$$\begin{aligned}\mathcal{D}^{(n)} = {} & \frac{l^{(n)}\left[n_x^{(n)}\left(x^{(n)}-\xi\right)+n_y^{(n)}\left(y^{(n)}-\eta\right)\right]}{\pi\sqrt{F^{(n)}}} \\ & \times\left[\arctan\frac{2A^{(n)}+B^{(n)}}{\sqrt{F^{(n)}}}-\arctan\frac{B^{(n)}}{\sqrt{F^{(n)}}}\right].\end{aligned}$$

On the other hand, if $F^{(n)}=0$, we need to compute

$$\begin{aligned}\mathcal{V}^{(n)} &= \frac{l^{(n)}}{2\pi}\left\{\ln l^{(n)}+\left[1+\frac{B^{(n)}}{2A^{(n)}}\right]\ln\left|1+\frac{B^{(n)}}{2A^{(n)}}\right|-\frac{B^{(n)}}{2A^{(n)}}\ln\left|\frac{B^{(n)}}{2A^{(n)}}\right|-1\right\}, \\ \mathcal{D}^{(n)} &= 0.\end{aligned}$$

The boundary element method is also applicable to non-stationary problems: time integration is performed separately. A superb introduction is given in [12].

10.6 Finite-Element Method ⋆

Finite-element methods (FEM) are five decades old, yet they remain the basic tool of engineers and natural scientists, especially for problems in elastomechanics, hydro- and aero-dynamics in complex geometries. In difference methods we use finite differences to approximate the differential equation, i.e. its space and time derivatives, and impose boundary conditions appropriately. But in complex geometries this is very hard to accomplish.

In the finite-element approach, we find the integral of the differential equation on its definition domain, and thereby express the equation in its variational form. The domain is then divided into smaller units (finite elements) on which the solution of the equation is approximated by a linear combination of some basis functions. The individual elements may be positioned over the domain quite freely (Fig. 10.9), and this liberty represents the main charm and strength of the method. Finally, the variational integral is computed by summing the contributions from all elements; we end up with a system of algebraic equations for the coefficients multiplying the basis functions in the solution expansion.

Here we present the essence of FEM. For greater clarity, the basic concepts are introduced in one space dimension, while in applications the method is overwhelmingly used in two and three dimensions. Further reading can be found in the textbooks [13] and [14] which we follow closely. See also [15].

10.6.1 One Space Dimension

The finite-element method started to bloom in construction problems, so it is fair to introduce it by a civil-engineering example: we are interested in the transverse

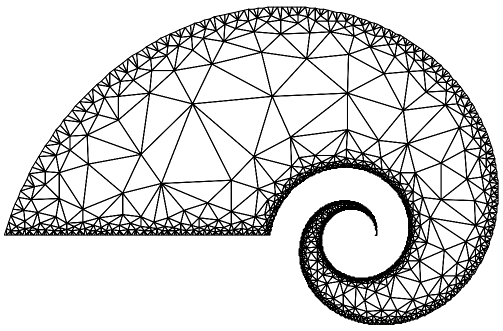


Fig. 10.9 Positioning the finite elements for the problem of propagating acoustic waves in the geometry of a "snail". Such complex geometries are virtually unmanageable by classical difference methods. Figure courtesy of M. Melenk and S. Langdon

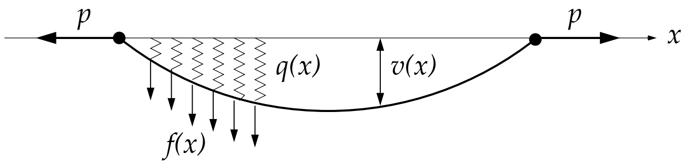


Fig. 10.10 The classical boundary-value problem illustrating the one-dimensional finite-element method: transverse deflections of a cable under non-uniform load

deflections $v(x)$ of a support cable pulled at its ends by the force p. There is another force (per unit length) acting in the transverse direction, $f(x)$, and the cable is held in equilibrium by springs with elastic modulus $q(x)$ (Fig. 10.10).

The deflection $v(x)$ from the horizontal is the solution of the boundary-value problem

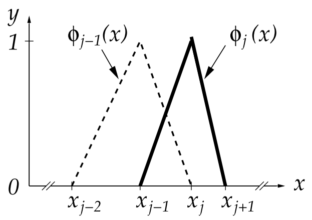
$$-p(x)v''(x) + q(x)v(x) = f(x), \quad x \in [0, 1], \tag{10.64}$$

with boundary conditions $v(0) = v(1) = 0$. For simplicity we assume constant $p > 0$ and $q \geq 0$. We have shown in Sect. 8.6 (see (8.82)) that seeking the solution of this problem is equivalent to determining the function v satisfying the equation

$$A(w, v) = \int_0^1 \left[w'pv' + wqv\right] \mathrm{d}x = \int_0^1 wf \, \mathrm{d}x = \langle w, f \rangle$$

for any weight function w.

We divide the interval $[0, 1]$ to N (not necessarily uniform) subintervals such that $0 = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = 1$. Each subinterval $[x_{j-1}, x_j]$ with length $h_j = x_j - x_{j-1}$ is called the *finite element*. Over the adjacent elements $[x_{j-1}, x_j]$ and $[x_j, x_{j+1}]$ we span the basis function ϕ_j with a peak at x_j and the *nodes* (zeros) at x_{j-1} and x_{j+1} (Fig. 10.11). We use piecewise linear basis functions here; quadratic and cubic functions are also widely used.



$$\phi_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}}; & x_{j-1} \le x < x_j, \\ \frac{x_{j+1}-x}{x_{j+1}-x_j}; & x_j \le x < x_{j+1}, \\ 0; & \text{otherwise.} \end{cases}$$

Fig. 10.11 Basis functions for the one-dimensional finite-element method. The function ϕ_j is non-zero only on the interval $(x_{j-1}, x_j] \cup [x_j, x_{j+1})$, and $\phi_j(x_k) = \delta_{j,k}$

The approximate solution is expanded in terms of the basis functions ϕ_j:

$$u(x) = \sum_{j=1}^{N-1} c_j \phi_j(x). \tag{10.65}$$

Since $\phi_j(x_k) = \delta_{j,k}$, we get $u(x_k) = \sum_j c_j \delta_{j,k} = c_k$. The coefficients c_k are therefore equal to the values of u at the interior nodes. For Dirichlet boundary conditions, $u(x_0) = c_0$ and $u(x_N) = c_N$, so we may include the contributions at the boundary nodes in (10.65) if we set $c_0 = c_N = 0$. The weight function is expanded similarly,

$$\widetilde{w}(x) = \sum_{j=1}^{N-1} d_j \psi_j(x).$$

We stay in the Galerkin framework, where the basis functions ϕ_j for the solution u are the same as the basis functions ψ_j for the weight function $\widetilde{w}$.

The coefficients c_k in the expansion (10.65) are computed by solving the variational equation (8.83), where the sum runs over all finite elements. The equation to be solved is

$$\sum_{j=1}^{N} \big[A_j(\widetilde{w}, u) - \langle \widetilde{w}, f \rangle_j \big] = 0 \quad \forall \widetilde{w}, \tag{10.66}$$

where u is the approximate solution (10.65), $\widetilde{w}$ is the weight function, and f is the right-hand side of (10.64). The subscript j denotes the element $[x_{j-1}, x_j]$ and runs from 1 to N, since all finite elements from $[x_0, x_1]$ through $[x_{N-1}, x_N]$ have to be considered. We split the expression for A_j in two parts,

$$A_j(\widetilde{w}, u) = A_j^{\mathrm{S}}(\widetilde{w}, u) + A_j^{\mathrm{M}}(\widetilde{w}, u) = \int_{x_{j-1}}^{x_j} p \widetilde{w}' u' \,\mathrm{d}x + \int_{x_{j-1}}^{x_j} q \widetilde{w} u \,\mathrm{d}x. \tag{10.67}$$

Stiffness Matrix, Mass Matrix, and Load Vector In engineering problems the first term of (10.67) corresponds to the internal energy of the body (due to compression or expansion), and the second term to external influences (e.g. due to the change of potential energy under load). Over the element $[x_{j-1}, x_j]$, the approximate solution and the weight function have the forms

$$u(x) = c_{j-1}\phi_{j-1}(x) + c_j\phi_j(x) = (c_{j-1}, c_j)\begin{pmatrix}\phi_{j-1}(x)\\ \phi_j(x)\end{pmatrix},$$

$$\widetilde{w}(x) = d_{j-1}\phi_{j-1}(x) + d_j\phi_j(x) = (d_{j-1}, d_j)\begin{pmatrix}\phi_{j-1}(x)\\ \phi_j(x)\end{pmatrix},$$

with the derivatives

$$u'(x) = (c_{j-1}, c_j)\begin{pmatrix}-1/h_j\\ 1/h_j\end{pmatrix}, \qquad \widetilde{w}'(x) = (d_{j-1}, d_j)\begin{pmatrix}-1/h_j\\ 1/h_j\end{pmatrix},$$

where $h_j = x_j - x_{j-1}$. From these expressions one finds

$$A_j^{\mathrm{S}}(\widetilde{w}, u) = (d_{j-1}, d_j)\left[\int_{x_{j-1}}^{x_j}\begin{pmatrix}1/h_j^2 & -1/h_j^2\\ -1/h_j^2 & 1/h_j^2\end{pmatrix}p(x)\,\mathrm{d}x\right]\begin{pmatrix}c_{j-1}\\ c_j\end{pmatrix},$$

$$A_j^{\mathrm{M}}(\widetilde{w}, u) = (d_{j-1}, d_j)\left[\int_{x_{j-1}}^{x_j}\begin{pmatrix}\phi_{j-1}^2 & \phi_{j-1}\phi_j\\ \phi_{j-1}\phi_j & \phi_j^2\end{pmatrix}q(x)\,\mathrm{d}x\right]\begin{pmatrix}c_{j-1}\\ c_j\end{pmatrix}.$$

For constant p and q, as assumed for (10.64), only elementary integrals are involved (trivial integration in A_j^{S} and integration of quadratic functions in A_j^{M}). In this case we get

$$A_j^{\mathrm{S}}(\widetilde{w}, u) = (d_{j-1}, d_j)S_j\begin{pmatrix}c_{j-1}\\ c_j\end{pmatrix}, \quad S_j = \frac{p}{h_j}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}, \tag{10.68}$$

$$A_j^{\mathrm{M}}(\widetilde{w}, u) = (d_{j-1}, d_j)M_j\begin{pmatrix}c_{j-1}\\ c_j\end{pmatrix}, \quad M_j = \frac{qh_j}{6}\begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}. \tag{10.69}$$

The matrix S_j is known as the *element (local) stiffness matrix*, and the matrix M_j is the *element (local) mass matrix*.

In general, the integral $\langle\widetilde{w}, f\rangle_j$ in (10.66) (the scalar product of functions $\widetilde{w}$ and f on the jth finite element) is computed numerically. But a good approximation can be obtained if f on $[x_{j-1}, x_j]$ is replaced by a piecewise linear function, $f(x) \approx f_{j-1}\phi_{j-1}(x) + f_j\phi_j(x)$. Then

$$\langle\widetilde{w}, f\rangle_j \approx (d_{j-1}, d_j)\boldsymbol{g}_j, \quad \boldsymbol{g}_j = \frac{h_j}{6}\begin{pmatrix}2f_{j-1} + f_j\\ f_{j-1} + 2f_j\end{pmatrix}. \tag{10.70}$$

The vector $\boldsymbol{g}_j$ is the *element (local) load vector*, as it describes the external force or loading upon the system.

Assembly The next step is the *assembly* of the element matrices into the global stiffness matrix by summing the contributions of all elements, and adding all element load vectors into the global load vector. To simplify, we consider a uniform mesh, $h_j = h = 1/N$. We sum the contributions (10.68), (10.69), and (10.70) over all elements $1, 2, \ldots, N$, where we consider the boundary condition $c_0 = d_0 = 0$ in the terms involving the element $j = 0$, and $c_N = d_N = 0$ in those involving $j = N$. We collect the coefficients c_j and d_j for the interior points of $[x_0, x_N]$ in the vectors $\boldsymbol{c} = (c_1, c_2, \ldots, c_{N-1})^{\mathrm{T}}$ and $\boldsymbol{d} = (d_1, d_2, \ldots, d_{N-1})^{\mathrm{T}}$, and put the components g_j in the global load vector $\boldsymbol{g} = (g_1, g_2, \ldots, g_{N-1})^{\mathrm{T}}$. The equation with the sum over the elements,

$$\sum_{j=1}^{N} \left[A_j^{\mathrm{S}}(\widetilde{w}, u) + A_j^{\mathrm{M}}(\widetilde{w}, u) \right] = \sum_{j=1}^{N} \langle \widetilde{w}, f \rangle_j,$$

can then be written in matrix form $\boldsymbol{d}^{\mathrm{T}}[(S + M)\boldsymbol{c} - \boldsymbol{g}] = 0$, where

$$S = \frac{p}{h} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}, \qquad M = \frac{qh}{6} \begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix},$$

are the $(N-1) \times (N-1)$ global stiffness and mass matrices, and

$$\boldsymbol{g} = \frac{h}{6} \begin{pmatrix} f_0 + 4f_1 + f_2 \\ f_1 + 4f_2 + f_3 \\ \vdots \\ f_{N-2} + 4f_{N-1} + f_N \end{pmatrix}$$

is the global load vector of dimension $N - 1$ (in the linear approximation on each element). This equation must hold true for any $\boldsymbol{d}$, so the coefficients c_j of the expansion (10.65) for $j = 1, 2, \ldots, N-1$ are obtained by solving the system

$$(S + M)\boldsymbol{c} = \boldsymbol{g}.$$

10.6.2 Two Space Dimensions

In two dimensions the problem (10.64) is generalized to a planar region R,

$$-\frac{\partial}{\partial x}\left(p(x, y) \frac{\partial v}{\partial x} \right) - \frac{\partial}{\partial y}\left(p(x, y) \frac{\partial v}{\partial y} \right) + q(x, y)v = f(x, y), \quad (x, y) \in R,$$

with the boundary condition $v(x, y) = \alpha(x, y)$ for $(x, y) \in \partial R$. (A problem with a non-homogeneous condition $\alpha \neq 0$ can be turned into the problem with $\alpha = 0$ by shifting $\hat{v} = v - \alpha$.) The variational formulation of the problem is analogous to the one-dimensional case; things start to become complicated when we attempt to divide (partition) the region R to finite elements.

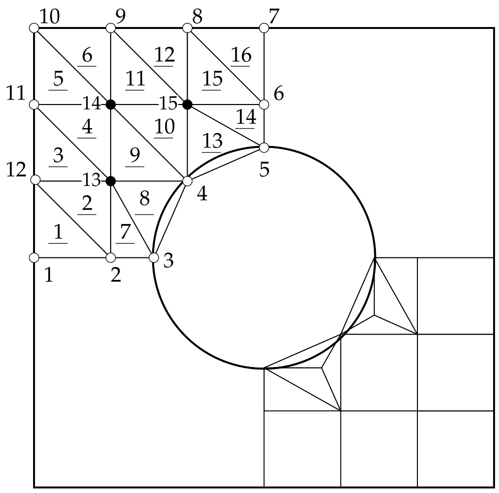


Fig. 10.12 Triangulation for the finite-element method in the geometry of a square with a cut-out circle. *Top left*: a coarse mesh of triangular elements with the enumeration of nodes and elements. *Bottom right*: in a part of the region we are allowed to use other shapes (e.g. squares) or refine the mesh in physically more interesting regions

In one dimension the finite elements are intervals; in a planar region the role of the elements is taken by various geometric shapes. Most frequently one uses (not necessarily equilateral or isosceles) triangles, rarely quadrilaterals (Fig. 10.12). The partitioning of the region is known as triangulation (or quadrangulation). For complex geometries (see Fig. 10.9) it almost amounts to art. Effective triangulation is accomplished by dedicated commercial programs (see e.g. [16]). Among the most well known is the Delaunay triangulation [17, 18]. A good triangulation guarantees that none of the interior angles in the triangles is too small (it maximizes the smallest used angles), thereby ensuring numerical stability.

On the chosen triangulation the corresponding basis functions are defined. Here we restrict the discussion to piecewise linear functions of x and y. The basis function on the nodes of the triangle $\boldsymbol{x}_j = (x_j, y_j)^{\mathrm{T}}$, $\boldsymbol{x}_{j+1} = (x_{j+1}, y_{j+1})^{\mathrm{T}}$, and $\boldsymbol{x}_{j+2} = (x_{j+2}, y_{j+2})^{\mathrm{T}}$, with an apex at $\boldsymbol{x}_j$ (Fig. 10.13(a)) has the form

$$\phi_j(x,y) = \left[\det\begin{pmatrix} 1 & x_j & y_j \\ 1 & x_{j+1} & y_{j+1} \\ 1 & x_{j+2} & y_{j+2} \end{pmatrix}\right]^{-1} \cdot \det\begin{pmatrix} 1 & x & y \\ 1 & x_{j+1} & y_{j+1} \\ 1 & x_{j+2} & y_{j+2} \end{pmatrix}. \tag{10.71}$$

The function ϕ_j is non-zero only over the triangle defined by the nodes $\boldsymbol{x}_j$, $\boldsymbol{x}_{j+1}$, and $\boldsymbol{x}_{j+2}$. It also holds that

$$\phi_j(x_k, y_k) = \delta_{j,k}.$$

Note that (10.71) expresses only the intersection of the planes defining the three side surfaces of the pyramid. For each pair of x and y one needs to check whether they lie inside the triangle defined by the three nodes.

Assembly When assembling the elements of local stiffness and mass matrices in the corresponding global matrices, one must distinguish between the local indices

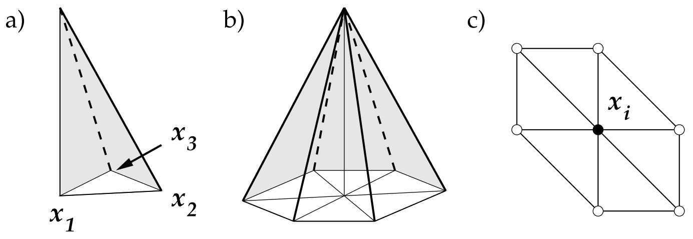


Fig. 10.13 Three-node element functions for the finite-element method on the plane: (**a**) the basic function above the triangular element with the value 1 at the node $\boldsymbol{x}_1$ and values 0 at the side nodes $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$; (**b**) the basis function above six elements; (**c**) the neighborhood of the node $\boldsymbol{x}_i$ as used in Problem 10.9.7

of nodes $m, n \in \{1, 2, 3\}$, and the global indices of nodes $j, k \in \{1, 2, \ldots, N_{\mathcal{N}}\}$ and elements $t \in \{1, 2, \ldots, N_{\mathcal{T}}\}$. Global indices label the whole triangulation with $N_{\mathcal{N}}$ nodes and $N_{\mathcal{T}}$ elements (triangles). In the variational requirement we sum over all triangles,

$$\sum_{t=1}^{N_{\mathcal{T}}} A^{(t)}(\widetilde{w}, u) = \sum_{t=1}^{N_{\mathcal{T}}} \langle \widetilde{w}, f\rangle^{(t)},$$

where t is the label of the triangle T_t. The surface integrals

$$A^{(t)}(\widetilde{w}, u) = \int_{T_t} \left[p(\nabla\widehat{w})^{\mathrm{T}} \cdot \nabla u + q\widetilde{w}u\right] \mathrm{d}x\, \mathrm{d}y, \qquad \langle \widetilde{w}, f\rangle^{(t)} = \int_{T_t} \widetilde{w} f \,\mathrm{d}x\, \mathrm{d}y,$$

are computed on each triangle T_t separately. This is relatively easily done in the Galerkin form of the method with piecewise linear basis functions, where $\phi_j = \psi_j$, as we need only the expressions for the functions ϕ_j (10.71) and their derivatives

$$\nabla\phi_j(x, y) = \frac{1}{2|T|} \begin{pmatrix} y_{j+1} - y_{j+2} \\ x_{j+2} - x_{j+1} \end{pmatrix},$$

where $|T| = \frac{1}{2}|(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)|$ is the surface area of the triangle. The local contributions of triplets of nodes to the stiffness matrix and the corresponding components of the local load vector are then

$$A_{mn}^{(t)}(\widetilde{w}, u) = \int_{T_t} \left[p(\nabla\phi_m)^{\mathrm{T}}\nabla\phi_n + q\phi_m\phi_n\right] \mathrm{d}x\, \mathrm{d}y,$$

$$\langle \widetilde{w}, f\rangle_m^{(t)} = \int_{T_t} \phi_m f \,\mathrm{d}x\, \mathrm{d}y.$$

Example How local matrices and vectors are assembled in the global matrix and global vector is best explained by an example. To make the discussion easier, we set $p = 1$ and $q = 0$ (we are solving the Poisson equation $-\nabla^2 v = f$). In this case the surface integrals over the triangles T_t are simple,

$$A_{mn}^{(t)} = \frac{1}{4|T|}(y_{m+1} - y_{m+2}, x_{m+2} - x_{m+1}) \begin{pmatrix} y_{n+1} - y_{n+2} \\ x_{n+2} - x_{n+1} \end{pmatrix},$$

$$\langle \widetilde{w}, f \rangle_m^{(t)} \approx \frac{1}{6} \det \begin{pmatrix} x_{m+1} - x_m & x_{m+2} - x_m \\ y_{m+1} - y_m & y_{m+2} - y_m \end{pmatrix} f(x_\mathrm{T}, y_\mathrm{T}),$$

where $(x_\mathrm{T}, y_\mathrm{T})$ are the coordinates of the center of mass of the triangle T_t.

As an illustration of the method, the top left portion of Fig. 10.12 shows a coarse triangulation of a square region with a cut-out circle. This problem (adopted from [14]) is still manageable by classical difference methods, but we use it here to serve as a typical example. We arrange the nodes in the matrix

$$\mathcal{N} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 11 & 12 & 13 & 14 & 15 \\ 0 & 1 & 1.59 & 2 & 3 & \cdots & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1.41 & \cdots & 2 & 1 & 1 & 2 & 2 \end{pmatrix}^\mathrm{T}$$

that includes the triplets {label of node, x, y}: here we specify the actual locations of the nodes in the planar region. The matrix

$$\mathcal{T} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 12 & 13 & 14 & 15 & 16 \\ 0 & 1 & 10 & 10 & 9 & \cdots & 7 & 3 & 4 & 5 & 5 \\ 1 & 12 & 11 & 12 & 10 & \cdots & 8 & 4 & 5 & 7 & 6 \\ 11 & 11 & 12 & 13 & 13 & \cdots & 14 & 14 & 14 & 14 & 7 \end{pmatrix}^\mathrm{T}$$

contains the quadruplets {label of element, label of node 1, label 2, label 3} relating the elements to the global indices of their nodes. The connection between the space of physical coordinates and the space of elements must be provided in our own code by some mapping between the sets $\mathcal{N}$ and $\mathcal{T}$. It is recommendable that the main loop in the code runs over the triangles $t = 1, 2, \ldots, N_\mathcal{T}$, not over the nodes, which would also be possible. Namely, it turns out that fewer surface integrals (evaluated numerically in general) need to be computed by using the element loop than by looping over the nodes [14]. The components of the global stiffness matrix S and the global load vector $\boldsymbol{g}$ are obtained by summing

```
loop over $t = 1, 2, \ldots, N_\mathcal{T}$
        $S_{\mathcal{T}(t,m),\mathcal{T}(t,n)} \mathrel{+}= A_{mn}^{(t)}$        $(m, n = 1, 2, 3)$
        $g_{\mathcal{T}(t,m)} \mathrel{+}= \langle \widetilde{w}, f \rangle_m^{(t)}$        $(m = 1, 2, 3)$
end loop
```

By solving $S\boldsymbol{c} = \boldsymbol{g}$ (as in the one-dimensional case) we finally obtain the vector $\boldsymbol{c}$ containing the expansion coefficients of the solution, $u(x, y) = \sum_j c_j \phi_j(x, y)$.

Try not to confuse the labellings of the nodes and the elements, since in general the number of nodes $N_\mathcal{N}$ is different from the number of elements $N_\mathcal{T}$. While the program loops over the elements t, the values are inserted in the global matrix, and the corresponding load vector according to the labeling of nodes. Compare Fig. 10.13(c) to Fig. 10.17: around the node $\boldsymbol{x}_i$ (in global enumeration) six triangles are arranged, and they form the support for the basis function shown in Fig. 10.13(b). Check your understanding in Problem 10.9.7!

We have merely traced the very first steps of FEM. A genuine expert use of the method only just starts here: non-uniform planar or spatial meshes are devised; basis functions are realized as splines of different degrees so that specific requirements at the boundaries between the elements are met; the mesh can be made denser during the computation if an increase in local precision is called for; discontinuities may be incorporated; non-stationary problems can be implemented; a road opens towards non-linear problems with a multitude of initial and boundary conditions. The finite-element method is already a part of modern numerical packages like MATLAB: nice pedagogical examples that can serve as a good vantage point for further study are given in [19, 20]. A myriad of commercial program packages is available for a more demanding use of the finite-element method [21]. For an excitingly propulsive free version, see [22].

10.7 Mimetic Discretizations ⋆

A powerful tool for solving PDE in complex geometries are the *mimetic (or compatible) discretizations* [23] that attempt to mimic the properties of the physical problem as closely as possible. A nice example is the diffusion in strongly heterogeneous and non-isotropic media described by the equation $v_t = \nabla \cdot D(v)\nabla v + Q$. The space is divided in "logically rectangular" convex cells onto which scalar and vector fields are attached (Fig. 10.14 (left)).

But the key step is the discretization of differential operators, like the diffusion operator $\nabla \cdot D(v)\nabla$ for the problem mentioned above [27, 28]. The discretization should, at least to some order, respect the symmetry and conservative properties of the underlying equation, for example, the conservation of mass, momentum, or energy (in studies of fluid flows) or Maxwell's equations (in electro-magnetic problems). Ultimately, the problems are translated to systems of algebraic equations and solved by preconditioned matrix relaxation methods.

A detailed presentation is beyond the scope of this book, but we let them lurk at the horizon due to their flexibility and development in the recent years. An introduction is given in [23] and the mathematical background in [29].

10.8 Multi-Grid and Mesh-Free Methods ⋆

Two further unique approaches to solving PDE in complex geometries should be mentioned. In the *multi-grid* approach the differential problem is discretized on