### 1.5.1 The Differential Equation

The elliptic fourth-order problems that most frequently appear in applications are of the general form

$$\big(q(x)\,u''(x)\big)'' - \big(k(x)\,u'(x)\big)' + c(x)\,u(x) = f(x)\,. \tag{1.149}$$

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:**

1.77 *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

$$\big(E(x)\,I(x)\,u''(x)\big)'' = f(x) \tag{1.150}$$

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.

1.78 *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

$$q\,u'''' - k\,u'' = f\,, \tag{1.151}$$

where $q$ represents a *diffusion coefficient* and the other coefficients arise from the linearization. This is certainly a particular case of (1.149).

1.79 *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

$$\big(q(x)\,u''\big)'' + u = u_0 \tag{1.152}$$

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 problem is:

**Problem 1.5.** (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*

$$\big(q(x)u''(x)\big)'' + c(x)u(x) = f(x) \qquad \forall\, x \in \Omega \tag{1.153a}$$
$$u(0) = g_0 \tag{1.153b}$$
$$u'(0) = d_0 \tag{1.153c}$$
$$u''(L) = m_L \tag{1.153d}$$
$$u'''(L) = n_L \tag{1.153e}$$

**Examples:**

1.80 *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,

$$u(L) = \frac{T}{2q} L^2 + \frac{W}{3q} L^3\ .$$

1.81 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 $x = 0$)

$$u''(0) - \alpha_0\, u'(0) = \beta_0\,, \qquad u'''(0) + \gamma_0\, u(0) = \delta_0\,, \tag{1.154}$$

where $\alpha_0$, $\beta_0$, $\gamma_0$ and $\delta_0$ are given constants. Similarly to the second-order case, making $\alpha_0$ very large *de facto* imposes the value of $u'(0)$, and making $\gamma_0$ 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 1.3.** *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 1.5 is well posed.*

### 1.5.2 A Variational Equation

Let us follow the procedure outlined in §1.1.2.3 to determine a suitable variational equation for Problem 1.5. The residual is

$$r(x) = \big(q(x)\,u''(x)\big)'' + c(x)\,u(x) - f(x). \tag{1.155}$$

If $u$ is the solution, then $r(x) = 0$ for all $x$, and thus for any smooth function $v(x)$ it must hold that

$$0 = \int_0^L r(x)\,v(x)\,dx = \int_0^L \Big[\big(q(x)\,u''(x)\big)'' + c(x)\,u(x) - f(x)\Big]\; v(x)\,dx. \tag{1.156}$$

After distributing the product inside the bracket, we integrate by parts twice the term $\int_0^L (q u'')''\, v\,dx$ so that the differentiation order is balanced between $u$ and $v$, arriving at

$$\begin{aligned}\int_0^L \big[q(x)\,u''(x)\,v''(x) + c(x)\,u(x)\,v(x)\big]\;dx = \int_0^L f(x)\,v(x)\,dx \\ - (q u''' + q' u'')(L)\,v(L) + (q u''' + q' u'')\,v(0) \\ + q(L)\,u''(L)\,v'(L) - q(0)\,u''(0)\,v'(0).\end{aligned} \tag{1.157}$$

The second and third lines of (1.157) 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

$$\int_0^L \left[q(x)u''(x)v''(x) + c(x)u(x)v(x)\right] dx = \int_0^L f(x)v(x)\,dx - (q(L)n_L + q'(L)m_L)v(L) + (qu''' + q'u'')v(0) + q(L)m_L v'(L) - q(0)u''(0)v'(0). \quad (1.158)$$

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.

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**.

A variational equation that the solution $u$ of Problem 1.5 satisfies is

$$a(u, v) = \ell(v) \qquad \forall v \in \mathcal{V}, \quad (1.159a)$$

where

$$a(u, v) = \int_0^L \left[q(x)u''(x)v''(x) + c(x)u(x)v(x)\right] dx, \quad (1.159b)$$

$$\ell(v) = \int_0^L f(x)v(x)\,dx - \left(q(L)n_L + q'(L)m_L\right) v(L) + q(L)m_L v'(L) \quad (1.159c)$$

and

$$\mathcal{V} = \left\{v\colon [0, L] \to \mathbb{R} \text{ smooth} \mid v(0) = 0 \text{ and } v'(0) = 0\right\}. \quad (1.159d)$$

Notice that **the bilinear form $a(\cdot,\cdot)$ is symmetric**. You can check that the Euler-Lagrange equations of this variational equation are (1.153a), (1.153d) and (1.153e).

For completeness, the weak form of Problem 1.5 reads:

**Problem 1.6.** *(Weak form of Problem 1.5) Let the trial space be*

$$\mathcal{S} = \left\{v\colon [0, L] \to \mathbb{R} \textit{ smooth } \mid v(0) = g_0 \textit{ and } v'(0) = d_0\right\}$$

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

In this weak form, the test space $\mathcal{V}$ is the direction of the trial space $\mathcal{S}$, so it has the general structure of Problem 1.4, with $\mathcal{W} = \{v\colon [0, L] \to \mathbb{R} \text{ smooth}\}$. To see this, let $v, w \in \mathcal{S}$ and $z = v - w$. Then, $z(0) = v(0) - w(0) = g_0 - g_0 = 0$ and $z'(0) = v'(0) - w'(0) = d_0 - d_0 = 0$, and thus $z \in \mathcal{V}$.

In the context of Euler-Bernouilli beams, it so happens that $-\left(q(L)n_L + q'(L)m_L\right)$ equals the applied force load $W$ at $L$, while $q(L)m_L$ equals the applied torque $T$ at $L$. So, an equivalent form of writing (1.159c) is

$$\ell(v) = \int_0^L f(x)v(x)\,dx + W\,v(L) + T\,v'(L)\,. \quad (1.160)$$

### 1.5.3 A Variational Method

A variational method for variational equation (1.159a) is constructed as in Problem 1.2. To this end, we need to select a vector space $\mathcal{W}_h$, and set

$$\mathcal{S}_h = \{w_h \in \mathcal{W}_h \mid w_h(0) = g_0 \text{ and } w_h'(0) = d_0\}$$
$$\mathcal{V}_h = \{w_h \in \mathcal{W}_h \mid w_h(0) = 0 \text{ and } w_h'(0) = 0\}.$$

The variational method then reads:

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

The bilinear and linear forms are those in (1.159b) and (1.159c)). Let's see how we follow the solution procedure in §1.3.2 to obtain a solvable linear system of equations that allows us to compute $u_h$.

#### 1.5.3.1 A variational method with global polynomials

To proceed, we need a finite-dimensional space $\mathcal{W}_h$. For this example, we set $\mathcal{W}_h = \mathbb{P}_k([0, L])$ (polynomials of degree $\leq k$). The next step is to choose a basis of $\mathcal{W}_h$ of which a subset is a basis of $\mathcal{V}_h$. Let

$$N_1(x) = 1, \quad N_2(x) = x, \quad \ldots \quad N_{k+1}(x) = x^k\,,$$

then

$$\{N_1, N_2, \ldots, N_{k+1}\} \qquad \text{is a basis of } \mathcal{W}_h$$

and

$$\{N_3, N_4, \ldots, N_{k+1}\} \qquad \text{is a basis of } \mathcal{V}_h.$$

In fact, $N_1$ and $N_2$ are the only two basis functions of $\mathcal{W}_h$ that do not satisfy $N_a(0) = N_a'(0) = 0$. Therefore, the set of active indices is $\eta_a = \{3, \ldots, k+1\}$ and the set of constrained indices is $\eta_g = \{1, 2\}$.

By direct inspection, we see that

$$m = \dim \mathcal{W}_h = k+1 \qquad \text{and} \qquad n = \dim \mathcal{V}_h = m-2 = k-1\,.$$

Following the same reasoning as in §1.3.2, we write

$$u_h(x) = u_1 N_1(x) + u_2 N_2(x) + \ldots + u_m N_m(x) \tag{1.161}$$

so that $u_b$ ($b = 1, \ldots, m$) are the components of the numerical solutions and the algebraic unknowns of our problem. We can then choose $\overline{u}_h \in \mathcal{S}_h$ to be $\overline{u}_h = g_0 + d_0 x$, or $\overline{u}_h = g_0 + d_0 x + x^3$, for example. In both cases, $u_1 = g_0$ and $u_2 = d_0$.

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 1.82** To compute actual numbers, let us consider the "simplest beam problem" introduced in Example 1.80, 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 $K_{ab}$ can be calculated by straightforward integration. For $a > 2$ and $b > 2$ this results in

$$\begin{aligned} a(N_b, N_a) &= \int_0^L q\, N_b'' N_a''\, dx \\ &= \int_0^L q(b-1)(b-2)x^{b-3}(a-1)(a-2)x^{a-3}\, dx \\ &= \frac{q(b-1)(b-2)(a-1)(a-2)L^{a+b-5}}{a+b-5} \end{aligned}$$

and $a(N_b, N_a) = \delta_{ab}$ 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 numerical 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 1.80**. This is always true for the a variational method that is consistent: **if the exact solution lies in $\mathscr{S}_h$, then the solution of the variational method coincides with the exact solution**.

#### 1.5.3.2 The Convenience of a Smoother Space

Some variational methods for fourth-order problems require the use of a space $\mathscr{W}_h$ made of $C^1$ functions, that is, continuous functions with continuous derivatives. In particular, the variational method based on variational equation (1.159a) does. The reason for this is *consistency*, and if $\mathscr{W}_h$ is not made of $C^1$ functions, such method is not consistent. As a result, the method may not converge to the exact solution of Problem 1.5, a fact that we will see in §**??**.

To evaluate consistency, we need to see if the exact solution of Problem 1.5 is also a solution of the method, or Problem 1.2. Specifically, is

$$a(u, v_h) = \ell(v_h) \qquad \forall\, v_h \in \mathscr{V}_h?$$

To see that this is not necessarily true, we consider the case with $T = W = 0$, $q(x) = 1$ and $c(x) = 0$, for simplicity. This implies that $m_L = n_L = 0$, and hence that $u''(L) = u'''(L) = 0$. In this case, we need to check if for all $v_h \in \mathscr{V}_h$

$$\int_0^L u''(x)\, v_h''(x)\, dx = \int_0^L f(x)\, v_h(x)\, dx. \tag{1.162}$$

If $\mathcal{V}_h$ contains discontinuous functions, or functions with discontinuous derivatives, then the integration by parts formula in §1.45 needs to be applied. For simplicity, let's assume that functions in $\mathcal{V}_h$ can have a discontinuity in the function or its derivative at $x = L/2$ only. In this case, integrating by parts the left hand side of (1.162) to transfer a derivative to $u$ we obtain

$$\begin{aligned}
&\int_0^L f(x)\, v_h(x)\, dx \\
&\qquad = \underbrace{u''(L)}_{=0}\, v_h'(L) - u''(0)\, \underbrace{v_h'(0)}_{=0} + [\![ u''(x)\, v_h'(x) ]\!]_{x=L/2} - \int_0^L u'''(x)\, v_h'(x)\, dx \qquad \text{($v_h \in \mathcal{V}_h$ and $m_L = 0$)} \\
&\qquad = u''(L/2)\, [\![ v_h'(x) ]\!]_{x=L/2} - \int_0^L u'''(x)\, v_h'(x)\, dx. \qquad \text{($u''$ is continuous at $x = L/2$)}
\end{aligned}$$

Integrating by parts once more we get

$$\begin{aligned}
\int_0^L f(x)\, v_h(x)\, dx &= u''(L)\, [\![ v_h'(x) ]\!]_{x=L/2} - \underbrace{u'''(L)}_{=0}\, v_h(L) + u'''(0)\, \underbrace{v_h(0)}_{=0} \qquad \text{($v_h \in \mathcal{V}_h$ and $n_L = 0$)} \\
&\quad - [\![ u'''(x)\, v_h(x) ]\!]_{x=L/2} + \int_0^L u''''(x)\, v_h(x)\, dx \\
\int_0^L \underbrace{(f(x) + u''''(x))}_{=0}\, v_h(x)\, dx &= u''(L/2)\, [\![ v_h'(x) ]\!]_{x=L/2} - u'''(L/2)\, [\![ v_h(x) ]\!]_{x=L/2}. \qquad \text{($u'''$ is continuous at $x = L/2$ and (1.153a))}
\end{aligned}$$

Therefore, the method would be consistent if and only if

$$0 = u''(L/2)\, [\![ v_h'(x) ]\!]_{x=L/2} - u'''(L/2)\, [\![ v_h(x) ]\!]_{x=L/2}$$

for *all* $v_h \in \mathcal{V}_h$. Since we can select the jumps of $v_h$ or $v_h'$ arbitrarily by selecting appropriate functions in $\mathcal{V}_h$, we conclude that the method is consistent if and only if $u''(L/2) = 0$ and $u'''(L/2)$. Since this is not necessarily true for the exact solution of Problem 1.5, this variational method is not necessarily consistent.

Instead, if $\mathcal{W}_h$ is made of $C^1$ functions, $[\![ v_h'(x) ]\!] = [\![ v_h(x) ]\!] = 0$ for all $x \in (0, L)$, and hence this method would be consistent.

It is possible to formulate consistent methods with a space $\mathcal{W}_h$ that contains functions that are not necessarily $C^1$, but it requires a different variational equation, inspired for example by the interior penalty method in Example 1.16.

### 1.5.4 The Simplest $C^1$ Finite Element Space

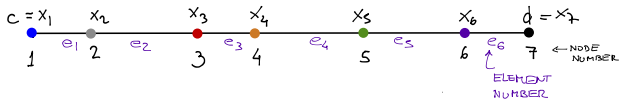
Let us now introduce the simplest **finite element space** that can approximate the fourth-order Problem 1.5, known as the **Hermite piecewise cubic space**. We will denote it here as $H_3$-space. We follow the same steps as in §1.4.1, but notice that some important differences arise.

**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

$$0 = x_1 < \ldots < x_{n_{\text{el}}+1} = L. \tag{1.163}$$

   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 a variational method based on variational equation (1.159a), since hat functions have discontinuous derivatives, and therefore the method would not be consistent.

   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 $n_{\text{vert}}$ 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 $n_{\text{el}}$ 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:

$$H_{2a-1}(x) = \begin{cases} 0 & \text{if } x < x_{a-1} \\ -2\left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^3 + 3\left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^2 & \text{if } x_{a-1} \le x < x_a \\ 1 & \text{if } x = x_a \\ 2\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^3 - 3\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^2 + 1 & \text{if } x_a < x \le x_{a+1} \\ 0 & \text{if } x_{a+1} < x \end{cases} \tag{1.164}$$

**b)** Even-numbered basis functions:

$$H_{2a}(x) = \tag{1.165}$$
$$= \begin{cases} 0 & \text{if } x < x_{a-1} \\ \left[\left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^3 - \left(\frac{x-x_{a-1}}{x_a-x_{a-1}}\right)^2\right](x_a - x_{a-1}) & \text{if } x_{a-1} \le x < x_a \\ 0 & \text{if } x = x_a \\ \left[\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^3 - 2\left(\frac{x-x_a}{x_{a+1}-x_a}\right)^2 + \left(\frac{x-x_a}{x_{a+1}-x_a}\right)\right](x_{a+1} - x_a) & \text{if } x_a < x \le x_{a+1} \\ 0 & \text{if } x_{a+1} < x \end{cases}$$

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 (1.164)-(1.165) 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., $H'_{2a}(x_a) = 1$ 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.,

$$\mathcal{W}_h = \text{span}\left(H_1, H_2, \ldots, H_{2n_{\text{vert}}}\right), \tag{1.166}$$

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

$$\begin{cases} c_1 &= w_h(x_1), \\ c_2 &= w'_h(x_1), \\ c_3 &= w_h(x_2), \\ c_4 &= w'_h(x_2), \\ \ldots & \ldots \\ c_{2n_{\text{vert}}-1} &= w_h(x_{n_{\text{vert}}}), \\ c_{2n_{\text{vert}}} &= w'_h(x_{n_{\text{vert}}}). \end{cases} \tag{1.167}$$

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 (1.159a). 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 $\mathcal{W}_h$,

$$w_h(x) = c_1 H_1(x) + c_2 H_2(x) + \ldots + c_{2n_{\text{vert}}} H_{2n_{\text{vert}}}(x) \,. \tag{1.168}$$

   Now, since $x_1 = 0$, we have that

$$w_h(0) = c_1, \qquad \text{and} \qquad w'_h(0) = c_2 \,. \tag{1.169}$$

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

$$\mathcal{V}_h = \left\{ v_h \in \mathcal{W}_h | v_h(0) = v_h'(0) = 0 \right\} = \text{span}\left( H_3, H_4, \ldots, H_{2n_{\text{vert}}} \right), \tag{1.170}$$

meaning that $\mathcal{V}_h$ consists of all functions of the form

$$v_h(x) = c_3 H_3(x) + c_4 H_4(x) + \ldots + c_{2n_{\text{vert}}} H_{2n_{\text{vert}}}(x) \,, \tag{1.171}$$

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 (1.169) 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

$$z_h(x) = g_0 H_1(x) + d_0 H_2(x) + c_3 H_3(x) + \ldots + c_{2n_{\text{vert}}} H_{2n_{\text{vert}}}(x) \,, \tag{1.172}$$

where the coefficients $c_3, \ldots, c_{2n_{\text{vert}}}$ are arbitrary. Another way of writing the definition of $\mathcal{S}_h$ is

$$\begin{aligned} \mathcal{S}_h &= \left\{ z_h \in \mathcal{W}_h | z_h = g_0 H_1 + d_0 H_2 + v_h,\, v_h \in \mathcal{V}_h \right\} \\ &= g_0 H_1 + d_0 H_2 + \mathcal{V}_h \,. \end{aligned} \tag{1.173}$$

If we go back to the discussion that led to (1.72b), we see that (1.173) *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 with $m$ unknowns.

4. **Compute $K$ and $F$.** Let the finite element solution be denoted by

$$u_h(x) = u_1 H_1(x) + u_2 H_2(x) + \ldots + u_m H_m(x) \,, \tag{1.174}$$

and let $U$ be the column vector of its coefficients,

$$U = (u_1, u_2, \ldots, u_m)^T \,.$$

Inserting (1.174) into $a(u_h, v_h)$, particularizing for $v_h = H_a$, with $a \in \eta_a$ and incorporating the essential boundary conditions, we have that, for $a, b \in \eta = \{1, \ldots, m\}$,

$$K_{ab} = \begin{cases} \delta_{ab} & \text{if } a \in \eta_g = \{1,2\}, \\ a(H_b, H_a) & \text{if } a \in \eta_a \,. \end{cases} \tag{1.175}$$

For the load vector,

$$F_a = \begin{cases} g_0 & \text{if } a = 1 \,, \\ d_0 & \text{if } a = 2 \,, \\ \ell(H_a) & \text{if } a \in \eta_a \,. \end{cases} \tag{1.176}$$

**Example 1.83 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 1.6 and those of the basis functions from (1.164)-(1.165). 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 $u_h(x) = \sum_{a\in\eta} u_a H_a(x)$.

   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

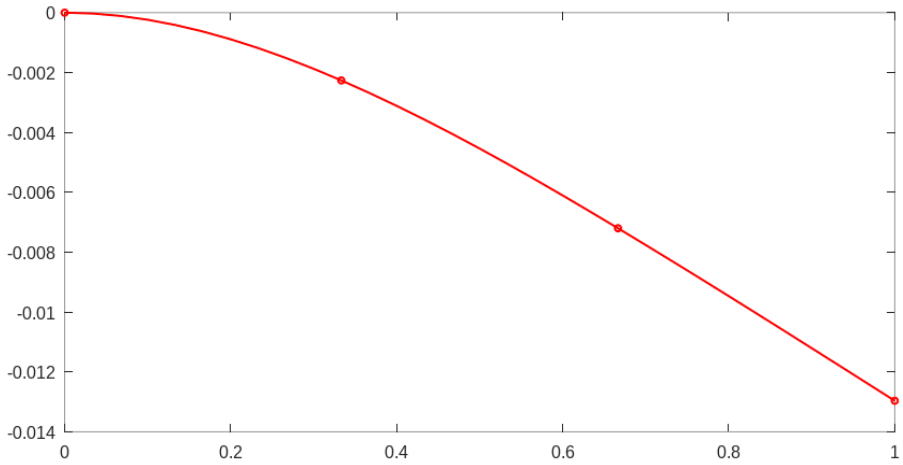



**Figure 1.20** The finite element solution $u_h$ corresponding to Equation 1.177.

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

$$u_h(x) = -0.0023H_3(x) - 0.012H_4(x) - 0.0072H_5(x) \\ - 0.0167H_6(x) - 0.013H_7(x) - 0.0176H_8(x). \quad (1.177)$$

This function is plotted in Fig. 1.20.

### 1.5.5 The Cubic Hermite Finite Element

The basis functions $H_1,\ldots,H_{2n_{\text{vert}}}$ introduced in (1.164)-(1.165) can also be viewed as generated by the following finite element:

$$\Omega_e = \left[x_1^e, x_2^e\right], \tag{1.178}$$

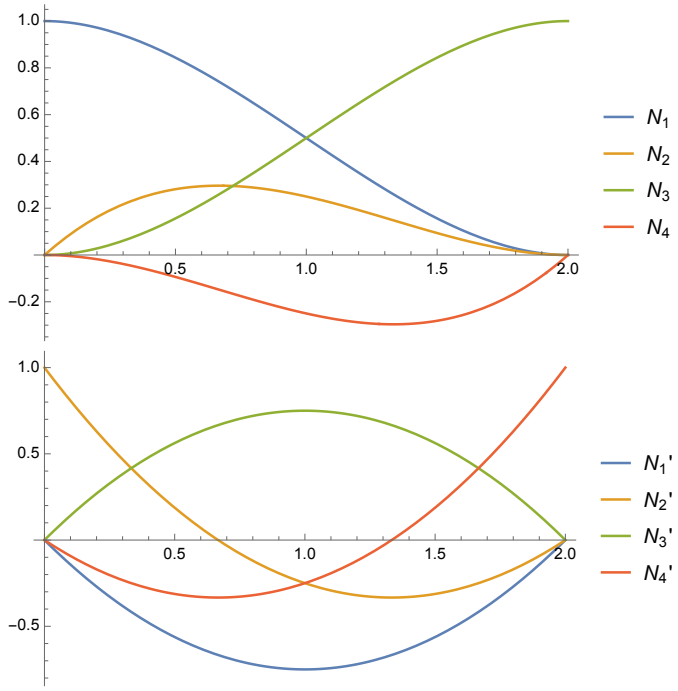
$$N_1^e(x) = \left(\frac{x_2^e - x}{x_2^e - x_1^e}\right)^2 \left(1 + 2\frac{x - x_1^e}{x_2^e - x_1^e}\right), \tag{1.179}$$

$$N_3^e(x) = \left(\frac{x_1^e - x}{x_1^e - x_2^e}\right)^2 \left(1 + 2\frac{x - x_2^e}{x_1^e - x_2^e}\right), \tag{1.180}$$

$$N_2^e(x) = \left(\frac{x_2^e - x}{x_2^e - x_1^e}\right)^2 \left(x - x_1^e\right), \tag{1.181}$$

$$N_4^e(x) = \left(\frac{x_1^e - x}{x_1^e - x_2^e}\right)^2 \left(x - x_2^e\right). \tag{1.182}$$

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

$$N_1^e(x_1^e) = 1, \qquad (N_1^e)'(x_1^e) = 0, \qquad N_1^e(x_2^e) = 0, \qquad (N_1^e)'(x_2^e) = 0,$$

$$N_2^e(x_1^e)=0, \quad (N_2^e)'(x_1^e)=1, \quad N_2^e(x_2^e)=0, \quad (N_2^e)'(x_2^e)=0,$$
$$N_3^e(x_1^e)=0, \quad (N_3^e)'(x_1^e)=0, \quad N_3^e(x_2^e)=1, \quad (N_1^e)'(x_2^e)=0,$$
$$N_4^e(x_1^e)=0, \quad (N_4^e)'(x_1^e)=0, \quad N_4^e(x_2^e)=0, \quad (N_4^e)'(x_2^e)=1,$$

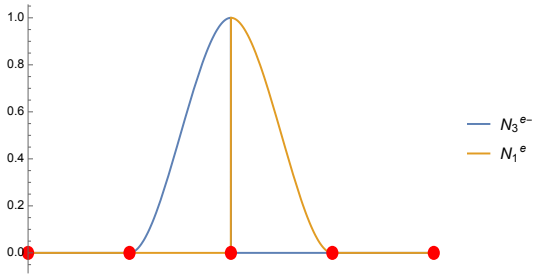
which implies that the **degrees of freedom** are

$$\phi_1^e = f^e(x_1^e), \quad \phi_2^e = (f^e)'(x_1^e), \quad \phi_3^e = f^e(x_2^e), \quad \phi_4^e = (f^e)'(x_2^e), \tag{1.183}$$

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 $\Omega$ is nothing but the function $H_{2a-1}(x)$, already introduced in (1.164), 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 (1.165), assuming that $a$ is the *left* node of $e$.

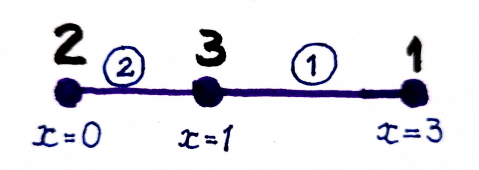
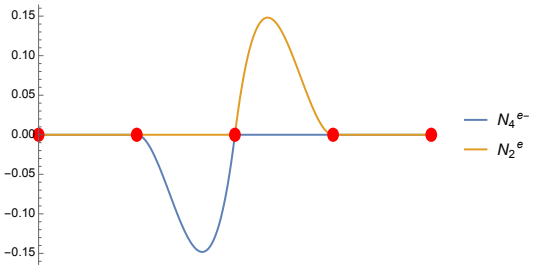



**Figure 1.21** Mesh of the example in the text. The circled numbers correspond to the elements, the bare numbers to the nodes.




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 1.4.4. This is best explained by carefully carrying out an example.

**Example 1.84**

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. 1.21. 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

$$N_A = \mathop{\mathring{\sum}}_{\{(a,e)|\mathsf{LG}(a,e)=A\}} N_a^e$$

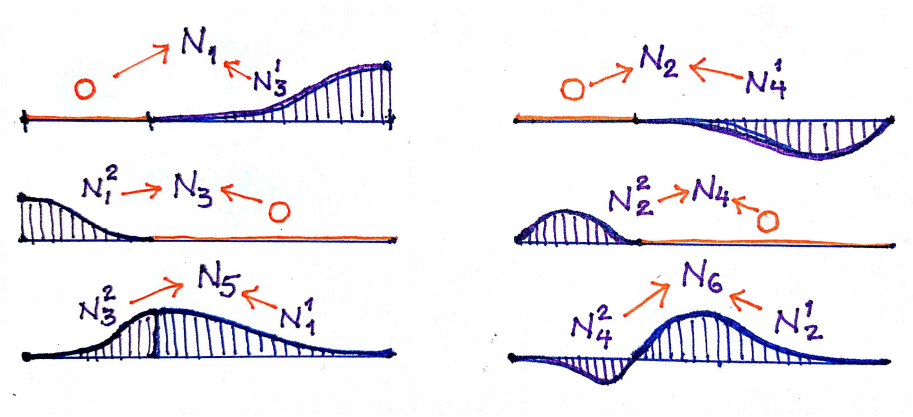



**Figure 1.22** The six basis functions generated by the local-to-global map LG provided in the example.

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. 1.22. They can be directly compared to the Hermite basis functions $H_k$ introduced in (1.164)-(1.165). 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. 1.21 (without renumbering the elements) produces basis functions that satisfy $N_k = H_k$ for all $k = 1,\ldots,m$.

### 1.5.6 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 1.83.

From the bilinear form (1.159b) we know that the contribution of each element is the element stiffness matrix

$$\mathsf{K}^e_{ij} = \int_{x_1^e}^{x_2^e} \left[ q\,(N_j^e)"\,(N_i^e)" + c\,N_j^e\,N_i^e \right] dx. \tag{1.184}$$

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:

$$\begin{array}{lcllcl}
\int_{x_1^e}^{x_2^e}(N_1^e)"\,(N_1^e)"\,dx &=& \frac{12}{h_e^3}, & \int_{x_1^e}^{x_2^e} N_1^e\,N_1^e\,dx &=& \frac{13h_e}{35},\\
\int_{x_1^e}^{x_2^e}(N_2^e)"\,(N_1^e)"\,dx &=& \frac{6}{h_e^2}, & \int_{x_1^e}^{x_2^e} N_2^e\,N_1^e\,dx &=& \frac{11h_e^2}{210},\\
\int_{x_1^e}^{x_2^e}(N_3^e)"\,(N_1^e)"\,dx &=& -\frac{12}{h_e^3}, & \int_{x_1^e}^{x_2^e} N_3^e\,N_1^e\,dx &=& \frac{9h_e}{70},\\
\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_1^e)"\,dx &=& \frac{6}{h_e^2}, & \int_{x_1^e}^{x_2^e} N_4^e\,N_1^e\,dx &=& -\frac{13h_e^2}{420},\\
\int_{x_1^e}^{x_2^e}(N_2^e)"\,(N_2^e)"\,dx &=& \frac{4}{h_e}, & \int_{x_1^e}^{x_2^e} N_2^e\,N_2^e\,dx &=& \frac{h_e^3}{105},\\
\int_{x_1^e}^{x_2^e}(N_3^e)"\,(N_2^e)"\,dx &=& -\frac{6}{h_e^2}, & \int_{x_1^e}^{x_2^e} N_3^e\,N_2^e\,dx &=& \frac{13h_e^2}{420},\\
\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_2^e)"\,dx &=& \frac{2}{h_e}, & \int_{x_1^e}^{x_2^e} N_4^e\,N_2^e\,dx &=& -\frac{h_e^3}{140},\\
\int_{x_1^e}^{x_2^e}(N_3^e)"\,(N_3^e)"\,dx &=& \frac{12}{h_e^3}, & \int_{x_1^e}^{x_2^e} N_3^e\,N_3^e\,dx &=& \frac{13h_e}{35},\\
\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_3^e)"\,dx &=& -\frac{6}{h_e^2}, & \int_{x_1^e}^{x_2^e} N_4^e\,N_3^e\,dx &=& -\frac{11h_e^2}{210},\\
\int_{x_1^e}^{x_2^e}(N_4^e)"\,(N_4^e)"\,dx &=& \frac{4}{h_e}, & \int_{x_1^e}^{x_2^e} N_4^e\,N_4^e\,dx &=& \frac{h_e^3}{105},
\end{array}$$

where $h_e = x_2^e - x_1^e$. Then the (symmetric) element stiffness matrix ends up being

$$\mathsf{K}^e = \begin{pmatrix}
\frac{12q_e}{h_e^3} + \frac{13c_eh_e}{35} & \frac{6q_e}{h_e^2} + \frac{11c_eh_e^2}{210} & -\frac{12q_e}{h_e^3} + \frac{9c_eh_e}{70} & \frac{6q_e}{h_e^2} - \frac{13c_eh_e^2}{420}\\
\text{symm} & \frac{4q_e}{h_e} + \frac{c_eh_e^3}{105} & -\frac{6q_e}{h_e^2} + \frac{13c_eh_e^2}{420} & \frac{2q_e}{h_e} - \frac{c_eh_e^3}{140}\\
\text{symm} & \text{symm} & \frac{12q_e}{h_e^3} + \frac{13c_eh_e}{35} & -\frac{6q_e}{h_e^2} - \frac{11c_eh_e^2}{210}\\
\text{symm} & \text{symm} & \text{symm} & \frac{4q_e}{h_e} + \frac{c_eh_e^3}{105}
\end{pmatrix} \tag{1.185}$$

Turning now to the element load vector, we will compute it without the end contributions, which will be added later on. From (1.159c) we have

$$\mathsf{F}_i^e = \int_{x_1^e}^{x_2^e} f(x)\, N_i^e(x)\, dx, \tag{1.186}$$

which again can in principle computed exactly.

As an interesting special case, let us compute $\mathsf{F}^e$ 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:

$$\mathsf{F}^e = \begin{pmatrix} \frac{f_e h_e}{2} \\ \\ \frac{f_e h_e^2}{12} \\ \\ \frac{f_e h_e}{2} \\ \\ -\frac{f_e h_e^2}{12} \end{pmatrix} \tag{1.187}$$

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
```

### 1.5.7 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 solution of the variational method 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 $n_{\text{nod}}$, $n_{\text{el}}$ 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 $\eta_g$ (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 1.85 (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. 1.20, 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. 1.23, which is very similar to that of Fig. 1.20 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. 1.24. Element number 4, being less stiff, acts as a hinge at which most of the deformation concentrates.

**Example 1.86 (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 1.25. 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.

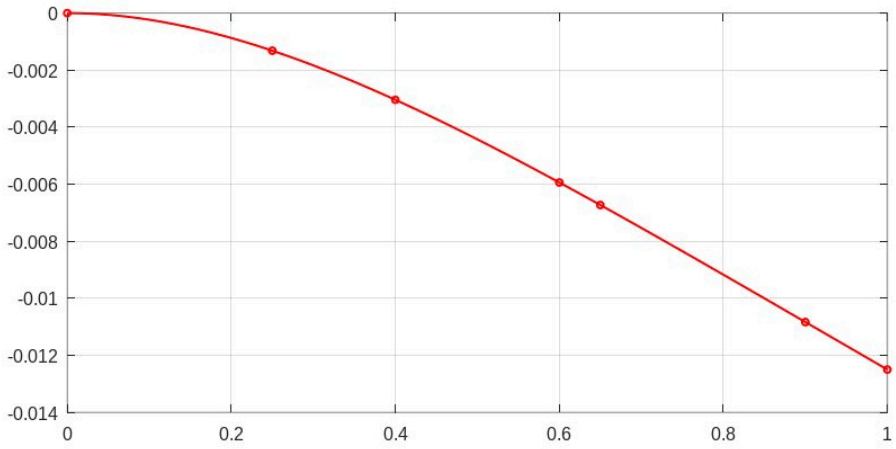



**Figure 1.23** Finite element solution from Example 1.85.

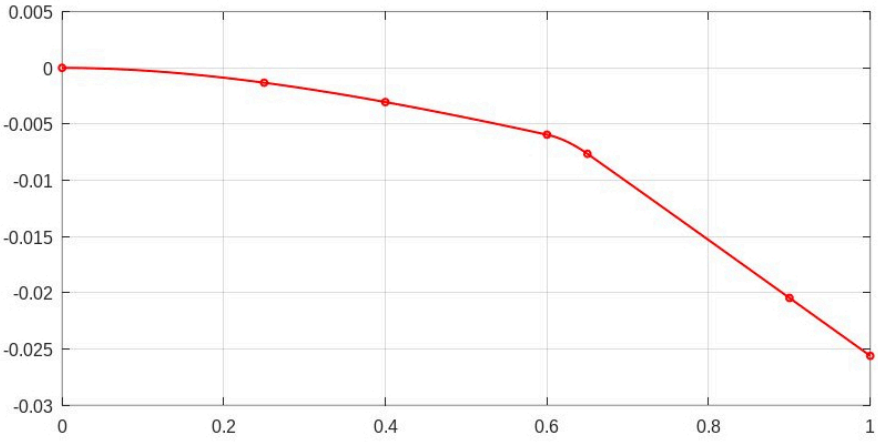



**Figure 1.24** Finite element solution from Example 1.85. The bending rigidity of the beam is equal to 10 in all elements except for element 4, where its value has been reduced to 0.1.

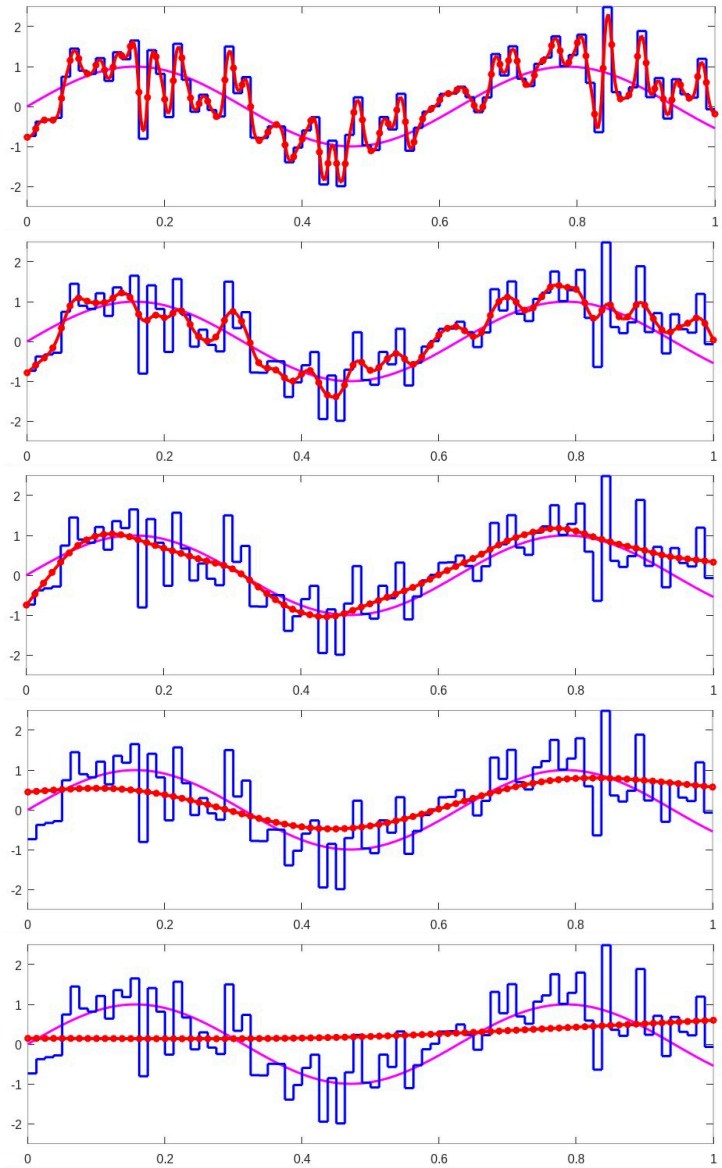



**Figure 1.25** Results from the image denoising example. In blue the original image $u_0$, which is a random perturbation of the exact function drawn in magenta. In red we plot the solution $u_h$ of the image-denoising example for $q$ taking values, from top to bottom, $q = 10^{-10}$, $10^{-8}$, $10^{-6}$, $10^{-4}$ and $10^{-2}$.