Chapter 4 Standard Methods for Standard Options

We now enter the part of the book that is devoted to the numerical solution of equations of the Black–Scholes type. Here we discuss "standard" options in the sense as introduced in Section 1.1 and assume the scenario characterized by the Assumptions 1.2. In case of European options the function $V(S,t)$ solves the Black–Scholes equation (1.2). It is not really our aim to solve this partial differential equation because it possesses an analytic solution (⟶ Appendix A4). Ultimately our intention is to solve more general equations and inequalities. In particular, American options will be calculated numerically. The goal is not only to calculate single values $V(S_0,0)$ —for this purpose binomial methods can be applied— but also to approximate the curve $V(S,0)$, or even the surface defined by $V(S,t)$ on the half strip $S>0$, $0 \le t \le T$. Thereby we collect information on early exercise, and on delta hedging by observing the derivative $\frac{\partial V}{\partial S}$.

American options obey *inequalities* of the type of the Black–Scholes equation (1.2). 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 because otherwise early exercise does not make sense for American calls.

The main part of this chapter outlines an 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 the full 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 that can calculate standard American (and European) options. If we work carefully, 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 Chapter 6.

The 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 into several fundamental concepts of numerical mathematics. A trained reader may like to skip Sections 4.2 and 4.3. The aim of this chapter is to introduce concepts,

R.U. Seydel, *Tools for Computational Finance*, Universitext,
DOI: 10.1007/978-3-540-92929-1_4,
© Springer-Verlag Berlin Heidelberg 2009

as well as a characterization of the free boundary (early-exercise curve), and of linear complementarity.

In addition to the classical finite-difference approach, "standard methods" include analytic methods, which to a significant part are based on nonnumerical methods. The final Section 4.8 will give an introduction.

4.1 Preparations

We assume that dividends are paid with a continuous yield of constant level. In case of a discrete payment of, for example, one payment per year, the payment can be converted into a continuous yield ($\longrightarrow$ Exercise 4.1). To this end one has to take into consideration that at the instant of a discrete payment the price $S(t)$ of the asset instantaneously drops by the amount of the payment. This holds true because of the no-arbitrage principle. The 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. To this end the standard model of a geometric Brownian motion represented by the SDE (1.33) is generalized to

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

The corresponding Black–Scholes equation for the value function $V(S,t)$ is

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

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 $0 \leq \tau$, $x \in \mathbb{R}$. This equivalence can be proved by means of the transformations

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

For the slightly simpler case of no dividend payments ($\delta = 0$) the derivation was carried out earlier ($\longrightarrow$ Exercise 1.2). Intrinsic to the transformation

(4.3) is that the constants r, σ, δ must be constants. The transformation 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

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

Linear differential equations with such terms are known as Euler's differential equations; their analysis suggests the transformation $S = K\mathrm{e}^x$. The transformed version in equation (4.2) has constant coefficients (=1), which simplifies implementing numerical algorithms.

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 $\tau = \frac{1}{2}\sigma^2 T$. 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 $S > 0$, $0 \le t \le T$ belonging to (4.1) becomes the strip

$$-\infty < x < \infty, \quad 0 \le \tau \le \tfrac{1}{2}\sigma^2 T\,,$$

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 (1.1C) and (1.1P) become **initial conditions** for $y(x, 0)$. A 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 $y(x, 0)$ in the new variables read

$$\text{call:} \quad y(x, 0) = \max\left\{\mathrm{e}^{\frac{x}{2}(q_\delta + 1)} - \mathrm{e}^{\frac{x}{2}(q_\delta - 1)},\ 0\right\} \tag{4.4C}$$

$$\text{put:} \quad y(x, 0) = \max\left\{\mathrm{e}^{\frac{x}{2}(q_\delta - 1)} - \mathrm{e}^{\frac{x}{2}(q_\delta + 1)},\ 0\right\} \tag{4.4P}$$

The boundary-value problem is completed by imposing boundary conditions for $x \to -\infty$ and $x \to +\infty$ (in Section 4.4).

The equation (4.2) is of the type of a parabolic partial differential equation and is the simplest diffusion or heat-conducting equation. Both equations (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 Chapter 6 in the context of Asian options. These difficulties will not occur for equation (4.2), which is well-suited for standard options with constant coefficients. The equation (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

$$f'(x_i) = \frac{f_{i+1} - f_i}{h} + O(h)\,. \tag{4.5}$$

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 (4.5) is the difference quotient that approximates the differential quotient f' of the left-hand side; the $O(h^p)$-term is the error. The one-sided (i.e. nonsymmetric) difference quotient of (4.5) is of the order $p = 1$. Error orders of $p = 2$ are obtained by central differences

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

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

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

Rearranging terms and indices provides the approximation formula

$$f_i \approx \frac{4}{3}f_{i-1} - \frac{1}{3}f_{i-2} + \frac{2}{3}hf'(x_i)\,, \qquad \text{(BDF2)}$$

which is of second order. The latter difference quotient leads to one example of a *backward differentiation formula* (BDF). Equidistant grids are advantagous in that algorithms are easy to implement, and error terms are easily derivated by Taylor's expansion. This chapter works with equidistant grids.

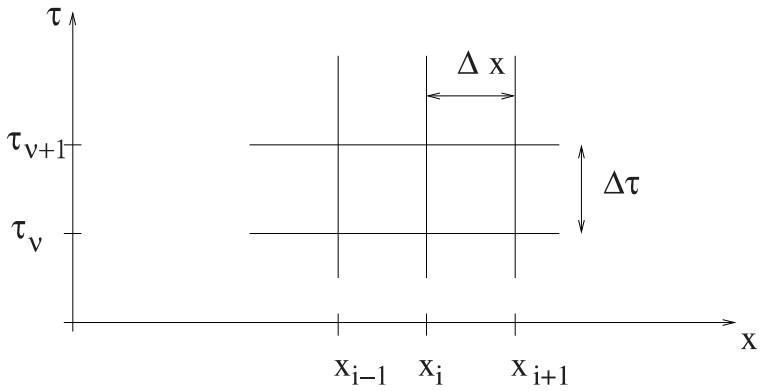

Fig. 4.1. Detail and notations of the grid

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 4.10 and in Section 4.8.3. Here we perform a full discretization leading to a two-dimensional grid.

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 $\nu_{\max}$. The choice of the x-discretization is more complicated. The infinite interval

$-\infty < x < \infty$ must be replaced by a finite interval $a \le x \le b$. Here the end values $a = x_{\min} < 0$ and $b = x_{\max} > 0$ must be chosen 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. For a suitable integer m the step length in x is defined by $\Delta x := (b-a)/m$. Additional notations for the grid are

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

This defines a two-dimensional uniform grid as illustrated in Figure 4.1.[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 $S = S_i = K\mathrm{e}^{x_i}$: The grid is increasingly dense close to $S_{\min}$. (This is not advantagous for the accuracy of the approximations of $V(S, t)$. We will come back to this in Section 5.2.) The 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 (Figure 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 only defined for the nodes. The error $w_{i\nu} - y_{i\nu}$ depends on the choice of parameters $\nu_{\max}$, m, $x_{\min}$, $x_{\max}$. A priori we do not know which choice of parameters matches a prespecified error tolerance. An example of the order of magnitude of these parameters is given by $x_{\min} = -5$, $x_{\max} = 5$, $\nu_{\max} = 100$, $m = 100$. This choice of $x_{\min}$, $x_{\max}$ has shown to be reasonable for a wide range of r, σ-values and accuracies. The actual error is then controlled via the numbers $\nu_{\max}$ und m of grid lines.

4.2.3 Explicit Method

Substituting

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

into (4.2) and discarding the error terms leads to the 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. Solving for $w_{i,\nu+1}$ we obtain

[1] Writing the indices in matrix notation as in $y_{i\nu}$ is meant in the sense $y_{i,\nu}$.

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

With the abbreviation

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

the result is written compactly

$$\boxed{w_{i,\nu+1} = \lambda w_{i-1,\nu} + (1-2\lambda)w_{i\nu} + \lambda w_{i+1,\nu}} \tag{4.6}$$

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

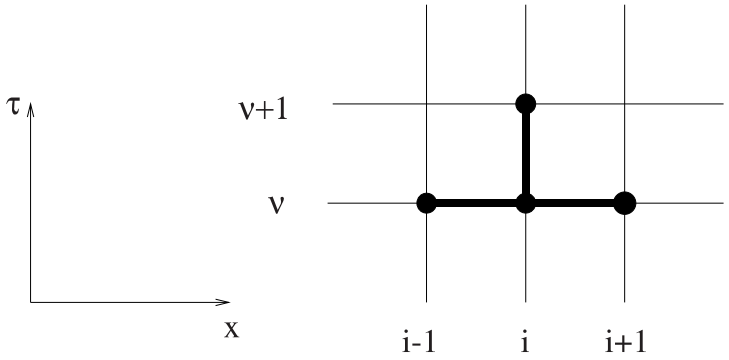

Fig. 4.2. Connection scheme of the explicit method

The equation (4.6) and the Figure 4.2 suggest an evaluation organized by *time levels*. All nodes with the same index ν form the ν-th time level. For a fixed ν the values $w_{i,\nu+1}$ for all i of the time level $\nu+1$ are calculated. Then we advance to the next time level. The formula (4.6) is an explicit expression for each of the $w_{i,\nu+1}$; the values w at level $\nu+1$ are not coupled. Since (4.6) provides an explicit formula for all $w_{i,\nu+1}$ $(i = 0, 1, ..., m)$, this method is called *explicit method* or *forward-difference method*.

Start: For $\nu = 0$ the values of w_{i0} are given by the initial conditions

$$w_{i0} = y(x_i, 0) \quad \text{for } y \text{ from (4.4)}, \; 0 \le i \le m \;\; .$$

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 $w_{0\nu} = w_{m\nu} = 0$. The correct boundary conditions are deferred to Section 4.4.

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

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

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

$$A := A_{\text{expl}} := \begin{pmatrix} 1-2\lambda & \lambda & 0 & \cdots & 0 \\ \lambda & 1-2\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \lambda \\ 0 & \cdots & 0 & \lambda & 1-2\lambda \end{pmatrix}. \tag{4.7a}$$

Now the explicit method in matrix-vector notation reads

$$w^{(\nu+1)} = Aw^{(\nu)} \quad \text{for } \nu = 0, 1, 2, ... \tag{4.7b}$$

The formulation of (4.7) with the matrix A and the iteration (4.7b) is needed only for theoretical investigations. An actual computer program would rather use the version (4.6). The inner-loop index i does not occur explicitly in the vector notation of (4.7).

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

$y_\tau = y_{xx}$, $y(x,0) = \sin \pi x$, $x_0 = 0$, $x_m = 1$, 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 $y(0.2, 0.5) = 0.004227....$ We carry out two calculations with the same $\Delta x = 0.1$ (hence $0.2 = x_2$), and two different $\Delta\tau$:

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

(b) $\Delta\tau = 0.01 \Longrightarrow \lambda = 1$,
$0.5 = \tau_{50}$, $w_{2,50} \doteq -1.5 * 10^8$ (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 the iteration $w^{(\nu+1)} = Aw^{(\nu)}$. In general we use the same notation w for the theoretical definition of w and for the values of w that are 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 (4.7). 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 $\bar{w}^{(\nu)}$ and the error vectors by

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

for $\nu \geq 0$. The result in a computer can be written

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

where the vectors $r^{(\nu+1)}$ amount to rounding errors that occur during the calculation of $A\bar{w}^{(\nu)}$. 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 (4.4), when $\bar{w}^{(0)}$ is calculated. According to this scenario we have $\bar{w}^{(\nu+1)} = A\bar{w}^{(\nu)}$. The relation

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

between consecutive errors is applied repeatedly and results in

$$e^{(\nu)} = A^{\nu} e^{(0)} . \tag{4.8}$$

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

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

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

$$\rho(A) := \max_i |\mu_i^A| ,$$

where $\mu_1^A, ..., \mu_{m-1}^A$ denote the eigenvalues of A. The proof can be found in text books of numerical analysis, for example, in [IK66]. As a consequence of Lemma 4.2 we require for stable behavior that $|\mu_i^A| < 1$ for all eigenvalues, here for $i = 1, ..., m-1$. To check the criterion of Lemma 4.2, the eigenvalues μ_i^A of A are needed. To this end we split the matrix A into

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

It remains to investigate the eigenvalues μ^G of the tridiagonal matrix G.[2]

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 and the eigenvectors $v^{(k)}$ of G are

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

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

Proof: Substitute into $Gv = \mu^G v$.

To apply the lemma observe $N = m-1$, $\alpha = 2$, $\beta = \gamma = -1$, and obtain the eigenvalues μ^G and finally the eigenvalues μ^A of A:

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

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

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

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

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

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

The largest sin-term is $\sin\frac{(m-1)\pi}{2m}$; for increasing m this term grows monotonically approaching 1.

[2] The zeros in the corner of the matrix symbolize the tringular zero structure of (4.7a).

In summary we have shown

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

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

$$0 < \Delta\tau \leq \frac{\Delta x^2}{2} \tag{4.9}$$

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

$$\text{(a)} \quad \lambda = 0.05 \leq \frac{1}{2}$$
$$\text{(b)} \quad \lambda = 1 > \frac{1}{2}$$

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 (4.9) 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 (4.9) bounds $\Delta\tau$ quadratically. This situation suggests searching for a method that is unconditionally stable.

4.2.5 An Implicit Method

When we introduced the explicit method in Subsection 4.2.3, we approximated the time derivative with a forward difference, "forward" as seen from the ν-th time level. Now we try the backward difference

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

which yields the alternative to (4.6)

$$-\lambda w_{i+1,\nu} + (2\lambda + 1)w_{i\nu} - \lambda w_{i-1,\nu} = w_{i,\nu-1} \tag{4.10}$$

The equation (4.10) 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 (4.10) is known, whereas on the left-hand side of the equation three unknown values of w wait to be computed. Equation (4.10) couples three unknowns.

The corresponding molecule is shown in Figure 4.3. There is no simple explicit formula with which the unknown can be obtained one after the other. Rather a system must be considered, all equations simultaneously. A vector notation reveals the structure of (4.10): With the matrix

$$A := A_{\text{impl}} := \begin{pmatrix} 2\lambda+1 & -\lambda & & 0 \\ -\lambda & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix} \tag{4.11a}$$

the vector $w^{(\nu)}$ is implicitly defined as solution of the system of linear equations

$$Aw^{(\nu)} = w^{(\nu-1)} \quad \text{for } \nu = 1, ..., \nu_{\max} \tag{4.11b}$$

Here we again have assumed $w_{0\nu} = w_{m\nu} = 0$. 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 C1) is calculated once. Then the costs for each ν are only of the order $O(m)$.

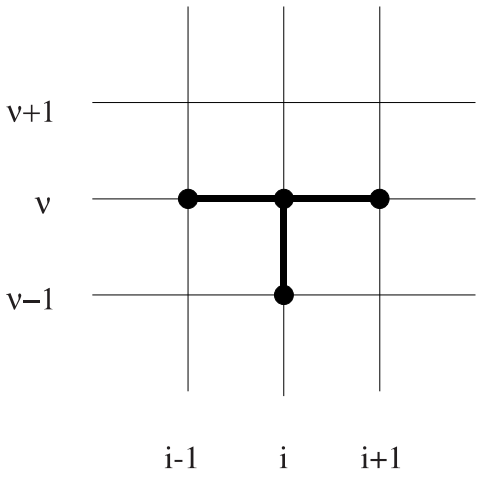

Fig. 4.3. Molecule of the backward-difference method (4.10)

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 equation (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

$$\frac{w_{i,\nu+1} - w_{i\nu}}{\Delta\tau} = \frac{1}{2\Delta x^2}(w_{i+1,\nu} - 2w_{i\nu} + w_{i-1,\nu} + w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1}) \tag{4.12}$$

The equation (4.12) involves in each of the time levels ν and $\nu + 1$ three values w (Figure 4.4). This is the basis of an efficient method. Its features are summarized in Theorem 4.4.

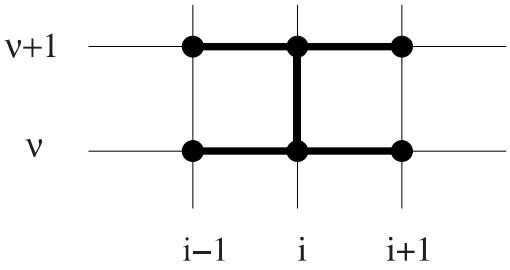

Fig. 4.4. Molecule of the Crank–Nicolson method (4.12)

Theorem 4.4 (Crank–Nicolson)
Suppose y is smooth in the sense $y \in \mathcal{C}^4$. Then:

1.) 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

$$\delta_x^2 w_{i\nu} := \frac{w_{i+1,\nu} - 2w_{i\nu} + w_{i-1,\nu}}{\Delta x^2}\,. \tag{4.13}$$

Apply the operator δ_x^2 to the exact solution y. Then by Taylor expansion for $y \in \mathcal{C}^4$ one can show

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

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

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

Applying the operator δ_x^2 of (4.13) 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 4.3)

2.) system of equations: With $\lambda := \frac{\Delta\tau}{\Delta x^2}$ the equation (4.12) is rewritten

$$\boxed{\begin{aligned} &-\frac{\lambda}{2} w_{i-1,\nu+1} + (1+\lambda) w_{i,\nu+1} - \frac{\lambda}{2} w_{i+1,\nu+1} \\ &= \frac{\lambda}{2} w_{i-1,\nu} + (1-\lambda) w_{i\nu} + \frac{\lambda}{2} w_{i+1,\nu} \end{aligned}} \tag{4.14}$$

The values of the new time level $\nu+1$ are implicitly given by the system of equations (4.14). For the simplest boundary conditions $w_{0\nu} = w_{m\nu} = 0$ equation (4.14) is a system of $m-1$ equations. With matrices

$$\begin{aligned} A := A_{\mathrm{CN}} &:= \begin{pmatrix} 1+\lambda & -\frac{\lambda}{2} & & 0 \\ -\frac{\lambda}{2} & \ddots & \ddots & \\ & \ddots & \ddots & -\frac{\lambda}{2} \\ 0 & & -\frac{\lambda}{2} & 1+\lambda \end{pmatrix}, \\ B := B_{\mathrm{CN}} &:= \begin{pmatrix} 1-\lambda & \frac{\lambda}{2} & & 0 \\ \frac{\lambda}{2} & \ddots & \ddots & \\ & \ddots & \ddots & \frac{\lambda}{2} \\ 0 & & \frac{\lambda}{2} & 1-\lambda \end{pmatrix} \end{aligned} \tag{4.15a}$$

the system (4.14) is rewritten

$$Aw^{(\nu+1)} = Bw^{(\nu)}\,. \tag{4.15b}$$

The eigenvalues of A are real and lie between 1 and $1+2\lambda$. (This follows from the Theorem of Gerschgorin, see Appendix C1). This rules out a zero eigenvalue, and so A must be nonsingular and the solution of (4.15b) is uniquely defined.

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

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

Now the equation (4.15b) 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

$$w^{(\nu+1)} = (4C^{-1} - I)w^{(\nu)}\,. \tag{4.16}$$

The eigenvalues μ_k^C of C for $k = 1, ..., m-1$ are known from Lemma 4.3,

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

In view of (4.16) 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 (4.15) is unconditionally stable for all $\lambda > 0$ $(\Delta\tau > 0)$.

Although the 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 (4.4), $0 \le i \le m$
Calculate the LR-decomposition of A
loop: *for* $\nu = 0, 1, ..., \nu_{\max} - 1$:
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. It is obvious that 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 the correct boundary conditions.

4.4 Boundary Conditions

The Black–Scholes equation (4.1), the transformed version (4.2), and the discretized versions of the previous sections, they all need boundary conditions. In particular, the values

$V(S, t)$ for $S = 0$ and $S \to \infty$, or
$y(x, \tau)$ for $x_{\min}$ and $x_{\max}$, or
$w_{0\nu}$ and $w_{m\nu}$ for $\nu = 1, ..., \nu_{\max}$,

respectively, must be prescribed by boundary conditions. The preliminary homogenous boundary conditions $w_{0\nu} = w_{m\nu} = 0$ of the previous sections do not match the scenario of Black, Merton and Scholes. In order to complete and adapt the Algorithm 4.5 we must define realistic boundary conditions.

The boundary conditions for the expiration time $t = T$ are obvious. They give rise to the simplest cases of boundary conditions for $t < T$: As motivated by the Figures 1.1 and 1.2 and the equations (1.1C), (1.1P), the value V_{C} of a call and the value V_{P} of a put must satisfy

$$\begin{aligned} V_{\mathrm{C}}(S, t) &= 0 \quad \text{for } S = 0, \text{ and} \\ V_{\mathrm{P}}(S, t) &= 0 \quad \text{for } S \to \infty \end{aligned} \tag{4.17}$$

also for all $t < T$. This follows from the integral representation (3.20), because discounting does not affect the value 0 of the payoff. And $S(0) = 0$ implies

$S(t) = 0$ for all $t > 0$ because of $\mathrm{d}S = S(\mu\,\mathrm{d}t + \sigma\,\mathrm{d}W)$; hence the value $V_{\mathrm{C}}(0,t) = 0$ can be predicted safely. The same holds true for $S(0) \to \infty$ and V of (1.1P). This holds 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 the American option to the next section, and investigate European options in this section.

From the put-call parity ($\longrightarrow$ Exercise 1.1) we deduce the additional boundary conditions for European options without dividend payment ($\delta = 0$). The result is

$$\begin{aligned} V_{\mathrm{C}}(S,t) &= S - K\mathrm{e}^{-r(T-t)} \quad && \text{for } S \to \infty \\ V_{\mathrm{P}}(S,t) &= K\mathrm{e}^{-r(T-t)} - S \quad && \text{for } S \approx 0\,. \end{aligned} \tag{4.18}$$

The lower bounds for European options ($\longrightarrow$ Appendix D1) are attained at the boundaries. In (4.18) for $S \approx 0$ we do not discard the term S, because the realization of the transformation (4.3) requires $S_{\min} > 0$, see Section 4.2.2. Boundary conditions analogous as in (4.18) hold for the case of a continuous flow of dividend payments ($\delta \neq 0$). We skip the derivation, which can be based on transformation (4.3) and the additional transformation $S = \overline{S}\mathrm{e}^{\delta(T-t)}$ ($\longrightarrow$ Exercise 4.4). In summary, the boundary conditions for European options in the (x,τ)-world are as follows:

Boundary Conditions 4.6 (European options)

$$\begin{aligned} &y(x,\tau) = r_1(x,\tau) \text{ for } x \to -\infty\,, \\ &y(x,\tau) = r_2(x,\tau) \text{ for } x \to \infty\,, \qquad \text{with} \\ &\text{call: } r_1(x,\tau) := 0\,, \\ &\qquad r_2(x,\tau) := \exp\left(\tfrac{1}{2}(q_\delta+1)x + \tfrac{1}{4}(q_\delta+1)^2\tau\right) \\ &\text{put: } r_1(x,\tau) := \exp\left(\tfrac{1}{2}(q_\delta-1)x + \tfrac{1}{4}(q_\delta-1)^2\tau\right)\,, \\ &\qquad r_2(x,\tau) := 0 \end{aligned} \tag{4.19}$$

Truncation: Instead of the theoretical domain $-\infty < x < \infty$ the practical realization truncates the infinite interval to the finite interval

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

see Section 4.2.2. This suggests 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 are explicit formulas and easy to implement. To this end return to the Crank–Nicolson equation (4.14), 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

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

and for $i = m - 1$:

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

These known boundary values are collected on the right-hand side of system (4.14). So we finally arrive at

$$\boxed{\begin{aligned} Aw^{(\nu+1)} &= Bw^{(\nu)} + d^{(\nu)} \\ d^{(\nu)} &:= \frac{\lambda}{2} \cdot \begin{pmatrix} r_1(a, \tau_{\nu+1}) + r_1(a, \tau_\nu) \\ 0 \\ \vdots \\ 0 \\ r_2(b, \tau_{\nu+1}) + r_2(b, \tau_\nu) \end{pmatrix} \end{aligned}} \tag{4.20}$$

The previous version (4.15b) 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 Section 4.2 can be adapted by analogous formulas. The stability is not affected by adding the vector d, which is constant with respect to w.

4.5 American Options as Free Boundary Problems

In Sections 4.1 through 4.3 we so far 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. So we have elementary lower bounds for the value of American options, but —as we will 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, Figure 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

$$\begin{aligned} V_{\mathrm{P}}^{\mathrm{Am}}(S,t) &\geq (K-S)^+ \quad \text{for all } (S,t) \\ V_{\mathrm{C}}^{\mathrm{Am}}(S,t) &\geq (S-K)^+ \quad \text{for all } (S,t) \end{aligned} \tag{4.21}$$

hold. This result is illustrated schematically for a put in Figure 4.5.

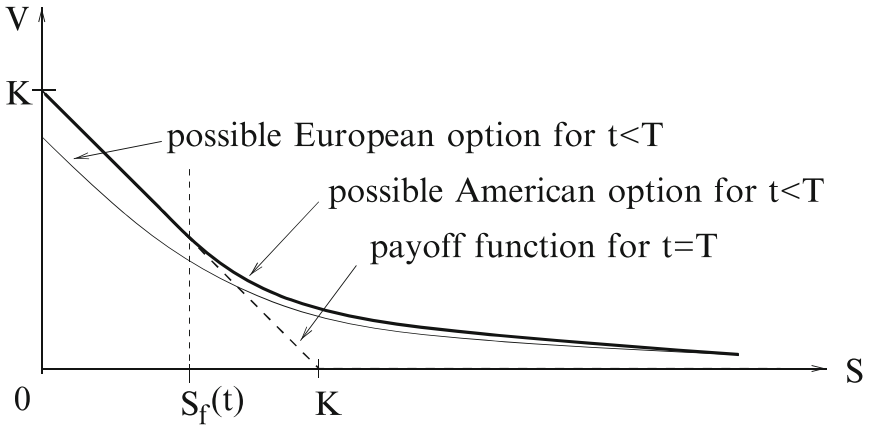

Fig. 4.5. $V(S,t)$ for a put and a $t < T$, schematically

For American options we have noted in (4.17) 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 (4.21) it is clear that the missing boundary conditions will be of a different kind than those for European options, which are listed in (4.18). Let us investigate the situation of an **American put**, which is illustrated in Figure 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 $V_{\mathrm{P}}(S,t) < K - S$ holds for $r > 0$ and sufficiently small S. But in view of (4.21) the American put should satisfy $V_{\mathrm{P}}(S,t) \equiv K - S$ 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 the curve $V_{\mathrm{P}}^{\mathrm{Am}}(S,t)$ hits the straight line of the payoff at some value S_{f} with $0 < S_{\mathrm{f}} < K$, see Figure 4.5. This **contact point** S_{f} is defined by

$$\begin{aligned} V_{\mathrm{P}}^{\mathrm{Am}}(S,t) &> (K-S)^+ \quad \text{for } S > S_{\mathrm{f}}(t), \\ V_{\mathrm{P}}^{\mathrm{Am}}(S,t) &= K - S \qquad \text{for } S \leq S_{\mathrm{f}}(t)\,. \end{aligned} \tag{4.22}$$

For $S < S_f$ the value V_P^{Am} equals the straight line of the payoff and nothing needs to be calculated. For each t, the *curve* $V_P^{Am}(S,t)$ reaches its left boundary at $S_f(t)$.

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

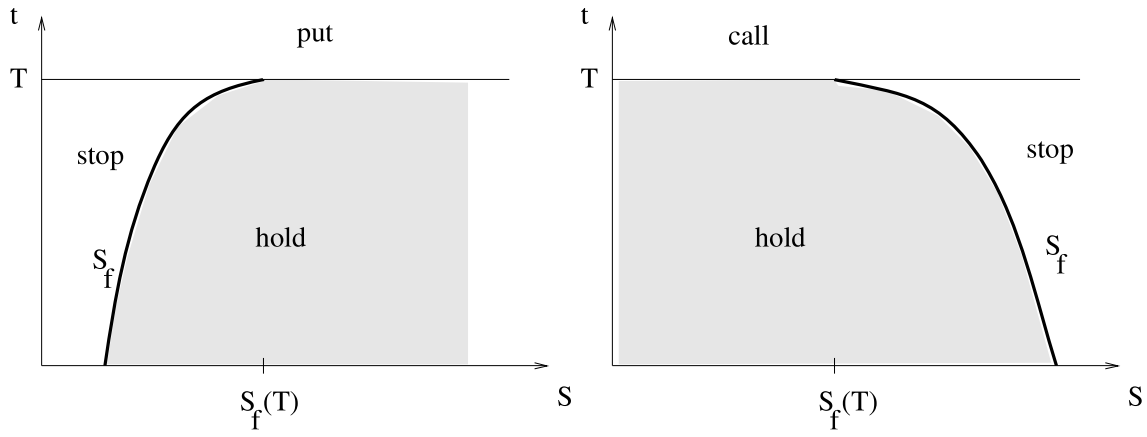

Fig. 4.6. Continuation region (shaded) and stopping region for American options

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

$$V_C^{Am} \ge V_C^{Eur} \ge S - Ke^{-r(T-t)} > S - K$$

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 Figure 4.9. And the free boundary S_f may look like the right-hand diagram of Figure 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 Figure 3.10. Let us explain this for the case of a put; for a call with $\delta \neq 0$ the argument is similar.

In case $S > S_f$, early-exercise causes an immediate loss, because (4.22) implies $-V + K - S < 0$. Receiving the strike price K does not compensate the loss of S and V. Accordingly, the holder of the option does not exercise when $S > S_f$. This explains why the area $S > S_f$ is called **continuation region** (shaded in Figure 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 for the remaining time period $T - t$. The resulting profit will be

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

To maximize the profit, the holder of the option will maximize $T - t$, and accordingly exercise 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**. — So much for the basic principle. Of course, the profit depends on $r > 0$, and the holder must watch the market, see [Hull00].

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 Figures 4.5 and 4.6. In fact, the curves $S_f(t)$ are continuously differentiable in t, and monotonous not decreasing / not increasing as illustrated. There are both upper and lower bounds to $S_f(t)$. For more details and proofs see Appendix A5. Here we confine ourselves to the bounds given by the limit $t \to T$ $(t < T,\ \delta > 0)$:

$$\text{put:} \quad \lim_{t \to T^-} S_f(t) = \min(K, \frac{r}{\delta}K) \tag{4.23P}$$

$$\text{call:} \quad \lim_{t \to T^-} S_f(t) = \max(K, \frac{r}{\delta}K) \tag{4.23C}$$

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 Figure 4.7 for a put. For the chosen numbers for all $\delta \leq 0.06$ the limit of (4.23P) is the strike K (lower diagram). Compare to Figures 1.4 and 1.5, and to the title figure of this book to get a feeling for the geometrical importance of the curve as contact line where two surfaces merge. For larger values of S the surface $V(S,t)$ approaches 0 in a way illustrated by Figure 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. In order to calculate the free boundary $S_f(t)$ we need an additional condition. To this end consider the slope $\frac{\partial V}{\partial S}$ with which $V_P^{Am}(S,t)$ touches at $S_f(t)$ the straight line $K - S$, which has the constant slope -1. By geometrical reasons we can rule out for V_P^{Am} the case $\frac{\partial V(S_f(t),t)}{\partial S} < -1$, because otherwise (4.21) and (4.22) would be violated. Using arbitrage arguments, the case $\frac{\partial V(S_f(t),t)}{\partial S} > -1$ can also be ruled out ($\longrightarrow$ Exercise 4.9). It remains the condition $\partial V_P^{Am}(S_f(t),t)/\partial S = -1$. 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

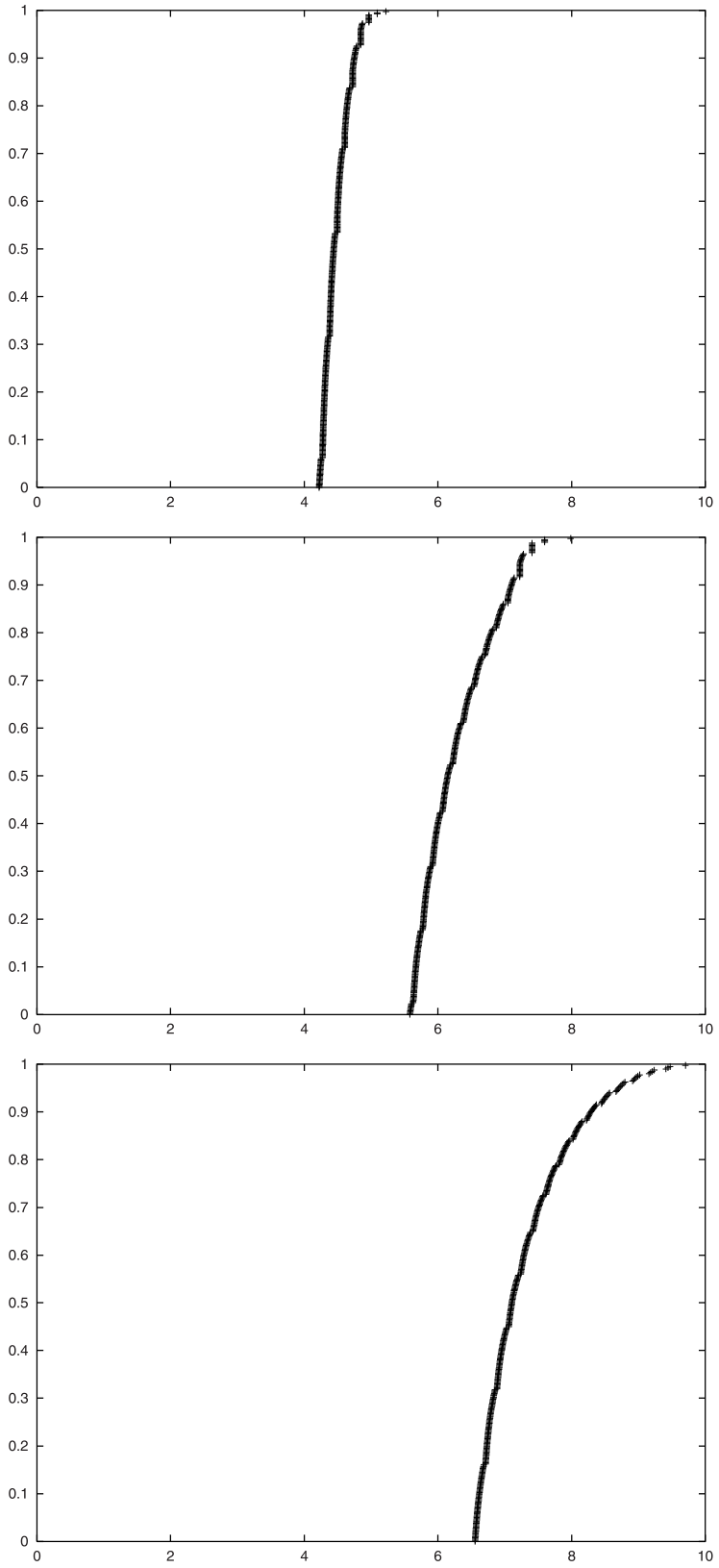

Fig. 4.7. Early-exercise curves of an American put, $r = 0.06$, $\sigma = 0.3$, $K = 10$, and dividend rates $\delta = 0.12$ (top figure), $\delta = 0.08$ (middle), $\delta = 0.04$ (bottom); raw data of a finite-difference calculation without interpolation or smoothing

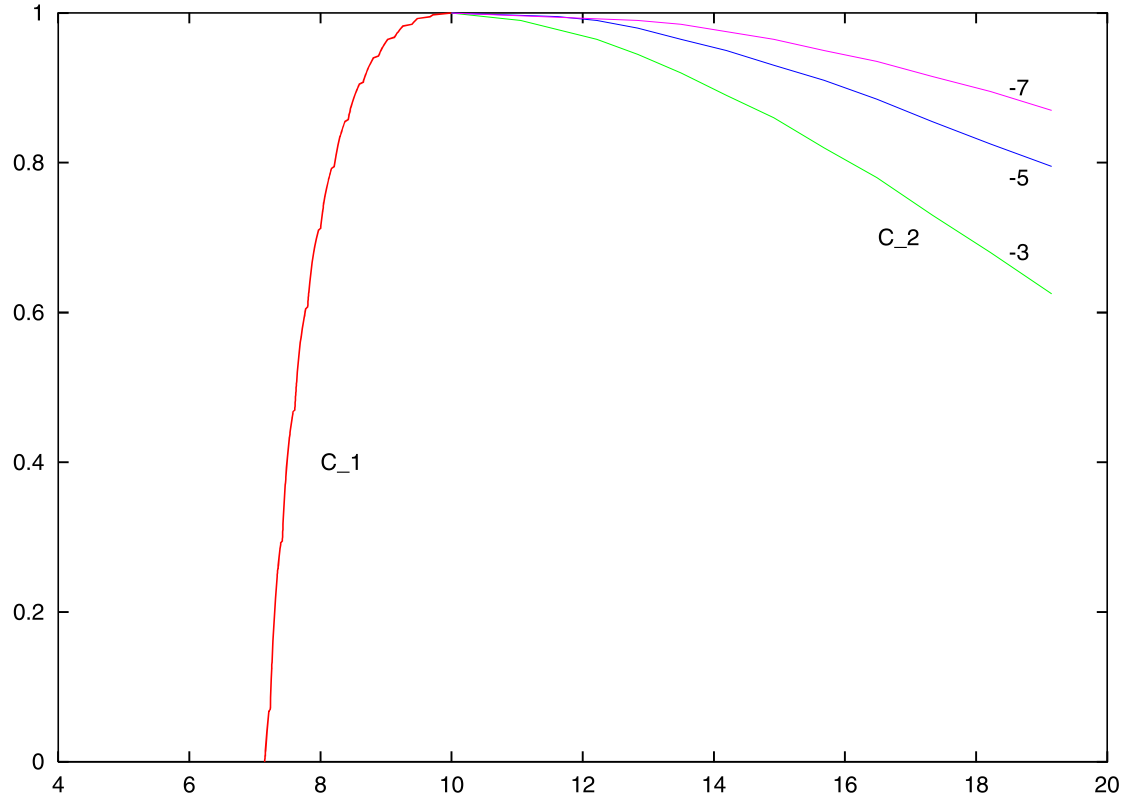

Fig. 4.8. Calculated curves of a put matching Figures 1.4, 1.5. C_1 is the curve S_f. The three curves C_2 have the meaning $V < 10^{-k}$ for $k = 3, 5, 7$.

somewhat hypothetical case of a *perpetual option* ($T = \infty$) the tangential touching can be calculated analytically ($\longrightarrow$ Exercise 4.8). In summary, *two boundary conditions* must hold at the contact point $S_\mathrm{f}(t)$:

$$\begin{aligned} V_\mathrm{P}^\mathrm{Am}(S_\mathrm{f}(t),t) &= K - S_\mathrm{f}(t) \\ \frac{\partial V_\mathrm{P}^\mathrm{Am}(S_\mathrm{f}(t),t)}{\partial S} &= -1 \end{aligned} \tag{4.24P}$$

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

$$\begin{aligned} V_\mathrm{C}^\mathrm{Am}(S_\mathrm{f}(t),t) &= S_\mathrm{f}(t) - K \\ \frac{\partial V_\mathrm{C}^\mathrm{Am}(S_\mathrm{f}(t),t)}{\partial S} &= 1 \end{aligned} \tag{4.24C}$$

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 an American call on a dividend-paying asset. The high contact on the payoff is visible.

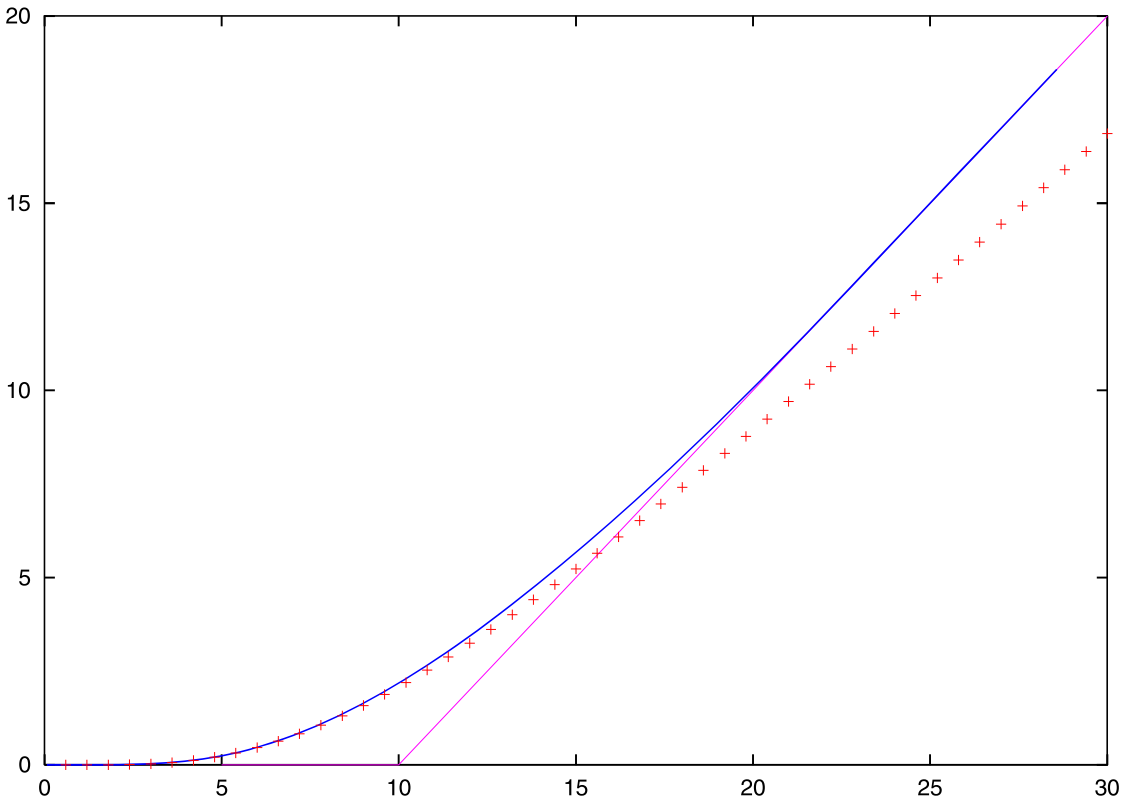

Fig. 4.9. Value $V(S,0)$ of an American call with $K = 10$, $r = 0.25$, $\sigma = 0.6$, $T = 1$ and dividend flow $\delta = 0.2$. Crosses indicate the corresponding curve of a European call; the payoff is shown. A special value is $V(K,0) = 2.18728$.

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. This *front-fixing* approach is numerically relevant ($\longrightarrow$ Exercise 4.11).

4.5.3 Black–Scholes Inequality

The Black–Scholes equation (4.1) is valid on the continuation region (shaded areas in Figure 4.6). For the numerical approach of the following Section 4.6 the computational domain will be the entire half strip with $S > 0$, including the stopping areas. 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 we substitute the payoff into $\frac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V)$ for the case of a put. (The reader may carry out the analysis for the case of a call.) For the put, for $S \le S_{\mathrm{f}}$,

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

Hence

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

From (4.23P) we have the bound $\delta S < rK$, which leads to conclude

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

The Black–Scholes equation changes to an *inequality* on the stopping region. The same inequality holds for the call. In summary, on the entire half strip American options must satisfy an *inequality* of the Black–Scholes type,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r-\delta)S\frac{\partial V}{\partial S} - rV \le 0\,. \tag{4.25}$$

The inequalities (4.21) and (4.25) hold for all (S,t). In case the strict inequality "$>$" holds in (4.21), equality holds in (4.25). The contact boundary S_{f} divides the half strip into the stopping region and the continuation region, each with appropriate version of V:

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

and

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

This shows that also for American options the Black–Scholes equation (4.1) must be solved, 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}.

Note that $\frac{\partial V}{\partial S}$ is continuous when S_{f} is crossed, 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 Obstacle Problem

A brief digression into obstacle problems will motivate the procedure. We assume an "obstacle" $g(x)$, say with $g(x) > 0$ for $\alpha < x < \beta$, $g \in \mathcal{C}^2$, $g'' < 0$ and $g(-1) < 0,\ g(1) < 0$, compare Figure 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, where the curve of u touches the obstacle tangentially. These two values $x = \alpha$ and $x = \beta$ are unknown initially. This obstacle problem is a simple free boundary problem.

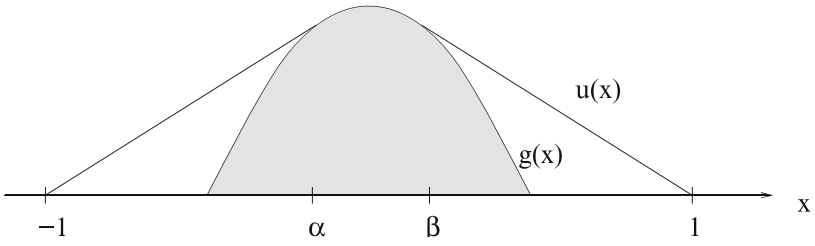

Fig. 4.10. Function $u(x)$ across an obstacle $g(x)$

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 Figure 4.10 is defined by the requirement $u \in \mathcal{C}^1[-1,1]$, and by:

$$\begin{array}{lll} \text{for } -1 < x < \alpha: & u'' = 0 & (\text{then } u > g) \\ \text{for } \alpha < x < \beta: & u = g & (\text{then } u'' = g'' < 0) \\ \text{for } \beta < x < 1: & u'' = 0 & (\text{then } u > g) \end{array}$$

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

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

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

$$\begin{aligned} &\text{if } V > \text{payoff, then Black–Scholes } \textit{equation } \tfrac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V) = 0 \\ &\text{if } V = \text{payoff, then Black–Scholes } \textit{inequality } \tfrac{\partial V}{\partial t} + \mathcal{L}_{\mathrm{BS}}(V) < 0 \end{aligned}$$

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

$$\begin{cases} \text{find a function } u \text{ such that} \\ u''(u-g) = 0, \quad -u'' \geq 0, \quad u - g \geq 0\,, \\ u(-1) = u(1) = 0,\ u \in \mathcal{C}^1[-1,1]\,. \end{cases} \tag{4.26}$$

The key line (4.26) 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. If a solution to (4.26) is known, then α and β are read off from the solution. So we construct a numerical solution procedure for the complementarity version (4.26) 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

$$\left\{ \begin{array}{l} (w_{i-1} - 2w_i + w_{i+1})(w_i - g_i) = 0, \\ -w_{i-1} + 2w_i - w_{i+1} \geq 0, \quad w_i \geq g_i \end{array} \right\} \; 0 < i < m \,,$$

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 this end define a vector notation using

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

Then the discretized complementarity problem is rewritten in the form

$$\left\{ \begin{array}{l} (w - g)^{tr} Bw = 0 \,, \\ Bw \geq 0 \,, \quad w \geq g \end{array} \right. \tag{4.27}$$

To calculate (4.27) one solves $Bw = 0$ under the side condition $w \geq g$. This will be explained in Section 4.6.2.

4.5.5 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 Section 4.6. The transformations (4.3) lead to

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

Also the side condition (4.21) 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

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

This function g allows to write the initial condition (4.4) as $y(x,0) = g(x,0)$. In summary, we require $y_\tau = y_{xx}$ as well as

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

and, in addition, the 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

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

The same holds for $x \to -\infty$ ($\longrightarrow$ Exercise 4.5). In practice, the boundary conditions are formulated for $x_{\min}$ and $x_{\max}$. Collecting all expressions, the American put is formulated as linear complementarity problem:

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

The exercise boundary is automatically captured by this formulation. An analogous formulation holds for the American call. Both of the formulations are listed in the beginning of the following section. We will return to the obstacle problem with a version as variational problem in Section 5.3.

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)

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

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

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

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

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

$$y(x,0) = g(x,0), \quad 0 \leq \tau \leq \frac{1}{2}\sigma^2 T$$

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

As outlined in Section 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 the numerical solution of Problem 4.7.

4.6.1 Discretization with Finite Differences

We use the same grid as in Section 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,

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

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

$$\begin{aligned} w_{i,\nu+1} &- \lambda\theta(w_{i+1,\nu+1} - 2w_{i,\nu+1} + w_{i-1,\nu+1}) \\ &- w_{i\nu} - \lambda(1-\theta)(w_{i+1,\nu} - 2w_{i\nu} + w_{i-1,\nu}) \geq 0\,, \end{aligned} \tag{4.28}$$

where we use again 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
$b^{(\nu)} := (b_{1\nu}, ..., b_{m-1,\nu})^{tr}$
$w^{(\nu)} := (w_{1\nu}, ..., w_{m-1,\nu})^{tr}$
$g^{(\nu)} := (g_{1\nu}, ..., g_{m-1,\nu})^{tr}$

and

$$A := \begin{pmatrix} 1+2\lambda\theta & -\lambda\theta & & 0 \\ -\lambda\theta & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix} \in \mathbb{R}^{(m-1)\times(m-1)} \tag{4.29}$$

(4.28) is rewritten in vector form as

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

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

$$w_{i0} = g_{i0}\,, \quad i = 1, ..., m-1\,, \quad (w^{(0)} = g^{(0)})\,;$$
$$w_{0\nu} = g_{0\nu}\,, \quad w_{m\nu} = g_{m\nu}\,, \quad \nu \geq 1$$

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

$$\begin{aligned} &b_{2\nu}, ..., b_{m-2,\nu} \quad \text{as defined above,} \\ &b_{1\nu} = w_{1\nu} + \lambda(1-\theta)(w_{2\nu} - 2w_{1\nu} + g_{0\nu}) + \lambda\theta g_{0,\nu+1} \\ &b_{m-1,\nu} = w_{m-1,\nu} + \lambda(1-\theta)(g_{m\nu} - 2w_{m-1,\nu} + w_{m-2,\nu}) + \lambda\theta g_{m,\nu+1} \end{aligned} \tag{4.30}$$

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

Algorithm 4.8 (computation of American options)

For $\nu = 0, 1, ..., \nu_{\max} - 1$:
Calculate the vectors $g := g^{(\nu+1)}$,
$b := b^{(\nu)}$ from (4.29), (4.30).
Calculate the vector w as solution of the problem
$Aw - b \geq 0, \quad w \geq g, \quad (Aw-b)^{tr}(w-g) = 0.$ (4.31)
$w^{(\nu+1)} := w$

This completes the chosen finite-difference discretization.

The remaining problem is to solve the complementarity problem in matrix-vector form (4.31). In principle, how to solve (4.31) 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.

Recall from Subsection 4.5.3 that $V(S,t)$ is not C^2-smooth over the free boundary S_f. This is a source of possible inaccuracies. The order two 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 (4.31).

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 [WDH96]) and a direct approach (following [BrS77]) 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 (4.31) must be solved. This is the bulk of work in Algorithm 4.8. Before entering the 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.[3] For the analysis we transform problem (4.31) from the w-world into an x-world with

$$\begin{aligned} x &:= w - g \\ y &:= Aw - b\,. \end{aligned} \tag{4.32}$$

Then it is easy to see (the reader may check) that the task of calculating a solution w for (4.31) is equivalent to the following problem:

[3] Notation: In this Subsection 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.

Problem 4.9 (Cryer)

$$\boxed{\begin{array}{l}\text{Find vectors } x \text{ and } y \text{ such that for } \hat{b} := b - Ag \\ Ax - y = \hat{b}\,, \quad x \geq 0\,, \quad y \geq 0\,, \quad x^{tr} y = 0\,.\end{array}} \tag{4.33}$$

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
The Problem 4.9 is equivalent to the minimization problem

$$\min_{x \geq 0} G(x), \quad \text{where } G(x) := \frac{1}{2}(x^{tr} Ax) - \hat{b}^{tr} x \quad \text{is strictly convex.} \tag{4.34}$$

Proof. The derivatives of G are $G_x = Ax - \widehat{b}$ 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 Kuhn and Tucker minimizes G under $H_i(x) \leq 0$, $i = 1, \ldots, m$. According to this theorem,[4] a vector x_0 to be a minimum is equivalent to the existence of a Lagrange multiplier $y \geq 0$ with

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

The set $x \geq 0$ leads to define $H(x) := -x$. Hence the Kuhn–Tucker condition is $Ax - \widehat{b} + (-I)^{tr} y = 0$, $y^{tr} x = 0$, and we have reached equation (4.33).

An iterative procedure can be derived from the minimization problem stated in Lemma 4.10. This algorithm is based on the SOR method [Cr71]. For an introduction into iterative methods for the solution of systems of linear equations $Ax = b$ we refer to Appendix C2. Note that (4.31) is not in the easy form of equation $Ax = b$ discussed in Appendix C2; a modification of the standard SOR will be necessary. The iteration of the SOR method for $Ax = \hat{b} = b - Ag$ is written componentwise ($\longrightarrow$ Exercise 4.6) as iteration for the correction vector $x^{(k)} - x^{(k-1)}$:

$$r_i^{(k)} := \hat{b}_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - a_{ii} x_i^{(k-1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \tag{4.35a}$$

$$x_i^{(k)} = x_i^{(k-1)} + \omega_{\mathrm{R}} \frac{r_i^{(k)}}{a_{ii}}\,. \tag{4.35b}$$

[4] For the Kuhn–Tucker theory we refer to [SW70], [St86].

Here k denotes the number of the iteration, $n = m-1$, and in the cases $i = 1$, $i = m-1$ one of the sums in (4.35a) 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 (4.33) starts from a vector $x^{(0)} \geq 0$ and is identical to the SOR method up to a modification on (4.35b) serving for $x_i^{(k)} \geq 0$.

Algorithm 4.11 (PSOR, projected SOR for Problem 4.9)

$$
\begin{aligned}
&\textit{outer loop: } k = 1, 2, \ldots \\
&\textit{inner loop: } i = 1, ..., m-1 \\
&\qquad r_i^{(k)} \quad \text{as in (4.35a)} \\
&\qquad x_i^{(k)} = \max\left\{0,\ x_i^{(k-1)} + \omega_{\mathrm{R}} \frac{r_i^{(k)}}{a_{ii}}\right\}, \\
&\qquad y_i^{(k)} = -r_i^{(k)} + a_{ii}\left(x_i^{(k)} - x_i^{(k-1)}\right)
\end{aligned}
\tag{4.36}
$$

We see that this method 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 against y_i, are not used explicitly for the algorithm. But since $y \geq 0$ is shown ($Aw \geq b$), the vector y serves an important role in the proof of convergence. Transformed back into the w-world of problem (4.31) by means of (4.32), the Algorithm 4.11 solves (4.31). It gives a solution to Problem 4.12:

Problem 4.12 (Cryer's problem restated)

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

Adapting Algorithm 4.11 with formula (4.36) from $x \geq 0$ to $w \geq g$ is easy.

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 $\{x | G(x) \leq G(x^{(0)})\}$;
The vector r from (4.35) converges toward $-y$;
Assuming $r \geq 0$ and $r^{\dagger} x \neq 0$ leads to a contradiction to $x^{(k+1)} - x^{(k)} \to 0$.
(For the proof see [Cr71].)

The above theory has established that it suffices to solve Problem 4.12; $Aw \geq b$ is guaranteed to hold. Because of the uniqueness of the solution, we will obtain the same result w when a direct method is applied instead of the iterative PSOR Algorithm 4.11. The structure of Problem 4.12 is not much different from the system $Aw = b$ without side condition. Recall that in a first phase a direct method establishes an equivalent system $\tilde{A}w = \tilde{b}$ with a triangular matrix $\tilde{A}$. The elimination of the components w_i is the second phase of a direct method. Obeying the side condition $w \geq g$ is easy to arrange for standard options. As analyzed earlier, for a put $w_i = g_i$ for small indexes i, and for a call this holds for the large indices. In both cases there is only one index i_f separating the components with $w_i = g_i$ from those with $w_i > g_i$. For a put and the unknown index i_f,

$$w_i = g_i \text{ for } 1 \leq i \leq i_\mathrm{f}, \quad \text{and } w_i > g_i \text{ for } i_\mathrm{f} < i \leq m\,.$$

The index i_f marks the location of the free boundary. So, as suggested by Brennan and Schwartz [BrS77], the elimination procedure runs *forward* for a put, starting with $i = 1$. To have the elimination phase run in a forward loop, the matrix $\tilde{A}$ must be a *lower* triangular matrix. That is, in the case of a put, the decomposion is a RL-decomposition (⟶ Appendix C1). After starting with $i = 1$, the algorithm for $i > 1$ then always calculates the next component w_i of $Aw = b$, and sets $w_i := g_i$ in case $w_i < g_i$. For the call, where the elimination phase runs in a backward loop, the traditional upper triangular matrix $\tilde{A}$ is calculated by the LR-decomposion. 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. (⟶ Exercise 4.12)

4.6.3 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 (4.31) into Algorithm 4.8. From the analysis of Subsection 4.6.2, we either apply the iterative Algorithm 4.11 (⟶ Exercise 4.7), or implement the fast direct method. The resulting algorithm is formulated in Algorithm 4.13 with an LCP-solving module that implements the iterative version. The implementation of the direct version is left to the reader (⟶ Exercise 4.12). Recall $g_{i\nu} := g(x_i, \tau_\nu)$ $(0 \leq i \leq m)$ and $g^{(\nu)} := (g_{1\nu}, \ldots, g_{m-1,\nu})^\mathrm{tr}$. The Figure 4.11 depicts a result of Algorithm 4.13 for Example 1.6. Here we obtain the contact point with value $S_\mathrm{f}(0) = 36.3$. Figure 4.13 shows the American put that corresponds to the call in Figure 4.9.

Algorithm 4.13 (prototype core algorithm)

Set up the function $g(x,\tau)$ listed in Problem 4.7.
Choose θ ($\theta = 1/2$ for Crank–Nicolson).
For PSOR: choose $1 \le \omega_{\mathrm{R}} < 2$ (for example, $\omega_{\mathrm{R}} = 1$),
fix an error bound ε (for example, $\varepsilon = 10^{-5}$).
Fix the discretization by choosing $x_{\min},\ x_{\max},\ m,\ \nu_{\max}$
(for example, $x_{\min} = -5,\ x_{\max} = 5,\ \nu_{\max} = m = 100$).
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 $\alpha := \lambda\theta$.

τ-loop: for $\nu = 0, 1, ..., \nu_{\max} - 1$:
$\tau_\nu := \nu\Delta\tau$
$b_i := w_i + \lambda(1-\theta)(w_{i+1} - 2w_i + w_{i-1})$ for $2 \le i \le m-2$
$b_1 := w_1 + \lambda(1-\theta)(w_2 - 2w_1 + g_{0\nu}) + \alpha g_{0,\nu+1}$
$b_{m-1} := w_{m-1} + \lambda(1-\theta)(g_{m\nu} - 2w_{m-1} + w_{m-2}) + \alpha g_{m,\nu+1}$
LCP solution, directly as in Exercise 4.12, or with PSOR:
| Set componentwise $v = \max(w, g^{(\nu+1)})$
| (v is the iteration vector of the projected SOR.)
| *PSOR-loop: for $k = 1, 2, ...$:*
| as long as $\|v^{\mathrm{new}} - v\|_2 > \epsilon$:
| *for $i = 1, 2, ..., m-1$:*
| $\rho := (b_i + \alpha(v_{i-1}^{\mathrm{new}} + v_{i+1}))/(1 + 2\alpha)$
| (with $v_0^{\mathrm{new}} = v_m = 0$)
| $v_i^{\mathrm{new}} = \max\{g_{i,\nu+1},\ v_i + \omega_{\mathrm{R}}(\rho - v_i)\}$
| $v := v^{\mathrm{new}}$ (after testing for convergence)
$w^{(\nu+1)} = w = v$

European options:

For completeness we mention that it is possible to calculate European options with Algorithm 4.13 after some modifications. In the iterative version, replacing the line

$v_i^{\mathrm{new}} = \max\{g_{i,\nu+1},\ v_i + \omega_{\mathrm{R}}(\rho - v_i)\}$

by the line

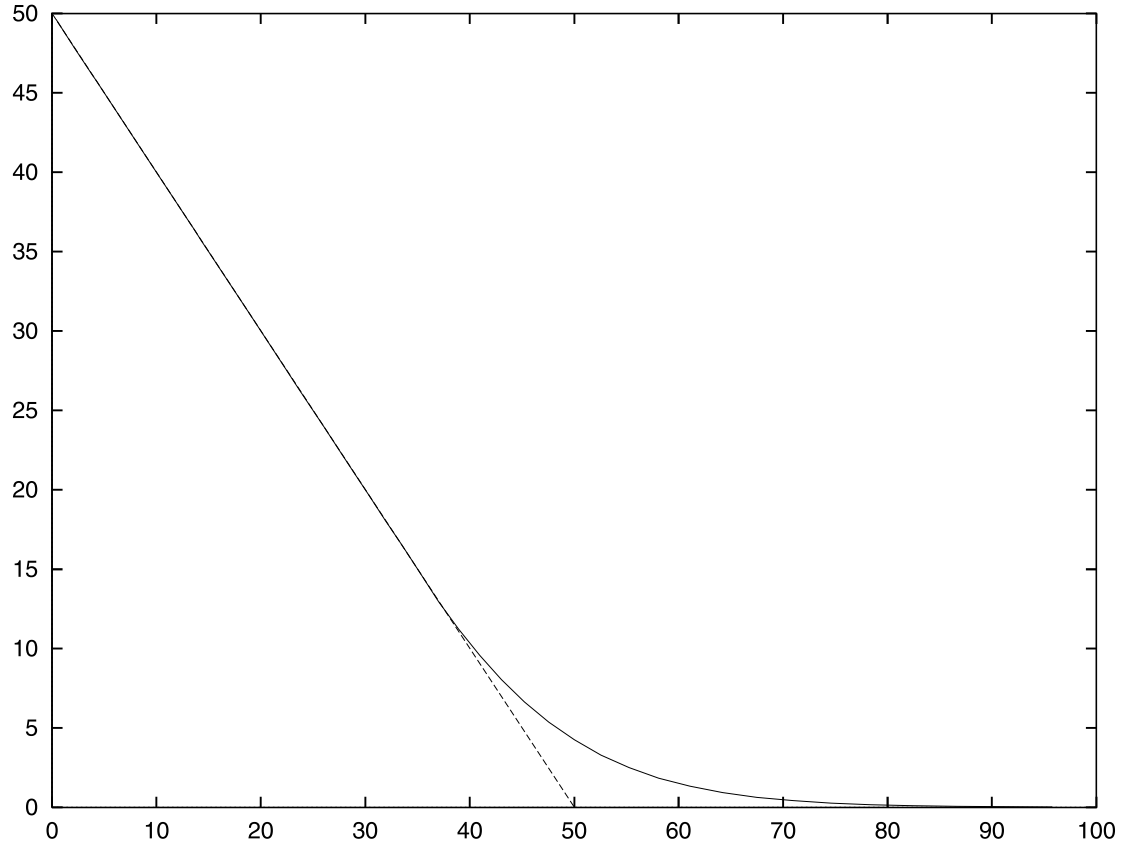

Fig. 4.11. (Example 1.6) American put, $K = 50$, $r = 0.1$, $\sigma = 0.4$, $T = \frac{5}{12}$. $V(S,0)$ (solid curve) and payoff $V(S,T)$ (dashed). Special value: $V(K,0) = 4.2842$

$$v_i^{\text{new}} = v_i + \omega_{\text{R}}(\rho - v_i)$$

recovers the standard SOR for solving $Aw = b$ (without $w \geq g$). If in addition the boundary conditions are adapted, then the program resulting from Algorithm 4.13 can be applied to European options. The same holds true for the direct method. And applying the analytic solution formula should be most economical, when the entire surface is not required. But for the purpose of testing Algorithm 4.13 it may be 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 4.13, 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 4.13 as another algorithm:

Algorithm 4.14 (American options)

Input: strike K, time to expiration T, spot price S_0, r, δ, σ
Perform the core Algorithm 4.13.
(The τ-loop ends at $\tau_{\text{end}} = \frac{1}{2}\sigma^2 T$.)
For $i = 1, \ldots, m-1$:
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 $S_{\text{f}}(0)$:
(in case PSOR was used)
Choose $\varepsilon^* = K \cdot 10^{-5}$ (for example)
For a put:
$i_{\text{f}} := \max\{i \ : \ |V(S_i, 0) + S_i - K| < \varepsilon^*\}$
$S_0 < S_{i_{\text{f}}}$: stopping region!
For a call:
$i_{\text{f}} := \min\{i \ : \ |K - S_i + V(S_i, 0)| < \varepsilon^*\}$
$S_0 > S_{i_{\text{f}}}$: stopping region!

In case the direct method was used, the index i_{f} is known from the algorithm. The Algorithm 4.14 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 Figure 4.8.

Modifications

The above Algorithm 4.13 (along with Algorithm 4.14) 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 lack in smoothness is illustrated by Figure 4.12. The turmoil at the corner is seen, but also the relatively rapid smoothing within a few time steps. In this context it may be advisable to start with a few fully implicit backward time steps ($\theta = 1$) before switching to Crank–Nicolson ($\theta = 1/2$), see [Ran84] and the Notes on Section 4.2. 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 Section 4.7.

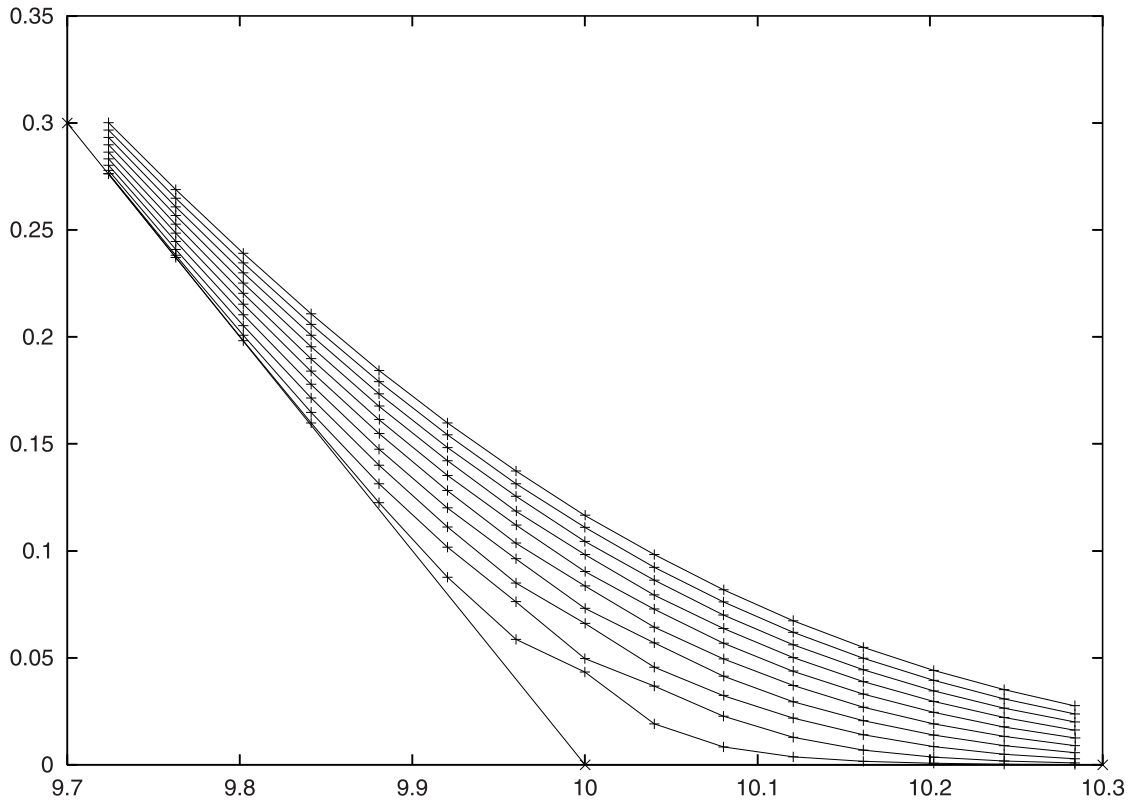

Fig. 4.12. Finite differences, Crank–Nicolson; American put with $r = 0.06$, $\sigma = 0.3$, $T = 1$, $K = 10$; $M = 1000$, $x_{\min} = -2$, $x_{\max} = 2$, $\Delta x = 1/250$, $\Delta t = 1/1000$, payoff and $V(S, t_\nu)$ for $t_\nu = 1 - \nu\Delta t$, $\nu = 1, \ldots, 10$.

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 Assumption 1.2, for example, will not exactly match the reality of a financial market. And the parameters of the equations (such as volatility σ) are unknown and must be estimated. Hence the equations of the model are only approximations of the "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 error between differential quotients and difference quotients. For example, a Crank–Nicolson discretization is of the