14 The Finite Element method

Lab Objective: *The finite element method is commonly used for numerically solving partial differential equations. We introduce the finite element method via a simple BVP describing the steady state distribution of heat in a pipe as fluid flows through.*

Advection-Diffusion of Heat in a Fluid

We begin with the heat equation

$$y_t = \epsilon y_{xx} + f(x)$$

where $f(x)$ represents any heat sources in the system, and ϵy_{xx} models the diffusion of heat. We wish to study the distribution of heat in a fluid that is moving at some constant speed a. This can be modelled by adding an advection or transport term to the heat equation, giving us

$$y_t + a y_x = \epsilon y_{xx} + f(x).$$

We consider a fluid flowing through a pipe from $x = 0$ to $x = 1$ with speed $a = 1$, and as it travels it is warmed at a constant rate $f = 1$. We will impose the condition that $y = 2$ at $x = 0$, so that the fluid is already at a constant temperature as it enters the pipe.

These conditions yield

$$\begin{aligned} &y_t + y_x = \epsilon y_{xx} + 1, \quad 0 < x < 1, \\ &y(0) = 2 \end{aligned}$$

As time increases we expect the temperature of the fluid in the pipe to reach a steady state distribution, with $y_t = 0$. The heat distribution then satisfies

$$\begin{aligned} &\epsilon y'' - y' = -1, \quad 0 < x < 1, \\ &y(0) = 2. \end{aligned}$$

This problem is not fully defined, since it has only one boundary condition. Suppose a device is installed on the end of the pipe that nearly instantaneously brings the heat of the water up to $y = 4$. Physically we expect this extra heat that is introduced at $x = 1$ to diffuse backward through the water in the pipe. This leads to a well defined BVP,

$$\begin{aligned} &\epsilon y'' - y' = -1, \quad 0 < x < 1, \\ &y(0) = 2, \quad y(1) = 4. \end{aligned} \tag{14.1}$$

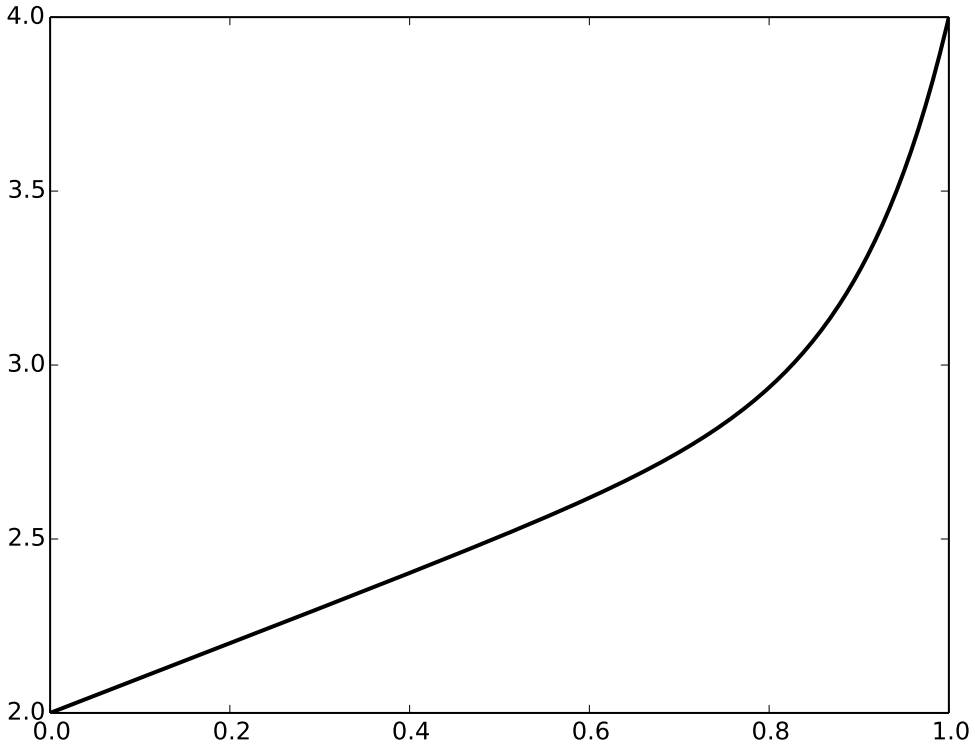

Figure 14.1: The solution of (14.1) for $\epsilon = .1$.

The Weak Formulation

Consider the equation

$$\begin{aligned} &\epsilon y'' - y' = f, \quad 0 < x < 1, \\ &y(0) = \alpha, \quad y(1) = \beta. \end{aligned} \tag{14.2}$$

To find the solution y using the finite element method, we reframe the problem and look at what is known as its weak formulation.

Let w be a smooth function on $[0, 1]$ satisfying $w(0) = w(1) = 0$. Multiplying (14.2) by w and integrating over $[0, 1]$ yields

$$\begin{aligned} \int_0^1 fw &= \int_0^1 \epsilon y''w - y'w, \\ &= \int_0^1 -\epsilon y'w' - y'w. \end{aligned}$$

Define a bilinear function a and a linear function l by

$$a(y,w) = \int_0^1 -\epsilon y'w' - y'w,$$
$$l(w) = \int_0^1 fw.$$

Rather than trying to solve (14.2), we instead consider the problem of finding a function y such that

$$a(y,w) = l(w), \quad \forall\, w \in V_0, \tag{14.3}$$

where V is some appropriate vector space that is expected to allow us to approximate the solution y, and $V_0 = \{w \in V | w(0) = w(1) = 0\}$. (For example, we could consider the space of functions that are piecewise linear with vertices at a fixed set of points. This example is discussed further below.) This equation is called the weak formulation of (14.2).

Let P_n be some partition of $[0,1]$, $0 = x_0 < x_1 < \ldots < x_n = 1$, and let V_n be the finite-dimensional vector space of continuous functions v on $[0,1]$ where v is linear on each subinterval $[x_j, x_{j+1}]$. These subintervals are the finite elements for which this method is named. V_n has dimension $n+1$, since there are $n+1$ degrees of freedom for continuous piecewise linear functions in V. Let V_{n0} be the subspace of V_n of dimension $n-1$ whose elements are zero at the endpoints of $[0,1]$, and let $\triangle x_n = \max_{0 \le j \le n-1} |x_{j+1} - x_j|$.

Let $\{\mathrm{P}_n\}$ be a sequence of partitions that are refinements of each other, such that $\triangle x_n \to 0$ as $n \to \infty$. Then in particular $V_1 \subset V_2 \subset \ldots \subset V_n \ldots \subset V$. For each partition P_n we can look for an approximation $y_n \in V_n$ for the true solution y; if this is done correctly then $y_n \to y$ as $n \to \infty$.

The Numerical Method

Consider a partition $\mathrm{P}_5 = \{x_0, x_1, \ldots, x_5\}$. We will define some basis functions ϕ_i, $i = 0, \ldots, 5$ for the corresponding vector space V_5. Let the ϕ_i be the hat functions

$$\phi_i(x) = \begin{cases} (x - x_{i-1})/h_i & \text{if } x \in [x_{i-1}, x_i] \\ (x_{i+1} - x)/h_{i+1} & \text{if } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

where $h_i = x_i - x_{i-1}$; see Figures 14.2 and 14.3.

We look for an approximation $\hat{y} = \sum_{i=0}^5 k_i \phi_i \in V_5$ of the true solution y; to do this we must determine appropriate values for the constants k_i. We impose the condition on $\hat{y}$ that

$$a(\hat{y}, w) = l(w) \quad \forall\, w \in V_5.$$

Equivalently, we require that

$$a\left(\sum_{i=0}^5 k_i \phi_i, \phi_j\right) = l(\phi_j) \quad \text{for } j = 1,2,3,4,$$

since $\phi_1, \phi_2, \phi_3, \phi_4$ form a basis for V_5.

Since a is bilinear, we obtain

$$\sum_{i=0}^5 k_i a(\phi_i, \phi_j) = l(\phi_j) \quad \text{for } j = 1,2,3,4.$$

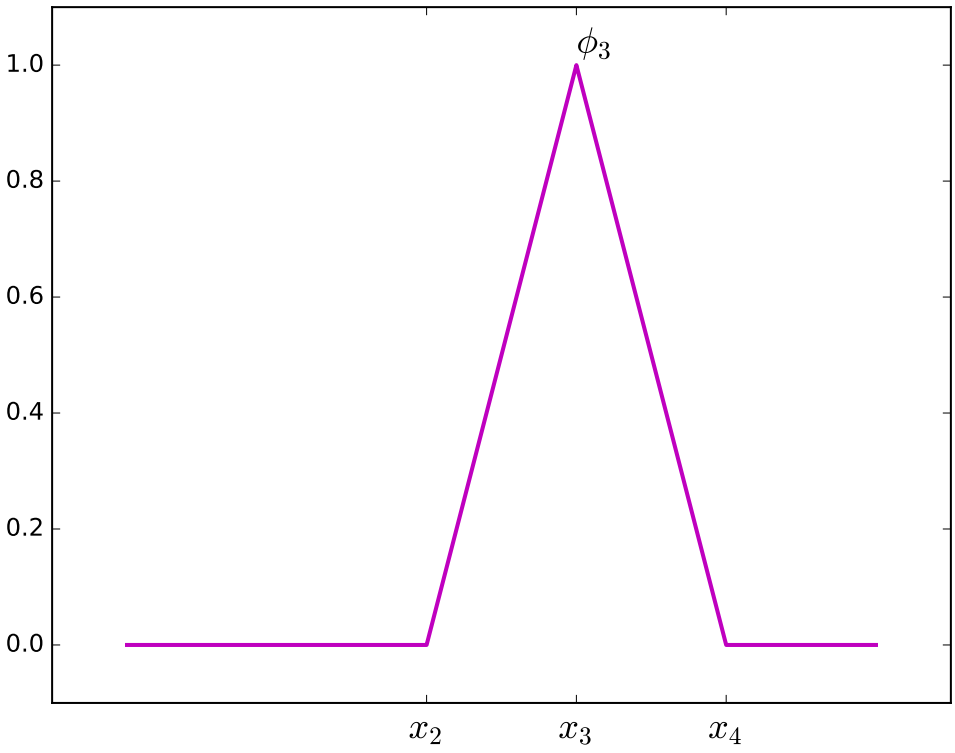

Figure 14.2: The basis function ϕ_3.

To satisfy the boundary conditions, we also require that $k_0 = \alpha$, $k_5 = \beta$. These equations can be written in matrix form as

$$AK = \Phi, \tag{14.4}$$

where

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ a(\phi_0,\phi_1) & a(\phi_1,\phi_1) & a(\phi_2,\phi_1) & 0 & 0 & 0 \\ 0 & a(\phi_1,\phi_2) & a(\phi_2,\phi_2) & a(\phi_3,\phi_2) & 0 & 0 \\ 0 & 0 & a(\phi_2,\phi_3) & a(\phi_3,\phi_3) & a(\phi_4,\phi_3) & 0 \\ 0 & 0 & 0 & a(\phi_3,\phi_4) & a(\phi_4,\phi_4) & a(\phi_5,\phi_4) \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$K = \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \end{bmatrix}, \quad \Phi = \begin{bmatrix} \alpha \\ l(\phi_1) \\ l(\phi_2) \\ l(\phi_3) \\ l(\phi_4) \\ \beta \end{bmatrix}.$$

Note that $a(\phi_i, \phi_j) = 0$ for most values of i, j (that is, when the hat functions do not have overlapping domains). Thus the finite element method results in a sparse linear system. To compute

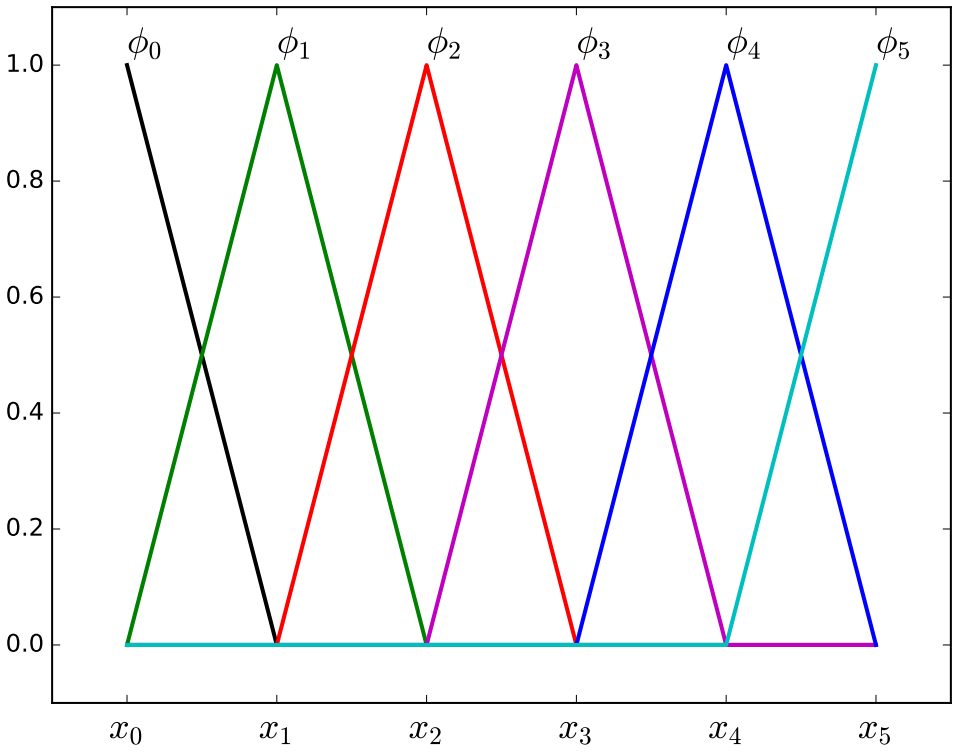

Figure 14.3: Basis functions for V_5.

the coefficients of (14.4) we begin by evaluating some integrals. Since

$$\phi_i'(x) = \begin{cases} 1/h_i & \text{for } x_{i-1} < x < x_i, \\ -1/h_{i+1} & \text{for } x_i < x < x_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

we obtain

$$\int_0^1 \phi_i'\phi_j' = \begin{cases} -1/h_{i+1} & \text{if } j = i+1, \\ 1/h_i + 1/h_{i+1} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases}$$

$$\int_0^1 \phi_i'\phi_j = \begin{cases} -1/2 & \text{if } j = i+1, \\ 1/2 & \text{if } j = i-1, \\ 0 & \text{otherwise,} \end{cases}$$

$$a(\phi_i, \phi_j) = \begin{cases} \epsilon/h_{i+1} + 1/2 & \text{if } j = i+1, \\ -\epsilon/h_i - \epsilon/h_{i+1} & \text{if } j = i, \\ \epsilon/h_i - 1/2 & \text{if } j = i-1, \\ 0 & \text{otherwise,} \end{cases}$$

$$l(\phi_j) = -(1/2)(h_j + h_{j+1}).$$

Equation (14.4) may now be solved using any standard linear solver. To handle the large number of elements required for Problem 3, you will want to use the tridiagonal algorithm provided in several of the earlier labs or the banded matrix solver included in `scipy.linalg`.

Problem 1. Use the finite element method to solve

$$\begin{aligned} \epsilon y'' - y' &= -1, \\ y(0) = \alpha, \quad y(1) &= \beta, \end{aligned} \tag{14.5}$$

where $\alpha = 2, \beta = 4$, and $\epsilon = 0.02$. Use $N = 100$ finite elements (101 grid points). Compare your solution with the analytic solution

$$y(x) = \alpha + x + (\beta - \alpha - 1)\frac{e^{x/\epsilon} - 1}{e^{1/\epsilon} - 1}.$$

Problem 2. One of the strengths of the finite element method is the ability to generate grids that better suit the problem. The solution of (14.5) changes most rapidly near $x = 1$. Compare the numerical solution when the grid points are unevenly spaced versus when the grid points are clustered in the area of greatest change; see Figure 14.4. Specifically, use the grid points defined by

```
even_grid = np.linspace(0,1,15)
clustered_grid = np.linspace(0,1,15)**(1./8)
```

Problem 3. Higher order methods promise faster convergence, but typically require more work to code. So why do we use them when a low order method will converge just as well, albeit with more grid points? The answer concerns the roundoff error associated with floating point

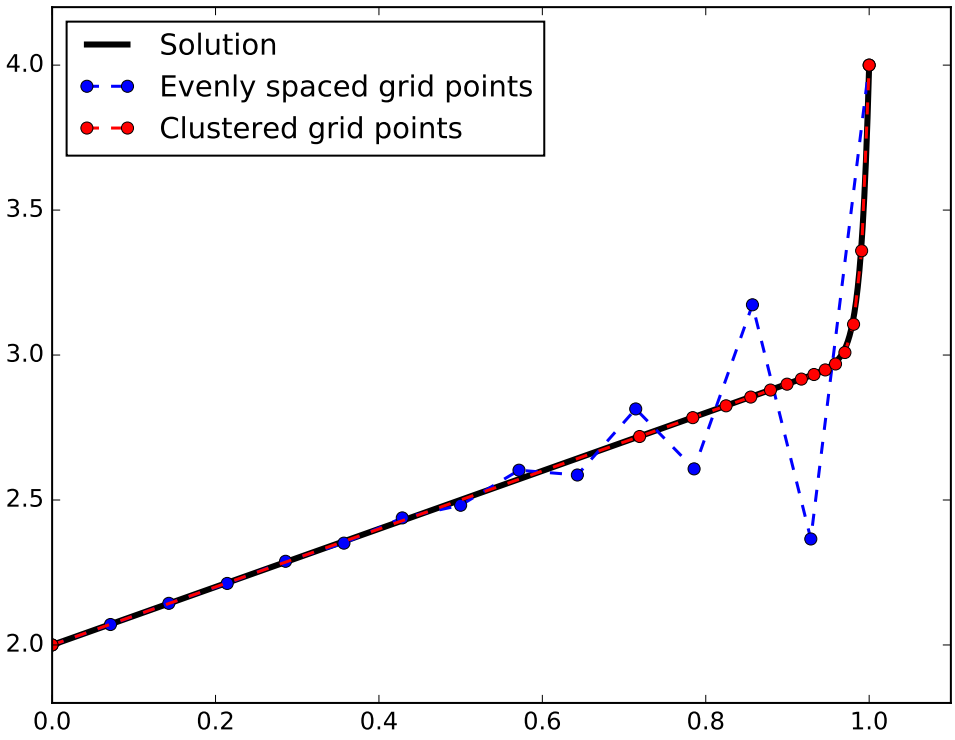

Figure 14.4: We plot two finite element approximations using 15 grid points.

arithmetic. Low order methods generally require more floating point operations, so roundoff error has a much greater effect.

The finite element method introduced here is a second order method, even though the approximate solution is piecewise linear. (To see this, note that if the grid points are evenly spaced, the matrix A in (14.4) is exactly the same as the matrix for the second order centered finite difference method.)

Solve (14.5) with the finite element method using $N = 2^i$ finite elements, $i = 4, 5, \ldots, 21$. Use a log-log plot to graph the error; see Figure 14.5.

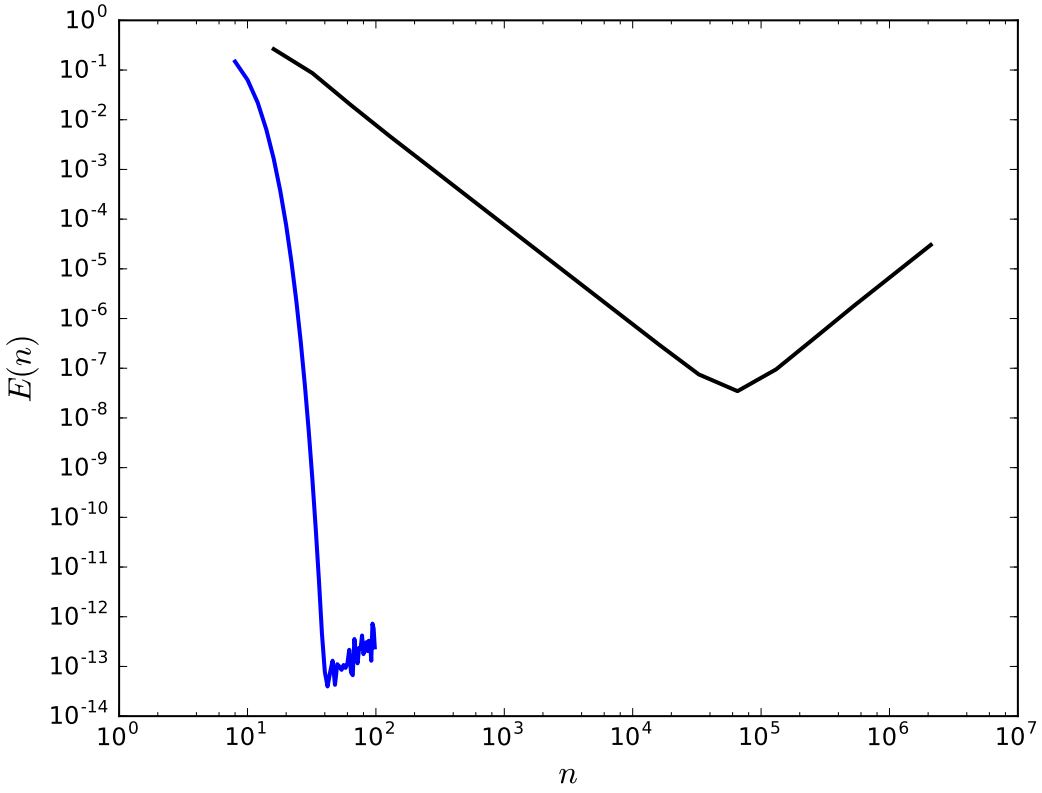

Figure 14.5: Error for the second order finite element method, as the number of subintervals n grows. Round-off error eventually overwhelms the approximation.