Chapter 4
Standard Methods for Standard Options

Now we enter the part of the book that is devoted to the numerical solution of equations of the Black–Scholes type. In this chapter, we discuss "standard" options in the sense as introduced in Sect. 1.1 and assume the scenario characterized by the Assumptions 1.2. In case of European vanilla options the value function $V(S,t)$ solves the Black–Scholes equation (1.5). It is not really our aim to solve this partial differential equation for vanilla payoff because it possesses an analytic solution ($\longrightarrow$ Appendix A.4). Ultimately our intention is to solve more general equations and inequalities. In particular, American options will be calculated numerically. But also European options without vanilla payoff are of interest; we encounter them for Bermudan options in Sect. 1.8.4, and for Asian options in Sect. 6.3.4. The goal is not only to calculate single values $V(S_0, 0)$—for this purpose tree methods can be applied—but also to approximate the curve $V(S, 0)$, or even the surface defined by the value function $V(S,t)$ on the half strip $S > 0$, $0 \leq t \leq T$. Based on the surface of the value function, we collect information on early exercise and on the greeks (1.25), for example, on delta hedging by observing the derivative $\frac{\partial V}{\partial S}$.

American options obey *inequalities* of the type of the Black–Scholes equation (1.5). To allow for early exercise, the Assumptions 1.2 must be weakened. As a further generalization, the payment of dividends must be taken into account; otherwise early exercise does not make sense for American calls.

The main part of this chapter outlines a PDE approach based on finite differences. We begin with unrealistically simplified boundary conditions in order to keep the explanation of the discretization schemes transparent. Later sections will discuss appropriate boundary conditions, which turn out to be tricky in the case of American options. At the end of this chapter we will be able to implement a finite-difference algorithm for standard American (and European) options. Note that this approach assumes constant coefficients of the Black-Scholes equation. If we work carefully,

R.U. Seydel, *Tools for Computational Finance*, Universitext,
DOI 10.1007/978-1-4471-7338-0_4

the resulting finite-difference computer program will yield correct approximations. But the finite-difference approach is not necessarily the most efficient one. Hints on other methods will be given at the end of this chapter. For nonstandard options we refer to Chap. 6.

The classic finite-difference methods will be explained in some detail because they are the most elementary approaches to approximate differential equations. As a side-effect, this chapter serves as introduction to several fundamental concepts of numerical mathematics. A trained reader may like to skip Sects. 4.2 and 4.3. The aim of this chapter is to introduce concepts, as well as a characterization of the free boundary (early-exercise curve), and of linear complementarity.

In addition to the finite-difference approach, "standard methods" include analytic methods, which to a significant part are based on nonnumerical analysis. The Sect. 4.8 will give an introduction to several such methods, including interpolation, a method of lines, and a method that solves an integral equation.

The broad field of available methods for pricing standard options calls for comparisons to judge on the relative merits of different approaches. Although such an endeavor goes beyond the scope of a text book, we offer some guidelines in Sect. 4.9.

4.1 Preparations

We allow for dividends paid with a continuous yield of constant level, because numerically this is a trivial extension from the case of no dividend. In case of a discrete dividend with, for example, one payment per year, a first remedy would be to convert the dividend to a continuous yield (⟶ Exercise 4.1).[1]

A continuous flow of dividends is modeled by a decrease of S in each time interval $\mathrm{d}t$ by the amount

$$\delta S \, \mathrm{d}t \, ,$$

with a constant $\delta \geq 0$. This continuous dividend model can be easily built into the Black–Scholes framework. The standard model of a geometric Brownian motion represented by the SDE (1.47) is generalized to

$$\frac{\mathrm{d}S}{S} = (\mu - \delta) \, \mathrm{d}t + \sigma \, \mathrm{d}W \, ,$$

[1] But the corresponding solutions $V(S, t)$ and their early-exercise structure will be different. The Notes and Comments summarize how to correctly compensate for a discrete dividend payment.

with $\mu = r$ according to Remark 1.14. This is the basis for this chapter. The corresponding Black–Scholes equation for the value function $V(S,t)$ is

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - \delta) S \frac{\partial V}{\partial S} - rV = 0\,. \tag{4.1}$$

For constant r, σ, δ, this equation is equivalent to the equation

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2} \tag{4.2}$$

for $y(x,\tau)$ with $0 \leq \tau$, $x \in \mathbb{R}$. The equivalence is proved by means of the transformations

$$\begin{aligned} & S = K\mathrm{e}^x, \quad t = T - \tfrac{2\tau}{\sigma^2}\,, \quad q := \tfrac{2r}{\sigma^2}\,, \quad q_\delta := \tfrac{2(r-\delta)}{\sigma^2}\,, \\ & V(S,t) = V\left(K\mathrm{e}^x, T - \tfrac{2\tau}{\sigma^2}\right) =: v(x,\tau) \quad \text{and} \\ & v(x,\tau) =: K \exp\left\{-\tfrac{1}{2}(q_\delta - 1)x - \left(\tfrac{1}{4}(q_\delta - 1)^2 + q\right)\tau\right\} y(x,\tau)\,. \end{aligned} \tag{4.3}$$

For the case of no dividend payments ($\delta = 0$) the derivation was carried out earlier (⟶ Exercise 1.4). For Black–Scholes-type equations with variable $\sigma(S,t)$, see Appendix A.6.

The transformation $S = K\mathrm{e}^x$ is motivated by the observation that the Black–Scholes equation in the version (4.1) has variable coefficients S^j with powers matching the order of the derivative with respect to S. That is, the relevant terms in (4.1) are of the type

$$S^j \frac{\partial^j V}{\partial S^j}, \quad \text{for } j = 0, 1, 2\,.$$

The transformed version in Eq. (4.2) has constant coefficients (= 1), which makes implementing numerical algorithms easier.

In view of the time transformation in (4.3) the expiration time $t = T$ is determined in the "new" time by $\tau = 0$, and $t = 0$ is transformed to $\tau_{\max} := \frac{1}{2}\sigma^2 T$. Up to the scaling by $\frac{1}{2}\sigma^2$ the new time variable τ represents the remaining life time of the option. And the original domain of the half strip $S > 0$, $0 \leq t \leq T$ belonging to (4.1) becomes the strip

$$-\infty < x < \infty, \quad 0 \leq \tau \leq \frac{1}{2}\sigma^2 T\,, \tag{4.4}$$

on which we are going to approximate a solution $y(x,\tau)$ to (4.2). After that calculation we again apply the transformations of (4.3) to derive out of $y(x,\tau)$ the value of the option $V(S,t)$ in the original variables.

Under the transformations (4.3) the terminal conditions (1.1) and (1.2) become **initial conditions** for $y(x, 0)$. A vanilla call, for example, satisfies

$$V(S, T) = \max\{S - K, 0\} = K \cdot \max\{\mathrm{e}^x - 1, 0\}\,.$$

From (4.3) we find

$$V(S, T) = K \exp\left\{-\frac{x}{2}(q_\delta - 1)\right\} y(x, 0)\,,$$

and thus

$$\begin{aligned} y(x, 0) &= \exp\left\{\frac{x}{2}(q_\delta - 1)\right\} \max\{\mathrm{e}^x - 1, 0\} \\ &= \begin{cases} \exp\left\{\frac{x}{2}(q_\delta - 1)\right\} (\mathrm{e}^x - 1) & \text{for } x > 0 \\ 0 & \text{for } x \le 0\,. \end{cases} \end{aligned}$$

Using

$$\exp\left\{\frac{x}{2}(q_\delta - 1)\right\} (\mathrm{e}^x - 1) = \exp\left\{\frac{x}{2}(q_\delta + 1)\right\} - \exp\left\{\frac{x}{2}(q_\delta - 1)\right\}$$

the initial conditions $y(x, 0)$ for vanilla options in the new variables read

$$\text{call: } y(x, 0) = \max\left\{\mathrm{e}^{\frac{x}{2}(q_\delta+1)} - \mathrm{e}^{\frac{x}{2}(q_\delta-1)},\ 0\right\}\,, \tag{4.5}$$

$$\text{put: } y(x, 0) = \max\left\{\mathrm{e}^{\frac{x}{2}(q_\delta-1)} - \mathrm{e}^{\frac{x}{2}(q_\delta+1)},\ 0\right\}\,. \tag{4.6}$$

Insofar the PDE (4.2) on the strip (4.4) with initial condition (4.5) or (4.6) defines an initial-value problem. In Sect. 4.4 we shall discuss possible boundary conditions needed when the boundaries $x \to -\infty$ and $x \to +\infty$ are truncated.

The Eq. (4.2) is of the type of a parabolic partial differential equation and is the simplest diffusion or heat-conducting equation. Both Eqs. (4.1) and (4.2) are linear in the dependent variables V or y. The differential equation (4.2) is also written $y_\tau = y_{xx}$ or $\dot{y} = y''$. The diffusion term is y_{xx}.

In principle, the methods of this chapter can be applied directly to (4.1). But the equations and algorithms are easier to derive for the algebraically equivalent version (4.2). Note that numerically the two equations are *not* equivalent. A direct application of this chapter's methods to version (4.1) can cause severe difficulties. This will be discussed in Chap. 6. These difficulties will not occur for Eq. (4.2), which is well-suited for standard options with constant coefficients. The Eq. (4.2) is integrated in forward time—that is, for increasing τ starting from $\tau = 0$. This fact is important for stability investigations. For increasing τ the version (4.2) makes sense; this is equivalent to the well-posedness of (4.1) for *de*creasing t.

4.2 Foundations of Finite-Difference Methods

This section describes the basic ideas of finite differences as they are applied to the PDE (4.2).

4.2.1 Difference Approximation

Each two times continuously differentiable function f satisfies

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\xi) \, ,$$

where ξ is an intermediate number between x and $x+h$. The accurate position of ξ is usually unknown. Such expressions are derived by Taylor expansions. We discretize $x \in \mathbb{R}$ by introducing a one-dimensional grid of discrete points x_i with

$$\ldots < x_{i-1} < x_i < x_{i+1} < \ldots$$

For example, choose an equidistant grid with mesh size $h := x_{i+1} - x_i$. The x is discretized, but the function values $f_i := f(x_i)$ are not discrete, $f_i \in \mathbb{R}$. For $f \in \mathcal{C}^2$ the derivative f'' is bounded, and the term $-\frac{h}{2} f''(\zeta)$ can be conveniently written as $O(h)$. This leads to the practical notation

$$f'(x_i) = \frac{f_{i+1} - f_i}{h} + O(h) \, . \tag{4.7}$$

Analogous expressions hold for the partial derivatives of $y(x, \tau)$, which includes a discretization in τ. This suggests to replace the neutral notation h by either Δx or $\Delta \tau$, respectively. The fraction in (4.7) is the difference quotient that approximates the differential quotient $f'(x_i)$; the $O(h^p)$-term is the error. The one-sided (i.e. nonsymmetric) difference quotient (4.7) is of the order $p = 1$. Error orders of $p = 2$ are obtained by central differences

$$f'(x_i) = \frac{f_{i+1} - f_{i-1}}{2h} + O(h^2) \qquad (\text{for } f \in \mathcal{C}^3) \, , \tag{4.8}$$

$$f''(x_i) = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + O(h^2) \qquad (\text{for } f \in \mathcal{C}^4) \, , \tag{4.9}$$

or by one-sided differences that involve more terms, such as

$$f'(x_i) = \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h} + O(h^2) \qquad (\text{for } f \in \mathcal{C}^3) \, . \tag{4.10}$$

Rearranging terms and indices of (4.10) provides the approximation formula

$$f_i \approx \frac{4}{3} f_{i-1} - \frac{1}{3} f_{i-2} + \frac{2}{3} h f'(x_i) \,. \tag{4.11}$$

The latter difference quotient is an example of a *backward differentiation formula* (BDF). Equidistant grids are advantageous in that algorithms are straightforward to implement, and error terms are easily derived by Taylor's expansion. This chapter works with equidistant grids.

4.2.2 The Grid

Either the x-axis, or the τ-axis, or both can be discretized. If only one of the two independent variables x or τ is discretized, one obtains a semidiscretization consisting of parallel lines. This is used in Exercise 4.3 and in Sect. 4.8.3. Here we perform a full discretization leading to a two-dimensional grid. A solution of the discretized problem will be different from the solution y of the initial-value problem on the strip (4.4). To emphasize the difference, we denote a solution of a discretized version w.

Let $\Delta\tau$ and Δx be the mesh sizes of the discretizations of τ and x. The step in τ is $\Delta\tau := \tau_{\max}/\nu_{\max}$ for $\tau_{\max} = \frac{1}{2}\sigma^2 T$ and a suitable integer $\nu_{\max}$. Selecting the x-discretization is more complicated. The infinite interval $-\infty < x < \infty$ must be replaced by a finite interval $x_{\min} \le x \le x_{\max}$, thereby the strip (4.4) changes to a rectangular domain for (x, τ). This truncation or *localization* will have an impact on the solutions w. The finite end values $a = x_{\min} < 0$ and $b = x_{\max} > 0$ must be chosen[2] such that for the corresponding $S_{\min} = K\mathrm{e}^a$ and $S_{\max} = K\mathrm{e}^b$ and the interval $S_{\min} \le S \le S_{\max}$ a sufficient quality of approximation is obtained, in the sense $w \approx y$. In addition, the interval $x_{\min} \le x \le x_{\max}$ must include the range of financial interest, namely, the x-values of S_0 and K. This requires

$$x_{\min} < \min\left\{0, \log\frac{S_0}{K}\right\}, \quad \max\left\{0, \log\frac{S_0}{K}\right\} < x_{\max} \,.$$

For simplicity, just think of $x_{\min} = -3$ and $x_{\max} = 3$. The localization will also need boundary conditions, which will be discussed in Sect. 4.4.

[2] Too large values of $|a|$ or b can lead to underflow or overflow when evaluating the exponential function.

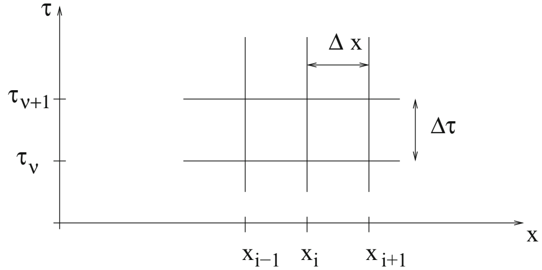

Fig. 4.1 Detail and notations of the grid

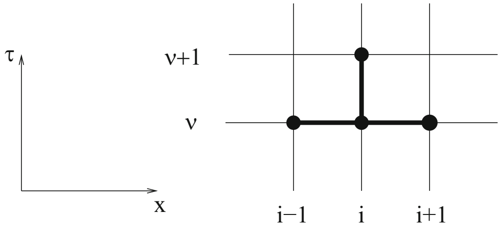

Fig. 4.2 Connection scheme (stencil) of the explicit method

For a suitable integer m the step length in x is defined by $\Delta x := (b-a)/m$. Additional notations for the grid are

$$\begin{aligned}
&\tau_\nu := \nu \cdot \Delta\tau \quad \text{for } \nu = 0, 1, \ldots, \nu_{\max}\\
&x_i := a + i\Delta x \quad \text{for } i = 0, 1, \ldots, m\\
&y_{i,\nu} := y(x_i, \tau_\nu),\\
&w_{i,\nu} \ \text{ approximation to } y_{i,\nu}.
\end{aligned}$$

This defines a two-dimensional uniform grid as illustrated in Fig. 4.1. Note that the equidistant grid in this chapter is defined in terms of x and τ, and not for S and t. Transforming the (x,τ)-grid via the transformation in (4.3) back to the (S,t)-plane, leads to a nonuniform grid with unequal distances of the grid lines $S = S_i = Ke^{x_i}$: The grid is increasingly dense close to $S_{\min}$. (This is not advantageous for the accuracy of the approximations of $V(S,t)$. We will come back to this in Sect. 5.2.) Figure 4.1 illustrates only a small part of the entire grid in the (x,τ)-strip. The grid lines $x = x_i$ and $\tau = \tau_\nu$ can be indicated by their indices (Fig. 4.2).

The points where the grid lines $\tau = \tau_\nu$ and $x = x_i$ intersect, are called *nodes*. In contrast to the theoretical solution $y(x,\tau)$, which is defined on a continuum, the $w_{i,\nu}$ are defined only for the nodes. The error $w_{i,\nu} - y_{i,\nu}$ depends on the choice of the discretization parameters $\nu_{\max}$, m, $x_{\min}$, $x_{\max}$. A priori we do not know which choice of the parameters matches a prespecified error tolerance. An example of the order of magnitude of these parameters is given by $x_{\min} = -5$, $x_{\max} = 5$, or smaller, and $\nu_{\max} = 100$, $m = 100$. Such a choice of $x_{\min}$, $x_{\max}$ has shown to be reasonable for a wide range of r, σ-values and accuracies. Then the actual error is essentially controlled via the numbers $\nu_{\max}$ and m of grid lines.

4.2.3 Explicit Method

Substituting the expressions from (4.7)/(4.10)

$$\frac{\partial y_{i,\nu}}{\partial \tau} := \frac{\partial y(x_i, \tau_\nu)}{\partial \tau} = \frac{y_{i,\nu+1} - y_{i,\nu}}{\Delta\tau} + O(\Delta\tau)$$
$$\frac{\partial^2 y_{i,\nu}}{\partial x^2} = \frac{y_{i+1,\nu} - 2y_{i,\nu} + y_{i-1,\nu}}{\Delta x^2} + O(\Delta x^2)$$

into (4.2) and discarding the O-error terms leads to the difference equation

$$\frac{w_{i,\nu+1} - w_{i,\nu}}{\Delta\tau} = \frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{\Delta x^2}$$

for the approximation w. Solved for $w_{i,\nu+1}$ this is

$$w_{i,\nu+1} = w_{i,\nu} + \frac{\Delta\tau}{\Delta x^2}(w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu})\,.$$

With the abbreviation

$$\lambda := \frac{\Delta\tau}{\Delta x^2}$$

the result is written compactly

$$w_{i,\nu+1} = \lambda w_{i-1,\nu} + (1 - 2\lambda)w_{i,\nu} + \lambda w_{i+1,\nu}\,. \tag{4.12}$$

Figure 4.2 accentuates the nodes that are connected by this formula. Such a graphical scheme illustrating the structure of the equation, is called *stencil* (or molecule).

The Eq. (4.12) and the Fig. 4.2 suggest an evaluation organized by *time levels*. All nodes with the same index ν form the νth time level. For a fixed ν the values $w_{i,\nu+1}$ of the time level $\nu + 1$ are calculated for all i. Then we advance to the next time level, $\nu \to \nu + 1$. The formula (4.12) is an explicit expression for each of the $w_{i,\nu+1}$; the values w at level $\nu + 1$ are not coupled. Since (4.12) provides an explicit formula for all $w_{i,\nu+1}$ $(i = 0, 1, \ldots, m)$, this method is called *explicit method* or *forward-difference method.*

Start: For $\nu = 0$ the values of $w_{i,0}$ are given by the initial conditions

$$w_{i,0} = y(x_i, 0) \quad \text{for } y \text{ from (4.5)/(4.6)},\ 0 \le i \le m.$$

Thereafter we proceed from $\nu = 0$ to $\nu = 1$, and so on. The $w_{0,\nu}$ and $w_{m,\nu}$ for $1 \le \nu \le \nu_{\max}$ are fixed by boundary conditions. For the next few pages, to simplify matters, we artificially set $w_{0,\nu} = w_{m,\nu} = 0$ for all ν. The correct boundary conditions are deferred to Sect. 4.4.

For the following analysis it is useful to collect all values w of the time level ν into a vector,

$$w^{(\nu)} := (w_{1,\nu}, \dots, w_{m-1,\nu})^{tr} .$$

The next step towards a vector notation of the explicit method is to introduce the constant $(m-1) \times (m-1)$ tridiagonal matrix

$$A := A_{\text{expl}} := \begin{pmatrix} 1-2\lambda & \lambda & 0 & \cdots & 0 \\ \lambda & 1-2\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \lambda \\ 0 & \cdots & 0 & \lambda & 1-2\lambda \end{pmatrix} . \tag{4.13}$$

Now the explicit method in matrix-vector notation reads

$$w^{(\nu+1)} = Aw^{(\nu)} \quad \text{for } \nu = 0, 1, 2, \dots \tag{4.14}$$

The formulation with the matrix A of (4.13) and the iteration (4.14) is needed only for theoretical investigations. An actual computer program would rather use the version (4.12). In the vector notation of (4.14), the inner-loop index i does not occur explicitly.

To illustrate the behavior of the explicit method, we perform an experiment with an artificial example, where initial conditions and boundary conditions are not related to finance.

Example 4.1 (Instability) The PDE is $y_\tau = y_{xx}$, initial condition $y(x,0) = \sin \pi x$, $x_0 = 0$, $x_m = 1$, and boundary conditions $y(0,\tau) = y(1,\tau) = 0$ (that is, $w_{0,\nu} = w_{m,\nu} = 0$).

The aim is to calculate an approximation w for one (x,τ), for example, for $x = 0.2$, $\tau = 0.5$. The exact solution is $y(x,\tau) = \mathrm{e}^{-\pi^2\tau} \sin \pi x$, such that $y(0.2, 0.5) = 0.004227\ldots$.

We carry out two calculations with the same $\Delta x = 0.1$ (hence $0.2 = x_2$), and two different $\Delta\tau$:

(a) $\Delta\tau = 0.0005 \Longrightarrow \lambda = 0.05$,
$0.5 = \tau_{1000}, \quad w_{2,1000} \doteq 0.00435$

(b) $\Delta\tau = 0.01 \Longrightarrow \lambda = 1$,
$0.5 = \tau_{50}, \quad w_{2,50} \doteq -1.5 * 10^8$ (the actual numbers depend on the computer)

It turns out that the choice of $\Delta\tau$ in (a) has led to a reasonable approximation, whereas the choice in (b) has caused a disaster. Here we have a stability problem!

4.2.4 Stability

Let us perform an error analysis of an iteration $w^{(\nu+1)} = Aw^{(\nu)} + d^{(\nu)}$. The iteration (4.14) is a special case, with matrix A_{expl}, and the vector $d^{(\nu)}$ vanishes for our preliminary boundary conditions $w_{0,\nu} = w_{m,\nu} = 0$. In general we use the same notation w for the theoretical definition of w and for the values of w obtained by numerical calculations in a computer. Since we now discuss rounding errors, we must distinguish between the two meanings. Let $w^{(\nu)}$ denote the vectors theoretically defined by the iteration. Hence, by definition, the $w^{(\nu)}$ are free of rounding errors. But in computational reality, rounding errors are inevitable. We denote the computer-calculated vector by $\overline{w}^{(\nu)}$ and the error vectors by

$$e^{(\nu)} := \overline{w}^{(\nu)} - w^{(\nu)} ,$$

for $\nu \geq 0$. The $\overline{w}$-result can be written

$$\overline{w}^{(\nu+1)} = A\overline{w}^{(\nu)} + d^{(\nu)} + r^{(\nu+1)} ,$$

where the vectors $r^{(\nu+1)}$ represent the rounding errors that occur during the calculation of $A\overline{w}^{(\nu)} + d^{(\nu)}$. Let us concentrate on the effect of the rounding errors that occur for an arbitrary ν, say for ν^*. We ask for the propagation of this error for increasing $\nu > \nu^*$. Without loss of generality we set $\nu^* = 0$, and for simplicity take $r^{(\nu)} = 0$ for $\nu > 1$. That is, we investigate the effect the initial rounding error $e^{(0)}$ has on the iteration. The initial error $e^{(0)}$ represents the rounding error during the evaluation of the initial condition (4.5)/(4.6), when $\overline{w}^{(0)}$ is calculated. According to this scenario, $\overline{w}^{(\nu+1)} = A\overline{w}^{(\nu)} + d^{(\nu)}$ for $\nu > 1$. The relation

$$Ae^{(\nu)} = A\overline{w}^{(\nu)} - Aw^{(\nu)} = \overline{w}^{(\nu+1)} - w^{(\nu+1)} = e^{(\nu+1)}$$

between consecutive errors is applied repeatedly and results in

$$e^{(\nu)} = A^{\nu} e^{(0)} . \tag{4.15}$$

For the method to be *stable*, previous errors must be damped. This leads to require $A^{\nu}e^{(0)} \to 0$ for $\nu \to \infty$. Elementwise this means $\lim_{\nu\to\infty}\{(A^{\nu})_{ij}\} = 0$ for $\nu \to \infty$ and for any pair of indices (i,j). The following lemma provides a criterion for this requirement.

Lemma 4.2

$$\begin{aligned} \rho(A) < 1 \iff & A^{\nu}z \to 0 \ \textit{for all } z \textit{ and } \nu \to \infty \\ \iff & \lim_{\nu\to\infty}\{(A^{\nu})_{i,j}\} = 0 \end{aligned}$$

Here $\rho(A)$ is the *spectral radius* of A,

$$\rho(A) := \max_k |\mu_k^A| \,,$$

where $\mu_1^A, \ldots, \mu_{m-1}^A$ denote the eigenvalues of A, labeled with index k. The proof can be found in text books on numerical analysis, for example, in [198]. As a consequence of Lemma 4.2 we require for stable behavior that $|\mu_k^A| < 1$ for all eigenvalues, here for $k = 1, \ldots, m-1$. To check the criterion of Lemma 4.2, the eigenvalues μ_k^A of A are needed. The matrix A can be written

$$A = I - \lambda \cdot \underbrace{\begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix}}_{=:G} .$$

It remains to investigate the eigenvalues μ^A or μ^G of the tridiagonal matrices A or G.[3]

Lemma 4.3 *Let*

$$G = \begin{pmatrix} \alpha & \beta & & 0 \\ \gamma & \ddots & \ddots & \\ & \ddots & \ddots & \beta \\ 0 & & \gamma & \alpha \end{pmatrix}$$

be an N^2-matrix. The eigenvalues μ_k^G are

$$\mu_k^G = \alpha + 2\beta \sqrt{\frac{\gamma}{\beta}} \cos \frac{k\pi}{N+1} \ , \ k = 1, \ldots, N \, .$$

Proof The eigenvectors $v^{(k)}$ of G are

$$v^{(k)} = \left(\sqrt{\frac{\gamma}{\beta}} \sin \frac{k\pi}{N+1}, \left(\sqrt{\frac{\gamma}{\beta}}\right)^2 \sin \frac{2k\pi}{N+1}, \ldots, \left(\sqrt{\frac{\gamma}{\beta}}\right)^N \sin \frac{Nk\pi}{N+1} \right)^{tr} .$$

Substitute this into $Gv = \mu^G v$. □

[3] The zeros in the corner of the matrix G symbolize the triangular zero structure of (4.13).

To apply Lemma 4.3 to A or to G observe $N = m-1$, and for G $\alpha = 2, \beta = \gamma = -1$. Accordingly, the eigenvalues μ^G and the eigenvalues μ^A are

$$\mu_k^G = 2 - 2\cos\frac{k\pi}{m} = 4\sin^2\left(\frac{k\pi}{\mathbf{2}m}\right),$$

$$\mu_k^A = 1 - \lambda\mu^G = 1 - 4\lambda\sin^2\frac{k\pi}{2m}.$$

Now we can state the stability requirement $|\mu_k^A| < 1$ as

$$\left|1 - 4\lambda\sin^2\frac{k\pi}{2m}\right| < 1, \quad k = 1, \ldots, m-1.$$

This implies the two inequalities $\lambda > 0$ and

$$-1 < 1 - 4\lambda\sin^2\frac{k\pi}{2m}, \quad \text{rewritten as} \quad \frac{1}{2} > \lambda\sin^2\frac{k\pi}{2m}.$$

The largest sin-term is $\sin\frac{(m-1)\pi}{2m}$; for increasing m this term grows monotonically approaching 1. In summary we have shown for (4.13)/(4.14)

$$\text{For } 0 < \lambda \le \frac{1}{2} \text{ the explicit method } w^{(\nu+1)} = Aw^{(\nu)} \text{ is stable.}$$

In view of $\lambda = \Delta\tau/\Delta x^2$ this stability criterion amounts to bounding the $\Delta\tau$ step size,

$$0 < \Delta\tau \le \frac{\Delta x^2}{2}. \tag{4.16}$$

This explains what happened with Example 4.1. The values of λ in the two cases of this example are

$$\text{(a)} \quad \lambda = 0.05 \le \frac{1}{2},$$

$$\text{(b)} \quad \lambda = 1 > \frac{1}{2}.$$

In case (b) the chosen $\Delta\tau$ and hence λ were too large, which led to an amplification of rounding errors resulting eventually in the "explosion" of the w-values.

The explicit method is stable only as long as (4.16) is satisfied. As a consequence, the parameters m and $\nu_{\max}$ of the grid resolution can not be chosen independent of each other. If the demands for accuracy are high, the step size Δx will be small, which in view of (4.16) bounds $\Delta\tau$ quadratically. This situation suggests searching for a method that is unconditionally stable.

4.2.5 An Implicit Method

Introducing the explicit method in Sect. 4.2.3, we have approximated the time derivative with a forward difference, "forward" as seen from the νth time level. Now we try the backward difference in

$$\frac{\partial y_{i,\nu}}{\partial \tau} = \frac{y_{i,\nu} - y_{i,\nu-1}}{\Delta\tau} + O(\Delta\tau)\,,$$

which yields the alternative to (4.12)

$$-\lambda w_{i+1,\nu} + (1+2\lambda)w_{i,\nu} - \lambda w_{i-1,\nu} = w_{i,\nu-1}\,. \tag{4.17}$$

The Eq. (4.17) relates the time level ν to the time level $\nu - 1$. For the transition from $\nu - 1$ to ν only the value $w_{i,\nu-1}$ on the right-hand side of (4.17) is known, whereas on the left-hand side of the equation three unknown values of w wait to be computed. Equation (4.17) couples three unknowns. The corresponding stencil is shown in Fig. 4.3. There is no simple explicit formula with which the unknowns can be obtained one after the other. Rather a system must be considered, all equations simultaneously. A vector notation reveals the structure of (4.17): With the matrix

$$A := A_{\text{impl}} := \begin{pmatrix} 1+2\lambda & -\lambda & & 0 \\ -\lambda & \ddots & \ddots & \\ & \ddots & \ddots & -\lambda \\ 0 & & -\lambda & 1+2\lambda \end{pmatrix} \tag{4.18}$$

the vector $w^{(\nu)}$ is implicitly defined as solution of the system of linear equations $Aw^{(\nu)} = w^{(\nu-1)}$. To have a consistent numbering, rewrite this as

$$Aw^{(\nu+1)} = w^{(\nu)} \quad \text{for } \nu = 0, \ldots, \nu_{\max} - 1\,. \tag{4.19}$$

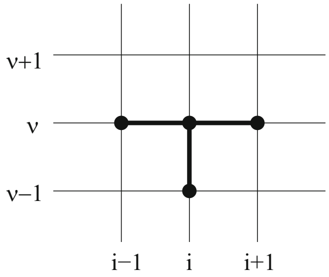

Fig. 4.3 Stencil of the backward-difference method (4.17)

(Again we set $w_{0,\nu} = w_{m,\nu} = 0$.) For each time level ν such a system of equations must be solved. This method is sometimes called *implicit method*. But to distinguish it from other implicit methods, we call it *fully implicit*, or *backward-difference method*, or more accurately *backward time centered space* scheme (BTCS). The method is unconditionally stable for all $\Delta\tau > 0$. This is shown analogously as in the explicit case (⟶ Exercise 4.2). The costs of this implicit method are low, because the matrix A is constant and tridiagonal. Initially, for $\nu = 0$, the *LR*-decomposition (⟶ Appendix C.1) is calculated once. Then the costs for each ν are only of the order $O(m)$.

4.3 Crank–Nicolson Method

For the methods of the previous section the discretizations of $\frac{\partial y}{\partial \tau}$ are of the order $O(\Delta\tau)$. It seems preferable to use a method where the time discretization of $\frac{\partial y}{\partial \tau}$ has the better order $O(\Delta\tau^2)$, and the stability is unconditional. Let us again consider Eq. (4.2), the equivalent to the Black–Scholes equation,

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2}\,.$$

Crank and Nicolson suggested to average the forward- and the backward difference method. For easy reference, we collect the underlying approaches from the above: forward for ν:

$$\frac{w_{i,\nu+1} - w_{i,\nu}}{\Delta\tau} = \frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{\Delta x^2}$$

backward for $\nu + 1$:

$$\frac{w_{i,\nu+1} - w_{i,\nu}}{\Delta\tau} = \frac{w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1}}{\Delta x^2}$$

Addition yields

$$\frac{w_{i,\nu+1} - w_{i,\nu}}{\Delta\tau} = \frac{1}{2\Delta x^2}(w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu} + w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1}) \tag{4.20}$$

The Eq. (4.20) involves in each of the time levels ν and $\nu + 1$ three values of w (Fig. 4.4). This is the basis of an efficient method. Its features are summarized in Theorem 4.4.

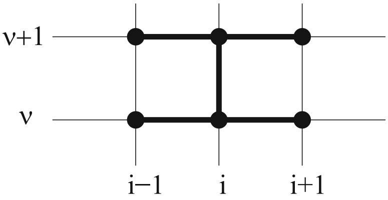

Fig. 4.4 Stencil of the Crank–Nicolson method (4.20)

Theorem 4.4 (Crank–Nicolson)

1.) For $y \in \mathcal{C}^4$ the order of the method is $O(\Delta\tau^2) + O(\Delta x^2)$.
2.) For each ν a linear system of a simple tridiagonal structure must be solved.
3.) Stability holds for all $\Delta\tau > 0$.

Proof

1.) order: A practical notation for the symmetric difference quotient of second order for y_{xx} is

$$\delta_{xx} w_{i,\nu} := \frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{\Delta x^2} . \tag{4.21}$$

Apply the operator δ_{xx} to the exact solution y. Then by Taylor expansion for $y \in \mathcal{C}^4$ one shows

$$\delta_{xx} y_{i,\nu} = \frac{\partial^2}{\partial x^2} y_{i,\nu} + \frac{\Delta x^2}{12}\frac{\partial^4}{\partial x^4} y_{i,\nu} + O(\Delta x^4) .$$

The *local discretization error* ϵ describes how well the exact solution y of (4.2) satisfies the difference scheme,

$$\epsilon := \frac{y_{i,\nu+1} - y_{i,\nu}}{\Delta\tau} - \frac{1}{2}(\delta_{xx} y_{i,\nu} + \delta_{xx} y_{i,\nu+1}) .$$

Applying the operator δ_{xx} of (4.21) to the expansion of $y_{i,\nu+1}$ at τ_ν and observing $y_\tau = y_{xx}$ leads to

$$\epsilon = O(\Delta\tau^2) + O(\Delta x^2) .$$

(⟶ Exercise 4.4)

2.) system of equations: With $\lambda := \frac{\Delta\tau}{\Delta x^2}$ the Eq. (4.20) is rewritten

$$\begin{aligned} &-\tfrac{\lambda}{2} w_{i-1,\nu+1} + (1+\lambda) w_{i,\nu+1} - \tfrac{\lambda}{2} w_{i+1,\nu+1} \\ &= \tfrac{\lambda}{2} w_{i-1,\nu} + (1-\lambda) w_{i,\nu} + \tfrac{\lambda}{2} w_{i+1,\nu} . \end{aligned} \tag{4.22}$$

The values of the new time level $\nu + 1$ are implicitly given by the system of Eqs. (4.22). For the simplest boundary conditions $w_{0,\nu} = w_{m,\nu} = 0$ Eq. (4.22) is a system of $m-1$ equations. With matrices

$$
\begin{aligned}
A :&= A_{\mathrm{CN}} := \begin{pmatrix} 1+\lambda & -\frac{\lambda}{2} & & 0 \\ -\frac{\lambda}{2} & \ddots & \ddots & \\ & \ddots & \ddots & -\frac{\lambda}{2} \\ 0 & & -\frac{\lambda}{2} & 1+\lambda \end{pmatrix}, \\
B :&= B_{\mathrm{CN}} := \begin{pmatrix} 1-\lambda & \frac{\lambda}{2} & & 0 \\ \frac{\lambda}{2} & \ddots & \ddots & \\ & \ddots & \ddots & \frac{\lambda}{2} \\ 0 & & \frac{\lambda}{2} & 1-\lambda \end{pmatrix}
\end{aligned}
\tag{4.23}
$$

the system (4.22) is rewritten

$$
Aw^{(\nu+1)} = Bw^{(\nu)} . \tag{4.24}
$$

The eigenvalues of A are real and lie between 1 and $1 + 2\lambda$ (follows from the Theorem of Gerschgorin, see Appendix C.1). This rules out a zero eigenvalue, and so A must be nonsingular and the solution $w^{(\nu+1)}$ of (4.24) is uniquely defined.

3.) stability: Both matrices A and B can be rewritten in terms of a constant tridiagonal matrix,

$$
A = I + \frac{\lambda}{2}G, \quad G := \begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix}, \quad B = I - \frac{\lambda}{2}G .
$$

Now the Eq. (4.24) reads

$$
\begin{aligned}
\underbrace{(2I+\lambda G)}_{=:C} w^{(\nu+1)} &= (2I-\lambda G)w^{(\nu)} \\
&= (4I - 2I - \lambda G)w^{(\nu)} \\
&= (4I - C)w^{(\nu)} ,
\end{aligned}
$$

which leads to the formally explicit iteration

$$
w^{(\nu+1)} = (4C^{-1} - I)w^{(\nu)} . \tag{4.25}
$$

The eigenvalues μ_k^C of C for $k = 1, \dots, m-1$ are known from Sect. 4.2.4,

$$\mu_k^C = 2 + \lambda\mu_k^G = 2 + \lambda(2 - 2\cos\frac{k\pi}{m}) = 2 + 4\lambda\sin^2\frac{k\pi}{2m}\,.$$

In view of (4.25) we require for a stable method that for all k

$$\left|\frac{4}{\mu_k^C} - 1\right| < 1\,.$$

This is guaranteed because of $\mu_k^C > 2$. Consequently, the Crank–Nicolson method (4.20)/(4.23)/(4.24) is unconditionally stable for all $\lambda > 0$ $(\Delta\tau > 0)$.

Although correct boundary conditions are still lacking, it makes sense to formulate the basic version of the Crank–Nicolson algorithm for the PDE (4.2).

Algorithm 4.5 (Crank-Nicolson)

```
Start: Choose $m$, $\nu_{\max}$; calculate $\Delta x$, $\Delta\tau$
    $w_i^{(0)} = y(x_i, 0)$ with $y$ from (4.5) or (4.6), $0 \le i \le m$.
    Calculate the LR-decomposition of $A$.
loop: for $\nu = 0, 1, \dots, \nu_{\max} - 1$:
    Calculate $c := Bw^{(\nu)}$    (preliminary).
    Solve $Ax = c$ using e.g. the LR-decomposition—
        that is, solve $Lz = Bw^{(\nu)}$ and $Rx = z$.
$w^{(\nu+1)} := x$
```

The *LR*-decomposition is the symbol for the solution of the system of linear equations. Later we shall see when to replace it by the *RL*-decomposition. Obviously the matrices A and B are not stored in the computer. Next we show how the vector c in Algorithm 4.5 is modified to realize correct boundary conditions.

4.4 Boundary Conditions

On the unbounded domain $-\infty < x < \infty$ the initial-value problem $y_\tau = y_{xx}$ with initial condition (4.5)/(4.6) and $\tau \ge 0$ is well-posed. But the truncation to the interval $x_{\min} \le x \le x_{\max}$ changes the type of the problem. To make the PDE-problem well-posed in the finite-domain case, boundary conditions must be imposed artificially. They are not stated in the option's contract, and are not needed by Monte Carlo or tree methods. Boundary conditions are the price one has to pay when PDE-based approaches are applied. Since boundary conditions are often approximations of the reality, the "localized solution" on the finite domain $x_{\min} \le x \le x_{\max}$ in general is different from the solution of the pure initial-value problem. For simplicity, we neglect this difference, and denote the localized solution again by y. We need to formulate boundary conditions such that the localized solution is close to the solution of the original problem. The choice of boundary conditions is not unique.

In the variety of possible boundary conditions there are two kinds so important and so frequent that they have names. For *Dirichlet conditions*, a value is assigned to y, whereas a *Neumann condition* assigns a value to the derivative $\mathrm{d}y/\mathrm{d}x$. For a call, for example, $y(x_{\min}) = 0$ is Dirichlet, and $\frac{\partial y(x_{\max})}{\partial x} = 1$ is Neumann. More generally, with x_b standing for $x_{\min}$ or $x_{\max}$,

$$y(x_\mathrm{b}, t) = \alpha(t)$$

for some function $\alpha(t)$ is an example of a Dirichlet condition. A discretized version is $w_{0,\nu} = \alpha(\tau_\nu)$. That is, our preliminary boundary conditions $w_{0,\nu} = w_{m,\nu} = 0$ have been of Dirichlet type. And a Neumann condition would be

$$\frac{\partial y(x_\mathrm{b}, t)}{\partial x} = \beta(t)$$

for some function $\beta(t)$. On our grid, a second-order approximation (4.8) for this Neumann condition is

$$w_{1,\nu} - w_{-1,\nu} = \beta(\tau_\nu)\, 2\Delta x\,,$$

which uses a fictive grid point x_{-1} outside the interval. The required information on $w_{-1,\nu}$ is provided by a discretized version of the PDE. Alternatively, the one-sided second-order difference quotient (4.10) can be applied. As a result, one or more entries of the matrix A would change, which makes a finite-difference realization of a Neumann condition a bit cumbersome. Dirichlet conditions are easier to cope with. Let us try to analyze the value function $V(S, t)$ for $S = 0$ and $S \to \infty$ in order to derive Dirichlet conditions for $S_{\min}$ and $S_{\max}$, and for

$$\begin{aligned} &y(x, \tau) \text{ for } x = x_{\min} \text{ and } x_{\max} \text{ and all } \tau\,, \quad \text{or} \\ &w_{0,\nu} \text{ and } w_{m,\nu} \text{ for } \nu = 1, \ldots, \nu_{\max}\,, \end{aligned}$$

all consistent with the Black-Scholes model.

Accordingly, we assume GBM paths satisfying (1.47). Then an initial value $S_0 = 0$ causes $S_t = 0$ for all t, and $S_0 \to \infty$ implies that S_t is arbitrarily large, at least larger than the strike K. The boundary conditions for the expiration time $t = T$ are obviously given by the payoff Ψ. This gives rise to the simplest cases of boundary conditions for $t < T$: As motivated by Figs. 1.1 and 1.2 and Eqs. (1.1), (1.2), the value V_C of a call and the value V_P of a put must satisfy

$$\begin{aligned} &V_\mathrm{C}(S, t) = 0 \quad \text{for } S = 0, \quad \text{and} \\ &V_\mathrm{P}(S, t) \to 0 \quad \text{for } S \to \infty \end{aligned} \tag{4.26}$$

also for all $t < T$. This follows, for example, from the integral representation (3.25), because discounting does not affect the value 0 of the payoff. Hence the value V_{C} for $S = 0$ can be predicted safely, as well as V_{P} for $S(0) \to \infty$. These arguments hold for European as well as for American options, with or without dividend payments.

The boundary conditions on each of the "other sides" of S, where $V \neq 0$, are more difficult. We postpone the boundary conditions for American options to the next section, and investigate European options in this section.

From (4.26) and the put-call parity (⟶ Exercise 1.1) we deduce the additional boundary conditions for European options. The result is

$$\begin{aligned} V_{\mathrm{C}}(S,t) &= S - K\mathrm{e}^{-r(T-t)} \quad \text{for } S \to \infty \\ V_{\mathrm{P}}(S,t) &= K\mathrm{e}^{-r(T-t)} - S \quad \text{for } S \to 0 \end{aligned} \tag{4.27}$$

(without dividend payment, $\delta = 0$). The lower bounds for European options (⟶ Appendix E.1) are attained at the boundaries. In (4.27) for $S \approx 0$ we do not discard the term S, because the realization of the transformation (4.3) requires $S_{\min} > 0$, see Sect. 4.2.2.[4] Boundary conditions analogous as in (4.27) hold for the case of a continuous flow of dividend payments ($\delta > 0$). We skip the derivation, which can be based on transformation (4.3) and the additional transformation $S = \overline{S}\mathrm{e}^{\delta(T-t)}$ (⟶ Exercise 4.5). In summary, the asymptotic boundary conditions for European options in the (x, τ)-world are as follows:

Boundary Conditions 4.6 (European Options)

$$\begin{aligned} &y(x,\tau) = r_1(x,\tau) \; \textit{for } x \to -\infty\,, \\ &y(x,\tau) = r_2(x,\tau) \; \textit{for } x \to \infty\,, \qquad \textit{with} \\ &\textit{call: } r_1(x,\tau) := 0\,, \\ &\qquad r_2(x,\tau) := \exp\left(\tfrac{1}{2}(q_\delta + 1)x + \tfrac{1}{4}(q_\delta + 1)^2\tau\right)\,, \\ &\textit{put: } r_1(x,\tau) := \exp\left(\tfrac{1}{2}(q_\delta - 1)x + \tfrac{1}{4}(q_\delta - 1)^2\tau\right)\,, \\ &\qquad r_2(x,\tau) := 0\,. \end{aligned} \tag{4.28}$$

Truncation What can we state about $S_{\min}$ and $S_{\max}$? We need boundary conditions for the finite interval

$$a := x_{\min} \leq x \leq x_{\max} =: b\,.$$

The probability that $S_T < K$ when $S_0 = S_{\min}$ can be estimated by the transition density (1.64). By the same argument the probability is known that $S_T > K$ when $S_0 = S_{\max}$. Both probabilities are large as long as $S_{\min}$ is small and $S_{\max}$ large enough. This situation suggests to apply the boundary conditions (4.26) and (4.27) also to the left-hand boundary $S_{\min}$ and to the right-hand boundary $S_{\max}$.

[4] For $S = 0$ the PDE is no longer parabolic.

Although (4.28) is valid only for $x \to -\infty$ and $x \to \infty$, we apply the dominant terms $r_1(x,\tau)$ and $r_2(x,\tau)$ to approximate boundary conditions at $x = a$ and $x = b$. This leads to the boundary conditions

$$\begin{aligned} w_{0,\nu} &= r_1(a,\tau_\nu) \\ w_{m,\nu} &= r_2(b,\tau_\nu) \end{aligned}$$

for all ν.

These approximations are explicit formulas and easy to implement. To this end return to the Crank–Nicolson equation (4.22), in which some of the terms on both sides of the equations are known by the boundary conditions. For the equation with $i = 1$ these are terms

$$\begin{aligned} &\text{from the left-hand side:} \quad -\frac{\lambda}{2} w_{0,\nu+1} = -\frac{\lambda}{2} r_1(a,\tau_{\nu+1}) \,, \\ &\text{from the right-hand side:} \quad \frac{\lambda}{2} w_{0,\nu} = \frac{\lambda}{2} r_1(a,\tau_\nu) \,, \end{aligned}$$

and for $i = m-1$

$$\begin{aligned} &\text{from the left-hand side:} \quad -\frac{\lambda}{2} w_{m,\nu+1} = -\frac{\lambda}{2} r_2(b,\tau_{\nu+1}) \,, \\ &\text{from the right-hand side:} \quad \frac{\lambda}{2} w_{m,\nu} = \frac{\lambda}{2} r_2(b,\tau_\nu) \,. \end{aligned}$$

These known boundary values are collected on the right-hand side of system (4.22). So we finally arrive at

$$\begin{aligned} Aw^{(\nu+1)} &= Bw^{(\nu)} + d^{(\nu)} \\ d^{(\nu)} : &= \tfrac{\lambda}{2} \cdot \begin{pmatrix} r_1(a,\tau_{\nu+1}) + r_1(a,\tau_\nu) \\ 0 \\ \vdots \\ 0 \\ r_2(b,\tau_{\nu+1}) + r_2(b,\tau_\nu) \end{pmatrix} \end{aligned} \tag{4.29}$$

The preliminary version (4.24) is included as special case, with $d^{(\nu)} = 0$. The statement in Algorithm 4.5 that defines c is modified to the statement

$$\text{Calculate } c := Bw^{(\nu)} + d^{(\nu)} \,.$$

The methods of Sect. 4.2 can be adapted by analogous formulas. The matrix A is not changed, and the stability is not affected by adding the vector d, which is constant with respect to w.

4.5 Early-Exercise Structure

In Sects. 4.1 through 4.3 we have considered tools for the Black–Scholes differential equation—that is, we have investigated European options. Now we turn our attention to American options. Recall that the value of an American option can never be smaller than the value of a European option,

$$V^{\mathrm{Am}} \geq V^{\mathrm{Eur}}.$$

In addition, an American option has at least the value of the payoff Ψ. So we have elementary lower bounds for the value of American options, but—as we shall see—additional numerical problems to cope with.

4.5.1 Early-Exercise Curve

A European option can have a value that is smaller than the payoff (compare, for example, Fig. 1.6). This can not happen with American options. Recall the arbitrage strategy: if for instance an American put would have a value $V_{\mathrm{P}}^{\mathrm{Am}} < (K-S)^{+}$, one would simultaneously purchase the asset and the put, and exercise immediately. An analogous arbitrage argument implies that for an American call the situation $V_{\mathrm{C}}^{\mathrm{Am}} < (S-K)^{+}$ can not prevail. Therefore the inequalities

$$\begin{aligned} V_{\mathrm{P}}^{\mathrm{Am}}(S,t) &\geq (K-S)^{+} \quad \text{for all } (S,t) \\ V_{\mathrm{C}}^{\mathrm{Am}}(S,t) &\geq (S-K)^{+} \quad \text{for all } (S,t) \end{aligned} \tag{4.30}$$

hold. For a put this is illustrated schematically in Fig. 4.5. The inequalities for V make the problem of calculating an American option nonlinear.

For American options we have stated in (4.26) the boundary conditions that prescribe $V = 0$. The boundary conditions at each of the other "ends" of the S-axis are still needed. In view of the inequalities (4.30) it is clear that the missing boundary conditions will be of a different kind than those for European options,

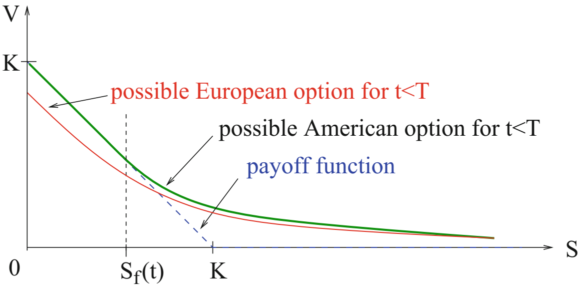

Fig. 4.5 $V(S,t)$ for a put and a $t<T$, schematically

which are listed in (4.27). Let us investigate the situation of an **American put**, which is illustrated in Fig. 4.5. First discuss the left-end part of the *curve* $V_{\mathrm{P}}(S,t)$, for small $S > 0$, and some $t < T$. Without the possibility of early exercise the inequality $V_{\mathrm{P}}^{\mathrm{Am}}(S,t) = V_{\mathrm{P}}^{\mathrm{Eur}}(S,t) < K - S$ holds for $r > 0$ and sufficiently small S. But in view of (4.30) the American put should satisfy $V_{\mathrm{P}}^{\mathrm{Am}}(S,t) \equiv K - S$ at least for small S. To understand what happens for "medium" values of S, imagine to approach from the right-hand side, where $V_{\mathrm{P}}^{\mathrm{Am}}(S,t) > (K-S)^+$. Continuity and monotony of V_{P} suggest that the curve $V_{\mathrm{P}}^{\mathrm{Am}}(S,t)$ merges into the straight line $K - S$ of the payoff at some value S_{f} in the interval $0 < S_{\mathrm{f}} < K$, see Fig. 4.5. This **contact point** S_{f} is defined by

$$\begin{aligned} &V_{\mathrm{P}}^{\mathrm{Am}}(S,t) > (K-S)^+ \quad &&\text{for } S > S_{\mathrm{f}}(t), \\ &V_{\mathrm{P}}^{\mathrm{Am}}(S,t) = K - S &&\text{for } S \le S_{\mathrm{f}}(t)\,. \end{aligned} \tag{4.31}$$

Convexity of $V(S,.)$ guarantees that there is only one contact point S_{f} for each t. For $S < S_{\mathrm{f}}$ the value $V_{\mathrm{P}}^{\mathrm{Am}}$ equals the straight line of the payoff and nothing needs to be calculated. For each t, the *curve* $V_{\mathrm{P}}^{\mathrm{Am}}(S,t)$ reaches its left boundary at $S_{\mathrm{f}}(t)$.

The above situation holds for any $t < T$, and the contact point S_{f} varies with t, $S_{\mathrm{f}} = S_{\mathrm{f}}(t)$. For all $0 \le t < T$, the contact points $S_{\mathrm{f}}(t)$ form a curve in the (S,t)-half strip. This curve S_{f} is the boundary separating the area with $V >$ payoff from the area with $V =$ payoff. The curve S_{f} of a put is illustrated in the left-hand diagram of Fig. 4.6. A priori the location of the boundary S_{f} is unknown, the curve is "free." This explains why the problem of calculating $V_{\mathrm{P}}^{\mathrm{Am}}(S,t)$ for $S > S_{\mathrm{f}}(t)$ is called **free boundary problem**.

For **American calls** the situation is similar, except that the contact only occurs for dividend-paying assets, $\delta \neq 0$. This is seen from

$$V_{\mathrm{C}}^{\mathrm{Am}} \ge V_{\mathrm{C}}^{\mathrm{Eur}} \ge S - K\mathrm{e}^{-r(T-t)} > S - K$$

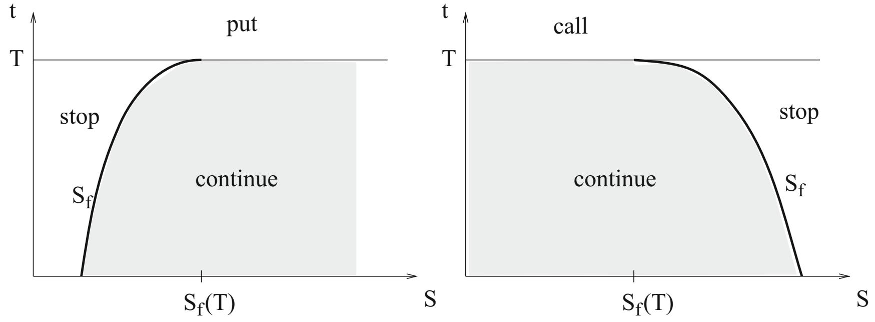

Fig. 4.6 Continuation region (*shaded*) and stopping region for American options: put (*left*) and call (*right*)

for $\delta = 0$, $r > 0$, $t < T$, compare Exercise 1.1. $V_C^{Am} > S - K$ for $\delta = 0$ implies that early-exercise does not pay. American and European calls on assets that pay no dividends are identical, $V_C^{Am} = V_C^{Eur}$. A typical curve $V_C^{Am}(S, t)$ for $\delta \neq 0$ contacting the payoff is shown in Fig. 4.9. And the free boundary S_f qualitatively looks like the right-hand diagram of Fig. 4.6.

The notation $S_f(t)$ for the free boundary is motivated by the process of solving PDEs. But the primary meaning of the curve S_f is economical. The free boundary S_f is the **early-exercise curve**. The time instance t_s when a price process S_t reaches the early-exercise curve is the optimal stopping time, compare also the illustration of Fig. 3.10. Let us explain this for the case of a put; for a call with dividend payment the argument is similar.

For a put, in case $S > S_f$, early-exercise causes an immediate loss, because (4.31) implies the exercise balance $-V + K - S < 0$. Receiving the strike price K does not compensate the loss of S and V. Accordingly, the rational holder of the option does not exercise when $S > S_f$. This explains why the area $S > S_f$ is called **continuation region**[5] (shaded in Fig. 4.6).

On the other side of the boundary curve S_f, characterized by $V = K - S$, each change of S is compensated by a corresponding move of V. Here the only way to create a profit is to exercise and invest the proceeds K at the risk-free rate r for the remaining time period $T - t$. The resulting profit will be

$$K\mathrm{e}^{r(T-t)} - K\,,$$

which relies on $r > 0$. (For $r = 0$ American and European put are identical.) To maximize the profit, the holder of the option will maximize $T - t$, and accordingly exercises as soon as $V \equiv K - S$ is reached. Hence, the boundary curve S_f is the early-exercise curve. And the area $S \leq S_f$ is called **stopping region**.[6]

Now that the curve S_f is recognized as having such a distinguished importance as early-exercise curve, we should make sure that the properties of S_f are as suggested by Figs. 4.6 and 4.7. In fact, the curves $S_f(t)$ are continuously differentiable in t, and monotone not decreasing/not increasing as illustrated. For more details see Appendix A.5. Here we confine ourselves to the bounds given by the limit $t \to T$ $(t < T)$:

$$\text{put:} \quad \lim_{t\to T^-} S_f(t) = \begin{cases} K & \text{for } 0 \leq \delta \leq r \\ \frac{r}{\delta}K & \text{for } r < \delta \end{cases} \tag{4.32}$$

$$\text{call:} \quad \lim_{t\to T^-} S_f(t) = \max(K, \frac{r}{\delta}K) \quad \text{for } \delta > 0 \tag{4.33}$$

[5] Of course, the holder may wish to sell the option.

[6] The final balance for a put after exercising is $K\mathrm{e}^{r(T-t)}$. The reader is encouraged to show that holding is less profitable ($S\mathrm{e}^{\delta(T-t)} < K\mathrm{e}^{r(T-t)}$), at least for small $r(T-t)$. When a discrete dividend is paid, the stopping region is not necessarily connected ($\longrightarrow$ Exercise 4.1b).

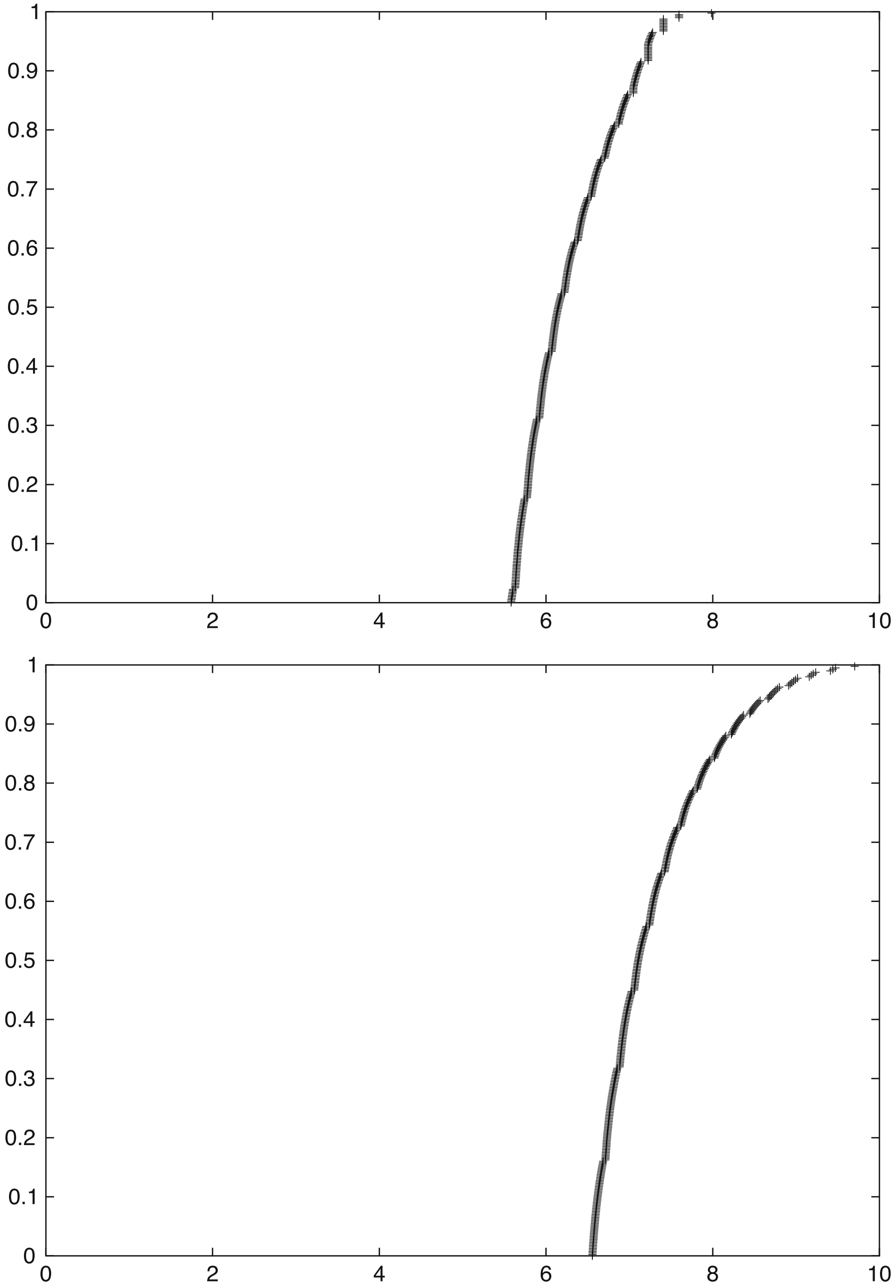

Fig. 4.7 Early-exercise curves of an American put in (S,t)-planes, $r = 0.06$, $\sigma = 0.3$, $K = 10$, and dividend rates $\delta = 0.08$ (*top*), $\delta = 0.04$ (*bottom*); raw data of a finite-difference calculation without interpolation or smoothing

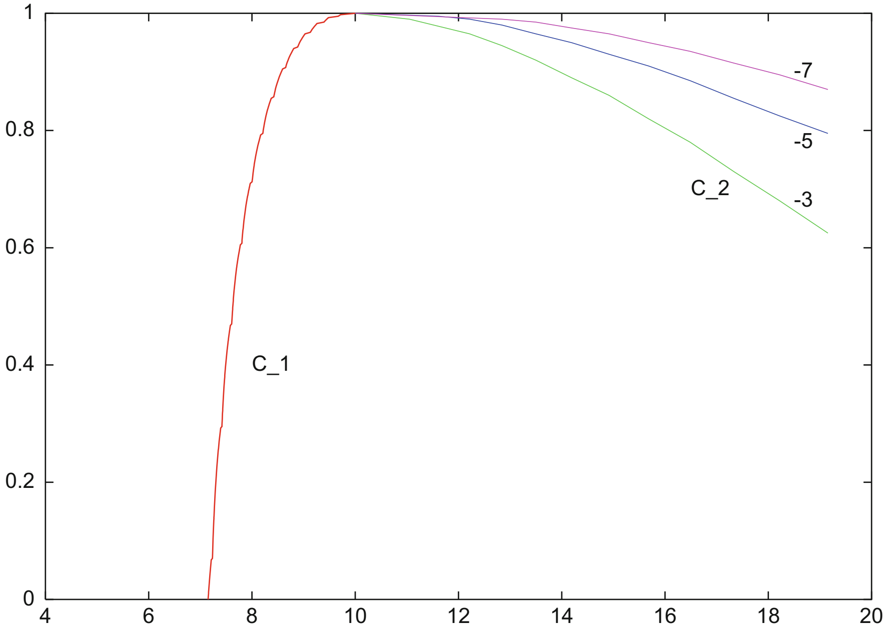

Fig. 4.8 (S, t)-plane, calculated curves of a put matching Figs. 1.4 and 1.5. C_1 is the curve S_f. The three curves C_2 have the meaning $V < 10^{-k}$ for $k = 3, 5, 7$

These bounds express a qualitatively different behavior of the early-exercise curve in the two situations $0 < \delta < r$ and $\delta > r$. This is illustrated in Fig. 4.7 for a put. For the chosen numbers, for all $\delta \leq 0.06$ the limit of (4.32) is the strike K (lower diagram). Compare to Figs. 1.4 and 1.5 to get a feeling for the geometrical importance of the curve as contact line where two surfaces merge. For large values of S the surface $V(S, t)$ approaches 0 in a way illustrated by Fig. 4.8.

4.5.2 Free-Boundary Problem

Again we start with a put. For the European option, the left-end boundary condition is formulated for $S = 0$. For the American option, the left-end boundary is given along the curve S_f (Fig. 4.5). In order to calculate the free boundary $S_\text{f}(t)$ one needs an additional condition. To this end consider the right-hand slope $\frac{\partial V}{\partial S}$ with which $V_\text{P}^\text{Am}(S, t)$ touches at $S_\text{f}(t)$ the straight line $K - S$, which has the constant slope -1. By geometrical reasons we can rule out the case $\frac{\partial V(S_\text{f}(t),t)}{\partial S} < -1$ for V_P^Am, because otherwise (4.30) and (4.31) would be violated. Using arbitrage arguments, the case $\frac{\partial V(S_\text{f}(t),t)}{\partial S} > -1$ can be ruled out as well ($\longrightarrow$ Exercise 4.6). It remains the condition $\partial V_\text{P}^\text{Am}(S_\text{f}(t), t)/\partial S = -1$. That is, $V(S, t)$ touches the payoff function *tangentially*.

This tangency condition is commonly called the *high-contact condition*, or *smooth pasting*. For the case of an option without maturity (*perpetual option*, $T = \infty$) the tangential touching can be calculated analytically ($\longrightarrow$ Exercise 4.7). In summary, *two* boundary conditions must hold at the contact point $S_\mathrm{f}(t)$:

$$\begin{aligned} V_\mathrm{P}^\mathrm{Am}(S_\mathrm{f}(t), t) &= K - S_\mathrm{f}(t) \\ \frac{\partial V_\mathrm{P}^\mathrm{Am}(S_\mathrm{f}(t), t)}{\partial S} &= -1 \end{aligned} \tag{4.34}$$

As before, the right-end boundary condition $V_\mathrm{P}(S,t) \to 0$ must be observed for $S \to \infty$.

For **American calls** analogous boundary conditions can be formulated. For a call in case $\delta > 0$, $r > 0$ the free boundary conditions

$$\begin{aligned} V_\mathrm{C}^\mathrm{Am}(S_\mathrm{f}(t), t) &= S_\mathrm{f}(t) - K \\ \frac{\partial V_\mathrm{C}^\mathrm{Am}(S_\mathrm{f}(t), t)}{\partial S} &= 1 \end{aligned} \tag{4.35}$$

must hold along the right-end boundary for $S_\mathrm{f}(t) > K$. The left-end boundary condition at $S = 0$ remains unchanged. Figure 4.9 shows the situation of an American call on a dividend-paying asset. The high contact on the payoff is visible.

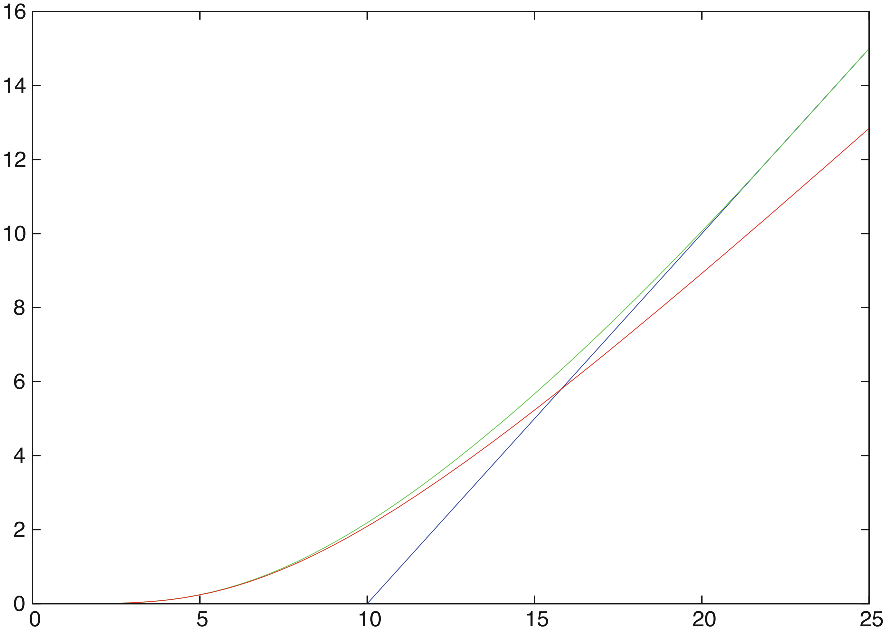

Fig. 4.9 Value $V(S,0)$ of an American call (in *green*) with $K = 10$, $r = 0.25$, $\sigma = 0.6$, $T = 1$ and dividend flow $\delta = 0.2$. The corresponding curve of a European call in *red*; the payoff in *blue*. A special calculated value is $V(K,0) = 2.18728$

We note in passing that the transformation $\zeta := S/S_{\mathrm{f}}(t)$, $y(\zeta, t) := V(S, t)$ allows to set up a Black–Scholes-type PDE on a rectangle. In this way, the unknown front $S_{\mathrm{f}}(t)$ is fixed at $\zeta = 1$, and is given implicitly by an ordinary differential equation as part of a nonlinear PDE ($\longrightarrow$ Exercise 4.8). Such a *front-fixing* approach is numerically relevant; see the Notes on Sect. 4.7.

4.5.3 Black–Scholes Inequality

The Black–Scholes equation (4.1) is valid on the continuation region (shaded areas in Fig. 4.6). For the numerical approach of the following Sect. 4.6 the computational domain will be the entire half strip $S > 0$, $0 \le t \le T$, *including* the stopping areas.[7] This will allow locating the early-exercise curve S_{f}. The approach requires to adapt the Black–Scholes equation in some way to the stopping areas.

To this end, define the Black–Scholes operator as

$$\mathcal{L}_{\mathrm{BS}}(V) := \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - \delta) S \frac{\partial V}{\partial S} - rV\,.$$

With this notation the Black–Scholes equation reads

$$\frac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V) = 0\,.$$

What happens with this operator on the stopping regions? To this end substitute the payoff function Ψ into $\frac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V)$. In the case of a put, for $S \le S_{\mathrm{f}}$, $V \equiv \Psi$ and

$$V = K - S\,,\quad \frac{\partial V}{\partial t} = 0\,,\quad \frac{\partial V}{\partial S} = -1\,,\quad \frac{\partial^2 V}{\partial S^2} = 0\,.$$

Hence

$$\frac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V) = -(r - \delta)S - r(K - S) = \delta S - rK\,.$$

Equation (4.32) implies the bound $\delta S < rK$, which leads to conclude

$$\frac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V) < 0\,.$$

[7] Up to localization.

That is, the Black–Scholes equation changes to an *inequality* on the stopping region. The same inequality holds for the call. (The reader may carry out the analysis for the case of a call.)

In summary, on the entire half strip $0 < S < \infty$, $0 < t < T$, American options must satisfy an inequality of the Black–Scholes type,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r-\delta)S\frac{\partial V}{\partial S} - rV \le 0\,. \tag{4.36}$$

Both inequalities (4.30) and (4.36) hold for all (S,t). In case the strict inequality "$>$" holds in (4.30), equality holds in (4.36). The contact boundary S_f divides the half strip into the stopping region and the continuation region, each with appropriate version of V:

$$\begin{aligned}
\text{put:}\quad & V_\mathrm{P}^\mathrm{Am} = K - S && \text{for } S \le S_\mathrm{f} && \text{(stop)}\\
& V_\mathrm{P}^\mathrm{Am} \text{ solves (4.1)} && \text{for } S > S_\mathrm{f} && \text{(hold)}\\
\text{call:}\quad & V_\mathrm{C}^\mathrm{Am} = S - K && \text{for } S \ge S_\mathrm{f} && \text{(stop)}\\
& V_\mathrm{C}^\mathrm{Am} \text{ solves (4.1)} && \text{for } S < S_\mathrm{f} && \text{(hold)}
\end{aligned}$$

This shows that the Black–Scholes equation (4.1) must be solved also for American options, however, with special arrangements because of the free boundary. We have to look for methods that simultaneously calculate V along with the unknown S_f.

Notice that $\frac{\partial V}{\partial S}$ is continuous when S crosses S_f, but $\frac{\partial^2 V}{\partial S^2}$ and $\frac{\partial V}{\partial t}$ are not continuous. It must be expected that this lack of smoothness along the early-exercise curve S_f affects the accuracy of numerical approximations.

4.5.4 Penalty Formulation

In this subsection we outline an approach that allows for a unified treatment of stopping region and continuation region. The inequality (4.36) can be written as an equality by introducing a *penalty* term $p(V) \ge 0$, and requesting

$$\frac{\partial V}{\partial t} + \mathcal{L}_\mathrm{BS}(V) + p(V) = 0\,. \tag{4.37}$$

The penalty term p should be zero for the continuation region, and should be positive for the stopping area. When calculating an approximation V, the distance to S_f is not known, but the distance $V - \Psi$ of V to the payoff Ψ is available and serves as decisive building block of a penalty term. There are several possibilities to construct a penalty p. One classic approach will be described in Sect. 7.2. Another way to set up a penalty can be accomplished by a term such as

$$p(V) := \frac{\epsilon}{V - \Psi} \quad \text{for a small } \epsilon > 0\,. \tag{4.38}$$

Let V^{ϵ} denote a solution of the penalty equation (4.37) with penalty function (4.38). Two extreme cases characterize the effect of the penalty term for (S, t) in the continuation area and in the stopping area:

- $V_\epsilon - \Psi \gg \epsilon$ implies $p \approx 0$. Then essentially the Black–Scholes equation results, and V_ϵ approximates the BS-solution.
- $0 < V_\epsilon - \Psi \ll \epsilon$ implies both $V_\epsilon \approx \Psi$ and a large value of p. The latter means that the BS-part of (4.37) is dominated by p; the BS equation is switched off.

The corresponding branches of the solution V_ϵ may be called the "continuation branch" ($p \approx 0$) and the "stopping branch" ($V_\epsilon \approx \Psi$). Obviously these two branches approximate the true solution V of the Black–Scholes problem. The intermediate range $V_\epsilon - \Psi \approx O(\epsilon)$ characterizes a boundary layer between the continuation branch and the stopping branch. In this layer around the early-exercise curve S_f the solution V_ϵ can be seen as a connection between the BS surface and the payoff plane.[8]

Notice that p and the resulting PDE are nonlinear in V, which complicates the numerical solution. The penalty formulation is advantageous especially in cases where an analysis of the early-exercise curve is difficult. See Sect. 6.7 for an exposition of the penalty approach in the two-dimensional situation. For the standard options of this chapter, we pursue another method, which effectively allows to preserve linearity.

4.5.5 Obstacle Problem

A brief digression into obstacle problems will motivate the procedure. We assume an "obstacle" $g(x)$, say with $g(x) > 0$ for a subinterval of $-1 < x < 1$, $g \in \mathcal{C}^2$, $g'' \leq 0$ and $g(-1) < 0,\ g(1) < 0$, compare Fig. 4.10. Across the obstacle a function u with minimal length is stretched like a rubber thread. Between $x = \alpha$ and $x = \beta$ the curve u clings to the boundary of the obstacle. For α and β we encounter high-contact conditions $u(\alpha) = g(\alpha)$, $u'(\alpha) = g'(\alpha)$, and $u(\beta) = g(\beta)$, $u'(\beta) = g'(\beta)$. Initially, the two values $x = \alpha$ and $x = \beta$ are unknown. This obstacle problem is a simple free-boundary problem.

The aim is to reformulate the obstacle problem such that the free boundary conditions do not show up explicitly. This may promise computational advantages. The function u shown in Fig. 4.10 is characterized by the requirements $u \geq g$,

[8] This is illustrated in Topic 9 of the *Topics fCF*.

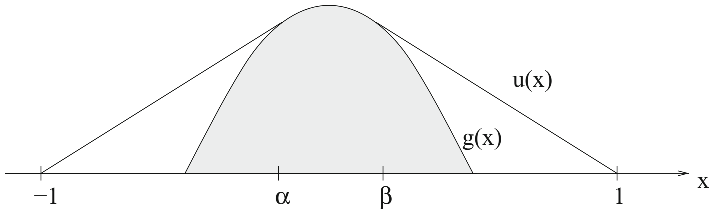

Fig. 4.10 Function $u(x)$ across an obstacle $g(x)$

$u(-1) = u(1) = 0$, $u \in \mathcal{C}^1$, and by: There is α, β such that

$$\begin{aligned} \text{for } -1 < x < \alpha: &\quad u'' = 0 \quad (u > g) \\ \text{for } \alpha < x < \beta: &\quad u = g \quad (u'' = g'' \le 0) \\ \text{for } \beta < x < 1: &\quad u'' = 0 \quad (u > g)\,. \end{aligned}$$

The characterization of the two outer intervals is identical. This manifests a complementarity in the sense

$$\begin{aligned} &\text{if } u - g > 0\ ,\ \text{then } u'' = 0\,; \\ &\text{if } u - g = 0\ ,\ \text{then } u'' \le 0\,. \end{aligned}$$

In retrospect it is clear that American options are complementary in an analogous way:

$$\begin{aligned} &\text{if} \quad V - \Psi > 0,\ \text{then Black–Scholes } \textit{equation}\ \frac{\partial V}{\partial t} + \mathcal{L}_{\text{BS}}(V) = 0\,; \\ &\text{if} \quad V - \Psi = 0,\ \text{then Black–Scholes } \textit{inequality}\ \frac{\partial V}{\partial t} + \mathcal{L}_{\text{BS}}(V) \le 0\,. \end{aligned}$$

This analogy motivates searching for a solution of the obstacle problem. The obstacle problem can be reformulated as

$$\begin{aligned} &\text{find a function } u \text{ such that} \\ &\quad u''(u-g) = 0\,, \quad -u'' \ge 0\,, \quad u - g \ge 0\,, \qquad (4.39) \\ &\quad u(-1) = u(1) = 0\,, \quad u \in \mathcal{C}^1[-1,1]\,. \end{aligned}$$

The key line (4.39) is a **linear complementarity problem** (LCP). This formulation does not mention the free boundary conditions at $x = \alpha$ and $x = \beta$ explicitly. This will be advantageous because α and β are unknown. After a solution to (4.39) is calculated, the values α and β are read off from the solution. To this end we construct a numerical solution procedure for the complementarity version (4.39) of the obstacle problem.

Discretization of the Obstacle Problem
A finite-difference approximation for u'' on the grid $x_i = -1 + i\Delta x$, with $\Delta x = \frac{2}{m}$, $g_i := g(x_i)$ leads to

$$\begin{aligned}&(w_{i-1} - 2w_i + w_{i+1})(w_i - g_i) = 0,\\ &-w_{i-1} + 2w_i - w_{i+1} \geq 0, \quad w_i \geq g_i\end{aligned}$$

for $0 < i < m$ and $w_0 = w_m = 0$. The w_i are approximations to $u(x_i)$. In view of the signs of the factors in the first line in this discretization scheme it can be written using a scalar product. To see this, define a vector notation using

$$G := \begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix} \text{ and } w := \begin{pmatrix} w_1 \\ \vdots \\ w_{m-1} \end{pmatrix}, \; g := \begin{pmatrix} g_1 \\ \vdots \\ g_{m-1} \end{pmatrix}.$$

Then the discretized complementarity problem is rewritten in the form

$$(w - g)^{tr} G w = 0\,, \quad Gw \geq 0\,, \quad w \geq g\,. \tag{4.40}$$

To calculate a solution of (4.40) one solves $Gw = 0$ under the side condition $w \geq g$. This will be explained in Sect. 4.6.4. In Sect. 5.3 we will return to the obstacle problem with a version as variational problem.

4.5.6 Linear Complementarity for American Put Options

In analogy to the simple obstacle problem described above we now derive a linear complementarity problem for American options. Here we confine ourselves to American puts without dividends ($\delta = 0$); the general case will be listed in Sect. 4.6. The transformations (4.3) lead to

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2} \text{ as long as } V_{\mathrm{P}}^{\mathrm{Am}} > (K - S)^+\,.$$

Also the side condition (4.30) is transformed: The relation

$$V_{\mathrm{P}}^{\mathrm{Am}}(S,t) \geq (K-S)^+ = K \max\{\,1-\mathrm{e}^x,\ 0\,\}$$

leads to the inequality

$$\begin{aligned} y(x,\tau) &\geq \exp\{\frac{1}{2}(q-1)x + \frac{1}{4}(q+1)^2\tau\}\max\{1-\mathrm{e}^x,\ 0\} \\ &= \exp\{\frac{1}{4}(q+1)^2\tau\}\max\{(1-\mathrm{e}^x)\mathrm{e}^{\frac{1}{2}(q-1)x},\ 0\} \\ &= \exp\{\frac{1}{4}(q+1)^2\tau\}\max\{\mathrm{e}^{\frac{1}{2}(q-1)x} - \mathrm{e}^{\frac{1}{2}(q+1)x},\ 0\} \\ &=:\, g(x,\tau)\,. \end{aligned}$$

This function g allows to write the initial condition (4.5)/(4.6) as $y(x,0) = g(x,0)$. In summary, we require $y_\tau = y_{xx}$ as well as

$$y(x,0) = g(x,0) \ \text{ and } \ y(x,\tau) \geq g(x,\tau)\,,$$

and, in addition, boundary conditions, and $y \in \mathcal{C}^1$ with respect to x. For $x \to \infty$ the function g vanishes, $g(x,\tau) = 0$, so the boundary condition $y(x,\tau) \to 0$ for $x \to \infty$ can be written

$$y(x,\tau) = g(x,\tau) \ \text{ for } \ x \to \infty\,.$$

The same holds for $x \to -\infty$ ($\longrightarrow$ Exercise 4.9). In the localizing practice, the boundary conditions are formulated for $x_{\min}$ and $x_{\max}$. Collecting all expressions, the American put is formulated as linear complementarity problem:

$$\begin{aligned} &\left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2}\right)(y-g) = 0\,, \\ &\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \geq 0\,, \qquad y - g \geq 0\,, \\ &y(x,0) = g(x,0), \quad y(x_{\min},\tau) = g(x_{\min},\tau)\,, \\ &y(x_{\max},\tau) = g(x_{\max},\tau)\,,\ y \in \mathcal{C}^1 \text{ with respect to } x\,. \end{aligned}$$

The exercise boundary is automatically captured by this formulation. An analogous formulation holds for the American call. Both of the formulations are comprised by Problem 4.7 below.

4.6 Computation of American Options

In the previous sections we have derived a linear complimentarity problem for both put and call of an American-style option. We summarize the results into Problem 4.7. This assumes for a put $r > 0$, and for a call $\delta > 0$; otherwise the American option is not distinct from the European counterpart.

Problem 4.7 (Linear Complementarity Problem)

Notations of (4.3), including

$$q = \frac{2r}{\sigma^2}, \quad q_\delta = \frac{2(r-\delta)}{\sigma^2},$$

$$\textit{put}: \; g(x,\tau) := \exp\{\frac{\tau}{4}((q_\delta - 1)^2 + 4q)\} \max\{\mathrm{e}^{\frac{x}{2}(q_\delta - 1)} - \mathrm{e}^{\frac{x}{2}(q_\delta + 1)}, 0\}$$

$$\textit{call}: \; g(x,\tau) := \exp\{\frac{\tau}{4}((q_\delta - 1)^2 + 4q)\} \max\{\mathrm{e}^{\frac{x}{2}(q_\delta + 1)} - \mathrm{e}^{\frac{x}{2}(q_\delta - 1)}, 0\}$$

$$\left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2}\right)(y - g) = 0$$

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

$$x_{\min} \leq x \leq x_{\max}, \quad 0 \leq \tau \leq \frac{1}{2}\sigma^2 T$$

$$y(x,0) = g(x,0)$$

$$y(x_{\min},\tau) = g(x_{\min},\tau), \quad y(x_{\max},\tau) = g(x_{\max},\tau)$$

As outlined in Sect. 4.5, the free boundary problem of American options is described in Problem 4.7 such that the free boundary condition does not show up explicitly. We now enter the discussion of how to solve Problem 4.7 numerically.

4.6.1 Discretization with Finite Differences

We use the same grid as in Sect. 4.2.2, with $w_{i,\nu}$ denoting an approximation to $y(x_i, \tau_\nu)$, and $g_{i,\nu} := g(x_i, \tau_\nu)$ for $0 \leq i \leq m$, $0 \leq \nu \leq \nu_{\max}$. The backward difference, the explicit, and the Crank–Nicolson method can be combined into one formula,

$$\begin{aligned}\frac{w_{i,\nu+1} - w_{i,\nu}}{\Delta\tau} = \; & \theta \frac{w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1}}{\Delta x^2} + \\ & (1-\theta)\frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{\Delta x^2},\end{aligned}$$

with the choices $\theta = 0$ (explicit), $\theta = \frac{1}{2}$ (Crank–Nicolson), $\theta = 1$ (backward-difference method). This family of numerical schemes parameterized by θ is often called *θ-method.*

The differential inequality $\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \geq 0$ becomes the discrete version

$$\begin{aligned} & w_{i,\nu+1} - \lambda\theta(w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1}) \\ & \qquad - w_{i,\nu} - \lambda(1-\theta)(w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}) \geq 0\,, \end{aligned} \tag{4.41}$$

again with the abbreviation $\lambda := \frac{\Delta\tau}{\Delta x^2}$. With the notations

$$b_{i,\nu} := w_{i,\nu} + \lambda(1-\theta)(w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu})\ ,\ i = 2, \ldots, m-2$$

$b_{1,\nu}$ and $b_{m-1,\nu}$ incorporate the boundary conditions

$$b^{(\nu)} := (b_{1,\nu}, \ldots, b_{m-1,\nu})^{tr}$$

$$w^{(\nu)} := (w_{1,\nu}, \ldots, w_{m-1,\nu})^{tr}$$

$$g^{(\nu)} := (g_{1,\nu}, \ldots, g_{m-1,\nu})^{tr}$$

and

$$A := \begin{pmatrix} 1+2\lambda\theta & -\lambda\theta & & 0 \\ -\lambda\theta & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix} \in \mathbb{R}^{(m-1)\times(m-1)} \tag{4.42}$$

(4.41) is rewritten in vector form as

$$Aw^{(\nu+1)} \geq b^{(\nu)} \ \text{ for all } \nu\,.$$

Such inequalities for vectors are understood componentwise. The inequality $y - g \geq 0$ leads to

$$w^{(\nu)} \geq g^{(\nu)}\,,$$

and $\left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2}\right)(y-g) = 0$ becomes

$$\left(Aw^{(\nu+1)} - b^{(\nu)}\right)^{tr}\left(w^{(\nu+1)} - g^{(\nu+1)}\right) = 0\,.$$

The initial and boundary conditions are

$$\begin{aligned} & w_{i,0} = g_{i,0}\,, \quad i = 1, \ldots, m-1\,, \quad (w^{(0)} = g^{(0)})\,; \\ & w_{0,\nu} = g_{0,\nu}\,, \quad w_{m,\nu} = g_{m,\nu}\,, \quad \nu \geq 1\,. \end{aligned}$$

Boundary conditions are realized in the vectors $b^{(\nu)}$ as follows:

$$\begin{aligned}
&b_{2,\nu}, \ldots, b_{m-2,\nu} \quad \text{as defined above,} \\
&b_{1,\nu} = w_{1,\nu} + \lambda(1-\theta)(w_{2,\nu} - 2w_{1,\nu} + g_{0,\nu}) + \lambda\theta g_{0,\nu+1} \\
&b_{m-1,\nu} = w_{m-1,\nu} + \lambda(1-\theta)(g_{m,\nu} - 2w_{m-1,\nu} + w_{m-2,\nu}) + \lambda\theta g_{m,\nu+1}
\end{aligned} \tag{4.43}$$

We summarize the discrete version of the Problem 4.7 into an Algorithm:

Algorithm 4.8 (Computation of American Options)

```
For $\nu = 0, 1, \ldots, \nu_{\max} - 1$ :
Calculate the vectors $g := g^{(\nu+1)}$,
  $b := b^{(\nu)}$ from (4.42), (4.43).
Calculate the vector $w$ as solution of the problem
  $Aw - b \geq 0\,, \quad w \geq g\,, \quad (Aw - b)^{tr}(w - g) = 0\,.$        (4.44)
$w^{(\nu+1)} := w$
```

This completes the chosen finite-difference discretization.

The remaining problem is to solve the complementarity problem in matrix-vector form (4.44). In principle, how to solve (4.44) is a new topic independent of the discretization background. But accuracy and efficiency will depend on the context of selected methods. We pause for a moment to become aware how broad the range of possible finite-difference methods is.

There are possible sources of inaccuracies. The payoff is not smooth. And recall from Sect. 4.5.3 that $V(S,t)$ is not C^2-smooth over the free boundary S_f. Second-order convergence of the basic Crank–Nicolson scheme must be expected to be deteriorated. The effect caused by lacking smoothness depends on the choice of several items, namely, the

(1) kind of transformation/PDE [from no transformation over a mere $\tau := T - t$ to the transformation (4.3)],
(2) kind of discretization (from backward-difference over Crank–Nicolson to more refined schemes like BDF2),
(3) method of solution for (4.44).

The latter can be a direct elimination method, or an iteratively working indirect method. Large systems as they occur in PDE context are frequently solved iteratively, in particular in high-dimensional spaces. Such approaches sometimes benefit from smoothing properties. Both an iterative procedure (following [376]) and a direct approach (following [52]) will be discussed below. It turns out that in the one-dimensional scenario of this chapter (one underlying asset), the direct approach is faster.

4.6.2 Reformulation and Analysis of the LCP

In each time level ν in Algorithm 4.8, a linear complementarity problem (4.44) must be solved. This is the bulk of work in Algorithm 4.8. Before entering a numerical solution, we analyze the LCP. Since this subsection is general numerical analysis independent of the finance framework, we momentarily use vectors x, y, r freely in other context.[9] For the analysis transform problem (4.44) from the w-world into an x-world with

$$\begin{aligned} x &:= w - g\,, \\ y &:= Aw - b\,. \end{aligned} \tag{4.45}$$

Then it is easy to see (the reader may check) that the task of calculating a solution w for (4.44) is equivalent to the following problem:

Problem 4.9 (Cryer) *Compute vectors x and y such that for $\hat{b} := b - Ag$*

$$Ax - y = \hat{b}\,, \quad x \geq 0\,, \quad y \geq 0\,, \quad x^{tr} y = 0\,. \tag{4.46}$$

First we make sure that the above problem has a unique solution. To this end one shows the equivalence of Problem 4.9 with a minimization problem.

Lemma 4.10 *Problem 4.9 is equivalent to the minimization problem*

$$\min_{x \geq 0} G(x), \quad \textit{with } G(x) := \frac{1}{2}(x^{tr} A x) - \hat{b}^{tr} x\,, \tag{4.47}$$

where G is strictly convex.

Proof The derivatives of G are $G_x = Ax - \hat{b}$ and $G_{xx} = A$. Lemma 4.3 implies that A has positive eigenvalues. Hence the Hessian matrix G_{xx} is symmetric and positive definite. So G is strictly convex, and has a unique minimum on each convex set in $\mathbb{R}^n$, for example on $x \geq 0$. The Theorem of Karush, Kuhn and Tucker minimizes G under $H_i(x) \leq 0$, $i = 1, \ldots, m$. According to this theorem,[10] a vector x_0 to be a minimum is equivalent to the existence of a Lagrange multiplier $y \geq 0$ with

$$\text{grad}\, G(x_0) + \left(\frac{\partial H(x_0)}{\partial x}\right)^{tr} y = 0\,, \quad y^{tr} H(x_0) = 0\,.$$

[9] Notation: In this Sect. 4.6.2, x does not have the meaning of transformation (4.3), and r not that of an interest rate, and y is no PDE solution. Here, $x, y \in \mathbb{R}^{m-1}$.

[10] For the KKT (Karush-Kuhn-Tucker or Kuhn-Tucker) theory we refer to [348, 350]. In our context, $m - 1$.

The set $x \geq 0$ leads to define $H(x) := -x$. Hence the KKT condition is $Ax - \hat{b} + (-I)^{tr} y = 0$, $y^{tr} x = 0$, and we have reached Eq. (4.46). □

4.6.3 Iterative Procedure for the LCP

An iterative procedure can be derived from the minimization problem stated in Lemma 4.10. This algorithm is based on the SOR[11] method [92]. Note that (4.44) is not in the easy form of equation $Ax = b$ discussed in Appendix C.2; a modification of the standard SOR will be necessary. The iteration of the SOR method for $Ax = \hat{b} = b - Ag$ is written componentwise (⟶ Exercise 4.10) as iteration for the correction vector $x^{(k)} - x^{(k-1)}$:

$$r_i^{(k)} := \hat{b}_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - a_{ii} x_i^{(k-1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} , \tag{4.48}$$

$$x_i^{(k)} = x_i^{(k-1)} + \omega_{\mathrm{R}} \frac{r_i^{(k)}}{a_{ii}} . \tag{4.49}$$

Here k denotes the number of the iteration, $n = m-1$, and a_{ij} is element of the matrix A. In the cases $i = 1$, $i = m - 1$ one of the sums in (4.48) is empty. The *relaxation parameter* ω_{R} is a factor chosen in a way that should improve the convergence of the iteration. The "projected" SOR method for solving (4.46) starts from a vector $x^{(0)} \geq 0$ and is identical to the SOR method up to a modification on (4.49) serving for $x_i^{(k)} \geq 0$.

Algorithm 4.11 (PSOR, Projected SOR for Problem 4.9)

```
*outer loop* :  $k = 1, 2, \ldots$
*inner loop* :  $i = 1, \ldots, m-1$
                $r_i^{(k)}$  *as in (4.48)*,
                $x_i^{(k)} = \max \left\{ 0,\; x_i^{(k-1)} + \omega_{\mathrm{R}} \frac{r_i^{(k)}}{a_{ii}} \right\}$ ,
                $y_i^{(k)} = -r_i^{(k)} + a_{ii} \left( x_i^{(k)} - x_i^{(k-1)} \right)$ .
```

[11] Successive overrelaxation, SOR. For an introduction to classic iterative methods for the solution of systems of linear equations $Ax = b$ we refer to Appendix C.2.

This algorithm solves $Ax = \hat{b}$ for $\hat{b} = b - Ag$ iteratively by *componentwise* considering $x^{(k)} \geq 0$. The vector y or the components $y_i^{(k)}$ converging to y_i, are not used explicitly for the algorithm. But since $y \geq 0$ is shown ($Aw \geq b$), the vector y plays an important role in the proof of convergence. Transformed back into the w-world of problem (4.44) by means of (4.45), the Algorithm 4.11 solves (4.44).

A proof of the convergence of Algorithm 4.11 is based on Lemma 4.10. One shows that the sequence defined in Algorithm 4.11 minimizes G. The main steps of the argumentation are sketched as follows:

For $0 < \omega_{\mathrm{R}} < 2$ the sequence $G(x^{(k)})$ is decreasing monotonically;

Show $x^{(k+1)} - x^{(k)} \to 0$ for $k \to \infty$;

The limit exists because $x^{(k)}$ moves in a compact set $\{x \mid G(x) \leq G(x^{(0)})\}$;

The vector r from (4.48) converges toward $-y$;

Assuming $r \geq 0$ and $r^{tr}x \neq 0$ leads to a contradiction to $x^{(k+1)} - x^{(k)} \to 0$. (For the proof see [92].)

4.6.4 Direct Method for the LCP

Another formulation has shown to be a basis for a direct solution by elimination:

Problem 4.12 (Cryer's Problem Restated)

Solve $Aw = b$ componentwise such that
the side condition $w \geq g$ is obeyed.

An implementation must be done carefully such that the boundary conditions and all the LCP requirements in (4.46) are met. The structure of Problem 4.12 is different from the system $Aw = b$ without side condition [201].

Recall that a direct method to solve a system $Aw = b$ of linear equations establishes in a first phase an equivalent system $\tilde{A}w = \tilde{b}$ with a triangular matrix $\tilde{A}$ (here bidiagonal since A is tridiagonal). After $\tilde{A}, \tilde{b}$ are calculated, the second phase (the solution of $\tilde{A}w = \tilde{b}$) is established in a single loop. For an upper (right) triangular matrix $\tilde{A}$ the fist phase is a forward loop, and the subsequent second phase is backward. This is the familiar form of Gauss elimination, which in this context may be called "forward-backward method."

Less familiar is the opposite procedure: In order to establish $\tilde{A}$ as *lower* triangular matrix, the first phase creates zeroes above the diagonal, and hence is done in the backward fashion. The second phase then solves $\tilde{A}w = \tilde{b}$ in a forward loop. This is the backward-forward version of Gauss elimination. In our context of solving Problem 4.12 both versions are not equivalent; renumbering the equations and variables does not help. This is caused by the side condition $w \geq g$, which adds a nonlinearity, with different conditions on w_1 and w_m.

When in the second phase in the ith step of the solution loop $\tilde{w}_i$ is a component of the solution of $\tilde{A}w = \tilde{b}$, then $w_i := \max\{\tilde{w}_i, g_i\}$ might appear the correct value.

But w depends on the orientation of the loop, backward or forward. Only one direction works. An implementation must make sure that the characteristic order of the underlying option is preserved. For a **put** this means:

> We denote i_f the index of the node S_i, which is closest to the contact point.[12] The index i_f marks the location of the free boundary. More precisely,
> $w_i = g_i$ for $1 \le i \le i_\mathrm{f}$, and
> $w_i > g_i$ for $i_\mathrm{f} < i \le m$.

This structure is characteristic for a put, but the index i_f is unknown. For a put the start is $w_1 = g_1$, and the $w_i := \max\{\tilde{w}_i, g_i\}$-loop must be *forward*. Accordingly, for a put, $\tilde{A}$ must be a lower triangular matrix, and hence the backward-forward variant of Gauss elimination is applied. This amounts to an *RL*-decomposition of A ($\longrightarrow$ Appendix C.1). The lower triangle $\tilde{A} := L$ is established, and the vector $\tilde{b}$ obtained by solving $R\tilde{b} = b$.

Algorithm 4.13 (American Put)

```
first phase:
    Calculate the RL-decomposition of A.
    Then set $\tilde{A} = L$ and calculate $\tilde{b}$ from $R\tilde{b} = b$ (backward loop).
second phase: forward loop for growing i:
    Start with $i = 1$. Calculate the next component of $\tilde{A}w = \tilde{b}$; denote it $\tilde{w}_i$.
    Set $w_i := \max\{\tilde{w}_i, g_i\}$.
```

This procedure was suggested by Brennan and Schwartz [52]. Since the matrix A from (4.42) is tridiagonal, the costs are low. In this way, a direct method for solving Problem 4.12 is established, which is as efficient as solving a standard system of linear equations. ($\longrightarrow$ Exercise 4.11) The elegant approach of Algorithm 4.13 allows to treat the nonlinear problem of valuing an American option as if it were linear.

For a call $w_i = g_i$ holds for large indices i, and the elimination phase runs in a backward loop. This requires the traditional upper triangular matrix $\tilde{A}$ as calculated by the *LR*-decomposition ($\longrightarrow$ Exercise 4.12). For both put and call there is only one index i_f separating the components with $w_i = g_i$ from those with $w_i > g_i$.

4.6.5 An Algorithm for Calculating American Options

We return to the original meaning of the variables x, y, r, as used for instance in (4.2), (4.3). It remains to substitute a proper algorithm for solving (4.44) in Algorithm 4.8. From the analysis of Sect. 4.6.2, we either apply the iterative Algorithm 4.11 ($\longrightarrow$ Exercise 4.13), or implement the fast direct method of

[12] The S-interval must be large enough, $S_1 < S_\mathrm{f}$.

Algorithm 4.13. The resulting algorithm is formulated in Algorithm 4.14 with an LCP-solving module. The implementation of the direct version is left to the reader (⟶ Exercise 4.11). Recall $g_{i,\nu} := g(x_i, \tau_\nu)$ $(0 \le i \le m)$ and $g^{(\nu)} := (g_{1,\nu}, \ldots, g_{m-1,\nu})^{tr}$. Figure 4.11 depicts a result of Algorithm 4.14 for Example 1.6. Here we obtain the contact point with value $S_{\mathrm{f}}(0) = 36.16$ (with $m = \nu_{\max} = 1600$). Figure 4.13 shows the American put that corresponds to the call in Fig. 4.9.

Algorithm 4.14 (Prototype Core Algorithm)

```
Set up the function $g(x, \tau)$ listed in Problem 4.7.
Choose $\theta$ ($\theta = 1/2$ for Crank–Nicolson).
For PSOR: choose $1 \le \omega_{\mathrm{R}} < 2$ (for example, $\omega_{\mathrm{R}} = 1$),
    fix an error bound $\varepsilon$ (for example, $\varepsilon = 10^{-5}$).
Fix the discretization by choosing $x_{\min},\ x_{\max},\ m,\ \nu_{\max}$
(for example, $x_{\min} = -5,\ x_{\max} = 5$ or $3,\ \nu_{\max} = m = 100$).
Calculate $\Delta x := (x_{\max} - x_{\min})/m$,
  $\Delta\tau := \frac{1}{2}\sigma^2 T/\nu_{\max}$,
  $x_i := x_{\min} + i\Delta x$ for $i = 0, \ldots, m$.
Initialize the iteration vector $w$ with
  $g^{(0)} = (g(x_1, 0), \ldots, g(x_{m-1}, 0))$.
Calculate $\lambda := \Delta\tau/\Delta x^2$ and $\alpha := \lambda\theta$.
(Now all elements of matrix $A$ from (4.42) are defined.)
$\tau$ − loop : for $\nu = 0, 1, \ldots, \nu_{\max} - 1$ :
  $\tau_\nu := \nu\Delta\tau$
  $b_i := w_i + \lambda(1-\theta)(w_{i+1} - 2w_i + w_{i-1})$ for $2 \le i \le m-2$
  $b_1 := w_1 + \lambda(1-\theta)(w_2 - 2w_1 + g_{0,\nu}) + \alpha g_{0,\nu+1}$
  $b_{m-1} := w_{m-1} + \lambda(1-\theta)(g_{m,\nu} - 2w_{m-1} + w_{m-2}) + \alpha g_{m,\nu+1}$
  Module: Calculate the LCP solution $w$ of Problem 4.12,
    preferably by direct elimination as Algorithm 4.13, Exercise 4.11, or
    alternatively by implementing an iterative method as Algorithm 4.11.
  $w^{(\nu+1)} = w$.
```

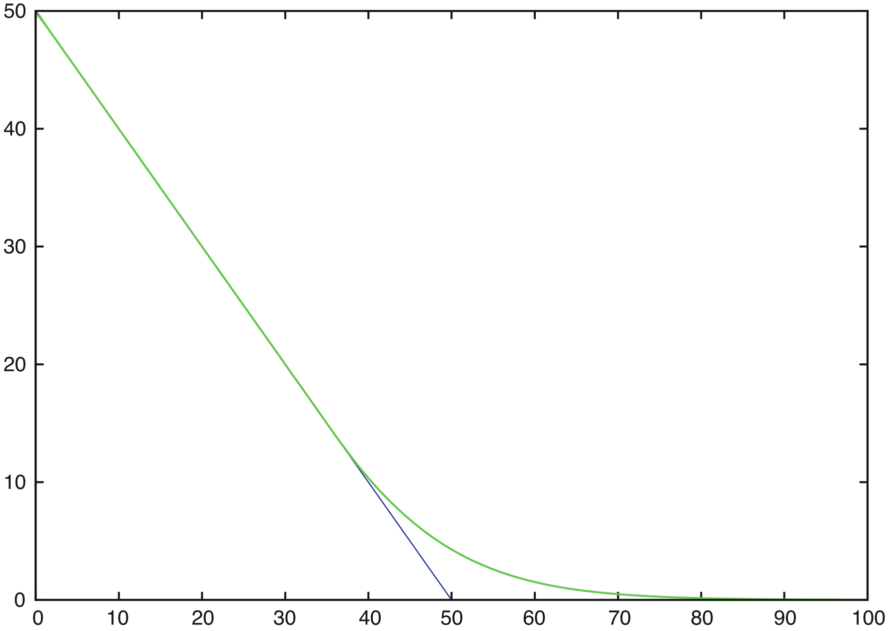

Fig. 4.11 Example 1.6: American put, $K = 50$, $r = 0.1$, $\sigma = 0.4$, $T = \frac{5}{12}$. $V(S, 0)$ (*green curve*) and payoff $V(S, T)$ (*blue*). Special calculated value: $V(K, 0) = 4.2842$

4.6.5.1 Valuing Options

For completeness we mention that it is possible to calculate European options with Algorithm 4.14 after simple modifications, which recover standard methods for solving $Aw = b$ (without $w \geq g$). If in addition the boundary conditions are adapted, then the computer program resulting from Algorithm 4.14 can be applied to European options. Of course, applying the analytic solution formula (A.15) or (A.17) should be most economical, when the entire surface $V(S, t)$ is not required. But for the purpose of testing Algorithm 4.14 it is recommendable to compare its results to something "known."

Back to American options, we complete the analysis, summarizing how a concrete financial task is solved with the core Algorithm 4.14, which is formulated in artificial variables such as $x_i, g_{i,\nu}, w_i$ and not in financial variables. This requires an interface between the real world and the core algorithm. The interface is provided by the transformations in (4.3). This important ingredient must be included for completeness. Let us formulate the required transition between the real world and the numerical machinery of Algorithm 4.14 as another algorithm:

Algorithm 4.15 (American Options)

```
Input: strike $K$, time to expiration $T$, spot price $S_0$, $r, \delta, \sigma$ .
Perform the core Algorithm 4.14.
    (The $\tau$-loop ends at $\tau_{\text{end}} = \frac{1}{2}\sigma^2 T$.)
For $i = 1, \dots, m-1$ :
  $w_i$ approximates $y(x_i, \frac{1}{2}\sigma^2 T)$,
  $S_i = K \exp\{x_i\}$
  $V(S_i, 0) = K w_i \exp\{-\frac{x_i}{2}(q_\delta - 1)\} \exp\{-\tau_{\text{end}}(\frac{1}{4}(q_\delta - 1)^2 + q)\}$
Test for early exercise: Approximate $S_{\text{f}}(0)$ and compare to $S_0$ .
```

For the direct method, an approximation for $S_{\text{f}}(0)$ is readily available via $S_{i_{\text{f}}}$. An indirect method checks the closeness of V to the payoff:

$$\text{Choose a small } \varepsilon^* > 0, \text{ for example, } \varepsilon^* = K \cdot 10^{-5}.$$

$$i_{\text{f}} := \max\{\, i \mid \ |V(S_i, 0) + S_i - K| < \varepsilon^* \,\} \text{ for a put,}$$

$$i_{\text{f}} := \min\{\, i \mid \ |K - S_i + V(S_i, 0)| < \varepsilon^* \,\} \text{ for a call.}$$

Criterion $S_0 < S_{i_{\text{f}}}$ indicates the stopping region for a put; for a call, this indication is $S_0 > S_{i_{\text{f}}}$.

Algorithm 4.15 evaluates the data at the final time level τ_{end}, which corresponds to $t = 0$. The computed information for the intermediate time levels can be evaluated analogously. In this way, the locations of $S_{i_{\text{f}}}$ can be put together to form an approximation of the free-boundary or stopping-time curve $S_{\text{f}}(t)$. But note that this approximation will be a crude step function. It requires some effort to calculate the curve $S_{\text{f}}(t)$ with reasonable accuracy, see the illustration of curve c_1 in Fig. 4.8 ($\longrightarrow$ Exercise 4.14).

4.6.5.2 Modifications

The above Algorithm 4.14 (along with Algorithm 4.15) is the prototype of a finite-difference algorithm. Improvements are possible. For example, the equidistant time step $\Delta\tau$ can be given up in favor of a variable time stepping. A few very small time steps initially will help to quickly damp the influence of the nonsmooth payoff. The effect of the kink of the payoff at the strike K is illustrated by Fig. 4.12. The turmoil at the corner is seen, but also the relatively rapid smoothing within a few

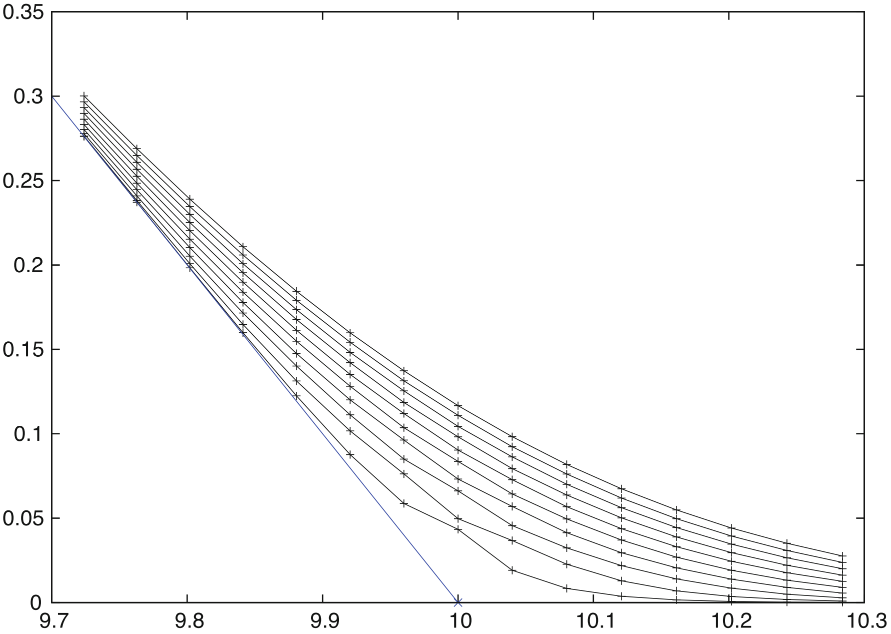

Fig. 4.12 Finite differences, Crank–Nicolson; American put with $r = 0.06$, $\sigma = 0.3$, $T = 1$, $K = 10$; $M = 1000$, $x_{\min} = -2$, $x_{\max} = 2$, $\Delta x = 1/250$, $\Delta t = 1/1000$, payoff (in *blue*) and $V(S, t_\nu)$ for $t_\nu = 1 - \nu\Delta t$, $\nu = 1, \ldots, 10$

time steps. Figure 4.12 shows explicitly the dependence of V on S; implicit in the figure is the dependence on t with corresponding oscillations. The effect of the lack of smoothness is heavier in case the payoff is discontinuous (binary option). In such a context it is advisable to start with a few fully implicit backward time steps ($\theta = 1$) before switching to Crank–Nicolson ($\theta = 1/2$). Such a procedure is called Rannacher stepping, see [305, 310], and the Notes on Sect. 4.3. After one run of the algorithm it is advisable to refine the initial grid to have a possibility to control the error. This simple strategy will be discussed in some more detail in Sect. 4.7.

Practical experience with boundary conditions (4.27) suggests working with $S_{\min} = 0.05$ and $S_{\max} = 5K$. For the transformation (4.3) $S = Ke^x$ this amounts to $x_{\min} = -3 - \log K$, $x_{\max} = 1.6$. This is to be modified for other transformations, see for instance the choice in Fig. 7.4.

4.6.5.3 Sensitivities

The greeks delta, gamma, theta are easily obtained by difference quotients. These approximations are formed by the V-values that were calculated on the finite-difference grid. For vega and rho, a recalculation is necessary, see Sect. 1.4.6.

In general, the comparably expensive solving of appropriate PDEs will not be necessary ($\longrightarrow$ Exercise 4.16).

4.7 On the Accuracy

Necessarily, each result obtained with the means of this chapter is subjected to errors in several ways. The most important errors have been mentioned earlier; in this section we collect them. Let us emphasize again that in general the *existence* of errors must be accepted, but not their magnitude. By investing sufficient effort, many of the errors can be kept at a tolerable level.

(a) **modeling error**
The assumptions defining the underlying financial model are restrictive. The Assumptions 1.2, for example, will not exactly match the reality of a financial market. Similarly this holds for other models. And the parameters of the models (such as volatility σ) are unknown and must be estimated. Hence the equations of the model are only crude approximations of "reality."

(b) **discretization errors**
Under the heading "discretization error" we summarize several errors that are introduced when the continuous PDE is replaced by a set of approximating equations defined on a grid. An essential portion of the discretization error is the difference between differential quotients and difference quotients. For example, a Crank–Nicolson discretization error is of the order $O(\Delta^2)$, if Δ is a measure of the grid size, and if the solution function is sufficiently smooth. Other discretization errors include the localization error caused by truncating the infinite interval $-\infty < x < \infty$ to a finite interval, or the implementation of the boundary conditions, or a quantification error when the strike ($x = 0$) is not part of the grid. In passing we recommend that the strike be one of the grid points, $x_k = 0$ for one k.

(c) **error from solving the linear equation**
An iterative solution of the linear systems of equation $Aw = b$ means that the error approaches 0 when $k \to \infty$, where k counts the number of iterations. By practical reasons the iteration must be terminated at a finite $k_{\max}$ such that the effort is bounded. Hence an error remains from the linear equations. The error tends to be neglectable for direct elimination methods.

(d) **rounding error**
The finite number of digits l of the mantissa is the reason for rounding errors.

In general, one has no *accurate* information on the size of these errors. Typically, the modeling errors are much larger than the discretization errors. In practice, in view of the uncertainties of modeling, it would be questionable to strive for an extremely small discretization error. For a stable method, the rounding errors are the least problem. The numerical analyst, as a rule, has limited potential in manipulating the modeling error. So the numerical analyst concentrates on the other errors, especially on discretization errors. To this end we may use the qualitative assertion of Theorem 4.4. But such an a priori result is only a basic step toward our ultimate goal formulated in Problem 4.16.

4.7.1 Elementary Error Control

Here we neglect modeling errors and try to solve the a posteriori error problem:

Problem 4.16 (Principle of an Error Control) *Let the exact result of a solution of the continuous equations be denoted η^*. The approximation η calculated by a given algorithm depends on a representative grid size Δ, on $k_{\max}$, on the word length l of the computer, and maybe on several additional parameters, symbolically written*

$$\eta = \eta(\Delta, k_{\max}, l)\,.$$

Choose $\Delta, k_{\max}, l$ such that the absolute error of η does not exceed a prescribed error tolerance ϵ,

$$|\eta - \eta^*| < \epsilon\,.$$

This problem is difficult to solve, because we implicitly assume an *efficient* approximation avoiding an overkill with extremely small values of Δ or large values of $k_{\max}$ or l. Time counts in real-time application. So we try to avoid unnecessary effort of achieving a tiny error $|\eta - \eta^*| \ll \epsilon$. The exact size of the error is unknown. But its order of magnitude can be estimated as follows.

Let us assume the method is of order p. We simplify this statement to

$$\eta(\Delta) - \eta^* = \gamma \Delta^p\,. \tag{4.50}$$

Here γ is a priori unknown. By calculating two approximations, say for grid sizes Δ_1 and Δ_2, the constant γ can be calculated. To this end subtract the two calculated approximations η_1 and η_2,

$$\eta_1 := \eta(\Delta_1) = \gamma \Delta_1^p + \eta^*$$
$$\eta_2 := \eta(\Delta_2) = \gamma \Delta_2^p + \eta^*$$

to obtain

$$\gamma = \frac{\eta_1 - \eta_2}{\Delta_1^p - \Delta_2^p}\,.$$

A simple choice of the grid size Δ_2 for the second approximation is the refinement $\Delta_2 = \frac{1}{2}\Delta_1$. This leads to

$$\gamma \left(\frac{\Delta_1}{2}\right)^p = \frac{\eta_1 - \eta_2}{2^p - 1} \, . \tag{4.51}$$

Especially for $p = 2$ the relation

$$\gamma \Delta_1^2 = \frac{4}{3}(\eta_1 - \eta_2)$$

results. In view of the scenario (4.50) the absolute error of the approximation η_1 is given by

$$\frac{4}{3}|\eta_1 - \eta_2|$$

and the error of η_2 by (4.51).

The above procedure does not guarantee that the error η is bounded by ϵ. This flaw is explained by the simplification in (4.50), and by neglecting the other type of errors of the above list (b)–(c). Here we have assumed γ constant, which in reality depends on the parameters of the model, for example, on the volatility σ. But testing the above rule of thumb (4.50)/(4.51) on European options shows that it works reasonably well. Here we compare the finite-difference results to the analytic solution formulas (A.15)/(A.17), the numerical errors of which are comparatively negligible. The procedure works similar well for American options, although then the function $V(S,t)$ is not $\mathcal{C}^2$-smooth at $S_\text{f}(t)$. (The effect of the lack in smoothness is similar as in Fig. 4.12.) In practical applications of Crank–Nicolson's method one can observe quite well that doubling of m and ν_max decreases the absolute error approximately by a factor of four. To obtain a minimum of information on the error, the core Algorithm 4.14 should be applied at least for two grids following the lines outlined above. The information on the error can be used to match the grid size Δ to the desired accuracy.

Let us illustrate the above considerations with an example, compare Figs. 4.13 and 4.14, and Table 4.1. For an American put and $x_\text{max} = -x_\text{min} = 5$ we calculate several approximations, and test Eq. (4.50) in the form $\eta(\Delta) = \eta^* + \gamma\Delta^2$. We illustrate the approximations as points in the (Δ^2, η)-plane. The better the assumption (4.50) is satisfied, the closer the calculated points lie on a straight line. Figure 4.14 suggests that this error-control model can be expected to work well.

In order to check the error quality of a computer program on standard American options, one may check the put-call symmetry relation (A.23). For example, for the parameters of Fig. 4.13/Table 4.1, the corresponding call with $S = K$ and switched parameters $r = 0.2$, $\delta = 0.25$ is calculated, and the results match very well: For the finest discretization in Table 4.1, about 8 digits match with the value of the

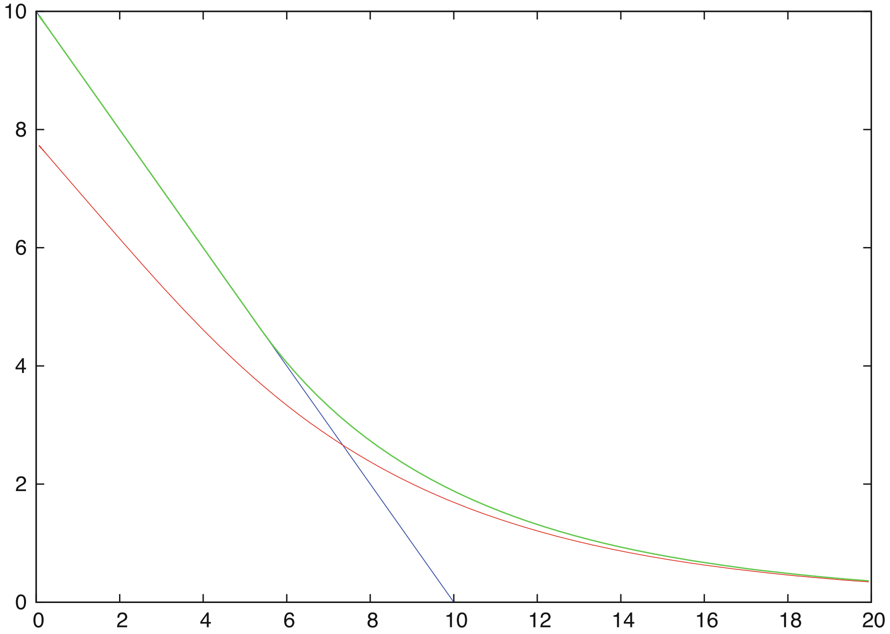

Fig. 4.13 Value $V(S, 0)$ of an American put (in *green*) with $K = 10$, $r = 0.25$, $\sigma = 0.6$, $T = 1$ and dividend flow $\delta = 0.2$. For special values see Table 4.1. The corresponding curve of a European option in *red*, the payoff in *blue*

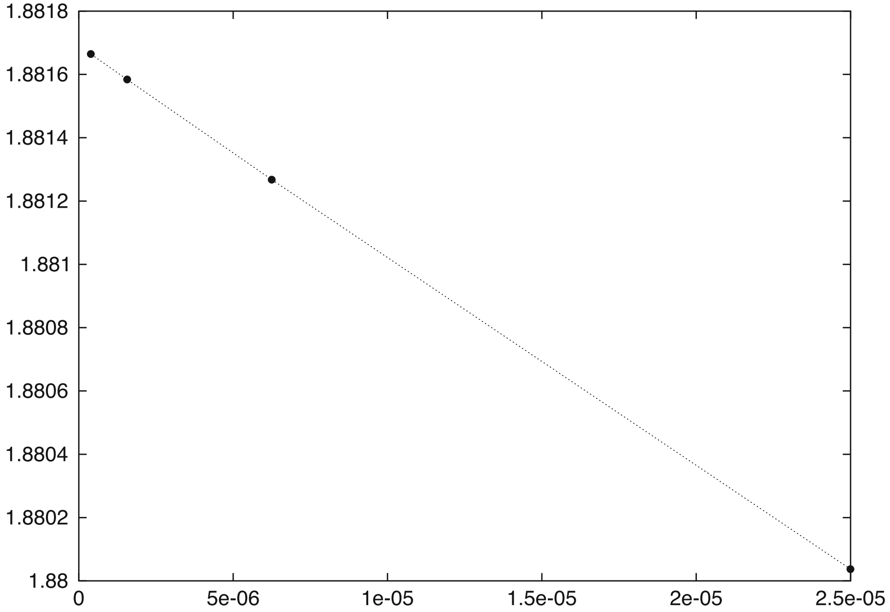

Fig. 4.14 Approximations V depending on Δ^2, with $\Delta = (x_{\max} - x_{\min})/m = 1/\nu_{\max}$; results of Fig. 4.13 and Table 4.1

Table 4.1 Results reported in Fig. 4.13

$m = \nu_{\max}$	$V(10,0)$
50	1.8562637
100	1.8752110
200	1.8800368
400	1.8812676
800	1.8815842
1600	1.8816652

corresponding call. But this is only a necessary criterion for accuracy; the number of matching digits of (A.23) does not relate to the number of correct digits of $V(S, 0)$.

4.7.2 *Extrapolation*

The obviously reasonable error model sketched above suggests applying (4.50) to obtain an improved approximation η at practically zero cost. Such a procedure is called *extrapolation* (⟶ Exercise 1.11). In a graphical illustration η over Δ^2 as in Fig. 4.14, extrapolation amounts to construct a straight line through two of the calculated points. The value of the straight line for $\Delta^2 = 0$ gives the extrapolated value from

$$\eta^* \approx \frac{4\eta_2 - \eta_1}{3}\,. \tag{4.52}$$

In our example, this procedure allows to estimate the correct value to be close to 1.8817. Combining, for example, two approximations of rather low quality, namely, $m = 50$ with $m = 100$, gives already an extrapolated approximation of 1.8815. And based on the two best approximations of Table 4.1, the extrapolated approximation is 1.881690.[13]

Typically, the extrapolation formula provided by (4.52) is significantly more accurate than η_2. But we have no further information on the accuracy of η_2 from the calculated η_1, η_2. Calculating a third approximation η_3 reveals more information. For example, a higher-order extrapolation can be constructed (⟶ Exercise 4.15). Figure 4.15 reports on the accuracies.

The convergence rate in Theorem 4.4 was derived under the assumptions of a structured equidistant grid and a $\mathcal{C}^4$-smooth solution. Practical experiments with nonuniform grids and nonsmooth data suggest that the convergence rate may still behave reasonably. But the finite-difference discretization error is not the whole story. The more flexible finite-element approaches in Chap. 5 will shed light on convergence under more general conditions.

[13] With $m = 20000$, our best result was 1.8816935.

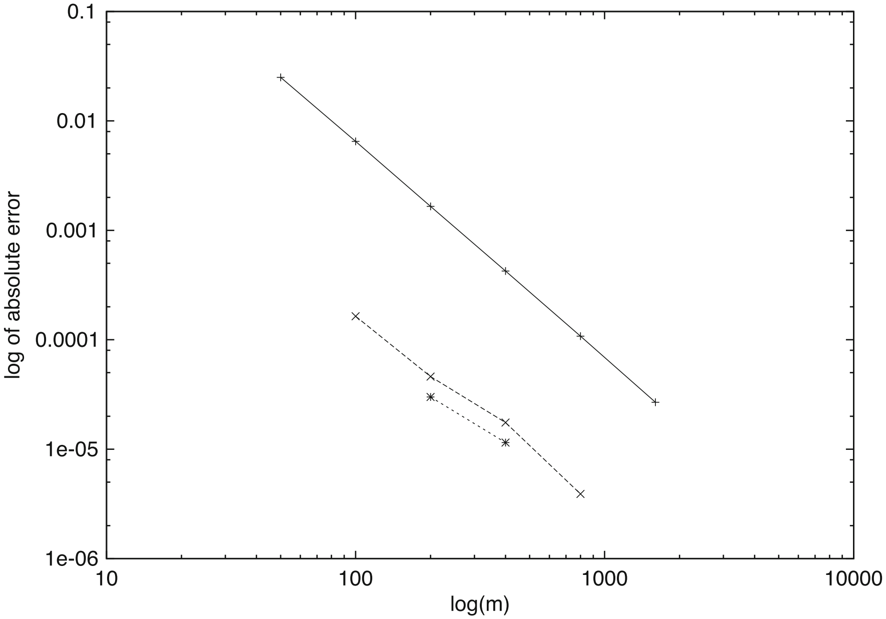

Fig. 4.15 Finite-difference methods, log of absolute error in $V(K, 0)$ over $\log(m)$, where $m = \nu_{\max}$, and the basis of the logarithm is 10. *Solid line*: plain algorithm, results in Table 4.1; *dashed line*: extrapolation (4.52) based on two approximations; *dotted line*: higher-order extrapolation of Exercise 4.15. Note that the axes in Fig. 4.15 are completely different from those of Fig. 4.14

4.8 Analytic Methods

Numerical methods typically are designed such that they achieve convergence. So, in principle, every accuracy can be reached, only limited by the available computer time and by hardware restrictions. In several cases this high potential of numerical methods is not needed. Rather, some analytic formula may be sufficient that delivers medium accuracy at low cost. Such "analytic methods" have been developed. Often their accuracy is reasonable as compared to the underlying modeling error. The limited accuracy goes along with a nice feature that is characteristic for analytic methods: their costs are clear, and known in advance.

In reality there is hardly a clear-cut between numerical and analytic methods. On the one hand, numerical methods require analysis for their derivation. And on the other hand, analytic methods involve numerical algorithms. These may be elementary evaluations of functions like the logarithm or the square root as in the Black–Scholes formula, or may consist of a sub-algorithm like Newton's iteration for zero finding.[14] There is hardly a purely analytic method.

[14] The latter situation might cause some uncertainty on the costs.

The finite-difference approach, which approximates the surface $V(S,t)$, requires intermediate values for $0 < t < T$ for the purpose of approximating $V(S,0)$. In the financial practice one is basically interested in values for $t = 0$, intermediate values are rarely asked for. So the only temporal input parameter is the time to maturity $T - t$ (or T in case the current time is set to zero, $t = 0$). Recall that also in the Black–Scholes formula, time only enters in the form $T - t$ (⟶ Appendix A.4). So it makes sense to write the formula in terms of the time to maturity τ,

$$\tau := T - t\,.$$

Setting $\tilde{V}(S,\tau) := V(S,T-\tau) = V(S,t)$ leads to a PDE for $\tilde{V}$. We drop the tilde (throughout Sect. 4.8), and arrive at a compact version of the Black–Scholes formulas (A.15) or (A.17),

$$\begin{aligned} d_1(S,\tau;K,r,\sigma) &:= \tfrac{1}{\sigma\sqrt{\tau}}\left\{\log\tfrac{S}{K} + \left(r + \tfrac{\sigma^2}{2}\right)\tau\right\}\,, \\ d_2(S,\tau;K,r,\sigma) &:= \tfrac{1}{\sigma\sqrt{\tau}}\left\{\log\tfrac{S}{K} + \left(r - \tfrac{\sigma^2}{2}\right)\tau\right\} = d_1 - \sigma\sqrt{\tau}\,, \\ V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau;K,r,\sigma) &= -SF(-d_1) + K\mathrm{e}^{-r\tau}F(-d_2)\,, \\ V_{\mathrm{C}}^{\mathrm{Eur}}(S,\tau;K,r,\sigma) &= SF(d_1) - K\mathrm{e}^{-r\tau}F(d_2)\,. \end{aligned} \tag{4.53}$$

(dividend-free case). F denotes the cumulative standard normal distribution function. For dividend-free vanilla options we only need an approximation formula for the American put $V_{\mathrm{P}}^{\mathrm{Am}}$; the other cases are covered by the Black–Scholes formula.

This Section introduces four analytic methods. The first two (Sects. 4.8.1 and 4.8.2) are described in detail such that the implementation of the algorithms is an easy matter. Of the method of lines (in Sect. 4.8.3) only basic ideas are set forth. More detail is presented on the integral representation (Sect. 4.8.4). We assume $r > 0$.

4.8.1 Approximation Based on Interpolation

If a lower bound V^{low} and an upper bound V^{up} on the American put are available,

$$V^{\mathrm{low}} \le V_{\mathrm{P}}^{\mathrm{Am}} \le V^{\mathrm{up}}\,,$$

then the idea is to construct an α aiming at

$$V_{\mathrm{P}}^{\mathrm{Am}} = \alpha V^{\mathrm{up}} + (1-\alpha)V^{\mathrm{low}}\,.$$

This is the approach of [204]. The parameter α, $0 \le \alpha \le 1$, defines an interpolation between V^{low} and V^{up}. Since $V_{\mathrm{P}}^{\mathrm{Am}}$ depends on the market data S, τ, K, r, σ, the single parameter α and the above interpolation can not be expected to provide an exact value of $V_{\mathrm{P}}^{\mathrm{Am}}$. (An exact value would mean that an exact formula for $V_{\mathrm{P}}^{\mathrm{Am}}$ would exist.) Rather a formula for α is developed as a function of S, τ, K, r, σ such that the interpolation formula

$$\alpha V^{\mathrm{up}} + (1-\alpha) V^{\mathrm{low}} \tag{4.54}$$

provides a good approximation for a wide range of market data. The smaller the gap between V^{low} and V^{up}, the better is the approximation.

An immediate candidate for the lower bound V^{low} is the value $V_{\mathrm{P}}^{\mathrm{Eur}}$ provided by the Black–Scholes formula,

$$V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau;K) \le V_{\mathrm{P}}^{\mathrm{Am}}(S,\tau;K)\,.$$

From (4.27) the left-hand boundary condition of a European put with strike $\tilde{K}$ is $\tilde{K}\mathrm{e}^{-r\tau}$ for all τ and all $\tilde{K}$. Clearly, for $\tilde{K} = K\mathrm{e}^{r\tau}$ and $S = 0$,

$$V_{\mathrm{P}}^{\mathrm{Am}}(0,\tau;K) = V_{\mathrm{P}}^{\mathrm{Eur}}(0,\tau;K\mathrm{e}^{r\tau})\,,$$

since both sides equal the payoff value K. From the properties of the American put we know $\frac{\partial V}{\partial S} \ge -1$ and $\frac{\partial^2 V}{\partial S^2} \ge 0$. Hence we conclude that

$$V_{\mathrm{P}}^{\mathrm{Am}}(S,\tau;K) \le V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau;K\mathrm{e}^{r\tau})$$

at least for small S. In fact, this inequality holds for all $S > 0$, which can be shown with Jensen's inequality, see Appendix B.1. In summary, the upper bound is

$$V^{\mathrm{up}} := V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau;K\mathrm{e}^{r\tau})\,,$$

see Fig. 4.16. The resulting approximation formula is

$$\overline{V} := \alpha V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau;K\mathrm{e}^{r\tau}) + (1-\alpha) V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau;K)\,. \tag{4.55}$$

The parameter α depends on S, τ, K, r, σ, so does $\overline{V}$. Actually, the Black–Scholes formula (4.53) suggests that α and $\overline{V}$ only depend on the three dimensionless parameters

$$S/K \text{ ("moneyness")}\,,\ r\tau\,,\ \text{and } \sigma^2\tau\,.$$

The approximation must be constructed such that the lower bound $(K-S)^+$ of the payoff is obeyed. As we will see, all depends on the free boundary S_{f}, which must be approximated as well.

Johnson [204] sets up a model for α with two free parameters a_0, a_1, which were determined by carrying out a regression analysis based on computed values of $V_{\mathrm{P}}^{\mathrm{Am}}$.

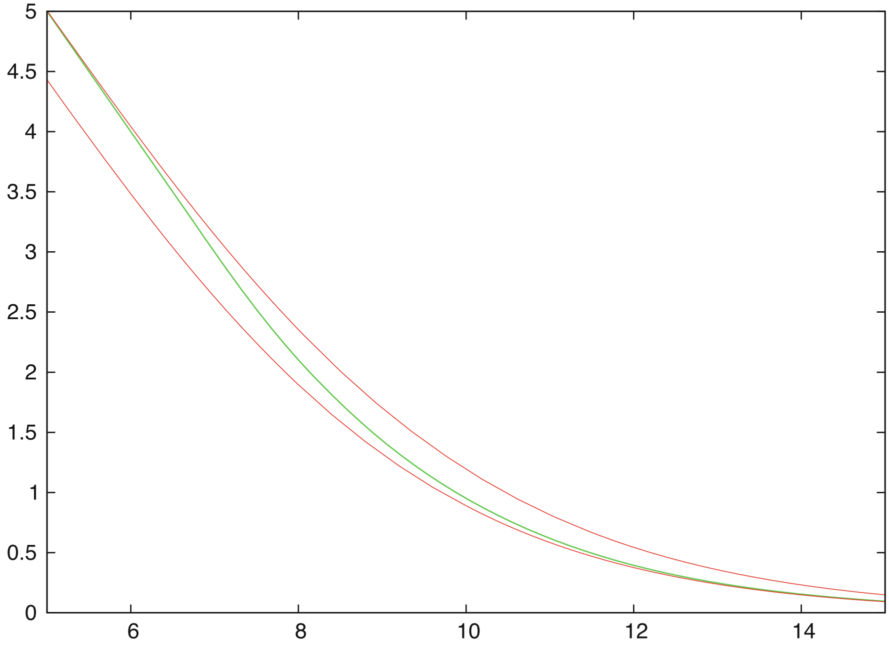

Fig. 4.16 Bounds on an American put $V(S,t;K)$ for $t = 0$ as function of S, with $K = 10$, $r = 0.06$, $\sigma = 0.3$, $\tau = 1$. Medium curve (in *green*): American put; lower curve (*red*): European put $V^{\text{Eur}}(S,0;K)$; upper curve (*red*): European put $V^{\text{Eur}}(S,0;\tilde{K})$, with $\tilde{K} = K\mathrm{e}^{r\tau}$

The result is

$$\alpha := \left(\frac{r\tau}{a_0 r\tau + a_1}\right)^{\beta}, \text{ where } \beta := \frac{\ln(S/S_\text{f})}{\ln(K/S_\text{f})}, \qquad (4.56)$$
$$a_0 = 3.9649\,, \quad a_1 = 0.032325\,.$$

The ansatz for α is designed such that for $S = K$ (and hence $\beta = 1$) upper and lower bound behavior and calculated option values can be matched with reasonable accuracy with only two parameters a_0, a_1. The S-dependent β is introduced to improve the approximation for $S < K$ and $S > K$. Obviously, $S = S_\text{f} \Rightarrow \beta = 0 \Rightarrow \alpha = 1$, which captures the upper bound. And for the lower bound, $\alpha = 0$ is reached for $S \to \infty$, and for $r\tau = 0$. (The reader may discuss (4.56) to check the assertions.)

The model for α of Eq. (4.56) involves the unknown free-boundary curve S_f. To approximate S_f, observe the extreme cases

$$S_\text{f} = K \quad \text{for} \quad \tau = 0\,,$$
$$S_\text{f} = K\frac{2r}{\sigma^2 + 2r} \quad \text{for} \quad T \to \infty\,.$$

(For the latter case consult Exercise 4.7 and Appendix A.5.) This motivates to set the approximation $\overline{S}_\mathrm{f}$ for S_f as

$$\overline{S}_\mathrm{f} := K \left(\frac{2r}{\sigma^2 + 2r} \right)^\gamma , \qquad (4.57)$$

for a suitably modeled exponent γ. To match the extreme cases, γ should vanish for $\tau = 0$, and $\gamma \approx 1$ for large values of τ. [204] suggests

$$\begin{aligned} \gamma &:= \frac{\sigma^2 \tau}{b_0 \sigma^2 \tau + b_1} , \\ b_0 &= 1.04083 , \quad b_1 = 0.00963 . \end{aligned} \qquad (4.58)$$

The constants b_0 and b_1 were again obtained by a regression analysis.

The analytic expressions of (4.57), (4.58) provide an approximation $\overline{S}_\mathrm{f}$ of S_f, and then by (4.56), (4.55) an approximation $\overline{V}$ of V_P^Am for $S > S_\mathrm{f}$, based on the Black–Scholes formulas (4.53) for $V_\mathrm{P}^\mathrm{Eur}$.

Algorithm 4.17 (Interpolation)

```
For given $S, \tau, K, r, \sigma$ evaluate $\gamma, \overline{S}_\mathrm{f}, \beta$ based on $\overline{S}_\mathrm{f}$ and $\alpha$ .
Evaluate the Black–Scholes formula for $V_\mathrm{P}^\mathrm{Eur}$
    for the arguments in (4.55).
Then $\overline{V}$ from (4.55) is an approximation to $V_\mathrm{P}^\mathrm{Am}$ for $S > \overline{S}_\mathrm{f}$ .
```

This purely analytic method is fast and simple. Numerical experiments show that the approximation quality of $\overline{S}_\mathrm{f}$ is poor. But for S not too close to $\overline{S}_\mathrm{f}$ the approximation quality of $\overline{V}$ is quite good. The error is small for $r\tau \le 0.125$, which is satisfied for average values of the risk-free rate r and time to maturity τ. For larger values of $r\tau$, when the gap between lower and upper bound widens, the approximation works less well. An extension to options on dividend-paying assets is given in [42].

4.8.2 Quadratic Approximation

Next we describe an analytic method due to [252]. Recall that in the continuation region both V_P^Am and $V_\mathrm{P}^\mathrm{Eur}$ obey the Black–Scholes equation. Since this equation is linear, also the difference

$$p(S, \tau) := V_\mathrm{P}^\mathrm{Am}(S, \tau) - V_\mathrm{P}^\mathrm{Eur}(S, \tau) \qquad (4.59)$$

satisfies the Black–Scholes equation. The relation $V^{\mathrm{Am}} \geq V^{\mathrm{Eur}}$ suggests to interpret the difference p as *early-exercise premium*. Since both $V_{\mathrm{P}}^{\mathrm{Am}}$ and $V_{\mathrm{P}}^{\mathrm{Eur}}$ have the same payoff, the terminal condition for $\tau = 0$ is zero, $p(S, 0) = 0$. The closeness of $p(S, \tau)$ to zero should scale roughly by

$$H(\tau) := 1 - \mathrm{e}^{-r\tau} . \tag{4.60}$$

This motivates introducing a scaled version f of p,

$$p(S, \tau) =: H(\tau) f(S, H(\tau)) \tag{4.61}$$

For the analysis we repeat the Black–Scholes equation, here for $p(S, \tau)$, where subscripts denote partial differentiation, and $q := \frac{2r}{\sigma^2}$:

$$-\frac{q}{r} p_\tau + S^2 p_{SS} + qSp_S - qp = 0 \tag{4.62}$$

Substituting (4.61) and

$$p_S = Hf_S \ , \ p_{SS} = Hf_{SS} \ , \ p_\tau = H_\tau f + Hf_H H_\tau$$

and using

$$\frac{1}{r} H_\tau = 1 - H$$

yields after a short calculation (the reader may check) the modified version of the Black–Scholes equation

$$S^2 f_{SS} + qSf_S - \frac{q}{H} f \left[1 + H(1 - H)\frac{f_H}{f}\right] = 0 . \tag{4.63}$$

H and q are nonzero for $r > 0$. Note that (4.63) is the "full" equation, nothing is simplified yet. No partial derivative with respect to t shows up, but instead the partial derivative f_H .

At this point, following [252], we introduce a simplifying approximation. The factor $H(H - 1)$ for the H varying in the range $0 \leq H < 1$ is a quadratic term with maximum value of $1/4$, and close to zero for $\tau \approx 0$ and for large values of τ, compare (4.60). This suggests that the term

$$H(1 - H)\frac{f_H}{f} \tag{4.64}$$

may be small compared to 1, and to neglect it in (4.63). (This motivates the name "quadratic approximation.") The resulting equation

$$S^2 f_{SS} + qSf_S - \frac{q}{H} f = 0 \tag{4.65}$$

is an ordinary differential equation with analytic solution, parameterized by H. An analysis similar as in Exercise 4.7 leads to the solution

$$f(S) = \alpha S^\lambda \ , \ \text{where} \ \lambda := -\frac{1}{2}\left\{(q-1) + \sqrt{(q-1)^2 + \frac{4q}{H}}\right\} \ , \tag{4.66}$$

for a parameter α. Combining (4.59), (4.61) and (4.66) we deduce for $S > S_\text{f}$ the approximation $\overline{V}$

$$V_\text{P}^\text{Am}(S,\tau) \approx \overline{V}(S,\tau) := V_\text{P}^\text{Eur}(S,\tau) + \alpha H(\tau) S^\lambda \, . \tag{4.67}$$

The parameter α must be such that $\overline{V}$ reaches the payoff at S_f,

$$V_\text{P}^\text{Eur}(S_\text{f},\tau) + \alpha H S_\text{f}^\lambda = K - S_\text{f} \, . \tag{4.68}$$

Here S_f is parameterized by H via (4.60), and therefore depends on τ. To fix the two unknowns S_f and α let us warm up the high-contact condition. This requires the partial derivative of $\overline{V}$ with respect to S. The main part is

$$\frac{\partial V_\text{P}^\text{Eur}(S,\tau)}{\partial S} = F(d_1) - 1$$

where F is the cumulative normal distribution function, and d_1 (and below d_2) are the expressions defined by (4.53). d_1 and d_2 depend on all relevant market parameters; we emphasize the dependence on S by writing $d_1(S)$. This gives the high-contact condition

$$\frac{\partial \overline{V}(S_\text{f},\tau)}{\partial S} = F(d_1(S_\text{f})) - 1 + \alpha\lambda H S_\text{f}^{\lambda-1} = -1 \, ,$$

and immediately α in terms of S_f:

$$\alpha = -\frac{F(d_1(S_\text{f}))}{\lambda H S_\text{f}^{\lambda-1}} \, . \tag{4.69}$$

Substituting into (4.68) yields one equation for the remaining unknown S_f,

$$V_\text{P}^\text{Eur}(S_\text{f},\tau) - F(d_1(S_\text{f}))\frac{1}{\lambda} S_\text{f} = K - S_\text{f} \, ,$$

which in view of the put-call parity (A.16) and $F(-d) = 1 - F(d)$ reads

$$S_\mathrm{f}F(d_1) - K\mathrm{e}^{-r\tau}F(d_2) - S_\mathrm{f} + K\mathrm{e}^{-r\tau} - F(d_1)\frac{S_\mathrm{f}}{\lambda} - K + S_\mathrm{f} = 0\,.$$

This can be summarized to

$$S_\mathrm{f}\,F(d_1(S_\mathrm{f}))\left[1 - \frac{1}{\lambda}\right] + K\mathrm{e}^{-r\tau}\left[1 - F(d_2(S_\mathrm{f}))\right] - K = 0\,. \tag{4.70}$$

Since d_1 and d_2 vary with S_f, (4.70) is an implicit equation for S_f and must be solved iteratively. In this way a sequence of approximations $S_1, S_2, \ldots$ to S_f is constructed. We summarize

Algorithm 4.18 (Quadratic Approximation)

For given S, τ, K, r, σ evaluate $q = \frac{2r}{\sigma^2}$, $H = 1 - \mathrm{e}^{-r\tau}$ and λ from (4.66).

Solve (4.70) iteratively for S_f.

(This involves a sub-algorithm, from which $F(d_1(S_\mathrm{f}))$ should be saved.)

Evaluate $V_\mathrm{P}^\mathrm{Eur}(S,\tau)$ using the Black–Scholes formula (4.53).

$$\overline{V} := V_\mathrm{P}^\mathrm{Eur}(S,\tau) - \frac{1}{\lambda}S_\mathrm{f}F(d_1(S_\mathrm{f}))\left(\frac{S}{S_\mathrm{f}}\right)^\lambda \tag{4.71}$$

is the approximation for $S > S_\mathrm{f}$,

and $\overline{V} = K - S$ for $S \le S_\mathrm{f}$.

Note that $\lambda < 0$, and λ depends on τ via $H(\tau)$. The time-consuming part of the quadratic-approximation method consists of the numerical root finding procedure. But here a moderate accuracy suffices, since a very small error in S_f does not affect the error in $\bar{V}$. ($\longrightarrow$ Exercises 4.17 and 4.18)

4.8.3 Analytic Method of Lines

In solving PDEs numerically, the *method of lines* is a well-known approach. It is based on a semidiscretization, where the domain (here the (S,τ) half strip) is replaced by a set of lines parallel to the S-axis, each defined by a constant value of τ. To this end, the interval $0 \le \tau \le T$ is discretized into ν_max sub-intervals by $\tau_\nu := \nu\Delta\tau$, $\Delta\tau := T/\nu_\mathrm{max}$, $\nu = 1, \ldots, \nu_\mathrm{max} - 1$. To deserve the attribute “analytic,” we assume ν_max to be small, say, work with three lines. We write the Black–Scholes equation as in Sect. 4.5.3,

$$-\frac{\partial V(S,\tau)}{\partial \tau} + \mathcal{L}_{\mathrm{BS}}(V(S,\tau)) = 0\,, \tag{4.72}$$

where the negative sign compensates for the transition from t to τ, and replace the partial derivative $\partial V/\partial\tau$ by the difference quotient

$$\frac{V(S,\tau) - V(S,\tau-\Delta\tau)}{\Delta\tau}\,.$$

This gives a semidiscretized version of (4.72), namely, the ordinary differential equation

$$w(S,\tau-\Delta\tau) - w(S,\tau) + \Delta\tau\,\mathcal{L}_{\mathrm{BS}}(w(S,\tau)) = 0\,,$$

which holds for $S > S_{\mathrm{f}}$. Here we use the notation w rather than V to indicate that a discretization error is involved. This semidiscretized version is applied for each of the parallel lines, $\tau = \tau_\nu$, $\nu = 1,\ldots,\nu_{\max}-1$. Figure 4.17 may motivate the procedure. For each line $\tau = \tau_\nu$, the function $w(S,\tau_{\nu-1})$ is known from the previous line, starting from the known payoff for $\tau = 0$. The equation to be solved for each line τ_ν is

$$\frac{1}{2}\Delta\tau\,\sigma^2 S^2 \frac{\partial^2 w}{\partial S^2} + \Delta\tau\, rS\frac{\partial w}{\partial S} - (1+\Delta\tau\, r)w = -w(\cdot,\tau_{\nu-1})\,. \tag{4.73}$$

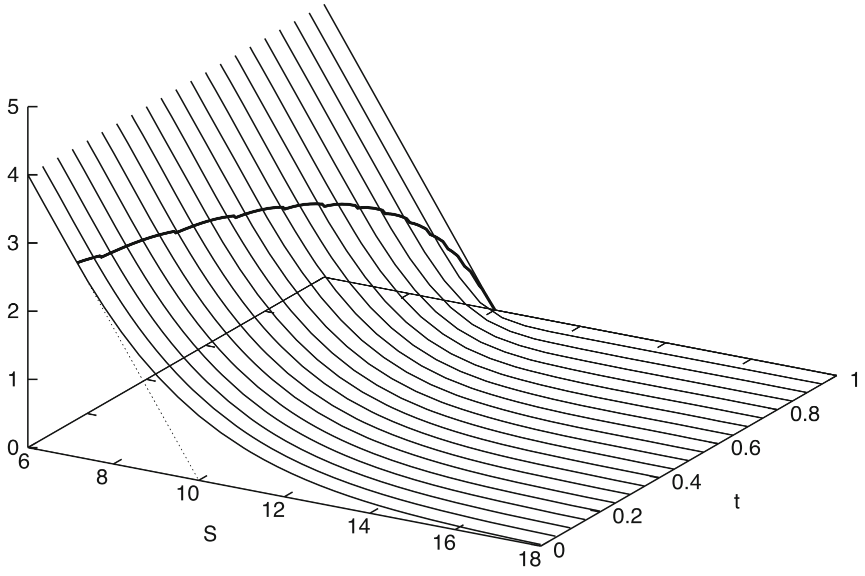

Fig. 4.17 Method of lines, situation as in Fig. 1.5. The early-exercise curve is indicated

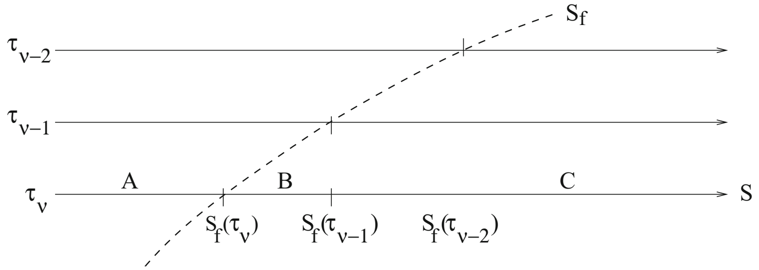

Fig. 4.18 Method of lines, situation along line τ_ν: *A*: solution is given by payoff; *B*: inhomogeneous term of differential equation given by payoff; *C*: inhomogeneous term given by $-w(., \tau_{\nu-1})$

This is a second-order ordinary differential equation for $w(S, \tau_\nu)$, with boundary conditions for $S_f(\tau_\nu)$ and $S \to \infty$. The solution is obtained analytically, similar as in Exercise 4.7. Hence there is no discretization error in S-direction. The right-hand function $-w(S, \tau_{\nu-1})$ is known, and is an inhomogeneous term of the ODE.

The resulting *analytic method of lines* is carried out in [66]. The above describes the basic idea. A complication arises from the early-exercise curve, which separates each of the parallel lines into two parts. Since for the previous line $\tau_{\nu-1}$ the separation point lies more "on the right" (recall that for a put the curve $S_f(\tau)$ is monotonically decreasing for growing τ), the inhomogeneous term $w(\cdot, \tau_{\nu-1})$ consists of two parts as well, but separated differently (see Fig. 4.18). Accordingly, neglecting for the moment the input of previous lines $\tau_{\nu-2}$, $\tau_{\nu-3}, \ldots$, the analytic solution of (4.73) for the line τ_ν consists of three parts, defined on the three intervals

$$\begin{aligned} &\text{A:} \quad 0 < S < S_f(\tau_\nu)\,, \\ &\text{B:} \quad S_f(\tau_\nu) \le S < S_f(\tau_{\nu-1})\,, \\ &\text{C:} \quad S_f(\tau_{\nu-1}) \le S\,. \end{aligned}$$

On the left-hand interval A, w equals the payoff; nothing needs to be calculated. For the middle interval B the inhomogeneous term $-w(., \tau_{\nu-1})$ is given by the payoff. Since the analytic solution involves two integration constants, and since the inhomogeneous terms differ on the intervals B and C, we encounter together with the unknown $S_f(\tau_\nu)$ five unknown parameters. One of the integration constants is zero because of the boundary condition for $S \to \infty$, similar as in Exercise 4.7. The unknown separation point $S_f(\tau_\nu)$ is again fixed by the high-contact conditions (4.34). Two remaining conditions are given by the requirement that both w and $\frac{dw}{dS}$ are continuous at the matching point $S_f(\tau_{\nu-1})$. This fixes all variables for the line τ_ν.

Over all lines, $\nu_{\max}$ type-B intervals are involved, and the only remaining type-C interval is that for $S \ge S_f(\tau_0) = K$. The resulting formulas are somewhat complex, for details see [66]. The method is used along with extrapolation. To this end, carry out the method three times, with $\nu_{\max} = 1, 2, 3$, and denote the results $\overline{V}_1, \overline{V}_2, \overline{V}_3$.

Then the three-point extrapolation formula

$$\overline{V} := \frac{1}{2}\,(\,9\overline{V}_3 - 8\overline{V}_2 + \overline{V}_1\,) \tag{4.74}$$

gives rather accurate results.

The method of lines can be carried out numerically [273]. For lines parallel to the t-axis, see Exercise 4.3 and Fig. 4.21.

4.8.4 Integral-Equation Method

Recall for European put options the integral representation (1.66)

$$V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau) = \mathrm{e}^{-r\tau}\int_0^\infty (K-S_T)^+ f_{\mathrm{GBM}}(S_T,\tau;\;S,r-\delta,\sigma)\,\mathrm{d}S_T\,,$$

where $\tau := T-t$ denotes the remaining time to expiration, and f_{GBM} is the density function from (1.64). Solving this integral one arrives at the Black–Scholes formula. We repeat from (4.53) the two functions (here with constant dividend yield rate $\delta \geq 0$),

$$d_1(S,\tau;K) := \frac{\log\frac{S}{K} + \left(r-\delta+\frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}\,,\quad d_2(S,\tau;K) := d_1 - \sigma\sqrt{\tau}\,, \tag{4.75}$$

for $\tau > 0$. With d_1, d_2 evaluated at S, τ, K, recall

$$V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau) = -S\mathrm{e}^{-\delta\tau}F(-d_1) + K\mathrm{e}^{-r\tau}F(-d_2)\,,$$

where F denotes the standard normal cumulative distribution. (See also Appendix A.4.) Further recall from (4.59) the early-exercise premium p, with

$$V_{\mathrm{P}}^{\mathrm{Am}}(S,\tau) = V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau) + p(S,\tau)\,.$$

As suggested by [222] and others, the premium function p can be represented as an integral over functions depending on the free boundary S_{f}. The result is

$$\begin{aligned} V_{\mathrm{P}}^{\mathrm{Am}}(S,\tau) = \quad & V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau) + \\ & + \int_0^\tau \,[\,rK\mathrm{e}^{-r\xi}F(-d_2(S,\xi;S_{\mathrm{f}}(\tau-\xi))) \\ & \qquad -\delta S\mathrm{e}^{-\delta\xi}F(-d_1(S,\xi;S_{\mathrm{f}}(\tau-\xi)))\,]\,\mathrm{d}\xi\,. \end{aligned} \tag{4.76}$$

The integral is identical to

$$\int_0^\tau [\, rK\mathrm{e}^{-r(\tau-\xi)} F(-d_2(S,\tau-\xi; S_\mathrm{f}(\xi))) \\ -\delta S\mathrm{e}^{-\delta(\tau-\xi)} F(-d_1(S,\tau-\xi; S_\mathrm{f}(\xi)))\,]\,\mathrm{d}\xi\,. \tag{4.77}$$

4.8.4.1 Integral Equation for S_f

Substitute $V(S_\mathrm{f}(\tau),\tau) = K - S_\mathrm{f}(\tau)$ into (4.76) and obtain

$$\begin{aligned} K - S_\mathrm{f}(\tau) = &- S_\mathrm{f}(\tau)\,\mathrm{e}^{-\delta\tau}\, F(-d_1(S_\mathrm{f}(\tau),\tau;K)) \\ &+ K\mathrm{e}^{-r\tau}\, F(-d_2(S_\mathrm{f}(\tau),\tau;K)) \\ &+ \int_0^\tau [\, rK\mathrm{e}^{-r\xi}\, F(-d_2(S_\mathrm{f}(\tau),\xi; S_\mathrm{f}(\tau-\xi))) \\ &\qquad -\delta S_\mathrm{f}(\tau)\,\mathrm{e}^{-\delta\xi}\, F(-d_1(S_\mathrm{f}(\tau),\xi; S_\mathrm{f}(\tau-\xi)))\,]\,\mathrm{d}\xi\,. \end{aligned} \tag{4.78}$$

This constitutes an integral equation for the free-boundary function (early-exercise curve) $S_\mathrm{f}(\tau)$ of an American put.

4.8.4.2 Numerical Solution of the Integral Equation

We denote the integrand in (4.78) by $g(S_\mathrm{f}(\tau), S_\mathrm{f}(\tau-\xi), \xi)$ (⟶ Exercise 4.19). So the integral equation reads

$$K - S_\mathrm{f}(\tau) = V_\mathrm{P}^\mathrm{Eur}(S_\mathrm{f}(\tau),\tau) + \int_0^\tau g(S_\mathrm{f}(\tau), S_\mathrm{f}(\tau-\xi),\xi)\,\mathrm{d}\xi\,.$$

Let the τ-interval be subdivided by discrete τ_ν into M subintervals, with $\tau_0 = 0$, $\tau_M = \tau$, and with equidistant steps $\Delta\tau = \tau/M$, and $\tau_\nu = \nu\Delta\tau$. The numerical treatment resembles that for ODE initial-value problems. Basically the integral is approximated by a composite trapezoidal sum (C.2). Note from Appendix A.5 that $S_\mathrm{f}(\tau)$ for $\tau \to 0^+$ is known,

$$S_\mathrm{f0} := \lim_{\tau\to 0^+} S_\mathrm{f}(\tau) = \min\{K,\ \frac{r}{\delta}K\}\,.$$

We use the notation $S_{\mathrm{f}\nu} := S_\mathrm{f}(\tau_\nu)$. Specifically for τ_1, the integral and (4.78) can be approximated by the trapezoidal rule

$$K - S_\mathrm{f1} = V_\mathrm{P}^\mathrm{Eur}(S_\mathrm{f1},\tau_1) + \frac{\Delta\tau}{2}[g(S_\mathrm{f1}, S_\mathrm{f1}, \tau_0) + g(S_\mathrm{f1}, S_\mathrm{f0}, \tau_1)]\,, \tag{4.79}$$

which is solved iteratively for its only unknown S_{f1} by any root-finding procedure. After S_{f1} is calculated to sufficient accuracy, the next equation is

$$\begin{aligned} K - S_{\text{f2}} &= V_{\text{P}}^{\text{Eur}}(S_{\text{f2}}, \tau_2) \\ &\quad + \frac{\Delta\tau}{2}[g(S_{\text{f2}}, S_{\text{f2}}, \tau_0) + 2g(S_{\text{f2}}, S_{\text{f1}}, \tau_1) + g(S_{\text{f2}}, S_{\text{f0}}, \tau_2)]\,, \end{aligned}$$

which is solved for S_{f2}. In this way, the composite trapezoidal sum builds up until we reach the final iteration for $S_{\text{f}M}$. So, recursively for $k = 2, \ldots, M$ solve

$$\begin{aligned} &K - S_{\text{f}k} = V_{\text{P}}^{\text{Eur}}(S_{\text{f}k}, \tau_k) + \\ &+ \frac{\Delta\tau}{2}\left[g(S_{\text{f}k}, S_{\text{f}k}, \tau_0) + 2\sum_{\nu=1}^{k-1} g(S_{\text{f}k}, S_{\text{f}(k-\nu)}, \tau_\nu) + g(S_{\text{f}k}, S_{\text{f0}}, \tau_k)\right] \end{aligned} \tag{4.80}$$

for $S_{\text{f}k}$. This recursion is run for $\tau = T$ to obtain values for $t = 0$.

The iterative solution of the above nonlinear equations [as (4.79), (4.80)] can be done, for example, by the secant method (C.5). The error control of the integral-equation method represented by (4.80) involves the discretization error of the trapezoidal sum as well as the error remaining when the secant iteration is stopped. Recall that the secant method requires *two* reasonable initial guesses. Alternatively, we recommend the highly robust bisection method. There is ample opportunity to test various strategies (⟶ Exercise 4.20).

4.8.4.3 Evaluation of the Premium

Now, the free boundary S_{f} is approximated by the chain of points

$$(\tau_0, S_{\text{f0}}), (\tau_1, S_{\text{f1}}), \ldots, (\tau_M, S_{\text{f}M})\,.$$

Based on this approximation, the evaluation of (4.76) is a simple task. Apply the analogous trapezoidal sum with the same discretization to approximate $V(S, \tau)$ for $\tau = \tau_M$:

$$\begin{aligned} &V(S, \tau) \approx V_{\text{P}}^{\text{Eur}}(S, \tau) + \\ &+ \frac{\Delta\tau}{2}[g(S, S_{\text{f}M}, 0) + 2\sum_{\nu=1}^{M-1} g(S, S_{\text{f}(M-\nu)}, \tau_\nu) + g(S, S_{\text{f0}}, \tau)]\,. \end{aligned} \tag{4.81}$$

The evaluation of (4.81) does not need any further iteration and is much cheaper than the preceding recursion (4.80).

4.8.4.4 Calculation of the Greeks

The same holds true for evaluating greeks. After calculating the partial derivatives of (4.76), one obtains corresponding formulas for the greeks. For example, delta is given by the formula

$$\Delta_{\mathrm{P}}^{\mathrm{Am}} = -\mathrm{e}^{-\delta\tau} F(-d_1) - \int_0^{\tau} g_{\mathrm{P}}^{\Delta} \,\mathrm{d}\xi$$

for a function g_{P}^{Δ} defined below. The calculation works as simply as in (4.81); the free boundary S_{f} is not calculated again. And similarly, other greeks are obtained, both for put and call. The resulting formulas are given in [190]. With the version of (4.77), and d_1 evaluated at the arguments $(S, \tau - \xi, S_{\mathrm{f}}(\xi))$,

$$g_{\mathrm{P}}^{\Delta} = \delta \mathrm{e}^{-\delta(\tau-\xi)} F(-d_1(S, \tau - \xi, S_{\mathrm{f}}(\xi))) + \frac{\mathrm{e}^{-d_1^2/2}}{\sqrt{2\pi}} \mathrm{e}^{-\delta(\tau-\xi)} \frac{rK - \delta S_{\mathrm{f}}(\xi)}{\sigma S_{\mathrm{f}}(\xi)\sqrt{\tau - \xi}} \, .$$

For these arguments and $\xi \to \tau$, $|d_1|$ is getting infinite, and

$$g_{\mathrm{P}}^{\Delta} = \begin{cases} 0 & \text{for } S > S_{\mathrm{f}} \, , \\ \delta & \text{for } S < S_{\mathrm{f}} \, . \end{cases}$$

4.8.5 Other Methods

The early-exercise curve $S_{\mathrm{f}}(\tau)$ can be approximated by pieces of exponential functions

$$B \exp(b\tau) \quad \text{for } \tau_1 \le \tau \le \tau_2 \, ,$$

for parameters B, b and suitable intervals for τ. Substituting this expression for $S_{\mathrm{f}}(\tau)$ into d_1 and d_2 in (4.76) leads to the observation that the integrals can be evaluated analytically in terms of the distribution function F. The parameters B, b are determined such that the high-contact boundary-condition condition is satisfied. Depending on the number of pieces of exponential functions, a good approximation of (4.76) is obtained. This is the method of [208]. The accuracy of the highly efficient three-piece approximation corresponds to that of the integral-equation method with about $M = 100$ subintervals.

[56] establishes LUBA, an analytic method for American calls. The derivation is beyond the scope of this textbook, but is worth at least a brief sketch because of its striking computational power. The method starts from a *capped call*, which is basically a vanilla European call, with the exception that for $t < T$ the option is exercised at the first time t such that S_t reaches the cap. The price

of the capped call can be replicated with two barrier options. Their analytical formulas constitute a lower bound LB on the option. This in turn, via the integral representation (4.76) lends to an upper bound UB. Then LB and UB are interpolated with a regression ansatz comparable to the interpolation of Sect. 4.8.1. The resulting specific approximation of [56] is called LUBA, which stands for lower upper bound approximation.

4.9 Criteria for Comparisons

In this chapter, we have learned about the basic structure of finite-difference methods, and we have studied several analytic approaches. How do these methods compare? As we shall see, this question is difficult to answer. There are several criteria to judge the performance of a computational method. The criteria include reliability, range of applicability, amount of information provided by the method, and speed, and error. Speed and error are relatively easy to compare, and we shall concentrate on these two criteria.

For the computational arena, we need to define a set of test examples, based on which we have to calculate a *benchmark* in high accuracy. Results of any chosen method will be compared to the benchmark. To measure the deviation, a suitable error must be defined. This Sect. 4.9 roughly sketches the steps of a comparison.

4.9.1 Set of Test Examples

We concentrate on the valuation of plain-vanilla options. This restriction to vanillas has the advantage that all kind of numerical methods are applicable and can be compared. And we confine ourselves to the valuation of American put options. The parameters $K, S, T, \sigma, r, \delta$ are chosen

- $K = 100$
- $S \in \{90, 100, 110, 150\}$
- $T \in \{0.5, 1, 2\}$
- $\sigma \in \{0.1, 0.3, 0.5\}$
- $r \in \{0.05, 0.1\}$ for $\delta = 0$; $\quad r \in \{0.15, 0.2\}$ for $\delta = 0.1$

Altogether these are 72 combinations with dividend rate $\delta = 0$ and as many for $\delta = 0.1$. But for $\sigma = 0.1$, in 12 of these cases, either

$$V(S, 0) \approx 0 \quad \text{or} \quad V(S, 0) = \text{ payoff}$$

occurs. In these cases, a relative error is meaningless, or nothing is to be calculated. Hence we remove theses 12 cases ($\sigma = 0.1, S = 90, S = 150$). The remaining 60 parameter combinations were organized into two files.[15]

For each set of parameters we calculated $V(S, 0)$ with rather high accuracy (7–8 decimal digits). To this end, we applied as reference method an extrapolation based on finite-difference approximations, as suggested in Sect. 4.7.2. The obtained values complete the benchmark files. Any method can be compared to the benchmark as long as its relative error is not smaller than 10^{-6}.

4.9.2 Measure of the Error

To measure performances, we calculate the *root mean square relative error*

$$\text{RMS} := \sqrt{\frac{1}{60}\sum_{i=1}^{60}\left(\frac{\overline{V}_i - V_i}{V_i}\right)^2} . \tag{4.82}$$

Here V_i denotes the "accurate" benchmark value of the ith parameter combination, and $\overline{V}_i$ denotes the value calculated with the method whose performance is to be measured.

4.9.3 Arena of Competing Methods

We have chosen the following prototypical methods:

B-M: binomial method with M time steps, Algorithm 1.4,
$M = 12, 25, 50, \ldots, 1600$;
FD-BS-M: finite differences Brennan–Schwartz, Algorithm 4.15,
with $M := m = \nu_{\max},\ M = 200, 400, \ldots, 6400$;
J: Johnson's interpolation, Algorithm 4.17;
Q: quadratic approximation, Algorithm 4.18;
I-M: integral-equation method with M subintervals, Sect. 4.8.4,
$M = 50, 100, \ldots, 3200$;
FD-BS-ex: version of FD-BS with two solutions with M and $M/2$
and extrapolation.

Keep in mind that the above methods provide different amount of information; in some sense we compare apples with oranges. The integer M represents a fineness of

[15] The files `BENCHMARK00` for $\delta = 0$ and `BENCHMARK01` for $\delta = 0.1$ can be found on www.compfin.de.

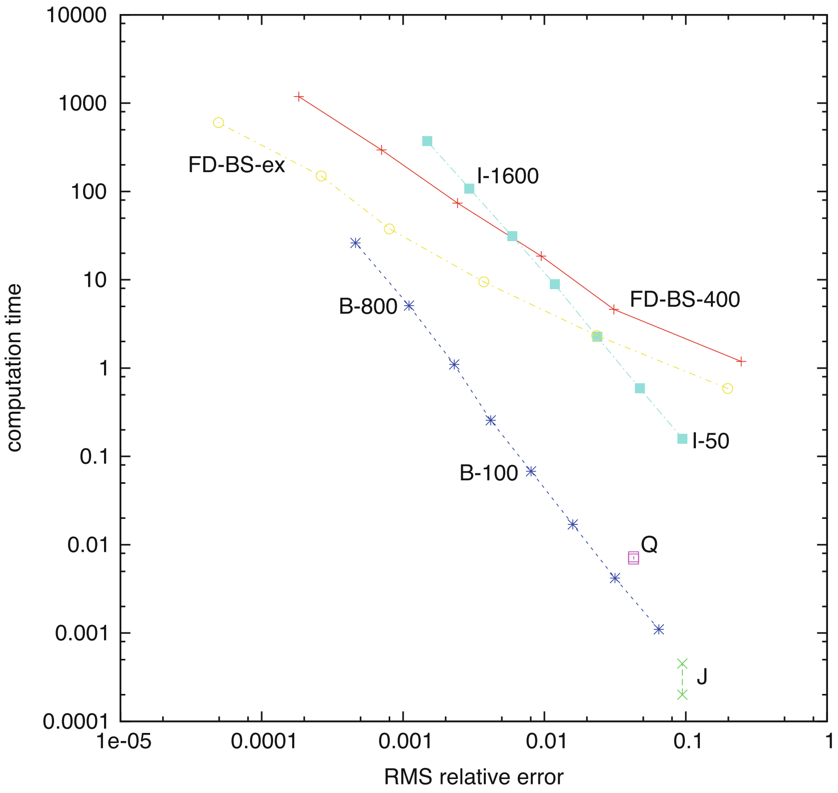

Fig. 4.19 Computing times and RMS errors of several methods, see the text. *Points mark* calculated RMS errors; corresponding *points* are connected by *lines*

discretization, which is consecutively doubled for clarity of exposition. Computing times in Fig. 4.19 report the time in seconds needed to valuate all of the 60 options for $\delta = 0$; overhead is subtracted.[16] The log scaling in Fig. 4.19 is most practical (⟶ Exercise 4.21). For the versions with shortest computing time (J), the time is hardly measurable, which is indicated by a bar of likely computing times.

In Fig. 4.19, the accuracy orders of the various methods can not be seen directly. The convergence rate would become apparent in case the absolute error is depicted over the grid size. Such a figure reveals the first-order convergence of the binomial method and the integral-equation method, and essentially a second-order convergence of the finite-difference method.

[16] All of the above methods were implemented in FORTRAN (F90 compiler) and run on a DS20 processor.

4.9.4 Preliminary Results

In the sense of Pareto optimization, smaller values in Fig. 4.19 are preferred to larger ones. Entries in the lower left part of the figure refer to methods with higher efficiency. The Pareto frontier in this figure is largely dominated by the binomial method (B). This holds at least for medium demands for accuracy. Both the analytic methods (J) and (Q) do not need the evaluation of the the Black–Scholes formula and hence $\surd$, log, exp in full accuracy. So their evaluation can be accelerated. Hence, for low accuracy, Johnson's interpolation method (J) and the quadratic approximation (Q) are competitive. This is not clear from the figure, where unnecessary accuracy of the underlying Black–Scholes formula falsely suggests that the quadratic approximation (Q) is dominated by the binomial method. For high demands for accuracy, the finite-difference method is competitive. The basic version of the binomial method dominates the basic version of the integral-equation method (I). The aspect of convergence applies to FD, B, I, but not to the fixed accuracy of Q, J. This may be seen as distinction between a numerical method and an analytic method.

4.9.5 Outlook

The above observations should not be considered as definite recommendations. It is important to realize that the conclusions refer to speed and RMS error only. Several aspects are neglected and lacking. For example, the finite-difference method calculates the surface of the value function $V(S,t)$, and provides more information than the binomial method. Or, the integral-equation method allows to calculate the greeks more effectively, and approximates the early-exercise curve very well (B does not). The above has selected one representative method of important classes of methods. These basic versions are implemented and compared. There are more efficient methods not shown in Fig. 4.19. For example, LUBA has shown to dominate the methods with comparable accuracy. Neither the highly efficient front-fixing methods are shown, nor the improvement [175] of the integral method, nor the fast approximation by exponential pieces. Improvements differ in the degree of speedup. Further, storage requirements are not taken into account. Implementation details do matter! And applied to a specific type of exotic option, the prototype methods chosen for Fig. 4.19 may behave and compare differently. Monte Carlo methods are not included at all, because their merits are beyond vanilla options. So the conclusions of this section aim at basic principles. They are tentative, and not comprehensive. We do not answer the question, what might be the "best" method for a particular application. For early comparisons, see [4, 56, 57, 211]. More recent developments have not been compared.

4.10 Notes and Comments

On Sect. 4.1

General references on numerical PDEs include [80, 281, 341, 356, 369]. A special solution of (4.2) is

$$y(x,\tau) = \frac{1}{2\sqrt{\pi\tau}} \exp\left(-\frac{x^2}{4\tau}\right) .$$

For small values of τ, the transformation (4.3) may take bad values in the argument of the exponential function because q_δ can be too large. The result will be an overflow. In such a situation, the transformation

$$\begin{aligned} \tau &:= \tfrac{1}{2}\sigma^2(T-t) \\ x &:= \log\left(\tfrac{S}{K}\right) + \left(r - \delta - \tfrac{\sigma^2}{2}\right)(T-t) \\ y(x,\tau) &:= \mathrm{e}^{-rt}V(S,t) \end{aligned}$$

can be used as alternative [28]. Again (4.2) results, but initial conditions and boundary conditions must be adapted appropriately (see also Appendix A.6). The equations also hold for options on foreign currencies. Then δ represents the foreign interest rate. As will be seen in Sect. 6.4, the quantities q and q_δ are basically the Péclet number. It turns out that large values of the Péclet number are a general source of difficulties. For other transformations see [381]. *Well-posed* means the existence of a unique solution that depends continuously on the data.

For the valuation of American options in case of **discrete dividend** payments there is a big difference between call and put. A call is exercised immediately prior to the dividend date, provided some analytically known criteria are satisfied [234]. In contrast, a put must be calculated numerically. By arbitrage reasons, the stock price jumps at the ex-dividend date t_D,

$$S_{t_D^+} = S_{t_D^-} - D \, ,$$

where D is the net amount paid at t_D. The price V_t of the put does not jump along the path S_t because the option's holder has no benefit from the payment. This continuity of $V(S_t, t)$ can be written

$$V(S,\, t_D^-) = V(S-D,\, t_D^+) \, ,$$

which amounts to a jump in the value function $V(S,t)$ at t_D.[17] For a numerical implementation, place a node t_ν at t_D, interrupt the integration of the PDE at t_D, and

[17] For tree methods, dividends are discussed in Appendix D.2.

apply interpolation to evaluate V at $S_i - D$ in case this is not a node. Then the PDE is applied again. For a method-of-lines approach see [273]. Exercise 4.1b provides some insight into the early-exercise structure. For $t_D < t < T$ the early-exercise curve is that of a non-dividend paying stock [27, 292].

On Sect. 4.2

We follow the notation $w_{i,\nu}$ for the approximation at the node (x_i, τ_ν), to stress the surface character of the solution y over a two-dimensional domain. In the literature a frequent notation is w_i^ν, which emphasizes the different character of the space variable (here x) and the time variable (here τ). Our vectors $w^{(\nu)}$ with components $w_i^{(\nu)}$ come close to this convention.

Finite differences work for nonuniform meshes as well. Then formally the discretization errors are of first order only. But under mild assumptions on a slowly varying mesh, second-order accuracy can be obtained [257].

Summarizing the Black–Scholes equation to

$$\frac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V) = 0 \tag{4.83}$$

where $\mathcal{L}_{\mathrm{BS}}$ represents the other terms of the equation, see Sect. 4.5.3, motivates an interpretation of the finite-difference schemes in the light of numerical ODEs. There the forward approach is known as *explicit Euler method* and the backward approach as *implicit Euler method*. The explicit scheme corresponds to the trinomial-tree method mentioned in Sect. 1.4 [191].

On Sect. 4.3

Crank and Nicolson suggested their approach in 1947 [91]. Theorem 4.4 discusses three main principles of numerical analysis, namely, order of convergence, stability, and efficiency. A Crank–Nicolson variant has been developed that is consistent with the volatility smile, which reflects the dependence of the volatility on the strike [10].

In view of the representation (4.20) the Crank–Nicolson approach corresponds to the ODE *trapezoidal rule*. Following these lines suggests to apply other ODE approaches, some of which lead to methods that relate more than two time levels. In particular, the backward difference formula BDF (4.11) is of interest, which evaluates $\mathcal{L}$ at one time level only. Using formula (4.11) for the time discretization, a three-term recursion involving $w^{(\nu+1)}, w^{(\nu)}, w^{(\nu-1)}$ replaces the two-term recursion (4.24) ($\longrightarrow$ Exercise 4.3). But multistep methods such as BDF may suffer from the lack of smoothness at the exercise boundary. This effect is mollified when the inequality is tackled by a penalty term. But even then it is interesting to consider other alternatives with better stability properties than Crank–Nicolson. Crank–Nicolson is A-stable, several other methods are L-stable, which better damp out high-frequency oscillation, see [71, 194, 221]. For numerical ODEs we refer to [165, 236]. From the ODE analysis circumstances are known where the implicit Euler method behaves superior to the trapezoidal rule. The latter method may show a *slowly damped* oscillating error. Accordingly, in several PDE situations the fully implicit method of Sect. 4.2.5 behaves better than the Crank–Nicolson method [310, 386].

On Sect. 4.4

The boundary condition $V_C(0,t) = 0$ in (4.26) can be shown independently of any underlying model [269]. If European options are evaluated via the analytic formulas (A.15)–(A.17), the boundary conditions in (4.28) are of no practical interest. When boundary conditions are not clear, it sometimes helps to set $V_{SS} = 0$ (or $y_{xx} = 0$), which amounts to assume linear behavior. See [353] for a discussion, and for the effect of boundary conditions on accuracy and stability. For bounds on the error caused by truncating the infinite x- or S-interval, see [214]. Boundary conditions for a term structure equation are discussed in [117].

On Sect. 4.5

For a proof of the Black–Scholes inequality, see [237, p. 111]. The obstacle problem in this chapter is described following [376]. Also the smooth pasting argument of Exercise 4.6 is based on that work. For other arguments concerning smooth pasting see [277], and [234], where you find a discussion of $S_f(t)$, and of the behavior of this curve for $t \to T$. There are several different possibilities to implement boundary conditions at $x_{\min}$, $x_{\max}$, see [353, p. 122]. The accuracy can be improved with artificial boundary conditions [169]. For direct methods, see also [99, 194]. Front-fixing goes back to Landau 1950, see [90]. For front-fixing applications to finance, consult, for example, [188, 288, 381], and the comments on Sect. 4.7.

The general definition of a linear complementarity problem is

$$\mathsf{AB} = 0\,, \quad \mathsf{A} \geq 0\,, \quad \mathsf{B} \geq 0\,,$$

where A and B are abbreviations of more complex expressions. This can be also written

$$\min(\mathsf{A}, \mathsf{B}) = 0\,.$$

A general reference on free boundaries and on linear complementarity is [119].

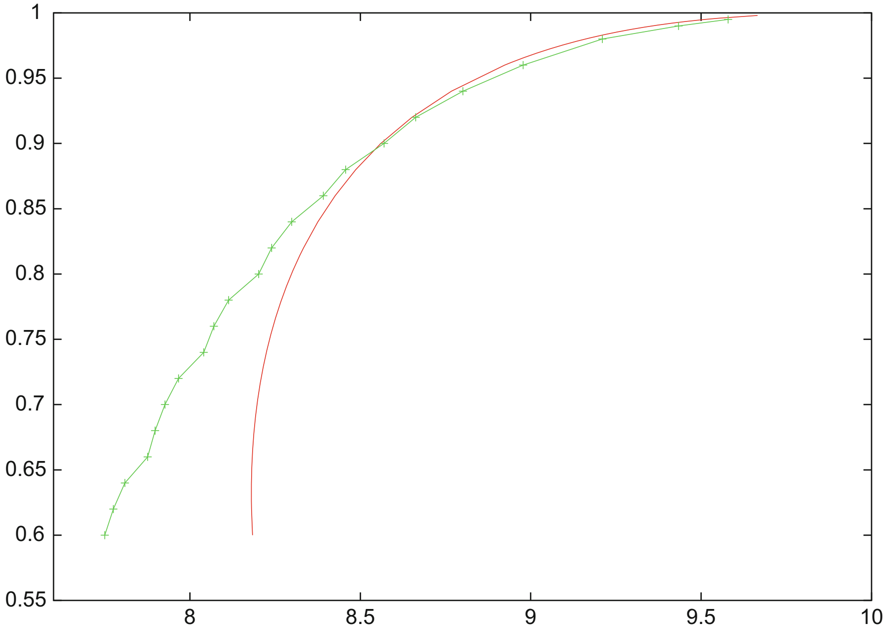

Fig. 4.20 (S,t)-plane. Approximations of an early-exercise curve of an American put ($T = 1$, $\sigma = 0.3$, $K = 10$); *green*: raw data out of an finite-difference approximation, *red*: asymptotic behavior for $t \approx T$. The asymptotic curve is valid only close to the strike K, much smaller than shown here

Figure 4.20 shows a detail of approximations to an early-exercise curve. The finite-difference calculated points are connected by straight lines. The figure also shows a local approximation valid close to maturity: For $t < T$ and $t \to T$, the asymptotic behavior of S_f can be approximated by, for example,

$$S_\mathrm{f}(t) \sim K\left(1 - \sigma\sqrt{(t-T)\log(T-t)}\right)$$

for an American put without dividends [22, 282]. For other asymptotic formulas, see [74, 75, 158]. Recall from the notes on Sect. 4.1 that discrete dividend payments change the early-exercise curve [273]; see also Appendix D.2.

For a proof of the high-contact condition or smooth-pasting principle see [277], p.114. For a discussion of the smoothness of the free boundary S_f see [282] and the references therein.

On Sect. 4.6

By choosing the θ in (4.41) one fixes at which position along the time axis the second-order spatial derivatives are focused. With

$$\theta = \frac{1}{2} - \frac{1}{12}\frac{\Delta x^2}{\Delta\tau}$$

a scheme results that is fourth-order accurate in x-direction. The application on American options requires careful compensation of the discontinuities [265]. One possibility of a variable $\Delta\tau$-time stepping is to set the nodes

$$\tau_\nu := \tau_{\max}\frac{\nu^2}{\nu_{\max}^2}\,,$$

suggested by [188].

Based on the experience of this author, an optimal choice of the relaxation parameter ω_{R} for the iterative variant in Algorithm 4.14 can not be given. The simple strategy $\omega_{\mathrm{R}} = 1$ appears recommendable. The method of Brennan and Schwartz has been analyzed in [201]. The formulation of Problem 4.12 reminds of the dynamic programming principle of (1.69).

On Sect. 4.7

Since the accuracy of the results is not easily guaranteed, it does seem advisable to hesitate before exposing wealth to a chance of loss or damage. After having implemented a finite-difference algorithm it may be recommendable to compare the results with those obtained by means of other algorithms.[18] The lacking smoothness of solutions near $(S,t) \approx (K,T)$ due to the nonsmooth payoff can be largely improved by solving for the difference function $V_{\mathrm{P}}^{\mathrm{Am}}(S,\tau) - V_{\mathrm{P}}^{\mathrm{Eur}}(S,\tau)$, see also Sect. 4.8.2. The lacking smoothness along the early-exercise curve can be diminished by using a front-fixing approach, which can be applied to the above difference. But one mast pay a price. Note that the nonlinearity has entered the front-fixing equation (4.86) ($\longrightarrow$ Exercise 4.8). The success of the front-fixing approach depends on whether the corresponding root-finding iteration finds a solution. Further, in our experience the lack of smoothness is only hidden and might lead to instabilities, such as oscillations in the early-exercise curve. A transformation such as $\log(S/S_{\mathrm{f}})$ does not lead to constant coefficients because one of the factors depends on the early-exercise curve. The alternative front-fixing approach of [188]

[18] As already mentioned in Sect. 4.7, the risk of having chosen an inappropriate model is mostly larger than the risk of inaccurate digits.

first applies the transformation $S = Ke^x$, $\tau = T - t$. Then the infinite (x, τ)-strip is truncated to a finite domain by the function $a(\tau) := x_f(\tau) - L$ for large enough $|L|$ ($L > 0$ for a put, $L < 0$ for a call), where $x_f(\tau) := \log(S_f(T - \tau)/K)$ denotes the transformed early-exercise curve. The final boundary-value problem localized on a rectangle is obtained by transforming the independent variable x to $z := x - a(\tau)$ (for a put). Front-fixing approaches have shown to be highly efficient.

The question how accurate different methods are has become a major concern in recent research; see for instance [83]. Clearly one compares a finite-difference European option with the analytic formulas (A.15)/(A.17). The latter are to be preferred, except the surface $V(S, t)$ is the ultimate object. The correctness of codes can be checked by testing the validity of symmetry relations (A.23).

Greeks such as delta= $\frac{\partial V}{\partial S}$ can be calculated accurately by solving specific PDEs that are derived from the Black–Scholes equation by differentiating. But delta can be approximated easily based on the a calculated approximation of V. To this end, calculate an interpolating Lagrange polynomial $L(S)$ on the line $t = 0$ based on three to five neighboring nodes (Appendix C.1), and take the derivative $L'(S)$.

We have introduced finite differences mainly in view of calculating standard American options. For exotic options PDEs occur, the solutions of which depend on three or more independent variables [21, 353, 376]; see also Chap. 6.

On Sect. 4.8

There are many analytic methods. For example, a binomial tree with a fixed number of nodes can be considered as analytic method. Classic approaches include [63, 150]. Seydel [338] suggests to analyze the attainable accuracy beforehand, depending on the parameters of options, for example, for the interpolation method. The quadratic approximation method has been extended to the more general situation of commodity options, where the cost of carry is involved [26], and a more ambitious initial guess is constructed. Integral representations are based on an inhomogeneous differential equation as that in Sect. 4.5.3. Kim's integral representation (4.76) can be derived via Mellin's transformation [294], or via Duhamel's principle [234], see also [202]. A condition number is derived by [174]. For implementations and improvements, see [175, 211]. The exponential function has been used for approximating the early-exercise curve already in [292]. There are other approaches with integral equations. From the Black–Scholes equation and the high-contact condition we recommend to derive

$$\frac{\partial V_P(S_f(t), t)}{\partial t} = 0 .$$

This equation enables an effective construction of the the early-exercise curve [74, 75].

On Other Methods

Here we give a few hints on methods neither belonging to this chapter on finite differences, nor to Chaps. 5 or 6. General hints can be found in [321], in particular with the references of [57]. Closely related to linear complementarity problems are minimization methods. An efficient realization by means of methods of linear optimization is suggested in [98]. The uniform grid can only be the first step toward more flexible approaches, such as the finite elements to be introduced in Chap. 5. For grid stretching and coordinate transformations see [197, 240]. Spectral methods have shown to be highly efficient, consult [381]. For penalty methods we refer to [133, 288], and to Sect. 6.7. Another possibility to enhance the power of finite differences is the *multigrid* approach; for general expositions see [161, 364]; for application to finance see [81, 293]. An irregular grid based on Sobol points is suggested in [36].

4.11 Exercises

4.1 (Discrete Dividend Payment)
Assume that a stock pays one dividend D at ex-dividend date t_D, with $0 < t_D < T$.

(a) Calculate a corresponding continuous dividend rate δ under the assumptions

$$\dot{S} = -\delta S\,, \quad S(T) = S(0) - D > 0\,.$$

(b) Define for an American put with strike K

$$\tilde{t} := t_D - \frac{1}{r}\log\left(\frac{D}{K} + 1\right)\,.$$

Assume $r > 0$, $D > 0$, and a time instant t in $\tilde{t} < t < t_D$. Argue that instead of exercising early it is reasonable to wait for the dividend.
Note: For $\tilde{t} > 0$, depending on S, early exercise may be reasonable for $0 \le t < \tilde{t}$.

4.2 (Stability of the Fully Implicit Method)
The backward-difference method is defined via the solution of the Eq. (4.18)/(4.19). Prove the stability.
Hint: Use the results of Sect. 4.2.4 and $w^{(\nu)} = A^{-1}w^{(\nu-1)}$.

4.3 (Semidiscretization, Method of Lines)
For a semidiscretization of the Black–Scholes equation (1.5) consider the semidiscretized domain

$$0 \le t \le T\,, \quad S = S_i := i\Delta S\,, \quad \Delta S := \frac{S_{\max}}{m}\,, \quad i = 0, 1, \ldots, m$$

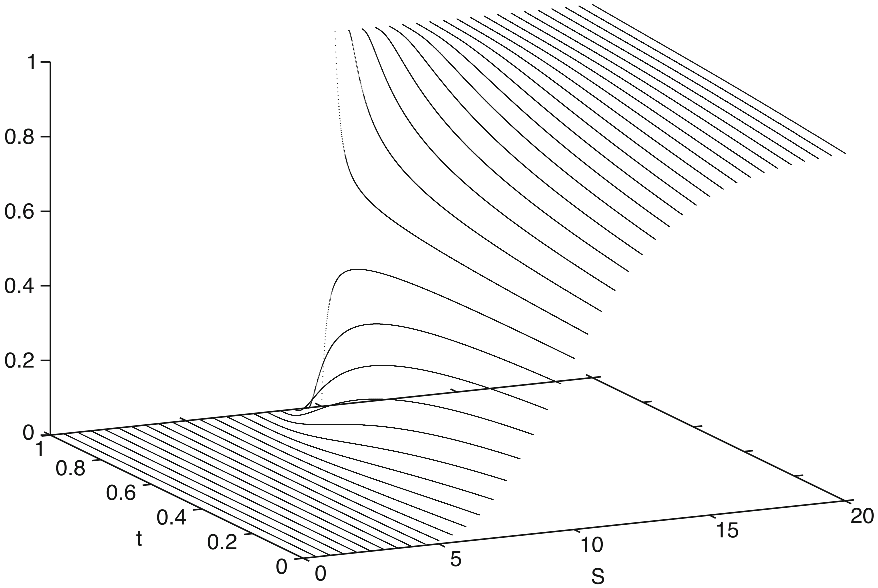

Fig. 4.21 V over (S, t); method of lines for a binary call option, compare Exercise 4.3 ($K = 10$, $T = 1$, $r = 0.06$, $\delta = 0$, $\sigma = 0.3$). With kind permission of Miriam Weingarten

for suitable values of $S_{\max} > K$ and m. On this set of lines parallel to the t-axis define for $\tau := T - t$ and $1 \le i \le m-1$ functions $w_i(\tau)$ as approximation to $V(S_i, \tau)$.

(a) Using the standard second-order difference schemes of Sect. 4.2.1, derive the ODE system $\dot{w} = Bw$ that up to boundary conditions approximates (1.5). Here w is the vector $(w_1, \ldots, w_{m-1})^{tr}$ and $\dot{w}$ denotes differentiation w.r.t. τ . Show that B is a tridiagonal matrix, and calculate its coefficients.

(b) For a European option assume Dirichlet boundary conditions for $w_0(\tau)$ and $w_m(\tau)$ and set up a vector c such that

$$\dot{w} = Bw + c \tag{4.84}$$

realizes the ODE system with correct boundary conditions, and with initial conditions taken from the payoff.

(c) Use the BDF formula (4.11) of Sect. 4.2.1, and implement this scheme for the initial-value problem with (4.84) and a European call option. (See Fig. 4.21 for an illustration.)

4.4 (Crank–Nicolson Order)
Let the function $y(x, \tau)$ solve the equation

$$y_\tau = y_{xx}$$

and be sufficiently smooth. With the difference quotient

$$\delta_{xx} w_{i,\nu} := \frac{w_{i+1,\nu} - 2w_{i,\nu} + w_{i-1,\nu}}{\Delta x^2}$$

the local discretization error ϵ of the Crank–Nicolson method is defined

$$\epsilon := \frac{y_{i,\nu+1} - y_{i,\nu}}{\Delta\tau} - \frac{1}{2}\left(\delta_{xx} y_{i,\nu} + \delta_{xx} y_{i,\nu+1}\right) .$$

Show

$$\epsilon = O(\Delta\tau^2) + O(\Delta x^2) .$$

4.5 (Boundary Conditions of a European Call)
Show that under the transformation (4.3)

$$\begin{aligned} &S\mathrm{e}^{-\delta(T-t)} - K\mathrm{e}^{-r(T-t)} = \\ &\exp\left\{\frac{x}{2}(q_\delta + 1) + \frac{\tau}{4}(q_\delta + 1)^2\right\} - \exp\left\{\frac{x}{2}(q_\delta - 1) + \frac{\tau}{4}(q_\delta - 1)^2\right\} \end{aligned}$$

holds, and prove (4.28).
Hints: Either transform the Black–Scholes equation (4.1) with

$$S := \bar{S}\exp(\delta(T-t))$$

into a dividend-free version to obtain the dividend version of (4.27), or apply the dividend version (A.16) of the put-call parity.

4.6 (Smooth Pasting of the American Put)
Suppose a portfolio consists of an American put and the corresponding underlying. Hence the value of the portfolio is $\Pi := V_{\mathrm{P}}^{\mathrm{Am}} + S$, where S satisfies the SDE (1.47). S_{f} is the value for which we have high contact, compare (4.31).

(a) Show that

$$\mathrm{d}\Pi = \begin{cases} 0 & \text{for } S < S_{\mathrm{f}} \\ \left(\dfrac{\partial V_{\mathrm{P}}^{\mathrm{Am}}}{\partial S} + 1\right)\sigma S\,\mathrm{d}W + O(\mathrm{d}t) & \text{for } S > S_{\mathrm{f}} . \end{cases}$$

(b) Use this to argue

$$\frac{\partial V_{\mathrm{P}}^{\mathrm{Am}}}{\partial S}(S_{\mathrm{f}}(t), t) = -1 .$$

Hint: Use $\mathrm{d}S > 0 \Rightarrow \mathrm{d}W > 0$ for small $\mathrm{d}t$. Assume $\frac{\partial V}{\partial S} > -1$ and construct an arbitrage strategy for $\mathrm{d}S > 0$.

4.7 (Perpetual Put Option)
For $T \to \infty$ it is sufficient to analyze the ODE

$$\frac{\sigma^2}{2}S^2\frac{\mathrm{d}^2V}{\mathrm{d}S^2} + (r-\delta)S\frac{\mathrm{d}V}{\mathrm{d}S} - rV = 0\,.$$

Consider an American put contacting the payoff $(K-S)^+$ at $S = \alpha < K$. Show:

(a) Upon substituting the boundary condition for $S \to \infty$ one obtains

$$V(S) = c\left(\frac{S}{K}\right)^{\lambda_2}\,, \tag{4.85}$$

where $\lambda_2 = \frac{1}{2}\left(1 - q_\delta - \sqrt{(q_\delta - 1)^2 + 4q}\right)$, $\quad q = \frac{2r}{\sigma^2}$, $\quad q_\delta = \frac{2(r-\delta)}{\sigma^2}$ and c is a positive constant. Fix c by using the left-hand boundary $V(\alpha) = K - \alpha$.
Hint: Apply the transformation $S = K\mathrm{e}^x$. (The other root λ_1 drops out.)

(b) V is decreasing and convex.

For $S < \alpha$ the option is exercised; then its intrinsic value is $K - S$. For $S > \alpha$ the option is not exercised and has a value $V(S) > K - S$. The holder of the option decides when to exercise. This means, the holder makes a decision on the contact $S = \alpha$ such that the value of the option becomes maximal [269].

(c) Show: $V'(\alpha_0) = -1$, if α_0 maximizes the value of the option.

4.8 (Front-Fixing for American Options)
Apply the transformation

$$\zeta := \frac{S}{S_\mathrm{f}(t)}\quad,\qquad y(\zeta,t) := V(S,t)$$

to the Black–Scholes equation (4.1).

(a) Show

$$\frac{\partial y}{\partial t} + \frac{\sigma^2}{2}\zeta^2\frac{\partial^2 y}{\partial\zeta^2} + \Big[(r-\delta) - \frac{1}{S_\mathrm{f}}\frac{\mathrm{d}S_\mathrm{f}}{\mathrm{d}t}\Big]\zeta\frac{\partial y}{\partial\zeta} - ry = 0\,. \tag{4.86}$$

(b) Set up the domain for (ζ, t) and formulate the boundary conditions for an American call. (Assume $\delta > 0$.)

(c) (Project) Set up a finite-difference scheme to solve boundary-value problem derived above. The curve $S_\mathrm{f}(t)$ is implicitly defined by the PDE (4.86), with final value $S_\mathrm{f}(T) = \max(K, \frac{r}{\delta}K)$.

4.9 (Boundary Conditions of American Options)
Show that the boundary conditions of American options satisfy

$$\lim_{x\to\pm\infty} y(x,\tau) = \lim_{x\to\pm\infty} g(x,\tau)\,,$$

where g is defined in Problem 4.7.

4.10 (Gauss–Seidel Method as Special Case of SOR)
Let the $n \times n$ matrix $A = (a_{ij})$ be partitioned additively into $A = D - L - U$, with D diagonal matrix, L strict lower triangular matrix, U strict upper triangular matrix, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$. The *Gauss–Seidel method* is defined by

$$(D - L)x^{(k)} = Ux^{(k-1)} + b$$

for $k = 1, 2, \ldots$. Show that with

$$r_i^{(k)} := b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i}^{n} a_{ij}x_j^{(k-1)}$$

and for $\omega_{\mathrm{R}} = 1$ the relation

$$x_i^{(k)} = x_i^{(k-1)} + \omega_{\mathrm{R}}\frac{r_i^{(k)}}{a_{ii}}$$

holds. For general $1 < \omega_{\mathrm{R}} < 2$ this defines the SOR (successive overrelaxation) method.

4.11 (Brennan–Schwartz Algorithm)
Let A be a tridiagonal matrix as in (C.6), and b and g vectors. The system of equations $Aw = b$ is to be solved such that the side condition $w \geq g$ is obeyed componentwise. Assume for the case of a put $w_i = g_i$ for $1 \leq i \leq i_{\mathrm{f}}$ and $w_i > g_i$ for $i_{\mathrm{f}} < i \leq n$, where i_{f} is unknown.

(a) Formulate an algorithm similar as Algorithm C.3 that solves $Aw = b$ in the backward/forward approach. In the final forward loop, for each i the calculated candidate $\tilde{w}_i$ is tested for $w_i \geq g_i$: Set $w_i := \max\{\tilde{w}_i,\ g_i\}$.

(b) Apply the algorithm to the case of a put with A, b, g from Sect. 4.6.1. For the case of a call adapt the forward/backward Algorithm C.3. Incorporate this approach into Algorithm 4.14.

4.12 (American Call)
Formulate the analogue of Algorithm 4.13 for the case of a call.

4.13
Implement Algorithms 4.14 and 4.15.
Test example: Example 1.6 and others.

4.14 (Approximating the Free Boundary)
Assume that after a finite-difference calculation of an American put three approximate values $V(S_i, t)$ are available, for a value of t and $i = k, k+1, k+2$. Assume further an index k such that these three (S, V)-pairs are close to the free boundary $S_\text{f}(t)$, and inside the continuation region.

(a) Derive an approximation $\bar{S}_\text{f}$ to $S_\text{f}(t)$ based on the available data.
(b) Discuss the error $O(\bar{S}_\text{f} - S_\text{f})$.

Hints: The derivative $\frac{\partial V}{\partial S}$ at S_f is -1. For (b) assume an equidistant spacing of the S_i.

4.15 (Extrapolation of Higher Order)
Similar as in Sect. 4.7 assume an error model

$$\eta^* = \eta(\Delta) - \gamma_1 \Delta^2 - \gamma_2 \Delta^3$$

and three calculated values

$$\eta_1 := \eta(\Delta)\,, \quad \eta_2 := \eta\left(\frac{\Delta}{2}\right)\,, \quad \eta_3 := \eta\left(\frac{\Delta}{4}\right)\,.$$

Show that

$$\eta^* = \frac{1}{21}(\eta_1 - 12\eta_2 + 32\eta_3)\,.$$

4.16 (PDE for the Greek Delta)
Derive a PDE-boundary-value problem for the greek delta $\Delta := \frac{\partial V}{\partial S}$ in case of a plain-vanilla put.
Hint: Differentiate the Black-Scholes equation, its terminal condition, and its boundary conditions with respect to S.

4.17

(a) Derive (4.63).
(b) Derive (4.70).

4.18 (Analytic Method for the American Put)
(Project) Implement both the Algorithm 4.17 and Algorithm 4.18. For Algorithm 4.18 choose as initial guess the average of the strike and the lower bound (A.21). A secant method (C.5) is a good choice for the iteration. Think of how to combine Algorithms 4.17 and 4.18 into a hybrid algorithm.

4.19 Consider the functions d_1 and d_2 of (4.75). For the three cases $S < S_\text{f}(\tau)$, $S = S_\text{f}(\tau)$, $S > S_\text{f}(\tau)$, calculate the limit for $\xi \to 0^+$ of

$$rK\text{e}^{-r\xi}\, F(-d_2(S, \xi; S_\text{f}(\tau - \xi))) - \delta S_\text{f}(\tau)\, \text{e}^{-\delta\xi}\, F(-d_1(S, \xi; S_\text{f}(\tau - \xi)))\,.$$

4.20
Implement Kim's integral-equation method (Sect. 4.8.4).

4.21 (Complexity)
With n underlyings and time t an option problem comprises $n + 1$ independent variables. Assume that we discretize each of the $n + 1$ axes with M grid points, then M^{n+1} nodes are involved. Hence the *complexity* C of the n-factor model is

$$C := O(M^{n+1}) ,$$

which amounts to an exponential growth with the dimension, nicknamed *curse of dimension*. Depending on the chosen method, the error E is of the order $M^{-\ell}$,

$$E := O\left(\frac{1}{M^{\ell}}\right) .$$

Argue

$$\log C = -\frac{n+1}{\ell} \log E + \gamma$$

for a method-dependent constant γ.