Chapter 5 Finite-Element Methods

The finite-difference approach with equidistant grids is easy to understand and straightforward to implement. The resulting uniform rectangular grids are comfortable, but in many applications not flexible enough. Steep gradients of the solution require locally a finer grid such that the difference quotients provide good approximations of the differentials. On the other hand, a flat gradient may be well modeled on a coarse grid. Such a flexibility of the grid is hard to obtain with finite-difference methods.

An alternative type of methods for solving PDEs that does provide the desired flexibility is the class of finite-element methods. A "finite element" (FE) designates a mathematical topic such as an interval and defined thereupon a piece of function. There are alternative names as *variational methods*, or *weighted residuals*, or *Galerkin methods*. These names hint at underlying principles that serve to derive suitable equations. As these different names suggest, there are several different approaches leading to finite elements. The methods are closely related.

The flexibility of finite-element methods is not only favorable to approximate functions, but also to approximate domains of computation that are not rectangular. This is important in higher-dimensional spaces. For the one-dimensional situation of standard options, the possible improvement of a finite-element method over the standard methods of the previous chapter is not that significant. With the focus on standard options, Chapter 5 may be skipped on first reading. But options with several underlyings naturally lead to domains of computation that may be more "fancy." This will be illustrated by the example in Section 5.4. In such situations, finite elements are ideally applicable and highly recommendable.

Faced with the huge field of finite-element methods, in this chapter we confine ourselves to a brief overview on several approaches and ideas (in Section 5.1). Then in Section 5.2, we describe the approximation with the simplest finite elements, namely, piecewise straight-line segments. These approaches will be applied to the calculation of standard options in Section 5.3. Section 5.4 will present an application to an exotic option with two underlyings. Finally, in Section 5.5, we will introduce into error estimates. Methods more subtle than just the Taylor expansion of the discretization error are required to show that quadratic convergence is possible with unstructured grids

R.U. Seydel, *Tools for Computational Finance*, Universitext,
DOI: 10.1007/978-3-540-92929-1_5,

and nonsmooth solutions. To keep this exposition short, many of the ideas will be explained for the one-dimensional situation. But the ideas extend to multidimensional scenarios.

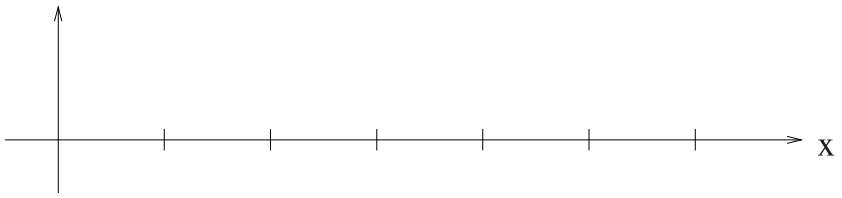

Fig. 5.1. Discretization of a continuum

5.1 Weighted Residuals

Many of the principles on which finite-element methods are based, can be interpreted as weighted residuals. What does this mean? This heading points at ways in which a discretization can be set up, and how an approximation can be defined. There lies a duality in a discretization. This is illustrated by means of Figure 5.1, which shows a partition of an x-axis. This discretization is either represented by

(a) discrete grid points x_i, or by
(b) a set of subintervals.

The two ways to see a discretization lead to different approaches of constructing an approximation w. Let us illustrate this with the one-dimensional situation of Figure 5.2. An approximation w based on finite differences is founded on the grid points and primarily consists of discrete points (Figure 5.2a). Finite elements are founded on subdomains (intervals in Figure 5.2b) with piecewise defined functions, which are defined by suitable criteria and constitute a global approximation w. In a narrower sense, a finite element is a pair consisting of one piece of subdomain and the corresponding function defined thereupon, mostly a polynomial. Figure 5.2 reflects the respective basic approaches; in a second step the isolated points of a finite-difference calculation can well be extended to continuous piecewise functions by means of interpolation (⟶ Appendix C1).

A two-dimensional domain can be partitioned into triangles, for example, where w is again represented with piecewise polynomials. Figure 5.3 depicts the simplest such situation, namely, a triangle in an (x, y)-plane, and a piece of a linear function defined thereupon. Figure 5.7 below will provide an example how triangles easily fill a seemingly "irregular" domain.

As will be shown next, the approaches of finite-element methods use integrals. If done properly, integrals require less smoothness. This often matches applications better and adds to the flexibility of finite-element methods. The integrals can be derived in a natural way from minimum principles, or are

constructed artificially. Finite elements based on polynomials make the calculation of the integrals easy.

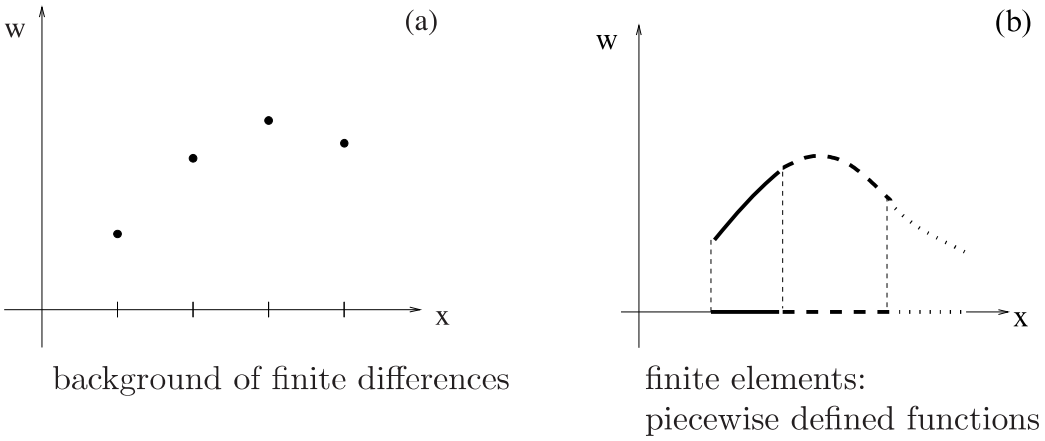

Fig. 5.2. Two kinds of approximations (one-dimensional situation)

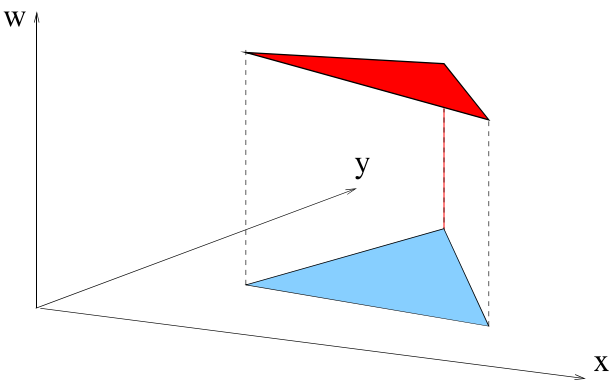

Fig. 5.3. A simple finite element in two dimensions, based on a triangle

5.1.1 The Principle of Weighted Residuals

To explain the principle of weighted residuals we discuss the formally simple case of the differential equation

$$Lu = f \ . \tag{5.1}$$

Here L symbolizes a linear differential operator. Important examples are

$$Lu := -u'' \quad \text{for} \quad u(x), \quad \text{or} \tag{5.2a}$$

$$Lu := -u_{xx} - u_{yy} \quad \text{for} \quad u(x,y) \ . \tag{5.2b}$$

Solutions u of the differential equation are studied on a domain $\mathcal{D} \subseteq \mathbb{R}^n$. The piecewise approach starts with a partition of the domain into a finite number of subdomains $\mathcal{D}_k$,

$$\mathcal{D} = \bigcup_k \mathcal{D}_k \ . \tag{5.3}$$

All boundaries should be included, and approximations to u are calculated on the closure $\bar{\mathcal{D}}$. The partition is assumed disjoint up to the boundaries of $\mathcal{D}_k$, so $\mathcal{D}_j^\circ \cap \mathcal{D}_k^\circ = \emptyset$ for $j \neq k$. In the one-dimensional case ($n = 1$), for example, the $\mathcal{D}_k$ are subintervals of a whole interval $\mathcal{D}$. In the two-dimensional case, (5.3) may describe a partition into triangles.

The ansatz for approximations w to a solution u is a basis representation,

$$w := \sum_{i=1}^{N} c_i \varphi_i \ . \tag{5.4}$$

In the case of one independent variable x the $c_i \in \mathbb{R}$ are constant coefficients, and the φ_i are functions of x. The φ_i are called **basis functions**, or *trial functions*. Typically the $\varphi_1, ..., \varphi_N$ are prescribed, whereas the free parameters $c_1, ..., c_N$ are to be determined such that $w \approx u$.

One strategy to determine the c_i is based on the residual function

$$R := Lw - f \ . \tag{5.5}$$

We look for a w such that R becomes "small." Since the φ_i are considered prescribed, in view of (5.4) N conditions or equations must be established to define and calculate the unknown $c_1, ..., c_N$. To this end we weight the residual by introducing N weighting functions *(test functions)* $\psi_1, ..., \psi_N$ and require

$$\boxed{\int_{\mathcal{D}} R\psi_j \, \mathrm{d}x = 0 \quad \text{for } j = 1, ..., N} \tag{5.6}$$

This amounts to the requirement that the residual be orthogonal to the set of weighting functions ψ_j. The "dx" in (5.6) symbolizes the integration that matches $\mathcal{D} \subseteq \mathbb{R}^n$; frequently it will be dropped. The system of equations (5.6) for the model problem (5.1) consists of the N equations

$$\int_{\mathcal{D}} Lw\psi_j = \int_{\mathcal{D}} f\psi_j \quad (j = 1, ..., N) \tag{5.7}$$

for the N unknowns $c_1, ..., c_N$, which are part of w. Often the equations in (5.7) are written using a formulation with inner products,

$$(Lw, \psi_j) = (f, \psi_j) \ ,$$

defined as the corresponding integrals in (5.7). For linear L the ansatz (5.4) implies

$$\int Lw\psi_j = \int \left(\sum_i c_i L\varphi_i \right) \psi_j = \sum_i c_i \underbrace{\int L\varphi_i \psi_j}_{=:a_{ij}} .$$

The integrals a_{ij} constitute a matrix A. The $r_j := \int f\psi_j$ set up a vector r and the coefficients c_j a vector $c = (c_1, ..., c_N)^{tr}$. This allows to rewrite the system of equations in vector notation as

$$Ac = r \; . \tag{5.8}$$

This outlines the general principle, but leaves open the questions how to handle boundary conditions and how to select the basis functions φ_i and the weighting functions ψ_j. The freedom to choose trial functions φ_i and test functions ψ_j allows to construct several different methods. For the time being suppose that these functions have sufficient potential to be differentiated or integrated. We will enter a discussion of relevant function spaces in Section 5.4.

5.1.2 Examples of Weighting Functions

We postpone the choice of basis functions φ_i and begin with listing important examples of how to select weighting functions ψ:

1.) **Galerkin method**, also Bubnov–Galerkin method:
Choose $\psi_j := \varphi_j$. Then $a_{ij} = \int L\varphi_i\varphi_j$

2.) **collocation:**
Choose $\psi_j := \delta(x - x_j)$. Here δ denotes Dirac's delta function, which in $\mathbb{R}^1$ satisfies $\int f\delta(x - x_j)\,\mathrm{d}x = f(x_j)$. As a consequence,

$$\int Lw\psi_j = Lw(x_j) \; ,$$

$$\int f\psi_j = f(x_j) \; .$$

That is, a system of equations $Lw(x_j) = f(x_j)$ results, which amounts to evaluating the differential equation at selected points x_j.

3.) **least squares:**
Choose

$$\psi_j := \frac{\partial R}{\partial c_j}$$

This choice of test functions deserves its name *least-squares*, because to minimize $\int (R(c_1, ..., c_N))^2$ the necessary criterion is the vanishing of the gradient, so

$$\int_{\mathcal{D}} R \frac{\partial R}{\partial c_j} = 0 \quad \text{for all } j \; .$$

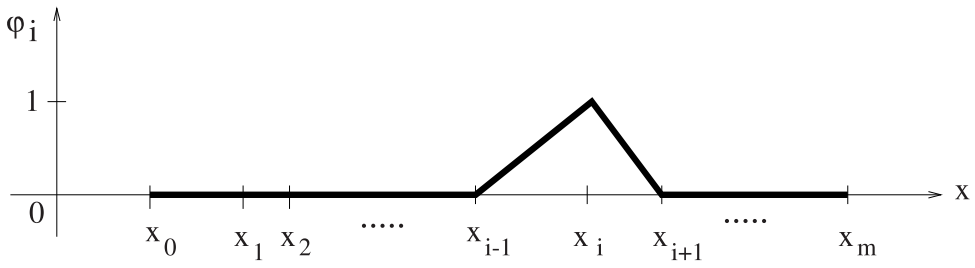

Fig. 5.4. "Hat function": simple choice of finite elements

5.1.3 Examples of Basis Functions

For the choice of suitable basis functions φ_i our concern will be to meet two aims: The resulting methods must be accurate, and their implementation should become efficient. We defer the aspect of accuracy to Section 5.5, and concentrate on the latter requirement, which can be focused on the sparsity of matrices. In particular, if the matrix A of the linear equations is sparse, then the system can be solved efficiently even when it is large. In order to achieve sparsity we require that $\varphi_i \equiv 0$ on most of the subdomains $\mathcal{D}_k$. Figure 5.4 illustrates an example for the one-dimensional case $n = 1$. This *hat function* of Figure 5.4 is the simplest example related to finite elements. It is piecewise linear, and each function φ_i has a support consisting of only two subintervals, $\varphi_i(x) \neq 0$ for $x \in$ support. A consequence is

$$\int_{\mathcal{D}} \varphi_i \varphi_j = 0 \quad \text{for } |i-j| > 1 \,, \tag{5.9}$$

as well as an analogous relation for $\int \varphi_i' \varphi_j'$. We will discuss hat functions in the following Section 5.2. More advanced basis functions are constructed using piecewise polynomials of higher degree. In this way, basis functions can be obtained with $\mathcal{C}^1$- or $\mathcal{C}^2$-smoothness ($\longrightarrow$ Exercise 5.1). Recall from interpolation ($\longrightarrow$ Appendix C1) that polynomials of degree three can lead to $\mathcal{C}^2$-smooth splines.

Remark on $Lu = -u''$, $\quad u, \varphi, \psi \in \{u : u(0) = u(1) = 0\}$:
Integration by parts implies formally

$$\int_0^1 \varphi'' \psi = -\int_0^1 \varphi' \psi' = \int_0^1 \varphi \psi'' \,,$$

because the boundary conditions $u(0) = u(1) = 0$ let the nonintegral terms vanish. These three versions of the integral can be distinguished by the smoothness requirements on φ and ψ, and by the question whether the integrals exist. One will choose the integral version that corresponds to the underlying method, and to the smoothness of the solution. For example, for Galerkin's approach the elements a_{ij} of A consist of the integrals

$$-\int_0^1 \varphi_i' \varphi_j' \ .$$

We will return to the topic of function spaces in Section 5.5 (with Appendix C3).

5.2 Galerkin Approach with Hat Functions

As mentioned before, any required flexibility is provided by finite-element methods. This holds to a larger extent in higher-dimensional spaces. In this section we stick to the one-dimensional situation, $x \in \mathbb{R}$.

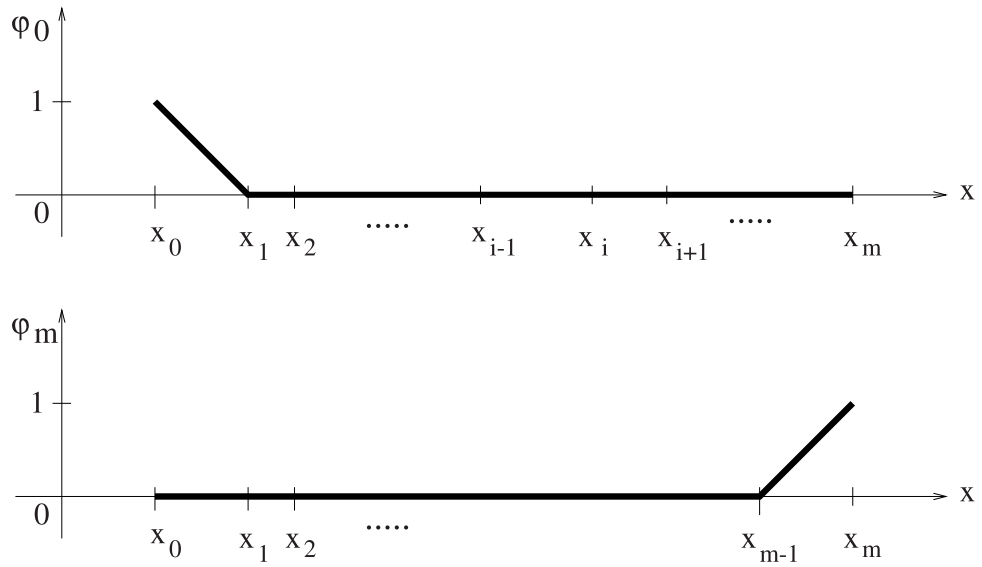

Fig. 5.5. Special "hat functions" φ_0 and φ_m

5.2.1 Hat Functions

We now explain the prototype of a finite-element method. This simple approach makes use of the hat functions, which we define formally (compare Figures 5.4 and 5.5).

Definition 5.1 (hat functions)
For $1 \le i \le m-1$ set

$$\varphi_i(x) := \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{for } x_{i-1} \le x < x_i \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i} & \text{for } x_i \le x < x_{i+1} \\ 0 & \text{elsewhere} \end{cases}$$

and for the boundary functions

$$\varphi_0(x) := \begin{cases} \dfrac{x_1 - x}{x_1 - x_0} & \text{for } x_0 \le x < x_1 \\ 0 & \text{elsewhere} \end{cases}$$

$$\varphi_m(x) := \begin{cases} \dfrac{x - x_{m-1}}{x_m - x_{m-1}} & \text{for } x_{m-1} \le x \le x_m \\ 0 & \text{elsewhere.} \end{cases}$$

These $m+1$ hat functions satisfy the following properties.

Properties 5.2 (hat functions)

(a) The $\varphi_0, ..., \varphi_m$ form a basis of the space of polygons

$$\{g \in \mathcal{C}^0[x_0, x_m] : g \text{ straight line on } \mathcal{D}_k := [x_k, x_{k+1}] \text{ for all } k = 0, ..., m-1\} .$$

That is to say, for each polygon v on $\mathcal{D}_0, ..., \mathcal{D}_{m-1}$ there are unique coefficients $c_0, ..., c_m$ with

$$v = \sum_{i=0}^{m} c_i \varphi_i .$$

(b) On $\mathcal{D}_k$ only φ_k and $\varphi_{k+1} \neq 0$ are nonzero. Hence

$$\varphi_i \varphi_k = 0 \quad \text{for} \quad |i-k| > 1 .$$

(c) A simple approximation of the integral $\int_{x_0}^{x_m} f\varphi_j \, \mathrm{d}x$ can be calculated as follows:
Substitute f by the interpolating polygon

$$f_{\mathrm{p}} := \sum_{i=0}^{m} f_i \varphi_i \quad \text{, where} \quad f_i := f(x_i) ,$$

and obtain for each j the approximating integral

$$I_j := \int_{x_0}^{x_m} f_{\mathrm{p}} \varphi_j \, \mathrm{d}x = \int_{x_0}^{x_m} \sum_{i=0}^{m} f_i \varphi_i \varphi_j \, \mathrm{d}x = \sum_{i=0}^{m} f_i \underbrace{\int_{x_0}^{x_m} \varphi_i \varphi_j \, \mathrm{d}x}_{=: b_{ji}}$$

The b_{ij} constitute a symmetric matrix B and the f_i a vector $\bar{f}$. If we arrange all integrals I_j $(0 \le j \le m)$ into a vector, then all integrals can be written in a compact way in vector notation as

$$B\bar{f} .$$

(d) The "large" $(m+1)^2$–matrix $B := (b_{ij})$ can be set up $\mathcal{D}_k$-elementwise by (2×2)-matrices (discussed below in Section 5.2.2). The (2×2)-matrices are

those integrals that integrate only over a single subdomain $\mathcal{D}_k$. For each $\mathcal{D}_k$ in our one-dimensional setting exactly the four integrals $\int \varphi_i \varphi_j \mathrm{d}x$ for $i, j \in \{k, k+1\}$ are nonzero. They can be arranged into a (2×2)-matrix

$$\int_{x_k}^{x_{k+1}} \begin{pmatrix} \varphi_k^2 & \varphi_k \varphi_{k+1} \\ \varphi_{k+1}\varphi_k & \varphi_{k+1}^2 \end{pmatrix} \mathrm{d}x \; .$$

(The integral over a matrix is understood elementwise.) These are the integrals on $\mathcal{D}_k$, where the integrand is a product of the factors

$$\frac{x_{k+1} - x}{x_{k+1} - x_k} \quad \text{and} \quad \frac{x - x_k}{x_{k+1} - x_k} \; .$$

The four numbers

$$\frac{1}{(x_{k+1} - x_k)^2} \int_{x_k}^{x_{k+1}} \begin{pmatrix} (x_{k+1} - x)^2 & (x_{k+1} - x)(x - x_k) \\ (x - x_k)(x_{k+1} - x) & (x - x_k)^2 \end{pmatrix} \mathrm{d}x$$

result. With $h_k := x_{k+1} - x_k$ integration yields the *element-mass matrix* ($\longrightarrow$ Exercise 5.2)

$$\frac{1}{6} h_k \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

(e) Analogously, integrating $\varphi_i' \varphi_j'$ yields

$$\begin{aligned} &\int_{x_k}^{x_{k+1}} \begin{pmatrix} \varphi_k'^2 & \varphi_k' \varphi_{k+1}' \\ \varphi_{k+1}' \varphi_k' & \varphi_{k+1}'^2 \end{pmatrix} \mathrm{d}x \\ &= \frac{1}{h_k^2} \int_{x_k}^{x_{k+1}} \begin{pmatrix} (-1)^2 & (-1)1 \\ 1(-1) & 1^2 \end{pmatrix} \mathrm{d}x = \frac{1}{h_k} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \; . \end{aligned}$$

These matrices are called *element-stiffness matrices*. They are used to set up the matrix A.

5.2.2 Assembling

The next step is to assemble the matrices A and B. It might to be tempting to organize this task as follows: Run a double loop on all i, j and check for each (i, j) on which $\mathcal{D}_k$ the integral

$$\int_{\mathcal{D}_k} \varphi_i \varphi_j = 0$$

is nonzero. It turns out that such a procedure is cumbersome as compared to the alternative of running a single loop on all k and calculate all relevant integrals on $\mathcal{D}_k$.

To this end, we split the integrals

$$\int_{x_0}^{x_m} = \sum_{k=0}^{m-1} \int_{\mathcal{D}_k}$$

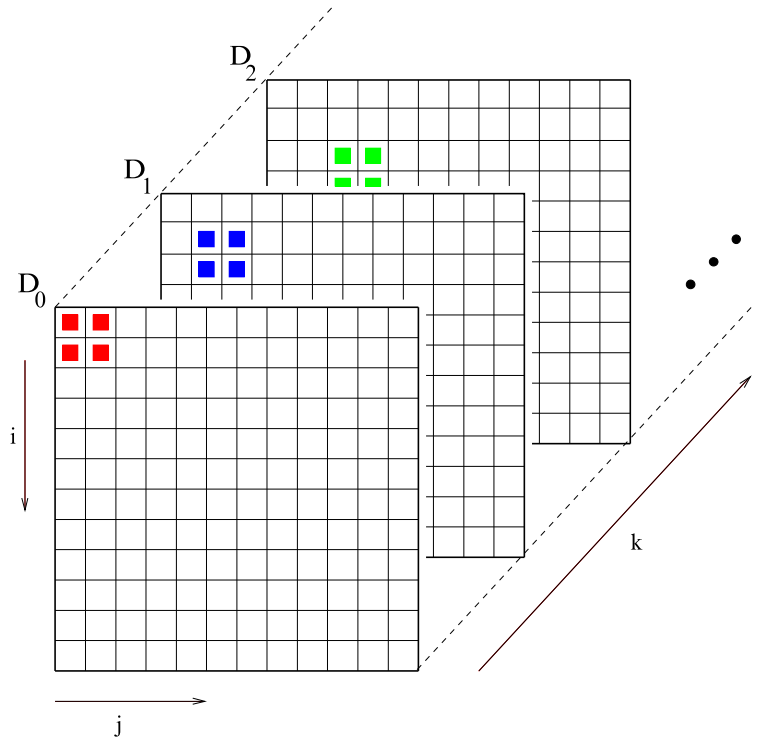

Fig. 5.6. Assembling in the one-dimensional setting

to construct the $(m+1)\times(m+1)$-matrices $A = (a_{ij})$ and $B = (b_{ij})$ *additively* out of the small element matrices. For the case of the one-dimensional hat functions with subintervals

$$\mathcal{D}_k = \{x \ : \ x_k \le x \le x_{k+1}\}$$

the element matrices are (2×2), see above. In this case only those integrals of $\varphi_i'\varphi_j'$ and $\varphi_i\varphi_j$ are nonzero, for which $i, j \in \mathcal{I}_k$, where

$$i, j \in \mathcal{I}_k := \{k, k+1\}\,. \tag{5.10}$$

$\mathcal{I}_k$ is the set of indices of those basis functions that are nonzero on $\mathcal{D}_k$. The *assembling algorithm* performs a loop over the subinterval index $k = 0, 1, \ldots, m-1$ and distributes the (2×2)-element matrices additively to the positions $(i, j) \in \mathcal{I}_k$. Before the assembling is started, the matrices A and B must be initialized with zeros. For $k = 0, ..., m-1$ one obtains for A the $(m+1)^2$-matrix

$$\begin{pmatrix} \frac{1}{h_0} & -\frac{1}{h_0} & & & \\ -\frac{1}{h_0} & \frac{1}{h_0}+\frac{1}{h_1} & -\frac{1}{h_1} & & \\ & -\frac{1}{h_1} & \frac{1}{h_1}+\frac{1}{h_2} & -\frac{1}{h_2} & \\ & & -\frac{1}{h_2} & \ddots & \ddots \\ & & & \ddots & \end{pmatrix} \tag{5.11a}$$

The matrix B is assembled in an analogous way. In the one-dimensional situation the matrices are tridiagonal. For an equidistant grid with $h = h_k$ this matrix A specializes to

$$A = \frac{1}{h}\begin{pmatrix} 1 & -1 & & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 2 & -1 \\ 0 & & & & -1 & 1 \end{pmatrix} \tag{5.11b}$$

and B to

$$B = \frac{h}{6}\begin{pmatrix} 2 & 1 & & & & 0 \\ 1 & 4 & 1 & & & \\ & 1 & 4 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 4 & 1 \\ 0 & & & & 1 & 2 \end{pmatrix} \tag{5.11c}$$

5.2.3 A Simple Application

In order to demonstrate the procedure, let us consider the simple model boundary-value problem

$$Lu := -u'' = f(x) \quad \text{with} \quad u(x_0) = u(x_m) = 0\,. \tag{5.12}$$

We perform a Galerkin approach and substitute $w := \sum_{i=0}^{m} c_i\varphi_i$ into the differential equation. In view of (5.7) this leads to

$$\sum_{i=0}^{m} c_i \int_{x_0}^{x_m} L\varphi_i\varphi_j \,\mathrm{d}x = \int_{x_0}^{x_m} f\varphi_j \,\mathrm{d}x\,.$$

Next we apply integration by parts on the left-hand side, and invoke Property 5.2(c) on the right-hand side. The resulting system of equations is

$$\sum_{i=0}^{m} c_i \underbrace{\int_{x_0}^{x_m} \varphi_i'\varphi_j' \,\mathrm{d}x}_{a_{ij}} = \sum_{i=0}^{m} f_i \underbrace{\int_{x_0}^{x_m} \varphi_i\varphi_j \,\mathrm{d}x}_{b_{ij}}, \quad j = 0, 1, ..., m\,. \tag{5.13}$$

This system is preliminary because the homogenous boundary conditions $u(x_0) = u(x_m) = 0$ are not yet taken into account.

At this state, the preliminary system of equations (5.13) can be written as

$$Ac = B\bar{f}\,. \tag{5.14}$$

It is easy to see that the matrix A from (5.11b) is singular, because $A(1,1,...,1)^{tr} = 0$. This singularity reflects the fact that the system (5.14) does not have a unique solution. This is consistent with the differential equation $-u'' = f(x)$: If $u(x)$ is solution, then also $u(x)+\alpha$ for arbitrary α. Unique solvability is attained by satisfying the boundary conditions; a solution u of $-u'' = f$ must be fixed by at least one essential boundary condition. For our example (5.12) we know in view of $u(x_0) = u(x_m) = 0$ the coefficients $c_0 = c_m = 0$. This information can be inserted into the system of equations in such a way that the matrix A changes to a nonsingular matrix without losing symmetry. For $c_0 = 0$ we replace the first equation of the system (5.14) by $(1,0,\ldots,0)^{tr}\, c = 0$. Adding the first equation to the second produces a zero in the first column of A. Analogously we realize $c_m = 0$ in the last row and column of A. Now the c_0 and c_m are decoupled, and the inner part of size $(m-1)\times(m-1)$ of A remains. The matrix B is $(m-1)\times(m+1)$. Finally, for the special case of an equidistant grid, the system of equations is

$$\begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ 0 & & & -1 & 2 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m-2} \\ c_{m-1} \end{pmatrix} = \frac{h^2}{6} \begin{pmatrix} 1 & 4 & 1 & & & 0 \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ 0 & & & & 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} \bar{f}_0 \\ \bar{f}_1 \\ \vdots \\ \bar{f}_{m-1} \\ \bar{f}_m \end{pmatrix} \tag{5.15}$$

In (5.15) we have used an equidistant grid for sake of a lucid exposition. Our main focus is the nonequidistant version, which is also implemented easily. In case nonhomogeneous boundary conditions are prescribed, appropriate values of c_0 or c_m are predefined. The importance of finite-element methods in structural engineering has lead to call the global matrix A the stiffness matrix, and B is called the mass matrix.

5.3 Application to Standard Options

The flexibility of finite elements is especially advantageous in higher-dimensional spaces (several underlyings). But it works also for the one-dimensional case of standard options. This is the theme of this section.

5.3.1 European Options

As emphasized earlier, the valuation of single-asset European options makes use of the Black–Scholes formula. But for sake of exposition, let us briefly sketch a finite-element approach. We apply the FE approach to the transformed version $y_\tau = y_{xx}$ of the Black–Scholes equation. The solution $y(x,\tau)$ is approximated by an ansatz that corresponds to (5.4), namely,

$$\sum_{i=1}^{N} w_i(\tau)\varphi_i(x) + \varphi_0(x,\tau)\ . \tag{5.16}$$

Here $\varphi_0(x,\tau)$ is constructed in advance such that φ_0 satisfies boundary conditions and —if possible— initial condition. So φ_0 can be considered to be known, and the sum $\sum w_i\varphi_i$ does not reflect any nonzero (Dirichlet-) boundary conditions. The basis functions $\varphi_1,\ldots,\varphi_N$ are chosen to be the hat functions, which incorporate the discretization of the x-axis. Hence, $N = m-1$, and x_0 corresponds to $x_{\min}$ and x_m to $x_{\max}$. The functions $w_1,\ldots,w_{m-1}$ are unknown. (5.16) represents a separation of the variables x and τ.

Calculating derivatives of (5.16) and substituting into $y_\tau = y_{xx}$ leads to the Galerkin approach

$$\int_{x_0}^{x_m} \left[\sum_{i=1}^{m-1} \dot{w}_i\varphi_i + \dot{\varphi}_0\right] \varphi_j \,\mathrm{d}x = \int_{x_0}^{x_m} \left[\sum_{i=1}^{m-1} w_i\varphi_i'' + \varphi_0''\right] \varphi_j \,\mathrm{d}x$$

for $j = 1,\ldots,m-1$. The overdot represents differentiation with respect to τ, and the prime with respect to x. Arranging the terms that invole derivatives of φ_0 into vectors $a(\tau)$, $b(\tau)$,

$$a(\tau) := \begin{pmatrix} \int \varphi_0''(x,\tau)\,\varphi_1(x)\,\mathrm{d}x \\ \vdots \\ \int \varphi_0''(x,\tau)\,\varphi_{m-1}(x)\,\mathrm{d}x \end{pmatrix}\ , \quad b(\tau) := \begin{pmatrix} \int \dot{\varphi}_0(x,\tau)\,\varphi_1(x)\,\mathrm{d}x \\ \vdots \\ \int \dot{\varphi}_0(x,\tau)\,\varphi_{m-1}(x)\,\mathrm{d}x \end{pmatrix}$$

and using the matrices A, B as in (5.11), we arrive after integration by parts at

$$B\dot{w} + b = -Aw - a \tag{5.17}$$

This completes the semidiscretization, and defines the unknown vector function $w(\tau) := (w_1,\ldots,w_{m-1})^{tr}$ as solution of a system of ordinary differential equations. Initial conditions for $\tau = 0$ are given by (5.16). Assume the initial condition as $y(x,0) = \alpha(x)$, then

$$\sum_{i=1}^{N} w_i(0)\varphi_i(x) + \varphi_0(x,0) = \alpha(x)\ .$$

Specifically for $x = x_j$ the sum reduces to $w_j(0)\cdot 1$, leading to

$$w_j(0) = \alpha(x_j) - \varphi_0(x_j, 0) \; .$$

We leave the derivation of a Crank–Nicolson type of discretization as an exercise to the reader. With the usual notation as in $w^{(\nu)} := w(t_\nu)$, the result can be written

$$\begin{aligned}(B + \frac{\Delta\tau}{2}A)\, w^{(\nu+1)} =& (B - \frac{\Delta\tau}{2}A)\, w^{(\nu)} \\ & - \frac{\Delta\tau}{2}\,(a^{(\nu)} + a^{(\nu+1)} + b^{(\nu)} + b^{(\nu+1)})\end{aligned} \tag{5.18}$$

The structure strongly resembles the finite-difference approach (4.15). This similarity suggests that the order is the same, because for the finite-element A's and B's we have (compare (5.11))

$$A = O\left(\frac{1}{\Delta x}\right) \quad , \quad B = O(\Delta x) \; .$$

The separation of the variables x and τ in (5.16) allows to investigate the orders of the discretizations separately. In $\Delta\tau$, the order $O(\Delta\tau^2)$ of the Crank–Nicolson type approach (5.18) is clear from the above. It remains to derive the order of convergence with respect to the discretization in x. Because of the separation of variables it is sufficient to derive the convergence for a one-dimensional model problem. This will be done in Section 5.5.

5.3.2 Variational Form of the Obstacle Problem

To warm up for the discussion of the American option, let us return to the simple obstacle problem of Section 4.5.4 with the obstacle function $g(x, \tau)$. This problem can be formulated as a variational inequality. The function u can be characterized by comparing it to functions v out of a set $\mathcal{K}$ of *competing functions*

$$\begin{aligned}\mathcal{K} := \{v \in \mathcal{C}^0[-1,1] :\; & v(-1) = v(1) = 0\,, \\ & v(x) \geq g(x) \;\text{ for }\; -1 \leq x \leq 1,\; v \text{ piecewise } \in \mathcal{C}^1\}\,.\end{aligned}$$

The requirements on u imply $u \in \mathcal{K}$. For $v \in \mathcal{K}$ we have $v - g \geq 0$ and in view of $-u'' \geq 0$ also $-u''(v - g) \geq 0$. Hence for all $v \in \mathcal{K}$ the inequality

$$\int_{-1}^{1} -u''(v - g)\,\mathrm{d}x \geq 0$$

must hold. By (4.26) the integral

$$\int_{-1}^{1} -u''(u - g)\,\mathrm{d}x = 0$$

vanishes. Subtracting yields

$$\int_{-1}^{1} -u''(v-u)\,\mathrm{d}x \geq 0 \quad \text{for any } v \in \mathcal{K}\,.$$

The obstacle function g does not occur explicitly in this formulation; the obstacle is implicitly defined in $\mathcal{K}$. Integration by parts leads to

$$\underbrace{[-u'(v-u)]_{-1}^{1}}_{=0} + \int_{-1}^{1} u'(v-u)'\,\mathrm{d}x \geq 0\,.$$

The integral-free term vanishes because of $u(-1) = v(-1),\ u(1) = v(1)$. In summary, we have derived the statement:

$$\boxed{\begin{array}{c}\text{If } u \text{ solves the obstacle problem (4.26), then} \\ \displaystyle\int_{-1}^{1} u'(v-u)'\,\mathrm{d}x \geq 0 \quad \text{for all } v \in \mathcal{K}\,.\end{array}} \tag{5.19}$$

Since v varies in the set $\mathcal{K}$ of competing functions, an inequality such as in (5.19) is called *variational inequality*. The characterization of u by (5.19) can be used to construct an approximation w: Instead of u, find a $w \in \mathcal{K}$ such that the inequality (5.19) is satisfied for all $v \in \mathcal{K}$,

$$\int_{-1}^{1} w'(v-w)'\,\mathrm{d}x \geq 0 \quad \text{for all } v \in \mathcal{K}$$

The characterization (5.19) is related to a minimum problem, because the integral vanishes for $v = u$.

5.3.3 American Options

Analogously as the simple obstacle problem also the problem of calculating American options can be formulated as variational problem, compare Problem 4.7. The class of comparison functions is defined as

$$\begin{aligned}\mathcal{K} := \{v \in \mathcal{C}^0 : \tfrac{\partial v}{\partial x} \ \text{ piecewise } \mathcal{C}^0\ , \\ v(x,\tau) \geq g(x,\tau) \ \text{ for all } x,\tau\ ,\ v(x,0) = g(x,0)\ , \\ v(x_{\max},\tau) = g(x_{\max},\tau),\ v(x_{\min},\tau) = g(x_{\min},\tau)\}\ .\end{aligned} \tag{5.20}$$

For the following, $v \in \mathcal{K}$. Let y denote the exact solution of Problem 4.7. As solution of the partial differential inequality, y is $\mathcal{C}^2$-smooth on the continuation region, and $y \in \mathcal{K}$. From

$$v \geq g, \quad \frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \geq 0$$

we deduce

$$\int_{x_{\min}}^{x_{\max}} \left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \right) (v - g) \, \mathrm{d}x \geq 0 \, .$$

Invoking the complementarity

$$\int_{x_{\min}}^{x_{\max}} \left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \right) (y - g) \, \mathrm{d}x = 0$$

and subtraction gives

$$\int_{x_{\min}}^{x_{\max}} \left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \right) (v - y) \, \mathrm{d}x \geq 0 \, .$$

Integration by parts leads to the inequality

$$\int_{x_{\min}}^{x_{\max}} \left(\frac{\partial y}{\partial \tau}(v - y) + \frac{\partial y}{\partial x} \left(\frac{\partial v}{\partial x} - \frac{\partial y}{\partial x} \right) \right) \mathrm{d}x - \frac{\partial y}{\partial x}(v - y) \Bigg|_{x_{\min}}^{x_{\max}} \geq 0 \, .$$

The nonintegral term vanishes, because at the boundary for $x_{\min}$, $x_{\max}$, in view of $v = g$, $y = g$ the equality $v = y$ holds. The final result is

$$I(y; v) := \int_{x_{\min}}^{x_{\max}} \left(\frac{\partial y}{\partial \tau} \cdot (v - y) + \frac{\partial y}{\partial x} \left(\frac{\partial v}{\partial x} - \frac{\partial y}{\partial x} \right) \right) \mathrm{d}x \geq 0 \quad \text{for all } v \in \mathcal{K} \, . \tag{5.21}$$

The exact y is characterized by the fact that the inequality (5.21) holds for all comparison functions $v \in \mathcal{K}$. For the special choice $v = y$ the integral takes its minimal value,

$$\min_{v \in \mathcal{K}} I(y; v) = I(y; y) = 0 \, .$$

A more general question is, whether the inequality (5.21) holds for a $\widehat{y} \in \mathcal{K}$ that is not $\mathcal{C}^2$-smooth on the continuation region. (Recall that the American option is widely $\mathcal{C}^2$-smooth, except across the early-exercise curve.) The aim is to construct a $\widehat{y} \in \mathcal{K}$ such that $I(\widehat{y}; v) \geq 0$ for all $v \in \mathcal{K}$, and

$$\inf_{v \in \mathcal{K}} I(\widehat{y}; v) = 0 \, .$$

This formulation of our problem is called *weak version*, because it does *not* use $\widehat{y} \in \mathcal{C}^2$. Solutions $\widehat{y}$ of this minimization problem, which are globally continuous but only piecewise $\in \mathcal{C}^1$ are called *weak solutions*. The original partial differential equation requires $y \in \mathcal{C}^2$ and hence more smoothness. Such $\mathcal{C}^2$-solutions are called *strong solutions* or *classical solutions* (⟶ Section 5.5).

Now we approach the inequality (5.21) with finite-element methods. As a first step to approximately solve the minimum problem, assume approximations for $\widehat{y}$ and v in the similar forms

$$\begin{aligned} &\sum_i w_i(\tau)\varphi_i(x) \quad \text{for } \hat{y}\ , \\ &\sum_i v_i(\tau)\varphi_i(x) \quad \text{for } v\ . \end{aligned} \tag{5.22}$$

The reduced smoothness of these expressions match the requirements of $\mathcal{K}$. The above setting assumes the independent variables τ and x to be separated. As a consequence of this simple approach, the same x-grid is applied for all τ, which results in a rectangular grid in the (x,τ)-plane. The time dependence is incorporated in the coefficient functions w_i and v_i. Since the basis functions φ_i represent the x_i-grid, we so far perform a semidiscretization. Plugging into (5.21) gives

$$\begin{aligned} &\int \left\{ \left(\sum_i \frac{\mathrm{d}w_i}{\mathrm{d}\tau}\varphi_i \right) \left(\sum_j (v_j - w_j)\varphi_j \right) + \right. \\ &\qquad \left. \left(\sum_i w_i \varphi_i' \right) \left(\sum_j (v_j - w_j)\varphi_j' \right) \right\} \mathrm{d}x \\ &= \sum_i \sum_j \frac{\mathrm{d}w_i}{\mathrm{d}\tau}(v_j - w_j) \int \varphi_i \varphi_j \,\mathrm{d}x + \sum_i \sum_j w_i (v_j - w_j) \int \varphi_i' \varphi_j' \,\mathrm{d}x \geq 0. \end{aligned}$$

Translated into vector notation this is equivalent to

$$\left(\frac{\mathrm{d}w}{\mathrm{d}\tau} \right)^{\mathrm{tr}} B(v-w) + w^{\mathrm{tr}} A(v-w) \geq 0$$

or

$$(v-w)^{\mathrm{tr}} \left(B\frac{\mathrm{d}w}{\mathrm{d}\tau} + Aw \right) \geq 0\ .$$

The matrices A and B are defined via the assembling described above; for equidistant steps the special versions in (5.11b), (5.11c) arise.

As a second step, the time is discretized. To this end let us define the vectors

$$w^{(\nu)} := w(\tau_\nu), \quad v^{(\nu)} := v(\tau_\nu)\ .$$

Upon substituting, and θ-averaging the Aw term as in Section 4.6.1, we arrive at the inequalities

$$\left(v^{(\nu+1)} - w^{(\nu+1)} \right)^{\mathrm{tr}} \left(B\frac{1}{\Delta\tau}(w^{(\nu+1)} - w^{(\nu)}) + \theta A w^{(\nu+1)} + (1-\theta) A w^{(\nu)} \right) \geq 0 \tag{5.23a}$$

for all ν. For $\theta = 1/2$ this is a Crank–Nicolson-type method.

Rearranging (5.23a) leads to

$$\left(v^{(\nu+1)} - w^{(\nu+1)} \right)^{\mathrm{tr}} \left((B + \Delta\tau\,\theta A)\, w^{(\nu+1)} + (\Delta\tau(1-\theta)A - B)\, w^{(\nu)} \right) \geq 0\ .$$

With the abbreviations

$$\begin{aligned} r &:= (B - \Delta\tau(1-\theta)A)\, w^{(\nu)} \\ C &:= B + \Delta\tau\,\theta A \end{aligned} \tag{5.23b}$$

the inequality can be rewritten as

$$\left(v^{(\nu+1)} - w^{(\nu+1)}\right)^{tr} \left(Cw^{(\nu+1)} - r\right) \geq 0\ . \tag{5.23c}$$

This is the fully discretized version of $I(\widehat{y}; v) \geq 0$.

Side Conditions

$\widehat{y}(x,\tau) \geq g(x,\tau)$ amounts to

$$\sum w_i(\tau)\varphi_i(x) \geq g(x,\tau)\ .$$

For hat functions φ_i (with $\varphi_i(x_i) = 1$ and $\varphi_i(x_j) = 0$ for $j \neq i$) and $x = x_j$ this implies $w_j(\tau) \geq g(x_j,\tau)$. With $\tau = \tau_\nu$ we have

$$w^{(\nu)} \geq g^{(\nu)}; \quad \text{analogously } v^{(\nu)} \geq g^{(\nu)}\ .$$

For each time level ν we must find a solution that satisfies both the inequality (5.23) and the side condition

$$w^{(\nu+1)} \geq g^{(\nu+1)} \ \text{ for all } \ v^{(\nu+1)} \geq g^{(\nu+1)}\ .$$

In summary, the algorithm is

Algorithm 5.3 (finite elements for American standard options)

$\theta := 1/2$. Calculate $w^{(0)}$.
For $\nu = 1, ..., \nu_{\max}$:
Calculate $r = (B - \Delta\tau(1-\theta)A)w^{(\nu-1)}$ and $g = g^{(\nu)}$
Construct a w such that for all $v \geq g$
$\quad (v-w)^{tr}(Cw - r) \geq 0, \quad w \geq g.$
Set $w^{(\nu)} := w$

Let us emphasize again the main step, which is the kernel of this algorithm and the main labor: Construct w such that

(FE)

for all $v \geq g$
$(v-w)^{tr}(Cw - r) \geq 0, \quad w \geq g$

(5.24)

This task (FE) can be reformulated into a task we already solved in Section 4.6. To this end recall the finite-difference equation (4.31), replacing A by C, and b by r. There the following holds for w:

$$\text{(FD)} \qquad \boxed{\begin{array}{l} Cw - r \geq 0, \quad w \geq g \\ (Cw - r)^{tr}(w - g) = 0 \end{array}} \tag{5.25}$$

Theorem 5.4 (equivalence)

The solution of the problem (FE) is equivalent to the solution of problem (FD).

Proof:

a) (FD) $\implies$ (FE):
Let w solve (FD), so $w \geq g$, and

$$(v - w)^{tr}(Cw - r) = (v - g)^{tr} \underbrace{(Cw - r)}_{\geq 0} - \underbrace{(w - g)^{tr}(Cw - r)}_{=0}$$

hence $(v - w)^{tr}(Cw - r) \geq 0$ for all $v \geq g$

b) (FE) $\implies$ (FD):
Let w solve (FE), so $w \geq g$, and

$$v^{tr}(Cw - r) \geq w^{tr}(Cw - r) \quad \text{for all } v \in \mathcal{K}$$

Suppose the kth component of $Cw - r$ is negative, and make v_k arbitrarily large. Then the left-hand side becomes arbitrarily small, which is a contradiction. So $Cw - r \geq 0$. Now

$$w \geq g \implies (w - g)^{tr}(Cw - r) \geq 0$$

Set in (FE) $v = g$, then $(w - g)^{tr}(Cw - r) \leq 0$.
Therefore $(w - g)^{tr}(Cw - r) = 0$.

Implementation

As a consequence of this equivalence, the solution of the finite-element problem (FE) can be calculated with the methods we applied to solve problem (FD) in Section 4.6. Following the exposition in Section 4.6.2, the kernel of the finite-element Algorithm 5.3 can be written as follows

$$(\mathbf{FE}') \qquad \boxed{\begin{array}{c} \text{Solve } Cw = r \text{ such that} \\ \text{componentwise } w \geq g\ . \end{array}}$$

The vector v is not calculated. The boundary conditions on w are set up in the same way as discussed in Section 4.4 and summarized in Algorithm 4.13.

Consequently, the finite-element algorithm parallels Algorithm 4.13 closely in the special case of an equidistant x-grid; there is no need to repeat this algorithm ($\longrightarrow$ Exercise 5.3). In the general nonequidistant case, the off-diagonal and the diagonal elements of the tridiagonal matrix C vary with i, and the formulation of the SOR-loop gets more involved. The details of the implementation are technical and omitted. The Algorithm 4.14 is the same in the finite-element case.

The computational results match those of Chapter 4 and need not be repeated. The costs of the presented simple version of a finite-element approach are slightly lower than that of the finite-difference approach.

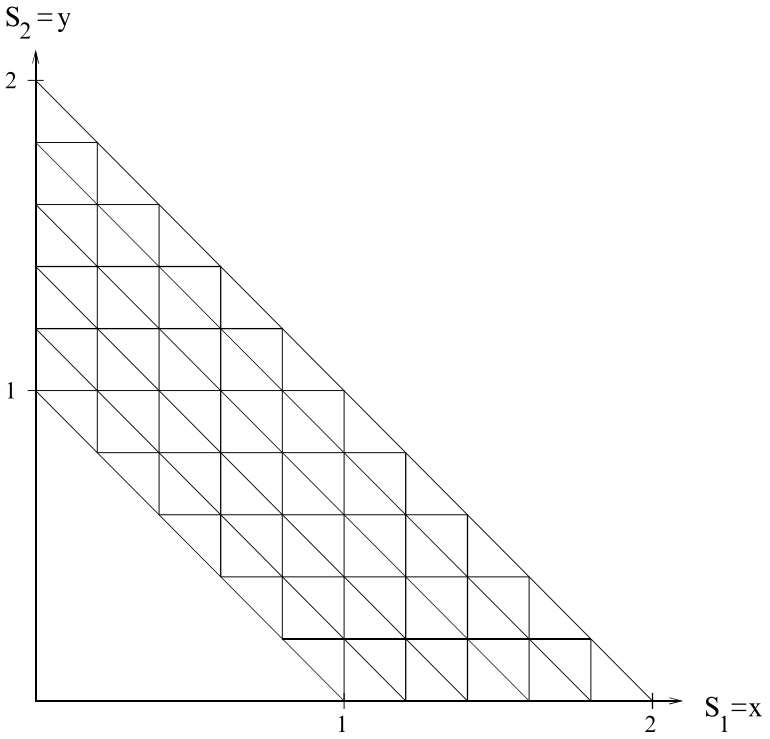

Fig. 5.7. Finite element discretization of a domain $\mathcal{D}$ into triangles $\mathcal{D}_k$ (see Section 5.4)

5.4 Application to an Exotic Call Option

As an example we consider an exotic European-style option, a two-asset basket-double-barrier call option with payoff

$$\Psi(S_1, S_2) = (S_1 + S_2 - K)^+ ,$$

and $V(S_1, S_2, T) = \Psi(S_1, S_2)$, up to the barriers. Assume two knock-out barriers B_1 and B_2, down-and-out with B_1, up-and-out with B_2. That is, the option ceases to exist when $S_1 + S_2 < B_1$, or when $S_1 + S_2 > B_2$; in both

cases $V = 0$. The mathematical model is that of the Black–Scholes market, see Section 3.5.5. The corresponding PDE for the value function $V(S_1, S_2, t)$ is

$$\begin{aligned} &\frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + rS_1 \frac{\partial V}{\partial S_1} - rV \\ &\quad + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + rS_2 \frac{\partial V}{\partial S_2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} = 0 \,. \end{aligned} \tag{5.26}$$

(For the general case see Section 6.2.) The computational domain $\mathcal{D}$ is bounded by the two lines $S_1 + S_2 = B_1$ and $S_1 + S_2 = B_2$. This shape of $\mathcal{D}$ naturally suggests applying a structured grid of triangular elements $\mathcal{D}_k$. One possible triangulation is sketched in Figure 5.7. For this example we choose the parameters

$$K = 1\,,\ T = 1\,,\ \sigma_1 = \sigma_2 = 0.25\,,\ \rho = 0.7\,,\ r = 0.05\,,\ B_1 = 1\,,\ B_2 = 2\,.$$

The boundary conditions for $S_1 \to 0$ and $S_2 \to 0$ are given by the one-dimensional Black–Scholes equation; just set either $S_1 = 0$ or $S_2 = 0$ in (5.26). Hence the boundary conditions for (5.26) are the values of single-asset double-barrier options and can be evaluated by a closed-form formula, see [Haug98].

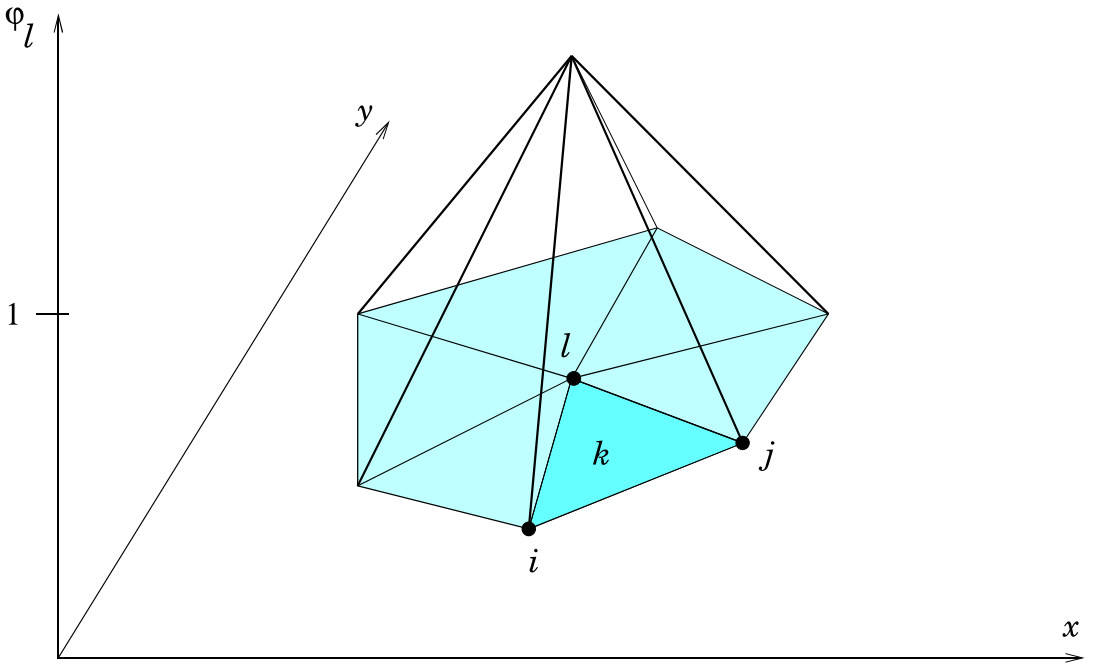

Fig. 5.8. Two-dimensional hat function $\varphi_l(x, y)$ (zero ouside the shaded structure)

It is convenient to solve the Black–Scholes equation in a divergence-free version. To this end, use standard PDE variables $x := S_1$, $y := S_2$, $\tau := T - t$ for the independent variables, and $u(x, y, \tau)$ for the dependent variable, and derive the PDE for u

$$-\nabla \cdot (D(x,y)\nabla u) + b(x,y)\nabla u + ru = -\frac{\partial}{\partial \tau} u \;, \tag{5.27a}$$

where the $\cdot$ corresponds to the scalar product, similar as ^{tr} for vectors. ∇u is the gradient of u. This makes use of

$$\begin{aligned} D(x,y) &:= \frac{1}{2} \begin{pmatrix} \sigma_1^2 x^2 & \rho\sigma_1\sigma_2 xy \\ \rho\sigma_1\sigma_2 xy & \sigma_2^2 y^2 \end{pmatrix} , \\ b(x,y) &:= - \begin{pmatrix} (r - \sigma_1^2 - \rho\sigma_1\sigma_2/2)\, x \\ (r - \sigma_2^2 - \rho\sigma_1\sigma_2/2)\, y \end{pmatrix} , \\ \nabla &:= \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix} . \end{aligned} \tag{5.27b}$$

The reader is invited to check the equivalence with (5.26). ($\longrightarrow$ Exercise 5.5)

To separate time τ and "space" (x, y), substitute u by the ansatz

$$\sum_i w_i(\tau)\, \varphi_i(x,y) \;.$$

Compared to (5.22) the basis functions φ_i are defined on planar regions $\mathcal{D} \subset \mathbb{R}^2$. The Galerkin ansatz creates integrals over $\mathcal{D}$

$$\int \varphi_i \, \nabla \cdot D \nabla \varphi_j \;, \quad \int \varphi_i \, b^{tr} \nabla \varphi_j \;, \quad \int \varphi_i \, r \, \varphi_j \;.$$

For basis functions, we choose the two-dimensional analogon of the hat functions, which matches perfectly triangular elements. The situation is shown schematically in Figure 5.8. There the central node l is node of several adjacent triangles, which are the support (shaded) on which φ_l is built by planar pieces. This approach defines a tent-like hat function φ_l, which is zero "outside." By linear combination of such basis functions, piecewise planar surfaces above the computational domain can be constructed. Locally, for one triangle, this may look like the element in Figure 5.3.

In this two-dimensional situation, the element matrices are 3×3. For each number k of a triangle, there are three nodes of the triangle, i, j, l in Figure 5.8. Hence the table that assigns nodes to triangles includes the entry $\mathcal{I}_k := \{i, j, l\}$. Accordingly, for each matrix, the assembling loop distributes 9 local integrals for each $\mathcal{D}_k$. For the calculation of the local integrals on an arbitrary triangle $\mathcal{D}_k$ consult the special FE literature. Basic ingredients are the relations in Exercise 5.6. The Figure 5.9[1] shows a FE solution with 192 triangles.

[1] Courtesy of A. Kvetnaia.