Chapter 1

Finite Element Methods for Elliptic Problems in 1D

1.1 Second-Order Problems

1.1.1 The Strong Form of the Boundary Value Problem

Linear, second-order elliptic differential equations in one dimension have the general form

$$-\big(k(x)\,u'(x)\big)' + b(x)\,u'(x) + c(x)\,u(x) = f(x)\ , \tag{1.1}$$

where x is the **independent variable**, typically a space variable, k, b, c and f are the **coefficients** of the equation, which may depend on x, and u is the relevant function being studied.

A differential equation is a condition for the function u that must be fulfilled at every point of the domain. At each $x \in \Omega$, the function u must **(a)** be smooth enough for all terms in the equation to be computable and **(b)** satisfy the algebraic equation (1.1). In this case, we say that the function u is a **solution** of the differential equation 1.1.

A multitude of physical problems can be modeled with Equation (1.1). We see some of them next, together with examples of solutions u to 1.1 in particular cases.

Examples:

1.1 Consider a vertical cylindrical column of uniform cross sectional area and height H, Young modulus $E(x)$ and density $\rho(x)$, x being the vertical coordinate. Then the vertical displacement $u(x)$ of the cross section at height x must satisfy the equilibrium equation

$$-\big(E(x)\,u'(x)\big)' = -\rho(x)\,g\ , \tag{1.2}$$

where g is the magnitude of the acceleration of gravity. Clearly, it reduces to (1.1) by taking $k = E$, $b = 0$, $c = 0$ and $f = -\rho\,g$. The vertical

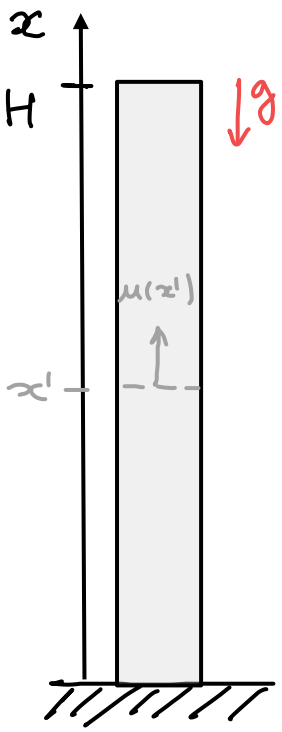


Figure 1.1 Sketch for Example 1.1.

stress (force per unit area) on the cross-section is given by

$$\sigma = E\,u' \,. \tag{1.3}$$

1.2 When heat is flowing through a wall, x being the through-the-wall coordinate, the temperature u obeys the **diffusion equation**

$$-\big(k(x)\,u'(x)\big)' = f(x)\,, \tag{1.4}$$

where $k(x)$ is the **thermal diffusivity** of the wall material at point x and f is a volumetric heat source (due for example to γ-radiation).

In the case of a homogeneous wall without volumetric sources, the temperature is affine in x. The temperature gradient, and thus also the **heat flux** $-ku'$, are constant.

1.3 **Steady-state convection-diffusion-reaction equation:** Let $u(x)$ denote $C(x) - C_{\text{eq}}$, where $C(x)$ the concentration of a species in a mobile one-dimensional reactive medium that moves with **velocity** $b(x)$ and C_{eq} is the equilibrium value. Then $u(x)$ satisfies (1.1), with k being the **diffusion coefficient**, c the **reaction coefficient** and f a possible volumetric source of the species.

In this case the first term is called **diffusion term** and consists of the spatial derivative of the **diffusive flux**

$$J_{\text{diff}} = -k\,u' \,.$$

It expresses the differential mass balance due to molecular diffusion. The second term is called **advection term** (sometimes **convection term**). It models the mass balance due to the transport of the species by the movement of the ambient medium. In fact, it is the *material time derivative* of the concentration in steady-state conditions (when the concentration does not depend on time), namely

$$\frac{Du}{Dt} = bu'.$$

In fact, the **advective flux** is

$$J_{\text{adv}} = b\,u \,.$$

The term $c(x)\,u(x)$ models the reaction of the species with the medium towards equilibrium. The coefficient c is proportional to the kinetic constant of the reaction. As c grows the local concentration is increasingly driven towards equilibrium and thus u gets closer to zero.

1.4 If we consider now a horizontal membrane under tension which is subject to vertical loads f, the vertical displacement u satisfies

$$-T\,u''(x) = f(x)\,, \tag{1.5}$$

where T is the **membrane tension**. This equation models the equilibrium of vertical forces on each arbitrary part of the membrane, which in this one-dimensional setting should better be visualized as an elastic string.

We hope the reader is by now convinced of the usefulness of models governed by (1.1). Of course the actual field $u(x)$ that arises for given coefficients (k, b, c and f) depends on the **boundary conditions**. In one dimension, the domain of analysis Ω is usually an interval, which we take as $0 < x < L$, i.e.,

$$\Omega = (0, L) .$$

In the physical systems considered in the previous examples, as in many other models of mathematical physics, the relevant field u arises as the solution of a **boundary value problem**. This means that *one* additional condition is imposed at $x = 0$ and *a second one* at $x = L$, and that these two conditions are necessary and sufficient to fully determine u. This is an intrinsic property of **second-order elliptic problems**: To fully define the solution u of the differential equation, it is necessary and sufficient to specify one condition at all points of the boundary. If there is some part of the boundary where no condition is specified, then there exist infinitely many solutions. In the one-dimensional setting considered here the boundary (denoted in general by $\partial\Omega$) consists of just the extreme points of the interval, i.e.

$$\partial\Omega = \{0, L\} .$$

The most popular boundary conditions are

- the **Dirichlet condition**, which imposes the value of u (for example, $u(0) = g_0$ or $u(L) = g_L$),
- and the **Neumann condition**, which imposes the value of u' (for example $u'(0) = d_0$ or $u'(L) = d_L$).

The problem in strong form that we introduce now has a Dirichlet condition at $x = 0$ and a Neumann condition at $x = L$. Other possibilities of boundary conditions will be discussed later.

Problem 1.1 (Strong Form)**.** *Given the coefficients k, b, c and f (as functions of x), together with the boundary constants g_0 and d_L, find a continuous function $u : \Omega \to \mathbb{R}$ satisfying*

$$-\big(k(x)u'(x)\big)' + b(x)u'(x) + c(x)u(x) = f(x) \qquad \forall\, x \in \Omega \tag{1.6a}$$

$$u(0) = g_0 \tag{1.6b}$$

$$u'(L) = d_L \tag{1.6c}$$

Examples:

1.5 Consider a purely diffusive ($b = c = 0$), homogeneous ($k(x) = k_0,\ \forall\, x$) case without source ($f = 0$). The constants g_0 and d_L remain arbitrary. Then the solution of the problem in strong form must be continuous and satisfy

$$-u''(x) = 0, \quad \forall x \in (0, L), \qquad u(0) = g_0, \qquad \text{and } \ u'(L) = d_L$$

Polynomials of degree ≤ 1, i.e., of the form $c_1 + c_2 x$, have vanishing second derivative in $(0, L)$, and they are continuous functions of x. They thus satisfy condition (a) and (b) above, and hence they are solution of the differential equation. Further, choosing $c_1 = g_0$ and $c_2 = d_L$ we identify the only polynomial solution to Problem 1.1, namely

$$u(x) = g_0 + d_L\, x\ .$$

Further, it is known that the *only* functions that have zero second derivative in an interval are polynomials of degree 1. Thus the function $u(x)$ above is the *unique* solution to Problem 1.1.

1.6 **Problems without solution.** Problem 1.1 does not always have a solution. Consider a case with constant diffusivity ($k = 1$), with no convection or source ($b = f = 0$) and with $c(x) = 1/x^2$. So, the equation reads

$$-u''(x) + \frac{u(x)}{x^2} = 0, \qquad \forall x \in (0, L)\ . \tag{1.7}$$

It can be checked by substitution that any function of the form

$$u(x) = c_1\, x^{(1+\sqrt{5})/2} + c_2\, x^{(1-\sqrt{5})/2}$$

satisfies (1.7). In fact, it is known that these functions are the *only* solutions of (1.7). However, notice that the exponent of x in the second term is *negative*, so that if $c_2 \neq 0$ the solution is $\pm\infty$ at $x = 0$ and thus different from g_0. So, for the problem to have a solution, $c_2 = 0$. But now, since the exponent of x in the first term is *positive*, the value of u at $x = 0$ is zero for any value of c_1! Unless the given value of g_0 is zero, there is no solution. Just for completeness, in the particular case $g_0 = 0$ there is indeed a unique solution, and the constant c_1 can be computed so that $u'(L) = d_L$.

1.7 **Deformed Column.** Consider again Example 1.1 of a vertical cylindrical column of uniform cross sectional area, height H, Young modulus $E(x)$ and density $\rho(x)$, with x being the vertical coordinate, see Fig. 1.1. Assume that the column is unloaded on its top face, and supported on a rigid foundation at its base. These define the boundary conditions of the problem by

$$u(0) = 0, \qquad \sigma(H) = E(H)\, u'(H) = 0,$$

which are of the Dirichlet and Neumann type, respectively. These equations, together with (1.2), define the strong form of this problem. To solve it, we integrate (1.3) over a slice of the column, from $x = h_1$ to $x = h_2$ to get

$$\sigma(h_1) - \sigma(h_2) = -g \int_{h_1}^{h_2} \rho(x)\, dx$$

so that the difference in σ between two positions equals the weight of the slice (per unit area). Since the column is unloaded on its top face, $\sigma(H) = 0$, and hence

$$\sigma(x) = -g \int_{x}^{H} \rho(\xi)\, d\xi.$$

Therefore, from (1.3), the displacement of the column follows as

$$u(x) - u(0) = \int_0^x \frac{\sigma(\xi)}{E(\xi)}\, d\xi = -\int_0^x \frac{g}{E(\xi)} \int_{\xi}^{H} \rho(y)\, dy. \tag{1.8}$$

Since the foundation is rigid, $u(0) = 0$. We can verify next that function $u(x)$ defined in (1.8) is a solution of the differential equation (1.2). First, we can compute $(E(x)u'(x))'$ for any point x by using the fundamental lemma of calculus, so it is smooth enough for all terms in the equation to be computable (condition (**a**) above), and these terms satisfy the algebraic equation (1.2) at any point x.
In the particular case in which $E(x) = E$ and $\rho(x) = \rho$, both constants through the length of the column, the solution u is

$$u(x) = -x(2H - x)\frac{g\rho}{2E}.$$

What about the solution of Problem 1.1? Does it exist at all? Is it unique? The answer to this question is derived from the general theory of linear ordinary differential equations, and the answer is **yes**, but of course conditional to some hypotheses. The hypotheses we consider here correspond to *elliptic* problems, and as we have seen allow us to model a wide variety of physical problems.

In general, sound physical models lead to well-posed mathematical problems, that is, problems for which a unique solution exists, and the solution changes smoothly when the coefficients of the equation or the boundary conditions do. However, theorems are helpful references to go to in case of doubt. We state here an existence and uniqueness theorem that covers most applications.

Theorem 1.1 (Existence and Uniqueness of Solutions)**.** *Assume that $k(x)$, $b(x)$, $c(x)$ and $f(x)$ are smooth and bounded, and also that $k(x) \geq k_0 > 0$. Further, let $c_0 = \min_x c(x)$ and assume that $c_0 \geq 0$. Then Problem 1.1 has a unique solution.*

This is more than what we need to know at this point about the strong form of the elliptic second-order boundary-value problem that we are set to analyze. The following sections will explain how a finite element method to calculate approximate solutions to Problem 1.1 is built.

1.1.2 The Strong Form with Robin Boundary Conditions*

Problem 1.2 (Strong Form (Robin conditions)). *Given the coefficients k, b, c and f (as functions of x), together with the boundary constants α_0, α_L, β_0 and β_L, find a continuous function $u : \Omega \to \mathbb{R}$ satisfying*

$$-\big(k(x)u'(x)\big)' + b(x)u'(x) + c(x)u(x) = f(x) \qquad \forall\, x \in \Omega \tag{1.9a}$$
$$u'(0) - \alpha_0\, u(0) = \beta_0 \tag{1.9b}$$
$$u'(L) + \alpha_L\, u(L) = \beta_L \tag{1.9c}$$

A Neumann condition at $x = 0$ corresponds to choosing $\alpha_0 = 0$ and $\beta_0 = d_0$, and similarly for $x = L$. Intuitively, a Dirichlet boundary condition with value g_0 at $x = 0$ is obtained by taking $\alpha_0 = M$ very large ($M \to +\infty$) and $\beta_0 = -Mg_0$. For the right boundary one must take $\alpha_L = M$ and $\beta_L = Mg_L$. Let us consider an example to help us understand the behavior of these boundary conditions.

Examples:

1.8 *The Dirichlet boundary condition as limit of the Robin condition.* Let us consider the boundary value problem corresponding to the diffusion equation,

$$-ku''(x) = 0 \qquad \forall\, x \in \Omega \tag{1.10a}$$
$$u'(0) = M\,(u(0) - g_0) \tag{1.10b}$$
$$u'(L) = -M\,(u(L) - g_L) \tag{1.10c}$$

which corresponds to Problem 1.2 with $k(x) = k$, $b = c = f = 0$, $\alpha_0 = \alpha_L = M$, $\beta_0 = -Mg_0$ and $\beta_L = Mg_L$. We expect that the exact solution satisfies $u(0) \to g_0$ and $u(L) \to g_L$ as $M \to +\infty$. Let us confirm this. Because u'' is zero everywhere, we know that

$$u(x) = u(0) + \frac{u(L) - u(0)}{L}\, x\,,$$

an affine function that is totally defined by its values at $x = 0$ and $x = L$. From this we compute $u'(x) = (u(L) - u(0))/L$ and insert into the boundary conditions to compute $u(0)$ and $u(L)$:

$$\frac{u(L) - u(0)}{L} = M(u(0) - g_0) \tag{1.11a}$$
$$\frac{u(L) - u(0)}{L} = -M(u(L) - g_L) \tag{1.11b}$$

which yields, if $M \neq 0$,

$$u(0) = g_0 + \frac{g_L - g_0}{ML + 2}\,, \qquad u(L) = g_L - \frac{g_L - g_0}{ML + 2},$$

and

$$u'(x) = \frac{ML}{ML+2}\,\frac{g_L - g_0}{L},$$

for all $x \in (0, L)$. If $M \gg 1/L$ (notice that M has units of length^{-1}) the Dirichlet conditions $u(0) = g_0$ and $u(L) = g_L$ are obtained in the limit.

We should mention here that if $M = 0$ (Neumann condition) the linear system is underdetermined, which is logical since only derivatives of u appear in the equation and boundary conditions and thus the solution is determined up to an additive constant. Also, if M is negative, and especially if $M < -2/L$, the condition is unphysical in the sense that there appear temperatures lower than $\min\{g_0, g_L\}$ and higher than $\max\{g_0, g_L\}$.

Physically, the problem (1.10) models heat transfer through a wall which is in contact with a fluid at temperature g_0 on the left side and with another fluid at temperature g_L on the right side. The heat flux from the left (resp., right) fluid to the wall is $h\,(g_0 - u(0))$ (resp., $h\,(g_L - u(L))$, i.e.,

$$-ku'(0) = h\,(g_0 - u(0)), \qquad ku'(L) = h\,(g_L - u(L))\,,$$

where h is called *heat transfer coefficient*. The corresponding value of M is thus h/k.

Our prototype Problem 1.2 thus allows us to consider a wide variety of physical situations with either Neumann, Robin or Dirichlet boundary conditions (in the latter case by taking, e.g., $M = 10^8/L$).

What about the solution of Problem 1.2? Does it exist at all? Is it unique? As mentioned earlier, in general sound physical models lead to well-posed mathematical problems. However, we will state an existence and uniqueness theorem that covers most applications.

Theorem 1.2 (Existence and Uniqueness of Solutions)**.** *Assume that L, $k(x)$, $b(x)$, $b'(x)$, $c(x)$ and $f(x)$ are bounded, and also that $k(x) \geq k_0 > 0$, $\alpha_0 \geq 0$ and $\alpha_L \geq 0$. Further, let $c_0 = \min_x\{c(x) + b'(x)\}$ and assume that*

a) *either $c_0 > 0$, or*

b) *$c_0 = 0$ and $\alpha_0 + \alpha_L > 0$ (i.e., they are not both zero),*

then Problem 1.2 has a unique solution. If neither (a) nor (b) above hold because $c(x) + b'(x) = 0$ for all x and $\alpha_0 = \alpha_L = 0$, but instead $\int_0^L f(x)\,dx - \beta_0\,k(0) + \beta_L\,k(L) = 0$, then the solution still exists but it is defined up to an additive constant.

This is more than what we need to know at this point about the strong form of the elliptic second-order boundary-value problem that we are set to analyze. In fact, in many situations one starts by assuming that an exact solution to the

problem of interest exists, and builds a Finite Element method to compute an approximation to the exact solution. Of course, if one proceeds in this way one must, at the time of analyzing the results, consider the possibility that maybe no solution exists at all.

The following sections will explain how the sentence "one builds a finite element method" is carried out for our problem of interest, i.e., Problem 1.1.

1.1.3 The Weak Form of the Boundary Value Problem

The finite element method is based on another way of stating Problem 1.1, called the *weak form* of the problem. We will discuss the origin of the name later. The weak form is not only a staple of finite element methods, but it is always a puzzling and welcome surprise to those who are introduced to it for the first time. Let's see what the weak form for Problem 1.1 looks like:

Problem 1.3 (Weak Form of Problem 1.1)**.** *Let*

$$\mathscr{S} = \left\{ w \colon \Omega \to \mathbb{R} \text{ smooth enough } \mid w(0) = g_0 \right\}, \tag{1.12a}$$

$$\mathscr{V} = \left\{ w \colon \Omega \to \mathbb{R} \text{ smooth enough } \mid w(0) = 0 \right\}. \tag{1.12b}$$

Find $u \in \mathscr{S}$ such that for any functions $v \in \mathscr{V}$

$$\int_\Omega \left[k(x)u'(x)v'(x) + b(x)u(x)'v(x) + c(x)u(x)v(x) \right] dx - k(L)d_L v(L) = \int_\Omega f(x)v(x)\, dx. \tag{1.13}$$

Remarkably, under very general conditions, solutions of the strong form of Problem 1.1 are solutions of the weak form of Problem 1.3, and conversely. We will discuss the potentially unfamiliar language we used in the statement a little bit later; let's focus on the main features first. The basic idea behind it is that we replace the requirement that equation (1.6a) should hold for all points in Ω, an infinite set of conditions, for the requirement that equation (1.13) should hold for *all* functions v in the set of functions $\mathscr{V}$, a different infinite set of conditions. Moreover, boundary condition (1.6c) also follows if (1.13) is satisfied for all $v \in \mathscr{V}$, while boundary condition (1.6b) is explicitly imposed by requiring $u \in \mathscr{T}$. So, how does this make sense? This is what we will discuss in the forthcoming paragraphs. First, however, we will exemplify how the weak form of the problem defines a solution.

Example 1.9 To illustrate how each function $v \in \mathscr{V}$ imposes a different condition on the solution u we are seeking, it is convenient to consider a simpler case first in which only a finite number of conditions are involved.

To this end, let $\Omega = (0,1)$, $k(x) = b(x) = c(x) = 1$ and $f(x) = -5\exp(-2x)$ for all $x \in \Omega$, $g_0 = 1$, and $d_L = -2\exp(-2)$. Equations (1.6) from the strong form

of Problem 1.1 become

$$-u''(x) + u'(x) + u(x) = -5\exp(-2x), \qquad \forall\, x \in \Omega, \tag{1.14a}$$
$$u(0) = 1, \tag{1.14b}$$
$$u'(1) = -2\exp(-2). \tag{1.14c}$$

The exact solution of this problem is $u(x) = \exp(-2x)$.

Additionally, we will replace the *infinite dimensional* sets $\mathcal{S}$ and $\mathcal{V}$ in (1.12) for *finite dimensional* ones as follows

$$\mathcal{S} = \left\{\exp(-2x) + c_1 N_1(x) + c_2 N_2(x) \mid c_1, c_2 \in \mathbb{R}\right\} \tag{1.15a}$$
$$\mathcal{V} = \left\{d_1 N_1(x) + d_2 N_2(x) \mid d_1, d_2 \in \mathbb{R}\right\}, \tag{1.15b}$$

where $N_1(x) = x$ and $N_2(x) = x^2$ are two functions over Ω. If you are unfamiliar with the set notation adopted in these equation, please read § 1.1.5. For example, the function $w_1(x) = \exp(-2x) + 2x \in \mathcal{S}$ by selecting $c_1 = 2$ and $c_2 = 0$, and the function $w_2(x) = 3x + 2x^2 \in \mathcal{V}$ by selecting $d_1 = 3$ and $d_2 = 2$. Additionally, $\exp(-2x) \in \mathcal{S}$, so the exact solution of the problem belongs to the set $\mathcal{S}$. Notice that selecting a function in $\mathcal{S}$ requires only specifying two values, c_1 and c_2, as it is for functions in $\mathcal{V}$ with d_1 and d_2. Because N_1 and N_2 are equal to zero at $x = 0$, functions in $\mathcal{S}$ attain the value 1 there and automatically satisfy boundary condition (1.14a), while at the same location functions in $\mathcal{V}$ are equal to zero, as in the sets in (1.12).

Under these conditions, the weak form is: *Find $u \in \mathcal{S}$ such that for any function $v \in \mathcal{V}$*

$$\int_0^1 u'(x)v'(x) + u'(x)v(x) + u(x)v(x)\,dx + 2\exp(-2)v(1) = -\int_0^1 5\exp(-2x)v(x)\,dx. \tag{1.16}$$

Finding u from the weak form is equivalent to determining the values of c_1 and c_2 that define it. How do we go about it? In other words, how does the weak form define u?

The answer is by *testing* with functions in $\mathcal{V}$. We can state that since (1.16) should be true for any $v \in \mathcal{V}$, it should be true for N_1 and for N_2 in particular. Functions N_1 and N_2 each impose a condition that u should satisfy, to wit:

$$\int_0^1 u'(x) + u'(x)x + u(x)x\,dx + 2\exp(-2) = -\int_0^1 5\exp(-2x)x\,dx, \qquad \text{(when } v = N_1\text{)} \tag{1.17}$$

$$\int_0^1 u'(x)2x + u'(x)x^2 + u(x)x^2\,dx + 2\exp(-2) = -\int_0^1 5\exp(-2x)x^2\,dx. \qquad \text{(when } v = N_2\text{)}$$

Since u is defined by the values of c_1 and c_2, these two equations could suffice to uniquely determine their values. To see this, we can substitute

with $u(x) = \exp(-2x) + c_1 x + c_2(1-x)$ and $u'(x) = -2\exp(-2x) + c_1 - c_2$, compute all integrals and simplify, to get

$$\begin{aligned} 11c_1 - 8c_2 &= 0, \\ 19c_1 - 15c_2 &= 0, \end{aligned}$$

whose sole solution is $c_1 = c_2 = 0$. Hence, the solution of this weak form is the exact solution $u = \exp(-2x)$.

To conclude the example, we note that to uniquely identify u it was enough to choose two out of the many functions in $\mathcal{V}$. We will have a chance to discuss more about this soon, and in particular, we will see that testing with only two functions, in this case N_1 and N_2, is enough to guarantee that (1.16) is satisfied for *any* $v \in \mathcal{V}$. This example also illustrates how each function $v \in \mathcal{V}$ imposes a condition that u needs to satisfy, and that u is well defined if enough conditions are imposed. On the other hand, not all conditions are independent, as the need to test with only two functions illustrates.

The Rationale Behind the Weak Form. Why does the weak form work, and how do we know what the weak form is? These are the questions we answer next. We first illustrate with an example how to obtain the weak form of a problem, and then discuss why and when they are equivalent. We expand on how to obtain the weak form more generally in the next section.

For this example we will consider obtaining the weak form for Problem 1.1. However, to begin with the simplest possible case, we will set $k(x) = 1$, $b(x) = c(x) = 0$ for all $x \in \Omega = (0, L)$, so that

$$-u''(x) = f(x) \qquad x \in \Omega, \tag{1.18a}$$

$$u(0) = g_0, \tag{1.18b}$$

$$u'(L) = d_L. \tag{1.18c}$$

To formulate the weak form, we begin by introducing two different sets of functions, the **trial space** $\mathcal{S}$ and the **test space** $\mathcal{V}$. For this problem, these are defined next.

1. We restrict the set of functions u among which we seek a solution: we search for u in the *trial space*

$$\mathcal{S} = \{u \colon \Omega \to \mathbb{R} \text{ smooth} \mid u(0) = g_0\}. \tag{1.19}$$

2. We restrict the set of functions v that impose conditions on u: we consider v in the *test space*

$$\mathcal{V} = \{v \colon \Omega \to \mathbb{R} \text{ smooth} \mid v(0) = 0\}. \tag{1.20}$$

Functions in the trial space are called **trial functions**, and those in the test space are called **test functions**. At this point it would be good to say a little more about what we mean by a **smooth function**. The word smooth indicates that functions in the set should have as many derivatives in Ω as required by the problem or the manipulation of expressions we perform (see also Example 1.17 in §1.1.5); we will have the opportunity to discuss some of these requirements later. The intention here is to focus on the essential aspects of the ideas we are introducing at this point, postponing the discussion on some of the details later.

The weak form of the problem is obtained from the strong form following three steps:

1. Multiply the strong form by an arbitrary function $v \in \mathcal{V}$, and integrate over the interval $[0, L]$ to obtain

$$0 = u''(x)\,v(x) + f(x)\,v(x) \Rightarrow 0 = \int_0^L u''(x)\,v(x) + f(x)\,v(x)\,dx$$

2. Integrate by parts[1] the second derivative of $u''(x)$, to pass the derivative to v, to get:

$$0 = u'(L)\,v(L) - u'(0)\,v(0) - \int_0^L u'(x)\,v'(x)\,dx + \int_0^L f(x)\,v(x)\,dx. \tag{1.21}$$

3. Use boundary condition (1.18c) to replace the value of $u'(L) = d_L$ in (1.21) and the fact that $v(0) = 0$, namely,

$$0 = \underbrace{u'(L)}_{=d_L,\text{ due to (1.18}b\text{)}}\, v(L) - u'(0)\, \underbrace{v(0)}_{=0,\text{ due to } v\in\mathcal{V}} - \int_0^L u'(x)\,v'(x)\,dx + \int_0^L f(x)\,v(x)\,dx, \tag{1.22}$$

from where it follows that

$$\int_0^L u'(x)\,v'(x)\,dx - d_L\,v(L) = \int_0^L f(x)\,v(x)\,dx. \tag{1.23}$$

Thus, we have proved that if u satisfies (1.18), the strong form of the simplest possible case, then it satisfies that (1.23) is true for any $v \in \mathcal{V}$.

The weak form of the problem can then be stated as:

$$\textit{Find } u \in \mathcal{S} \textit{ such that (1.23) holds for any } v \in \mathcal{V}. \tag{1.24}$$

[1] For two smooth functions w and v, $(wv)' = w'v + wv'$ by the product formula, and integrating on both sides we get the integration by parts formula:

$$\lim_{x\to b^-} w(x)\,v(x) - \lim_{x\to a^+} w(x)\,v(x) = \int_a^b w'(x)\,v(x)\,dx + \int_a^b w(x)\,v'(x)\,dx.$$

We use it by setting $w = u'$.

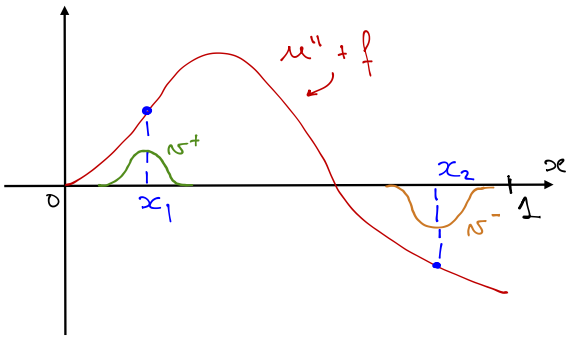


Figure 1.2 Potential choice of weighting functions to show that a weak solution is a strong solution.

If u satisfies the weak form of the problem, it is called **a weak solution**. Equation (1.23) is often called a **variational equation**, parallel to the denomination of (1.18a) as a differential equation.

Notice that, by integrating by parts, we reduced the number of derivatives of u involved in the variational equation relative to the number involved in the strong form. So, the conditions for a function to satisfy the weak form are *weaker* than for the strong form. This means that, potentially, functions that have fewer derivatives than those needed in the strong form could be solutions of the weak form (a *weak solution*), but for such functions the strong form may not make sense (not a *strong solution*). These are situations in which the weak form may have a solution that is not solution of the strong form. However, and as aforementioned, under very general conditions, a weak solution *is* a strong solution. In the context of our discussions, we will not make a distinction between the two. These weaker conditions imposed on the solution is what gives rise to the name, the *weak form*. For the same reasons, Problem 1.1 is called the *strong form* of the problem.

☞ The space $\mathcal{V}$ specified in (1.15) is an example of a *finite dimensional vector space* of functions, since only two scalar quantities (d_1, d_2) are needed to specify a function in it. Instead, the space $\mathcal{V}$ in (1.20) is an **infinite dimensional vector** space.

Why and when are the strong and weak from equivalent? As aforementioned, the weak form is characterized by the fact that we replaced a condition or equality at each $x \in (0, L)$ in (1.18a) for a condition or equality for each $v \in \mathcal{V}$ in (1.23). Each $v \in \mathcal{V}$ provides a new equation that u needs to satisfy. In the same way that there is an infinite number of points in $(0, L)$, there generally is an infinite number of functions in $\mathcal{V}$.

This observation intuitively suggests why the weak form of the problem works. We just saw that a strong solution is a weak solution, and based on the above observation, we may also ask the converse question: is a weak solution a strong solution? In other words, are the strong and weak forms of the problem equivalent? It turns out that, under very general conditions, the answer to these two questions is "yes:" a weak solution is a strong solution, and one problem (or form) is a restatement of the other.

We discuss a formal[2] argument for this next. Assume that $u \in \mathscr{S}$ is a weak solution. Then, integrating by parts the left hand side of the variational equation (1.23) to remove the derivative of v', we get

$$0 = (u'(L) - d_L)\, v(L) - u'(0)\, v(0) - \int_0^L (u''(x) + f(x))\, v(x)\, dx. \tag{1.25}$$

Next, we use that we can choose any $v \in \mathscr{V}$. Referring to Fig. 1.2, we proceed by contradiction and assume that $u''(x_1) + f(x_1) > 0$ for some $x_1 \in (0, L)$. Then we can choose $v^+ \in \mathscr{V}$ as sketched in the figure[3]. In this case, using that $v^+(L) = v^+(0) = 0$, (1.25) reads

$$0 = -\int_0^L (u''(x) + f(x))\, v^+(x)\, dx. \tag{1.26}$$

But $v^+(x)$ is not negative anywhere, it is zero wherever $u''(x) + f(x)$ is negative, and both $u''(x) + f(x)$ and v^+ are positive in a neighborhood of x_1, so the integral in the right hand side of (1.26) needs to be positive, a contradiction. We conclude then that $u''(x) + f(x)$ cannot be positive anywhere. A similar argument can be made around a point in which $u'' + f$ is negative by selecting a weighting function v^-, as in the figure. We can then conclude that $u'' + f$ is neither positive nor negative anywhere in $(0, L)$, so $u''(x) + f(x) = 0$ for any $x \in (0, L)$. The weak solution satisfies (1.18a) then.

An important detail that often gets lost in a first view of the last argument is that each one of the different v's described above imposes a different condition on u, and that the only way for u to satisfy them all is by satisfying the differential equation (1.18a). Therefore, even though we may not have considered *all* functions $v \in \mathscr{V}$, we considered enough of them to conclude that u satisfies the differential equation.

☞ Often the following question is asked: "Wait... shouldn't u satisfy (1.18a) *only* when we consider those functions $v \in \mathscr{V}$ in Fig. 1.2? This question does not merit further consideration after realizing that whether u satisfies the differential equation or not depends on u only, and is independent of v.

And what about the boundary conditions at $x = 0, L$? How are they imposed in the weak form? The condition $u(0) = g_0$ is explicitly required from u in the definition of $\mathscr{S}$. We show what happens with the condition at $x = L$ next.

Since we just concluded that the weak solution u satisfies (1.18a), and $v \in \mathscr{V}$, the variational equation reads

$$0 = (u'(L) - d_L)\, v(L) - u'(0) \underbrace{v(0)}_{=0} - \int_0^L \underbrace{(u''(x) + f(x))}_{=0}\, v(x)\, dx = (u'(L) - d_L)\, v(L)$$

for *any* $v \in \mathscr{V}$. In this case we can choose any $v \in V$ that satisfies $v(L) \neq 0$. This implies that we need $u'(L) - d_L = 0$. So, the weak solution satisfies (1.18c).

[2] "Formal" in this context means that it is not mathematically rigorous.

[3] For example, we can set

$$v^+(x) = \begin{cases} 0 & \text{if } |x - x_1| > \epsilon \\ \left[1 - \left(\frac{x - x_1}{\epsilon}\right)^2\right]^3 & \text{if } |x - x_1| \leq \epsilon, \end{cases}$$

where $\epsilon > 0$ is the "half-width" of v^+, and can be chosen as small as desired.

Boundary conditions that are obtained from the weak form are called **natural boundary conditions (NBC)**. Instead, boundary conditions that need to be explicitly requested in the definition of the trial space $\mathcal{S}$ are labeled **essential boundary conditions (EBC)**. In our example, the boundary condition $u'(L) = d_L$ is a natural boundary condition, and boundary condition $u(0) = g_0$ is an essential boundary condition.

As a result of this discussion, we can conclude that the weak solution u satisfies all the equations in (1.18), so it is a strong solution.

☞ It emerges from the discussion here that the set $\mathcal{V}$ should at least contain all functions v^+ and v^- for this argument to be made; the precise conditions for what functions $\mathcal{V}$ should include take the form of the "Fundamental lemma of the calculus of variations," see e.g. [3].

Finally, this discussion also explains why the trial and test spaces need to be different. The boundary condition at $x = 0$ does not emanate from the weak form[4], so it needs to be required from the candidate solutions to the problem in $\mathcal{S}$. Had the condition $v(0) = 0$ not been required from functions in $\mathcal{V}$, we would have concluded that the boundary condition $u'(0) = 0$ was needed, by a similar argument to that we made for the boundary condition at $x = L$.

Exercises:

1. Verify that the strong solution $u(x) = x/L - (x/L)^2/2$ is a weak solution when $f(x) = 1$.

1.1.4 The Weak Form of the Model Problem*

In the following we will formalize the path we took in the last section to go from a strong form to a weak form, and back. We will proceed again in a formal way, which means that we will do it without fixing many of the details.

From the strong to the weak form. As a way of illustrating with an example as we discuss the process, we will consider the strong form of Problem 1.1.

To obtain the weak form, we proceed in a series of steps:

1. **Form the residual:** Begin by forming the residual of (1.6a): subtract the right hand side from the left hand side of the equality (or vice versa). That is, for a function u we define a function $r : [0, L] \to \mathbb{R}$ as

$$r = -(k(x)u'(x))' + b(x)u'(x) + c(x)u(x) - f(x). \tag{1.27}$$

   Then, according to (1.6a), for the solution u of the strong form we should have

$$r(x) = 0 \qquad x \in (0, L). \tag{1.28}$$

2. **Multiply by a test function and integrate.** We then proceed and multiply this equation by a function $v \in \mathcal{V}$ and integrate over $(0, L)$, where $\mathcal{V}$ is some set of smooth functions over $(0, L)$ that we shall specify later. As

[4] There is indeed a weak form in which this is possible, which we do not discuss here.

aforementioned, functions $v \in \mathcal{V}$ are called *test* functions, but are also labeled **weight** functions. For any such $v \in \mathcal{V}$, we have

$$\int_0^L r(x)\, v(x)\, dx = 0. \tag{1.29}$$

Again, we are replacing the requirement $r(x) = 0$ for all $x \in (0, L)$, for (1.29) to be satisfied for all functions in $\mathcal{V}$. Because the residual functions are multiplied by the weight functions, this form of formulating the problem is also called the *Method of Weighted Residuals* (MWR) [2].

In our example, this means:

$$\int_0^L \left(-(k(x)u'(x))' + b(x)u'(x) + c(x)u(x) - f(x)\right) v\, dx = 0 \tag{1.30}$$

for all $v \in \mathcal{V}$.

3. **Integrate the residual by parts.** Assuming that both u and v are smooth enough for the integration by parts formula to hold, integrate the residual by parts $m \geq 0$ times, with m less or equal than the maximum number of derivatives of u in the residual. For each value of m a *different* variational equation and hence weak form is obtained. Typically, we integrate by parts until the number of derivatives of u is equal or only one higher than the number of derivatives of v in each term. In other words, use integration by parts to "transfer" derivatives from u to v, until u has either an equal number of or one more derivative than v in any term in the resulting expression. In our example, $m = 1$, and this leads to

$$\int_0^L k(x)u'(x)v'(x) + b(x)u'(x)v(x) + c(x)u(x)v(x) - f(x)v(x)\, dx \\ - k(L)u'(L)v(L) + k(0)u'(0)v(0) = 0 \tag{1.31}$$

4. **Use boundary conditions and identify conditions for $\mathcal{V}$.** Out of the boundary terms that appear from integrating by parts, identify those for which the value has been provided or can be solved for from the boundary conditions, and replace them in the boundary terms. As with the integration by parts, there could be some ambiguity here, leading to different weak formulations. In this case, (1.6c) gives the value of $u'(L)$. Replacing in our example,

$$\int_0^L k(x)u'(x)v'(x) + b(x)u'(x)v(x) + c(x)u(x)v(x) - f(x)v(x)\, dx \\ - k(L)d_L v(L) + k(0)u'(0)v(0) = 0 \tag{1.32}$$

However, we do not know anything about the value of $u'(0)$; we only know about $u(0)$. In this case, we request the value of the accompanying test function to be zero through the definition of $\mathcal{V}$. In a general case, we proceed similarly with any other boundary term for which we do not have any

boundary condition, and require the value of the accompanying (derivative of the) test function to be zero in the definition of $\mathcal{V}$. This is the process to identify condition that we need to impose for functions in $\mathcal{V}$.

In our example, we are going to request that if $v \in \mathcal{V}$, then $v(0) = 0$. For any such v,

$$\int_0^L k(x)u'(x)v'(x) + b(x)u'(x)v(x) + c(x)u(x)v(x) - f(x)v(x)\,dx - k(L)d_L v(L) = 0 \quad (1.33)$$

Those boundary conditions that we are able to incorporate into the variational equation by replacing some of the boundary terms are the natural boundary conditions for the problem, since as we will see shortly, they will be enforced by the weak form. The remaining boundary conditions need to be explicitly imposed and requested as part of the definition of $\mathcal{S}$; they are the essential boundary conditions. Hence, for our example, boundary condition (1.18b) needs to be requested as part of the conditions for a function to belong to the trial space $\mathcal{S}$.

If we do not request $v(0) = 0$ in the definition of $\mathcal{V}$, we would arrive to a *different* weak form, still satisfied by the exact solution. This is an alternative weak form, just as we obtain a different weak from for each number of derivatives we decide to transfer from u to v. We would still need to request the essential boundary condition as part of the definition of $\mathcal{S}$.

5. **Formulate the weak form.** Define a set of smooth enough trial functions $\mathcal{S}$ in which to seek u that incorporates the boundary conditions *not yet* used in the weak form. Define also the set of test functions $\mathcal{V}$ by using the boundary conditions we unearthed for v in the last step. We then formulate the weak form by requesting to seek a function in the trial space such that the variational equation holds for any test function. For our example, this leads to Problem 1.3.

As remarked in step 3, it would also be possible to choose a different distribution of derivatives between u and v than the guideline provided above, leading to a different type of weak form. For example, we could transfer all derivatives to v, or leave all derivatives in u. In Finite Element Analysis, we are generally interested in weak forms with minimal smoothness requirements for u and v, since it is simpler to build spaces to approximate solutions in this case.

Example 1.10 Weak form for a third-order problem. Consider the problem

whose strong form is: Given $f\colon [a,b] \to \mathbb{R}$, find $u\colon [a,b] \to \mathbb{R}$ such that

$$u_{,xxx} = f \qquad x \in (a,b) \tag{1.34a}$$
$$u(a) = 1 \tag{1.34b}$$
$$u_{,x}(b) = 2 \tag{1.34c}$$
$$u_{,xx}(a) = 3. \tag{1.34d}$$

Here the notation $u_{,x}$ denotes the derivative, $u_{,xx}$ the second derivative, etc. The exact solution of this problem is obtained by repeated integration of (1.34a):

$$\begin{aligned}
\int_a^x f(y)\,dy &= \int_a^x u_{,yyy}(y)\,dy \\
&= u_{,xx}(x) - u_{,xx}(a) = u_{,xx}(x) - 3 \\
\int_b^x \left[\int_a^z f(y)\,dy\right] dz &= \int_b^x u_{,zz}(z) - 3\,dz \\
&= u_{,x}(x) - u_{,x}(b) - 3(x-b) \\
&= u_{,x}(x) - 2 - 3(x-b) \\
\int_a^x \left[\int_b^w \left[\int_a^z f(y)\,dy\right] dz\right] dw &= \int_a^x u_{,w}(w) - 2 - 3(w-b)\,dw \\
&= u(x) - \underbrace{u(a)}_{=1} - 2(x-a) - \frac{3}{2}(x^2-a^2) \\
&\quad + 3b(x-a).
\end{aligned}$$

Therefore,

$$u(x) = 1 + (2-3b)(x-a) + \frac{3}{2}(x^2-a^2) + \int_a^x \left[\int_b^w \left[\int_a^z f(y)\,dy\right] dz\right] dw. \tag{1.35}$$

To identify the weak form, we proceed as above:

(a) *Form the residual*:

$$r = u_{,xxx} - f.$$

(b) *Multiply by a test function and integrate*:

$$\int_a^b (u_{,xxx} - f)\,v\,dx = 0$$

for all v that is smooth enough.

(c) *Integrate the residual by parts*: In this case, we will integrate by parts only once,

$$u_{,xx}(b)\,v(b) - u_{,xx}(a)\,v(a) - \int_a^b u_{,xx}\,v_{,x} + f\,v\,dx = 0$$

for all v that is smooth enough.

(d) *Use boundary conditions and identify conditions for $\mathcal{V}$*: Here we know the value of $u_{,xx}(a)$, so we need to request $v(b) = 0$. For such v we have

$$-3v(a) - \int_a^b u_{,xx}\, v_{,x} + f v \, dx = 0$$

Then, $u_{,xx}(a) = 3$ is a natural boundary condition.

(e) *Formulate the weak form*: The essential boundary conditions are then $u(a) = 1$ and $u_{,x}(b) = 2$. Let then

$$\mathcal{S} = \{u\colon [a,b] \to \mathbb{R} \text{ smooth enough } \mid u(a) = 1, u_{,x}(b) = 2\},$$
$$\mathcal{V} = \{v\colon [a,b] \to \mathbb{R} \text{ smooth enough } \mid v(b) = 0\}.$$

The weak form of the problem is then: Find $u \in \mathcal{S}$ such that for all $v \in \mathcal{V}$

$$-\int_a^b u_{,xx}\, v_{,x}\, dx = \int_a^b f v \, dx - 3v(a).$$

From the weak to the strong form. We now illustrate the process of obtaining the strong form of a problem starting from the weak form. We do this beginning from Problem 1.3, and would like to show that this implies that its solution should satisfy the strong form, Problem 1.1.

We proceed as follows:

1. **Integrate the variational equation by parts to eliminate all derivatives from the test function.** In this case, we want to eliminate all derivatives that appear on v, so we integrate by parts as many times as needed to do that. For our example in (1.13),

$$\begin{aligned} 0 &= \int_0^L k(x)u'(x)v'(x) + b(x)u'(x)v(x) + c(x)u(x)v(x) - f(x)v(x)\, dx \\ &\quad - k(L)d_L v(L) \\ &= \int_0^L -k(x)u''(x)v(x) + b(x)u'(x)v(x) + c(x)u(x)v(x) - f(x)v(x)\, dx \\ &\quad + (k(L)u'(L) - k(L)d_L)v(L) - k(0)u'(0)v(0) \end{aligned}$$

2. **Group terms with v at the same location, and use conditions in $\mathcal{V}$.** A number of boundary terms will appear as a result of the integration by parts. Since functions in $\mathcal{V}$ often need to satisfy conditions at the boundary, use such conditions at this point. Then, gather all terms that involve the same values of the test function v, or its derivatives, in the domain and at the boundary. For our example, $v(0) = 0$, and we collect the terms containing $v(L)$ and $v(x)$:

$$\begin{aligned} 0 = \int_0^L \left[-k(x)u''(x) + b(x)u'(x) + c(x)u(x) - f(x)\right] v(x)\, dx \\ + k(L)(u'(L) - d_L)v(L). \end{aligned} \quad (1.36)$$

3. **Obtain the differential equation and potential boundary conditions.** We use the fact that the resulting expression should be valid for any $v \in \mathcal{V}$. Again, appealing to a simple and formal argument, this means that every term that multiplies a test function v at some point x should be equal to zero at that point, since the value of $v(x)$ can be chosen arbitrarily. For our example, this means that

$$0 = -k(x)u''(x) + b(x)u'(x) + c(x)u(x) - f(x) \qquad x \in (0, L) \tag{1.37a}$$
$$0 = k(L)(u'(L) - d_L), \tag{1.37b}$$

since they multiply $v(x)$ for $x \in (0, L)$ and $v(L)$, respectively.

The set of equations (1.37) are called the *Euler-Lagrange* equations of the weak form, or in the context of the Calculus of Variations, of the variational principle. Notice that the natural boundary conditions have been recovered as one of the Euler-Lagrange equations of the weak form.

4. **Formulate the strong form.** Gather the Euler-Lagrange equations and the essential boundary conditons and state the strong form. For our example, these are equations (1.18).

The procedure here justifies why, in building the weak form from a strong form, we require v or its derivative to be zero on the boundary terms for which we do not have a prescribed boundary condition, and why we request the remaing (essential) boundary conditions in $\mathcal{S}$.

1.1.5 Sets of Functions*

The weak form of a problem involves describing a set of functions for which we require an equation such as (1.13) to hold (the statement "for all continuous functions v"). Let's describe some common sets of functions and the notation we use to specify them. We proceed by examining some examples.

Examples:

1.11 The set $C^0(I)$ is the set of continuous scalar(real)-valued functions over the interval $I \subset \mathbb{R}$. For example:

i. Let $f(x) = \sin x$. Then, if $I = [0, 1]$ we have that $f \in C^0([0, 1])$, since $\sin x$ assigns a real value to each point in the interval $[0, 1]$, and f is continuous over that interval. Moreover, we have that $f \in C^0(\mathbb{R})$, in which we set $I = \mathbb{R}$, since $\sin x$ is continuous over the entire real line.

ii. The *Heaviside step function* is defined as

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0, \end{cases}$$

for $x \in \mathbb{R}$. Then, $H \in C^0((0,1))$ because it is continuous in the open interval $(0,1)$, but $H \notin C^0([-1,1])$, because H is discontinuous at $x = 0$.

1.12 The set $C^k(I)$, for $k \in \mathbb{N}$, is the set of continuous functions with k continuous derivatives over the interval $I \subset \mathbb{R}$.
The set $C^\infty(I)$ is the set of functions in which *all* derivatives are continuous. For example:

i. Let $f(x) = \cos x$. Then, $f \in C^2(\mathbb{R})$ since it has two continuous derivatives anywhere in the real line. Moreover, $f \in C^\infty(\mathbb{R})$, since all derivatives of f are continuous.

ii. Let $g(x) = |x|$, the absolute value function, whose derivative exists anywhere but at $x = 0$. Then, $g \in C^2((0,1])$, $g \in C^0([-1,1])$, but $g \notin C^1([-1,1])$ since g' is discontinuous at $x = 0$, i.e., $\lim_{x\to 0^-} g'(x) \neq \lim_{x\to 0^+} g'(x)$.

1.13 The set $\mathbb{P}_k(I)$, for $k \in \mathbb{N} \cup \{0\}$, is the set of all polynomials of degree less or equal than k over the interval $I \subset \mathbb{R}$. For example:

i. Let $f(x) = x^3 + 1$, then $f \in \mathbb{P}_k(\mathbb{R})$ for any $k \geq 3$.

ii. Let $f(x) = (x-2)^{10}$, then $f \in \mathbb{P}_{10}([0,1])$ for any $k \geq 10$.

A set of functions often contains an infinite number of functions, and it is impossible to enumerate all members of the set. Nevertheless, it is possible to test whether a given function belongs to the set, as we did in the above examples. Additionally, sets of functions are often defined by imposing additional conditions for a function to belong to a set. For example, we could define a set by writing

$$V_1 = \left\{ f \in C^0([0,1]) \mid f(0) = 2 \right\},$$

which indicates the set of all continuous functions over the interval $[0,1]$ whose value at $x = 0$ is 2.

We introduced new notation here, which we proceed to explain: The curly brackets $\{\cdot\}$ indicate that what is inside describes the members of the set, and the separator "|" should be read as "*such that*." So, if we write $V = \{f \in C^0([0,1])\}$ we are saying that the set contains all functions f that are in $C^0([0,1])$; f stands for a generic member of the set. It is equivalent to writing $V = C^0([0,1])$. The expression that defines V_1 above should be read as "*all functions f in $C^0([0,1])$ such that $f(0) = 2$*." The "|" serves the function of allowing us to add conditions for a function to belong to a set, and we do so by indicating the conditions on the generic member of the set f.

Examples:

1.14 Let $f(x) = x^2$ and $g(x) = x^2 + 2$. Then $f \notin V_1$ and $g \in V_1$.

1.15 Let $V_2 = \{g \in C^2([-1,1]) \mid g(-1) = 1, g'(1) = 2\}$. Then, $x^2 \in V_2$ but $h(x) = x^4 \notin V_2$, since $h'(1) = 4 \neq 2$.

1.16 Let $V_3 = \{h \in C^2([0,L]) \mid h(0) = 0, h(L) = 0\}$. Then $V_3 \subset C^2([0,L])$, that is, the set $C^2([0,L])$ contains all functions in V_3. This is a trivial statement, since in the definition of V_3 we are requesting functions to be in $C^2([0,L])$ as one of the conditions they should satisfy to belong to V_3. However, in the next section we will use the idea that the set $C^2([0,L])$ contains all functions in V_3, so it is a good idea to become familiar with this now.

1.17 Let $V_4 = \{h\colon [a,b] \to \mathbb{R} \text{ smooth} \mid h(a) = 0\}$. In this example, functions in V_4 take a real value for each point in the interval $[a,b]$, and the word smooth indicates that they should have as many continuous derivatives as required by the problem or the manipulation of expressions we perform; we will talk about what this precisely means later. For example, $\sin(x-a) \in V_4$, since all derivatives exists and are continuous, but the membership of $|x-a|$ will depend on the specifics of the problem.

1.1.6 Integration by Parts of Piecewise Smooth Functions*

Up to now we have been looking at examples in which all functions and a their derivatives are continuous, and we used the integration-by-parts formula on these functions to obtain the weak form. There are incentives, however, to expand the class of functions we consider so as to include functions in which either the function or some of its relevant derivatives are discontinuous. In particular, the Finite Element Method provides a way to construct sets of functions, and the less continuity requirements functions in a set have, the easier it is to construct the set of Finite Element functions. This is particularly true in two and three spatial dimensions, and over domains that cannot smoothly be mapped to a cube (curved domains, domains with holes, etc.). The integration by parts formula needs to be modified for functions with discontinuities, and this is the focus of the forthcoming discussion.

The type of functions we want to consider are illustrated by the following two (see Fig. 1.3):

- The hat function $N\colon \mathbb{R} \to \mathbb{R}$,

$$N(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & |x| \geq 1. \end{cases} \tag{1.38}$$

- The function $M\colon \mathbb{R} \to \mathbb{R}$

$$M(x) = N(x) + \begin{cases} 1 & x \geq 0 \\ 0 & x < 0. \end{cases} \tag{1.39}$$

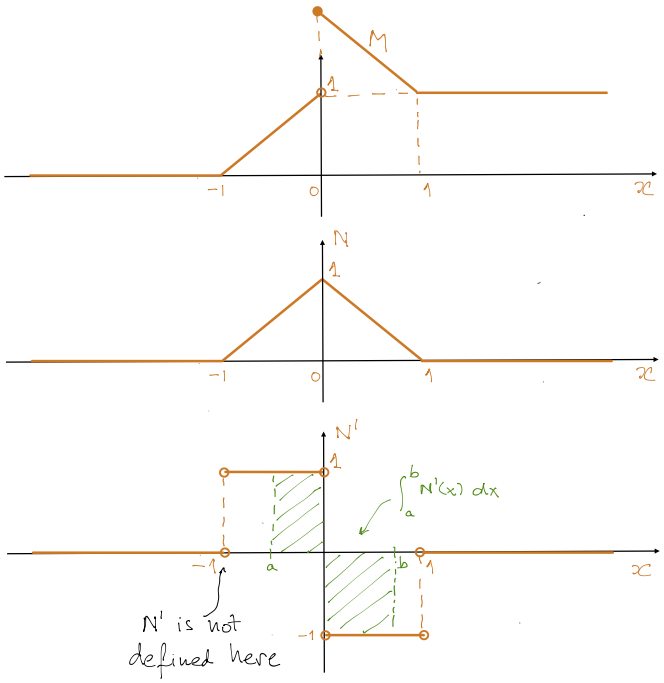


Figure 1.3 A hat function (middle), another function (top), and the common classical derivative (bottom).

The hat function N is continuous, while M is not continuous at $x = 0$. They have the same derivative[5], it is

$$N'(x) = M'(x) = \begin{cases} 0 & x < -1 \\ 1 & -1 < x < 0 \\ -1 & 0 < x < 1 \\ 0 & 1 < x \end{cases}$$

and it is not defined at $x \in \{-1, 0, 1\}$. So, the domain of $N'(x)$ and $M'(x)$ is $\mathbb{R} \setminus \{-1, 0, 1\}$.

Consider then the following question. Let v be a smooth function over $\mathbb{R}$, such as $v(x) = \sin x$, how do we apply the integration-by-parts formula to the following integrals?

$$\int_{-1}^{1} N(x)\, v'(x)\, dx, \qquad \int_{-1}^{1} M(x)\, v'(x)\, dx. \tag{1.41}$$

Neither N nor M are smooth over the interval $(-1, 1)$, so we need to proceed with caution. Notice, however, that both M and N are smooth over the intervals $(-1, 0)$

[5]Precisely, they have the same **classical derivative**. For a function $N\colon \mathbb{R} \to \mathbb{R}$, the classical derivative at a point x is computed as

$$N'(x) = \lim_{h \to 0} \frac{N(x+h) - N(x)}{h}. \tag{1.40}$$

The classical derivative is defined wherever this limit is.

and $(0,1)$, so we can proceed as follows, using u for either N or M,

$$\begin{aligned}\int_{-1}^{1} u(x)\,v'(x)\,dx &= \int_{-1}^{0} u(x)\,v'(x)\,dx + \int_{0}^{1} u(x)\,v'(x)\,dx \\ &= \lim_{x\to 0^-} u(x)\,v(x) - u(-1)\,v(-1) - \int_{-1}^{0} u'(x)\,v(x)\,dx \\ &\quad + u(1)\,v(1) - \lim_{x\to 0^+} u(x)\,v(x) - \int_{0}^{1} u'(x)\,v(x)\,dx \qquad (1.42)\\ &= u(1)\,v(1) - u(-1)\,v(-1) - \int_{-1}^{1} u'(x)\,v(x)\,dx \\ &\quad + v(0)\left(\lim_{x\to 0^-} u(x) - \lim_{x\to 0^+} u(x)\right)\end{aligned}$$

The value

$$[\![u]\!]_{x=c} = \lim_{x\to c^-} u(x) - \lim_{x\to c^+} u(x) \qquad (1.43)$$

is called the **jump** of u at $x = c \in \mathbb{R}$. Its value is equal to the jump discontinuity of u at $x = c$, so it is zero when u is continuous at $x = c$, and different than zero otherwise. For example, $[\![M]\!]_{x=0} = -1$. Hence, (1.42) for N and M is

$$\begin{aligned}\int_{-1}^{1} N(x)\,v'(x)\,dx &= -\int_{-1}^{1} N'(x)\,v(x)\,dx \\ &= -\int_{-1}^{0} v(x)\,dx + \int_{0}^{1} v(x)\,dx \\ \int_{-1}^{1} M(x)\,v'(x)\,dx &= v(1)M(1) + v(0)[\![M]\!]_{x=0} - \int_{-1}^{1} M'(x)\,v(x)\,dx \\ &= v(1) - v(0) - \int_{-1}^{0} v(x)\,dx + \int_{0}^{1} v(x)\,dx.\end{aligned}$$

So, the integration-by-parts formula applies as we know it for N over the interval $(-1,1)$, but not to M because of the discontinuity it has at $x = 0$. We generalize this observation next.

Functions N and M are called piecewise smooth, since each one of them has derivatives of any order in the open intervals $(-1,0)$ and $(0,1)$, but not on the entire real line. More generally, in the context of this class, we say that a function $f\colon (a,b) \subset \mathbb{R} \to \mathbb{R}$ is **piecewise smooth** over (a,b) if there are $k \in \mathbb{N}$ points $a = c_0 \le \ldots \le c_k = b$ such that f is smooth in each interval (c_i, c_{i+1}) for $i = 0, k-1$.

☞ If both functions u and v are continuous and piecewise smooth, then the integration-by-parts formula used for smooth functions holds.

Theorem 1.3 (Integration by Parts Formula for Piecewise Smooth Functions)**.** *Let $(a,b) \subset \mathbb{R}$, and u, v be piecewise smooth functions. Let $c_0 = a \le \ldots \le c_k = b$ for $k \in \mathbb{N}$ be such that both u and v are smooth in each interval (c_i, c_{i+1}) for $i = 0, \ldots, k-1$. Then,*

$$\int_a^b u'(x)\,v(x)\,dx = \sum_{i=0}^{k} [\![u(x)\,v(x)]\!]_{x=c_i} - \int_a^b u(x)\,v'(x)\,dx, \qquad (1.44)$$

where

$$\begin{aligned}[\![u(x)\,v(x)]\!]_{x=c_0} &= -\lim_{x\to a^+} u(x)\,v(x) \\ [\![u(x)\,v(x)]\!]_{x=c_k} &= \lim_{x\to b^-} u(x)\,v(x).\end{aligned}$$

The proof of this theorem is simple, and it follows the ideas we used in (1.42). It consists of decomposing the integral over (a,b) into a sum of integrals over (c_i, c_{i+1}) for $i = 0,\ldots,k-1$, and then integrating by parts in each one of these intervals, in which the two functions are smooth.

It follows from Thm. 1.3 that *if both u and v are continuous in (a,b) and piecewise smooth, then the same integration-by-parts used for smooth functions holds.*

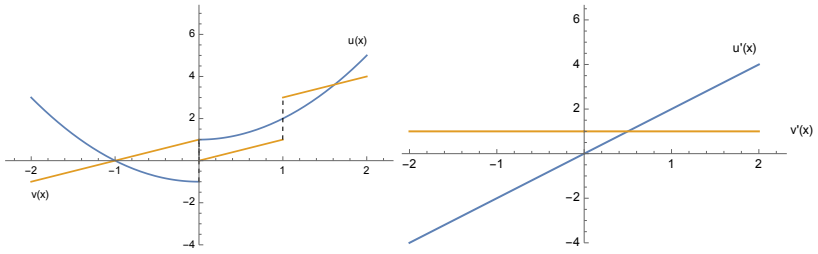
Example 1.18 Consider the functions

$$u(x) = \begin{cases} x^2 - 1 & x < 0, \\ x^2 + 1 & x \geq 0, \end{cases} \quad \text{and} \quad v(x) = \begin{cases} x+1 & x < 0, \\ x & x \in [0,1], \\ x+2 & x > 1. \end{cases}$$

Their derivatives are

$$u'(x) = 2x \text{ for } x \neq 0, \quad \text{and} \quad v'(x) = 1 \text{ for } x \notin \{0,1\}.$$

The two functions and their derivatives are plotted below:



Consider the following expression

$$\int_{-2}^{2} u(x)\, v'(x)\, dx. \tag{1.45}$$

Its value can be readily computed by direct integration, and it is

$$\int_{-2}^{2} u(x)\, v'(x)\, dx = \int_{-2}^{0} x^2 - 1\, dx + \int_{-2}^{0} x^2 + 1\, dx = \frac{16}{3}.$$

Let's integrate (1.45) by parts, and verify that returns the same value. To this end, notice that u is smooth in the intervals $(-2,0)$ and $(0,2)$, while v is smooth in the intervals $(-2,0)$, $(0,1)$, and $(1,2)$. The two functions are piecewise smooth in $(-2,2)$ if we select $k = 3$ and $c_0 = -2, c_1 = 0, c_2 = 1, c_3 =$

2. Notice that u is smooth in $(0,2)$, and hence it is automatically smooth in the two intervals (c_1,c_2) and (c_2,c_3). From (1.44),

$$\begin{aligned}\int_{-2}^{2} u(x)v'(x)\,dx &= [\![u(x)v(x)]\!]_{x=-2} + [\![u(x)v(x)]\!]_{x=0} + [\![u(x)v(x)]\!]_{x=1} \\ &\quad + [\![u(x)v(x)]\!]_{x=2} - \int_{-2}^{2} u'(x)v(x)\,dx \\ &= -u(-2)v(-2) + \lim_{x\to 0^-} u(x)v(x) - \lim_{x\to 0^+} u(x)v(x) \\ &\quad + \lim_{x\to 1^-} u(x)v(x) - \lim_{x\to 1^+} u(x)v(x) + u(2)v(2) \\ &\quad - \int_{-2}^{0} 2x(x+1)\,dx - \int_{0}^{1} 2x.x\,dx - \int_{1}^{2} 2x(x+2)\,dx \\ &= -3.(-1) + (-1).1 - 1.0 + 2.1 - 2.3 + 5.4 - \frac{38}{3} \\ &= \frac{16}{3},\end{aligned}$$

and we verified the identity.

An alternative way to obtain the same result is to split the integral as a sum of integrals over $(-2,0)$, $(0,1)$ and $(1,2)$, and then integrate each one of these integrals by parts. By rearranging the terms, we will arrive to the expression that we used from Thm. 1.3.

1.2 Galerkin Method

1.2.1 Vector Spaces of Functions

The first encounter with finite element methods is for many the first encounter with the use of vector spaces in a context other than one in which vectors represent elementary physics quantities, such as forces or velocities. It is also the first encounter with infinite-dimensional vector spaces. In studying finite element methods, we are interested in vector spaces in which each vector is a function, that is, in *vector spaces of functions*.

For example, consider the set V of all real quadratic polynomials that are zero at zero, namely, functions of the form

$$f(x) = ax^2 + bx$$

for any $a,b \in \mathbb{R}$, such as $f_1(x) = 3x^2$ and $f_2(x) = x$. Notice that $f_1(x) + f_2(x) = 3x^2 + x$ is also a function in V, and $3f_2(x) = 3x$ is another one, so addition of two polynomials in V returns a polynomial in V, and multiplication of a polynomial in V by a scalar (real number) is also a polynomial in V.

We can think of each polynomial in V as the vector in $\mathbb{R}^2$ that starts at the origin and ends at the point with coordinates (a,b), see Fig. 1.4. The sum of

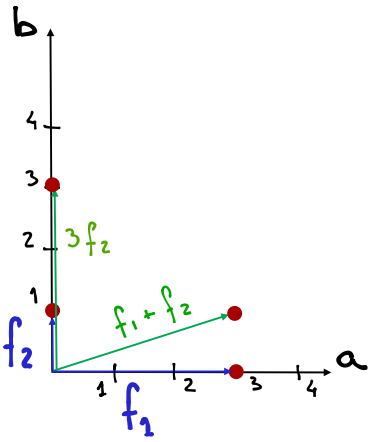


Figure 1.4 Identification of polynomials with vectors in $\mathbb{R}^2$.

two functions in V corresponds to adding the two vectors, and similarly with the multiplication by a scalar (real number). You may be wondering about why to call each function a vector, or simply why to talk about vector spaces of functions? With this identification we can define the *dimension* and a basis for V, and by using a basis, we will be able to build any function (vector) in the space.

We begin by reviewing the definition of vector spaces.

Definition 1.1 (Vector Space)**.** *A Vector Space V is a set for which two operations $+$ and $\cdot$ are defined, called* **vector addition** *and* **multiplication by a scalar**, *such that for all $u, v, w \in V$ and all $\alpha, \beta \in \mathbb{R}$ they satisfy:*

1. **Closure***: $u + v \in V$, and $\alpha \cdot u \in V$.*
2. **Commutativity:** $u + v = v + u$.
3. **Associativity:** $u + (v + w) = (u + v) + w$, *and* $\alpha \cdot (\beta \cdot u) = (\alpha\beta) \cdot u$.
4. **Identity:** *There exists an element $0 \in V$, called "zero," such that $u + 0 = u$, and $1 \cdot u = u$.*
5. **Additive Inverse:** *For any $u \in V$, there exists $v \in V$ such that $v + u = 0$.*
6. **Distributivity:** $(\alpha + \beta) \cdot u = \alpha \cdot u + \beta \cdot u$ *and* $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$.

The elements of V are called vectors.

In the following, we will drop the symbol $\cdot$ to indicate multiplication by a scalar, unless there is ambiguity. So, for example, for $\alpha \cdot u$ will write $\alpha\, u$.

You are by now very familiar with $\mathbb{R}^n$ as a vector space, $n \in \mathbb{N}$. Each point of $\mathbb{R}^n$ defines a vector under the standard definition of vector addition and multiplication by a scalar in $\mathbb{R}^n$. As aforementioned, what might be new for you is that sets of functions can also be vector spaces. In this case, each function in the set is a "vector."

To complete the depiction of functions as elements of a vector space, we need to specify the **vector-addition** and **multiplication-by-a-scalar** operations. Fortunately, they are defined exactly as you would expect: Let V be a set of functions over a domain $\Omega \subseteq \mathbb{R}^n$, and let $f, g \in V$ and $\alpha \in \mathbb{R}$, then

- **Vector addition:** the function defined as $h(x) = f(x) + g(x)$ for all $x \in \Omega$.
- **Multiplication by a scalar:** the function defined as $w(x) = \alpha f(x)$ for all $x \in \Omega$.

To illustrate this definition, we will consider sets of *smooth functions*, which as in previous sections, are functions that have as many continuous derivatives as required by the problem or the manipulation of expressions we perform.

Examples:

1.19 The set $V_1 = \{f : [a,b] \to \mathbb{R} \text{ smooth }\}$ is a vector space. We can check each one of the non-trivial properties:

i. Closure: If $u, v \in V_1$ and $\alpha \in \mathbb{R}$, then $u + v \in V_1$ and $\alpha u \in V_1$, since the sum of smooth functions is another smooth function, and multiplication of a smooth function by a scalar is anoter smooth function.

ii. Identity: the function $z(x) = 0$ for all $x \in [a,b]$ is the "zero" of the space.

The rest of the properties are easy to check.

1.20 The set $V_2 = \{f : [a,b] \to \mathbb{R} \text{ smooth } \mid w(a) = w(b) = 0\}$ is a vector space. Since $V_2 \subset V_1$ in Example 1.19, V_2 inherits commutativity, associativity, and distributivity in Def. 1.1 from V_1. We need to only check closure, identity and additive inverse. Closure follows because if $u, v \in V_2$, and $w = u + v$, then $w(a) = u(a) + v(a) = 0$ and $w(b) = u(b) + v(b) = 0$, and hence $w \in V_2$. Since the zero function is in V_2, identity is satisfied. Finally, if $u \in V_2$, then $-u \in V_2$ because $u(a) = -u(a) = 0$ and $u(b) = -u(b) = 0$, and hence additive inverse is also satisfied.

1.21 The set of polynomials of degree less or equal than $k \in \mathbb{N} \cup \{0\}$ over an interval $I \subset \mathbb{R}$, $\mathbb{P}_k(I)$, is a vector space. This is because the sum of polynomials in $\mathbb{P}_k$ and the product by a scalar is still a polynomial of degree less or equal than k (closure), and because the function $0 \in \mathbb{P}_k$ (identity). The rest of the properties are easy to check.

In addition to vector spaces, we will use a closely related concept, that of an affine subspace, which we define next.

Definition 1.2 (Affine Subspace)**.** *Let W be a vector space. An affine subspace of W is a set $S \subset W$ such that for any $s_1 \in S$ the set*

$$V = \{s_2 - s_1 \mid s_2 \in S\}$$

is a vector subspace of W. The vector space V is called the **direction** *of S, or the associated vector space to S.*

The direction V is independent of the choice of s_1.

The direction V is independent of the choice of s_1.

To see this, let $s_a, s_b \in S$ and

$$V_a = \{s - s_a \mid s \in S\}, \qquad V_b = \{s - s_b \mid s \in S\}$$

be the associated vector spaces to S. We will prove that $V_a = V_b$, and hence that the direction is independent of s_1. We will use the fact that $\Delta s = s_a - s_b$ belongs to both V_a and V_b, by definition. So, if $v \in V_a$, then there exists $\overline{s} \in S$ such that

$$\overline{s} = v + s_a = v + \Delta s + s_b,$$

and hence $w = v + \Delta s = \bar{s} - s_b \in V_b$. But $w = v + \Delta s \in V_a$, since $\Delta s \in V_a$, from where $v = w - \Delta s \in V_b$ as well, since $\Delta s \in V_b$. We conclude then that if $v \in V_a$, $v \in V_b$, or $V_a \subseteq V_b$. A similar argument leads to $V_a \supseteq V_b$, and hence to $V_a = V_b$.

Of course, using the notation from the definition, a vector space $V \subset W$ is an affine subspace of W. Elements of an affine subspace are called points and not vectors, since it is not a vector space.

Examples:

1.22 Let $v = (-1,-1) \in W = \mathbb{R}^2$. The set $S_1 = \{\alpha v \mid \alpha \in \mathbb{R}\}$ is a vector space. Instead, the set $S_2 = \{\alpha v + (0,1) \mid \alpha \in \mathbb{R}\}$ is *not* a vector space. This is because if $s_1 = \alpha_1 v + (0,1)$ and $s_2 = \alpha_2 v + (0,1)$, then $h = s_1 + s_2 = \alpha_1 v + (0,1) + \alpha_2 v + (0,1) = (\alpha_1 + \alpha_2) v + (0,2)$, and hence $h \notin S_2$.

Instead, the set S_2 is an affine subspace of W, since $s_1 - s_2 = (\alpha_1 - \alpha_2) v$, and hence $s_1 - s_2$ can be any element of the vector space S_1. Please see Fig. 1.5 for a sketch.

1.23 The set $V_3 = \{w\colon [a,b] \to \mathbb{R} \text{ smooth } \mid w(a) = 1 = w(b) = 1\}$ is *not* a vector space. This is because if $u, v \in V_3$, and $w = u + v$, then $w(a) = u(a) + v(a) = 2$, and similarly for $w(b)$, and hence $w \notin V_3$.

Instead, V_3 is an affine subspace of the vector space $W = \{w\colon [a,b] \to \mathbb{R} \text{ smooth}\}$. To see this, first notice first that for any $v_1, v_2 \in V_3$, $v_2 - v_1 \in V_2$ of Example 1.20, a vector space, since $v_1(a) - v_2(a) = 0 = v_1(b) - v_2(b)$. This implies that for any $v_1 \in V_3$,

$$V = \{v_2 - v_1 \mid v_2 \in V_3\} \subseteq V_2.$$

Second, notice that if $v \in V_2$, then $v_1 + v = v_2 \in V_3$, so $v_2 - v_1 = v$, from where

$$V \supseteq V_2.$$

Therefore, $V = V_2$, and hence V_3 is an affine subspace of W with V_2 as its direction.

Vector spaces that are subspaces of $\mathbb{R}^n$ can be identified with (hyper-)planes that contain the origin. Affine subspaces of $\mathbb{R}^n$ are (hyper-)planes that may not contain the origin, and hence, will be parallel to a vector subspace, their direction, as illustrated by S_1 and S_2 in Fig. 1.5. Also in the figure, notice that the role of the vector $(0,1)$ is to "transport" the vector space S_1 parallel to itself to become the affine subspace S_2. More generally, if s_1 is any element of an affine subspace S, then any other element $s \in S$ can be written as $s = s_1 + w$ for $w \in V$, since by the definition of an affine subspace, $s - s_1 = w \in V$. In Fig. 1.5, when $s = s_2$, then $w = \alpha_2 v$.

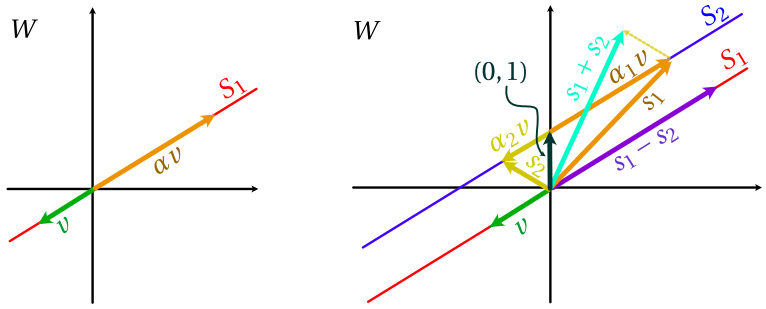


Figure 1.5 Sketch of sets S_1 and S_2 in Example 1.22. The former is a vector space, while the latter is not; it is an affine subspace of W instead.

More importantly, if $s_1 \in S$, then we can reach all elements in S by adding an element in its direction V, namely,

$$S = \{s_1 + w \mid w \in V\}. \tag{1.46}$$

$S = \{s_1 + w \mid w \in V\}$ **for any** $s_1 \in S$**.**

To see this, let

$$U = \{s_1 + w \mid w \in V\}.$$

We want to show that $\mathcal{U} = S$. For any $s_2 \in S$, $w = s_2 - s_1 \in V$, by definition, so $s_2 = w + s_1 \in U$. Therefore $U \supseteq S$.

To see that $U \subseteq S$, notice that for any $w \in V$, there exists $s_2 \in S$ such that $s_2 - s_1 = w$, since all elements of V are generated by such differences. This implies that $s_2 = s_1 + w$, and hence that $w + s_1 \in S$, or $U \subseteq S$.

Bases in a vector space of functions. Next, we review the definition of a basis for a vector space, and see examples of bases in vector spaces of functions.

Definition 1.3 (Linear combinations, or span)**.** *Let V be a vector space and $U = \{e_1, \ldots, e_n\} \subset V$ be a set of vectors in V. The* **span** *of U, span(U), is the set*

$$span(U) = \{v \in V \mid v = \sum_{i=1}^{n} c_i e_i, \text{ for } (c_1, \ldots, c_n) \in \mathbb{R}^n\}. \tag{1.47}$$

The set span(U) contains all linear combinations of vectors in the set U.

In the following, we denote vectors in $\mathbb{R}^n$ by the Cartesian coordinates of their end points.

Examples:

1.24 $U_1 = \{e_1, e_2\} \subset \mathbb{R}^3$, where $e_1 = (1,0,0)$ and $e_2 = (1,0,1)$. Then span(U_1) is the plane that contains 0 and vectors e_1 and e_2, or whose normal is in the direction $e_1 \times e_2$, where $'\times'$ is the vector cross-product. Notice that vectors in U do not need to be unit vectors.

1.25 $U_2 = \{1, x, x^2\}$. Then span(U_2) is the set of all quadratic polynomials. The vector with components $(3,4,5)$, or $c_1 = 3$, $c_2 = 4$ and $c_3 = 5$ is the polynomial

$$p(x) = 3 + 4x + 5x^2,$$

and the quadratic polynomial $q(x) = 5 - 2x - 6x^2$ has components $(5,-2,-6)$ in this basis.

Definition 1.4 (Linearly Independent and Linearly Dependent Set of Vectors)**.** *Let V be a vector space and let $e_i \in V$ for $i = 1,\ldots,n$. The set of vectors $U = \{e_1,\ldots,e_n\}$ is linearly independent whenever*

$$\sum_{i=1}^{n} c_i e_i = 0 \Longleftrightarrow c_i = 0 \text{ for } i = 1,\ldots,n. \tag{1.48}$$

Otherwise, the set of vectors U is linearly dependent.

Examples:

1.26 $U_1 = \{e_1, e_2\} \subset \mathbb{R}^3$, where $e_1 = (1,0,0)$ and $e_2 = (1,0,1)$. The set of vectors in U_1 is linearly independent:

$$c_1(1,0,0) + c_2(1,0,1) = (0,0,0) \Longleftrightarrow \begin{cases} c_1 + c_2 = 0 \\ c_2 = 0 \end{cases} \Longleftrightarrow c_1 = c_2 = 0.$$

1.27 $U_2 = \{1, x, x^2\}$. The set of vectors U_2 is linearly independent:

$$p(x) = c_1 + c_2 x + c_3 x^2 = 0 \quad x \in [0,1] \Longleftrightarrow c_1 = c_2 = c_3 = 0.$$

To see this, it is enough to evaluate $p(x)$ at three different locations, for example. Say, $p(0) = 0$, $p(1/2) = 0$, $p(1) = 0$. The resulting system of equations has $c_1 = c_2 = c_3 = 0$ as the only solution.

1.28 $U_3 = \{1, x, 2 + 3x\}$. This is a linearly dependent set of vectors, since if we let $e_1 = 1$, $e_2 = x$, and $e_3 = 2 + 3x$, then $2e_1 + 3e_2 - e_3 = 0$.

1.29 Consider the set $U_4 = \{\min\{0, x\}, x\}$ of functions with domain $[a, b]$. If $[a, b] = [-1, 1]$, U_4 is a set of linearly independent functions, since for

$$f(x) = c_1 \min\{0, x\} + c_2 x,$$

we have that

$$0 = f(-1) = -c_2, \qquad 0 = f(1) = c_1 + c_2 \Longrightarrow c_1 = c_2 = 0.$$

Instead, if $[a,b] = [0,1]$, this is a linearly dependent set. To see this, notice that for $x \in [0,1]$, $\min\{0,x\} = x$, so the two functions are precisely the same function over this interval.

Definition 1.5 (Basis and Dimension of a Vector Space)**.** *Let V be a vector space and $e_i \in V$ for $i = 1,\ldots,n$, $n \in \mathbb{N}$. The set $U = \{e_1,\ldots,e_n\}$ is a basis of V if U is linearly independent and span$(U) = V$. The number of vectors in a basis is the dimension of V.*

Given a basis $U = \{e_1,\ldots,e_n\}$ in a vector space V, and a vector $v \in V$, there there exists a *unique* set of numbers $(c_1,\ldots,c_n) \in \mathbb{R}^n$ such that

$$v = c_1 e_1 + \ldots + c_n e_n. \tag{1.49}$$

The numbers $c_1,\ldots,c_n$ are called the **components** of v in basis U.

Conversely, when the components $(c_1,\ldots,c_n)$ span all points in $\mathbb{R}^n$, the vector v in (1.49) spans the space V. Because all possible vectors in V are obtained by evaluating all possible values of $(c_1,\ldots,c_n)$, the variables $c_1,\ldots,c_n$ are called **degrees of freedom** of V.

Examples:

1.30 Consider the set $U_1 = \{e_1, e_2\} \subset \mathbb{R}^3$, where $e_1 = (1,0,0)$ and $e_2 = (1,0,1)$. The set of vectors in U_1 is not a basis for $\mathbb{R}^3$, since the vector $(0,1,0) \notin$ span(U_1).

1.31 Consider the set $U_4 = \{e_1, e_2, e_3\} \subset \mathbb{R}^3$, where $e_1 = (1,0,0)$, $e_2 = (1,0,1)$, and $e_3 = (0,1,0)$. The set U_4 is a basis for $\mathbb{R}^3$, since it can be seen to be linearly independent, and any vector in $\mathbb{R}^3$ is a linear combination of the basis: If $(x,y,z) \in \mathbb{R}^3$, then $(x,y,z) = (x-y)e_1 + ye_2 + ze_3$. The dimension of $\mathbb{R}^3$ is then 3.

1.32 The set $U_2 = \{1, x, x^2\} \subset V_1 = \{f\colon (0,1) \to \mathbb{R} \text{ smooth}\}$. The set of vectors U_2 is not a basis for V_1. In fact, there is no basis for V_1, and hence it is an **infinite dimensional space**.

The set U_2 is a basis for the vector space $\mathbb{P}_2$ formed by all quadratic polynomials, whose dimension is 3.

1.33 The set $U_5 = \{1 + x, x - x^2, x^2 - 1\}$ is another basis for $\mathbb{P}_2$. To see this, notice that given a polynomial $p(x) = a + bx + cx^2 \in \mathbb{P}_2$ for $a,b,c \in \mathbb{R}$, we can write it as

$$p(x) = \frac{a+b+c}{2}(1+x) + \frac{b-c-a}{2}(x-x^2) + \frac{b+c-a}{2}(x^2-1).$$

Instead, the set $U_5' = \{1 + x, x - x^2, x^2 + 1\}$ is not a basis, since the three functions are not linearly independent: $(x^2+1) + (x-x^2) = 1 + x$.

1.34 The set $U_6 = \{\sin(x), \sin(2x), \sin(3x), \sin(4x)\}$ is a basis for $\text{span}(U_6)$ over the interval $(0, 2\pi)$, since the 4 functions are linearly independent. One way to see the linear independence is as follows: If we have

$$\sum_{i=0}^{n} c_i \sin(ix) = 0$$

with $n = 4$ here, we need to show that this implies that $c_i = 0$ for any i. This follows by multiplying the last equation by $\sin(jx)$ for any $j \in \{1, \ldots, n\}$ and integrating over the interval $(0, 2\pi)$. In this case we get that

$$\sum_{i=0}^{n} c_i \int_0^{2\pi} \sin(ix)\sin(jx)\, dx = 0. \tag{1.50}$$

We then notice that

$$\int_0^{2\pi} \sin(ix)\sin(jx)\, dx = \begin{cases} \pi & i = j, \\ 0 & i \neq j. \end{cases}$$

Using this in (1.50) allows us to conclude that

$$0 = \sum_{i=0}^{n} c_i \underbrace{\int_0^{2\pi} \sin(ix)\sin(jx)\, dx}_{\neq 0 \text{ only if } i=j} \Longrightarrow c_j = 0,$$

and since this is true for any j, we can conclude that U_6 is a linearly independent set, and hence it is a basis for $\text{span}(U_6)$.

Linear functional and bilinear form. We conclude this section by introducing two more definitions, which will allow us to talk about weak forms in an abstract way.

Definition 1.6 (Linear Functional)**.** *Let V be a vector space. A linear functional is a function $\ell : V \to \mathbb{R}$ such that for any $u, v \in V$ and $\alpha \in \mathbb{R}$*

$$\ell(u + \alpha v) = \ell(u) + \alpha \ell(v). \tag{1.51}$$

Examples:

1.35 Let $V_1 = \{f : [0,1] \to \mathbb{R} \text{ smooth }\}$, then

$$\ell(v) = \int_0^1 x^2 v(x)\, dx$$

is a linear functional. This is because:

- The value of $\ell(v)$ can be computed for any function $v \in V_1$, so it is defined for *any* function in V_1.
- It is simple to see that (1.51) is true, to wit, for $u, v \in V_1$ and $\alpha \in \mathbb{R}$,

$$\begin{aligned}\ell(u+\alpha v) &= \int_0^1 x^2(u(x)+\alpha v(x))\,dx \\ &= \int_0^1 x^2 u(x)\,dx + \alpha\int_0^1 x^2 v(x)\,dx \\ &= \ell(u)+\alpha\ell(v).\end{aligned}$$

Let's compute the value of the linear functional for a couple of functions:

- $\ell(\cos(x)) = \int_0^1 x^2\cos(x)\,dx = 2\cos(1)-\sin(1)$.
- $\ell(x^4) = \int_0^1 x^2 x^4\,dx = \frac{x^7}{7}\Big|_{x=1} = \frac{1}{7}$.

This is an example of linear functionals of the form

$$\ell(v) = \int_a^b f(x)v(x)\,dx$$

for some function f, which we will encounter often in class.

1.36 Let $V = \mathbb{R}^2$, and $f = (f_1, f_2) \in \mathbb{R}^2$. For $v = (v_1, v_2) \in V$,

$$\ell(v) = v_1 f_1 + v_2 f_2 \tag{1.52}$$

is a linear functional.

1.37 Let V be the set of continuous functions over $\mathbb{R}$. For $v \in V$, let

$$\ell(v) = v(0) \tag{1.53}$$

This is a linear functional. You may have encountered this functional written in a different way:

$$\ell(v) = \int_{\mathbb{R}} \delta(x)v(x)\,dx,$$

namely, using the *Dirac delta function*. A problem with the denomination of $\delta(x)$ as a function is that $\delta(x)$ is not a function, but a linear functional.

A linear functional is also called a **one-form**.

Definition 1.7 (Bilinear Form)**.** *Let V be a vector space. A bilinear form is a function $a\colon V \times V \to \mathbb{R}$ that is linear in each argument. More precisely, for any $u, v, w \in V$ and $\alpha \in \mathbb{R}$*

$$\begin{aligned}a(u+\alpha v, w) &= a(u,w)+\alpha a(v,w) \\ a(w, u+\alpha v) &= a(w,u)+\alpha a(w,v).\end{aligned} \tag{1.54}$$

If, additionally, for all $u, v \in V$

$$a(u,v) = a(v,u), \tag{1.55}$$

then "a" is a symmetric bilinear form.

Examples:

1.38 Let $V_1 = \{f : [0,1] \to \mathbb{R} \text{ smooth}\}$, then

$$a(u,v) = \int_0^1 u'(x)\,v'(x)\,dx$$

is a bilinear form, since $a(u,v)$ can be computed for any functions $u, v \in V_1$, and it is simple to see that (1.54) is true. To wit,

$$\begin{aligned} a(u+\alpha v, w) &= \int_0^1 (u' + \alpha v')\,w'\,dx \\ &= \int_0^1 u'w'\,dx + \alpha \int_0^1 v'w'\,dx \\ &= a(u,w) + \alpha a(v,w), \end{aligned}$$

and similarly with the other slot.

This bilinear form is symmetric.

Let's compute the value of the bilinear form for a few functions:

- $a(\sin(x), x^2) = \int_0^1 \cos(x) 2x\,dx = 2(\sin(1) - \cos(1))$.
- $a((x-1)^2, x^3) = \int_0^1 2(x-1)3x^2\,dx = -1/2$.

1.39 Let $V = \mathbb{R}^2$, and

$$M = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

where $u = (u_1, u_2)$ and $v = (v_1, v_2)$ are vectors in $\mathbb{R}^2$, written in a column matrix form. Then, we can define

$$\begin{aligned} a(u,v) &= u^{\mathsf{T}} M v \\ &= \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \\ &= 2u_1v_1 + 3u_2v_1 + u_1v_2 + 2u_2v_2. \end{aligned} \tag{1.56}$$

This is a bilinear form in V. It is *not* a symmetric bilinear form. For example, $a((0,1),(2,0)) = 6$, and $a((2,0),(0,1)) = 2$.

Now, if

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{1.57}$$

then $a(u,v) = u \cdot v$, or the dot product between vectors u and v. So, the dot product *is* a bilinear form.

1.40 Let V_1 be that of Example 1.38, and set

$$a(u,v) = \int_0^1 \sin(x)u(x)v'(x)\,dx + u'(1/2)v(1/2).$$

This is a bilinear form, since it is defined for any pair of smooth functions, and it is linear in each argument. It is not symmetric, as it can be inferred from the different roles u and v play in each term. An alternative way to see this is by choosing two functions u and v and evaluating $a(u,v)$; there is a high chance that $a(u,v)$ will be different than $a(v,u)$ if a is not symmetric, and single pair of functions for which $a(u,v) \neq a(v,u)$ is enough to show that it is not symmetric. Set $u(x) = x^2$ and $v(x) = x^3$, then

$$a(x^2,x^3) = \int_0^1 \sin(x)x^2 3x^2\,dx + 2(1/2)(1/2)^3 = 577/8 - 39\cos(1) - 60\sin(1),$$
$$a(x^3,x^2) = \int_0^1 \sin(x)x^3 2x\,dx + 3(1/2)^2(1/2)^2 = 771/16 - 26\cos(1) - 40\sin(1).$$

so $a(x^2,x^3) \neq a(x^3,x^2)$, and this proves that a is not symmetric.

An Abstract Way to Write the Weak Form. Having defined linear functionals and bilinear forms, we can now write the weak forms we have seen so far in a simple, abstract way.

Problem 1.4 (Abstract Weak Form)**.** *Let $\mathcal{W}$ be a vector space, $a\colon \mathcal{W}\times\mathcal{W}\to\mathbb{R}$ be a bilinear form, and $\ell\colon \mathcal{W}\to\mathbb{R}$ be a linear functional. Let the trial space $\mathcal{S}$ be an affine subspace of $\mathcal{W}$, and let the test space $\mathcal{V}\subset\mathcal{W}$ be the direction of $\mathcal{S}$.*

$$\text{Find } u\in\mathcal{S} \text{ such that } a(u,v) = \ell(v) \text{ for all } v\in\mathcal{V}. \tag{1.58}$$

The best way to begin the discussion about this abstraction is by seeing how one weak form we have seen so far is expressed in this way.

Example 1.41 Bilinear form and linear functional for the model problem
Consider then the weak form in Problem 1.3, reproduced here for this discussion:

Let $\Omega = [0,L]\subset\mathbb{R}$, and

$$\mathcal{S} = \left\{w\colon \Omega\to\mathbb{R} \text{ smooth} \mid w(0) = g_0\right\}, \tag{1.59a}$$
$$\mathcal{V} = \left\{w\colon \Omega\to\mathbb{R} \text{ smooth} \mid w(0) = 0\right\}. \tag{1.59b}$$

Find $u\in\mathcal{S}$ such that for any functions $v\in\mathcal{V}$

$$\int_\Omega \left[k(x)u'(x)v'(x) + b(x)u(x)'v(x) + c(x)u(x)v(x)\right]\,dx - k(L)d_L v(L) = \int_\Omega f(x)v(x)\,dx. \tag{1.60}$$

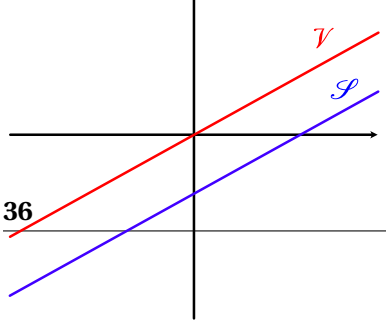


Figure 1.6

Let then

$$\mathcal{W} = \{w\colon \Omega \to \mathbb{R} \text{ smooth enough}\},$$

and we can then say that $\mathcal{S} \subset \mathcal{W}$ and $\mathcal{V} \subset \mathcal{W}$, that $\mathcal{S}$ is an affine subspace of $\mathcal{W}$ (see Example 1.23), and that $\mathcal{V}$ is the direction of $\mathcal{S}$, a vector space. Notice, that $\mathcal{S}$ is *not* a vector space unless $g_0 = 0$. The relationship between these three spaces is sketched in Fig. 1.6 when $\mathcal{W} = \mathbb{R}^2$. Additionally, we identify

$$a(u,v) = \int_\Omega \left[k(x)u'(x)v'(x) + b(x)u(x)'v(x) + c(x)u(x)v(x)\right]\,dx,$$
$$\ell(v) = \int_\Omega f(x)v(x)\,dx - k(L)d_L v(L),$$

as a bilinear form and a linear functional, respectively, both of which defined for any function in $\mathcal{W}$.

1.2.2 Galerkin Method

We show now the process of, given a weak form, formulating Galerkin Method. We will first look at Galerkin Methods over a general vector space of functions. Later, in §1.3.1 , we show an example of its application to vector spaces of functions built with Finite Elements.

Given the abstract weak form in Problem 1.4, *Galerkin Method* follows by restricting the trial and test functions to finite dimensional subspaces of $\mathcal{S}$ and $\mathcal{V}$.

Problem 1.5 (Abstract Galerkin Method for Problem 1.4)**.** *Let $\mathcal{W}_h \subset \mathcal{W}$ be a vector space, and let $\mathcal{S}_h = \mathcal{S} \cap \mathcal{W}_h$ and $\mathcal{V}_h = \mathcal{V} \cap \mathcal{W}_h$.*

$$\text{Find } u_h \in \mathcal{S}_h \text{ such that } a(u_h, v_h) = \ell(v_h) \text{ for all } v_h \in \mathcal{V}_h. \tag{1.61}$$

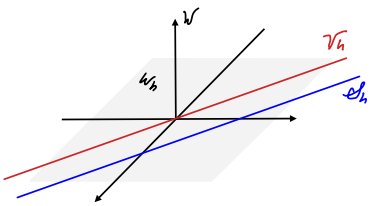


Figure 1.7

The space $\mathcal{W}_h$ is a finite dimensional space which we will construct through finite element techniques. However, to illustrate how $\mathcal{S}_h$ and $\mathcal{V}_h$ follow from it, we build a space $\mathcal{W}_h$ with polynomials in the next example.

Example 1.42 Spaces $\mathcal{V}_h$ and $\mathcal{S}_h$. Consider the spaces $\mathcal{W}$, $\mathcal{S}$ and $\mathcal{V}$ in Example 1.41 with $\Omega = [0,1]$ and $g_0 = 2$. We build example spaces $\mathcal{W}_h$, $\mathcal{S}_h$ and $\mathcal{V}_h$ next.

Let $\mathcal{W}_h = \text{span}(1, x, x^2, x^3)$ so that if $w_h \in \mathcal{W}_h$, then $w_h = w_0 \cdot 1 + w_1 x + w_2 x^2 + w_3 x^3$. Functions in $\mathcal{V}_h = \mathcal{W}_h \cap \mathcal{V}$ are those in $\mathcal{W}_h$ such that $w_h(0) = 0$, which implies that $w_0 = 0$. So we conclude that

$$\begin{aligned}\mathcal{V}_h &= \{w_h \in \mathcal{W}_h \mid w_h(0) = 0\}\\ &= \text{span}(x, x^2, x^3).\end{aligned}$$

Similarly, functions in $\mathcal{S}_h = \mathcal{W}_h \cap \mathcal{S}$ are those in $\mathcal{W}_h$ such that $w_h(0) = 2$, which implies that $w_0 = 2$. So we conclude that

$$\begin{aligned}\mathcal{S}_h &= \{w_h \in \mathcal{W}_h \mid w_h(0) = 2\} \\ &= \{w_h = 2 + c_1 x + c_2 x^2 + c_3 x^3 \mid (c_1, c_2, c_3) \in \mathbb{R}^3\} \\ &= \{w_h = 2 + v_h \mid v_h \in \mathcal{V}_h\}. \end{aligned} \tag{1.62}$$

In the statement of Problem 1.4, $\mathcal{V}$ is the direction of $\mathcal{S}$. It is also true that $\mathcal{V}_h$ is the direction of $\mathcal{S}_h$. This is what we see in (1.62), as part of Example 1.42.

$\mathcal{V}_h$ is the direction of $\mathcal{S}_h$

To see this, for any $u_h \in \mathcal{S}_h$, let

$$\mathcal{U}_h = \{w_h - u_h \mid w_h \in \mathcal{S}_h\}.$$

We first see that $w_h - u_h \in \mathcal{V}_h$ for any $w_h \in \mathcal{S}_h$, and hence that $\mathcal{U}_h \subseteq \mathcal{V}_h$. To this end, notice that since $\mathcal{S}_h = \mathcal{S} \cap \mathcal{W}_h$, then $u_h, w_h \in \mathcal{S}$, and hence $u_h - w_h \in \mathcal{V}$. Since $u_h - v_h \in \mathcal{W}_h$ as well, then $u_h - v_h \in \mathcal{V} \cap \mathcal{W}_h = \mathcal{V}_h$.

Next, we need to show that $\mathcal{U}_h \supseteq \mathcal{V}_h$. Given $v_h \in \mathcal{V}_h$, let $w_h = v_h + u_h$. Since $v_h \in \mathcal{V}$ and $u_h \in \mathcal{S}$, it follows that $w_h \in \mathcal{S}$, from (1.46). But $w_h \in \mathcal{W}_h$, so $w_h \in \mathcal{S} \cap W_h = \mathcal{S}_h$. Therefore, $v_h = w_h - u_h \in \mathcal{U}_h$, and thus $\mathcal{V}_h \subseteq \mathcal{U}_h$.

We can conclude then that $\mathcal{U}_h = \mathcal{V}_h$, or that $\mathcal{V}_h$ is the direction of $\mathcal{S}_h$.

1.2.3 Solution to Galerkin Method

We next describe the general procedure to find the solution of (3.7), regardless of the way we construct the discrete space $\mathcal{W}_h$, finite elements or another method.

Let $\{N_a\}_{a=1,\dots,m} = \{N_1, \dots N_m\}$, $m \in \mathbb{N}$, be a basis for $\mathcal{W}_h$. Then, the approximate solution $u_h \in \mathcal{S}_h$ of Problem 1.5 and any test function $v_h \in \mathcal{V}_h$ can be written as

$$\begin{aligned} u_h(x) &= \sum_{b=1}^{m} u_b N_b(x) \\ v_h(x) &= \sum_{a=1}^{m} v_a N_a(x). \end{aligned}$$

Additionally, we will assume that the subset of basis functions $\{N_a\}_{a=1,\dots,n}$ with $n \leq m$ is a basis for $\mathcal{V}_h$. Graphically,

$$\underbrace{\overbrace{N_1, \dots, N_n}^{\text{Basis for } \mathcal{V}_h}, , \dots, N_m}_{\text{Basis for } \mathcal{W}_h}. \tag{1.63}$$

This automatically means that $v_a = 0$ for $n < a \leq m$.

The solution u_h satisfies (3.7) for any $v_h \in \mathcal{V}_h$. To find u_h, we will take advantage that we can choose the test functions v_h we can "test" with, and of the fact that u_h belongs to $\mathcal{S}_h$. If we choose enough test functions, we will get enough equations to define u_h completely. We can then show that such u_h satisfies (3.7) for any $v_h \in \mathcal{V}_h$, not only for those chosen as particular test functions.

For this plan, we will select each basis function of $\mathcal{V}_h$ as a test function, namely,

$$\ell(N_a) = a(u_h, N_a) \qquad a = 1, \dots, n. \tag{1.64a}$$

This gives us n algebraic equations, for the m unknown components $\{u_1, \dots, u_m\}$ of u in the basis $\{N_a\}_{a=1,\dots,m}$. The remaining $n-m$ equations follow from the fact that $u_h \in \mathcal{S}_h$. Typically, this means that the remaining equations come from the boundary conditions. To impose them, it is enough to select *any* element $\overline{u}_h$ of $\mathcal{S}_h$, write

$$\overline{u}_h = \underbrace{\overline{u}_1 N_1 + \dots + \overline{u}_n N_n}_{\in \mathcal{V}_h} + \underbrace{\dots + \overline{u}_m N_m}_{\notin \mathcal{V}_h}$$

and set

$$u_b = \overline{u}_b \qquad n < b \le m, \tag{1.64b}$$

which provide the remaining $n-m$ equations needed to completely determine the m components $u_1, \dots, u_m$ of u_h in the basis $\{N_1, \dots, N_m\}$.

The solution to (1.64) amounts to the solution of a linear system of equations. To see this, we first expand u_h in components inside (1.64a) and use the bilinearity of a to get:

$$\ell(N_a) = a(u_h, N_a) = a\left(\sum_{b=1}^{m} u_b N_b, N_a\right) = \sum_{b=1}^{m} a(N_b, N_a)\, u_b \qquad a = 1, \dots, n. \tag{1.65}$$

We then label

$$F_a = \ell(N_a), \qquad K_{ab} = a(N_b, N_a), \qquad 1 \le a \le n,\ 1 \le b \le m$$

and from (1.64b),

$$F_a = \overline{u}_a, \qquad K_{ab} = \delta_{ab}, \qquad\qquad n < a \le m,\ 1 \le b \le m$$

where δ_{ab} is called the **Kronecker Delta** [6], and arrange them in a matrix and two columns vectors

$$K = \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1m} \\ K_{21} & K_{22} & \dots & K_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1} & K_{m2} & \dots & K_{mm} \end{bmatrix}, \quad F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}, \text{ and } U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}. \tag{1.66}$$

☞ We could be a little bit more specific about K, and use that we know the values for $a, b > n$. Namely,

$$K = \begin{bmatrix} K_{11} & \dots & K_{1(n+1)} & \dots & K_{1m} \\ \vdots & \ddots & \vdots & & \vdots \\ K_{n1} & \dots & K_{n(n+1)} & \dots & K_{nm} \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 \end{bmatrix}$$

[6] It is defined as $\delta_{ab} = \begin{cases} 1 & a = b, \\ 0 & a \neq b. \end{cases}$

Then, (1.64) is expressed as the linear system of equations

$$KU = F. \tag{1.67}$$

The matrix K is often called the **stiffness matrix** and the vector F is often called the **load vector**, for their origins in mechanical problems.

Solving the linear system (1.67) for U defines the components $u_1, \ldots, u_m$ needed to construct the function $u_h = u_1 N_1 + \ldots + u_m N_m$, the solution to Galerkin Method.

Example 1.43 Consider the weak form in Example 1.41 with $k(x) = f(x) = 1$ and $b(x) = c(x) = 0$ for $x \in [0,1]$ over the domain $\Omega = (0,1)$, and set $g_0 = 2$ and $d_L = 0$. The bilinear form and linear functional in this case take the form

$$a(u,v) = \int_0^1 u'(x)v'(x)\,dx$$
$$\ell(x) = \int_0^1 v(x)\,dx.$$

We will seek a solution to Galerkin Method with the spaces in Example 1.42, namely,

$$\mathcal{W}_h = \text{span}(1, x, x^2, x^3)$$
$$\mathcal{V}_h = \text{span}(x, x^2, x^3)$$
$$\mathcal{S}_h = \{2 + v_h \mid v_h \in \mathcal{V}_h\}.$$

Hence, we have $m = 4$ and $n = 3$, and index the basis functions in $\mathcal{W}_h$ so that indices 1 to 3 form a basis for $\mathcal{V}_h$. To wit, we set $N_1(x) = x$, $N_2(x) = x^2$, $N_3(x) = x^3$ and $N_4(x) = 1$. We also need to choose one element $\overline{u}_h$ of $\mathcal{S}_h$. Among the infinite options we have here, one is $\overline{u}_h(x) = 2N_4(x) = 2$, and another one is $\overline{u}_h(x) = 3N_1(x) + 2N_4(x)$. Notice that regardless of what we choose for $\overline{u}_h$, all of them will have $\overline{u}_4 = 2$, since this is the only way to construct the constant function 2 needed to belong to $\mathcal{S}_h$.

With these choices, the equations imposed by Galerkin Method, (1.64a), are

$$a(u_h, N_1) = \ell(N_1)$$
$$a(u_h, N_2) = \ell(N_2)$$
$$a(u_h, N_3) = \ell(N_3)$$

while the equations that impose that $u_h \in \mathcal{S}_h$, (1.64b), is

$$u_4 = 2.$$

Replacing, the load vector is

$$F = \begin{bmatrix} \ell(N_1) \\ \ell(N_2) \\ \ell(N_3) \\ u_4 \end{bmatrix} = \begin{bmatrix} \int_0^1 x\,dx \\ \int_0^1 x^2\,dx \\ \int_0^1 x^3\,dx \\ \overline{u}_4 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/3 \\ 1/4 \\ 2 \end{bmatrix}.$$

The stiffness matrix is

$$
\begin{aligned}
K &= \begin{bmatrix} a(N_1,N_1) & a(N_2,N_1) & a(N_3,N_1) & a(N_4,N_1) \\ a(N_1,N_2) & a(N_2,N_2) & a(N_3,N_2) & a(N_4,N_2) \\ a(N_1,N_3) & a(N_2,N_3) & a(N_3,N_3) & a(N_4,N_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \int_0^1 1\cdot 1\,dx & \int_0^1 2x\cdot 1\,dx & \int_0^1 3x^2\cdot 1\,dx & \int_0^1 0\cdot 1\,dx \\ \int_0^1 1\cdot 2x\,dx & \int_0^1 2x\cdot 2x\,dx & \int_0^1 3x^2\cdot 2x\,dx & \int_0^1 0\cdot 2x\,dx \\ \int_0^1 1\cdot 3x^2\,dx & \int_0^1 2x\cdot 3x^2\,dx & \int_0^1 3x^2\cdot 3x^2\,dx & \int_0^1 0\cdot 3x^2\,dx \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 4/3 & 3/2 & 0 \\ 1 & 3/2 & 9/5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

The components of u_h are then

$$
U = K^{-1}F = \begin{bmatrix} 1 \\ -1/2 \\ 0 \\ 2 \end{bmatrix}, \tag{1.68}
$$

from where

$$
u_h(x) = 1.N_1(x) - 1/2N_2(x) + 0.N_3(x) + 2N_4(x) = 2 + x - \frac{x^2}{2}. \tag{1.69}
$$

This happens to be the *exact* solution of the problem, whose strong form consists of the following three equations:

$$
\begin{aligned}
-u''(x) &= 1 \qquad x \in (0,1) \\
u(0) &= 2 \\
u'(1) &= 0.
\end{aligned}
$$

This can be understood because in this case $\mathscr{S}_h$ *contains* the exact solution of the problem. Galerkin method will *always* return the exact solution if it belongs to $\mathscr{S}_h$. In general, however, this will not be the case.

We complete this section by answering two questions about this solution procedure to Galerkin Method. The first question is: *Why do we test with the basis functions only, if the variational equation* (3.7) *should hold for* all *test functions?* The answer is that if the variational equation is satisfied for every function in a basis for the test space $\mathscr{V}_h$, it is satisfied for *every* function in the test space.

The proof is simple, and worth reading, and it takes advantage of the bilinearity of a and the linearity of ℓ:

$$\begin{aligned} a(u_h, v_h) &= a\left(u_h, \sum_{b=1}^{n} v_b N_b\right) \\ &= \sum_{b=1}^{n} v_b a(u_h, N_b) && \text{bilinearity of } a \\ &= \sum_{b=1}^{n} v_b \ell(N_b) && \text{use of (1.64a)} \\ &= \ell\left(\sum_{b=1}^{n} v_b N_b\right) && \text{linearity of } \ell \\ &= \ell(v_h). \end{aligned}$$

So, for $u_h \in \mathcal{S}_h$, (1.64a) implies (3.7). The converse is trivially true, namely, if (3.7) is satisfied for any $v_h \in \mathcal{V}_h$, it is satisfied for any basis function $N_b \in \mathcal{V}_h$ in particular, and hence it implies (1.64a). In summary, if $u_h \in \mathcal{S}_h$,

$$u_h \text{ is a solution of (3.7)} \iff u_h \text{ is a solution of (1.64a)}.$$

In words, this implies that the solution of Galerkin Method is a solution of the linear system of equations defined by the basis functions of the test space, and conversely[7].

The second question we answer is: *Why does the solution u_h belong to $\mathcal{S}_h$, and why is it independent of our choice of $\overline{u}_h \in \mathcal{S}_h$?* The answer to this relies on the fact that $\mathcal{S}_h$ is an affine subspace of $\mathcal{W}_h$ and $\mathcal{V}_h$ is its direction. To see that $u_h \in \mathcal{S}_h$, notice that since $u_b = \overline{u}_b$ for $n < b \le m$, then

$$\Delta u_h = u_h - \overline{u}_h = (u_1 - \overline{u}_1)N_1 + \ldots + (u_n - \overline{u}_n)N_n,$$

from where we conclude that $\Delta u_h \in \mathcal{V}_h$, or $u_h = \overline{u}_h + \Delta u_h$, and hence it follows from (1.46) that $u_h \in \mathcal{S}_h$. To see that the choice of $\overline{u}_h$ does not affect the u_h we compute, consider another function $\overline{w}_h \in \mathcal{S}_h$. Then, by the definition of affine subspace, $\overline{u}_h - \overline{w}_h \in \mathcal{V}_h$, or in terms of the basis for $\mathcal{W}_h$,

$$\overline{u}_h = \overline{w}_h + \sum_{b=1}^{n} v_b N_b.$$

So, $\overline{u}_h$ and $\overline{w}_h$ can only differ in the values of the components $v_1, \ldots, v_n$, but need to have precisely the same values for the components v_b with $n < b \le m$. Since the latter are the only components that participate in (1.64b), the solution u_h does not change if we choose $\overline{w}_h$ instead. In other words, u_h does not depend on our choice of $\overline{u}_h$.

[7] It is possible to regard (1.64a) as the Euler-Lagrange equations of (3.7).

Galerkin Method with an arbitrarily-ordered basis. In general, an ordered basis as in (1.63) is not readily available, nor is it necessary. We discuss next how to proceed in the case in which the basis functions for $\mathcal{W}_h$ and $\mathcal{V}_h$ are not neatly ordered as in the earlier discussion.

Again, let $\{N_a\}_{a=1,\ldots,m}$ be a basis for $\mathcal{W}_h$, and again we will assume that a subset of $n \le m$ of these basis functions is a basis for $\mathcal{V}_h$. However, the basis for $\mathcal{V}_h$ need *not* be the set $\{N_a\}_{a=1,\ldots,n}$. To indicate the basis for $\mathcal{V}_h$, it is convenient to introduce three sets of indices, or **index sets**. First, we denote by $\eta = \{1,\ldots,m\}$ the set of indices of all basis functions in $\mathcal{W}_h$. The basis functions for $\mathcal{V}_h$ can be indicated by a subset of η. The set of indices of basis functions for $\mathcal{V}_h$ is denoted $\eta_a \subset \eta$; it is called the set of **active indices**, and we can write

$$\mathcal{V}_h = \text{span}\left(\bigcup_{a\in\eta_a} \{N_a\} \right)$$

or

$$w_h \in \mathcal{V}_h \Longleftrightarrow w_h = \sum_{a\in\eta_a} w_a N_a.$$

The remaining indices in η, those that are *not* in η_a, is denoted $\eta_g = \eta \setminus \eta_a$; it is called the set of **constrained indices**.

We next rewrite the equations to solve Galerkin Method using these index sets. First, testing with each basis function in $\mathcal{V}_h$, (1.64a), is restated as

$$\ell(N_a) = a(u_h, N_a) \qquad a \in \eta_a \tag{1.70a}$$

The arbitrary element $\overline{u}_h \in \mathcal{S}_h$ used to impose the fact that $u_h \in \mathcal{S}_h$ is still written as $\overline{u}_h = \overline{u}_1 N_1 + \ldots + \overline{u}_m N_m$, but (1.64b) is restated as

$$u_b = \overline{u}_b \qquad b \in \eta_g. \tag{1.70b}$$

We then label

$$\begin{aligned} &F_a = \ell(N_a), \quad K_{ab} = a(N_b, N_a) \quad a \in \eta_a, b \in \eta \\ &F_a = \overline{u}_a, \qquad K_{ab} = \delta_{ab} \qquad\quad a \in \eta_g, b \in \eta \end{aligned} \tag{1.70c}$$

which define the stiffness matrix K and load vector F. To illustrate these ideas, let's consider Example 1.43 again.

Example 1.44 Consider Example 1.43 again, but in this case we set $N_1(x) = x$, $N_2(x) = 1$, $N_3(x) = x^2$ and $N_4(x) = x^3$. Therefore, the basis for $\mathcal{V}_h$ is $\{N_1, N_3, N_4\}$, and the index sets are $\eta = \{1,2,3,4\}$, $\eta_a = \{1,3,4\}$, and $\eta_g = \{2\}$. We can then set $\overline{u}_h = 2N_2(x) = 2$.

The stiffness matrix and load vector in this case are

$$F = \begin{bmatrix} \int_0^1 x\,dx \\ \overline{u}_2 \\ \int_0^1 x^2\,dx \\ \int_0^1 x^3\,dx \end{bmatrix} = \begin{bmatrix} 1/2 \\ 2 \\ 1/3 \\ 1/4 \end{bmatrix}.$$

The stiffness matrix is

$$K = \begin{bmatrix} \int_0^1 1\cdot 1\,dx & \int_0^1 0\cdot 1\,dx & \int_0^1 2x\cdot 1\,dx & \int_0^1 3x^2\cdot 1\,dx \\ 0 & 1 & 0 & 0 \\ \int_0^1 1\cdot 2x\,dx & \int_0^1 0\cdot 2x\,dx & \int_0^1 2x\cdot 2x\,dx & \int_0^1 3x^2\cdot 2x\,dx \\ \int_0^1 1\cdot 3x^2\,dx & \int_0^1 0\cdot 3x^2\,dx & \int_0^1 2x\cdot 3x^2\,dx & \int_0^1 3x^2\cdot 3x^2\,dx \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 4/3 & 3/2 \\ 1 & 0 & 3/2 & 9/5 \end{bmatrix}.$$

The components of the solution are

$$U = \begin{bmatrix} 1 \\ 2 \\ -1/2 \\ 0 \end{bmatrix},$$

and the solution is

$$u_h(x) = 1.N_1(x) + 2N_2(x) - 1/2N_3(x) + 0N_4(x) = 2 + x - \frac{x^2}{2}, \tag{1.71}$$

which is exactly the same function we obtained in Example 1.43.

Comparing the stiffness matrix and load vector in Examples 1.43 and 1.44, notice that they have the same entries, but reordered: the last row and column in Example 1.43 were moved to be the second row and column in Example 1.44. The solution U in the former has the last row moved to the be the second row in the latter. Of course, the solution u_h is the same in both cases, since the entries in U are multiplied by the reordered basis functions as well.

To conclude this discussion, notice that reordering the basis functions does not change the spaces $\mathcal{V}_h$, $\mathcal{S}_h$ and $\mathcal{W}_h$, and hence it should not change the solution to Galerkin Method, as we have just seen.

Example 1.45 Let's look at another example of Galerkin Method, in this case with a basis of trigonometric functions. To this end, we will revisit Example 1.9 in a domain $\Omega = [0, \pi/2]$. In this case, the strong form of the problem is given by

$$-u''(x) + u'(x) + u(x) = -5\exp(-2x), \qquad \forall\, x \in \Omega, \tag{1.72a}$$
$$u(0) = 1, \tag{1.72b}$$
$$u'(\pi/2) = -2\exp(-\pi). \tag{1.72c}$$

and the exact solution is $u(x) = \exp(-2x)$. The weak form of the problem is: *Find $u \in \mathcal{S}$ such that for any function $v \in \mathcal{V}$*

$$\int_0^{\pi/2} u'(x)v'(x) + u'(x)v(x) + u(x)v(x)\,dx + 2\exp(-\pi)v(\pi/2) = -\int_0^{\pi/2} 5\exp(-2x)v(x)\,dx, \quad (1.72d)$$

where

$$\mathcal{S} = \{w\colon [0,1] \to \mathbb{R} \text{ smooth } \mid w(0) = 1\}$$
$$\mathcal{V} = \{w\colon [0,1] \to \mathbb{R} \text{ smooth } \mid w(0) = 0\}.$$

The bilinear form and linear functional here are

$$a(u,v) = \int_0^{\pi/2} u'(x)v'(x) + u'(x)v(x) + u(x)v(x)\,dx \quad (1.72e)$$
$$\ell(v) = -\int_0^{\pi/2} 5\exp(-2x)v(x)\,dx - 2\exp(-\pi)v(\pi/2). \quad (1.72f)$$

The bilinear form is not symmetric.

To formulate Galerkin Method, we consider the space

$$\mathcal{W}_h^n = \text{span}(1, \sin x, \ldots, \sin nx)$$

for $n \in \mathbb{N}$. We included a dependence on n for generality, but we will proceed with $n = 2$ next. To this end, we will label $N_1(x) = 1, N_2(x) = \sin x, N_3(x) = \sin 2x$.

Let's find spaces $\mathcal{S}_h$ and $\mathcal{V}_h$ next. For $w_h \in \mathcal{W}_h^2$, we can write

$$\begin{aligned} w_h(x) &= w_1 N_1(x) + w_2 N_2(x) + w_3 N_3(x) \\ &= w_1 .1 + w_2 \sin x + w_3 \sin 2x. \end{aligned}$$

For $w_h \in \mathcal{V}_h$, we need $0 = w_h(0) = w_1$, and for $w_h \in \mathcal{S}_h$, we need $1 = w_h(0) = w_1$. These are also sufficient conditions: if $w_1 = 0$, then $w_h \in \mathcal{V}_h$, and if $w_1 = 1$, then $w_h \in \mathcal{S}_h$. Therefore,

$$\begin{aligned} \mathcal{S}_h &= \{w_h \in W_h^2 \mid w_h(0) = 1\} \\ &= \{1 + w_2 \sin x + w_3 \sin 2x \mid (w_2, w_3) \in \mathbb{R}^2\} \\ \mathcal{V}_h &= \{w_h \in W_h^2 \mid w_h(0) = 0\} \\ &= \{w_2 \sin x + w_3 \sin 2x \mid (w_2, w_3) \in \mathbb{R}^2\}. \end{aligned}$$

To proceed, we need to identify active and constrained indices. In this case, $\eta_c = \{1\}$ and $\eta_a = \{2,3\}$. The stiffness matrix is then (careful because this is a non-symmetric bilinear form):

$$K = \begin{bmatrix} 1 & 0 & 0 \\ a(N_1,N_2) & a(N_2,N_2) & a(N_3,N_2) \\ a(N_1,N_3) & a(N_2,N_3) & a(N_3,N_3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & (1+\pi)/2 & 2/3 \\ 1 & 2 & 5\pi/4 \end{bmatrix}.$$

We set $\overline{u}_h(x) = 1$, so that $\overline{u}_h \in \mathcal{S}_h$, and $u_1 = 1$. The load vector is

$$F = \begin{bmatrix} 1 \\ \ell(N_2) \\ \ell(N_3) \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -5(1+\exp(-\pi))/4 \end{bmatrix}.$$

The components of the solution are obtained from $U = K^{-1}F$, or

$$U = \begin{bmatrix} 1 \\ -\frac{36+20\exp(-\pi)-60\pi}{32-15\pi-15\pi^2} \\ -\frac{3\exp(-\pi)\left(5(1+\pi)+\exp(\pi)(9\pi-23)\right)}{-32+15\pi+15\pi^2} \end{bmatrix} \approx \begin{bmatrix} 1 \\ -0.93 \\ -0.11 \end{bmatrix}.$$

Hence,

$$u_h(x) \approx 1 - 0.93\sin x - 0.11\sin 2x.$$

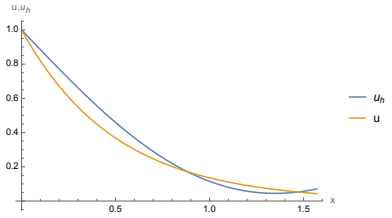


Figure 1.8

A plot of the exact versus the approximate solutions is shown in Fig. 1.8. By selecting a larger value of n, a better approximation is obtained. You can check that.

Example 1.46 Let's consider a twist of Example 1.45 to illustrate the effect of additional boundary conditions on the method. To this end, we change the problem in that example to have a Dirichlet boundary condition at $x = \pi/2$ as well, keeping the same exact solution. The problem is

$$-u''(x) + u'(x) + u(x) = -5\exp(-2x), \qquad \forall\, x \in \Omega, \tag{1.73a}$$

$$u(0) = 1, \tag{1.73b}$$

$$u(\pi/2) = \exp(-\pi). \tag{1.73c}$$

and the exact solution is still $u(x) = \exp(-2x)$. The weak form of the problem is: *Find $u \in \mathcal{S}$ such that for any function $v \in \mathcal{V}$*

$$\int_0^{\pi/2} u'(x)v'(x) + u'(x)v(x) + u(x)v(x)\,dx = -\int_0^{\pi/2} 5\exp(-2x)v(x)\,dx, \tag{1.73d}$$

where

$$\mathcal{S} = \{w\colon [0,1] \to \mathbb{R} \text{ smooth } \mid w(0) = 1, w(\pi/2) = \exp(-\pi)\}$$
$$\mathcal{V} = \{w\colon [0,1] \to \mathbb{R} \text{ smooth } \mid w(0) = 0, w(\pi/2) = 0\}.$$

Notice that there is no longer a term that appears from the natural boundary condition.

The bilinear form and linear functional here are

$$a(u,v) = \int_0^{\pi/2} u'(x)v'(x) + u'(x)v(x) + u(x)v(x)\,dx \tag{1.73e}$$

$$\ell(v) = -\int_0^{\pi/2} 5\exp(-2x)v(x)\,dx. \tag{1.73f}$$

The bilinear form is not symmetric.

In this case, we consider the space

$$\mathcal{W}_h = \text{span}(1, \sin x, \sin 2x, \sin 4x).$$

We will label $N_1(x) = 1, N_2(x) = \sin x, N_3(x) = \sin 2x$, $N_4(x) = \sin 4x$. Notice that we did not include the function $\sin 3x$.

Let's find spaces $\mathcal{S}_h$ and $\mathcal{V}_h$ next. For $w_h \in \mathcal{W}_h$, we can write

$$\begin{aligned} w_h(x) &= w_1 N_1(x) + w_2 N_2(x) + w_3 N_3(x) + w_4 N_4(x) \\ &= w_1.1 + w_2 \sin x + w_3 \sin 2x + w_4 \sin 4x. \end{aligned}$$

For $w_h \in \mathcal{V}_h$, we need $0 = w_h(0) = w_1$ and $0 = w_h(\pi/2) = w_1 + w_2$, or $w_1 = w_2 = 0$. For $w_h \in \mathcal{S}_h$, we need $1 = w_h(0) = w_1$ and $\exp(-\pi) = w_h(\pi/2) = w_1 + w_2$, or $w_1 = 1$ and $w_2 = \exp(-\pi) - 1$. These are also sufficient conditions: if $w_1 = w_2 = 0$, then $w_h \in \mathcal{V}_h$, and if $w_1 = 1$ and $w_2 = \exp(-\pi) - 1$, then $w_h \in \mathcal{S}_h$. Therefore,

$$\begin{aligned} \mathcal{S}_h &= \{w_h \in W_h \mid w_h(0) = 1, w_h(\pi/2) = \exp(-\pi)\} \\ &= \{1 + (\exp(-\pi) - 1)\sin x + w_3 \sin 2x + w_4 \sin 4x \mid (w_3, w_4) \in \mathbb{R}^2\} \\ \mathcal{V}_h &= \{w_h \in W_h \mid w_h(0) = w_h(\pi/2) = 0\} \\ &= \{w_3 \sin 2x + w_4 \sin 4x \mid (w_3, w_4) \in \mathbb{R}^2\}. \end{aligned}$$

Had we included a term with $\sin 3x$, the characterization of $\mathcal{S}_h$ and $\mathcal{V}_h$ would have been somewhat more complicated, because we would have had a total of 3 functions that are non-zero at $x = \pi/2$.

The active and constrained indices are $\eta_c = \{1, 2\}$ and $\eta_a = \{3, 4\}$. The stiffness matrix is then (careful because this is a non-symmetric bilinear form):

$$\begin{aligned} K &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a(N_1, N_3) & a(N_2, N_3) & a(N_3, N_3) & a(N_4, N_3) \\ a(N_1, N_4) & a(N_2, N_4) & a(N_3, N_4) & a(N_4, N_4) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 5\pi/4 & -4/3 \\ 0 & -4/15 & 4/3 & 17\pi/4 \end{bmatrix}. \end{aligned}$$

We set $\overline{u}_h(x) = 1 + (\exp(-\pi) - 1)\sin x$, so that $\overline{u}_h \in \mathcal{S}_h$, $u_1 = 1$ and $u_2 = \exp(-\pi) - 1$. The load vector is

$$F = \begin{bmatrix} 1 \\ \exp(-\pi) - 1 \\ \ell(N_3) \\ \ell(N_4) \end{bmatrix} = \begin{bmatrix} 1 \\ \exp(-\pi) - 1 \\ 5(1 + \exp(-\pi))/4 \\ \exp(-\pi) - 1 \end{bmatrix}.$$

The components of the solution are obtained from $U = K^{-1}F$, or

$$U = \begin{bmatrix} 1 \\ \exp(-\pi) - 1 \\ \frac{\exp(-\pi)(-e^{\pi}(1216+765\pi)-9945\pi+1216)}{5(256+765\pi^2)} \\ \frac{12\exp(-\pi)(e^{\pi}(4-19\pi)+19\pi+52)}{256+765\pi^2} \end{bmatrix} \approx \begin{bmatrix} 1 \\ -0.96 \\ -0.13 \\ -0.08 \end{bmatrix}.$$

The solution is then

$$1 - 0.96\sin x - 0.13\sin 2x - 0.08\sin 4x,$$

and it is plotted in Fig. 1.9.

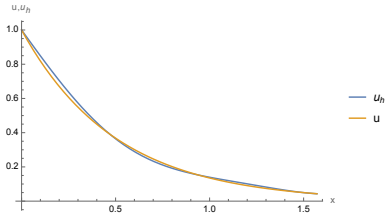


Figure 1.9

1.3 The Finite Element Method

By now we have learned about how to obtain the weak form from the strong form of a problem, and to formulate a Galerkin Method to obtain an approximation of the solution. This last step relied on the construction of a trial and a test space in which to seek the approximate solution. We next look at how Finite Elements provide a systematic way to construct such spaces.

The Finite Element Method (FEM) is obtained by seeking the solution of Galerkin Method in trial and test spaces constructed with Finite Elements.

We will also describe *how* to compute the stiffness matrix and load vector of Galerkin Method in Finite Element spaces. The way this computation is performed, called **assembly**, is a distinctive feature and a virtue of the Finite Element method, since it can be done very efficiently in a computer.

1.3.1 The Simplest C^0 Finite Element Space

We proceed now to show a first example of a solution computed with the finite element method, building the simplest finite element space of continuous functions. We do this for the problem in Example 1.43, so that we can contrast the use of Galerkin Method with and without Finite Element spaces.
Steps:

1. **Build the mesh of the domain.** Let the domain of the problem be the interval $\Omega = [c,d]$. We partition the domain into $n_{\text{el}} \in \mathbb{N}$ intervals by selecting $\{x_i\}_{i=1,\ldots,n_{\text{el}}+1}$ such that

$$c = x_1 < \ldots < x_{n_{\text{el}}+1} = d. \tag{1.74}$$

Each point x_i is a **vertex**, and i is its **vertex number**. Interval $[x_i, x_{i+1}]$ is called *element* i, for $i = 1,\ldots,n_{\text{el}}$. Strictly speaking, this is the domain of the

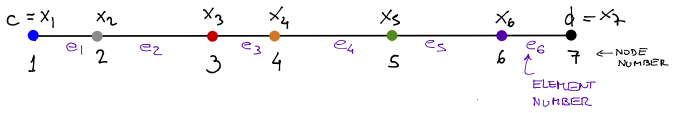


Figure 1.10

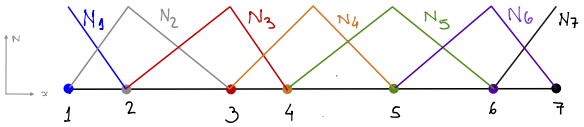
element, but it is common to denote the domain of the element simply as "element." The collection of nodes and elements is the **mesh**; we shall give a more complete definition of the mesh when we look at 2D problems.

For our example, we choose a uniform mesh, so $c = 0, d = 1$, and $x_a = (a-1)/n_{\text{el}}$ for $i = 1, \ldots, n_{\text{el}}+1$.

2. **Build basis functions.** For this example, we build the so-called continuous "piecewise affine" elements, or hat functions. These functions have domain $[c,d]$, and for $a = 1, \ldots, n_{\text{el}}+1$ are defined as

$$N_a(x) = \begin{cases} 0 & x < x_{a-1} \\ \frac{x-x_{a-1}}{x_a - x_{a-1}} & x_{a-1} \le x < x_a \\ 1 & x = x_a \\ \frac{x_{a+1}-x}{x_{a+1}-x_a} & x_a < x \le x_{a+1} \\ 0 & x_{a+1} < x \end{cases} \tag{1.75}$$
$$= \max\left[0, \min\left(\frac{x-x_{a-1}}{x_a - x_{a-1}}, \frac{x-x_{a+1}}{x_a - x_{a+1}}\right)\right].$$

Notice that when $a = 1$, $x \in [c,d]$ implies that we only have the case $x \ge x_a$ [8]. Similarly, when $a = n_{\text{el}}+1$, $x \in [c,d]$ implies that we only have the case $x \le x_a$ [9]. These functions are plotted below.



The Finite Element space is $\mathcal{W}_h = \text{span}(N_1, \ldots, N_{n_{\text{el}}+1})$, so for this example, $m = n_{\text{el}}+1$. Some properties that will be generalized later to many other shape functions:

(a) You can check that all of the add up to 1, i.e., $\sum_{a=1}^{n_{\text{el}}+1} N_a(x) = 1$ for $x \in [c,d]$.

A simple way to see this, is to notice that in each element e, the two non-zero functions N_e and N_{e+1} are affine, and hence their sum is

[8] We do not know what x_{a-1} is in this case, nor do we need it
[9] We do not know what x_{a+1} is in this case, but again, we do not need it

affine. But $N_e + N_{e+1}$ is equal to 1 at $x = x_e$ and $x = x_{e+1}$, and hence their sum is the only affine function that is equal to 1 at both locations: this is the constant function equal to 1.

(b) Notice that $N_b(x_a) = \delta_{ba}$. This is a particular version of a more general property we will see later in the class, and it has the following neat consequence. A function $w_h \in \mathcal{W}_h$ can be written as $w_h = w_1 N_1 + w_2 N_2 + \ldots + w_{n_{\text{el}}+1} N_{n_{\text{el}}+1}$, where $w_1, \ldots, w_{n_{\text{el}}+1}$ are the components of w_h in the basis. At the same time, $w_a = w_h(x_a)$ for $a = 1, \ldots, n_{\text{el}} + 1$, that is, the component w_a is the value of the function w_h at x_a. This follows because

$$w_h(x_a) = w_1 \underbrace{N_1(x_a)}_{=\delta_{1a}} + \ldots + w_b \underbrace{N_b(x_a)}_{=\delta_{ba}} + \ldots + w_{n_{\text{el}}+1} \underbrace{N(x_a)}_{=\delta_{(n_{\text{el}}+1)a}} = w_a,$$

so it is a special property of the basis we chose for $\mathcal{W}_h$.

Had we chosen the basis $\{N_1 + N_2, N_2, \ldots, N_{n_{\text{el}}+1}\}$, for example, then $w_h(x_2) = w_1 + w_2$, and in this case w_2 does not necessarily coincide with the value of w_h at x_2.

To indicate that the degrees of freedom $\{w_1, \ldots, w_{n_{\text{el}}+1}\}$ of the function w_h in the basis $\{N_1, \ldots, N_{n_{\text{el}}+1}\}$ are precisely the values of w_h at each vertex x_a, we say that there is a **node** of the finite element space at each vertex of this mesh, and graphically depict it with a filled disk at the vertex; see Fig. 1.10.

(c) Notice that $N_a(x) \neq 0$ only in a small part of the domain. This is normally referred to by saying that the basis functions have "compact support." In the Finite Element context, this (generally) means that basis functions are non-zero in at most one element and its neighbors.

In this case we defined the space $\mathcal{W}_h$ as the span of a set of basis functions. Alternatively, it could have been defined as

$$\mathcal{W}_h = \{w_h \colon [c,d] \to \mathbb{R} \text{ continuous } \mid w_h \text{ is affine on each element } e\}. \quad (1.76)$$

This space would often be referred to as the "space of piecewise affine functions over $[c,d]$," with the tacit understanding that functions would be affine over each element. Both definitions are equivalent; it is simple to see that functions in $\text{span}(N_1, \ldots, N_{n_{\text{el}}+1})$ are piecewise affine, and that any piecewise affine function can be expressed as a linear combination of functions in the basis $\{N_1, \ldots, N_{n_{\text{el}}+1}\}$.

3. **Build $\mathcal{V}_h$ and $\mathcal{S}_h$.** In Galerkin Method, we want $\mathcal{V}_h = \mathcal{W}_h \cap \mathcal{V}$ and $\mathcal{S}_h = \mathcal{W}_h \cap \mathcal{S}$. As a reminder, in our example

$$\mathcal{S} = \{u \colon [c,d] \to \mathbb{R} \text{ smooth} \mid u(0) = 2\}, \mathcal{V} \quad = \{v \colon [c,d] \to \mathbb{R} \text{ smooth} \mid v(0) = 0\}.$$

So, since any function $v_h \in \mathcal{W}_h$ is already smooth (continuous), it belongs to $\mathcal{V}$ is an only if it satisfies $v_h(0) = 0$. This is true if and only if $v_1 = 0$. Similarly,

and function $u_h \in \mathcal{W}_h$ is smooth, and hence it belongs to $\mathcal{S}_h$ if and only if $u_h(0) = 2$, which is true if and only if $u_1 = 2$. So we can write

$$\begin{aligned}
\mathcal{V}_h &= \{v_h \in \mathcal{W}_h \mid v_h(0) = 0\} \\
&= \{v_2 N_2 + \ldots + v_{n_{\text{el}}+1} N_{n_{\text{el}}+1} \mid v_2, \ldots, v_{n_{\text{el}}+1} \in \mathbb{R}\} \\
&= \text{span}\left(N_2, \ldots, N_{n_{\text{el}}+1}\right). \\
\mathcal{S}_h &= \{u_h \in \mathcal{W}_h \mid u_h(0) = 2\} \\
&= \{u_h \in \mathcal{W}_h \mid u_1 = 2\} \\
&= \{2N_1 + v_h \mid v_h \in \mathcal{V}_h\}.
\end{aligned}$$

We can then identify the index set η_a that identifies the basis functions for $\mathcal{V}_h$, and its complement η_g. These are

$$\begin{aligned}
\eta_a &= \{2, \ldots, n_{\text{el}} + 1\} \\
\eta_g &= \{1\}.
\end{aligned}$$

Finally, we need to identify the components $\overline{u}_a$ for $a \in \eta_g$, to impose the fact that $u_h \in \mathcal{S}_h$. In this case, based on the identification we did of $\mathcal{S}_h$ above, it is $\overline{u}_1 = 2$.

As an exercise, we can also identify a function $\overline{u}_h \in \mathcal{S}_h$. For example, we can choose $\overline{u}_h = 2N_1$, which gives $\overline{u}_1 = 2$. Alternatively, we can set $\overline{u}_h = 2 = 2\sum_{a\in\eta} N_a$, which is also in $\mathcal{S}_h$ because it is in $\mathcal{W}_h$ and $\overline{u}_h(0) = 2$.

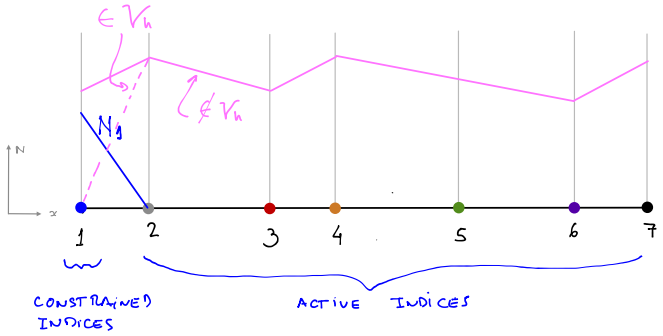


Figure 1.11 If $w_1 \neq 0$, then $w_h \notin \mathcal{V}_h$.

Notice that the constraint imposed on v_1 stems from the essential boundary condition (EBC) included in $\mathcal{V}$. Should more EBC be present in the definition of $\mathcal{V}$, more constraints should be imposed on functions in $\mathcal{W}_h$ to belong to $\mathcal{V}_h$. The EBC will also determine components of $\overline{u}_a$ for $a \in \eta_g$.

Not every constraint can be imposed by simply selecting a subset of a set of basis functions for $\mathcal{W}_h$, as we have assumed so far. For example, had we chosen the basis $\{N_1, N_1 + N_2, N_3, \ldots, N_{n_{\text{el}}+1}\}$ for $\mathcal{W}_h$, then $w_h(0) = w_1 + w_2$, and the condition for $w_h \in \mathcal{W}_h$ to belong to $\mathcal{V}_h$ is to have $w_1 + w_2 = 0$, see Fig.

1.12. For example, the function $w_h = N_1 + (N_2 - N_1)$ has $w_1 = 1$ and $w_2 = 1$ and is in $\mathcal{V}_h$. Therefore, just setting either w_1 or w_2 (or both) to zero does not lead to a basis for $\mathcal{V}_h$. It is not possible in this case to extract a subset of $\{N_1, N_1 + N_2, N_3, \ldots, N_{n_{\text{el}}+1}\}$ to serve as a basis for $\mathcal{V}_h$[10]. As a result, it is not possible to define η_a or η_g.

In the finite element method, this type of situations need a different treatment (e.g., with Lagrange multipliers), and the most commonly used finite element bases are constructed so that essential boundary conditions can be imposed by setting the values of some components, such as $v_1 = 0$ here. In other words, in the finite element method it is common for the basis for $\mathcal{V}_h$ to be a subset of the basis for $\mathcal{W}_h$. This is going to be the case for the examples we will see.

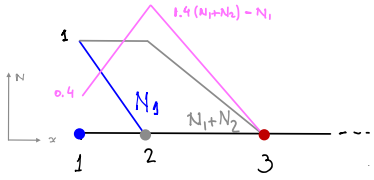


Figure 1.12 The function $w_h = 1.4(N_1 + N_2) - N_1$ has $w_1 = -1$ and $w_2 = 1.4$ in the basis $\{N_1, N_1 + N_2, \ldots\}$, and $w_h(0) = w_1 + w_2 = 0.4 \neq w_1$, so $w_1 = 0$ is not enough to impose the EBC.

Summarizing, in this step we:

(a) Identify active and constrained index sets, η_a and η_g (this assumes that the basis for $\mathcal{V}_h$ is a subset of the basis for $\mathcal{W}_h$).

(b) Build $\mathcal{V}_h = \text{span}\left(\{N_a \in \{N_1, \ldots, N_m\} \mid a \in \eta_a\}\right)$.

(c) Identify $\overline{u}_h \in \mathcal{W}_h$ so that $\mathcal{S}_h = \{v + \overline{u}_h \mid v \in \mathcal{V}_h\} = \left\{v + \sum_{a\in\eta_g} \overline{u}_a N_a \mid v \in \mathcal{V}_h\right\}$.

4. **Compute K and F.** We proceed as we did earlier and compute the stiffness matrix and load vector. We compute its entries according to (1.70).

   For our example, we can set $h = 1/n_{\text{el}}$ with $n_{\text{el}} = 5$ and hence $m = 6$,

$$\ell(N_a) = \int_0^1 1 \cdot N_a(x)\, dx = \begin{cases} \dfrac{h}{2} & a \in \{1, m\} \\ h & a \in \{2, \ldots, m-1\}. \end{cases}$$

$$a(N_b, N_a) = \int_0^1 N_b'(x) N_a'(x)\, dx = \begin{cases} 0 & |a-b| > 1 \\ -\dfrac{1}{h} & |a-b| = 1 \\ \dfrac{2}{h} & a = b \in \{2, \ldots, m-1\}. \\ \dfrac{1}{h} & a = b \in \{1, m\}. \end{cases}$$

[10] Of course, $\mathcal{V}_h$ has a basis, but it is not a subset of the chosen basis for $\mathcal{W}_h$.

The only index in η_g is 1. Therefore, according to (1.70c),

$$\begin{aligned}
K_{21} &= a(N_1, N_2) = -\frac{1}{h},\\
K_{12} &= \delta_{12} = 0,\\
K_{11} &= \delta_{11} = 1,\\
K_{22} &= a(N_2, N_2) = \frac{2}{h},\\
K_{23} &= a(N_3, N_2) = -\frac{1}{h},\\
K_{24} &= a(N_4, N_2) = 0,\\
K_{66} &= a(N_6, N_6) = \frac{1}{h},\\
F_1 &= \overline{u}_1 = 2,\\
F_5 &= \ell(N_5) = h,\\
F_6 &= \ell(N_6) = \frac{h}{2}.
\end{aligned}$$

We have not replaced $h = 1/m = 1/5$ yet, for clarity. In this case, the stiffness matrix and load vector are, now replacing $h = 1/5$,

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ -5 & 10 & -5 & 0 & 0 & 0\\ 0 & -5 & 10 & -5 & 0 & 0\\ 0 & 0 & -5 & 10 & -5 & 0\\ 0 & 0 & 0 & -5 & 10 & -5\\ 0 & 0 & 0 & 0 & -5 & 5 \end{bmatrix} \qquad F = \begin{bmatrix} 2\\ 0.2\\ 0.2\\ 0.2\\ 0.2\\ 0.1 \end{bmatrix}. \tag{1.77}$$

5. **Solve and Compute the Finite Element Solution.** We now solve the system $KU = F$, and then build the finite element solution as $u_h(x) = \sum_{a=1}^{m} u_a N_a(x)$.

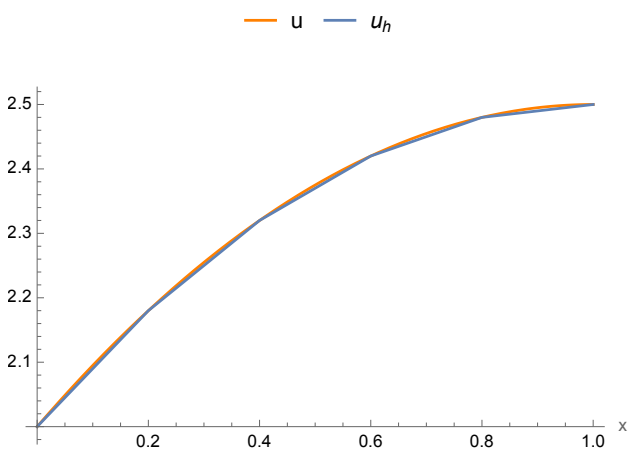
   For our example,

$$U = \begin{bmatrix} 2\\ 2.18\\ 2.32\\ 2.42\\ 2.48\\ 2.5 \end{bmatrix}$$

   and hence

$$u_h(x) = 2N_1(x) + 2.18N_2(x) + 2.32N_3(x) + 2.42N_4(x) + 2.48N_5(x) + 2.5N_6(x).$$

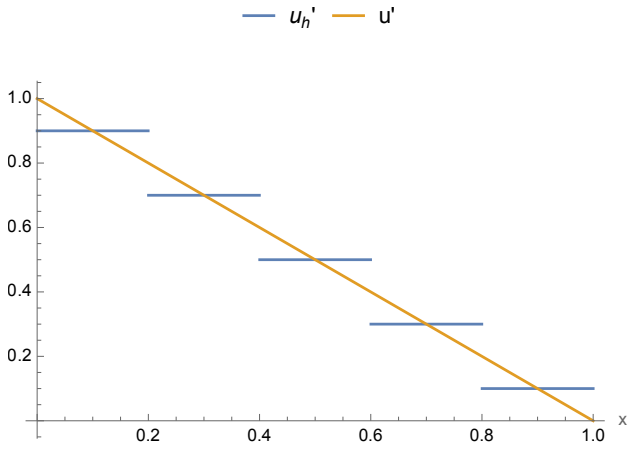
   This function is plotted below, together with the exact solution $u(x) = 2 + x - x^2/2$.



The derivative of u_h follows as

$$u_h'(x) = 2N_1'(x) + 2.18N_2'(x) + 2.32N_3'(x) + 2.42N_4'(x) + 2.48N_5'(x) + 2.5N_6'(x).$$

The derivative of the finite element approximation is a piecewise constant function, a fact reflected in its graph, shown below.



We conclude by showing the stiffness matrix of this problem for a constant h and any m, in the case in which $\mathcal{V}_h = \mathcal{W}_h$ (no boundary EBC). This is a matrix also found in finite differences in the same problem, and hence it is a commonly found matrix in elementary numerical analyses textbooks. According to step 4, the matrix is

$$K = \frac{1}{h}\begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix}.$$

1.3.2 What is a Finite Element?

We proceed now to describe how we construct finite element spaces beyond the span of hat functions that we introduced in §1.3.1. The construction of finite element spaces is done in two steps: (1) definition of vector spaces of functions over each element, and (2) adding functions defined over different elements to form functions whose domain is the entire interval $[c,d]$. We describe the first step next, and the second step in the next section, §1.3.3.

We begin by introducing the definition of a Finite Element.

Definition 1.8 (Finite Element)**.** *A finite element is a pair* $e = (\Omega_e, \mathscr{N}^e)$ *of an element domain* Ω_e *and a finite set of basis functions* $\mathscr{N}^e = \{N_1^e, \ldots, N_k^e\}$ *defined over* Ω_e*.*

Given a finite element $e = (\Omega_e, \mathscr{N}_e)$ with element domain Ω_e and a set $\mathscr{N}^e = \{N_1^e, \ldots, N_k^e\}$ of linearly independent functions $N_i^e : \Omega_e \to \mathbb{R}$, the space of functions $\mathscr{P}^e$ over Ω_e is defined as

$$\mathscr{P}^e = \text{span}\{N_1^e, \ldots, N_k^e\} \tag{1.78}$$

for $k \geq 1$, and it is called the **element space**. The set $\mathscr{N}^e$ is a basis for $\mathscr{P}^e$. Functions in $\mathscr{N}^e$ are known as **shape functions**. The number of shape functions k, or dimension of $\mathscr{P}^e$, is the **number of degrees of freedom** of the element. The **degrees of freedom** of the element are the components $\{\phi_1^e, \ldots, \phi_k^e\}$ of functions in this basis. Each one of the components ϕ_i^e, $i = 1, \ldots, k$, is a variable that can take any real value, and hence the k-tuple $(\phi_1^e, \ldots, \phi_k^e)$ can take any value in $\mathbb{R}^k$. For each such value, a unique function $f^e \in \mathscr{P}^e$, $f^e : \Omega_e \to \mathbb{R}$, is defined through

$$f^e(x) = \phi_1^e N_1^e(x) + \ldots + \phi_k^e N_k^e(x) = \sum_{a=1}^{k} \phi_a N_a^e(x). \tag{1.79}$$

The symbol e will be used interchangeably to denote an element or an **element index**, often a natural number, given that the index is another way to identify what element we are referring to. It is also common to use the word element in lieu of element domain; for example, wording such as *... integrating over an element...*, as a way to say integrating over Ω_e.

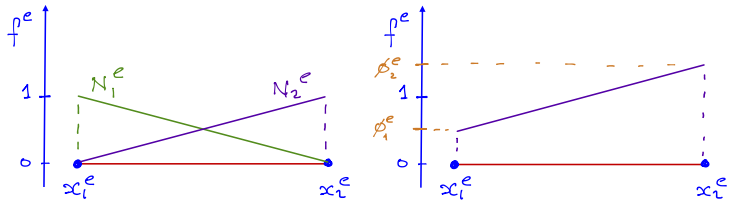
For the following examples we will consider a generic elements with element domain $\Omega_e = [x_1^e, x_2^e]$, with vertex 1 at x_1^e and vertex 2 at x_2^e.

Examples:

1.47 P_1**-element.** One of the simplest element spaces is generated by the basis functions

$$\begin{aligned} N_1^e(x) &= \frac{x - x_2^e}{x_1^e - x_2^e}, \\ N_2^e(x) &= \frac{x - x_1^e}{x_2^e - x_1^e}, \end{aligned} \tag{1.80}$$

which satisfy that $N_a^e(x_b^e) = \delta_{ab}$.



To see that the two are linearly independent, let

$$f^e(x) = \phi_1^e \frac{x - x_2^e}{x_1^e - x_2^e} + \phi_2^e \frac{x - x_1^e}{x_2^e - x_1^e}$$

and assume that $f^e(x) = 0$ for all $x \in \Omega_e$. In particular, $f^e(x_1^e) = \phi_1^e = 0$, and similarly, $f^e(x_2^e) = \phi_2^e = 0$. Therefore, this is a set of linearly independent functions.

The space $\mathscr{P}^e$ has 2 degrees of freedom, it is the space $\mathbb{P}_1(\Omega_e)$ of all polynomials of degree 1 or less over Ω_e. To see this, notice that $N_1^e(x) + N_2^e(x) = 1$ for all x, and $x_1^e N_1^e(x) + x_2^e N_2^e(x) = x$, so $\{1, x\} \in \mathscr{P}^e$. The degrees of freedom here are the values of f^e at x_1^e and x_2^e; this is the interpretation of ϕ_1^e and ϕ_2^e. Thus, we say that this element has a node at x_1^e and a node at x_2^e, and indicated them with a filled disc as follows

1.48 A variation of the P_1-element has the basis

$$\begin{aligned} N_1^e(x) &= 1 \\ N_2^e(x) &= x. \end{aligned} \tag{1.81}$$

The space span$\{N_1^e, N_2^e\}$ is still $\mathbb{P}_1(\Omega_e)$. However, the degrees of freedom in this case do not always lend themselves to be interpreted as pointwise values of the function $f^e = \phi_1^e\, 1 + \phi_2^e x$ somewhere in the element. There is no standard graphical depiction of this element.

1.49 **P_1-element+bubble.** Next, consider the basis functions

$$\begin{aligned} N_1^e(x) &= \frac{x - x_2^e}{x_1^e - x_2^e}, \\ N_2^e(x) &= \frac{x - x_1^e}{x_2^e - x_1^e}, \\ N_3^e(x) &= 4 N_1^e(x) N_2^e(x). \end{aligned} \tag{1.82}$$

Their plot is

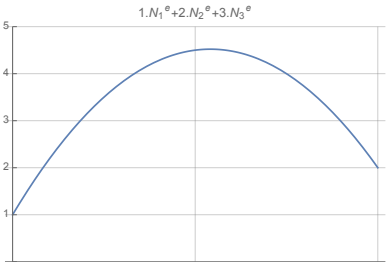
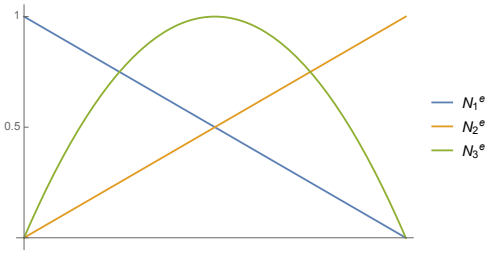


Figure 1.13 A function in the P_1-element+bubble.



It is simple to check that this is a set of linearly independent functions. A function $f^e \in \mathscr{P}^e$ has the form

$$f^e(x) = \phi_1^e \frac{x - x_2^e}{x_1^e - x_2^e} + \phi_2^e \frac{x - x_1^e}{x_2^e - x_1^e} + \phi_3^e 4 \frac{(x - x_1^e)(x_2^e - x)}{(x_1^e - x_2^e)^2},$$

and one example of such function is shown in Fig. 1.13.

The space $\mathscr{P}^e$ has 3 degrees of freedom, and it is the space $\mathbb{P}_2(\Omega_e)$ of all polynomials of degree 2 or less over Ω_e. To see this, notice that $1, x \in \mathscr{P}^e$ from example 1.47, and that $x^2 = (x_2^e)^2 N_2^e(x) + (x_1^e)^2 N_1^e(x) - (x_2^e - x_1^e)^2/4\, N_3^e(x)$, so $x^2 \in \mathscr{P}^e$.

The degrees of freedom of this element do not all have a simple interpretation: ϕ_1^e and ϕ_2^e are the values of f^e at x_1^e and x_2^e, but ϕ_3^e lacks one. The name *bubble* comes from the shape of N_3^e, which is zero at the two boundaries of the element.

What is a node? In general, whenever a degree of freedom of an element is the value of the function f^e or one of its derivatives at a location $\bar{x}$, we say that the element has a *node* at $\bar{x}$. When the degree of freedom is the value of the function, we depict it with a filled disk at $\bar{x}$. The symbol to depict the value of a derivative as a degree of freedom will be introduced later.

As a counterexample, degree of freedom ϕ_3^e in the P_1-element+bubble (Example 1.49) does not always correspond to the value of a function in the space at the midpoint between x_1^e and x_2^e, so such degree of freedom cannot be indicated by a node, c.f. Fig. 1.13.

The pictorial depiction of nodes in an element is a way to graphically indicate the degrees of freedom of an element, and it is commonly used in the finite element literature.

It is important to retain a strict distinction between the vertices of an element, which are used to define the geometry of the element domain, and the degrees of freedom indicated by the nodes, which are used to define functions over the element domain.

Examples:

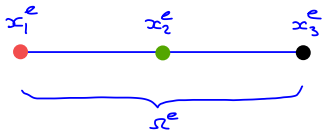
1.50 P_0**-element.** The simplest element space is that of a single constant function over the domain of the element Ω_e, or in a fancy way, a polynomial in $\mathbb{P}_0(\Omega_e)$. A single basis function is needed,

$$N_1^e(x) = 1, \tag{1.83}$$

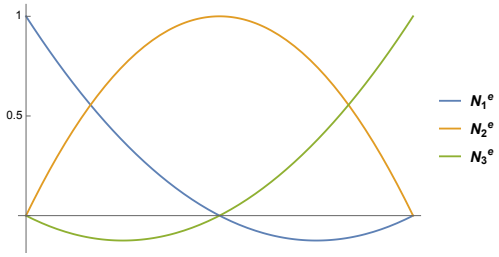
so the space $\mathscr{P}^e$ has one degree of freedom.

It can be represented with a node at the center of the element, or elsewhere.

1.51 P_2**-element.** The second most common element has the following basis functions over an element domain $\Omega_e = [x_1^e, x_3^e]$, with $x_2^e = (x_1^e + x_3^e)/2$,



$$\begin{aligned} N_1^e(x) &= \frac{(x-x_2^e)(x-x_3^e)}{(x_1^e-x_2^e)(x_1^e-x_3^e)}, \\ N_2^e(x) &= \frac{(x-x_1^e)(x-x_3^e)}{(x_2^e-x_1^e)(x_2^e-x_3^e)}, \\ N_3^e(x) &= \frac{(x-x_1^e)(x-x_2^e)}{(x_3^e-x_1^e)(x_3^e-x_2^e)}. \end{aligned} \tag{1.84}$$



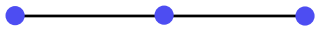
This is a linearly independent set of functions, a fact that follows from similar arguments to those for the P_1-element.

The space $\mathscr{P}^e$ has 3 degrees of freedom, it is also the space $\mathbb{P}_2(\Omega_e)$ of all polynomials of degree 2 or less over Ω_e. This is a set of three linearly independent quadratic polynomials, precisely the dimension of $\mathbb{P}_2(\Omega_e)$, and hence they need to span $\mathbb{P}_2(\Omega_e)$.

In this element we have that $N_1^e(x) + N_2^e(x) + N_3^e(x) = 1$ for any $x \in \Omega_e$. To see this, let $f^e(x) = N_1^e(x) + N_2^e(x) + N_3^e(x)$, and notice that $f^e(x) = 1$ for $x \in \{x_1^e, x_2^e, x_3^e\}$ and that $f^e \in \mathscr{P}^e$. Thus, $f^e(x) - 1$ is a quadratic polynomial that is equal to 0 at these three points. We conclude then

that $f^e(x) - 1 = 0$ for all $x \in \Omega_e$. You can also check this by simply adding the three expressions in (1.84).

Since the basis functions satisfy that $N_a^e(x_b^e) = \delta_{ab}$, the degrees of freedom in the space are the values of a function $f^e \in \mathscr{P}^e$ at x_1^e, x_2^3 and x_3^e. Because these three spatial locations have the value of a function therein as a degree of freedom, the element has three nodes, each one represented with a filled disc, one at each vertex and one at x_3^e, to wit:



1.52 **P_k-element, for $k = 1, \ldots$.** The Lagrange P_k-elements, often known simply as the P_k-elements, are a generalization of the P_0, P_1 and P_2 elements to any positive integer k. To simplify notation, we will denote the position of the vertices of the element by $z_1 < z_2$, so that $\Omega_e = [z_1, z_2]$. Additionally, we introduce $k+1$ nodes at locations

$$x_a^e = z_1 + (a-1)\frac{(z_2 - z_1)}{k}$$

for $a = 1, \ldots, k+1$. The basis functions for this element are

$$N_a^e(x) = \frac{\Pi_{b=1, b\neq a}^{k+1}(x - x_b^e)}{\Pi_{b=1, b\neq a}^{k+1}(x_a^e - x_b^e)} \tag{1.85}$$

for $a = 1, \ldots, k+1$. Each of these functions is a polynomial of degree k, and as will see next, they form a linearly independent set of $k+1$ functions. Therefore, $\mathscr{P}^e = \text{span}(N_1^e, \ldots, N_{k+1}^e) = \mathbb{P}_k(\Omega_e)$, or the set of all polynomials of degree less or equal than k over Ω_e, since the number of linearly independent vectors functions is equal to the dimension of $\mathbb{P}_k(\Omega_e)$. The plots of these basis functions for $k = 3, 4, 5$ are shown in Fig. 1.14.

The first noteworthy feature of this set of functions is that

$$N_a^e(x_b^e) = \delta_{ab}, \tag{1.86}$$

so that the degrees of freedom of this element are the values of a function at $\{x_1^e, \ldots, x_{k+1}^e\}$. Therefore, this element has nodes at these locations. To see (1.86), notice that if $a \neq b$, then x_b^e is a zero of the numerator of N_a^e. Instead, if $a = b$, then the numerator and denominator of (1.86) are equal, and hence $N_a^e(x_a^e) = 1$.

To see that this is a basis, consider $(\phi_1^e, \ldots, \phi_{k+1}^e) \in \mathbb{R}^{k+1}$ such that

$$f(x) = \phi_1^e N_1^e(x) + \ldots + \phi_{k+1}^e N_{k+1}^e(x) = 0 \qquad \forall x \in \Omega_e.$$

Then, for any $a = 1, \ldots, k+1$, $f(x_a^e) = \phi_a^e N_a^e(x_a^e)$, because $N_b^e(x_a^e) = 0$ for $a \neq b$, from where $\phi_a^e = 0$. It then follows that this is a basis.

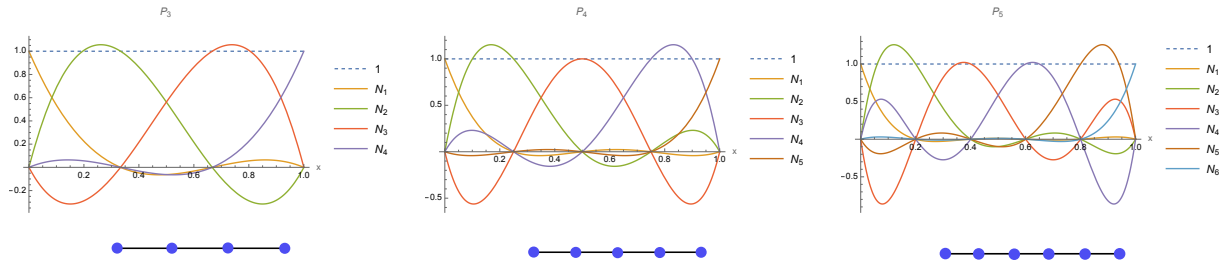


Figure 1.14 Shape functions for elements P_3, P_4 and P_5 over $\Omega_e = [0,1]$, together with the constant function $f(x) = 1$ for comparison. The graphical depiction of each element is shown as well.

The final interesting property of this basis is that if $f \in \mathbb{P}_k(\Omega_e)$, then

$$f(x) = f(x_1^e)N_1^e(x) + \ldots + f(x_{k+1}^e)N_{k+1}^e(x) \qquad \forall x \in \Omega_e. \tag{1.87}$$

In particular, if $f(x) = 1$, then $N_1^e(x) + \ldots + N_{k+1}^e(x) = 1$ for all $x \in \Omega_e$. To see this, let $g(x) = f(x_1^e)N_1^e(x) + \ldots + f(x_{k+1}^e)N_{k+1}^e(x) - f(x)$. Notice then that $g(x_a^e) = 0$ for $a = 1, \ldots, k+1$, and that $g(x)$ is a polynomial of degree less or equal than k that is equal to zero at $k+1$ distinct points. This can only happen if $g(x) = 0$ for all $x \in \Omega_e$, from where (1.87) follows.

Elements in which all the degrees of freedom are values of the function at predefined locations in the element are called **Lagrange elements**. For example, the P_k-element is a Lagrange element, while the P_1-element+bubble is not.

1.3.3 Construction of Finite Element Spaces

Once we define a mesh over the interval Ω and element spaces on each element, we have what is called a **finite element mesh**. A vector space $\mathcal{W}_h$ of functions over the interval Ω can be constructed by defining a basis for it using the shape functions in each finite element.

What type of basis functions for $\mathcal{W}_h$ can we construct with the shape functions in each element? Let's look at some examples (we do not specify the entire basis yet). To this end, we consider a mesh of three P_1-elements in Fig. 1.15.

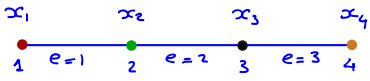
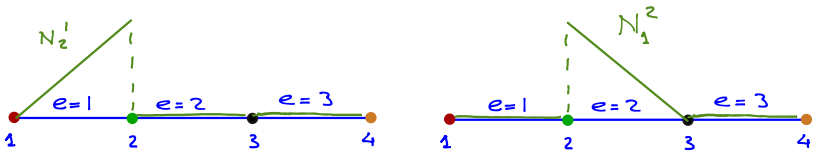


Figure 1.15

Examples:

1.53 Function N_2^1 could be a basis function, if we define it as equal to zero for points outside element $e = 1$. The function is discontinuous at x_2, so has two one-sided limits, $\lim_{x \to x_2^-} N_2^1(x) = 1$, and $\lim_{x \to x_2^+} N_2^1(x) = 0$. Similarly, N_2^1 could be a basis function, if we define it as equal to zero outside element $e = 2$, and it is discontinuous at x_2 as well. These two functions are sketched next:



1.54 The hat function N_2 can be constructed as the sum of N_1^2 and N_2^1, $N_2 = N_2^1 + N_1^2$, when each of them is defined as equal to zero outside elements $e = 1$ and $e = 2$, respectively, see Fig. 1.16. Because N_2^1 and N_1^2 are discontinuous at x_2, the value of N_2 at x_2 depends on what values each one of them takes at x_2. If we defined $N_2^1(x_2) = N_1^2(x_2) = 1$, then $N_2(x_2) = 2$, and N_2 would be discontinuous at x_2.

Instead, it is convenient to define

$$N_2(x) = N_2^1(x) + N_1^2(x) \qquad \text{for } x \neq x_2,$$

and define the value of N_2 at x_2 only if the two one-sided limits are the same. In this case they are, so

$$N_2(x_2) = \lim_{x \to x_2^-} N_2(x) = \lim_{x \to x_2^+} N_2(x) = \lim_{x \to x_2} N_2(x) = 1.$$

We will not consider cases in which the two one-sided limits are not equal. But if they were, the value of $N_2(x_2)$ need not be defined, and instead the method would use the values of the two one-sided limits. This is what is typically done in a class of finite element method called *Discontinuous Galerkin Methods*.

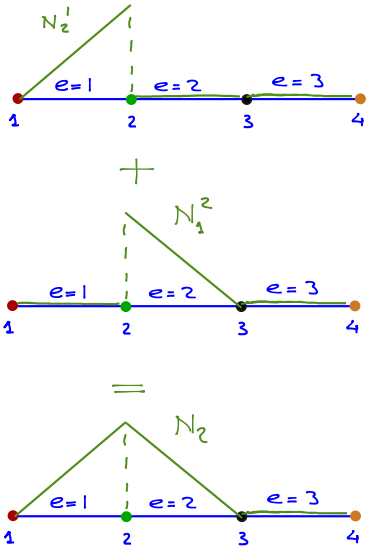


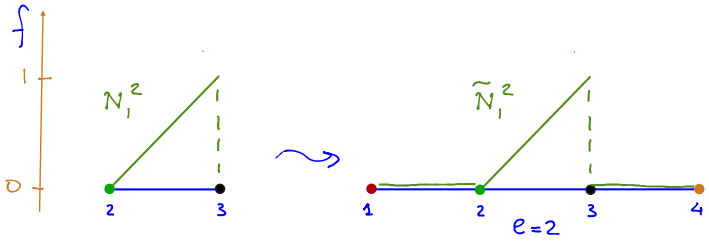
Figure 1.16

In general, to build a basis for $\mathcal{W}_h$, we proceed as follows:

1. **Extend Shape Functions by Zero.** For each element e in the mesh, we extend each basis function in the element by zero outside the element. More precisely, if we denote by $\tilde{N}_a^e \colon \Omega \to \mathbb{R}$ the extension-by-zero of the function $N_a^e \colon \Omega_e \to \mathbb{R}$, we can write

$$\tilde{N}_a^e(x) = \begin{cases} N_a^e(x) & x \in \Omega_e \\ 0 & x \notin \Omega_e. \end{cases} \tag{1.88}$$

See the example below.



In the following, we will not make an explicit distinction between $\tilde{N}_a^e$ and N_a^e, and simply use N_a^e for both.

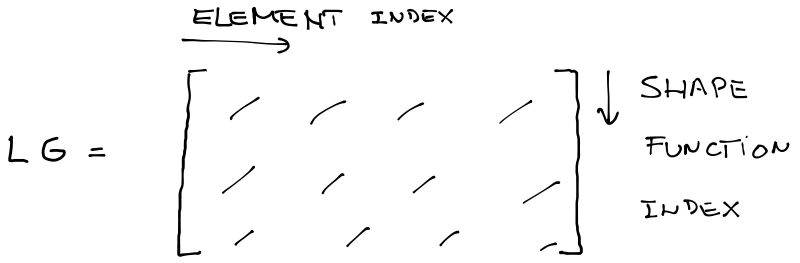
2. **Define a Local-to-Global Map.** We will define *every* basis functions N_A for $\mathcal{W}_h$, with $A \in \{1,\ldots,m\}$, by *adding* shape functions from one or more elements, with the condition that *each shape function in an element contributes to exactly one basis function in* $\mathcal{W}_h$. Then, $\mathcal{W}_h = \text{span}(N_1,\ldots,N_m)$ and its dimension is m. Symbolically, if N_A is the result of adding $r \geq 1$ shape functions, we can write

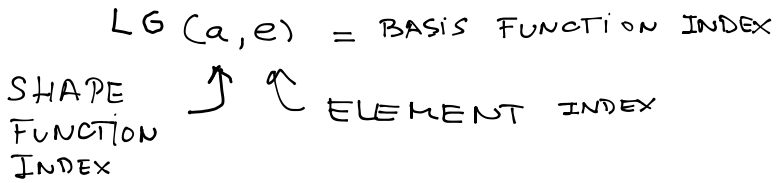
$$N_A = N_{a_1}^{e_1} + \ldots + N_{a_r}^{e_r},$$

and each shape function N_a^e appears in exactly one of such sums.

We specify what basis function N_A a shape function N_a^e contributes to through a **local-to-global map**.

In this class, the local-to-global map is indicated with an $n_{\text{el}} \times k$ matrix termed LG, so that $A = \text{LG}(a,e)$ is the entry in row a and column e in LG. Graphically:





The entry $\text{LG}(a,e)$ in the matrix is a number in $\{1,\ldots,m\}$ that defines that shape function N_a^e should be added when constructing basis function N_A. Alternatively, function N_A is obtained by adding all shape functions with index A in the matrix LG. Because each shape function contributes to exactly one basis function, every entry in the LG matrix is well-defined. The range of LG needs to be $\{1,\ldots,m\}$, so that all basis functions in $\mathcal{W}_h$ are constructed in this way.

The name local-to-global map originates in the fact that it maps the indices of shape functions in each element, defined only locally over the domain of the element to form the element space, to indices of basis functions whose domain is the entire interval Ω to form the space $\mathcal{W}_h$.

3. **Add Shape Functions.** Because we are adding functions that are potentially discontinuous at the interfaces between neighboring elements, some care is needed in the definition of the basis functions for $\mathcal{W}_h$, as in Example 1.54.

   With the local-to-global map, we define the basis functions for $\mathcal{W}_h$. For any $A \in \{1, \dots, m\}$, let $N_A \colon \Omega \to \mathbb{R}$ be given by

$$N_A(x) = \sum_{\{(a,e) \mid \mathsf{LG}(a,e)=A\}} N_a^e(x), \tag{1.89a}$$

   for $x \neq x_i$ and

$$N_A(x_i) = \lim_{x \to x_i} N_A(x), \tag{1.89b}$$

   for all $i \in \{1, \dots, n_{\text{el}} + 1\}$.

   The set

$$\{(a,e) \mid \mathsf{LG}(a,e) = A\} \tag{1.90}$$

   says that we should seek all pairs (a,e) of shape function index a and element number e that are mapped to basis function index A. In other words, we should add all shape functions that contribute to basis function N_A. Because each basis function N_A is the sum of some shape functions, the set in 1.90 is never empty. Finally, if the limit in (1.89b) is not defined, the value of $N_A(x_i)$ is left undefined.

   To remember that in performing this special sum we add the values everywhere except at the nodes as in (1.89a), and evaluate limits to find their values at the nodes as in (1.89b), we introduce a special name and symbol for it. We call it the **broken sum**, $\mathring{+} \colon \mathcal{W}_h \times \mathcal{W}_h \to \mathcal{W}_h$, so that for $f_h, g_h \in \mathcal{W}_h$,

$$(f_h \mathring{+} g_h)(x) = f_h(x) + g_h(x), \qquad x \neq x_i, \tag{1.91a}$$

$$\tag{1.91b}$$

   and

$$(f_h \mathring{+} g_h)(x_i) = \lim_{x \to x_i} (f_h \mathring{+} g_h)(x). \tag{1.91c}$$

   Consistently with the new symbol, the broken summation sum is $\mathring{\sum}$.

   With this notation, we can write

$$N_A = \mathring{\sum_{\{(a,e) \mid \mathsf{LG}(a,e)=A\}}} N_a^e. \tag{1.92}$$

☞ Basis functions for the finite element space are called *global basis functions*, and basis functions for an element space are called *local basis functions*, or shape functions. Global basis functions are defined in the entire domain of the problem, while local basis functions are defined only over an element.

Notice that we are now regularly referring to different sets of basis functions in the same context: the basis functions for $\mathcal{W}_h$ and the basis functions for the element spaces $\mathcal{P}^e$, or shape functions. To distinguish them, the basis functions for $\mathcal{W}_h$, whose domain is the entire interval, are called **global basis functions**. Conversely, basis functions for the element spaces $\mathcal{P}^e$, defined only over the element domain, are called **local basis functions**.

As a convention and whenever possible, uppercase letters will be used for indices of global degrees of freedom or basis functions, while local degrees of freedom or basis functions will use indices that are lowercase letters.

Examples:

1.55 **A space with discontinuous functions.** The simplest basis to define a space $\mathcal{W}_h$ over the mesh in Fig. 1.15 is the one in which each shape function defines a single global basis function. In this case, $m = k \times n_{\text{el}} = 2 \times 3 = 6$, and a local-to-global map is a 2×3 matrix that can be defined as

$$\mathsf{LG} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}. \tag{1.93}$$

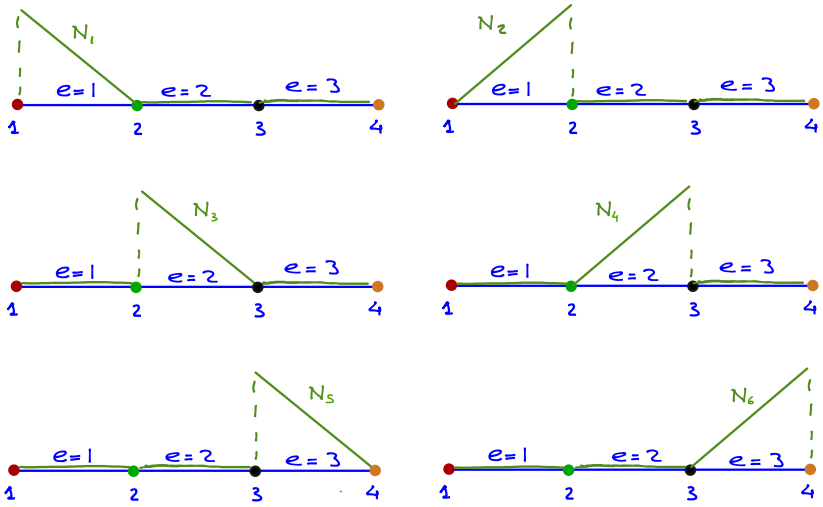
For example,

$$\mathsf{LG}(2,1) = 2,$$
$$\mathsf{LG}(1,3) = 5,$$
$$\mathsf{LG}(2,3) = 6.$$

With this local-to-global map, each global basis function index appears only once, so the basis functions for $\mathcal{W}_h$ can be written as

$$N_1 = N_1^1, \quad N_2 = N_2^1$$
$$N_3 = N_1^2, \quad N_4 = N_2^2$$
$$N_5 = N_1^3, \quad N_6 = N_2^3.$$

A sketch of the global basis functions is



An example of a function defined on this space is

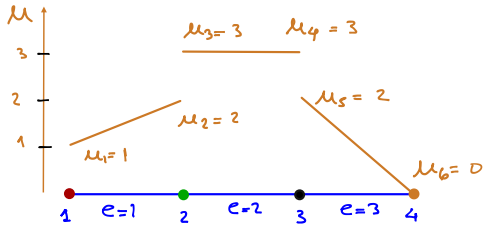
$$u = 1\, N_1 \mathring{+} 2\, N_2 \mathring{+} 3\, N_3 \mathring{+} 3\, N_4 \mathring{+} 2\, N_5 \mathring{+} 0\, N_6,$$

☞ As defined by the broken sum, values at the discontinuities are left undefined, since the limit in (1.91c) does not exist. The two one-sided limits do exist. We will not need to worry about this, since we are going to require functions in the finite element space to

with components in this basis

$$U = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \\ 2 \\ 0 \end{bmatrix}.$$

These components can be interpreted as the one-sided limits of the function at the nodes, as sketched next:



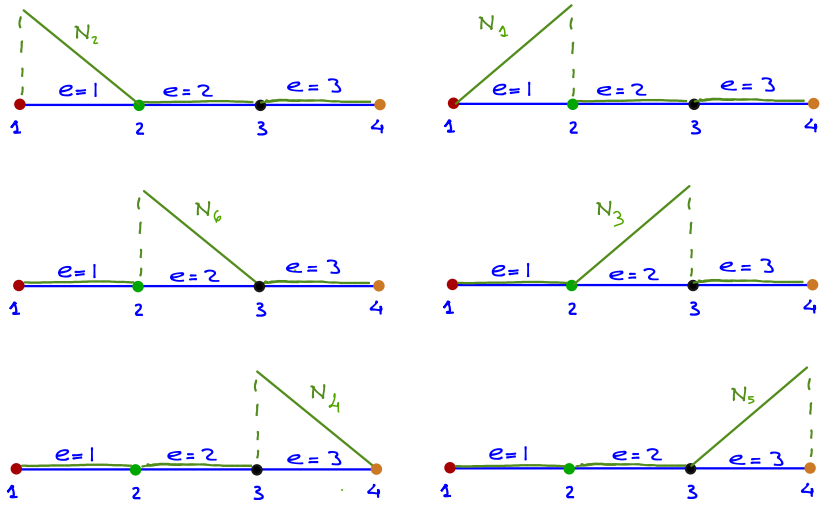
The index we assigned to each basis function of $\mathcal{W}_h$ is immaterial, since $\mathcal{W}_h$ does not change upon altering the name of each basis function. For example, we could have used the following local-to-global map

$$\mathsf{LG} = \begin{bmatrix} 2 & 6 & 4 \\ 1 & 3 & 5 \end{bmatrix}. \tag{1.94}$$

In this case, the global basis functions can be written as

$$N_1 = N_2^1, \quad N_2 = N_1^1$$
$$N_3 = N_2^2, \quad N_4 = N_1^3$$
$$N_5 = N_2^3, \quad N_6 = N_1^2.$$

A sketch of the global basis functions with this new label is



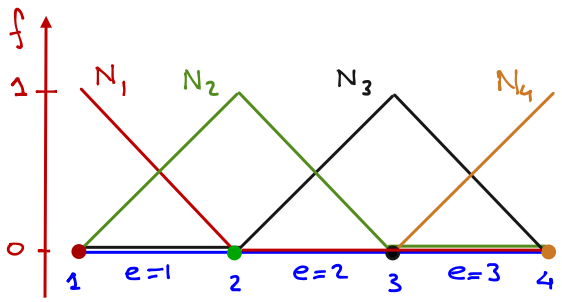
The function u is now written as

$$\begin{aligned} u &= 1\, N_2 \mathring{+} 2\, N_1 \mathring{+} 3\, N_6 \mathring{+} 3\, N_3 \mathring{+} 2\, N_4 \mathring{+} 0\, N_5 \\ &= 2\, N_1 \mathring{+} 1\, N_2 \mathring{+} 3\, N_3 \mathring{+} 2\, N_4 \mathring{+} 0\, N_5 \mathring{+} 3\, N_6, \end{aligned}$$

and its components are

$$U = \begin{bmatrix} 2 \\ 1 \\ 3 \\ 2 \\ 0 \\ 3 \end{bmatrix}.$$

1.56 **The simplest space of continuous functions.** The basis made of hat functions over the mesh in Fig. 1.15 is:



We can then build each basis function of $\mathscr{W}_h$ as a sum of shape functions as follows:

$$\begin{aligned} N_1 &= N_1^1 \\ N_2 &= N_2^1 \mathring{+} N_1^2 \\ N_3 &= N_2^2 \mathring{+} N_1^3 \\ N_4 &= N_2^2. \end{aligned} \tag{1.95}$$

This space has dimension $m = 4$, and the local-to-global map is defined by the following 2×3 matrix:

$$\mathsf{LG} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}. \tag{1.96}$$

What is then the set

$$\{(a,e) \mid \mathsf{LG}(a,e) = 2\}?$$

It is $\{(2,1),(1,2)\}$, that is, the rows and columns of the two entries equal to 2 in the LG matrix in (1.96).

What about the set

$$\{(a,e) \mid \mathsf{LG}(a,e) = 4\}?$$

It is the set $\{(2,3)\}$. You can now check that (1.89a) reduces to (1.95) for this example.

Examples of functions in this space are shown next:

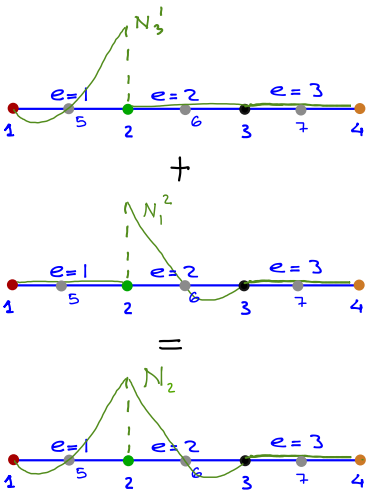
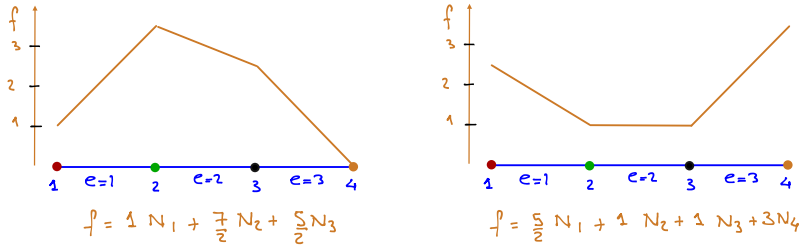


Figure 1.17



1.57 **A space of continuous piecewise quadratic functions.** Consider the mesh of Fig. 1.15 with P_2-elements, c.f. (1.84). A space $\mathscr{W}_h$ of continuous functions that are polynomials of degree less or equal than 2 over each element can be built with the basis functions sketched in Fig. 1.18. These basis functions can be written as

$$
\begin{aligned}
N_1 &= N_1^1, \\
N_2 &= N_3^1 \mathring{+} N_1^2 \\
N_3 &= N_3^2 \mathring{+} N_1^3 \\
N_4 &= N_3^3 \\
N_5 &= N_2^1, \\
N_6 &= N_2^2, \\
N_7 &= N_2^3.
\end{aligned}
$$

For example, the construction of N_2 is sketched in Fig. 1.17.

The space has dimension $m = 7$, and it is obtained from the local-to-global map given by the $k \times n_{\text{el}} = 3 \times 3$ matrix

$$
\mathsf{LG} = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 2 & 3 & 4 \end{bmatrix}.
$$

For example,

$$
\mathsf{LG}(2,2) = 6, \qquad \mathsf{LG}(3,1) = 2.
$$

It is clear from Examples 1.55 and 1.56 that different local-to-global maps can be defined over the same finite element mesh, in this case a mesh of 3 P_1-elements. Each combination of a finite element mesh and a local-to-global map defines a (potentially different) space $\mathscr{W}_h$. This is a very general and flexible framework, which starting from the definition of finite elements enables the construction of very rich and varied vector spaces of functions.

Similarly to the basis functions, the degrees of freedom of the element space $\mathscr{P}^e$ are labeled **local degrees of freedom**, while those of $\mathscr{W}_h$ are called **global degrees of freedom**. We see next that, by construction, *the local-to-global map is also a map from the index of the local degree of freedom to the index of the global*

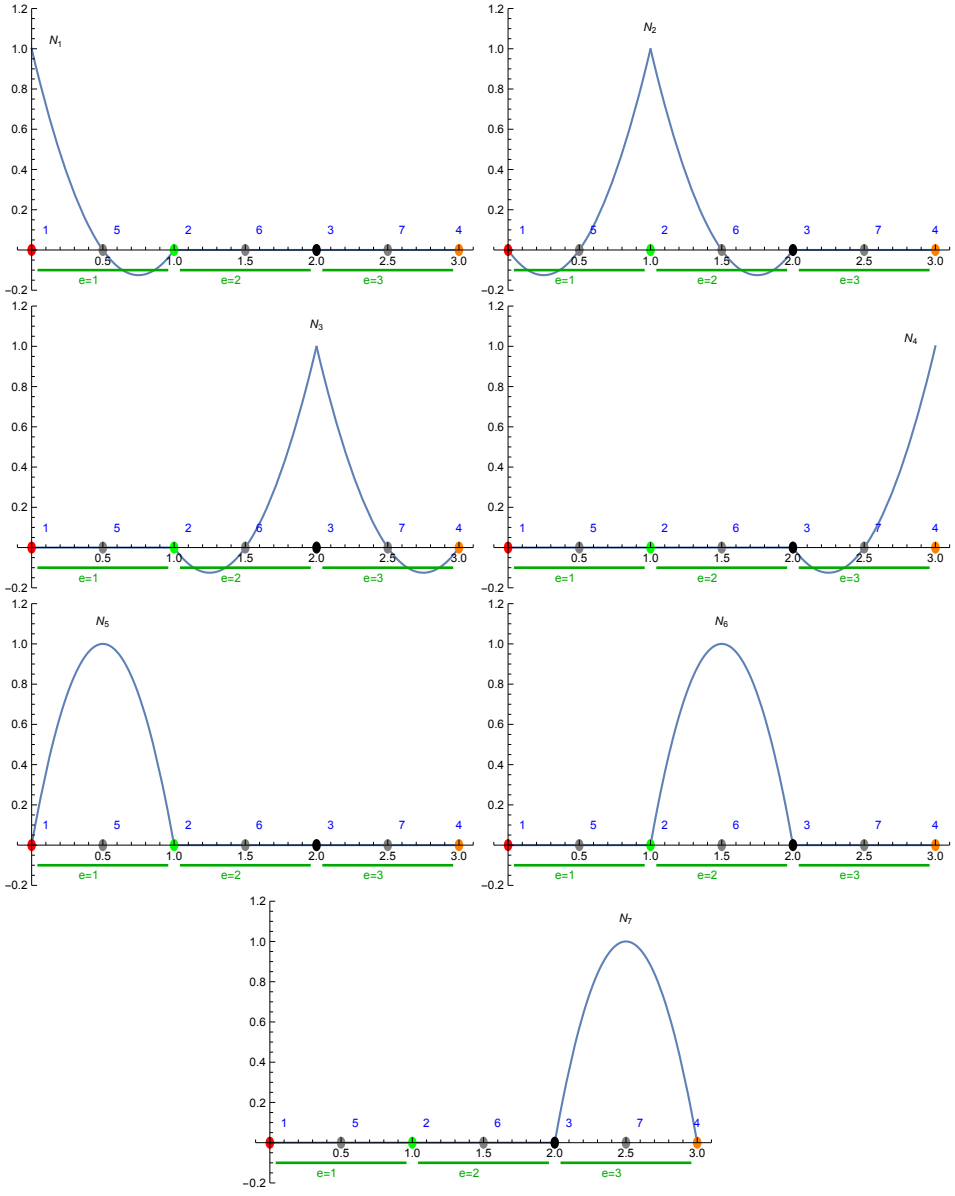


Figure 1.18 Basis functions for a finite element space of 3 P_2-elements.

degree of freedom. Specifically, consider a function $u \in \mathscr{W}_h$ defined by the values $(u_1, \ldots, u_m)$ of the global degrees of freedom, namely,

$$u = \sum_{A=1}^{m} u_A N_A,$$

then

$$u = \sum_{e=1}^{n_{\text{el}}} \sum_{a=1}^{k} u_{\mathsf{LG}(a,e)} N_a^e.$$

Therefore, for any element e, the function $f^e : \Omega_e \to \mathbb{R}$ defined by

$$f^e(x) = u(x) \qquad x \in \Omega_e,$$

belongs to $\mathscr{P}^e$. Its components in the local basis are

$$\phi_a^e = u_{\mathsf{LG}(a,e)}. \tag{1.97}$$

The function f^e is called the **restriction** of u to element e.

So each set of values for the global degrees of freedom define a set of values for the local degrees of freedom to describe the same function over an element. The process of obtaining the local degrees of freedom from the global ones through (1.97) is called **localization**.

Map between local and global degrees of freedom (1.97)**.**

Consider $u \in \mathscr{W}_h$, then

$$\begin{aligned}
u(x) &= \sum_{A=1}^{m} u_A N_A(x) \\
&= \sum_{A=1}^{m} u_A \sum_{\{(a,e)|\mathsf{LG}(a,e)=A\}} N_a^e(x) && \text{from (1.92)} \\
&= \sum_{A=1}^{m} \sum_{\{(a,e)|\mathsf{LG}(a,e)=A\}} u_{\mathsf{LG}(a,e)} N_a^e(x) && \text{from definition of } \mathsf{LG} \\
&= \sum_{e=1}^{n_{\text{el}}} \sum_{a=1}^{k} u_{\mathsf{LG}(a,e)} N_a^e(x) && \text{see below}
\end{aligned}$$

The last step uses the fact that in spanning all values of A with the first sum, the two sums together effectively guarantee that all pairs (a,e) will be added exactly once, since $\{1,\ldots,m\}$ is precisely the range of LG, so its pre-image is the entire domain. This is again a consequence of the fact that every shape function contributes to exactly one global basis function, and all global basis functions are built in this way. It is also a consequence of defining global basis functions as sums of shape functions. Had global basis functions been defined as more general linear combinations of shape functions, each local degree of freedom would not be directly equal to a global degree of freedom.

So, because of the construction of the basis functions, it follows that: (a) the function u restricted to element e belongs to $\mathscr{P}^e$, (b) the values of the degrees of freedom of u restricted to element e are $\phi_a^e = u_{\mathsf{LG}(a,e)}$, so the local-to-global map also maps the local degrees of freedom to local ones.

Finite Element Spaces of Continuous Functions. The choice of the finite element space $\mathcal{W}_h$ and finite element basis to use with Galerkin Method is conditioned by the requirement $\mathcal{W}_h \subset \mathcal{W}$ and by how easy it is to find a basis for $\mathcal{V}_h = \mathcal{V} \cap \mathcal{W}_h$. Commonly, $\mathcal{W}$ requires functions to be continuous, so $\mathcal{W}_h$ cannot be the span of a finite element basis with discontinuous functions (since a discontinuous basis function does not belong to $\mathcal{W}$). If non-zero boundary conditions need to be imposed, then it should be possible to identify constrained indices in the (finite element) basis for $\mathcal{W}_h$ to select the basis functions for $\mathcal{V}_h$. Standard finite element bases will make this trivial.

For example, the space of hat functions in Example 1.56 has a continuous basis, so its span $\mathcal{W}_h$ contains only continuous functions. Additionally, if $w_h \in \mathcal{W}_h$, then $w_h(x_1) = w_1$ and $w_h(x_4) = w_4$, so it is enough to constrain indices 1 and/or 4 to impose boundary conditions on the value of w_h at either endpoint. The space of continuous piecewise quadratic functions in Example 1.57 has the same properties.

Some problems require only continuity of functions in $\mathcal{W}_h$, but others require continuity of one or more derivatives, such as the fourth-order problems in §1.4. Constructing the bases for such spaces can be a delicate task for two and three dimensional domains, but it is rather simple in one dimension.

1.3.4 Assembly of the Stiffness Matrix and Load Vector

The computation of the stiffness matrix and load vector generally involves the calculation of integrals over the domain, such as those involved in the bilinear form and linear functional. For example, for model Problem 1.3 with $b(x) = 0$,

$$a(u,v) = \int_\Omega \left[k(x)u'(x)v'(x) + c(x)u(x)v(x)\right]\,dx, \tag{1.98a}$$

$$\ell(v) = k(L)d_L v(L) + \int_\Omega f(x)v(x)\,dx. \tag{1.98b}$$

This task highlights key functions that elements provide: The decomposition of the domain into elements afford us the ability to decompose integrals over the domain as a sum of integrals over elements, construct *elemental stiffness matrices* and *elemental load vectors*, and *"assemble"* them over the mesh to form the stiffness matrix and load vector of the problem.

Let's have a brief look at the main ideas of what we will be discussing. Consider again the simplest space of continuous functions over a mesh with 3 elements and basis functions in Fig. 1.19. If, for example, we wanted to compute the

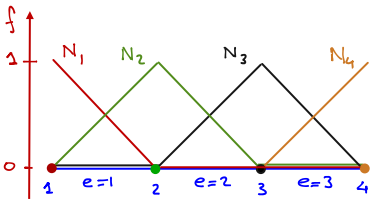


Figure 1.19

stiffness matrix entry $K_{33} = a(N_3, N_3)$, we could split the integral in (1.98a) as

$$\begin{aligned} K_{33} &= \int_\Omega \left[k(x)N_3'(x)N_3'(x) + c(x)N_3(x)N_3(x)\right]\, dx \\ &= \underbrace{\int_{\Omega_2} \left[k(x)\left(N_2^2\right)'(x)\left(N_2^2\right)'(x) + c(x)N_2^2(x)N_2^2(x)\right]\, dx}_{K_{22}^2} \\ &\quad + \underbrace{\int_{\Omega_3} \left[k(x)\left(N_1^3\right)'(x)\left(N_1^3\right)'(x) + c(x)N_1^3(x)N_1^3(x)\right]\, dx}_{K_{11}^3} \end{aligned} \tag{1.99}$$

There is no need to compute an integral over element $e = 1$, given that $N_3(x) = 0$ for $x \in \Omega_1$ and hence the value of the integral is zero. Only elements 2 and 3 contribute non-zero values to K_{33}. The contribution of each element, K_{22}^2 and K_{11}^3, are entries of the element stiffness matrices for elements 2 and 3. Over each element we can replace the global basis functions (N_3) by the local ones (N_2^2 and N_1^3), or shape functions. The value of K_{33} is obtained by *accumulating* the contributions to its value by all elements in the mesh. This is what is called *assembling* K_{33}.

For problems with two and three-dimensional domains, computing in this way simplifies the integration problem enormously, since it is only necessary to learn how to compute integrals over each element, rather than over arbitrarily-shaped domains.

Element Stiffness Matrix and Element Load Vector. The element stiffness matrix K^e and element load vector F^e are inspired and emerge from the decomposition of the integrals involved in the definition of the bilinear form and linear functional as sums of integrals over elements. Symbolically, we can write

$$\int_\Omega (\cdot) = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega_e} (\cdot), \tag{1.100}$$

where Ω is the domain over which an integral is performed. Many commonly found bilinear forms and linear functionals can be written as

$$\begin{aligned} a(u,v) &= \sum_{e=1}^{n_{\text{el}}} a^e(u,v) \quad \text{where} \quad a^e(u,v) &= \int_{\Omega_e} \ldots\, dx \\ \ell(v) &= \sum_{e=1}^{n_{\text{el}}} \ell^e(v) \quad \text{where} \quad \ell^e(v) &= \int_{\Omega_e} \ldots\, dx, \end{aligned} \tag{1.101}$$

For model Problem 1.3 with $b(x) = 0$ and $d_L = 0$ (c.f. (1.98)) this is

$$\begin{aligned} a(u,v) &= \sum_{e=1}^{n_{\text{el}}} \underbrace{\int_{\Omega_e} \left[k(x)u'(x)v'(x) + c(x)u(x)v(x)\right]\, dx}_{=a^e(u,v)} = \sum_{e=1}^{n_{\text{el}}} a^e(u,v), \\ \ell(v) &= \sum_{e=1}^{n_{\text{el}}} \underbrace{\int_{\Omega_e} f(x)v(x)\, dx}_{\ell^e(v)} = \sum_{e=1}^{n_{\text{el}}} \ell^e(v). \end{aligned} \tag{1.102}$$

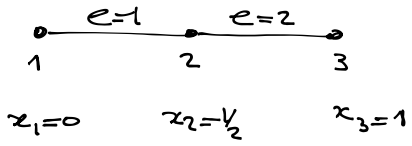
We are now in position to define K^e and F^e as

$$K^e_{ab} = a^e(N^e_b, N^e_a) \qquad \text{Element Stiffness Matrix} \tag{1.103a}$$
$$F^e_a = \ell^e(N^e_a) \qquad \text{Element Load Vector} \tag{1.103b}$$

for any $a, b = 1, \ldots, k$. Before we discuss how these are used to construct K and F, let's look at an example.

The definition of the element stiffness matrix and element load vector is also notable for what it does not define: the values of $a^e(N^{e_1}_b, N^{e_2}_a)$ and $\ell^e(N^{e_1}_a)$, for elements e, e_1 and e_2. The reason for this is that if either $e_1 \neq e$ or $e_2 \neq e$, then both values are identically zero, given that either $N^{e_1}_b$ and/or $N^{e_2}_a$ are zero in Ω_e. Since the values of any basis function N_A in Ω_e are exclusively defined by linear combinations of shape functions in an element, these are the only potentially non-zero contributions to the stiffness matrix or load vector, and hence the only ones included in the element stiffness matrix and element load vector. Therefore, out of the $m \times m$ combinations of basis functions (generally a large number), the only non-zero contributions of $a^e(N_A, N_B)$ are accounted for by the $k \times k$ elemental stiffness matrix (generally a much smaller number).

Example 1.58 We compute the element stiffness matrix K^e and element load vector F^e defined by (1.102) for every element in a simple case. Consider a mesh with two P_1 elements over the interval $[0, 1]$, and let $f(x) = 10$, $k(x) = 1$ and $c(x) = 3x$ for $x \in (0, 1)$.



We do not yet specify the local-to-global map LG because it is not needed to compute K^e and F^e; it will be needed later to build K and F.

Since we have two elements, we need to compute K^1, K^2, F^1 and F^2. From (1.102) and the values for k, c and f,

$$a^e(u, v) = \int_{\Omega_e} \left[u'(x)v'(x) + 3xu(x)v(x) \right] dx, \qquad \ell^e(v) = \int_e 10v \, dx. \tag{1.104}$$

The shape functions over each element of this finite element mesh are

$$N^1_1(x) = \frac{1/2 - x}{1/2}$$
$$N^1_2(x) = \frac{x}{1/2}$$
$$N^2_1(x) = \frac{1 - x}{1/2}$$
$$N^2_2(x) = \frac{x - 1/2}{1/2}.$$

Notice here the superindex N_a^e with $e = 1,2$ is the element index, and not exponentiation. To simplify the notation, we will also use $N_{,x}$ to indicate the derivative of N, instead of N'.

The element stiffness matrices follow as:

$$K_{ab}^1 = \int_0^{1/2} N_{a,x}^1 N_{b,x}^1 + 3x N_a^1 N_b^1 \, dx, \qquad K_{ab}^2 = \int_{1/2}^1 N_{a,x}^2 N_{b,x}^2 + 3x N_a^2 N_b^2 \, dx.$$

This results in

$$\begin{aligned}
K^1 &= \begin{bmatrix} \int_0^{1/2} \left(-\frac{1}{1/2}\right)\left(-\frac{1}{1/2}\right) + 3x \frac{1/2-x}{1/2}\frac{1/2-x}{1/2}\, dx & \int_0^{1/2} \left(-\frac{1}{1/2}\right)\left(\frac{1}{1/2}\right) + 3x \frac{1/2-x}{1/2}\frac{x}{1/2}\, dx \\ \int_0^{1/2} \left(\frac{1}{1/2}\right)\left(-\frac{1}{1/2}\right) + 3x \frac{x}{1/2}\frac{1/2-x}{1/2}\, dx & \int_0^{1/2} \left(\frac{1}{1/2}\right)\left(\frac{1}{1/2}\right) + 3x \frac{x}{1/2}\frac{x}{1/2}\, dx \end{bmatrix} \\
&= \begin{bmatrix} \frac{33}{16} & -\frac{31}{16} \\ -\frac{31}{16} & \frac{35}{16} \end{bmatrix}, \\
K^2 &= \begin{bmatrix} \int_{1/2}^1 \left(-\frac{1}{1/2}\right)\left(-\frac{1}{1/2}\right) + 3x \frac{1-x}{1/2}\frac{1-x}{1/2}\, dx & \int_{1/2}^1 \left(-\frac{1}{1/2}\right)\left(\frac{1}{1/2}\right) + 3x \frac{1-x}{1/2}\frac{x-1/2}{1/2}\, dx \\ \int_{1/2}^1 \left(\frac{1}{1/2}\right)\left(-\frac{1}{1/2}\right) + 3x \frac{x-1/2}{1/2}\frac{1-x}{1/2}\, dx & \int_{1/2}^1 \left(\frac{1}{1/2}\right)\left(\frac{1}{1/2}\right) + 3x \frac{x-1/2}{1/2}\frac{x-1/2}{1/2}\, dx \end{bmatrix} \\
&= \begin{bmatrix} \frac{37}{16} & -\frac{29}{16} \\ -\frac{29}{16} & \frac{39}{16} \end{bmatrix}.
\end{aligned}$$

The element load vectors are:

$$\begin{aligned}
F_a^1 = \int_0^{1/2} 10 N_a^1 \, dx \quad &\Longrightarrow F^1 = \begin{bmatrix} \int_0^{1/2} 10 \frac{1/2-x}{1/2}\, dx \\ \int_0^{1/2} 10 \frac{x}{1/2}\, dx \end{bmatrix} = \begin{bmatrix} 5/2 \\ 5/2 \end{bmatrix}, \\
F_a^2 = \int_{1/2}^1 10 N_a^2 \, dx \quad &\Longrightarrow F^2 = \begin{bmatrix} \int_{1/2}^1 10 \frac{1-x}{1/2}\, dx \\ \int_{1/2}^1 10 \frac{x-1/2}{1/2}\, dx \end{bmatrix} = \begin{bmatrix} 5/2 \\ 5/2 \end{bmatrix}.
\end{aligned} \tag{1.105}$$

To encompass the most common types of weak forms, we further need to consider load vectors that have contributions from evaluating test functions on the boundary of the domain. An example of such contribution is found in (1.98b), in the term

$$k(L) d_L v(L).$$

This term cannot be written as an integral over an element, and it involves the value of the test function v at the end of the interval, $x = L$. When terms of this type are present, the linear form of the problem is written as

$$\ell(v) = \sum_{e=1}^{n_{\text{el}}} \ell^e(v) + h_0 v(0) + h_L v(L), \tag{1.106}$$

where h_0 and h_L are two real numbers defined by the bilinear form. For our example in (1.98b), $h_0 = 0$ and $h_L = k(L) d_L$.

For simplicity, we will incorporate the potential contributions of the two boundary terms into the element load vector of the first and last elements of the

mesh. Specifically, we modify the definition (1.103a) of the element load vector for $e = 1$ and $e = n_{\text{el}}$ to become

$$\begin{aligned} F_a^1 &= \ell^1(N_a^1) + h_0 N_a^1(0) && \text{Element load vector for } e = 1 \\ F_a^e &= \ell^e(N_a^e) && \text{Element load vector for } e \neq 1, n_{\text{el}} \\ F_a^{n_{\text{el}}} &= \ell^1(N_a^{n_{\text{el}}}) + h_L N_a^{n_{\text{el}}}(L) && \text{Element load vector for } e = n_{\text{el}}, \end{aligned} \tag{1.107}$$

for $a = 1, \ldots, k$.

Assembly. The construction of the stiffness matrix K and load vector F in terms of the ones from the elements is called the finite element **assembly** operation. It is also called the **Direct Stiffness Method**. It is a result of the way global basis functions are constructed (c.f. (1.89a)). Recall that basis functions in $\mathcal{W}_h$ can be written in terms of the shape functions, or local basis functions, as

$$N_A(x) = \mathop{\sum\nolimits^{\circ}}_{\{(a,e)|\mathsf{LG}(a,e)=A\}} N_a^e(x). \tag{1.108}$$

for $A \in \{1, \ldots, m\}$. Based on this relation, we show below that

$$\boxed{F_A = \sum_{e=1}^{n_{\text{el}}} \sum_{\{a|\mathsf{LG}(a,e)=A\}} F_a^e, \qquad A \in \eta_a} \tag{1.109a}$$

and

$$\boxed{K_{AB} = \sum_{e=1}^{n_{\text{el}}} \sum_{\substack{\{a|\mathsf{LG}(a,e)=A\} \\ \{b|\mathsf{LG}(b,e)=B\}}} K_{ab}^e, \qquad A \in \eta_a, B \in \eta} \tag{1.109b}$$

These identities directly connect entries of the stiffness matrix and load vector in active indices' rows to entries in the corresponding contributions from the elements. Specifically, each such entry in K and F is obtained by accumulating the contributions of some elements in the mesh.

Example 1.59 We show next that (1.99) is a result of (1.109b). The local-to-global map for the mesh in Fig. 1.19 is (1.96), namely,

$$\mathsf{LG} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}.$$

To compute K_{33} from (1.109b), we need to identify the set

$$\{a \,|\, \mathsf{LG}(a,e) = 3\},$$

for each $e \in \{1,2,3\}$, since $A = B = 3$. By inspection of LG, it follows that

$$\begin{aligned} \{a \,|\, \mathsf{LG}(a,1) = 3\} &= \emptyset, \\ \{a \,|\, \mathsf{LG}(a,2) = 3\} &= \{2\}, \\ \{a \,|\, \mathsf{LG}(a,3) = 3\} &= \{1\}. \end{aligned}$$

Replacing with these indices in (1.109b), we obtain

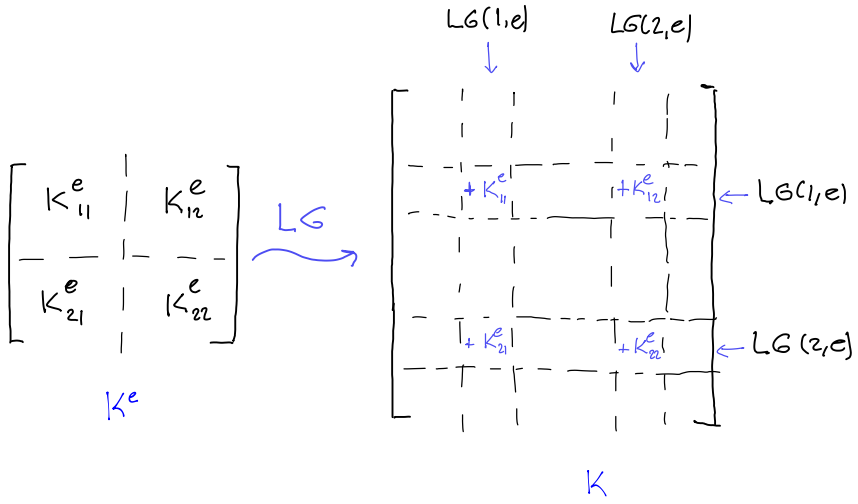
$$K_{33} = K^2_{22} + K^3_{11},$$

and we recover (1.99).

This example illustrates that computing a value of K_{AB} for a single pair of indices AB or the value of F_A for a single index A requires searching for those elements that contain indices of local degrees of freedom that are mapped to A and/or B by LG. However, since we want to compute K_{AB} and F_A for *all* active indices A and *all* indices B, it is more efficient to proceed in a different way:

1. Initially set $K = 0$ and $F = 0$.
2. Visit every element e in the mesh to compute K^e and F^e and
3. Add K^e_{ab} to $K_{\mathsf{LG}(a,e)\mathsf{LG}(b,e)}$ for all $a, b\{1,\dots,k\}$ if $\mathsf{LG}(a,e) \in \eta_a$, and
4. Add F^e_a to $F_{\mathsf{LG}(a,e)}$ for all $a \in \{1,\dots,k\}$ if $\mathsf{LG}(a,e) \in \eta_a$.

In this way, there is no need to search for which elements contribute to an entry. This is the distinguishing aspect of the assembly. A sketch of the way the local-to-global map defines where to add the element stiffness matrix is shown next:



Example 1.60 Let's revisit Example 1.58 to assemble a stiffness matrix and a load vector.

The mesh contains two elements, and to build the finite element space we need to specify the local-to-global map LG. For a space of continuous functions, this map is

$$\mathsf{LG} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix},$$

and the dimension of $\mathcal{W}_h$ is $m = 3$. Therefore, K is a 3×3 matrix, and F is a 3×1 matrix. Furthermore, we will assume that all indices are active [11], namely, $\eta_a = \eta = \{1, 2, 3\}$.

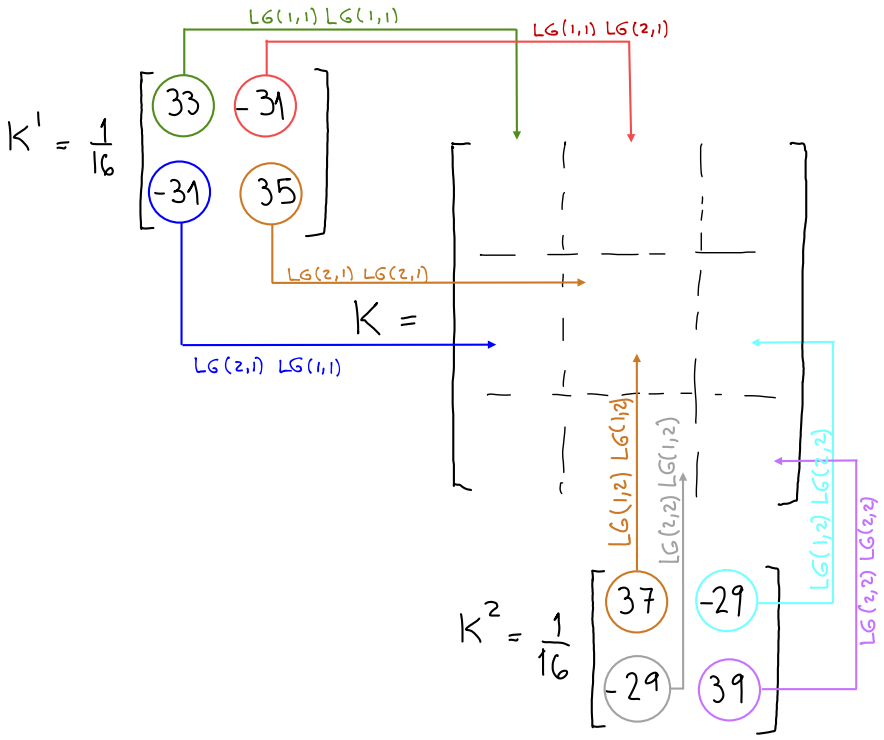
The element stiffness matrices are, from Example 1.58,

$$K^1 = \frac{1}{16}\begin{bmatrix} 33 & -31 \\ -31 & 35 \end{bmatrix} \qquad K^2 = \frac{1}{16}\begin{bmatrix} 37 & -29 \\ -29 & 39 \end{bmatrix},$$

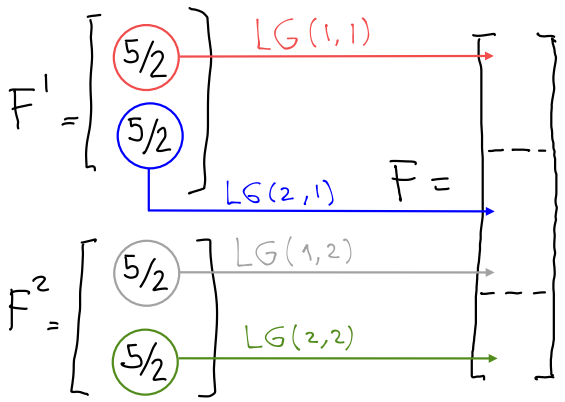
and the element load vectors are

$$F^1 = F^2 = \begin{bmatrix} 5/2 \\ 5/2 \end{bmatrix}.$$

The assembly process is sketched in the following figures



[11] this is the case when the boundary conditions are natural and *homogeneous*, i.e., equal to zero



The results of the assembly are

$$K = \frac{1}{16}\begin{bmatrix} 33 & -31 & 0 \\ -31 & 35+37 & -29 \\ 0 & -29 & 39 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 33 & -31 & 0 \\ -31 & 72 & -29 \\ 0 & -29 & 39 \end{bmatrix},$$

and

$$F = \begin{bmatrix} 5/2 \\ 5/2+5/2 \\ 5/2 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 5 \\ 5/2 \end{bmatrix}. \tag{1.110}$$

Insofar we have discussed how to assemble the rows of K and F whose indices are active. Rows with constrained indices are still defined by (1.70c), i.e., for $A \in \eta_g$ and $B \in \eta$,

$$K_{AB} = \delta_{AB}, \qquad F_A = \overline{u}_A.$$

Taking this into account, the pseudocode for the assembly procedure is

```
K = 0,  F = 0
FOR e ∈ {1, ..., n_el}
    FOR a ∈ {1, ..., k}
        IF LG(a,e) ∈ η_a
            FOR b ∈ {1, ..., k}
                K(LG(a,e), LG(b,e)) += K^e_ab
            END FOR
            F(LG(a,e)) += F^e_a
        END IF
    END FOR
END FOR
FOR A ∈ η_g
    K(A,A) = 1
    F(A)   = ū_A
END FOR
```

Example 1.61 Let's modify Example 1.60 and assemble K and F by assuming that $\eta_a = \{2,3\}$ and that $\overline{u}_1 = 4$, instead of the original assumption that all indices are active.

The results of the assembly are

$$K = \begin{bmatrix} 1 & 0 & 0 \\ -31/16 & 35/16+37/16 & -29/16 \\ 0 & -29/16 & 39/16 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -31/16 & 72/16 & -29/16 \\ 0 & -29/16 & 39/16 \end{bmatrix},$$

and

$$F = \begin{bmatrix} 4 \\ 5/2+5/2 \\ 5/2 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 5/2 \end{bmatrix}. \tag{1.111}$$

Example 1.62 As a final twist, let's examine the assembly for meshes with a different element type and with an increasing number of elements, so

as to observe the pattern of non-zero entries that emerges. For simplicity, we consider meshes with elements of equal length, and the bilinear form and linear functional in 1.102 with $f(x) = k(x) = c(x) = 1$. In this way, all element stiffness matrices and element load vectors are the same, since none of the three functions depends on x.

Since $\Omega = (0,1)$, the vertices for this mesh are at $x_i = (i-1)h$ for $i = 1,\dots,n_{\text{el}}+1$ and $h = 1/n_{\text{el}}$. We will consider a finite element space made of continuous piecewise quadratic functions (P_2-elements), and as customary, we add a node at the midpoint of each elements to indicate the third local degree of freedom. Thus, we set $x_{n_{\text{el}}+1+i} = (x_i + x_{i+1})/2$ for $i = 1,\dots,n_{\text{el}}$. Figure 1.17 shows the mesh for $n_{\text{el}} = 3$.

The element stiffness matrix is a 3×3 matrix, the load vector has length 3, and are computed as

$$K^e_{ab} = \int_{x_e}^{x_{e+1}} \left[N^e_{a,x}(x) N^e_{b,x}(x) + N_a(x) N_b(x) \right] dx,$$
$$F^e_a = \int_{x_e}^{x_{e+1}} N_a(x)\, dx,$$

where the shape functions are those in (1.84). We explicitly show the computation for two entries of K^e, and leave the rest for the reader to verify. From (1.84), we will use that $x^e_1 - x^e_2 = h$, $x^e_3 - x^e_1 = x^e_2 - x^e_3 = h/2$, and that

$$\begin{aligned} N^e_1\left(x^e_1 + \xi h\right) &= (\xi-1)(2\xi-1), & N^e_{1,x}(x^e_1 + h\xi) &= \frac{4\xi-3}{h} \\ N^e_3\left(x^e_1 + \xi h\right) &= 4(1-\xi)\xi, & N^e_{3,x}(x^e_1 + h\xi) &= \frac{4-8\xi}{h} \end{aligned} \tag{1.112}$$

for $\xi \in [0,1]$. Then,

$$\begin{aligned} K^e_{11} &= \int_{x_e}^{x_{e+1}} \left[N^e_{1,x}(x) N^e_{1,x}(x) + N_1(x) N_1(x) \right] dx \\ &= \int_0^1 \left[N^e_{1,x}(x_e+\xi h) N^e_{1,x}(x_e+\xi h) + N_1(x_e+\xi h) N_1(x_e+\xi h) \right] h\, d\xi \qquad \text{Change of variables } \xi = \tfrac{x-x_e}{h} \\ &= \int_0^1 \left[\frac{(4\xi-3)^2}{h^2} + (\xi-1)^2(2\xi-1)^2 \right] h\, d\xi \qquad \text{From (1.112)} \\ &= \frac{7}{3h} + \frac{2h}{15}. \\ K^e_{13} &= \int_{x_e}^{x_{e+1}} \left[N^e_{1,x}(x) N^e_{3,x}(x) + N_1(x) N_3(x) \right] dx \\ &= \int_0^1 \left[N^e_{1,x}(x_e+\xi h) N^e_{3,x}(x_e+\xi h) + N_1(x_e+\xi h) N_3(x_e+\xi h) \right] h\, d\xi \qquad \text{Change of variables } \xi = \tfrac{x-x_e}{h} \\ &= \int_0^1 \left[\frac{(4\xi-3)(4-8\xi)}{h^2} - 4(\xi-1)^2(2\xi-1)\xi \right] h\, d\xi \qquad \text{From (1.112)} \\ &= -\frac{8}{3h} + \frac{h}{15}. \end{aligned}$$

Proceeding with the computation, the element stiffness matrix is then

$$K^e = \begin{bmatrix} \frac{7}{3h} + \frac{2h}{15} & \frac{1}{3h} - \frac{h}{30} & -\frac{8}{3h} + \frac{h}{15} \\ \frac{1}{3h} - \frac{h}{30} & \frac{7}{3h} + \frac{2h}{15} & -\frac{8}{3h} + \frac{h}{15} \\ -\frac{8}{3h} + \frac{h}{15} & -\frac{8}{3h} + \frac{h}{15} & \frac{16}{3h} + \frac{8h}{15} \end{bmatrix}, \tag{1.113}$$

and the element load vector is

$$F^e = \begin{bmatrix} \frac{h}{6} \\ \frac{h}{6} \\ \frac{2h}{3} \end{bmatrix}. \tag{1.114}$$

Next, we assemble the stiffness matrix and load vector for a mesh with 3, 6, and 9 elements of the same length, with $\eta_c = \{1\}$ and $\overline{u}_1 = -1$. As always, we need to decide how to index the global degrees of freedom. In this case, it is customary to adopt the number of the node as the index of the global degree of freedom, so the local-to-global maps for each case are

$$\begin{aligned} n_{\text{el}} = 3, \quad \mathsf{LG} &= \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix} \\ n_{\text{el}} = 6, \quad \mathsf{LG} &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 & 12 & 13 \end{bmatrix} \\ n_{\text{el}} = 9, \quad \mathsf{LG} &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 \end{bmatrix}. \end{aligned} \tag{1.115}$$

When $n_{\text{el}} = 3$, $h = 1/3$, the stiffness matrix is

$$K = \frac{1}{90} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 89 & 1268 & 89 & 0 & -718 & -718 & 0 \\ 0 & 89 & 1268 & 89 & 0 & -718 & -718 \\ 0 & 0 & 89 & 634 & 0 & 0 & -718 \\ -718 & -718 & 0 & 0 & 1456 & 0 & 0 \\ 0 & -718 & -718 & 0 & 0 & 1456 & 0 \\ 0 & 0 & -718 & -718 & 0 & 0 & 1456 \end{bmatrix},$$

and the (transpose of the) force vector is

$$F^{\mathsf{T}} = \frac{1}{18} \begin{bmatrix} -18 & 2 & 2 & 1 & 4 & 4 & 4 \end{bmatrix}.$$

When $n_{\text{el}} = 6$, $h = 1/6$, the stiffness matrix is

$$K = \frac{1}{180} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 359 & 5048 & 359 & 0 & 0 & 0 & 0 & -2878 & -2878 & 0 & 0 & 0 & 0 \\ 0 & 359 & 5048 & 359 & 0 & 0 & 0 & 0 & -2878 & -2878 & 0 & 0 & 0 \\ 0 & 0 & 359 & 5048 & 359 & 0 & 0 & 0 & 0 & -2878 & -2878 & 0 & 0 \\ 0 & 0 & 0 & 359 & 5048 & 359 & 0 & 0 & 0 & 0 & -2878 & -2878 & 0 \\ 0 & 0 & 0 & 0 & 359 & 5048 & 359 & 0 & 0 & 0 & 0 & -2878 & -2878 \\ 0 & 0 & 0 & 0 & 0 & 359 & 2524 & 0 & 0 & 0 & 0 & 0 & -2878 \\ -2878 & -2878 & 0 & 0 & 0 & 0 & 0 & 5776 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2878 & -2878 & 0 & 0 & 0 & 0 & 0 & 5776 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2878 & -2878 & 0 & 0 & 0 & 0 & 0 & 5776 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2878 & -2878 & 0 & 0 & 0 & 0 & 0 & 5776 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2878 & -2878 & 0 & 0 & 0 & 0 & 0 & 5776 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2878 & -2878 & 0 & 0 & 0 & 0 & 0 & 5776 \end{bmatrix},$$

$$K = \frac{1}{270}\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 809 & 11348 & 809 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 809 & 5674 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 \\
-6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6478 & -6478 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12976
\end{bmatrix},$$

Figure 1.20 Stiffness matrix for $n_{\text{el}} = 9$ in Example 1.62.

and the (transpose of the) force vector is

$$F^{\mathsf{T}} = \frac{1}{36}\begin{bmatrix} -36 & 2 & 2 & 2 & 2 & 2 & 1 & 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}.$$

Finally, when $n_{\text{el}} = 9$, $h = 1/9$, the stiffness matrix is shown in Fig. 1.20 and the (transpose of the) force vector is

$$F^{\mathsf{T}} = \frac{1}{54}\begin{bmatrix} -54 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}.$$

The reader may want to follow this example closely to understand how the assembly works.

Derivation of Assembly Formulas (1.109)**.**

In addition to the way in which global basis functions are constructed, in the following, we will use the facts that: (a) shape functions are zero outside the elements over which they are defined, and (b) the integrals over each element in the definition of ℓ^e and a^e will therefore be zero when computed for a shape function of another element. More precisely, for $a \in \{1,\dots,m\}$ and $e, e', e'' \in \{1,\dots,n_{\text{el}}\}$,

$$N_a^e(x) = 0 \text{ if } x \notin \Omega_e, \text{ from where, } \begin{cases} \ell^e\left(N_a^{e'}\right) = 0 & \text{if } e \neq e', \\ a^e(N_a^{e''}, N_b^{e'}) = 0 & \text{if } e' \neq e \text{ or } e'' \neq e. \end{cases} \tag{1.116}$$

We examine how we arrive to (1.109a) first. For simplicity, we proceed by assuming

that $h_0 = 0$, since it reduces the bookkeeping needed in the derivation. For $A \in \eta_a$,

$$
\begin{aligned}
F_A &= \ell(N_A) && \text{from (1.70}c\text{)}\\
&= \sum_{e=1}^{n_{\rm el}} \ell^e(N_A) + h_L N_A(L) && \text{from (1.106)}\\
&= \sum_{e=1}^{n_{\rm el}} \ell^e\left(\mathop{\sum^{\circ}}_{\{(a,e')|\mathsf{LG}(a,e')=A\}} N_a^{e'}\right) + h_L \mathop{\sum^{\circ}}_{\{(a,e)|\mathsf{LG}(a,e)=A\}} N_a^e(L) && \text{from (1.108)}\\
&= \sum_{e=1}^{n_{\rm el}} \sum_{\{(a,e')|\mathsf{LG}(a,e')=A\}} \ell^e\left(N_a^{e'}\right) + \mathop{\sum^{\circ}}_{\{(a,e)|\mathsf{LG}(a,e)=A\}} h_L N_a^e(L) && \text{linearity}\\
&= \sum_{e=1}^{n_{\rm el}} \sum_{\{a|\mathsf{LG}(a,e)=A\}} \ell^e(N_a^e) + \sum_{\{a|\mathsf{LG}(a,n_{\rm el})=A\}} h_L N_a^{n_{\rm el}}(L), && \text{from (1.116)}\\
&= \sum_{e=1}^{n_{\rm el}-1} \sum_{\{a|\mathsf{LG}(a,e)=A\}} \ell^e(N_a^e) + \sum_{\{a|\mathsf{LG}(a,n_{\rm el})=A\}} \left[\ell^{n_{\rm el}}(N_a^{n_{\rm el}}) + h_L N_a^{n_{\rm el}}(L)\right]\\
&= \sum_{e=1}^{n_{\rm el}} \sum_{\{a|\mathsf{LG}(a,e)=A\}} F_a^e. && \text{from (1.107).}
\end{aligned}
$$

It should be evident from this derivation that the equality in the last line does not change if $h_0 \neq 0$. This proves (1.109a).

Similarly, to obtain (1.109b), for $A \in \eta_a, B \in \eta$,

$$
\begin{aligned}
K_{AB} &= a(N_B, N_A) && \text{from (1.70}c\text{)}\\
&= \sum_{e=1}^{n_{\rm el}} a^e(N_B, N_A) && \text{from (1.101)}\\
&= \sum_{e=1}^{n_{\rm el}} a^e\left(\mathop{\sum^{\circ}}_{\{(b,e'')|\mathsf{LG}(b,e'')=B\}} N_b^{e''}, \mathop{\sum^{\circ}}_{\{(a,e')|\mathsf{LG}(a,e')=A\}} N_a^{e'}\right) && \text{from (1.108)}\\
&= \sum_{e=1}^{n_{\rm el}} \sum_{\{(a,e')|\mathsf{LG}(a,e')=A\}} \sum_{\{(b,e'')|\mathsf{LG}(b,e'')=B\}} a^e(N_b^{e''}, N_a^{e'}) && \text{bilinearity}\\
&= \sum_{e=1}^{n_{\rm el}} \sum_{\substack{\{a|\mathsf{LG}(a,e)=A\}\\ \{b|\mathsf{LG}(b,e)=B\}}} a^e(N_b^e, N_a^e) && \text{from (1.116)}\\
&= \sum_{e=1}^{n_{\rm el}} \sum_{\substack{\{a|\mathsf{LG}(a,e)=A\}\\ \{b|\mathsf{LG}(b,e)=B\}}} K_{ab}^e && \text{from (1.103}a\text{).}
\end{aligned}
$$

Symmetrization of the Stiffness Matrix. When the bilinear from is symmetric, it is possible to manipulate the stiffness matrix to obtain a symmetric matrix. Symmetric matrices are needed when some iterative solvers for the linear system are adopted, such as Conjugate Gradients. Additionally, symmetric matrices can be stored with less memory, or more efficiently. Problems in structural mechanics, such as a elasticity, have a symmetric bilinear form and can benefit from a symmetric stiffness matrix.

Given a stiffness matrix K such that (a) $K_{AB} = K_{BA}$ for $A \in \eta_a, B \in \eta_a$ and, (b) $K_{AB} = \delta_{AB}$ if $A \notin \eta_a$, we can construct a symmetric matrix K^S and load vector F^S such that U is the solution of both

$$KU = F \quad \text{and} \quad K^S U = F^S. \tag{1.117}$$

It is then possible to solve $K^S U = F^S$ to find U, instead of $KU = F$.

Any stiffness matrix that emerges from Galerkin Method, satisfies condition (a) if the bilinear form is symmetric, and satisfies condition (b) automatically, c.f. (1.70c).

The symmetric stiffness matrix and associated load vector follow as

$$K^S_{AB} = \begin{cases} K_{AB} & \text{if } A \in \eta_a, B \in \eta_a, \\ \delta_{AB} & \text{otherwise.} \end{cases} \tag{1.118a}$$

$$F^S_A = \begin{cases} F_A - \sum_{B \in \eta_g} K_{AB} F_B & \text{if } A \in \eta_a, \\ F_A & \text{otherwise} \end{cases} \tag{1.118b}$$

Example 1.63 Let's symmetrize the matrix in Example 1.61. Notice that the matrix in the earlier example, Example 1.60, is already symmetric. The stiffness matrix and load vector from Example 1.61 are

$$K = \begin{bmatrix} 1 & 0 & 0 \\ -31/16 & 72/16 & -29/16 \\ 0 & -29/16 & 39/16 \end{bmatrix}, \qquad F = \begin{bmatrix} 4 \\ 5 \\ 5/2 \end{bmatrix},$$

with $\eta_a = \{2,3\}$.

Before we proceed with the direct construction of the matrix, we look at why it works. The linear system defined by the stiffness matrix and load vector is

$$\begin{array}{llll} 1 \cdot u_1 & +0 \cdot u_2 & +0 \cdot u_3 & = 4, \\ -31/16 \cdot u_1 & +72/16 \cdot u_2 & -29/16 \cdot u_3 & = 5, \\ 0 \cdot u_1 & -29/16 \cdot u_2 & +39/16 \cdot u_3 & = 5/2. \end{array}$$

Because the first line is a constrained index, it defines directly the value of $u_1 = 4$. We can then replace this in the first column of lines 2 and 3, and move them to the right hand side. The linear system can then be written as

$$\begin{array}{llll} 1 \cdot u_1 & +0 \cdot u_2 & +0 \cdot u_3 & = 4, \\ 0 \cdot u_1 & +72/16 \cdot u_2 & -29/16 \cdot u_3 & = 5 + 31/16 \cdot 4, \\ 0 \cdot u_1 & -29/16 \cdot u_2 & +39/16 \cdot u_3 & = 5/2. \end{array}$$

The matrix associated to this linear system is then symmetric. This is what (1.118) is doing.

Notice that the conditions for symmetrization are satisfied. Condition (a) is satisfied because the submatrix formed by the entries (2,2), (2,3), (3,2), and (3,3) is symmetric. Condition (b) is satisfied because the first row is identically zero, except for the diagonal, in which it is equal to 1. This is of course expected, since this matrix emerged from Galerkin Method and the symmetric bilinear form in Example 1.58.

The symmetrized matrix is

$$K^S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 72/16 & -29/16 \\ 0 & -29/16 & 39/16 \end{bmatrix},$$

The associated load vector F^S follows as

$$F^S = \begin{bmatrix} 4 \\ 5 \\ 5/2 \end{bmatrix} - 4 \begin{bmatrix} 0 \\ -31/16 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 51/4 \\ 5/2 \end{bmatrix}.$$

To conclude, you can check that both systems lead to the same solution

$$U = \begin{bmatrix} 4 \\ 9116/1967 \\ 8796/1967 \end{bmatrix}.$$

Derivation of Symmetrization Formulas (1.118)

To see that $KU = F$ if and only if $K^S U = F^S$, we consider first the equations for $A \notin \eta_a$. In this case,

$$F_A = K_{AB} u_B = \delta_{AB} u_B = K^S_{AB} u_B = F^S_A. \tag{1.119}$$

Next, we consider the equations for $A \in \eta_a$. We will need to use that from 1.119,

$$F_B = u_B \text{ for } B \notin \eta_a, \tag{1.120}$$

and from (1.118a),

$$K^S_{AB} = 0 \text{ if } A \in \eta_a, B \notin \eta_a. \tag{1.121}$$

Then,

$$\begin{aligned}
0 &= \sum_{B \in \eta} K_{AB} u_B - F_A \\
&= \sum_{B \in \eta_a} K_{AB} u_B + \sum_{B \notin \eta_a} K_{AB} F_B - F_A && \text{from (1.118}a\text{)} \\
&= \sum_{B \in \eta_a} K^S_{AB} u_B - \left(F_A - \sum_{B \in \eta_g} K_{AB} F_B \right) && \text{from (1.120)} \\
&= \sum_{B \in \eta_a} K^S_{AB} u_B - F^S_A && \text{from (1.118}b\text{)} \\
&= \sum_{B \in \eta_a} K^S_{AB} u_B + \sum_{B \notin \eta_a} K^S_{AB} u_B - F^S_A && \text{from (1.121)} \\
&= \sum_{B \in \eta} K^S_{AB} u_B - F^S_A.
\end{aligned}$$

Together with (1.119), this proves that if U satisfies one set of equations, it satisfies the others.

Finite Element Bases and Sparse Stiffness Matrices. A glance at the stiffness matrix computed with P_1-elements in (1.77) reveals that the matrix has non-zero entries only along three of its diagonals, so it is called a **tri-diagonal** matrix. Similarly, an examination of the stiffness matrices in Example 1.62 shows that the only non-zero entries lie along 7 different diagonal directions. This means that as the number of elements grows, the majority of the entries in the matrix are equal to zero. In fact, the number of non-zero entries grows linearly with the number of rows $n_{\text{el}}+1$, since it is proportional to the length of the diagonals. In contrast, the number of zeros grows quadratically with the number of rows, since it corresponds to the rest of the $(n_{\text{el}}+1)^2$ entries. Matrices in which the fraction of non-zero entries is small are called **sparse**.

The sparsity of the matrix has consequences in the computational efficiency of a method. If the matrix is going to be stored in memory[12], it is convenient to store the non-zero entries only. This drastically reduces the memory requirements as the number of elements grows, since the amount of memory needed is $\mathcal{O}(n_{\text{el}})$ instead of $\mathcal{O}({n_{\text{el}}}^2)$ as $n_{\text{el}} \to \infty$. The same scaling law applies to matrix-vector products, which is the main operation used in the solution of linear systems of equations through iterative methods .

For the forthcoming discussion, it is convenient to introduce the definition of support of a function. Given a real-valued function f with domain Ω, the set

$$\text{supp}(f) = \overline{\{x \in \Omega \mid f(x) \neq 0\}} \tag{1.122}$$

is called the **support** of f. In (1.122), the line over set indicates its closure: it says that the support of f is formed by all the points in the set, but also by those that may not be in the set but that can be reached as limits of sequences of points in the set. For the forthcoming discussion, it is enough to think about the support of f as essentially the set of all points in Ω at which f is equal to zero.

Back to the main discussion then, when are stiffness matrices in the Finite Element Method sparse? For active indices, entries in the stiffness matrix of Galerkin Method are computed as $K_{AB} = a(N_B, N_A)$. The most commonly found scenario is represented by the bilinear form in Example 1.43

$$a(u,v) = \int_0^1 u_{,x}(x)\, v_{,x}(x)\, dx. \tag{1.123}$$

In this case, $a(N_B, N_A) = 0$ if $N_{B,x}$ and $N_{A,x}$ are different than zero over non-intersecting regions of the domain, or more precisely, when the intersection of their supports has zero length. For a finite element space of continuous functions over P_1-elements, $N_{B,x}N_{A,x}$ is a non-zero function only when $|A-B| \leq 1$, that is,

[12] The so-called matrix-free methods never build the matrix, see e.g. CITE

when A and B are indices of the same node or of neighboring nodes. This is the reason for the appearance of the tri-diagonal matrix in (1.77), since for each row A, only the entries $K_{A(A-1)}$, K_{AA} and $K_{A(A+1)}$ are non-zero. There are always only at most three non-zero entries per row regardless of the number of elements n_{el}, so the matrix becomes increasingly sparse as n_{el} grows.

A similar scenario is found with the bilinear form of Example 1.62,

$$a(u,v) = \int_0^1 u_{,x} v_{,x} + uv \, dx. \tag{1.124}$$

In this case, if A is the index of a node between elements in the mesh, row A has at most 5 non-zero entries: $K_{AB} \neq 0$ only if $|A-B| \leq 1$ (indices of the neighboring nodes or the same node), $B = A + n_{\text{el}}$, or $B = A + n_{\text{el}} + 1$ (indices of the neighboring nodes in the middle of an element); see Fig. 1.20. Alternatively, if A is the index of a node in the middle of an element, then $K_{AB} \neq 0$ only if $B = A - n_{\text{el}}$ (the index of the node on its right) or $B = A - n_{\text{el}} - 1$ (the index node of the node on its left). As in the previous example, the number of non-zero entries per row is the same for all values of n_{el}, so the matrix becomes increasingly sparse as n_{el} grows.

These two examples should be contrasted with one of Galerkin Method with non-finite element bases. For example, it is possible to select $\mathcal{W}_h = \mathbb{P}_{m-1}(\Omega)$ for $m \geq 1$, with basis functions $N_A(x) = x^{A-1}$ for $A = 1, \ldots, m$. In this case, $\text{supp}(N_A) = \Omega$ for all A, so the support of each basis function is the entire domain. For any of the two bilinear forms (1.123) or (1.124), choosing this basis leads to non-sparse matrices, as we can expect from the non-empty intersection of the support of any pair of basis functions. Specifically, assuming all indices are active, we have

$$K_{AB} = \int_0^1 (A-1)(B-1)x^{A+B-4} \, dx = \frac{(A-1)(B-1)}{A+B-3}, \qquad \text{for (1.123)}$$

$$K_{AB} = \int_0^1 (A-1)(B-1)x^{A+B-4} + x^{A+B-2} \, dx = \frac{(A-1)(B-1)}{A+B-3} + \frac{1}{A+B-1}. \qquad \text{for (1.124)}$$

These are all non-zero entries (except for the $A = 1$ row or $B = 1$ column for (1.123)). Matrices in which most of the entries are non-zero are called **dense**.

Sufficient Conditions for Stiffness Matrices in the Finite Element Method to be Sparse

The following are sufficient conditions on a finite element basis and the bilinear form to generate a sparse stiffness matrix.

If

(a) $\text{supp}(f) \cap \text{supp}(g) = \emptyset \implies a(f,g) = 0$, and

(b) There exists $n_{\text{width}} \in \mathbb{N}$ such that for all $A = 1, \ldots, m$ the number of elements of the set $s_A = \{e \mid \mathsf{LG}(a,e) = A\}$ is less or equal than n_{width}, i.e., $\#s_A \leq n_{\text{width}}$,

then the number of non-zero entries in each row of K is less or equal than $n_{\text{width}} \times k$.

To see this, notice first that $\text{supp}(N_A) = \cup_{e \in s_A} \Omega_e$, since these are all of the elements in which a shape function is added to form N_A. Therefore, due to the first condition, $K_{AB} = 0$ if $s_A \cap s_B = \emptyset$. For each element $e \in s_A$, there are at most k other basis functions whose support includes Ω_e, given that each shape function is added to one and only one global basis function, c.f., 1.3.3. Thus, due to the second condition, the set $\{B \in \{1, \ldots, m\} \mid s_A \cap s_B \neq \emptyset\}$ has at most $n_{\text{width}} \times k$ elements, and hence each row of K has at most that number of non-zero elements.

Let's briefly discuss the conditions and the implications of this result. The first condition is satisfied by essentially all commonly found bilinear forms. The second condition examines the local-to-global map to request global basis functions to be formed by adding at most n_{width} shape functions, regardless of the number of elements in the mesh. Because the number of non-zero entries is less or equal than $n_{\text{width}} \times k$, a quantity that does not change as n_{el} grows, the associated stiffness matrix grows increasingly sparse as $n_{\text{el}} \to \infty$.

We conclude this section with a couple of final remarks:

- The use of methods that generate non-sparse matrices can be convenient when they lead to numerical solutions of similar accuracy with fewer global degrees of freedom than a Finite Element Method, such as in some Boundary Element Methods or Spectral Methods.
- The property of the finite element basis functions to be non-zero only in a few elements is typically referred to in the literature by stating that *finite element basis functions have compact support.* It is worth reflecting a bit on the meaning of this statement. Mathematically, a function f is said to have **compact support** if $\text{supp}(f)$ is a bounded set. Hence, any function whose support is included in the domain $\Omega = (0, L)$ of our example has compact support. It is clear then that a literal interpretation of this statement does not capture that the support of a finite element basis function is included in the union of at most a fixed number of elements, regardless of the number of elements in the mesh.

1.4 Elliptic Fourth-Order Problems

Galerkin method, as we have seen, can be summarized as the **restriction** of a suitable **weak formulation** of the problem to subspaces of $\mathscr{S}$ and $\mathscr{V}$. The **finite element method**, on the other hand, is the combination of the Galerkin method with some specially designed **finite element spaces**, which are very efficient (because the involved matrices are sparse), easy to code (because the equation system is built element by element) and mathematically rigorous.

The goal of this section is to reinforce this description by showing how it applies to fourth-order problems, but it will also introduce one more consideration we have so far stayed silent about: the need for additional smoothness of the finite element functions. For this purpose, we will reenact the same sequence of the previous sections. We will first formulate the strong and weak forms of the

problem and understand the path from one to the other. Then we will deduce and implement the Galerkin method and build an adequate finite element space.

1.4.1 The Strong Form

The elliptic fourth-order problems that most frequently appear in applications are of the general form

$$\big(q(x)\,u''(x)\big)'' - \big(k(x)\,u'(x)\big)' + c(x)\,u(x) = f(x)\,. \tag{1.125}$$

The field under study is, as before, u, and q, k, c and f are the coefficients. This class of equations is broad enough to model very interesting problems.

Examples:

1.64 *Euler-Bernoulli beam equation:* Consider a rectilinear beam with Young modulus $E(x)$ whose cross-section has moment of inertia relative to a neutral horizontal axis $I(x)$. Let $u(x)$ model the (small) vertical deflection of the beam when it is subjected to a vertical load (per unit length) $f(x)$. Then the vertical displacement must satisfy the equation

$$\big(E(x)\,I(x)\,u''(x)\big)'' = f(x) \tag{1.126}$$

at all points x for the beam to be in static equilibrium. In this case $k(x) = c(x) = 0$ and $q(x) = E(x)\,I(x)$ is the *bending rigidity* of the beam.

1.65 *Diffuse-interfaces in material science:* When a material separates into two distinct phases, the process is often modeled by a *diffuse interphase* equation formulated in terms of a *concentration variable* u. A classical example is the *Cahn-Hilliard equation.* Let us consider here a steady-state, linearized form of this equation, which reads

$$q\,u'''' - k\,u'' = f\,, \tag{1.127}$$

where q represents a *diffusion coefficient* and the other coefficients arise from the linearization. This is certainly a particular case of (1.125).

1.66 *Image denoising:* In this application an input image u_0 (a function of x) is to be transformed so as to remove its noise. To do this, the "denoised image" $u(x)$ is defined as the solution to

$$\big(q(x)\,u''\big)'' + u = u_0 \tag{1.128}$$

for a carefully chosen coefficient $q(x)$ (in fact, $q(x)$ is often a function of u itself, but this would turn the equation *nonlinear*, which is outside the scope of the chapter).

For simplicity, in the following we restrict our attention to the case $k = 0$.

It is known from the theory of ordinary differential equations that, if $q(x) \neq 0$, four boundary conditions are required to completely specify u. Of the many possibilities, the class of problems we consider (*elliptic* problems) impose *two conditions on each boundary point of the domain* $\Omega = (0, L)$. The most popular boundary conditions are:

- **Clamped conditions**, which specify the values of $u(0)$ *and* of $u'(0)$ (and/or of $u(L)$ *and* of $u'(L)$, depending on which boundary is considered).
- **Applied load conditions**, which specify the values of $u''(0)$ and of $u'''(0)$ (and/or $u''(L)$ and $u'''(L)$).

To study both types of conditions simultaneously, let us consider a problem with clamped conditions at $x = 0$ and applied load conditions at $x = L$. Other combinations are easy to understand by analogy. The strong form of the problem is:

Problem 1.6. (Strong form of the fourth-order problem) *Given the coefficients q, c and f (as functions of x), together with the boundary constants g_0, d_0, m_L and n_L, find a continuously differentiable function $u : \Omega \to \mathbb{R}$ satisfying*

$$\big(q(x)u''(x)\big)'' + c(x)u(x) = f(x) \qquad \forall\, x \in \Omega \tag{1.129a}$$

$$u(0) = g_0 \tag{1.129b}$$

$$u'(0) = d_0 \tag{1.129c}$$

$$u''(L) = m_L \tag{1.129d}$$

$$u'''(L) = n_L \tag{1.129e}$$

Examples:

1.67 *The simplest beam problem* is a homogeneous cantilever beam (q independent of x, $c = 0$) with no distributed load $f = 0$, clamped horizontally at $x = 0$ (i.e., $u(0) = u'(0) = 0$) and with a vertical force W and a bending moment T applied at $x = L$. The strong problem simplifies to

$$u''''(x) = 0,\ \forall x \in \Omega, \quad u(0) = u'(0) = 0, \quad u''(L) = \frac{T}{q}, \quad u'''(L) = -\frac{W}{q}.$$

Since $u'''' = 0$, the solution is necessarily a cubic polynomial, and because of the clamped conditions at $x = 0$ it must be of the form

$$u(x) = c_1\, x^2 + c_2\, x^3\ .$$

It only remains to calculate c_1 and c_2 so that the boundary conditions at $x = L$ are satisfied. The exact solution is

$$u(x) = \frac{T + WL}{2q}\, x^2 - \frac{W}{6q}\, x^3\ .$$

The tip displacement is, in particular,

$$u(L) = \frac{T}{2q}L^2 + \frac{W}{3q}L^3 .$$

1.68 Other interesting boundary conditions are **elastic support** conditions, which are an analog to the Robin conditions discussed in Section 1.1.1. Their expression is (considering the boundary at $x = 0$)

$$u''(0) - \alpha_0\, u'(0) = \beta_0 , \qquad u'''(0) + \gamma_0\, u(0) = \delta_0 , \tag{1.130}$$

where α_0, β_0, γ_0 and δ_0 are given constants. Similarly to the second-order case, making α_0 very large *de facto* imposes the value of $u'(0)$, and making γ_0 very large imposes $u(0)$. In the limit $\alpha_0 \to +\infty$, $\gamma_0 \to +\infty$ we end up with *clamped* conditions. On the other hand, when $\alpha_0 = \gamma_0 = 0$ we recover the *applied load* conditions.

It is always important to check that the problem we are considering is well posed, in the sense that one can expect to have a unique solution which depends continuously on the coefficients and boundary conditions. In this case we have

Theorem 1.4. *Under the hypotheses that the coefficients q and c are piecewise smooth and non-negative, with $q(x) \geq q_{\min} > 0$, $\forall x$, and that $\int_0^L |f(x)|\,dx < \infty$, Problem 1.6 is well posed.*

1.4.2 The Weak Form

Let us follow the procedure outlined in 1.1.4 to determine a suitable weak formulation of Problem 1.6. The residual is

$$r(x) = \big(q(x)\,u''(x)\big)'' + c(x)\,u(x) - f(x). \tag{1.131}$$

If u is the solution, then $r(x) = 0$ for all x, and thus for any smooth function $v(x)$ it must hold that

$$0 = \int_0^L r(x)\,v(x)\,dx = \int_0^L \Big[\big(q(x)\,u''(x)\big)'' + c(x)\,u(x) - f(x)\Big]\,v(x)\,dx. \tag{1.132}$$

After distributing the product inside the bracket, we integrate by parts twice the term $\int_0^L (qu'')''\,v\,dx$ so that the differentiation order is balanced between u and v, arriving at

$$\begin{aligned}\int_0^L \big[q(x)\,u''(x)\,v''(x) + c(x)\,u(x)\,v(x)\big]\,dx = \int_0^L f(x)\,v(x)\,dx \\ - (qu''' + q'u'')(L)\,v(L) + (qu''' + q'u'')\,v(0) \\ + q(L)\,u''(L)\,v'(L) - q(0)\,u''(0)\,v'(0).\end{aligned} \tag{1.133}$$

The second and third lines of (1.133) contain the boundary values of u and v and their derivatives. We next replace with the boundary conditions we have information about and that appear in the boundary terms. In this case, these are the values of $u''(L) = m_L$ and $u'''(L) = n_L$, to get

$$\int_0^L \left[q(x)u''(x)v''(x) + c(x)u(x)v(x)\right] dx = \int_0^L f(x)v(x)\,dx - (q(L)n_L + q'(L)m_L)v(L) + (qu''' + q'u'')v(0) + q(L)m_L v'(L) - q(0)u''(0)v'(0). \quad (1.134)$$

Since we do not know the values of $u'''(0)$ and $u''(0)$, we will request $v(0) = 0$ and $v'(0) = 0$ in the definition of the test space. If we do not request these, complexities arise in the formulation of the finite element method that are better left to be discussed at a later stage of learning. Traditional methods are based on constraining the test space as described.

The **natural boundary conditions** for this problem are then $u''(L) = m_L$ and $u'''(L) = n_L$, while $u(0) = g_0$ and $u'(0) = d_0$ need to be considered **essential boundary conditions** and requested as constrains in the definition of the trial space.

The trial space is

$$\mathcal{S} = \left\{v\colon [0,L] \to \mathbb{R} \text{ smooth } \mid v(0) = g_0 \text{ and } v'(0) = d_0\right\}, \quad (1.135)$$

while the test space is

$$\mathcal{V} = \left\{v\colon [0,L] \to \mathbb{R} \text{ smooth} \mid v(0) = 0 \text{ and } v'(0) = 0\right\}. \quad (1.136)$$

The test space is indeed the direction of the trial space: Let $v, w \in \mathcal{S}$ and $z = v - w$. Then,

$$z(0) = v(0) - w(0) = g_0 - g_0 = 0, \quad \text{and} \quad z'(0) = v'(0) - w'(0) = d_0 - d_0 = 0\,,$$

and thus $z \in \mathcal{V}$. This choice of $\mathcal{S}$, and thus of $\mathcal{V}$, zeroes out the terms containing $v(0)$ and $v'(0)$ in (1.134). The containing vector space $\mathcal{W}$ needs functions to just be smooth enough for the bilinear form and linear functional to be well defined, namely

$$\mathcal{W} = \{v\colon [0,L] \to \mathbb{R} \text{ smooth}\}.$$

Now we know all terms in the right-hand side of (1.133), and the weak formulation reads:

Problem 1.7. (Weak form of Problem 1.6) *Find $u \in \mathcal{S}$ such that*

$$a(u,v) = \ell(v) \qquad \forall\, v \in \mathcal{V}, \quad (1.137a)$$

where

$$a(u,v) = \int_0^L \left[q(x)u''(x)v''(x) + c(x)u(x)v(x)\right] dx, \quad (1.137b)$$

$$\ell(v) = \int_0^L f(x)v(x)\,dx - \left(q(L)n_L + q'(L)m_L\right)v(L) + q(L)m_L v'(L). \quad (1.137c)$$

Notice that **the bilinear form $a(\cdot,\cdot)$ is symmetric**.

In more mechanical terms, it so happens that $-\big(q(L)n_L + q'(L)m_L\big)$ equals the applied force load W at L, while $q(L)m_L$ equals the applied torque T at L. So, an equivalent form is

$$\ell(v) = \int_0^L f(x)v(x)\,dx + W\,v(L) + T\,v'(L)\,. \tag{1.138}$$

A smoother space $\mathscr{W}$. The bilinear form in this problem involves the second derivative of u and v. A necessary condition for the weak form to have a unique solution that depends smoothly on the data of the problem (functions q, c, f and boundary values) is for the second first derivative of u and v to be at least continuous. Therefore, we will require functions in the space $\mathscr{W}$, and hence in $\mathscr{S}, \mathscr{V} \subset \mathscr{W}$, to contain functions whose first derivative is continuous. We will return to this point when we discuss about the numerical analysis of the FEM.

1.4.3 Galerkin Method

The Galerkin Method applied to Problem 1.7 is exactly the same as in Problem 1.5: *Let $\mathscr{W}_h \subset W$, $\mathscr{S}_h = \mathscr{S} \cap \mathscr{W}_h$ and $\mathscr{V}_h = \mathscr{V} \cap \mathscr{W}_h$. Find $u_h \in \mathscr{S}_h \subset \mathscr{S}$ such that*

$$a(u_h, v_h) = \ell(v_h) \tag{1.139}$$

for all $v_h \in \mathscr{V}_h$.

The bilinear and linear forms are different, because they depend on the problem, but the formulation is the same. Let us work out an example to show how (1.139) results in a solvable linear system of equations that allows us to compute u_h.

Galerkin method with global polynomials. To proceed, we need a finite-dimensional space $\mathscr{W}_h$ that is **contained in $\mathscr{W}$**. Remembering that $\mathscr{W}$ is the space of smooth functions in $[0, L]$, let us first apply the Galerkin Method with the space $\mathscr{W}_h = \mathbb{P}_k([0, L])$ (polynomials of degree $\le k$). Polynomials are certainly smooth functions, in fact infinitely so.

The next step is to find a basis of $\mathscr{W}_h$ of which a subset is a basis of $\mathscr{V}_h$. Let

$$N_1(x) = 1, \quad N_2(x) = x, \quad \ldots \quad N_{k+1}(x) = x^k\,,$$

then

$$\{N_1, N_2, \ldots, N_{k+1}\} \qquad \text{is a basis of } \mathscr{W}_h$$

and

$$\{N_3, N_4, \ldots, N_{k+1}\} \qquad \text{is a basis of } \mathscr{V}_h.$$

In fact, N_1 and N_2 are the only two basis functions of $\mathscr{W}_h$ that do not satisfy $N_a(0) = N_a'(0) = 0$.

By direct inspection, we see that

$$m = \dim \mathscr{W}_h = k+1 \qquad \text{and} \qquad n = \dim \mathscr{V}_h = m-2 = k-1\,.$$

Following the same reasoning as in 1.2.3, we write

$$u_h(x) = u_1 N_1(x) + u_2 N_2(x) + \ldots + u_m N_m(x) \tag{1.140}$$

so that u_b $(b = 1, \ldots, m)$ are the coefficients that define the numerical solutions and the algebraic unknowns of our problem. To compute them one has to solve the linear system that results from the essential boundary conditions ($u_1 = g_0$, $u_2 = d_0$) and from successively taking $v_h = N_3$, $v_h = N_4$, etc., in (1.139). The linear system thus reads

$$\begin{aligned}
u_1 &= g_0 \\
u_2 &= d_0 \\
\sum_{b=1}^{m} u_a\, a(N_b, N_3) &= \ell(N_3) \\
\sum_{b=1}^{m} u_a\, a(N_b, N_4) &= \ell(N_4) \\
\cdots &\quad \cdots \\
\sum_{b=1}^{m} u_a\, a(N_b, N_m) &= \ell(N_m)
\end{aligned}$$

Written in matrix form, we have

$$K\,U = F\,,$$

with

$$K_{ab} = \begin{cases} \delta_{ab} & \text{if } a = 1 \text{ or } a = 2 \\ a(N_b, N_a) & \text{if } a > 2 \end{cases},$$

and

$$F_a = \begin{cases} g_0 & \text{if } a = 1 \\ d_0 & \text{if } a = 2 \\ \ell(N_a) & \text{if } a > 2 \end{cases},$$

Example 1.69 To compute actual numbers, let us consider the "simplest beam problem" introduced in Example 1.67, so that $q(x) = q$ is constant, $c = f = 0$, $m_L = T/q$ and $n_L = -W/q$. Let us choose $k = 4$, so that $W_h = \mathbb{P}_4([0, L])$ and thus $m = 5$.

The elements of K can be calculated by straightforward integration, **evaluating to zero if $a < 2$ or $b < 2$ and**

$$\begin{aligned}
a(N_b, N_a) &= \int_0^L q\, N_b'' N_a''\, dx \\
&= \int_0^L q(b-1)(b-2)x^{b-3}(a-1)(a-2)x^{a-3}\, dx \\
&= \frac{q(b-1)(b-2)(a-1)(a-2)L^{a+b-5}}{a+b-5}
\end{aligned}$$

otherwise. Thus,

$$K = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 4qL & 6qL^2 & 8qL^3 \\ 0 & 0 & 6qL^2 & 12qL^3 & 18qL^4 \\ 0 & 0 & 8qL^3 & 18qL^4 & (144/5)qL^5 \end{pmatrix}.$$

The value of the determinant of K is $\det K = 9.6\, q^3 L^9$.

The entries of the load vector F for $a > 2$ are

$$F_a = W N_a(L) + T N_a'(L) = W L^{a-1} + T(a-1) L^{a-2} \qquad (a > 2),$$

giving

$$F = \begin{pmatrix} 0 \\ 0 \\ WL^2 + 2TL \\ WL^3 + 3TL^2 \\ WL^4 + 4TL^3 \end{pmatrix}.$$

It only remains to solve $KU = F$. The solution reads

$$U = \begin{pmatrix} 0 \\ 0 \\ (T + WL)/2q \\ -W/(6q) \\ 0 \end{pmatrix}$$

(as can be easily checked by substitution) implying that the Galerkin solution is

$$u_h = 0 \cdot 1 + 0 \cdot x + \frac{T + WL}{2q} \cdot x^2 - \frac{W}{6q} \cdot x^3 + 0 \cdot x^4 \,,$$

coincident with the exact solution computed in Example 1.67. This is always true for the Galerkin method: **if the exact solution lies in $\mathscr{S}_h$, then the Galerkin solution coincides with the exact solution**.

1.4.4 The Simplest C^1 Finite Element Space

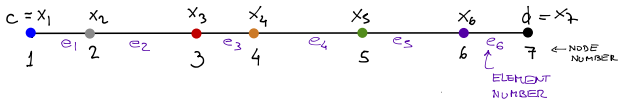
Let us now introduce the simplest **finite element space** that can approximate the fourth-order Problem 1.6, known as **Hermite piecewise cubic space**. We will denote it here as H_3-space. We follow the same steps as in Section 1.3.1, but notice that some important differences arise.

Steps:

1. **Build the mesh of the domain.** Let the domain of the problem be the interval $[0, L]$. We partition the domain into $n_{\text{el}} \in \mathbb{N}$ intervals by selecting vertices $\{x_i\}_{i=1,\ldots,n_{\text{el}}+1}$ such that

$$0 = x_1 < \ldots < x_{n_{\text{el}}+1} = L. \tag{1.141}$$

Interval $[x_i, x_{i+1}]$ is the element domain for element i, for $i = 1, \ldots, n_{\text{el}}$.



2. **Build basis functions.** Remarkably, the "hat functions" $\{N_a\}$ introduced earlier are not useful to approximate fourth-order problems with Galerkin Method. For such problems, $\mathcal{W}$ needs to contain functions with continuous first derivatives. Since for Galerkin Method we require $\mathcal{W}_h \subset \mathcal{W}$, functions in $\mathcal{W}_h$ need to have a continuous first derivative as well. The "hat functions" have discontinuous first derivatives, and hence cannot form the space $\mathcal{W}_h$.

   More generally, finite element spaces built with Lagrange P_k-elements will contain functions whose first derivative is discontinuous across element boundaries, and hence cannot be used to built $\mathcal{W}_h$. We thus proceed to introduce a *new* set of basis functions $\{H_k(x), k = 1, 2, \ldots, 2n_{\text{el}} + 2\}$, which are known as **Hermite basis functions**. Their most important feature is that their first derivative is continuous in $[0, L]$, which is why we say that they generate a C^1 **finite element space**. The number of Hermite basis functions equals *twice* the number n_{vert} of vertices. Their second derivative is discontinuous along element boundaries but this does not preclude us from computing the necessary integrals, in the same way that we were able to compute $\int_0^L N_a'(x) N_b'(x)\, dx$ for the hat functions although their first derivatives are discontinuous.

   The Hermite basis functions are **piecewise cubic polynomials**, but not any piecewise cubic polynomial since only a subset of them (in fact, a subspace) is contained in $C^1([0, L])$. The dimension of the vector space Z of piecewise cubic polynomials in a mesh of n_{el} elements is $4\, n_{\text{el}}$, because there are 4 linearly independent cubic polynomials *per element*. The subspace that consists only of C^1 functions, $\mathcal{W}_h = Z \cap C^1([0, L])$, incorporates 2 linear restrictions (continuity of function and derivative) at each of the $n_{\text{el}} - 1$ inter-element boundaries and thus has dimension $m = 4n_{\text{el}} - 2(n_{\text{el}} - 1) = 2n_{\text{el}} + 2 = 2n_{\text{vert}}$. We thus need *two basis functions per vertex* to provide a basis for W_h.

   The Hermite basis functions, for $a = 1, \ldots, m = 2n_{\text{vert}}$, are defined as

a) Odd-numbered basis functions:

$$H_{2a-1}(x) = \begin{cases} 0 & \text{if } x < x_{a-1} \\ -2\left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^3 + 3\left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^2 & \text{if } x_{a-1} \le x < x_a \\ 1 & \text{if } x = x_a \\ 2\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^3 - 3\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^2 + 1 & \text{if } x_a < x \le x_{a+1} \\ 0 & \text{if } x_{a+1} < x \end{cases} \tag{1.142}$$

b) Even-numbered basis functions:

$$H_{2a}(x) = \tag{1.143}$$
$$= \begin{cases} 0 & \text{if } x < x_{a-1} \\ \left[\left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^3 - \left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^2\right](x_a - x_{a-1}) & \text{if } x_{a-1} \le x < x_a \\ 0 & \text{if } x = x_a \\ \left[\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^3 - 2\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^2 + \left(\frac{x-x_a}{x_{a+1}-x_a}\right)\right](x_{a+1} - x_a) & \text{if } x_a < x \le x_{a+1} \\ 0 & \text{if } x_{a+1} < x \end{cases}$$

As before, when $a = 1$, $x \in [0, L]$ implies that we only have the case $x \ge x_a = x_1 = 0$ and when $a = n_{\text{vert}}$, $x \in [0, L]$ implies that we only have the case $x \le x_a = x_{n_{\text{vert}}} = L$. These functions are plotted below for a mesh of three elements and four vertices, $x_0 = 0$, $x_1 = 1$, $x_2 = 2$.

By direct inspection of (1.142)-(1.143) it is evident that these functions are piecewise cubic polynomials. It is not difficult to check that they belong to $C^1([0, L])$ for any valid mesh positions ($x_{a+1} - x_a$ must be strictly positive for all $a = 1, \ldots, n_{\text{vert}} - 1$). You can also verify the following properties:

(a) They add up to 1, i.e., $\sum_{a=1}^{2n_{\text{vert}}} H_a(x) = 1$ for $x \in [0, L]$.

(b) They are linearly independent.

(c) They have **compact support**. The support of H_{2a-1} and H_{2a} is the interval $[x_{a-1}, x_{a+1}]$.

(d) The odd-numbered functions satisfy $H_{2a-1}(x_a) = 1$ and $H'_{2a-1}(x_a) = 0$ at their associated vertices, while $H_{2a-1}(x_b) = H'_{2a-1}(x_b) = 0$ for all other vertices $x_b \neq x_a$.

(e) The even-numbered functions satisfy $H_{2a}(x_b) = 0$ for all vertices x_b of the mesh. On the other hand, their derivative is one at the associated vertex and zero at all other vertices, i.e., $H'_{2a}(x_a) = 0$ and $H'_{2a}(x_b) = 0$ for all $x_b \neq x_a$.

The H_3-space is the vector space of all linear combinations of the functions $H_1, H_2, \ldots, H_{2n_{\text{vert}}}$. From the previous properties, we conclude that the H_3-space is exactly the same space as $\mathcal{W}_h = Z \cap C^1([0,L])$ (the piecewise cubic polynomials that are C^1), i.e.,

$$\mathcal{W}_h = \text{span}\left(H_1, H_2, \ldots, H_{2n_{\text{vert}}}\right), \tag{1.144}$$

and that $H_1, H_2, \ldots, H_{2n_{\text{vert}}}$ is a basis of $\mathcal{W}_h$.

Further, from items (d) and (e) above, we know that for w_h arbitrary in $\mathcal{W}_h$,

$$w_h(x) = c_1 H_1(x) + c_2 H_2(x) + \ldots + c_{2n_{\text{vert}}} H_{2n_{\text{vert}}}(x) \,,$$

it holds that

$$\left\{\begin{array}{rcl} c_1 &=& w_h(x_1), \\ c_2 &=& w'_h(x_1), \\ c_3 &=& w_h(x_2), \\ c_4 &=& w'_h(x_2), \\ \ldots & & \ldots \\ c_{2n_{\text{vert}}-1} &=& w_h(x_{n_{\text{vert}}}), \\ c_{2n_{\text{vert}}} &=& w'_h(x_{n_{\text{vert}}}). \end{array}\right. \tag{1.145}$$

The arbitrary coefficients c_i ($i = 1, \ldots, m = 2n_{\text{vert}}$) are the **degrees of freedom of the space**. The *odd* degree of freedom c_{2k-1} is the value of w_h at vertex x_k. The *even* degree of freedom c_{2k} is the value of w'_h at vertex x_k. Because of these, this finite element mesh has a node at each vertex. Henceforth, we will refer to them as nodes.

3. **Build $\mathcal{V}_h$.** The boundary conditions at $x = L$, which are *natural* boundary conditions, have already been incorporated into the weak form (1.137a). On the other hand, the boundary conditions at $x = 0$ are *essential* and thus need to be imposed explicitly in the definition of the trial and test spaces, $\mathcal{S}_h$ and $\mathcal{V}_h$.

   How do we do that? We define both $\mathcal{S}_h$ and $\mathcal{V}_h$ as **suitable subsets** of $\mathcal{W}_h$. Let w_h be an arbitrary function in $\mathcal{W}_h$,

$$w_h(x) = c_1 H_1(x) + c_2 H_2(x) + \ldots + c_{2n_{\text{vert}}} H_{2n_{\text{vert}}}(x) \,. \tag{1.146}$$

   Now, since $x_1 = 0$, we have that

$$w_h(0) = c_1, \qquad \text{and} \qquad w'_h(0) = c_2 \,. \tag{1.147}$$

Because the value of $w_h(0)$ involves only c_1 and that of $w_h'(0)$ involves only c_2 it is straightforward to build the trial and test spaces for our problem. The basis was purposefully designed to make things easy.

Functions v_h belonging to the **test space** $\mathcal{V}_h$, to begin with, need to satisfy $v_h(0) = v_h'(0) = 0$. These two linear restrictions are automatically satisfied if

$$\mathcal{V}_h = \left\{ v_h \in \mathcal{W}_h | v_h(0) = v_h'(0) = 0 \right\} = \text{span}\left(H_3, H_4, \ldots, H_{2n_{\text{vert}}} \right), \tag{1.148}$$

meaning that $\mathcal{V}_h$ consists of all functions of the form

$$v_h(x) = c_3 H_3(x) + c_4 H_4(x) + \ldots + c_{2n_{\text{vert}}} H_{2n_{\text{vert}}}(x) \,, \tag{1.149}$$

or, equivalently, all functions in $\mathcal{W}_h$ that have $c_1 = c_2 = 0$.

Functions z_h in the **trial space** $\mathcal{S}_h$, in turn, need to satisfy $z_h(0) = g_0$ and $z_h'(0) = d_0$. From (1.147) we know that this takes place if and only if $c_1 = g_0$ and $c_2 = d_0$. This means that the functions in $\mathcal{S}_h$ can be written as

$$z_h(x) = g_0 H_1(x) + d_0 H_2(x) + c_3 H_3(x) + \ldots + c_{2n_{\text{vert}}} H_{2n_{\text{vert}}}(x) \,, \tag{1.150}$$

where the coefficients $c_3, \ldots, c_{2n_{\text{vert}}}$ are arbitrary. Another way of writing the definition of $\mathcal{S}_h$ is

$$\begin{aligned} \mathcal{S}_h &= \left\{ z_h \in \mathcal{W}_h | z_h = g_0 H_1 + d_0 H_2 + v_h,\, v_h \in \mathcal{V}_h \right\} \\ &= g_0 H_1 + d_0 H_2 + \mathcal{V}_h \,. \end{aligned} \tag{1.151}$$

If we go back to the discussion that led to (1.64b), we see that (1.151) *invites* us to select $\overline{u}_h = g_0 H_1 + d_0 H_2$ (we could add any linear combination of $H_3, \ldots, H_{2n_{\text{vert}}}$ to it, but … adding nothing is simpler and better!)

The set of indices of all basis functions in $\mathcal{W}_h$ is

$$\eta = \{1, 2, \ldots, 2n_{\text{vert}}\},$$

the set of constrained indices is

$$\eta_g = \{1, 2\},$$

and the set of active indices follows as

$$\eta_a = \{3, 4, \ldots, 2n_{\text{vert}}\}.$$

The number of indices in $\eta \setminus \eta_g$ is n, the dimension of $\mathcal{V}_h$, and is thus the number of linearly independent equations generated by our weak form when $v_h \in \mathcal{V}_h$. In our example, this number is $2n_{\text{vert}} - 2$ (i.e., $m - 2$). Adding the two equations coming from the boundary conditions $u_h(0) = c_1 = g_0$ and $u_h'(0) = c_2 = d_0$ we arrive at m equations in m unknowns.

4. **Compute K and F.** Let the finite element solution be denoted by

$$u_h(x) = u_1 H_1(x) + u_2 H_2(x) + \ldots + u_m H_m(x) \,, \tag{1.152}$$

and let U be the column vector of its coefficients,

$$U = (u_1, u_2, \ldots, u_m)^T \,.$$

Inserting u_h into (1.139), particularizing for $v_h = H_a$, with $a \in \eta_a$ and incorporating the essential boundary conditions we have that, for $a, b \in \eta = \{1, \ldots, m\}$,

$$K_{ab} = \begin{cases} \delta_{ab} & \text{if } a \in \eta_g = \{1,2\}, \\ a(H_b, H_a) & \text{if } a \in \eta_a \,. \end{cases} \tag{1.153}$$

For the load vector,

$$F_a = \begin{cases} g_0 & \text{if } a = 1 \,, \\ d_0 & \text{if } a = 2 \,, \\ \ell(H_a) & \text{if } a \in \eta_a \,. \end{cases} \tag{1.154}$$

Example 1.70 Fourth-order problem with uniform mesh and constant coefficients. Let us carry out the explicit computations corresponding to a uniform mesh with mesh size $h = L/n_{\text{el}}$. For this, we bring back the definitions of $a(\cdot,\cdot)$ and $\ell(\cdot)$ from Problem 1.7 and those of the basis functions from (1.142)-(1.143). We assume that $q > 0$, $c \geq 0$ and f are constants.

A direct calculation shows that, for $a = 1, \ldots, n_{\text{vert}}$,

$$H''_{2a-1}(x) = \begin{cases} 0 & \text{if } x < x_{a-1} \\ -12(x - x_{a-1})/h^3 + 6/h^2 & \text{if } x_{a-1} < x < x_a \\ 12(x - x_a)/h^3 - 6/h^2 & \text{if } x_a < x < x_{a+1} \\ 0 & \text{if } x > x_{a+1} \end{cases}$$

$$H''_{2a} = \begin{cases} 0 & \text{if } x < x_{a-1} \\ 6(x - x_{a-1})/h^2 - 4/h & \text{if } x_{a-1} < x < x_a \\ 6(x - x_a)/h^2 - 2/h & \text{if } x_a < x < x_{a+1} \\ 0 & \text{if } x > x_{a+1} \end{cases}$$

The next step is the (tedious) computation of all the system matrix

and load vector components. The result is the following:

$$K = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ C_1 & C_2 & C_3 & 0 & C_1 & -C_2 & 0 & 0 & \dots & 0 & 0 \\ -C_2 & C_4 & 0 & C_5 & C_2 & C_4 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & C_1 & C_2 & C_3 & 0 & C_1 & -C_2 & \dots & 0 & 0 \\ 0 & 0 & -C_2 & C_4 & 0 & C_5 & C_2 & C_4 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & C_1 & C_2 & C_3 & 0 & C_1 & -C_2 \\ 0 & 0 & \dots & 0 & 0 & -C_2 & C_4 & 0 & C_5 & C_2 & C_4 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & C_1 & C_2 & C_3/2 & C_6 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & -C_2 & C_4 & C_6 & C_5/2 \end{pmatrix}$$

where

$$\begin{aligned} C_1 &= -\frac{12q}{h^3} + \frac{9ch}{70} \\ C_2 &= -\frac{6q}{h^2} + \frac{13ch^2}{420} \\ C_3 &= \frac{24q}{h^3} + \frac{26ch}{35} \\ C_4 &= \frac{2q}{h} - \frac{ch^3}{140} \\ C_5 &= \frac{8q}{h} + \frac{2ch^3}{105} \\ C_6 &= -\frac{6q}{h^2} - \frac{11ch^2}{210} \end{aligned}$$

and

$$F = \begin{pmatrix} g_0 \\ d_0 \\ fh \\ 0 \\ fh \\ 0 \\ \dots \\ fh \\ 0 \\ fh/2 - qn_L \\ qm_L \end{pmatrix}.$$

5. **Solve and Compute the Finite Element Solution.** We now solve the system $KU = F$, and then build the finite element solution as $u_h(x) = \sum_{a\in\eta} u_a N_a(x)$.

   Taking $L = 1$, $q = 10$, $c = 0$, $f = -1$, $g_0 = 0$, $d_0 = 0$, $n_L = 0$ and $m_L = 0$ we get the conditions of a bar of constant cross section, clamped on the left

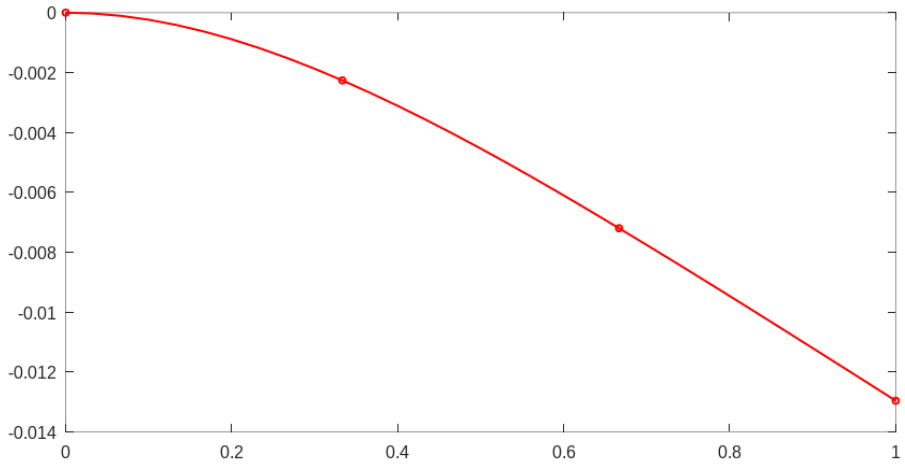


Figure 1.21 The finite element solution u_h corresponding to Equation 1.155.

boundary and free on the right boundary, with uniform load. For a small mesh with just three elements ($h = 1/3$, $n_{\text{vert}} = 4$, and thus $m = 8$) we get the solution

$$U = \begin{bmatrix} 0 \\ 0 \\ -2.2634e-03 \\ -1.2037e-02 \\ -7.2016e-03 \\ -1.6667e-02 \\ -1.2963e-02 \\ -1.7593e-02 \end{bmatrix}$$

and hence

$$u_h(x) = -0.0023H_3(x) - 0.012H_4(x) - 0.0072H_5(x) - 0.0167H_6(x) - 0.013H_7(x) - 0.0176H_8(x). \quad (1.155)$$

This function is plotted in Fig. 1.21.

1.4.5 The Cubic Hermite Finite Element

The basis functions $H_1,\ldots,H_{2n_{\text{vert}}}$ introduced in (1.142)-(1.143) can also be viewed as generated by the following finite element:

$$\Omega_e = \left[x_1^e, x_2^e\right], \tag{1.156}$$

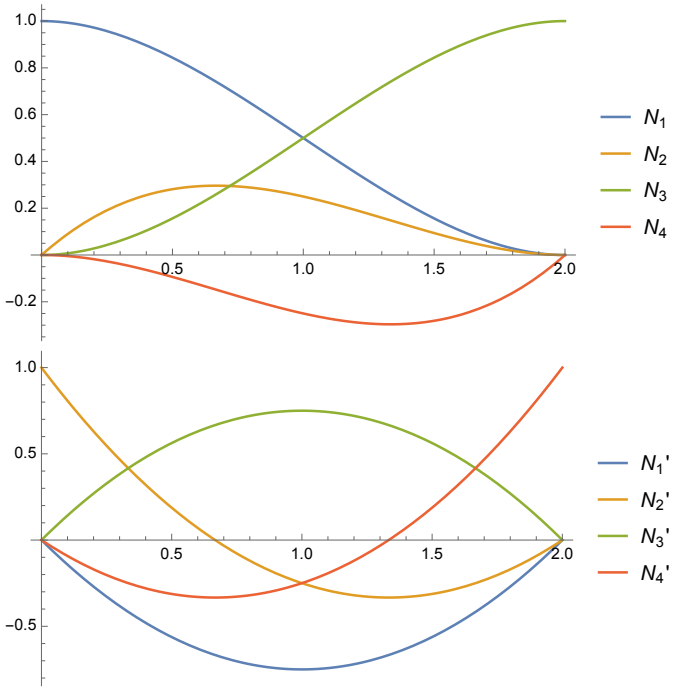
$$N_1^e(x) = \left(\frac{x_2^e - x}{x_2^e - x_1^e}\right)^2 \left(1 + 2\frac{x - x_1^e}{x_2^e - x_1^e}\right), \tag{1.157}$$

$$N_3^e(x) = \left(\frac{x_1^e - x}{x_1^e - x_2^e}\right)^2 \left(1 + 2\frac{x - x_2^e}{x_1^e - x_2^e}\right), \tag{1.158}$$

$$N_2^e(x) = \left(\frac{x_2^e - x}{x_2^e - x_1^e}\right)^2 \left(x - x_1^e\right), \tag{1.159}$$

$$N_4^e(x) = \left(\frac{x_1^e - x}{x_1^e - x_2^e}\right)^2 \left(x - x_2^e\right). \tag{1.160}$$

These functions and their derivatives are plotted next:



Any cubic polynomial in e can be written as

$$f^e(x) = \phi_1^e N_1^e(x) + \phi_2^e N_2(x) + \phi_3^e N_3(x) + \phi_4^e N_4(x)\,.$$

Furthermore, it is easy to verify that

$$N_1^e(x_1^e) = 1, \qquad (N_1^e)'(x_1^e) = 0, \qquad N_1^e(x_2^e) = 0, \qquad (N_1^e)'(x_2^e) = 0,$$

$$
\begin{aligned}
&N_2^e(x_1^e)=0, \quad (N_2^e)'(x_1^e)=1, \quad N_2^e(x_2^e)=0, \quad (N_2^e)'(x_2^e)=0,\\
&N_3^e(x_1^e)=0, \quad (N_3^e)'(x_1^e)=0, \quad N_3^e(x_2^e)=1, \quad (N_1^e)'(x_2^e)=0,\\
&N_4^e(x_1^e)=0, \quad (N_4^e)'(x_1^e)=0, \quad N_4^e(x_2^e)=0, \quad (N_4^e)'(x_2^e)=1,
\end{aligned}
$$

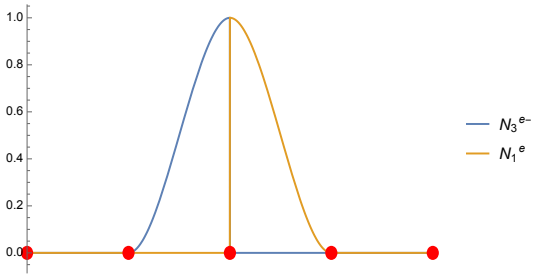
which implies that the **degrees of freedom** are

$$
\phi_1^e = f^e(x_1^e), \qquad \phi_2^e = (f^e)'(x_1^e), \qquad \phi_3^e = f^e(x_2^e), \qquad \phi_4^e = (f^e)'(x_2^e), \tag{1.161}
$$

which are the values of the function and its derivative at vertices x_1^e and x_2^e. The fact that the degrees of freedom involve not just the value of the function but also the value of its derivative is what qualifies this element as being an **Hermite** finite element. The element then has two nodes, one at each vertex. To pictorially indicate that a degree of freedom at each node is the derivative therein, we draw a ring around the node, i.e.,

These elements can easily be combined in such a way to obtain global basis functions that are $\mathscr{C}^1$ and generate the Hermite space. The basic procedure is as follows:

- The function N_1^e of element e (extended by zero to the rest of the domain) is added to the function N_3^{e-} (also extended by zero), where $e-$ is the element to the left of e (if any). The resulting function over Ω is nothing but the function $H_{2a-1}(x)$, already introduced in (1.142), assuming that a is the *left* node of e.



- For the even-numbered basis functions the construction is analogous. The functions N_2^e and N_4^{e-} are added up. The resulting function is $H_{2a}(x)$, introduced in (1.143), assuming that a is the *left* node of e.

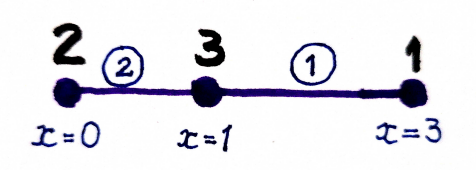
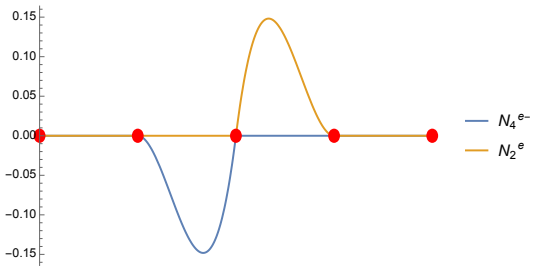


Figure 1.22 Mesh of the example in the text. The circled numbers correspond to the elements, the bare numbers to the nodes.



A key issue is that **these operations can be encoded in a local-to-global map**, so as to use **the same assembly procedure** as in Section 1.3.4. This is best explained by carefully carrying out an example.

Example 1.71

Consider a mesh of just two elements and three nodes. Furthermore, let us adopt an arbitrary numbering of nodes and elements so as to show how general the procedure is. The nodal coordinates are

$$x_1 = 3,\ x_2 = 0,\ x_3 = 1$$

and we specify that element domain number 1 is (x_3, x_1) and element domain number 2 is (x_2, x_3), as shown in Fig. 1.22. Having 3 nodes, the dimension of the cubic Hermite space that we will generate is $m = 2n_{\text{vert}} = 6$.

In this mesh, the following local-to-global map yields a basis of $\mathcal{W}_h$

$$\mathsf{LG} = \begin{pmatrix} 5 & 3 \\ 6 & 4 \\ 1 & 5 \\ 2 & 6 \end{pmatrix}.$$

In fact, from the definition

$$N_A = \mathop{\mathring{\sum}}_{\{(a,e)|\mathsf{LG}(a,e)=A\}} N_a^e$$

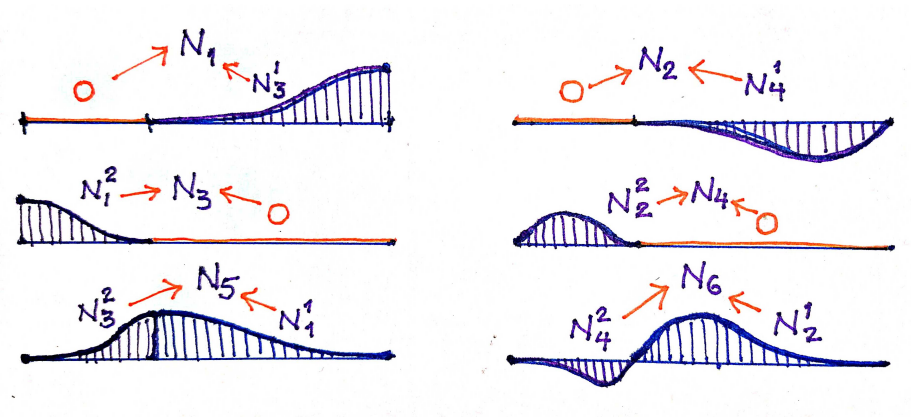


Figure 1.23 The six basis functions generated by the local-to-global map LG provided in the example.

we can write down the sum explicitly for $A = 1, \ldots, 6$, yielding

$$\begin{aligned} N_1 &= N_3^1 \,, \\ N_2 &= N_4^1 \,, \\ N_3 &= N_1^2 \,, \\ N_4 &= N_2^2 \,, \\ N_5 &= N_1^1 \mathring{+} N_3^2 \,, \\ N_6 &= N_2^1 \mathring{+} N_4^2 \,. \end{aligned}$$

These functions, together with their elementwise contributions, are shown in Fig. 1.23. They can be directly compared to the Hermite basis functions H_k introduced in (1.142)-(1.143). The local-to-global map has done its magic, since in fact we have

$$N_1 = H_5, \quad N_2 = H_6, \quad N_3 = H_1, \quad N_4 = H_2, \quad N_5 = H_3, \quad N_6 = H_4 \,.$$

The Hermite finite element, together with the local-to-global map, have generated the Hermite global basis functions. The numbering is different, but just because we numbered the nodes differently.

Exercise: Verify that with the local-to-global map

$$\mathsf{LG} = \begin{pmatrix} 3 & 1 \\ 4 & 2 \\ 5 & 3 \\ 6 & 4 \end{pmatrix}$$

with the mesh of Fig. 1.22 (without renumbering the elements) produces basis functions that satisfy $N_k = H_k$ for all $k = 1, \ldots, m$.

1.4.6 The element stiffness matrix and load vector

It is clear by now that a mesh of cubic Hermite finite elements can be "assembled," provided a correct local-to-global map is provided, into the piecewice cubic Hermite C^1 space that we denoted H_3-space and introduced as the "simplest" C^1 space and which we have seen to work quite well to solve fourth order problems.

We now look for the element matrices and vectors that will allow us to implement codes in which the mesh is arbitrary and the coefficients are not constant. This will be a significant gain in generality with respect to the method discussed in Example 1.70.

From the bilinear form (1.137b) we know that the contribution of each element is the element stiffness matrix

$$\mathsf{K}^e_{ij} = \int_{x_1^e}^{x_2^e} \left[q\,(N_j^e)"\,(N_i^e)" + c\,N_j^e\,N_i^e \right] dx. \tag{1.162}$$

Both terms in the integrand are products of the (assumed known) functions q and c by polynomials, so that in principle the integral can be computed exactly. We provide next the exact expressions that arise when q and c are **constant within the element**, equal to real numbers q_e and c_e.

The following calculations can be checked by hand:

$$\begin{aligned}
\textstyle\int_{x_1^e}^{x_2^e}(N_1^e)"\,(N_1^e)"\,dx &= \tfrac{12}{h_e^3}, & \textstyle\int_{x_1^e}^{x_2^e} N_1^e\,N_1^e\,dx &= \tfrac{13h_e}{35},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_2^e)"\,(N_1^e)"\,dx &= \tfrac{6}{h_e^2}, & \textstyle\int_{x_1^e}^{x_2^e} N_2^e\,N_1^e\,dx &= \tfrac{11h_e^2}{210},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_3^e)"\,(N_1^e)"\,dx &= -\tfrac{12}{h_e^3}, & \textstyle\int_{x_1^e}^{x_2^e} N_3^e\,N_1^e\,dx &= \tfrac{9h_e}{70},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_1^e)"\,dx &= \tfrac{6}{h_e^2}, & \textstyle\int_{x_1^e}^{x_2^e} N_4^e\,N_1^e\,dx &= -\tfrac{13h_e^2}{420},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_2^e)"\,(N_2^e)"\,dx &= \tfrac{4}{h_e}, & \textstyle\int_{x_1^e}^{x_2^e} N_2^e\,N_2^e\,dx &= \tfrac{h_e^3}{105},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_3^e)"\,(N_2^e)"\,dx &= -\tfrac{6}{h_e^2}, & \textstyle\int_{x_1^e}^{x_2^e} N_3^e\,N_2^e\,dx &= \tfrac{13h_e^2}{420},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_2^e)"\,dx &= \tfrac{2}{h_e}, & \textstyle\int_{x_1^e}^{x_2^e} N_4^e\,N_2^e\,dx &= -\tfrac{h_e^3}{140},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_3^e)"\,(N_3^e)"\,dx &= \tfrac{12}{h_e^3}, & \textstyle\int_{x_1^e}^{x_2^e} N_3^e\,N_3^e\,dx &= \tfrac{13h_e}{35},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_3^e)"\,dx &= -\tfrac{6}{h_e^2}, & \textstyle\int_{x_1^e}^{x_2^e} N_4^e\,N_3^e\,dx &= -\tfrac{11h_e^2}{210},\\
\textstyle\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_4^e)"\,dx &= \tfrac{4}{h_e}, & \textstyle\int_{x_1^e}^{x_2^e} N_4^e\,N_4^e\,dx &= \tfrac{h_e^3}{105},
\end{aligned}$$

where $h_e = x_2^e - x_1^e$. Then the (symmetric) element stiffness matrix ends up being

$$\mathsf{K}^e = \begin{pmatrix}
\frac{12q_e}{h_e^3} + \frac{13c_eh_e}{35} & \frac{6q_e}{h_e^2} + \frac{11c_eh_e^2}{210} & -\frac{12q_e}{h_e^3} + \frac{9c_eh_e}{70} & \frac{6q_e}{h_e^2} - \frac{13c_eh_e^2}{420}\\
\text{symm} & \frac{4q_e}{h_e} + \frac{c_eh_e^3}{105} & -\frac{6q_e}{h_e^2} + \frac{13c_eh_e^2}{420} & \frac{2q_e}{h_e} - \frac{c_eh_e^3}{140}\\
\text{symm} & \text{symm} & \frac{12q_e}{h_e^3} + \frac{13c_eh_e}{35} & -\frac{6q_e}{h_e^2} - \frac{11c_eh_e^2}{210}\\
\text{symm} & \text{symm} & \text{symm} & \frac{4q_e}{h_e} + \frac{c_eh_e^3}{105}
\end{pmatrix} \tag{1.163}$$

Turning now to the element load vector, we will compute it without the end contributions, which will be added later on. From (1.137d) we have

$$\mathsf{F}_i^e = \int_{x_1^e}^{x_2^e} f(x)\, N_i^e(x)\, dx, \tag{1.164}$$

which again can in principle computed exactly.

As an interesting special case, let us compute F^e explicitly for the case in which $f(x) = f_e$, constant within the element. This will allows us to solve problems with piecewise-constant distributed load.

From the straightforward integrals

$$\int_{x_1^e}^{x_2^e} N_1^e(x)\, dx = \int_{x_1^e}^{x_2^e} N_3^e(x)\, dx = \frac{h_e}{2}\,, \qquad \int_{x_1^e}^{x_2^e} N_2^e(x)\, dx = -\int_{x_1^e}^{x_2^e} N_4^e(x)\, dx = \frac{h_e^2}{12}\,,$$

we get the required expression:

$$\mathsf{F}^e = \begin{pmatrix} \frac{f_e h_e}{2} \\ \\ \frac{f_e h_e^2}{12} \\ \\ \frac{f_e h_e}{2} \\ \\ -\frac{f_e h_e^2}{12} \end{pmatrix} \tag{1.165}$$

These expressions can be coded in the element routine:

```
function [Ke, Fe]=elementKandF(xe,qe,ce,fe)
  he=xe(2)-xe(1);
  qh=qe/he;qh2=qh/he;qh3=qh2/he;
  ch=ce*he;ch2=ch*he;ch3=ch2*he;
  Ke=[12*qh3+13*ch/35, 6*qh2+11*ch2/210, -12*qh3+9*ch/70, 6*qh2-13*ch2/420;...
      6*qh2+11*ch2/210, 4*qh+ch3/105, -6*qh2+13*ch2/420, 2*qh-ch3/140;...
      -12*qh3+9*ch/70, -6*qh2+13*ch2/420, 12*qh3+13*ch/35, -6*qh2-11*ch2/210;...
      6*qh2-13*ch2/420, 2*qh-ch3/140, -6*qh2-11*ch2/210, 4*qh+ch3/105];
  fh=fe*he;fh2=fh*he;
  Fe=[fh/2; fh2/12; fh/2; -fh2/12];
end
```

1.4.7 Solving Fourth-order Elliptic Problems with H_3 Hermite Finite Elements

We assume that a mesh of H_3 finite elements is provided by means of a **list of coordinates** X and a **local-to-global map** LG.

The specified values g_0, d_0, T and F are also provided, together with the piecewise constant values for q_e, c_e and f_e.

We are thus in a position to code the assembly of the stiffness matrix and load vector. We follow the same procedure as in the case of P_1 elements. Notice

that, as before, we are looking for an array $\mathsf{U} = (u_1, u_2, \ldots, u_{2n_{\text{nod}}})^T$ that defines the Galerkin solution as

$$u_h(x) = u_1 N_1(x) + u_2 N_2(x) + \ldots + u_{2n_{\text{nod}}} N_{2n_{\text{nod}}}(x)$$

where we used $n_{\text{nod}} = n_{\text{vert}}$, since the number of nodes is equal to the number of vertices in this case.

The code starts by identifying n_{nod}, n_{el} and m from the data, and initializing K and F to zero.

```
nod=length(X); nunk=2*nod; nel=size(LG,2);
K=zeros(nunk,nunk);F=zeros(nunk,1);
```

Then, assuming that the elementwise values of $q(x)$, $c(x)$ and $f(x)$ are stored in the arrays `qq`, `cc` and `ff`, respectively, we proceed to assemble the contributions of the element stiffness matrices and load vectors.

```
for iel=1:nel
%% setting the local data
  lge=LG(:,iel);
  xe(1,1:npe)=X(1,iel:iel+1);
  qe=qq(iel);   ce=cc(iel);  fe=ff(iel);
%% computing element K and F
  [Ke Fe]=elementKandF(xe,qe,ce,fe);
%% assembly, from local to global
  for ii=1:4
     if (sum(EtaG==lge(ii))==0)
        for jj=1:4
            K(lge(ii),lge(jj))+=Ke(ii,jj);
        end
        F(lge(ii))+=Fe(ii);
     end
  end
end
```

Notice that this procedure is exactly the same as that used for all other finite element spaces.

Finally, we impose the **essential boundary conditions**, i.e., the specified values (array `GG`) of the unknowns listed in η_g (array `EtaG`)

```
ng=length(EtaG);
for ig=1:ng
 K(EtaG(ig),EtaG(ig))=1;
 F(EtaG(ig))=GG(ig);
end
```

and the **natural boundary conditions**, i.e., the torque T and the force F

```
% adding natural boundary conditions at last two unknowns
F(nunk-1)+=FL;
F(nunk)+=TL;
```

With this, we can compute the coefficients $u_1, u_2, \ldots, u_{2n_{\text{nod}}}$ by solving the linear system $\mathsf{KU} = \mathsf{F}$.

```
%% solve algebraic system
U=K\F;
```

Omitting the input and output sections of the code, the whole finite element procedure consists of about 40 lines of Octave/MATLAB code.

Example 1.72 (The Euler-Bernoulli beam equation with non-constant bending rigidity and non-uniform mesh)

Consider as example a beam with the same parameters as that plotted in Fig. 1.21, but now we will use a non-uniform mesh with the following array of coordinates

$$\mathsf{X} = \begin{pmatrix} 0 & 0.25 & 0.4 & 0.6 & 0.65 & 0.9 & 1 \end{pmatrix},$$

so that $n_{\text{nod}} = 7$, $n_{\text{el}} = 6$, and a suitable local-to-global map, such as the natural one

$$\mathsf{LG} = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 & 11 \\ 2 & 4 & 6 & 8 & 10 & 12 \\ 3 & 5 & 7 & 9 & 11 & 13 \\ 4 & 6 & 8 & 10 & 12 & 14 \end{pmatrix}.$$

Because of the boundary conditions we have $\eta_g = \{1,2\}$ with $\mathsf{GG} = (0,0)$, and natural boundary conditions on the right $T = 0$ and $F = 0$.

Running the code one obtains the solution shown in Fig. 1.24, which is very similar to that of Fig. 1.21 up to small discretization errors. But notice that, in this case, the mesh is non-uniform.

Furthermore, we can vary the values of the bending rigidity of each element at will. Notice that element number 4 is quite small, it goes from $x = 0.6$ to $x = 0.65$. Let us change the bending stiffness of just this element to 1/100-th that of the rest of the beam, from $q_e = 10$ to $q_e = 0.1$. The solution radically changes, as seen in Fig. 1.25. Element number 4, being less stiff, acts as a hinge at which most of the deformation concentrates.

Example 1.73 (Image denoising) We can apply the same method and code as before to an image denoising problem by solving the equation

$$q\,u''''(x) + u(x) = u_0(x),$$

where u_0 is the raw image and q an adjustable parameter, with homogeneous natural boundary conditions ($T = F = 0$ at both ends). We use the H_3-space with one element per pixel in the image. The results are shown in Figure 1.26. The denoising effect of the fourth-order term is evident. A value $q = 10^{-6}$ seems the most appropriate for this image. For $q > 10^{-5}$ the solution is too smoothed, loosing the underlying signal. For $q < 10^{-7}$ the solution follows the local noise, which is not sufficiently removed.

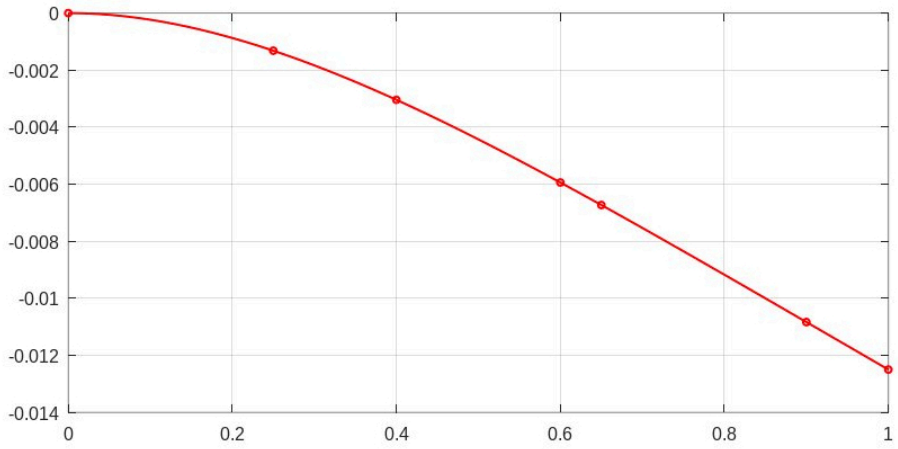


Figure 1.24 Finite element solution from Example 1.72.

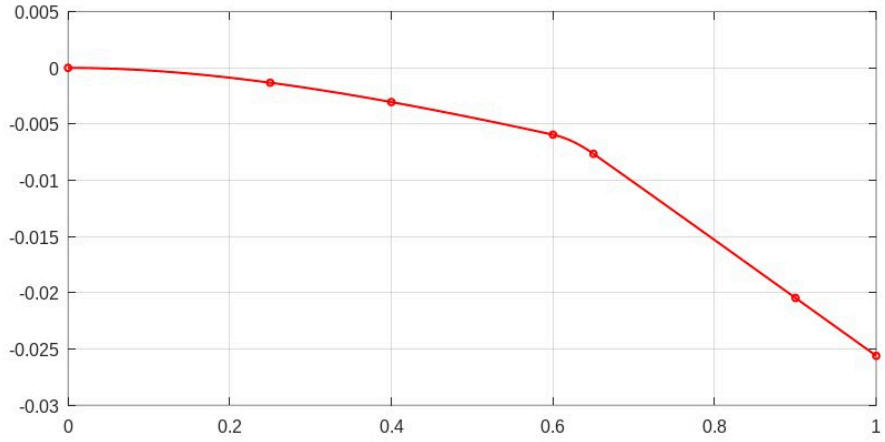


Figure 1.25 Finite element solution from Example 1.72. The bending rigidity of the beam is equal to 10 in all elements except for element 4, where its value has been reduced to 0.1.

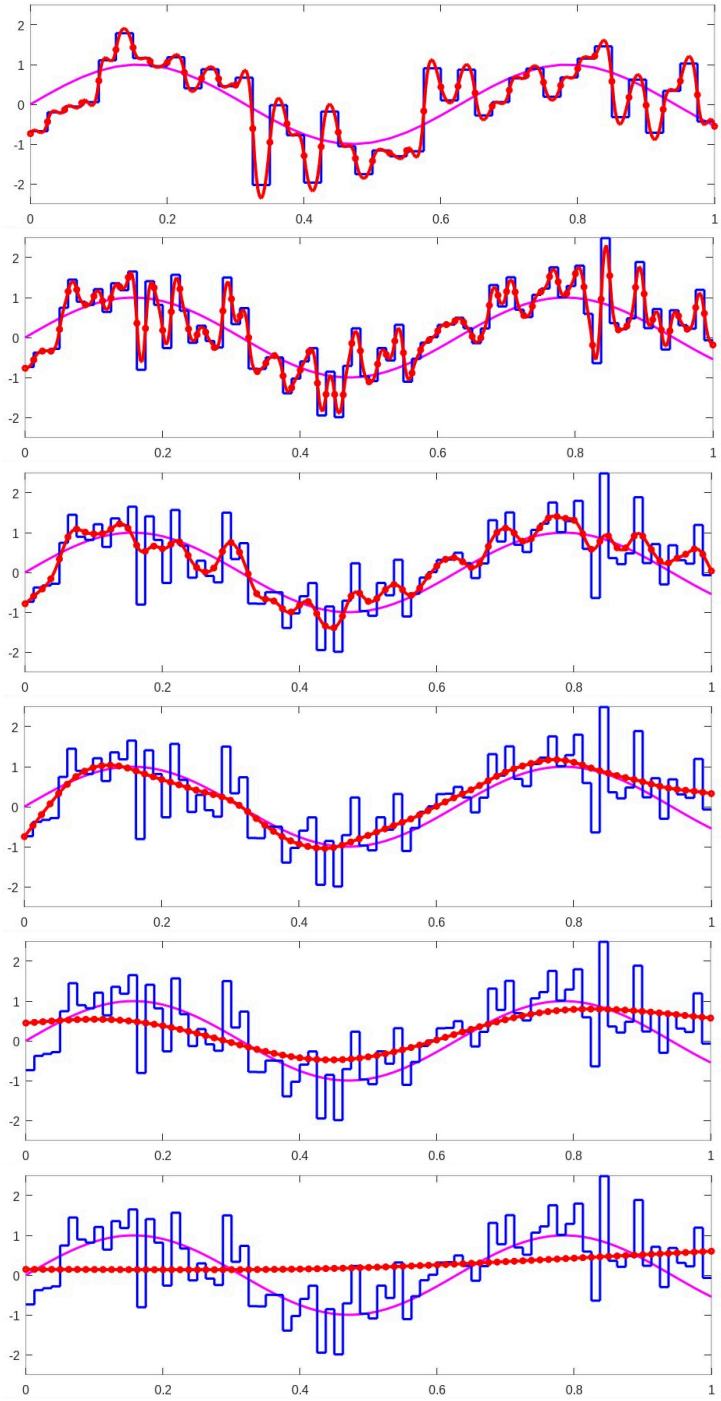


Figure 1.26 Results from the image denoising example. In blue the original image u_0, which is a random perturbation of the exact function drawn in magenta. In red we plot the solution u_h of the image-denoising example for q taking values, from top to bottom, $q = 0$, 10^{-10}, 10^{-8}, 10^{-6}, 10^{-4} and 10^{-2}.

Chapter 2

Diffusion Problems in 2D

We now turn to consider diffusion problems in two dimensions, which govern heat conduction in solids, electrostatics and some mass transfer situations. We follow the same methodology as in Chapter 1, beginning with the strong and weak forms of the mathematical problem, discussing the novelties brought by the higher dimensionality of the domain and finally introducing the simplest finite element method to compute an approximate solution.

2.1 The Strong Form of the Boundary Value Problem

Let us define a general **diffusion equation**

$$-\operatorname{div}(K\nabla u) = f\,, \tag{2.1}$$

to be satisfied in a domain $\Omega \subset \mathbb{R}^2$. It is convenient to denote the coordinates by $x = (x_1, x_2)^T$ instead of $(x, y)^T$, and the partial derivatives of a function $u(x_1, x_2)$ by $\partial_1 u$ and $\partial_2 u$. The components of a vector v in a basis to be specified will be likewise denoted by $(v_1, v_2)^T$.

Above, div is the divergence operator which applied to a vector $v(x)$ yields

$$\operatorname{div} v = \partial_1 v_1 + \partial_2 v_2,$$

K is a positive-definite symmetric matrix (all eigenvalues are positive), ∇u is the gradient vector

$$\nabla u = (\partial_1 u\,, \partial_2 u\,)^T$$

and f is a source density (per unit area). When u represents the temperature of a solid, (2.1) is known as the **heat conduction equation**.

Of utmost physical importance is the **diffusive flux** (or **heat flux** in the thermal setting), defined as

$$J = -K\nabla u\,, \tag{2.2}$$

which obeys $\operatorname{div} J = f$. In the context of the heat conduction equation, this relationship is called **Fourier's law**. In the context of mass transport, u is the concentration of mass and (2.2) is called **Fick's law**.

All the previous expressions have been written in operator form, which is a concise way of writing formulae involving partial derivatives. They can of course be rewritten in coordinates, as

$$-\sum_{i=1}^{2}\partial_i\left(\sum_{j=1}^{2}K_{ij}\partial_j u\right) = f\,, \tag{2.3}$$

$$J_i = -\sum_{j=1}^{2}K_{ij}\partial_j u \tag{2.4}$$

Example 2.1 (The Poisson equation) When K is a multiple of the identity matrix,

$$K(x) = k(x)\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

we say that the diffusive medium is **isotropic**. If we further assume that k is independent of x we have the case in which (2.1) is a **Poisson equation**. In such a case the expressions simplify considerably. Since $K_{ij} = k\delta_{ij}$, we have

$$K = \begin{pmatrix}k & 0\\ 0 & k\end{pmatrix} \tag{2.5}$$

$$J = -K\nabla u = -k\nabla u = \begin{pmatrix}-k\partial_1 u\\ -k\partial_2 u\end{pmatrix} \tag{2.6}$$

and thus the equation reads

$$\partial_{11}^2 u + \partial_{22}^2 u = -\frac{f}{k}\,. \tag{2.7}$$

The notation of the second partial derivatives is

$$\partial_{ij}^2 u = \frac{\partial^2 u}{\partial x_i \partial x_j}$$

and one can recognize in the left-hand side of (2.7) the **Laplacian** of u, namely

$$\Delta u = \left(\partial_{11}^2 + \partial_{22}^2\right)u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}\,. \tag{2.8}$$

There exist many analytical solutions of $\Delta u = -f/k$ even if $f = c$ is a constant. In the case $c = 0$ the following functions are solutions for any α, β and γ real numbers,

$$u(x_1, x_2) = \alpha + \beta x_1 + \gamma x_2\,. \tag{2.9}$$

In other words, all *affine functions* are solutions to $\Delta u = 0$. This is also the case in 1D, where all functions of the form $u(x) = \alpha + \beta x$ are solutions to

$u'' = 0$. There is a crucial difference though: In 1D the affine functions are *all possible solutions* to $u'' = 0$, while in 2D there are infinitely many linearly independent functions that satisfy $\Delta u = 0$, also called **harmonic functions**. For example, the function

$$u(x_1, x_2) = \ln\left((x_1 - X_1)^2 + (x_2 - X_2)^2\right) \tag{2.10}$$

defined for all $x \neq X$ satisfies $\Delta u = 0$ in $\mathbb{R}^2 \setminus X$, for *any choice of* X. These functions are smooth in Ω if $X \notin \overline{\Omega}$.[1] Notice that two such functions with different choices of X are linearly independent, so that the set of harmonic functions has infinite dimensions.

Two-dimensional domains. Going back to (2.1), it is assumed to hold at all points x of the **two-dimensional domain** Ω. In 1D the domains could be intervals, or at most groups of intervals. The variety of domains in 2D is certainly much higher. The shape of the domain usually comes from the geometry of the physical system under study. The theory and methods we describe below hold for **bounded** domains that do not have cusps or cracks. A visual guide of the **admissible domains** is given in Fig. 2.1. To avoid technicalities, however, we restrict until further notice to **polygonal domains**, with the possibility of them having one or several polygonal holes. The **boundary** of a domain Ω, denoted as $\partial\Omega$, is a closed non-intersecting polygonal line (if the domain is simply connected) or a finite number of such lines.

Boundary conditions. Boundary conditions are necessary to uniquely identify a solution of (2.1). A salient feature of elliptic second-order problems is that at all points in $\partial\Omega$ one (and only one) information is needed about the solution u. We assume that $\partial\Omega$ is decomposed into two parts, depending on the available boundary information.

- If at $x \in \partial\Omega$ we know the **value** of u, we say that x belongs to the **Dirichlet boundary** $\partial\Omega_D$.
- On the other hand, if at x we know the value of the **normal flux** $J \cdot \check{n}$, with $\check{n}$ the **exterior unit normal to** $\partial\Omega$ at x, we say that x belongs to the **Neumann boundary** $\partial\Omega_N$.
- There exist other possibilities, such as knowing the value of a linear combination of the function and the normal flux. These would be Robin boundary conditions, that will not be considered here.

We assume that both $\partial\Omega_D$ and $\partial\Omega_N$ consist of a subset of edges of $\partial\Omega$, which is a polygonal line (or several).

The problem we aim at solving, at least approximately, reads as follows.

[1] The set $\overline{\Omega}$ is the **closure** of Ω, or the set of points in Ω and all those points that can be reached as limits of sequences of points in Ω; as in the discussion on the sparsity of the stiffness matrix in §1.3.4.

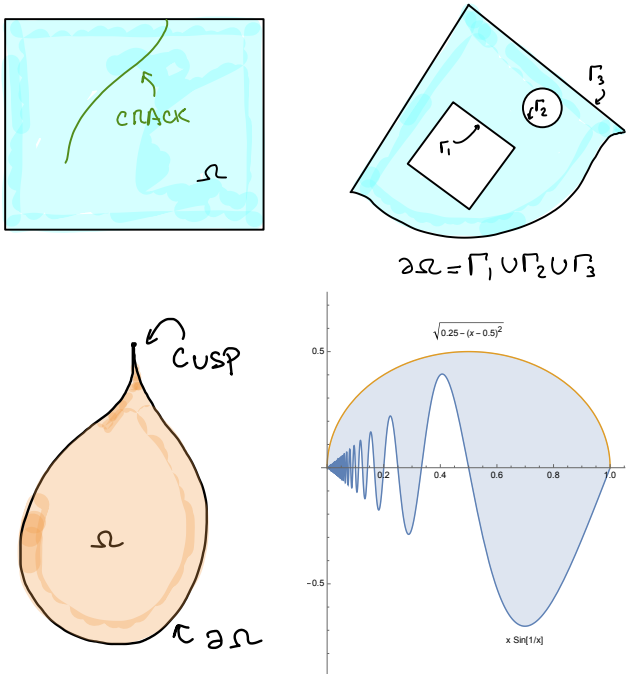


Figure 2.1 Domains that we will consider can contain smooth cracks, holes, and have a boundary made of a finite number of smooth curves, each of finite length, such as polygonal domains, as shown in the top row. Domains that we will not consider have boundaries that form cusps, have boundaries that have infinite length, or that do not have normal to the boundary defined at almost every point of the boundary, as illustrated in the bottom row. Typical engineering domains are idealized to be of the type in the top row.

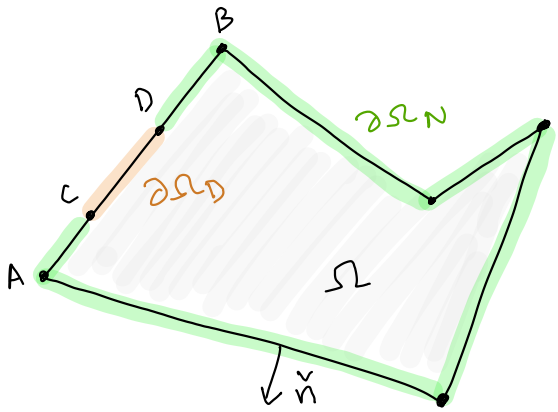


Figure 2.2 Sketch of the domain for Problem 2.1

Problem 2.1 (Strong Form of the 2D Diffusion Problem)**.** *Given the coefficients K and f as functions of $x \in \Omega$, and given a real function g defined in $\partial\Omega_D$ and another real function H defined in $\partial\Omega_N$, determine a function $u : \Omega \to \mathbb{R}$ satisfying*

$$-div\,(K\nabla u) = f(x) \qquad \forall\, x \in \Omega \tag{2.11a}$$

$$u = g \qquad \forall\, x \in \partial\Omega_D \tag{2.11b}$$

$$(K\nabla u)\cdot \check{n} = H \qquad \forall\, x \in \partial\Omega_N \tag{2.11c}$$

Problem 2.1 admits one and only one solution under sufficient regularity of the data (in particular, g must be continuous) plus the two essential hypotheses

H1) K is everywhere a bounded, positive definite matrix, with all eigenvalues greater than some $\kappa_0 > 0$, and

H2) the length of $\partial\Omega_D$ is strictly positive.

If K is a multiple of the identity matrix, i.e., $K(x) = k(x)\mathbf{I}_{2\times 2}$, then H1 requires that $k(x) > \kappa_0 > 0$ for all $x \in \Omega$. Concerning H2, it requires that u is known not just at a point or a finite set of points of $\partial\Omega$, the condition $u = g$ must hold all along the full length of an edge of $\partial\Omega$. Notice that if $\partial\Omega_D$ is just a segment $\overline{CD}$ within a larger edge $\overline{AB}$ one can redefine the polygon incorporating C and D as vertices, so that $\partial\Omega_D$ is a full edge.

Example 2.2 (A uniformly heated rod) Consider the circular cross section of a homogeneous and isotropic rod, in which heat is generated uniformly at rate f. The domain is thus $\Omega = \{(x_1, x_2) \in \mathbb{R}^2,\ x_1^2 + x_2^2 < R^2\}$.

Let $g \in \mathbb{R}$ be the temperature at the rod's surface, assumed uniform. We are interested in the temperature field **inside** the rod.

The corresponding differential equation is

$$\Delta u = -\frac{f}{k}\,,$$

so, we are looking for $u(x,y)$, satisfying $u = g$ on the circle $x_1^2 + x_2^2 = R^2$ and having (constant) Laplacian equal to $-f/k$ in the enclosed region. The Dirichlet boundary $\partial\Omega_D$ is the whole boundary $\partial\Omega$ of the domain, and thus $\partial\Omega_N$ is empty. The constant k is assumed positive, so that both H1 and H2 are satisfied.

It is easy to check that

$$u(x,y) = g - \frac{f}{4k}\left(x_1^2 + x_2^2 - R^2\right)$$

satisfies these conditions and is thus the unique solution to the boundary value problem.

By differentiating u we can compute the heat flux

$$J = -k\nabla u = -k\left(-\frac{f x_1}{2k}, -\frac{f x_2}{2k}\right)^T = \frac{f}{2}x$$

and see that it is constant along the boundary circle, pointing outwards and with magnitude $fR/2$.

The maximum temperature takes place at the center (if $f > 0$), with value $g + fR^2/(4k)$.

Important: The same solution $u(x_1, x_2)$ also satisfies the **Neumann** boundary value problem, in which the **normal flux** is specified as

$$(k\nabla u)\cdot \check{n} = -\frac{fR}{2}$$

over the boundary circle. However, this problem **does not satisfy H2** (because $\partial\Omega_D$ is empty) and in facts admits not just u as solution but also any function $v = u + C$, with C an arbitrary real constant.

2.2 The weak form

As already discussed in the 1D case, the finite element method is built upon a *weak* formulation of the problem under consideration, which at present is Problem 2.1. Getting to the weak formulation in general involves integration by parts, so let us recall a useful result.

Theorem 2.1. (Integration by parts in 2D or 3D) *Let w be a smooth vector field in Ω (an admissible domain), and v a smooth scalar function. Then,*

$$\int_\Omega v \operatorname{div} w \, d\Omega = \int_{\partial\Omega} v\, w\cdot \check{n}\, d\Gamma - \int_\Omega w\cdot\nabla v \, d\Omega\,. \tag{2.12}$$

Now, applying the same procedure as in 1D, we multiply the differential equation (2.11a) by a smooth $v : \Omega \to \mathbb{R}$ and integrate over Ω, to get

$$-\int_\Omega \operatorname{div}(K\nabla u)\, v \, d\Omega = \int_\Omega f\, v \, d\Omega \,.$$

Using (2.12) with $w = K\nabla u$ for the left-hand side and decomposing the integral over $\partial\Omega$ into the sum of $\int_{\partial\Omega_D}$ and $\int_{\partial\Omega_N}$, we arrive at

$$\int_\Omega (K\nabla u)\cdot\nabla v \, d\Omega = \int_\Omega f\, v \, d\Omega + \int_{\partial\Omega_N} v\, K\nabla u\cdot\check{n}\, d\Gamma + \int_{\partial\Omega_D} v\, K\nabla u\cdot\check{n}\, d\Gamma \,. \quad (2.13)$$

In the integral over $\partial\Omega_N$, $K\nabla u\cdot\check{n}$ can be replaced by H, since (2.11c) holds. The Neumann conditions are **natural**. On the other hand, in the integral over $\partial\Omega_D$ we have no way to know $K\nabla u\cdot\check{n}$, *but we know that* $u = g$ *there*. The Dirichlet boundary condition needs to be imposed as an **essential** boundary condition, meaning that it is incorporated into the definition of the **trial space** as

$$\mathscr{S} = \left\{ v\colon \Omega \to \mathbb{R} \text{ smooth } \mid v(x) = g(x) \text{ for all } x \in \partial\Omega_D \right\} . \quad (2.14)$$

The **test space** (i.e., the direction of $\mathscr{S}$) is then given by

$$\mathscr{V} = \{ v\colon \Omega \to \mathbb{R} \text{ smooth } \mid v(x) = 0 \text{ for all } x \in \partial\Omega_D \} \,, \quad (2.15)$$

so that the last integral in (2.13) is zero. The solution u, which is assumed to be smooth, thus satisfies the following weak formulation:

Problem 2.2. (Weak Form of the 2D Diffusion Problem) *Find* $u \in \mathscr{S}$ *such that*

$$a(u,v) = \ell(v) \qquad \forall\, v \in \mathscr{V} \,, \quad (2.16)$$

where the bilinear and linear forms are given by

$$a(u,v) = \int_\Omega (K\nabla u)\cdot\nabla v \, d\Omega \,, \quad (2.17)$$

$$\ell(v) = \int_\Omega f\, v \, d\Omega + \int_{\partial\Omega_N} H\, v \, d\Gamma \,. \quad (2.18)$$

As you can see, the weak form involves just first-order derivatives (the gradient ∇), but instead of having the pointwise requirement that $\operatorname{div}(K\nabla u) + f = 0$ at all points we have integral expressions in two dimensions that must hold for all $v \in \mathscr{V}$.

As in previous examples, both $\mathscr{S}$ and $\mathscr{V}$ are subspaces of a space $\mathscr{W}$ where the bilinear form and the linear functional are defined, in this case,

$$\mathscr{W} = \{ v\colon \Omega \to \mathbb{R} \text{ smooth} \}.$$

We can see then that $\mathscr{S}$ is an affine subspace of $\mathscr{W}$ with direction $\mathscr{V}$. The meaning of smoothness of functions in $\mathscr{W}$ for this two-dimensional case is subtler than in one dimension, but at the very least functions in $\mathscr{W}$ cannot have discontinuities across any smooth curve. As with the harmonic function involving a ln above, they could have discontinuities at points.

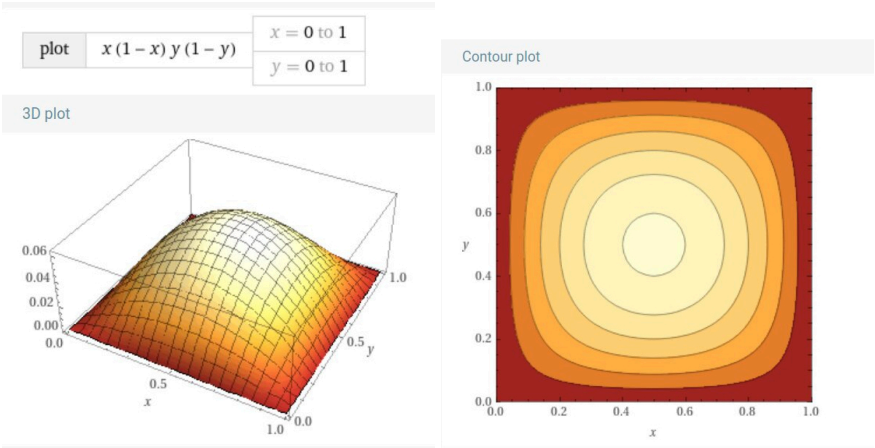


Figure 2.3 The function $N_1(x_1, x_2)$ when $L = 1$. Its maximum value (at $(1/2, 1/2)$) is $1/16$.

2.3 Galerkin method

As in the 1D case, the Galerkin Method for finding an approximation to the exact solution u consists of:

Let $\mathcal{W}_h \subset \mathcal{W}$. Find $u_h \in \mathcal{S}_h = \mathcal{S} \cap \mathcal{W}_h$ such that

$$a(u_h, v_h) = \ell(v_h) \tag{2.19}$$

for all $v_h \in \mathcal{V}_h = \mathcal{V} \cap \mathcal{W}_h$.

There are, however, significant differences with the one-dimensional case:

- The spaces $\mathcal{S}_h$ and $\mathcal{V}_h$ must now be composed of two-dimensional functions.
- The domain boundary is a closed line, which we assume to be a polygon for simplicity. The construction of the basis of $\mathcal{V}_h$ must ensure that the functions are zero on the boundary $\partial\Omega_D$.
- The requirement $\mathcal{S}_h \subset \mathcal{W}$ (and thus $\mathcal{V}_h \subset \mathcal{W}$), when the finite element space $\mathcal{W}_h$ consists of piecewise polynomials, translates into: **The functions in $\mathcal{S}_h$ and $\mathcal{V}_h$ must be continuous**. So, if the domain is subdivided into element domains, continuity of the global basis functions between elements must be enforced. This continuity, contrary to the 1D case, must hold not just at the nodes but along all edges of the subdivision.

Let us begin by discussing an example of using a global (instead of piecewise) polynomial basis on Ω, which is possible when Ω is a square.

Example 2.3 (Uniformly heated square rod) Let us revisit Example 2.2 but now considering a square geometry, so that $\Omega = \{(x_1, x_2) \in \mathbb{R}^2, 0 < x_1 < L, 0 <$

$x_2 < L\}$. The surface temperature is $g \in \mathbb{R}$ and the governing equation is $\Delta u = -f/k$ as before, where the heat source is $f \in \mathbb{R}$.

We want to select the space $\mathcal{W}_h$ as a subset of

$$\mathbb{P}_r(\Omega) = \{\text{polynomials of degree} \leq r \text{ in two variables in } \Omega\}\,. \tag{2.20}$$

If $r = 2$ this space consists of functions of the form

$$v(x_1, x_2) = c_1 + c_2 x_1 + c_3 x_2 + c_4 x_1^2 + c_5 x_1 x_2 + c_6 x_2^2\,,$$

if $r = 3$ the following terms are added

$$\ldots + c_7 x_1^3 + c_8 x_1^2 x_2 + c_9 x_1 x_2^2 + c_{10} x_2^3\,,$$

and so on. We will use this example to illustrate that the choice of a basis for $\mathcal{W}_h$ that can easily accommodate the constraints on $\mathcal{V}_h$ and $\mathcal{S}_h$ is not always trivial.

Both spaces $\mathcal{V}_h = \mathcal{V} \cap \mathcal{W}_h$ and $\mathcal{S}_h = \mathcal{S} \cap \mathcal{W}_h$ require functions to be constant along the boundary of the domain, that is, whenever x_1 or x_2 are either equal to 0 or equal to L. If, for example, we choose $r = 1$, then $p(x_1, x_2) = c_1 + c_2 x_1 + c_3 x_2$ for any $(c_1, c_2, c_3) \in \mathbb{R}^3$. The fact that p is constant at $x_2 = 0$ implies that $\partial^i p/\partial x_1^i(x_1, 0) = 0$ for all $i \in \mathbb{N}$ and any $x_1 \in [0, L]$. A similar argument can be made for the boundary conditions at $x_2 = L$, $x_1 = 0$ and $x_1 = L$. Hence,

$$\begin{aligned}
0 &= \frac{\partial p}{\partial x_1}(x_1, 0) = c_2,\\
0 &= \frac{\partial p}{\partial x_1}(x_1, L) = c_2,\\
0 &= \frac{\partial p}{\partial x_2}(0, x_2) = c_3,\\
0 &= \frac{\partial p}{\partial x_2}(L, x_2) = c_3,
\end{aligned}$$

and in this case we only need to evaluate the first derivative. As a result, we conclude that functions in $\mathcal{V}_h$ or $\mathcal{S}_h$ need $c_2 = c_3 = 0$, so they can only be constant functions. Then, if $r = 1$, $\mathcal{V}_h = \mathcal{V} \cap \mathcal{W}_h = \{0 \cdot N_2\}$ and $\mathcal{S}_h = \mathcal{S} \cap \mathcal{W}_h = \{g \cdot N_2\}$, where $N_2(x_1, x_2) = 1$ constant for all $x \in \Omega$. These spaces contain a single function each, so they are useless for any approximation.

In a similar way, it can be verified that, if $r < 4$, $\mathcal{V}_h = \{0 \cdot N_2\}$ and $\mathcal{S}_h = \{g \cdot N_2\}$. For $r \geq 4$, due to the simplicity of the geometry, we have that the set of function in $\mathbb{P}_r(\Omega)$ that are constant on $\partial\Omega$ is (see the explanation after the example)

$$\mathcal{W}_h = \{v(x_1, x_2) = c_1 + \underbrace{x_1(L - x_1)x_2(L - x_2)}_{=0 \text{ on } \partial\Omega}\, p(x_1, x_2) \mid p \in \mathbb{P}_{r-4}, c_1 \in \mathbb{R}\},$$

and this will be our choice for $\mathcal{W}_h$. **Take** $r = 4$, which is the simplest case. Then $p(x_1, x_2)$ is a constant and $\mathcal{V}_h$ **has dimension 1**. Define the basis function

$$N_1(x_1, x_2) = x_1(L - x_1)x_2(L - x_2), \text{ so that } \nabla N_1 = \begin{pmatrix} (L - 2x_1)x_2(L - x_2) \\ x_1(L - x_1)(L - 2x_2) \end{pmatrix}.$$

Therefore, we can write

$$\mathcal{W}_h = \text{span}(N_1, N_2).$$

In particular, functions in $\mathcal{V}_h$ are zero on $\partial\Omega$, so

$$\mathcal{V}_h = \{v_1 N_1 \mid v_1 \in \mathbb{R}\}$$

and functions in $\mathcal{S}_h$ are equal to g on $\partial\Omega$, so

$$\mathcal{S}_h = \{v_1 N_1 + g N_2 \mid v_1 \in \mathbb{R}\}.$$

We then have $\eta_a = \{1\}$, $\eta_g = \{2\}$, we can choose $\overline{u}_h = g N_2$, and

$$u_h(x_1, x_2) = u_1 N_1(x_1, x_2) + g N_2(x_1, x_2) = g + u_1 N_1(x_1, x_2).$$

For u_h to satisfy (2.19) with $a(\cdot, \cdot)$ given by (2.17) and $\ell(\cdot)$ given by (2.18) it must hold that

$$\int_\Omega k\nabla(g + u_1 N_1) \cdot \nabla N_1 \, dx_1 dx_2 = \int_\Omega f N_1 \, dx_1 dx_2 .$$

Here we directly used that $u_2 = g$, so we do not need to add an equation for the constrained index.

Noticing that $\nabla g = 0$ and taking u_1 out of the integral by linearity, the final equation to compute $U = [u_1]$ is

$$KU = F$$

where the 1×1 stiffness matrix and load vector are

$$K = \int_\Omega k\nabla N_1 \cdot \nabla N_1 \, dx_1 dx_2, \qquad F = \int_\Omega f N_1 \, dx_1 dx_2 .$$

Performing the double integrals we obtain

$$K = \frac{kL^8}{45}, \quad F = \frac{fL^6}{36}, \quad \text{and thus} \quad u_1 = \frac{5f}{4kL^2}.$$

This means that the Galerkin solution is

$$u_h(x_1, x_2) = g + \frac{5f}{4kL^2} N_1(x_1, x_2) = g + \frac{5f}{4kL^2} x_1(L - x_1)x_2(L - x_2) .$$

The maximum temperature takes place at the center (if $f > 0$), with value $g + 5fL^2/(64k)$. The temperature contours are shown in Fig. 2.4, where we compare our Galerkin solution (obtained by solving an equation with just one unknown!) with the exact solution. They are qualitatively very similar. The maximum difference is, in fact, less than 6%.

It is important to remark that the Galerkin solution u_h is **not** an exact solution of the differential equation. To check this, simply compute

$$\Delta u_h(x_1, x_2) = -\frac{5f}{2kL^2}\,[x_1(L - x_1) + x_2(L - x_2)]$$

and compare to the exact equation $\Delta u = -f/k$. The **residual of the differential equation**, evaluated on the Galerkin solution, is

$$r_h = \Delta u_h(x_1, x_2) + \frac{f}{k} = \frac{f}{k}\left(1 - \frac{5}{2}\frac{x_1(L - x_1) + x_2(L - x_2)}{L^2}\right).$$

Its average value over the domain is $f/(6k)$. The residual is maximum at the vertices, where its value is f/k. At the center the value is $-1/4$. The residual is plotted in Fig. 2.5. It must not be mistaken for the actual **error of the approximate solution** $e_h = u - u_h$, which in this case we can compute because the exact solution is known and is also shown in the same figure.

Functions in $\mathbb{P}_r(\Omega)$, $r \geq 4$ that are constant on $\partial\Omega$, where $\Omega = [0, L] \times [0, L]$

To obtain the result, we will repeatedly use the following observation. Let $q \in \mathbb{P}_r(\Omega)$, and subtract a constant c_1 so that $q_r(x_1, x_2) = q(x_1, x_2) - c_1$ is equal to zero on $\partial\Omega$. Let $h(x_1, x_2) = h_0 + h_1 x_1 + h_2 x_2$ for $h_0, h_1, h_2 \in \mathbb{R}$ such that either h_1 or h_2 are not zero, and such that if $h(\bar{x}_1, \bar{x}_2) = 0$ then $q(\bar{x}_1, \bar{x}_2) = 0$. Then,

$$q_r(x_1, x_2) = h(x_1, x_2)\, q_{r-1}(x_1, x_2), \tag{2.21}$$

where $q_{r-1}(x_1, x_2)$ is a polynomial of degree $r - 1$. To see this, without loss of generality assume that $h_1 \neq 0$, and let $z = h(x_1, x_2)$, so that $x_1 = g(z, x_2) = (z - h_2 x_2 - h_0)/h_1$. Consider then the polynomial $\hat{q}_r(z, x_2) = q_r(g(z, x_2), x_2)$, which satisfies that $\hat{q}_r(0, x_2) = 0$ for any x_2. Then, it admits a factorization of the form

$$\hat{q}_r(z, x_2) = z\,\hat{q}_{r-1}(z, x_2)$$

for a polynomial $\hat{q}_{r-1}$ of degree $r - 1$. Defining $q_{r-1}(x_1, x_2) = \hat{q}_{r-1}(h(x_1, x_2), x_2)$, we arrive to (2.21).

We can apply this to our case, by sequentially considering $h(x_1, x_2)$ equal to $x_1, L - x_1, x_2$ and $L - x_2$. It then follows that

$$q_r(x_1, x_2) = x_1(L - x_1)x_2(L - x_2)\, q_{r-4}(x_1, x_2), \tag{2.22}$$

where q_{r-4} is a polynomial of degree $r - 4$.

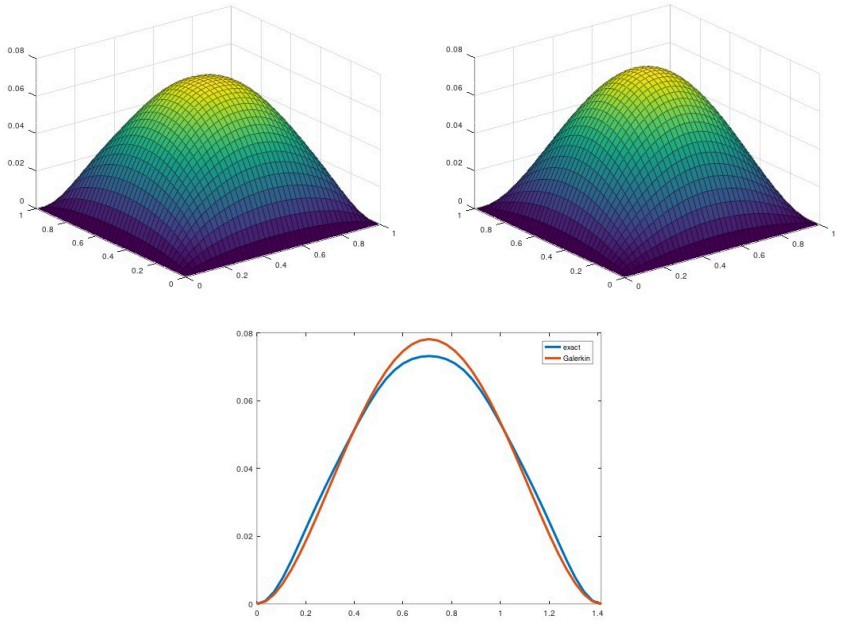


Figure 2.4 Exact solution, Galerkin solution, and comparison along the diagonal.

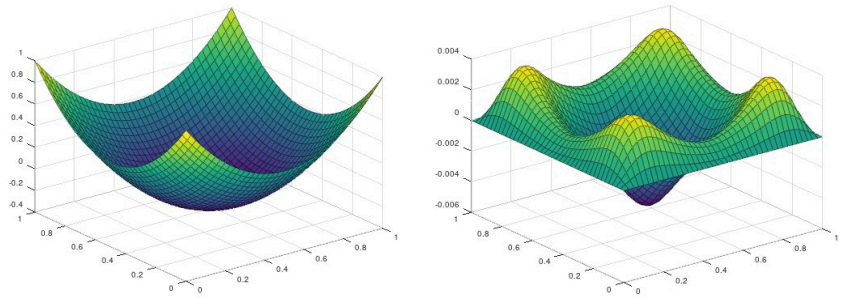


Figure 2.5 The residual function $r_h(x_1, x_2)$ (left) and the error function $e_h(x_1, x_2)$ (right).

2.4 Finite element spaces in two dimensions

Over the years many finite element spaces have been introduced with ever-increasing sophistication for different specific applications. For diffusion problems the classical and still most popular ones consist of **piecewise polynomial functions that are continuous in** Ω. These are the ones that we discuss next.

2.4.1 The simplest C^0 finite element space in two dimensions

How to define and build a space of piecewise polynomials that only consist of continuous functions? Consider the domain subdivided into element subdomains, over which the functions of the space need to be polynomials of degree r in two variables. If r is zero, the function is continuous if and only if its value is the same in all element subdomains, which makes the *piecewise* constant functions to be *globally* constant. So, $r = 0$ does not produce a space of continuous functions that can be used to approximate anything.

But what about piecewise *linear* polynomials ($r = 1$)? Or polynomials of higher degree? To consider the simplest case, lets answer this question for $r = 1$. The answer will introduce us to the **continuous** P_1 **finite element space**, also known simply as P_1 space.

Definition: Given a partition of Ω into element subdomains, the **continuous** P_1 **finite element space** associated to that partition is the space $\mathcal{W}_h$ of all functions that belong to $C^0(\Omega)$ and are a polynomial of degree $r = 1$ in each of the subdomains.

A polynomial $p(x_1, x_2)$ of degree less or equal than 1 in two dimensions has the form

$$p(x_1, x_2) = c_1 + c_2 x_1 + c_3 x_2$$

Therefore, if we know the value of p at three different non-colinear points in the plane, we can uniquely identify the coefficients c_1, c_2 and c_3.

Given any partition of Ω (just a decomposition into non-overlapping subdomains) it may be very hard to exhibit a basis of the P_1 space, but there are some special partitions, called **conformal triangulations** that make it very simple.

Definition: A **conformal triangulation** of a polygonal domain Ω is a decomposition of Ω into a finite number n_T of **triangles** such that the intersection of any two triangles K and K' is either (a) empty, or (b) a whole edge of **both** K and K', or (c) a vertex of **both** K and K'.

In Fig. 2.6 we can see several decompositions of a polygonal domain into non-overlapping triangles. The one labeled B is a conformal triangulation. The ones labeled A and C are not conformal.

Conformal triangulations are remarkable. Let $\mathcal{T}$ be a triangulation and let us number the vertices of the triangulation from 1 to n_V. Since every triangle

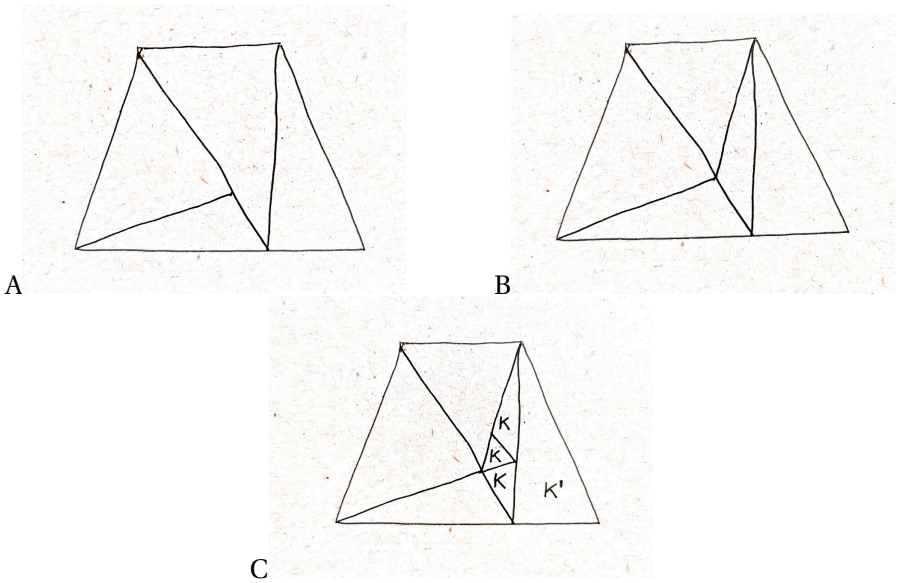


Figure 2.6 Examples of triangulations of a polygonal domain. B is conforming, A and C are not. The elements K in triangulation C that fail the definition of conforming triangulation when considered vis-à-vis K' are indicated.

has three vertices, by specifying the values $f_1, f_2, \ldots, f_{n_V}$ at all vertices of $\mathcal{T}$ one defines a *unique* $\mathbb{P}_1$-polynomial function f^K in each triangle K of $\mathcal{T}$. From these functions $\{f^K\}$, each defined in one triangle of $\mathcal{T}$, we can build a function f over Ω as

$$f(x) = f^K(x)\,, \quad \text{if } x \in K\,. \tag{2.23}$$

In other words, f takes at a point x the value corresponding to $f^K(x)$, if the point belongs to triangle K. All points belong to at least one triangle, so that $f(x)$ always exists. However, at some edges of a general triangulation the function may be multi-valued and thus f **may not be a continuous function**. This is easily understood by considering, in Fig. 2.6 (case A), the function f that is zero at all vertices except for the one and only interior vertex, at which the value is 1. This function, drawn in Fig. 2.7, is discontinuous along the edge over which the interior vertex is "hanging".

Definition: A **hanging vertex** in a triangulation $\mathcal{T}$ is a vertex x for which there exists a triangle K in $\mathcal{T}$ satisfying

$$x \in K, \qquad \text{and} \qquad x \text{ is not a vertex of } K\,.$$

An alternative definition of **conformal triangulations** is **triangulations with no hanging vertices**.

The remarkable fact about conformal triangulations is the following

Theorem 2.2. *Given arbitrary values $f_1, f_2, \ldots, f_{n_V}$ at the vertices, if the triangulation $\mathcal{T}$ is conforming then*

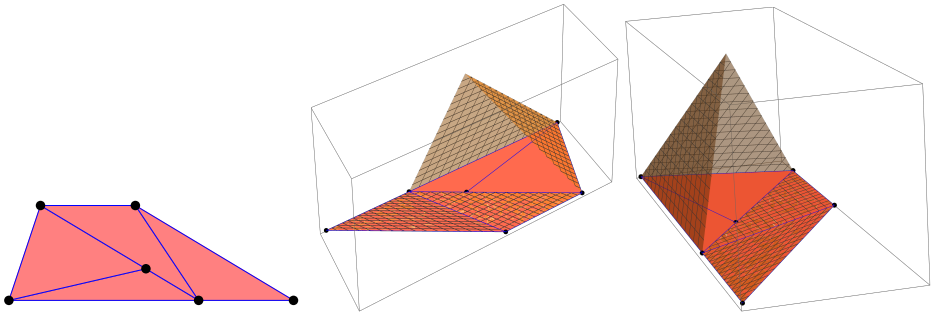


Figure 2.7 The non-conformal mesh in Fig. 2.6(Case A) is shown on the left, and two views of a piecewise linear polynomial function over it are displayed at the center and on the right. The function is obtained by setting the value of the interior vertex to 1, and to 0 at all the remaining vertices. Because the triangulation is not conformal, the resulting function is discontinuous along one of the interior edges.

- *the piecewise* $\mathbb{P}_1$ *function* f *defined by (2.23) is* **always continuous**, *and*
- *every function in the space* $\mathcal{W}_h$ *of all continuous piecewise* $\mathbb{P}_1$ *functions corresponds to a specific set of vertex values* $(f_1, f_2, \ldots, f_{n_V}) \in \mathbb{R}^{n_V}$.

As a consequence, the set of functions $\{N_a,\ a = 1, \ldots, n_V\}$ *which take the value 1 at vertex* j *and the value zero at all other vertices* **is a basis of** $\mathcal{W}_h$.

If the position of vertex i is $x_i = (x_{i1}, x_{i2})$, then the **Krönecker-delta property** holds, i.e.,

$$N_j(x_i) = \delta_{ij} \tag{2.24}$$

with $\delta_{ij} = 1$ if $i = j$ and $= 0$ otherwise.

Any function w_h in $\mathcal{W}_h$ is a linear combination of the basis functions. The coefficients of the linear combination, or **degrees of freedom**, are the values at the vertices, which by such property are called **nodes of the** P_1 **finite element space**. We thus have $n_V = m$ and

$$w_h(x) = w_1 N_1(x) + w_2 N_2(x) + \ldots + w_m N_m(x)\ , \tag{2.25}$$

where

$$w_1 = w_h(x_1), \quad w_2 = w_h(x_2), \qquad \ldots \qquad w_m = w_h(x_m)\ . \tag{2.26}$$

Once more we observe the linear correspondence between the **column array** of nodal values $\mathsf{W} = (w_1, w_2, \ldots, w_m)^T$ and the **function** $w_h(x)$.

How is a triangulation handled within the code? The typical way in which triangulations are handled is by means of two basic arrays:

- The **vertex coordinates array**, denoted here by X. It is a matrix of n_V columns, such that each column is the coordinate vector of a vertex.
- The **list of vertices**, denoted here by LV. It is a matrix of n_T columns and 3 rows. Each column contains the three numbers identifying the three vertices of the corresponding triangle.

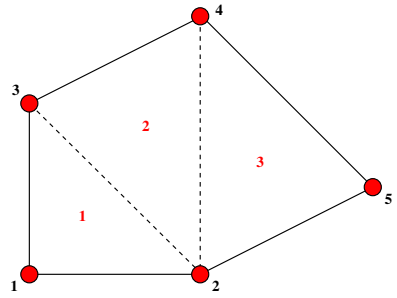


Figure 2.8 A simple conforming triangulation

Example 2.4 (A simple conforming triangulation) The triangulation of Fig. 2.8 has $n_V = 5$ vertices and $n_T = 3$ triangles and is conforming, as can easily be checked. With the specified numbering of triangles and vertices, the corresponding arrays are

$$\mathsf{X} = \begin{pmatrix} 4 & 8 & 4 & 8 & 12 \\ 2 & 2 & 6 & 8 & 4 \end{pmatrix} \tag{2.27}$$

$$\mathsf{LV} = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 2 & 2 \\ 3 & 4 & 5 \end{pmatrix} \tag{2.28}$$

The basis functions $N_1, \ldots, N_5$ of the P_1 space corresponding to this triangulation can be seen in Fig. 2.9.

2.4.2 Barycentric coordinates and the basis functions of the P_1 space

The **geometry** of a P_1 triangle K is determined by the positions its vertices $\mathbf{X}^1$, $\mathbf{X}^2$ and $\mathbf{X}^3$. It is the **only** triangle that has such vertices. It is also the **convex hull** of $\{\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3\}$, defined as the convex linear combinations (CLC) of the vertex positions:

$$K = \mathscr{C}(\{\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3\}) = \left\{ \mathbf{x} \in \mathbb{R}^d \,|\, \mathbf{x} = \sum_{j=1}^{d+1} \lambda_j \mathbf{X}^j, \underbrace{0 \le \lambda_j \le 1, \forall j, \text{ and } \sum_{j=1}^{d+1} \lambda_j = 1}_{\textbf{CLC}} \right\} \tag{2.29}$$

Remark: This definition of the geometry, in fact, works equally well in 2D ($d = 2$) and 3D ($d = 3$, in which case the triangle turns into a tetrahedron). It is independent of d. For each $\mathbf{x} \in K$ there exists a unique triplet $(\lambda_1, \lambda_2, \lambda_3) \in$ **CLC** such that $\mathbf{x} = \sum_{j=1}^{3} \lambda_j \mathbf{X}^j$, thus

$$\mathbf{x} \in K \quad \longleftrightarrow \quad (\lambda_1, \lambda_2, \lambda_3) \in \textbf{CLC}$$

is one-to-one, and thus a **reparameterization** (change of coordinates) of K. The parameters λ_i are called **area coordinates** of K (also **barycentric coordinates**).

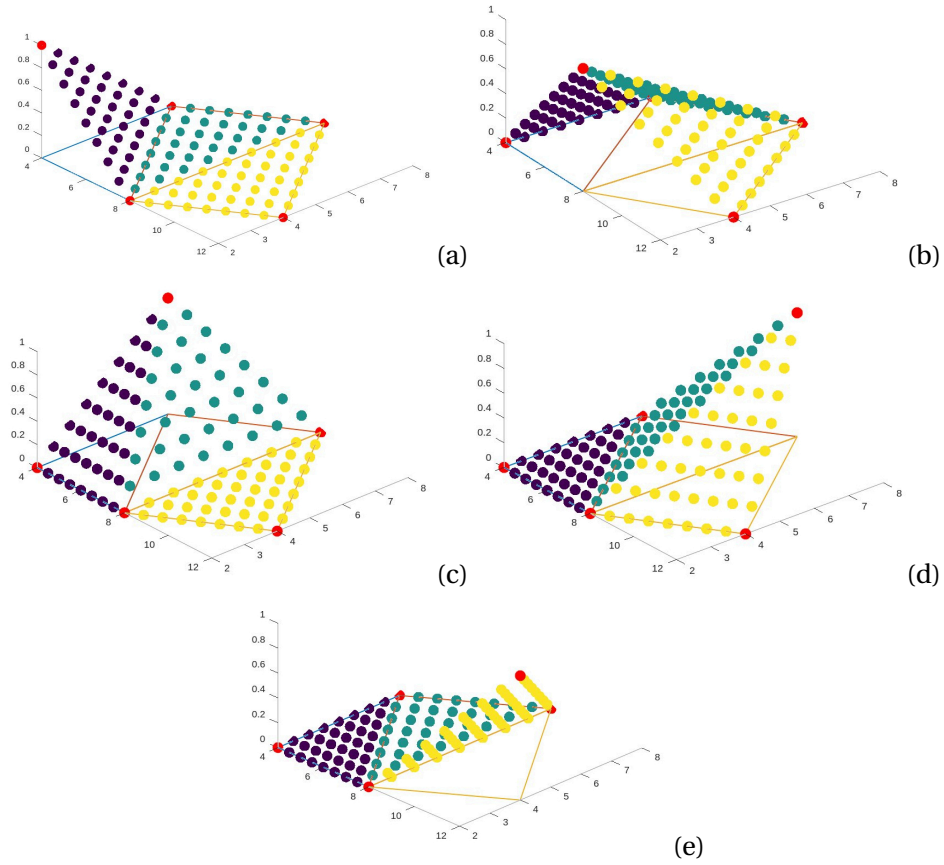


Figure 2.9 Basis functions of the triangulation of Example 2.4. (a) N_1, (b) N_2, …, (e) N_5. The elevation corresponds to the value of the function. The nodal values are shown as red dots. Sample points belonging to triangle 1, 2 or 3 are shown in purple, green or yellow, respectively.

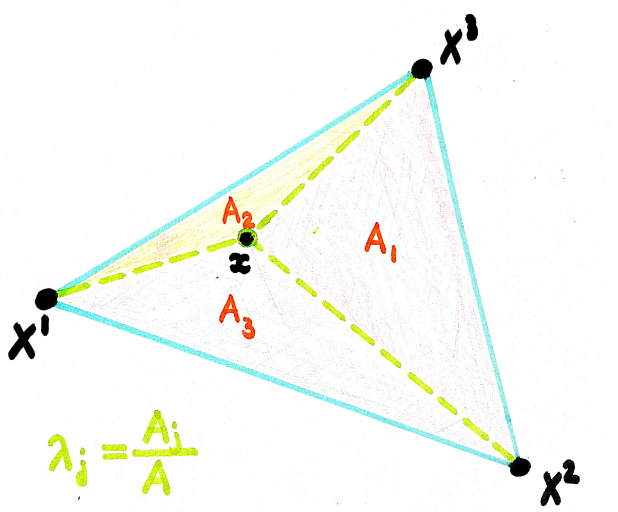


Figure 2.10 Barycentric coordinates

Barycentric coordinates have the following properties:

a) $\mathbf{x} = \mathbf{X}^j \iff \lambda_j = 1, \lambda_{k\neq j} = 0$. This is analogous to the Krönecker delta property.

b) $\mathbf{x}$ belongs to edge $\mathbf{X}^i\mathbf{X}^j$ iff $\lambda_i + \lambda_j = 1$, and as a result, $\lambda_{k\notin\{i,j\}} = 0$. In other words, λ_j is zero along the edge opposite to vertex $\mathbf{X}^j$.

c) $\mathbf{x}$ belongs to the (relative) interior of K iff all λ_j's are different from 0 and 1.

d) The barycentric coordinates satisfy that

$$\lambda_i = \frac{A_i}{A},$$

where A is the area of the triangle K and A_i is the area of the triangle formed by $\mathbf{x}$ and the two vertices $\mathbf{X}^j$ with $j \neq i$, see Fig. 2.10.

e) The inverse mapping to $(\lambda_1, \lambda_2, \lambda_3) \mapsto \mathbf{x} = \sum_{j=1}^3 \lambda_j \mathbf{X}^j$ is given by (in 2D, with $\mathbf{x} = (x_1, x_2)^T$)

$$\lambda_1(x_1, x_2) = \frac{1}{2A}\left[-(X_2^3 - X_2^2)(x_1 - X_1^2) + (X_1^3 - X_1^2)(x_2 - X_2^2)\right] \quad (2.30)$$

$$\lambda_2(x_1, x_2) = \frac{1}{2A}\left[-(X_2^1 - X_2^3)(x_1 - X_1^3) + (X_1^1 - X_1^3)(x_2 - X_2^3)\right] \quad (2.31)$$

$$\lambda_3(x_1, x_2) = \frac{1}{2A}\left[-(X_2^2 - X_2^1)(x_1 - X_1^1) + (X_1^2 - X_1^1)(x_2 - X_2^1)\right] \quad (2.32)$$

where $2A$ is twice the area of K,

$$2A = (X_1^2 - X_1^1)(X_1^3 - X_1^1) - (X_2^2 - X_2^1)(X_2^3 - X_2^1)\,.$$

It is a general convention that the *vertices are ordered either clockwise or counter-clockwise*. We are adopting the latter. Otherwise, A would be negative the area of K, but the other formulae would remain true.

f) The **barycenter B** of K corresponds to

$$\mathbf{B} = \frac{\mathbf{X}^1 + \mathbf{X}^2 + \mathbf{X}^3}{3} \leftrightarrow (\lambda_1, \lambda_2, \lambda_3) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$$

g) The **edges midpoints** correspond to $(\lambda_1, \lambda_2, \lambda_3)$ equal to $(0, 1/2, 1/2)$, $(1/2, 0, 1/2)$ and $(1/2, 1/2, 0)$. Notice that the midpoints have been numbered according to the opposite vertex.

The barycentric coordinates are not just another set of coordinates (instead of $x_1 - x_2$) that one could choose to parameterize the points of a triangle, for **triangular finite elements**,

$$N_1^e = \lambda_1, \quad N_2^e = \lambda_2 \quad \text{and} \quad N_3^e = \lambda_3, \tag{2.33}$$

given by (2.30)-(2.32) above, are **the local basis of the P_1-finite element space** restricted to element e. In fact, they are **three polynomials of degree 1 and linearly independent**, and thus a basis of $\mathbb{P}_1$. Since they satisfy the delta property at the vertices, the **vertices are the nodes of this space**.

The gradient of the basis functions can be obtained by differentiation of (2.30)-(2.32) with respect to x_1 and x_2, which gives

$$\nabla N_1^e = \frac{1}{2A} \begin{pmatrix} X_2^2 - X_2^3 \\ X_1^3 - X_1^2 \end{pmatrix}, \tag{2.34}$$

$$\nabla N_2^e = \frac{1}{2A} \begin{pmatrix} X_2^3 - X_2^1 \\ X_1^1 - X_1^3 \end{pmatrix}, \tag{2.35}$$

$$\nabla N_3^e = \frac{1}{2A} \begin{pmatrix} X_2^1 - X_2^2 \\ X_1^2 - X_1^1 \end{pmatrix}. \tag{2.36}$$

The second derivatives are of course zero all over the element.

Now, **notice what happens if we take the local-to-global array equal to the list-of-vertices array**,

$$\mathsf{LG} = \mathsf{LV}\,. \tag{2.37}$$

This means that we consider the P_1 element with **the vertices as nodes and the triangles as element domains**.

Following exactly the same methodology that was developed for the 1D case, the global basis functions are defined as

$$N_A(x_1, x_2) = \sum_{(a,e)|\mathsf{LG}(a,e)=A}^{\circ} N_a^e(x_1, x_2)\,. \tag{2.38}$$

Remember, the summation is only performed when $x = (x_1, x_2)$ is **interior** to some triangle in the mesh. The value of N_A at the mesh vertices and edges is not equal to the sum above, but as the continuous extension (if it exists) from the element interiors. Let us see how this works in the triangulation of Figure 2.8.

Example 2.5 (Using the list of vertices as local-to-global map) Going back to the triangulation in Example 2.4, with LG equal to LV given in (2.28), i.e.,

$$\mathsf{LG} = \mathsf{LV} = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 2 & 2 \\ 3 & 4 & 5 \end{pmatrix}, \tag{2.39}$$

the explicit expressions for the global basis functions (corresponding to (2.38)) are

$$N_1 = N_1^1 \tag{2.40}$$
$$N_2 = N_2^1 \mathring{+} N_2^2 \mathring{+} N_2^3 \tag{2.41}$$
$$N_3 = N_3^1 \mathring{+} N_1^2 \tag{2.42}$$
$$N_4 = N_3^2 \mathring{+} N_1^3 \tag{2.43}$$
$$N_5 = N_3^3 \tag{2.44}$$

Take for example N_3, which is depicted in Fig. 2.11. We know that N_3 is different from zero just in elements 1 and 2 because in LG the number 3 only appears in columns 1 and 2. Inside element $e = 1$ the function N_3 coincides with the N_3^1, depicted in purple in the figure, because 3 appears in row 3 of column $e = 1$ of LG. Similarly, in element $e = 2$ the function N_3 coincides with N_1^2, depicted in green in the figure, because 3 appears in row 1 of column $e = 2$ of LG. In column $e = 3$ of LG the number 3 does not appear. This means that the function N_3 is identically zero in element $e = 3$, as shown in yellow in Fig. 2.11. By the magic of conforming triangulations, the three pieces fit together in such a way that the resulting function N_3 is a continuous function. **In fact, the functions N_1-N_5 defined by (2.40)-(2.40) are exactly the same as those defined in Theorem 2.2 and depicted in Figure 2.9.**

The fundamental message is that, if the arrays X and LV of a conforming triangulation of a domain Ω are available, then the global basis functions obtained from (2.38) by taking LG = LV generate the space $\mathscr{W}_h$ of P_1 continuous finite elements.

In what follows, we proceed and use the basis functions just developed to solve a 2D diffusion problem in any polygonal domain by the Galerkin method.

2.4.3 The element stiffness matrix

Remember that our goal is to solve the Diffusion Problem 2.2 in weak form by the Galerkin method, so as to obtain the only function $u_h \in \mathscr{S}_h$ that satisfies

$$\int_\Omega (K\nabla u_h)\cdot\nabla v_h \, d\Omega = \int_\Omega f\, v_h \, d\Omega + \int_{\partial\Omega_N} H\, v_h \, d\Gamma \tag{2.45}$$

for all $v_h \in V_h$.

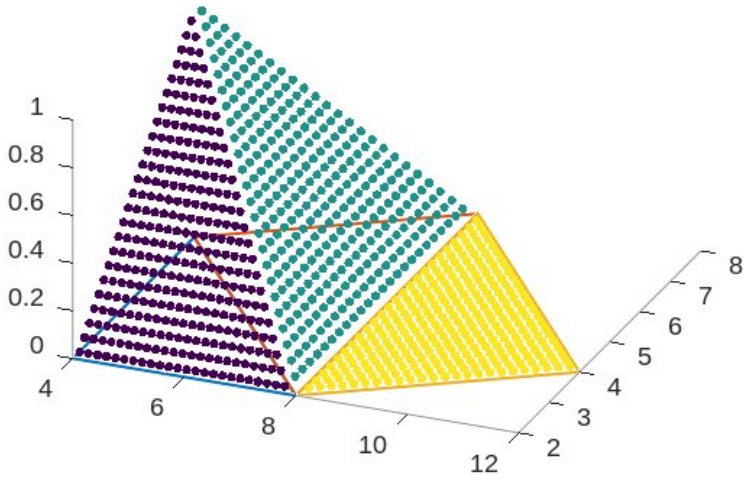


Figure 2.11 The basis function N_3 sampled only at interior points. Sample points belonging to triangle 1, 2 or 3 are shown in purple, green or yellow, respectively.

The element stiffness matrix of the diffusion problem for the P_1 triangular element is as follows. Let the indices a and b run from 1 to 3, the number of nodes per element. The three local basis functions at element e are barycentric coordinates, as introduced in (2.33). As a consequence, the element matrix results from

$$\mathsf{K}^e_{ab} = \int_{\Omega_e} (K\nabla N^e_b)\cdot\nabla N^e_a \, d\Omega \,. \tag{2.46}$$

Let us assume for simplicity that K is isotropic, so that $K(x) = k(x)\mathbb{I}$. Further, notice that ∇N^e_b is constant over Ω_e for any b, so that

$$\mathsf{K}^e_{ab} = \left(\int_{\Omega_e} k(x)\, d\Omega\right) \nabla N^e_b \cdot \nabla N^e_a = k_e\, A_e\, \nabla N^e_b \cdot \nabla N^e_a \,.$$

Notice that k_e is the average diffusion coefficient in the element, since A_e is the element's area.

Consider, for each element e, the array dN which contains the partial derivatives of the basis functions and is defined by

$$\mathsf{dN}_{ib} = \frac{\partial N^e_b}{\partial x_i} \,,$$

that is, one column per basis function, with $i = 1,2$. Then K^e is given by

$$\mathsf{K}^e = k_e\, A_e\, \mathsf{dN}^T\, \mathsf{dN} \,.$$

In Octave/MATLAB code, given the array of nodal coordinates of element e,

$$\mathsf{xe} = \begin{pmatrix} X^1_1 & X^2_1 & X^3_1 \\ X^1_2 & X^2_2 & X^3_2 \end{pmatrix}$$

☞ The matrix dN has the form

$$\mathsf{dN} = \begin{bmatrix} \frac{\partial N^e_1}{\partial x_1} & \frac{\partial N^e_2}{\partial x_1} & \frac{\partial N^e_3}{\partial x_1} \\ \frac{\partial N^e_1}{\partial x_2} & \frac{\partial N^e_2}{\partial x_2} & \frac{\partial N^e_3}{\partial x_2} \end{bmatrix}.$$

in which each column corresponds to one of the three nodes of the element, and given k_e, the computation of K^e is as simple as

```
dN=[xe(2,2)-xe(2,3),xe(2,3)-xe(2,1),xe(2,1)-xe(2,2);...
      xe(1,3)-xe(1,2),xe(1,1)-xe(1,3),xe(1,2)-xe(1,1)];
Ae2=dN(2,3)*dN(1,2)-dN(1,3)*dN(2,2);
dN=dN/Ae2;
Ke=Ae2/2*ke*dN'*dN;
```

2.4.4 The element load vector

Let us for now assume that either Dirichlet conditions are imposed all over the boundary or the Neumann datum H is zero. Then the element load vector is

$$\mathsf{F}^e_a = \int_{\Omega_e} f\, N^e_a \, d\Omega \,.$$

The result of this integral depends of course on the function $f(x)$. Let us assume that $f = f_e$ is constant over the element, then $f N^e_a$ is a polynomial of degree 1 in x_1 and x_2 which has as integral

$$\mathsf{F}^e_a = f_e \int_{\Omega_e} N^e_a \, d\Omega = \frac{f_e A_e}{3}, \qquad \forall\, a = 1,2,3. \tag{2.47}$$

The computation of the element load vector is thus immediate

```
Fe=Ae2*fe*ones(3,1)/6;
```

Another possibility is to assume that f is affine within the element, in which case the three nodal values need to be provided. Whatever the values are, it is clear that the product $f\, N^e_a$ is a polynomial of degree 2 in the variables x_1 and x_2.

To compute F^e we can use the following well-known integration rule: *The integral of a quadratic polynomial over a triangle is equal to $\frac{1}{3}$ times the triangle's area times the sum of the values of the polynomial at the three edge midpoints.*

Let $f_e = (f_{e,1}, f_{e,2}, f_{e,3})$ be an array containing the three nodal values of f_e. Then the integration rule above states that

$$\begin{aligned} \mathsf{F}^e_a &= \int_{\Omega_e} f\, N^e_a \, d\Omega \\ &= \frac{A_e}{3}\left(\frac{f_{e,1}+f_{e,2}}{2}\,\frac{\delta_{a1}+\delta_{a2}}{2} + \frac{f_{e,2}+f_{e,3}}{2}\,\frac{\delta_{a2}+\delta_{a3}}{2} + \frac{f_{e,3}+f_{e,1}}{2}\,\frac{\delta_{a3}+\delta_{a1}}{2} \right) \end{aligned}$$

Particularizing for $a = 1,2,3$ we obtain the element load vector

$$\mathsf{F}^e = \frac{A_e}{12}\begin{pmatrix} 2f_{e,1}+f_{e,2}+f_{e,3} \\ f_{e,1}+2f_{e,2}+f_{e,3} \\ f_{e,1}+f_{e,2}+2f_{e,3} \end{pmatrix} \tag{2.48}$$

which would lead to the code (notice that `fe` is stored as a row array)

```
auxmat=[2,1,1;1,2,1;1,1,2];
Fe=(Ae2/24)*auxmat*fe';
```

2.4.5 Solving 2D diffusion problems with P_1 finite elements (Dirichlet case)

Let us work out the procedure and code that solves the diffusion problem with P_1 finite elements, assuming that we only have Dirichlet boundary conditions. We assume, as before, that a mesh is provided by means of a **list of coordinates** X and a **list of vertices** LV.

Further, let us assume that a **list of boundary nodes**, denoted by η_g is provided, together with a **list of boundary values**, GG.

Also, aproximate k and f as piecewise constant, with value k_e and f_e in each triangle.

Then we can build a function that computes the elementary stiffness matrix and load vector, for example,

```
function [Ke, Fe]=elementKandF(xe,ke,fe)
  dN=[xe(2,2)-xe(2,3),xe(2,3)-xe(2,1),xe(2,1)-xe(2,2);...
      xe(1,3)-xe(1,2),xe(1,1)-xe(1,3),xe(1,2)-xe(1,1)];
  Ae2=dN(2,3)*dN(1,2)-dN(1,3)*dN(2,2);
  dN=dN/Ae2;
  Ke=Ae2/2*ke*dN'*dN;
  Fe=Ae2*fe*ones(3,1)/6;
end
```

It only remains to **assemble** the element contributions and impose the Dirichlet boundary conditions to end up with the global stiffness matrix and load vector.

Here we will use a trick that simplifies the coding: Instead of taking care of the boundary conditions during the assembly procedure, we will *assemble the global matrix and load vector as if there were no boundary conditions, and then correct the lines that correspond to unknowns with imposed value.*

If this trick is adopted, the assembly procedure of finite element stiffness matrices and load vectors is quite intuitive and straightforward to code. In Octave/MATLAB it reads

```
LG=LV; nod=size(X,2); nel=size(LG,2);
K=zeros(nod,nod); F=zeros(nod,1);
for iel=1:nel
%% setting the local data
  lge=LG(:,iel);
  xe(1:dd,1:npe)=X(1:dd,lge(1:npe));
  ke=difcoeff(iel);
  fe=source(iel);
%% computing element K and F
  [Ke Fe]=elementKandF(xe,ke,fe);
%% assembly, from local to global
  K(lge,lge)+=Ke;
  F(lge)+=Fe;
end
```

where the arrays `difcoeff` and `source` contain the element-wise values of k and f, respectively.

Notice that `lge` is an index array that contains the numbering of the three nodes of the element. In this way, the matrix `K(lge,lge)` is the submatrix (or "slice") of `K` consisting of just the rows and columns present in `lge`. The key

assembly operations

```
K(lge,lge)+=Ke;
F(lge)+=Fe;
```

are thus just a shorter way of coding

```
for j=1:3
   for k=1:3
      K(lge(j),lge(k))+=Ke(j,k);
   end
   F(lge(j))+=Fe(j);
end
```

which is the lengthier code we introduced originally. For example, if `lge=[7 4 9]`, then the operation

```
K(lge,lge)+=Ke;
```

is equivalent to

```
K(7,7)+=Ke(1,1); K(7,4)+=Ke(1,2); K(7,9)+=Ke(1,3);
K(4,7)+=Ke(2,1); K(4,4)+=Ke(2,2); K(4,9)+=Ke(2,3);
K(9,7)+=Ke(3,1); K(9,4)+=Ke(3,2); K(9,9)+=Ke(3,3);
```

It only remains to fix the lines corresponding to the boundary nodes and we will be in a position to solve our first two-dimensional problem with finite elements! Remember what we need to do: If node A is a Dirichlet node with imposed value g, we must modify the linear system (i.e., the arrays K and F) so that the A-th equation reads, simply, $\mathsf{U}_A = g$. This is easily coded as follows:

```
ng=length(EtaG); II=eye(nod);
for ig=1:ng
 K(EtaG(ig),:)=II(EtaG(ig),:);
 F(EtaG(ig))=GG(ig);
end
```

The code is now complete, we can solve for the unknown vector U which contains the nodal values of u_h and plot the solution.

```
U=K\F;
trisurf(LG',X(1,:),X(2,:),U)
```

The command `trisurf` plots functions defined on arbitrary conforming triangulations.

Example 2.6 A uniformly heated rod of arbitrary polygonal shape.

Consider the geometry shown in Fig. 2.8, only that now we will work with the much finer discretization shown in Fig. 2.12. Let us assume that the rod is homogeneous, with diffusion coefficient equal to 1 and heat source f equal to 10, and that the surface temperature is $g = 20$. We aim to compute the temperature distribution inside the rod as given by the Galerkin solution.

The whole code needed for this purpose is provided as `octavefemp1a.m` in the accompanying material and reads as shown in Table 2.1. After running

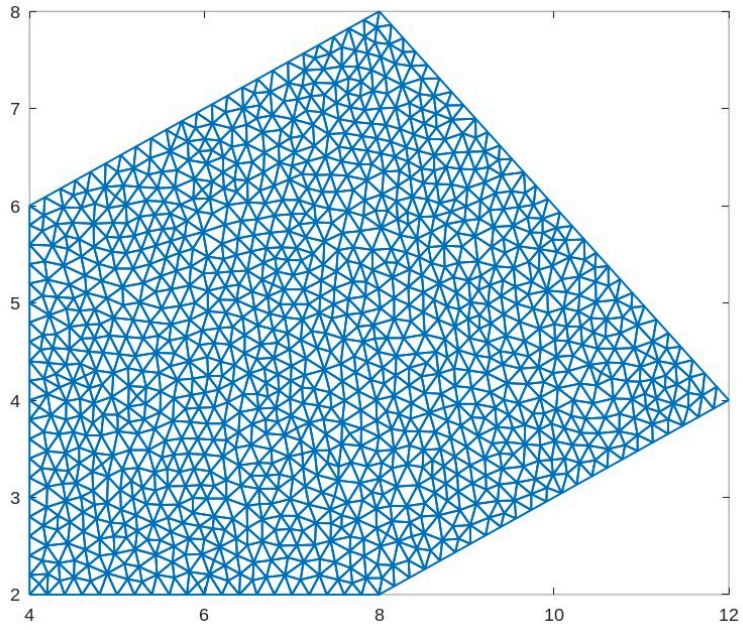


Figure 2.12 Refined triangulation of a polygonal domain.

it, we have computed our first 2D finite element solution! The function u_h can be seen in Figure 2.13. It provides a good estimation of the exact temperature distribution, since the mesh is quite refined. It can be used to estimate, for example, the maximum temperature in the rod, which yields

$$\max_{x\in\Omega} u_h(x) = 43.077\,.$$

Example 2.7 (The uniformly heated square rod revisited with P_1 elements) By simply changing the geometry of the previous example we can revisit the square-rod problem discussed in Example 2.3.

The results obtained with different meshes are plotted in Figure 2.14. The maximum of u_h is 20.719, 20.733 and 20.737, respectively, for the meshes in part (a), (b) and (c) of the Figure. The exact maximum is 20.737. The corresponding number of elements is 68, 242 and 1054, and the number of nodes is 45, 142 and 568. The maximum absolute value of the error $u - u_h$ for each mesh is 0.0193, 0.00557 and 0.00134. We observe that the Galerkin solution remains stable as the mesh is refined, and converges at all points of the domain to the exact solution.

For comparison purposes we report also the Galerkin results obtained, for this specific case, with the space $\mathcal{W}_h$ of quartic polynomials used in Example 2.3. The maximum of the discrete solution is 20.781 and the maximum error is 0.0445.

```
function [Ke Fe]=elementKandF(xe,ke,fe)
  dN=[xe(2,2)-xe(2,3),xe(2,3)-xe(2,1),xe(2,1)-xe(2,2);...
      xe(1,3)-xe(1,2),xe(1,1)-xe(1,3),xe(1,2)-xe(1,1)];
  Ae2=dN(2,3)*dN(1,2)-dN(1,3)*dN(2,2);
  dN=dN/Ae2;
  Ke=Ae2/2*ke*dN'*dN;
  Fe=Ae2*fe*ones(3,1)/6;
end
%% finite element solver begins (X, LV, EtaG and GG are given)
LG=LV; nod=size(X,2); nel=size(LG,2);
difcoeff=ones(1,nel);
source=10*ones(1,nel);
GG=20*ones(1,length(EtaG));
K=zeros(nod,nod); F=zeros(nod,1);
for iel=1:nel
%% setting the local data
  lge=LG(:,iel);
  xe(1:dd,1:npe)=X(1:dd,lge(1:npe));
  ke=difcoeff(iel);
  fe=source(iel);
%% computing element K and F
  [Ke Fe]=elementKandF(xe,ke,fe);
%% assembly, from local to global
  K(lge,lge)+=Ke;
  F(lge)+=Fe;
end
%% boundary nodes
ng=length(EtaG); II=eye(nod);
for ig=1:ng
 K(EtaG(ig),:)=II(EtaG(ig),:);
 F(EtaG(ig))=GG(ig);
end
%% solve algebraic system
U=K\F;
%% plot
trisurf(LG',X(1,:),X(2,:),U)
```

Table 2.1 Code **octavefemp1a.m**. It solves Example 2.6.

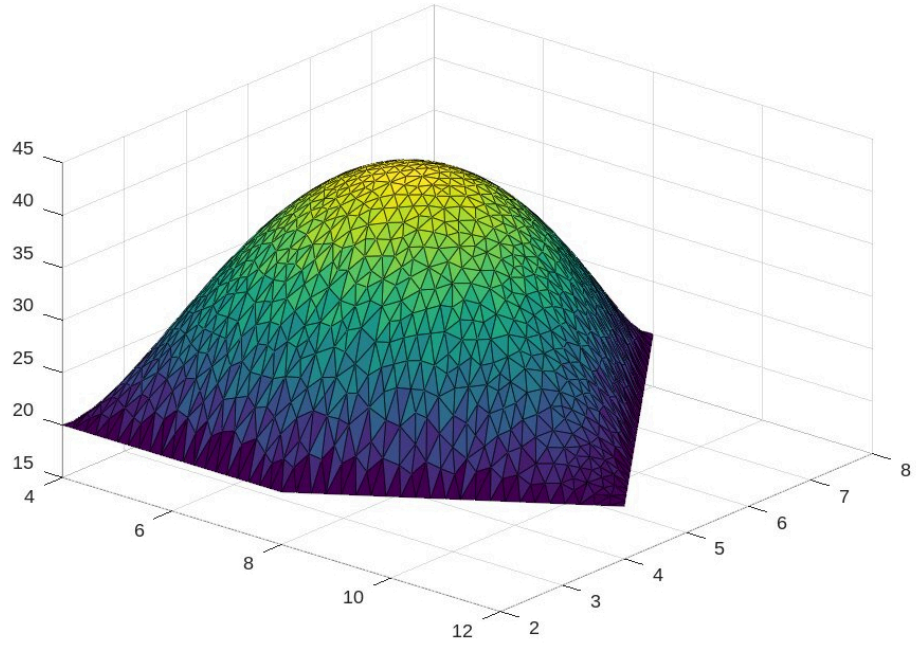


Figure 2.13 Galerkin solution u_h of Example 2.6.

2.4.6 Solving problems with Neumann boundaries

We have presented a method that approximates the solution of the diffusion equation in any 2D polygonal domain with P_1 finite elements, but just when the solution value is known at the whole boundary (i.e., when the Neumann boundary $\partial\Omega_N$ is empty).

Frequently, however, we have information of the value of the normal flux $H = k\nabla u \cdot \check{n}$ on some parts of the boundary (where $\check{n}$ is the outward normal to $\partial\Omega$). In such cases, the solution is specified on the Dirichlet boundary as before, so that

$$\mathcal{S}_h = \{w_h \in \mathcal{W}_h \,|\, w_h = g \,\text{on}\, \partial\Omega_D\} \quad \text{and} \quad \mathcal{V}_h = \{w_h \in \mathcal{W}_h \,|\, w_h = 0 \,\text{on}\, \partial\Omega_D\}.$$

There remains the part $\partial\Omega_N$, where there exist basis functions of $\mathcal{V}_h$ that are not identically zero.

Assuming that node A belongs to $\partial\Omega_N$, the A-th component of the load vector involves two terms,

$$\mathsf{F}_A = \int_\Omega f\, N_A \, d\Omega + \int_{\partial\Omega_N} H\, N_A \, d\Gamma \,,$$

while the corresponding line of the stiffness matrix remains the same.

Symmetry lines and homogeneous Neumann conditions. If the domain and data of the problem are symmetric with respect to some line that crosses the domain, then the solution will also be. This is a property of all *linear* problems. Then

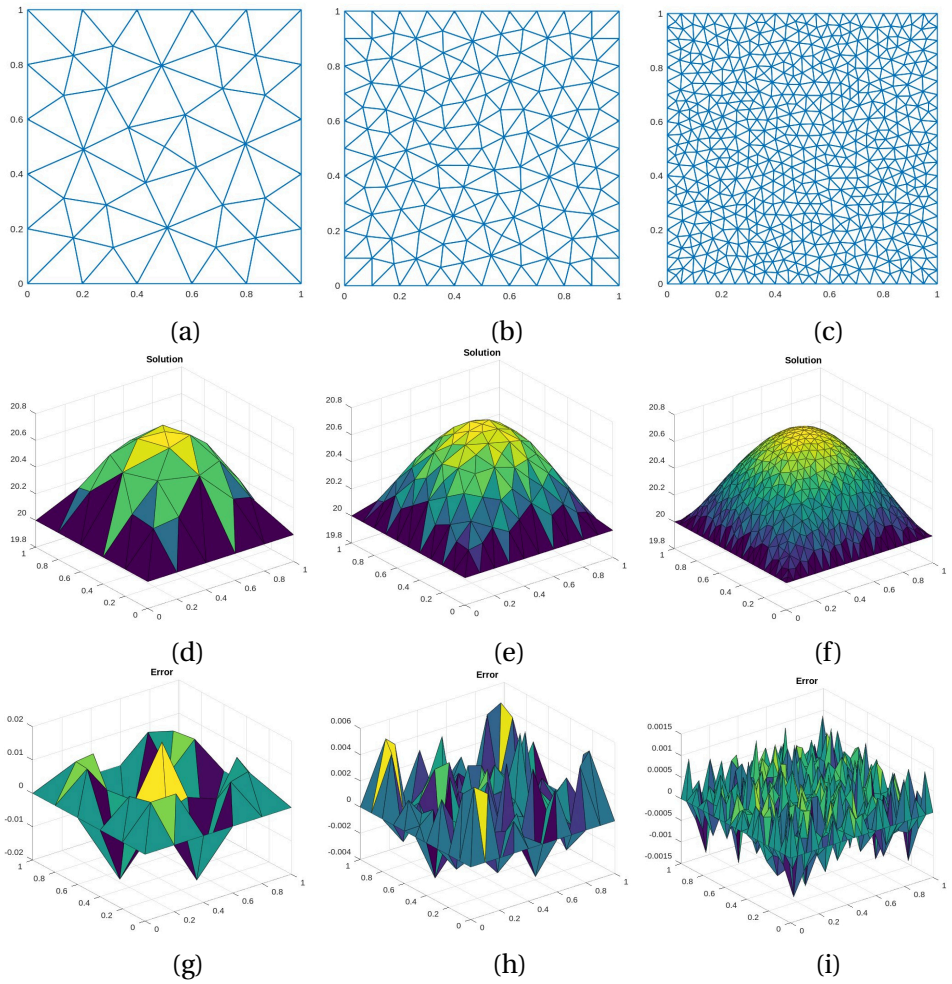


Figure 2.14 The P_1 finite element solutions of the uniformly heated square-rod example. Each column corresponds to a different mesh, which is plotted at the top of the column. The second row shows the Galerkin solution u_h and the third row shows the error $u - u_h$.

one can solve the corresponding equation on a *reduced domain*, consisting of just one half of the original one, and extend the solution found in the reduced domain to the original one by symmetry. The symmetry line is thus a part of the boundary of the reduced domain. What boundary condition should be imposed there? It can be shown that the correct boundary condition is a *Neumann condition* with $H = 0$, because the normal derivative of u must be zero there. These are called **homogeneous Neumann conditions**, and also apply at any isolated (or zero-flux) boundary.

If all parts of the boundary that do not have Dirichlet conditions have homogeneous Neumann conditions, then the code in Table 2.1 works perfectly well as is. It was built considering that the Dirichlet boundary occupies the whole of $\partial\Omega$ and thus the load vector F lacks the contribution

$$\int_{\partial\Omega_N} H N_i \, d\Gamma \,,$$

but, if $H = 0$ all over $\partial\Omega_N$, **there is nothing to be added**. The Galerkin finite element method automatically imposes homogeneous Neumann conditions all over the part of the boundary where the solution value is not imposed. This is because homogeneous Neumann conditions are sometimes called "do-nothing boundary conditions" for this problem.

Example 2.8 (Exploiting symmetries) The case of the uniformly heated square rod exhibits several symmetries. The central horizontal and vertical lines are lines of symmetry, as well as both diagonals. By exploiting those symmetries, we end up with the reduced domain defined as the triangle ABC shown in Fig. 2.15(a). Along the edge BC the Dirichlet condition $u = g$ is applied. The other two edges (AB and CA) must satisfy **homogeneous Neumann boundary conditions** for the solutions in the original and reduced domains to coincide. This is quite useful, since meshing the reduced domain requires significantly less elements of any given size than those required by the original domain (approximately 1/8).

We modified the previous code to set the triangle ABC as domain and only set the nodes in the edge BC as Dirichlet nodes, with imposed value $g = 20$. Nothing special was programmed for the other boundary nodes, they were treated just like internal nodes. The other constants are as in Example 2.7, i.e., $k = 1$ and $f = 10$. The result obtained with a mesh of 380 elements and 220 nodes is shown in Fig. 2.15(b). The maximum error of the discrete solution is 0.00064, smaller than the error attained in the whole domain with a mesh of 1054 elements and 568 nodes.

Non-homogeneous Neumann boundary conditions. When $H \neq 0$ one needs to build the complete load vector, which, as said, if A is a non-Dirichlet node reads

$$\mathsf{F}_A = \int_\Omega f N_A \, d\Omega + \int_{\partial\Omega_N} H N_A \, d\Gamma \,.$$

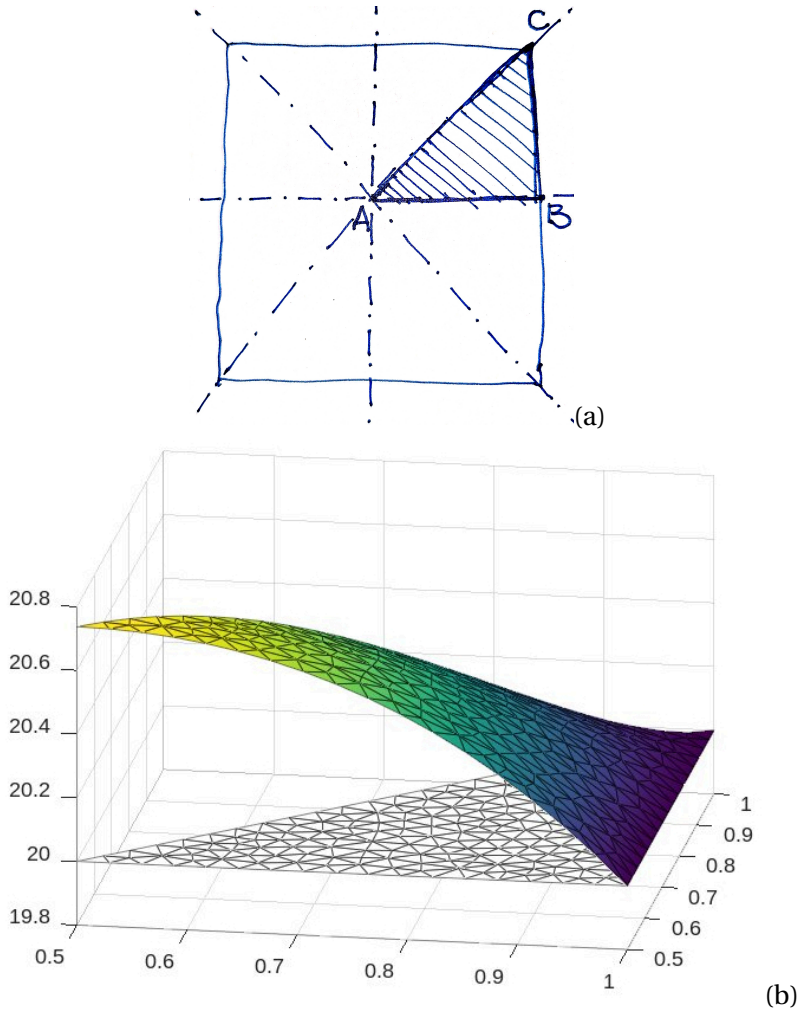


Figure 2.15 (a) Original domain of the square rod problem, and reduced domain (the triangle ABC after exploiting the several symmetries. (b) Galerkin solution in the reduced domain, with homogeneous Neumann conditions at symmetry boundaries.

The second integral is our focus of attention now. The differential $d\Gamma$ is a differential of length, because $\partial\Omega$ is one-dimensional. Our first task is to describe the **data structure** with which we handle the definition of $\partial\Omega_N$ and of the function H inside the code.

The **mesh generator** builds the domain boundary by joining together ℓ individual lines provided by the user, of which some (or all) belong to the Neumann boundary. Let us assume that the produced triangulation has n_{be} boundary edges. By construction, each boundary edge belongs to one and only one of these lines. The generator provides an **array of boundary edges** BE of dimension $2 \times n_{\text{be}}$. Each column of BE contains three numbers. The first two indicate the two nodes of the corresponding edge. The third one indicates the line to which the edge belongs.

To keep things simple, we can assume that H is constant over each edge. Then it can be provided by an array H of n_{be} entries, as shown in the example below.

Example 2.9 (Boundary arrays of a triangulation) In Figure 2.16 we show a small triangulation of a polygon, defined by points 1-5 and lines 1-5. For this triangulation $n_{\text{be}} = 11$, and BE is as follows:

$$\mathsf{BE} = \begin{pmatrix} 1 & 6 & 2 & 7 & 5 & 8 & 9 & 4 & 10 & 3 & 11 \\ 6 & 2 & 7 & 5 & 8 & 9 & 4 & 10 & 3 & 11 & 1 \\ 1 & 1 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 5 & 5 \end{pmatrix}$$

Now suppose that along **line 2** the imposed heat flux H is constant and equal to $\frac{2}{3}$, while along **line 3** it is also constant but equal to $\frac{1}{2}$. Along lines 1, 4 and 5 the Dirichlet condition $u = 20$ is imposed. This is specified in the array H that provides the value of H over each edge, assuming it is constant at each edge.

$$\mathsf{H} = \begin{pmatrix} 0 & 0 & \frac{2}{3} & \frac{2}{3} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \end{pmatrix}$$

The construction of H is easily coded as

```
for k=1:nbe
   if (BE(3,k)==2)
      HH(k)=2./3;
   end
   if (BE(3,k)==3)
      HH(k)=1./2;
   end
end
```

It should be clear by now that a P_1 function restricted to an edge of the triangulation is *affine*. The function interpolates linearly along the edge between the two nodal values at its ends. This implies that, if the k-th boundary edge (let

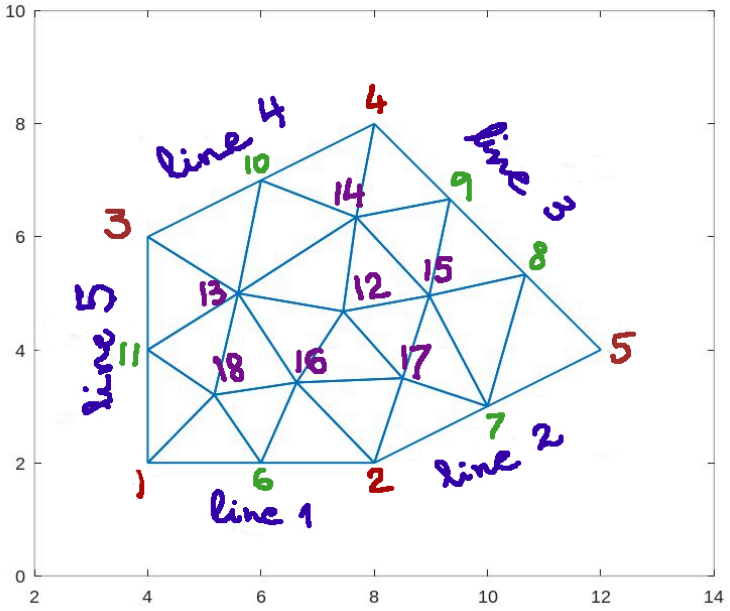


Figure 2.16 A small triangulation, showing the numbering of the defining points (in red), of the defining lines (in blue), and of the rest of the vertices.

us denote it by E_k) joins the nodes $A_1 = \mathsf{BE}(1,k)$ and $A_2 = \mathsf{BE}(2,k)$, then the only two basis functions that are different from zero along the edge are N_{A_1} and N_{A_2}. Further, these two basis functions go affinely from 0 to 1 over the edge, so that their average value is $\frac{1}{2}$. We can thus compute the integral over E_k as

$$\int_{E_k} H\, N_{A_1}\, d\Gamma = \int_{E_k} H\, N_{A_2}\, d\Gamma = \frac{1}{2}\,\mathsf{H}(k)\,|E_k|\,,$$

where $|E_k|$ is the length of the edge,

$$|E_k| = \|X^{A_1} - X^{A_2}\|\,.$$

We can implement the addition of the Neumann part of the load vector as an assembly procedure.

```
%% Neumann contributions
for ied=1:nbe
  lged=BE(1:2,ied);
  xed(1:dd,1:2)=X(1:dd,lged);
  Led=norm(xed(:,1)-xed(:,2));
  Hed=HH(ied);
  F(lged)+=Hed*Led/2;
end
```

Remark: Of course, the contribution of each edge will depend on the specific function H. Piecewise polynomial functions of higher degree can easily be implemented.

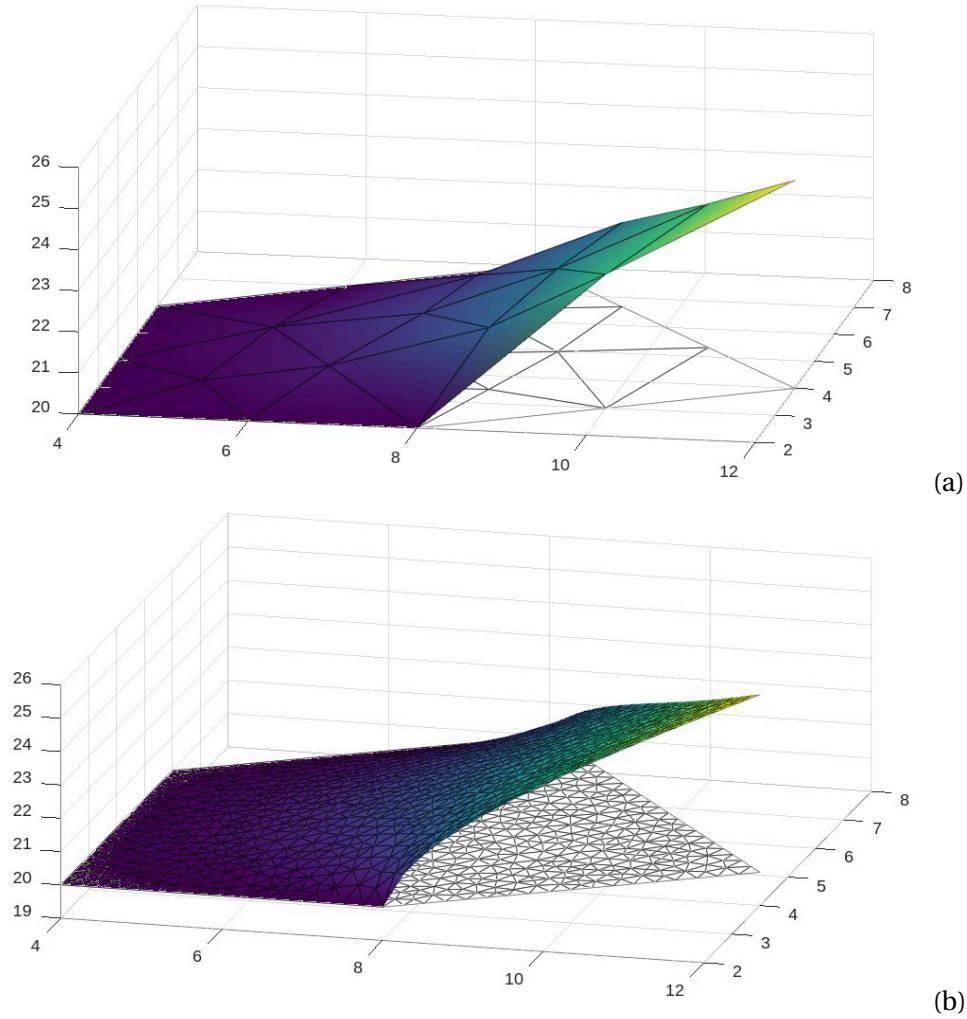


Figure 2.17 Galerkin solutions of the diffusion problem of Examples 2.9 and 2.10. In (a) the mesh of Figure 2.16 is used, while in (b) the adopted mesh is much finer.

Example 2.10 (Solution of the Neumann problem of Example 2.9) The procedure above has been implemented in **octavefemp1b.m**. The source f was set to zero and the diffusion coefficient to $q = 1$. The results obtained for the mesh of Figure 2.16 and for a finer mesh on the same geometry are shown in Figure 2.17.