

Finite Difference Methods

In Chapter 1 we described how the pricing of a derivative security typically requires either the solution of a parabolic partial differential equation (PDE) or the evaluation of an expectation of a random variable. In realistic applications, both of these price formulations often do not allow for closed-form solution, in which case we must resort to either analytical approximations or, more generally, numerical techniques. In the next two chapters we will describe a number of numerical algorithms useful in derivatives pricing. Analytical approximations will receive ample treatment later in this book, in the context of specific problems.

Our treatment of numerical methods is broken into two main subjects. In this chapter, we cover finite difference solutions of PDEs; and in Chapter 3 we turn to Monte Carlo evaluation of expectations. Many excellent specialist books exist on both topics, including Mitchell and Griffiths [1980], Tavella and Randall [2000], and Glasserman [2004]; our treatment only surveys the most important concepts, as required for our needs in this book. We do provide, however, a number of schemes rarely described in detail in the finance literature and also supplement our analysis with a number of “tricks of the trade”, particularly in the application of finite difference grids.

The analysis of numerical PDE solutions in this chapter is arranged in two blocks. First, in Sections 2.1–2.8 we study the basic mechanics of the finite difference grid method for one-dimensional PDEs. Subsequently, Sections 2.9–2.12 then apply operator splitting techniques to extend the finite difference method to PDE of dimensions two and higher. The analysis culminates with a presentation of *ADI schemes* for multi-dimensional PDEs with mixed partial derivatives.

2.1 1-Dimensional PDEs: Problem Formulation

Initially, we will consider the numerical solution of the general one-dimensional terminal value PDE problem

$$\frac{\partial V}{\partial t} + \mathcal{L}V = 0, \quad (2.1)$$

where \mathcal{L} is the operator

$$\mathcal{L} = \mu(t, x) \frac{\partial}{\partial x} + \frac{1}{2} \sigma(t, x)^2 \frac{\partial^2}{\partial x^2} - r(t, x),$$

and where $V = V(t, x)$ satisfies a terminal condition $V(T, x) = g(x)$. We recognize the PDE as being an extension of the Black-Scholes PDE (1.47) to general time- and state-dependent drift (μ), volatility (σ), and interest rate (r). Underneath the PDE lies a physical model where a state variable process $x(\cdot)$ follows an SDE of the form

$$dx(t) = \mu(t, x(t)) dt + \sigma(t, x(t)) dW(t) \quad (2.2)$$

where $W(t)$ is a Brownian motion in the risk-neutral probability measure Q . Let the range of values attainable by $x(t)$ on $t \in [0, T]$ be denoted $\mathcal{B} \subseteq \mathbb{R}$, and assume that the functions $\mu, \sigma, r : [0, T] \times \mathcal{B} \rightarrow \mathbb{R}$ are sufficiently regular to make (2.1) and (2.2) meaningful (see Chapter 1).

The terminal value problem above is, as discussed earlier, a *Cauchy problem* to be solved for $V(t, x)$ on $(t, x) \in [0, T] \times \mathcal{B}$. In many cases of practical interest, further boundary conditions are applied in the spatial (x) domain. If such boundary conditions are expressed directly in terms of V (rather than its derivatives) we have a *Dirichlet boundary problem*. For instance, a so-called *up-and-out barrier option* will pay out $g(x(T))$ at time T if and only if $x(t)$ stays strictly below a contractually specified barrier level H at all times $t \leq T$. If, on the other hand, $x(t)$ touches H at any time during the life of the contract, it will expire worthless (or “knock out”). In this case, the PDE is only to be solved on $(t, x) \in [0, T] \times (\mathcal{B} \cap (-\infty, H))$ and is subject to the Dirichlet boundary condition

$$V(t, H) = 0, \quad t \in [0, T],$$

which expresses that the option has no value for $x \geq H$. We note that it is not uncommon to encounter options where the spatial domain boundaries are functions of time, a situation we shall deal with in Section 2.7.1. Also, as we shall see shortly, sometimes boundary conditions are conveniently expressed in terms of derivatives of V .

For numerical solution of the PDE (2.1), we often need to assume that the domain of the state variable x is finite, even in situations where (2.1) is supposed to hold for an infinite domain. Suitable truncation of the domain can often be done probabilistically, based on a confidence interval for $x(T)$. To illustrate the procedure, consider the Black-Scholes PDE (1.47) applied to a call option with strike K . A common first step is to use the transformation $x = \ln S$, such that the PDE has constant coefficients,

$$\frac{\partial V}{\partial t} + \left(r - \frac{1}{2} \sigma^2 \right) \frac{\partial V}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 V}{\partial x^2} - rV = 0, \quad (2.3)$$

with terminal value (for a call option) $V(T, x) = (e^x - K)^+$. The domain of x is here the entire real line, $\mathcal{B} = \mathbb{R}$. We know (from (1.39)) that

$$x(T) = x(0) + \left(r - \frac{1}{2}\sigma^2 \right) T + \sigma (W(T) - W(0)), \quad (2.4)$$

which is a Gaussian random variable with mean $\bar{x} = x(0) + (r - \frac{1}{2}\sigma^2)T$ and variance $\sigma^2 T$. Consider now replacing the domain $(-\infty, \infty)$ with the finite interval $[\bar{x} - \alpha\sigma\sqrt{T}, \bar{x} + \alpha\sigma\sqrt{T}]$ for some positive constant α . The likelihood of $x(T)$ falling outside of this interval is easily seen to be $2\Phi(-\alpha)$ (where, as always, $\Phi(z)$ is the standard Gaussian cumulative distribution function). If, say, we set α to 4, $2\Phi(-4) = 6.3 \times 10^{-5}$, which is an insignificant probability for most applications. Larger (smaller) values of α will make the truncation error smaller (larger) and will ultimately require more (less) effort in a numerical scheme. We recommend values of α somewhere between 3 and 5 for most applications. For the Black-Scholes case, a rigorous estimate of the error imposed by domain truncation is given in Kangro and Nicolaides [2000].

In many cases of practical interest, it is not possible to write down an exact confidence interval for $x(T)$. In such cases, one instead may use an approximate confidence interval, found by, for instance, using “average” values for $\mu(t, x)$ and $\sigma(t, x)$. High precision in these estimates is typically not needed.

2.2 Finite Difference Discretization

In order to solve the PDE (2.1) numerically, we now wish to discretize it on the rectangular domain $(t, x) \in [0, T] \times [\underline{M}, \overline{M}]$, where \overline{M} and \underline{M} are finite constants, possibly found by a truncation procedure such as the one outlined above. We first introduce two equidistant¹ grids $\{t_i\}_{i=0}^n$ and $\{x_j\}_{j=0}^{m+1}$ where $t_i = iT/n \triangleq i\Delta_t$, $i = 0, 1, \dots, n$, and $x_j = \underline{M} + j(\overline{M} - \underline{M})/(m+1) \triangleq \underline{M} + j\Delta_x$, $j = 0, 1, \dots, m+1$. The terminal value $V(T, x) = g(x)$ is imposed at $t_n = T$, and spatial boundary conditions are imposed at x_0 and x_{m+1} .

2.2.1 Discretization in x -Direction. Dirichlet Boundary Conditions

We first focus on the spatial operator \mathcal{L} and restrict x to take values in the interior of the spatial grid $x \in \{x_j\}_{j=1}^m$. Consider replacing the first- and second-order partial derivatives with first- and second-order difference operators:

¹Non-equidistant grids are often required in practice and will be covered in Section 2.4.

$$\delta_x V(t, x_j) \triangleq \frac{V(t, x_{j+1}) - V(t, x_{j-1})}{2\Delta_x}, \quad (2.5)$$

$$\delta_{xx} V(t, x_j) \triangleq \frac{V(t, x_{j+1}) + V(t, x_{j-1}) - 2V(t, x_j)}{\Delta_x^2}. \quad (2.6)$$

These operators are accurate to second order. Formally²,

Lemma 2.2.1.

$$\delta_x V(t, x_j) = \frac{\partial V(t, x_j)}{\partial x} + O(\Delta_x^2),$$

$$\delta_{xx} V(t, x_j) = \frac{\partial^2 V(t, x_j)}{\partial x^2} + O(\Delta_x^2).$$

Proof. A Taylor expansion of $V(t, x)$ around the point $x = x_j$ gives

$$\begin{aligned} V(t, x_{j+1}) &= V(t, x_j) + \Delta_x \frac{\partial V(t, x_j)}{\partial x} \\ &\quad + \frac{1}{2} \Delta_x^2 \frac{\partial^2 V(t, x_j)}{\partial x^2} + \frac{1}{6} \Delta_x^3 \frac{\partial^3 V(t, x_j)}{\partial x^3} + O(\Delta_x^4), \end{aligned}$$

and

$$\begin{aligned} V(t, x_{j-1}) &= V(t, x_j) - \Delta_x \frac{\partial V(t, x_j)}{\partial x} \\ &\quad + \frac{1}{2} \Delta_x^2 \frac{\partial^2 V(t, x_j)}{\partial x^2} - \frac{1}{6} \Delta_x^3 \frac{\partial^3 V(t, x_j)}{\partial x^3} + O(\Delta_x^4). \end{aligned}$$

Insertion of these expressions into (2.5) and (2.6) gives the desired result. \square

In other words, if we introduce the discrete operator

$$\widehat{\mathcal{L}} = \mu(t, x)\delta_x + \frac{1}{2}\sigma(t, x)^2\delta_{xx} - r(t, x),$$

we have, for $x \in \{x_j\}_{j=1}^m$,

$$\mathcal{L}V(t, x) = \widehat{\mathcal{L}}V(t, x) + O(\Delta_x^2).$$

With attention restricted to values on the grid $\{x_j\}_{j=1}^m$, we can view $\widehat{\mathcal{L}}$ as a matrix, once we specify the side boundary conditions at x_0 and x_{m+1} . For the Dirichlet case, assume for instance that

$$V(x_0, t) = \underline{f}(t, x_0), \quad V(x_{m+1}, t) = \bar{f}(t, x_{m+1}),$$

²Recall that a function $f(h)$ is of order $O(e(h))$ if $|f(h)|/|e(h)|$ is bounded from above by a positive constant in the limit $h \rightarrow 0$.

for given functions $\underline{f}, \bar{f} : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$. With³ $\mathbf{V}(t) \triangleq (V(t, x_1), \dots, V(t, x_m))^\top$ and, for $j = 1, \dots, m$,

$$c_j(t) \triangleq -\sigma(t, x_j)^2 \Delta_x^{-2} - r(t, x_j), \quad (2.7)$$

$$u_j(t) \triangleq \frac{1}{2}\mu(t, x_j)\Delta_x^{-1} + \frac{1}{2}\sigma(t, x_j)^2\Delta_x^{-2}, \quad (2.8)$$

$$l_j(t) \triangleq -\frac{1}{2}\mu(t, x_j)\Delta_x^{-1} + \frac{1}{2}\sigma(t, x_j)^2\Delta_x^{-2}, \quad (2.9)$$

we can write

$$\hat{\mathcal{L}}\mathbf{V}(t) = \mathbf{A}(t)\mathbf{V}(t) + \boldsymbol{\Omega}(t), \quad (2.10)$$

where \mathbf{A} is a *tri-diagonal matrix*

$$\mathbf{A}(t) = \begin{pmatrix} c_1(t) & u_1(t) & 0 & 0 & 0 & \dots & 0 \\ l_2(t) & c_2(t) & u_2(t) & 0 & 0 & \dots & 0 \\ 0 & l_3(t) & c_3(t) & u_3(t) & 0 & \dots & 0 \\ 0 & 0 & l_4(t) & c_4(t) & u_4(t) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & l_{m-1}(t) & c_{m-1}(t) & u_{m-1}(t) \\ 0 & 0 & 0 & 0 & 0 & l_m(t) & c_m(t) \end{pmatrix} \quad (2.11)$$

and $\boldsymbol{\Omega}(t)$ is a vector containing boundary values

$$\boldsymbol{\Omega}(t) = \begin{pmatrix} l_1(t)\underline{f}(t, x_0) \\ 0 \\ \vdots \\ 0 \\ u_m(t)\bar{f}(t, x_{m+1}) \end{pmatrix}.$$

As discussed earlier, sometimes one or both of the functions \bar{f} and \underline{f} are explicitly imposed as part of the option specification (as is the case for a knock-out options). In other cases, asymptotics may be necessary to establish these functions. For instance, for the case of a simple call option on a stock paying no dividends, we can set

$$\begin{aligned} \bar{f}(t, x) &= e^x - Ke^{-r(T-t)}, \\ \underline{f}(t, x) &= 0, \end{aligned}$$

where we, as before, have set $x = \ln S$ (S being the stock price) and assumed that the strike K is positive. The result for \underline{f} is obvious; the result for \bar{f} follows from the fact that a deep in-the-money call option will almost certainly pay at maturity the stock (the present value of which is just $S = e^x$) minus the strike (the present value of which is $Ke^{-r(T-t)}$).

³For clarity, this chapter uses boldface type for all vectors and matrices.

2.2.2 Other Boundary Conditions

Deriving asymptotic Dirichlet conditions can be quite involved for complicated option payouts and is often inconvenient in implementations. Rather than having to perform an asymptotic analysis for each and every type of option payout, it would be preferable to have a general-purpose mechanism for specifying the boundary condition. One common idea involves making assumptions on the form of the functional dependency between V and x at the grid boundaries, often from specification of relationships between spatial derivatives. For instance, if we impose the condition that the second derivative of V is zero at the upper boundary (x_{m+1}) — that is, V is a linear function of x — we can write (effectively using a downward discretization of the second derivative)

$$\frac{V(t, x_{m+1}) + V(t, x_{m-1}) - 2V(t, x_m)}{\Delta_x^2} = 0 \\ \Rightarrow V(t, x_{m+1}) = 2V(t, x_m) - V(t, x_{m-1}).$$

A similar assumption at the lower spatial boundary yields

$$V(t, x_0) = 2V(t, x_1) - V(t, x_2).$$

For PDEs discretized in the logarithm of some asset, it may be more natural to assume that $V(t, x) \propto e^x$ at the boundaries; equivalently, we can assume that $\partial V / \partial x = \partial^2 V / \partial x^2$ at the boundary. When discretized in downward fashion at the upper boundary (x_{m+1}), this implies that

$$\frac{V(t, x_{m+1}) - V(t, x_m)}{\Delta_x} = \frac{V(t, x_{m+1}) + V(t, x_{m-1}) - 2V(t, x_m)}{\Delta_x^2}$$

or (assuming that $\Delta_x \neq 1$)

$$V(t, x_{m+1}) = V(t, x_{m-1}) \frac{1}{\Delta_x - 1} + V(t, x_m) \frac{\Delta_x - 2}{\Delta_x - 1}.$$

Similarly,

$$V(t, x_0) = V(t, x_1) \frac{2 + \Delta_x}{1 + \Delta_x} - V(t, x_2) \frac{1}{\Delta_x + 1}.$$

Common for both methods above — and for the Dirichlet specification discussed earlier — is that they give rise to boundary specifications through simple linear systems of the general form

$$V(t, x_{m+1}) = k_m(t)V(t, x_m) + k_{m-1}(t)V(t, x_{m-1}) + \bar{f}(t, x_{m+1}), \quad (2.12)$$

$$V(t, x_0) = k_1(t)V(t, x_1) + k_2(t)V(t, x_2) + \underline{f}(t, x_0). \quad (2.13)$$

This boundary specification can be captured in the matrix system (2.10) by simply rewriting a few components of $\mathbf{A}(t)$; specifically, we must set

$$\begin{aligned}
c_m(t) &= -\sigma(t, x_m)^2 \Delta_x^{-2} - r(t, x_m) + k_m(t) u_m(t), \\
l_m(t) &= -\frac{1}{2} \mu(t, x_m) \Delta_x^{-1} + \frac{1}{2} \sigma(t, x_m)^2 \Delta_x^{-2} + k_{m-1}(t) u_m(t), \\
c_1(t) &= -\sigma(t, x_j)^2 \Delta_x^{-2} - r(t, x_j) + k_1(t) l_1(t), \\
u_1(t) &= \frac{1}{2} \mu(t, x_1) \Delta_x^{-1} + \frac{1}{2} \sigma(t, x_1)^2 \Delta_x^{-2} + k_2(t) l_1(t).
\end{aligned}$$

All other components of \mathbf{A} remain as in (2.11); note that \mathbf{A} remains tri-diagonal.

An alternative approach to specification of boundary conditions in the x -domain involves using the PDE itself to determine the boundary conditions, through replacement of all central difference operators with one-sided differences at the boundaries. Section 10.1.5.2 contains a detailed example of this idea; ultimately, this approach leads to boundary conditions that can also be written in the form (2.12)–(2.13).

2.2.3 Time-Discretization

To simplify notation, assume for now that $\Omega(t) = 0$ for all t , as will be the case if, say, we use the linear or linear-exponential boundary conditions outlined earlier. On the spatial grid, our original PDE can be written

$$\frac{\partial \mathbf{V}(t)}{\partial t} = -\mathbf{A}(t)\mathbf{V}(t) + O(\Delta_x^2)$$

which, ignoring the error term⁴, defines a system of coupled ordinary differential equations (ODEs).

A number of methods are available for the numerical solution of coupled ODEs; see, e.g., Press et al. [1992]. We here only consider basic two-level time-stepping schemes, where grid computations at time t_i involve only PDE values at times t_i and t_{i+1} . Focusing the attention on a particular bucket $[t_i, t_{i+1}]$, the choice for the finite difference approximation of $\partial V/\partial t$ is obvious:

$$\frac{\partial \mathbf{V}}{\partial t} \approx \frac{\mathbf{V}(t_{i+1}) - \mathbf{V}(t_i)}{\Delta_t}.$$

Not so obvious, however, is to which time in the interval $[t_i, t_{i+1}]$ we should associate this derivative. To be general, consider picking a time $t_i^{i+1}(\theta) \in [t_i, t_{i+1}]$, given by

$$t_i^{i+1}(\theta) = (1 - \theta)t_{i+1} + \theta t_i, \quad (2.14)$$

where $\theta \in [0, 1]$ is a parameter. We then write

$$\frac{\partial \mathbf{V}(t_i^{i+1}(\theta))}{\partial t} \approx \frac{\mathbf{V}(t_{i+1}) - \mathbf{V}(t_i)}{\Delta_t}.$$

⁴Note that the error term $O(\Delta_x^2)$ is here to be interpreted as an m -dimensional vector. We will use such short-hand notation throughout this chapter.

By a Taylor expansion, it is easy to see that this expression is first-order accurate in the time step when $\theta \neq \frac{1}{2}$, and second-order accurate when $\theta = \frac{1}{2}$. Written compactly,

$$\frac{\partial \mathbf{V}(t_i^{i+1}(\theta))}{\partial t} = \frac{\mathbf{V}(t_{i+1}) - \mathbf{V}(t_i)}{\Delta_t} + 1_{\{\theta \neq \frac{1}{2}\}} O(\Delta_t) + O(\Delta_t^2). \quad (2.15)$$

This result on the convergence order is intuitive since only in the case $\theta = \frac{1}{2}$ is the difference coefficient precisely central; for all other cases, the difference coefficient is either predominantly backward in time or predominantly forward in time.

The time-discretization technique introduced above is known as a *theta scheme*. The special cases of $\theta = 1$, $\theta = 0$, and $\theta = \frac{1}{2}$ are known as the *fully implicit scheme*, the *fully explicit scheme*, and the *Crank-Nicolson scheme*, respectively. In light of the convergence result (2.15), one may wonder why anything other than the Crank-Nicolson scheme is ever used. The CN method is, indeed, often the method of choice, but there are situations where a straight application of the Crank-Nicolson scheme can lead to oscillations in the numerical solution or its spatial derivatives. Judicial application of the fully implicit method can often alleviate these problems, as we shall discuss later. The fully explicit method should never be used due to poor convergence and stability properties (see Section 2.3), but has nevertheless managed to survive in a surprisingly large number of finance texts and papers.

2.2.4 Finite Difference Scheme

We now proceed to combine the discretizations (2.10) and (2.15) into a complete finite difference scheme. First, we expand

$$\begin{aligned} \mathbf{A}(t_i^{i+1}(\theta)) \mathbf{V}(t_i^{i+1}(\theta)) &= \theta \mathbf{A}(t_i^{i+1}(\theta)) \mathbf{V}(t_i) \\ &\quad + (1 - \theta) \mathbf{A}(t_i^{i+1}(\theta)) \mathbf{V}(t_{i+1}) + 1_{\{\theta \neq \frac{1}{2}\}} O(\Delta_t) + O(\Delta_t^2), \end{aligned}$$

such that our PDE can be represented as

$$\begin{aligned} \frac{\mathbf{V}(t_{i+1}) - \mathbf{V}(t_i)}{\Delta_t} + 1_{\{\theta \neq \frac{1}{2}\}} O(\Delta_t) + O(\Delta_t^2) \\ &= -\mathbf{A}(t_i^{i+1}(\theta)) \mathbf{V}(t_i^{i+1}(\theta)) + O(\Delta_x^2) \\ &= -\theta \mathbf{A}(t_i^{i+1}(\theta)) \mathbf{V}(t_i) - (1 - \theta) \mathbf{A}(t_i^{i+1}(\theta)) \mathbf{V}(t_{i+1}) \\ &\quad + 1_{\{\theta \neq \frac{1}{2}\}} O(\Delta_t) + O(\Delta_t^2) + O(\Delta_x^2). \end{aligned}$$

Multiplying through with Δ_t gives rise to the complete finite difference representation of the PDE solution at times t_i and t_{i+1} :

Proposition 2.2.2. *On the grid $\{x_j\}_{j=1}^m$, the solution to (2.1) at times t_i and t_{i+1} is characterized by*

$$(\mathbf{I} - \theta \Delta_t \mathbf{A}(t_i^{i+1}(\theta))) \mathbf{V}(t_i) = (\mathbf{I} + (1 - \theta) \Delta_t \mathbf{A}(t_i^{i+1}(\theta))) \mathbf{V}(t_{i+1}) + e_i^{i+1}, \quad (2.16)$$

where \mathbf{I} is the $m \times m$ identity matrix, and e_i^{i+1} is an error term

$$e_i^{i+1} = \Delta_t O(\Delta_x^2) + 1_{\{\theta \neq \frac{1}{2}\}} O(\Delta_t^2) + O(\Delta_t^3). \quad (2.17)$$

Let $\widehat{\mathbf{V}}(t_i, x_j)$ denote the approximation to the true solution $V(t_i, x_j)$ obtained by using (2.16) without the error term. Defining

$$\widehat{\mathbf{V}}(t) = (\widehat{V}(t, x_1), \dots, \widehat{V}(t, x_m))^\top,$$

we have

$$(\mathbf{I} - \theta \Delta_t \mathbf{A}(t_i^{i+1}(\theta))) \widehat{\mathbf{V}}(t_i) = (\mathbf{I} + (1 - \theta) \Delta_t \mathbf{A}(t_i^{i+1}(\theta))) \widehat{\mathbf{V}}(t_{i+1}). \quad (2.18)$$

For a known value of $\widehat{\mathbf{V}}(t_{i+1})$, (2.18) defines a simple linear system of equations that can be solved for $\widehat{\mathbf{V}}(t_i)$ by standard methods. Simplifying matters is the fact that the matrix $(\mathbf{I} - \theta \Delta_t \mathbf{A}(t_i^{i+1}(\theta)))$ is tri-diagonal, allowing us to solve (2.18) in only $O(m)$ operations; see Press et al. [1992] for an algorithm⁵.

Starting from the prescribed terminal condition $V(t_n, x_j) = g(x_j)$, $j = 1, \dots, m$, we can now use (2.18) to iteratively step backward in time until we ultimately recover $\widehat{\mathbf{V}}(0)$. This procedure is known as *backward induction*.

Proposition 2.2.3. *The theta scheme (2.18) recovers $\widehat{\mathbf{V}}(0)$ in $O(mn)$ operations. If the scheme converges, the error on $\widehat{\mathbf{V}}(0)$ compared to the exact solution $\mathbf{V}(0)$ is of order*

$$O(\Delta_x^2) + 1_{\{\theta \neq \frac{1}{2}\}} O(\Delta_t) + O(\Delta_t^2).$$

Proof. The backward induction algorithm requires the solution of n tri-diagonal systems, one per time step, for a total computational cost of $O(mn)$. The local truncation error on $\widehat{\mathbf{V}}(t_i)$ is e_i^{i+1} , making the global truncation error after n time steps of order ne_i^{i+1} . Combining (2.17) with the fact that $n = T/\Delta_t = O(\Delta_t^{-1})$ gives the order result listed in the proposition. \square

⁵The special case of an explicit scheme ($\theta = 0$) provides us with a direct expression for $V(t_i, x_j)$ in terms of $V(t_{i+1}, x_{j-1})$, $V(t_{i+1}, x_j)$, and $V(t_{i+1}, x_{j+1})$, a scheme that is easily visualized as a “trinomial tree”. The intuitive nature of the explicit scheme coupled with the fact that no matrix equation must be solved may explain the popularity of this scheme in the finance literature, despite its poor numerical qualities (see Section 2.3). We stress that the workload of the explicit scheme is still $O(m)$ per time step, as is the case for all theta schemes.

It follows from Proposition 2.2.3 that the Crank-Nicolson scheme is second-order convergent in the time step, and all other theta schemes are first-order convergent in the time step. All theta-schemes are second-order convergent in the spatial step Δ_x .

In deriving (2.18), we assumed earlier that the boundary vector was zero, $\Omega(t) = 0$. Including a non-zero boundary vector into the scheme is, however, straightforward and results in a time-stepping scheme of the form

$$\begin{aligned} (\mathbf{I} - \theta \Delta_t \mathbf{A}(t_i^{i+1}(\theta))) \hat{\mathbf{V}}(t_i) &= (\mathbf{I} + (1 - \theta) \Delta_t \mathbf{A}(t_i^{i+1}(\theta))) \hat{\mathbf{V}}(t_{i+1}) \\ &\quad + (1 - \theta) \Omega(t_{i+1}) + \theta \Omega(t_i). \end{aligned} \quad (2.19)$$

Again, this system is easily solved for $\hat{\mathbf{V}}(t_i)$ by a standard tri-diagonal equation solver.

As a final point, we stress that the finite difference scheme above ultimately yields a full vector of values $\hat{\mathbf{V}}(0)$ at time 0, with one element per value of x_j , $j = 1, \dots, m$. In general, we are mainly interested in $V(0, x(0))$, where $x(0)$ is the known value of x at time 0. There is no need to include $x(0)$ in the grid, as we can simply employ an interpolator (e.g., a cubic spline) on this vector $\hat{\mathbf{V}}(0)$ to compute $V(0, x(0))$. Clearly, such an interpolator should be at least second-order accurate to avoid interfering with the overall $O(\Delta_x^2)$ convergence of the finite difference scheme. Assuming the interpolator is sufficiently smooth, we can also use it to compute various partial derivatives with respect to x that we may be interested in. Alternatively, these can be computed by the same type of finite difference coefficients discussed in Section 2.2.1. The derivative $\partial V(0, x(0))/\partial t$ — the *time decay* — can be picked up from the grid in the same fashion.

Remark 2.2.4. The scheme (2.18) may, without affecting convergence order, be replaced with

$$(\mathbf{I} - \theta \Delta_t \mathbf{A}(t_i)) \hat{\mathbf{V}}(t_i) = (\mathbf{I} + (1 - \theta) \Delta_t \mathbf{A}(t_{i+1})) \hat{\mathbf{V}}(t_{i+1}).$$

2.3 Stability

2.3.1 Matrix Methods

Ignoring the contributions from boundary conditions, the finite difference scheme developed in the previous section can be rewritten

$$\hat{\mathbf{V}}(t_i) = \mathbf{B}_i^{i+1} \hat{\mathbf{V}}(t_{i+1}), \quad (2.20)$$

where

$$\mathbf{B}_i^{i+1} \triangleq (\mathbf{I} - \theta \Delta_t \mathbf{A}(t_i^{i+1}(\theta)))^{-1} (\mathbf{I} + (1 - \theta) \Delta_t \mathbf{A}(t_i^{i+1}(\theta))).$$

That is, for any $0 \leq k < n$,

$$\widehat{\mathbf{V}}(t_k) = \mathbf{B}_k^n \widehat{\mathbf{V}}(t_n), \quad \mathbf{B}_k^n \triangleq \mathbf{B}_k^{k+1} \mathbf{B}_{k+1}^{k+2} \dots \mathbf{B}_{n-1}^n.$$

We say that the scheme is *stable* if $|\widehat{\mathbf{V}}(t_k)|$ is bounded for all $0 \leq k < n$. Assuming $|\widehat{\mathbf{V}}(T)| < \infty$, a necessary and sufficient condition for stability is that there exists a constant K such that for all $0 \leq k < n$

$$|\mathbf{B}_k^n| \leq K, \quad (2.21)$$

where $|\cdot|$ is any matrix norm, e.g. the spectral norm or the infinity norm⁶. See Mitchell and Griffiths [1980] for further details.

2.3.2 Von Neumann Analysis

For simple problems with time- and space-independent coefficients, it may be possible to establish the spectral norm of \mathbf{B}_k^n by direct methods (see e.g. Mitchell and Griffiths [1980], Kraaijevanger et al. [1987], Lenferink and Spijker [1991], Spijker and Straetemans [1997]), but generally the stability criterion (2.21) is difficult to evaluate. While certain somewhat simpler matrix-based methods exist to establish necessary conditions for stability (again, see Mitchell and Griffiths [1980]), we shall here only consider a “local” method, known as the *von Neumann method*. In principle, the von Neumann method only holds for finite difference schemes where the underlying PDE has constant coefficients, but there is much numerical evidence to support wider application⁷. The von Neumann method does not directly consider the effect of boundary conditions on stability, but (for constant coefficient problems) provides a necessary condition for stability irrespective of the type of boundary condition.

The basis for the von Neumann analysis is the observation that a real function sampled on a finite number of points is uniquely defined by a complex Fourier series. For our PDE solution sampled on the spatial grid, the precise result is

$$V(t_k, x_j) = \sum_l H_l(t_k) e^{-i\omega_l j \Delta_x},$$

where $H_l(t_k)$ and ω_l are the amplification factor (discrete Fourier transform) and wave number for the l -th mode, respectively. Notice that i here denotes

⁶The spectral norm of a matrix \mathbf{C} is defined as the largest absolute eigenvalue of $(\mathbf{C}^\top \mathbf{C})^{1/2}$. The infinity norm is defined as $\max_i \sum_j |C_{i,j}|$.

⁷In the application to PDEs with non-constant coefficients, it may help to think of the von Neumann analysis as being applied to the PDE locally with “frozen” coefficients, followed by an examination of the worst case among all frozen coefficients.

the imaginary unit, $i^2 = -1$, with k (momentarily) having taken the role of the time index in the finite difference grid. For the constant coefficient case, a key fact for our PDE problem is that

$$H_l(t_k) = H_l(t_{k+1})\xi_l^{-1},$$

where ξ_l is a mode-specific *amplification factor* independent of time. To determine how a solution is propagated back through the finite difference grid, it thus suffices to consider a test function of the form

$$v(t_k, x_j) = \xi(\omega)^{n-k} e^{i\omega j \Delta_x}. \quad (2.22)$$

According to the Von Neumann criterion, stability of (2.20) requires that the *modulus of the amplification factor* $\xi(\omega)$ is less or equal to one, independent of the wave number:

$$\forall \omega : |\xi(\omega)| \leq 1. \quad (2.23)$$

This criterion is natural and merely expresses that all eigenmodes should be damped, and not exponentially amplified, by the finite difference scheme.

Turning to our system (2.20), assume for simplicity that $r(t, x) = 0$. A positive interest rate (we will nearly always have $r(t, x) > 0$) introduces some extra dampening through discounting effects and will, if anything, lead to better stability properties than the case of zero interest rates. Writing $v(t_k, x_j) = v_{k,j}$, $\sigma(t_k^{k+1}(\theta), x_j) = \sigma_{k,j}$, and $\mu(t_k^{k+1}(\theta), x_j) = \mu_{k,j}$, the von Neumann analysis gives the following result:

Proposition 2.3.1. Define $\alpha = \Delta_t / (\Delta_x)^2$. For (2.20) with $r(t, x) = 0$, the von Neumann stability criterion is

$$1 \geq \theta \geq \frac{1}{2} - \frac{1}{\alpha} \left(\frac{\sigma_{k,j}^2}{\sigma_{k,j}^4 + \mu_{k,j}^2 \Delta_x^2 + |\mu_{k,j}^2 \Delta_x^2 - \sigma_{k,j}^4|} \right), \quad (2.24)$$

to hold for all $k = 0, 1, \dots, n-1$, $j = 1, 2, \dots, m$.

Proof. Define $\varsigma_{k,j}^\pm = \sigma_{k,j}^2 \pm \Delta_x \mu_{k,j}$. A local application of (2.20) gives

$$\begin{aligned} v_{k,j-1} \left(-\frac{\alpha\theta}{2} \varsigma_{k,j}^- \right) + v_{k,j} (1 + \alpha\theta\sigma_{k,j}^2) + v_{k,j+1} \left(-\frac{\alpha\theta}{2} \varsigma_{k,j}^+ \right) = \\ v_{k+1,j} \left(\frac{\alpha(1-\theta)}{2} \varsigma_{k,j}^- \right) + v_{k+1,j} (1 - \alpha(1-\theta)\sigma_{k,j}^2) + v_{k+1,j+1} \left(\frac{\alpha(1-\theta)}{2} \varsigma_{k,j}^+ \right) \end{aligned}$$

with α defined above. Inserting (2.22) and rearranging (using Euler's formulas for sin and cos) yields

$$\xi(\omega) = \frac{1 - (1-\theta)\alpha\sigma_{k,j}^2(1 - \cos \omega \Delta_x) + i(1-\theta)\alpha\Delta_x \mu_{k,j} \sin \omega \Delta_x}{1 + \theta\alpha\sigma_{k,j}^2(1 - \cos \omega \Delta_x) - i\theta\alpha\Delta_x \mu_{k,j} \sin \omega \Delta_x}.$$

Note that ξ is a function of k and j , due to the non-constant PDE parameters. As discussed earlier (see also Mitchell and Griffiths [1980]), we expect the system to be stable if the criterion (2.23) holds for all k and j in the grid. Computing the modulus of ξ and requiring that it does not exceed one leads, after straightforward manipulations, to the stability criterion

$$\forall \omega : 2\alpha\sigma_{k,j}^2 + (2\theta - 1)\alpha^2 [\sigma_{k,j}^4 + \mu_{k,j}^2 \Delta_x^2 + \cos \omega \Delta_x (\mu_{k,j}^2 \Delta_x^2 - \sigma_{k,j}^4)] \geq 0.$$

As $\cos \omega \Delta_x \in [-1, 1]$, this expression can be simplified to (2.24). \square

From (2.24) we can immediately conclude that the finite difference scheme is always stable if $\frac{1}{2} \leq \theta \leq 1$, irrespective of the magnitudes of Δ_x and Δ_t . For $\frac{1}{2} \leq \theta \leq 1$, we therefore say that the theta scheme is *absolutely stable*, or simply *A-stable*. Both the fully implicit ($\theta = 1$) and the Crank-Nicolson ($\theta = \frac{1}{2}$) finite difference schemes are thus *A-stable*. For the explicit scheme ($\theta = 0$), however, stability is *conditional*, requiring

$$\frac{2}{\alpha}\sigma_{k,j}^2 \geq \sigma_{k,j}^4 + \mu_{k,j}^2 \Delta_x^2 + |\mu_{k,j}^2 \Delta_x^2 - \sigma_{k,j}^4|.$$

For small drifts, this expression amounts to the restriction $\sigma_{k,j}^2 \leq \Delta_x^2/\Delta_t$ which can be quite onerous, often requiring the (laborious) use of thousands of time steps in the finite difference grid. We shall not consider fully explicit methods any further in this book.

Returning to the case $\frac{1}{2} \leq \theta \leq 1$, let us introduce a stronger definition of stability. A time-stepping method is said to be *strongly A-stable* if the modulus of the amplification factor ξ is strictly below 1 for any value of the time step, including the limit⁸ $\Delta_t \rightarrow \infty$. From (2.24), we see that if $\Delta_t \rightarrow \infty$ (which implies $\alpha \rightarrow \infty$), then the modulus of the amplification factor could reach 1 in the special case of $\theta = 1/2$. In other words, the Crank-Nicolson scheme is *not* strongly *A-stable*. For large time steps, harmonics in the Crank-Nicolson finite difference solution will effectively not be damped from one time step to the next, opening up the possibility that unwanted high-frequency oscillations can creep into the numerical solution. In practice, this is primarily a problem if high-frequency eigenmodes have high amplification factors, as can happen if there is an outright discontinuity in the terminal value function g . The problem is especially noticeable if the discontinuity in the value function is “close” in both time and space to $t = 0$ and $x = x(0)$ (as would be the case for a short-dated option with a discontinuity close to the starting value of x). Oscillations can be prevented by setting the time step smaller than twice the maximum stable explicit time step (see Tavella and Randall [2000]), but this can often be computationally expensive. We shall deal with other methods to suppress oscillations in Section 2.5.

We conclude this section by noting a deep connection between the stability of a finite difference scheme and its convergence to the true solution

⁸If further $|\xi|$ approaches zero for $\Delta_t \rightarrow 0$, the scheme is said to be *L-stable*.

of the PDE as $\Delta_t \rightarrow 0$ and $\Delta_x \rightarrow 0$. First, we define a finite difference scheme to be *consistent* if local (Taylor) truncation errors approach zero for $\Delta_t \rightarrow 0$ and $\Delta_x \rightarrow 0$. All the schemes we have encountered so far are consistent. Further, define a finite difference scheme to be *convergent* if the difference between the numerical solution and the exact PDE solution at a fixed point in the domain converges to zero uniformly as $\Delta_t \rightarrow 0$ and $\Delta_x \rightarrow 0$ (not necessarily independently of each other). We then have

Theorem 2.3.2 (Lax Equivalence Theorem). *For a well-posed⁹ linear terminal value PDE, a consistent 2-level finite difference scheme is convergent if and only if it is stable.*

A more precise statement of the above result, as well as a proof, can be found in Mitchell and Griffiths [1980].

2.4 Non-Equidistant Discretization

In practice, we often wish to align the finite difference grid to particular dates (e.g., those on which a coupon or a dividend is paid) and particular values of x (e.g., those on which strikes and barriers are positioned). Also, for numerical reasons we may want to make certain important parts of the finite difference grid more densely spaced to concentrate computational effort on domains of particular importance to the solution of the PDE. To do so, we will now relax our earlier assumption of equidistant discretization in time and space. Doing so for the time domain is actually trivial and merely requires us to replace Δ_t in (2.18) with $\Delta_{t,i} \triangleq t_{i+1} - t_i$, where the spacing of the time grid $\{t_i\}_{i=0}^n$ is now no longer constant. The backward induction algorithm can proceed as before. We note that the ability to freely select the time grid will allow us to line up perfectly with dates that carry high significance for the product in question (e.g. dates on which cash flows take place, see Section 2.7.3) or to, say, use coarser time steps for the part of the finite difference grid that is far in the future. For an adaptive algorithm to automatically select the time-step, see d'Halluin et al. [2001].

For the spatial step, we have a number of options to induce non-equidistant spacing. One method involves a non-linear change of variables $y = h(x)$ in the PDE, followed by a regular equidistant discretization in the new variable y . This maps into a non-equidistant discretization in x which, provided that $h(\cdot)$ is chosen carefully, will have the desired geometry. Discussion of this method along with guidelines for choosing $h(\cdot)$ can be found in Chapter 5 of Tavella and Randall [2000]. We will here pursue a more direct alternative, where we simply introduce an irregular grid $\{x_j\}_{j=0}^{m+1}$

⁹Well-posed means that the PDE we are solving has a unique solution that depends continuously on the problem data (PDE coefficients, domain, boundary conditions, etc.)

and redefine the finite difference operators (2.5)–(2.6) to achieve maximum precision. For this, define

$$\Delta_{x,j}^+ \triangleq x_{j+1} - x_j, \quad \Delta_{x,j}^- \triangleq x_j - x_{j-1},$$

and set

$$\delta_x^+ V(t, x_j) = \frac{V(t, x_{j+1}) - V(t, x_j)}{\Delta_{x,j}^+}, \quad \delta_x^- V(t, x_j) = \frac{V(t, x_j) - V(t, x_{j-1})}{\Delta_{x,j}^-}.$$

By a Taylor expansion, we get

$$\begin{aligned} \delta_x^+ V(t, x_j) &= \frac{\partial V(t, x_j)}{\partial x} + \frac{1}{2} \frac{\partial^2 V(t, x_j)}{\partial x^2} \Delta_{x,j}^+ \\ &\quad + \frac{1}{6} \frac{\partial^3 V(t, x_j)}{\partial x^3} (\Delta_{x,j}^+)^2 + O((\Delta_{x,j}^+)^3), \end{aligned} \quad (2.25)$$

$$\begin{aligned} \delta_x^- V(t, x_j) &= \frac{\partial V(t, x_j)}{\partial x} - \frac{1}{2} \frac{\partial^2 V(t, x_j)}{\partial x^2} \Delta_{x,j}^- \\ &\quad + \frac{1}{6} \frac{\partial^3 V(t, x_j)}{\partial x^3} (\Delta_{x,j}^-)^2 + O((\Delta_{x,j}^-)^3). \end{aligned} \quad (2.26)$$

Maximum accuracy on the first-order derivative approximation is achieved by selecting a weighted combination of (2.25)–(2.26) such that the terms of order $O(\Delta_{x,j}^+)$ and $O(\Delta_{x,j}^-)$ cancel. That is, we set

$$\begin{aligned} \delta_x V(t, x_j) &= \frac{\Delta_{x,j}^-}{\Delta_{x,j}^+ + \Delta_{x,j}^-} \cdot \delta_x^+ V(t, x_j) + \frac{\Delta_{x,j}^+}{\Delta_{x,j}^+ + \Delta_{x,j}^-} \cdot \delta_x^- V(t, x_j) \quad (2.27) \\ &= \frac{\partial V(t, x_j)}{\partial x} + O\left(\frac{(\Delta_{x,j}^+)^2 \Delta_{x,j}^- + (\Delta_{x,j}^-)^2 \Delta_{x,j}^+}{\Delta_{x,j}^+ + \Delta_{x,j}^-}\right) \end{aligned}$$

which is second-order accurate, in the sense that reducing both $\Delta_{x,j}^+$ and $\Delta_{x,j}^-$ by a factor of k will reduce the error by a factor of k^2 . To estimate the derivative $\partial^2 V(t, x_j)/\partial x^2$ we set

$$\begin{aligned} \delta_{xx} V(t, x_j) &= \frac{\delta_x^+ V(t, x_j) - \delta_x^- V(t, x_j)}{\frac{1}{2} (\Delta_{x,j}^+ + \Delta_{x,j}^-)} \quad (2.28) \\ &= \frac{\partial^2 V(t, x_j)}{\partial x^2} + O\left(\frac{(\Delta_{x,j}^+)^2 - (\Delta_{x,j}^-)^2}{\Delta_{x,j}^+ + \Delta_{x,j}^-} + \frac{(\Delta_{x,j}^+)^3 + (\Delta_{x,j}^-)^3}{\Delta_{x,j}^+ + \Delta_{x,j}^-}\right) \end{aligned}$$

which is only first-order accurate, unless $\Delta_{x,j}^+ = \Delta_{x,j}^-$. Despite this, the global discretization error will typically remain second-order in the spatial step, even for a non-equidistant grid. A proof of this perhaps somewhat

surprising result can be found in the monograph Axelsson and Barker [1991] on finite element methods.

Development of a theta scheme around the definitions (2.27) and (2.28) proceeds in the same way as in Section 2.2. The resulting time-stepping scheme is identical to (2.18), after a modification of the matrix \mathbf{A} . Specifically, we must simply redefine the c -, u -, and l -arrays in (2.7)–(2.9) as follows:

$$c_j(t) \triangleq \frac{\Delta_{x,j}^+ - \Delta_{x,j}^-}{\Delta_{x,j}^+ \Delta_{x,j}^-} - \frac{1}{\Delta_{x,j}^- \Delta_{x,j}^+} \sigma(t, x_j)^2 - r(t, x_j), \quad (2.29)$$

$$u_j(t) \triangleq \frac{\Delta_{x,j}^-}{(\Delta_{x,j}^+ + \Delta_{x,j}^-) \Delta_{x,j}^+} \mu(t, x_j) + \frac{1}{(\Delta_{x,j}^+ + \Delta_{x,j}^-) \Delta_{x,j}^+} \sigma(t, x_j)^2, \quad (2.30)$$

$$l_j(t) \triangleq -\frac{\Delta_{x,j}^+}{(\Delta_{x,j}^+ + \Delta_{x,j}^-) \Delta_{x,j}^-} \mu(t, x_j) + \frac{1}{(\Delta_{x,j}^+ + \Delta_{x,j}^-) \Delta_{x,j}^-} \sigma(t, x_j)^2. \quad (2.31)$$

For an example where having a non-equidistant grid is essential to the numerical performance of the scheme, see Section 9.4.3.

2.5 Smoothing and Continuity Correction

2.5.1 Crank-Nicolson Oscillation Remedies

As discussed earlier, for discontinuous terminal conditions, the Crank-Nicolson scheme may exhibit localized oscillations if the time step is too coarse relative to the spatial step. Depending on the timing and spatial position of the discontinuities, these spurious oscillations may negatively affect the computed option value or, more likely, its first (“delta”) or second (“gamma”) x -derivatives. Further, in the presence of discontinuous terminal conditions, the expected $O(\Delta_t^2)$ convergence order of the Crank-Nicolson scheme may not be realized. While $O(\Delta_t^2)$ convergence is possible without spurious oscillations in some multi-level time-stepping schemes, there is evidence that these schemes are less robust than the Crank-Nicolson scheme for many financially relevant problems, see, e.g., Windcliff et al. [2001]. Fortunately, it is relatively easy to remedy the problems in the Crank-Nicolson scheme. Specifically, a theoretical result by Rannacher [1984] shows that second-order convergence can be achieved for the Crank-Nicolson scheme, provided that two simple algorithm modifications are taken:

- The discontinuous terminal payout is least-squares (L^2) projected onto the space of linear Lagrange basis functions¹⁰.

¹⁰Recall that the linear Lagrange basis functions (also called “hat” functions) are simply small triangles given by $l_j(x) = 1_{\{x_{j-1} < x \leq x_j\}} \cdot \frac{x - x_{j-1}}{x_j - x_{j-1}} + 1_{\{x_j < x \leq x_{j+1}\}} \cdot \frac{x_{j+1} - x}{x_{j+1} - x_j}$, $j = 1, \dots, m$. For an algorithm to perform the L^2 -projection, see Pooley et al. [2003].

- Two fully implicit time steps ($\theta = 1$) are taken before we switch to Crank-Nicolson ($\theta = \frac{1}{2}$) time stepping (“Rannacher stepping”).

Both techniques effectively smoothen out the discontinuity before the Crank-Nicolson scheme is applied, dampening the problematic high-frequency modes of the numerical solution. As demonstrated in Pooley et al. [2003] (see also Giles and Carter [2006]), applying either technique in isolation will typically not suffice; both are jointly required to ensure smooth second-order convergence. That said, the application of Lagrange basis function projection may conveniently be substituted with simpler smoothing techniques, with no loss of convergence order. The usefulness of such payoff smoothing extends beyond the case of discontinuous boundary conditions, so we proceed to discuss a few common techniques next.

2.5.2 Continuity Correction

By the Shannon sampling theorem, (see Shannon [1949]) if the spectrum of $g(x)$ contains frequencies higher than $1/(2\Delta_x)$ (the *Nyquist frequency*), information is lost when we sample $g(x)$ on our mesh $\{x_j\}_{j=0}^{m+1}$. In other words, whenever $g(x)$ or its derivatives are non-smooth, we will incur a *quantization* error where important features of the payout (e.g., the discontinuity of the slope of a call option at the strike) will be lost between grid points. As the grid geometry is modified, and the location of critical points (strikes, barriers, etc.) relative to x -grid changes, the computed finite difference solution will jump back and forth in erratic fashion. This so-called *odd-even effect* will result in poor convergence and an undesirably strong dependence of the solution on the grid geometry.

One straightforward way to reduce the odd-even effect (and to smooth out the high-frequency components of the payoff) is to apply a common technique from probability theory known as a *continuity correction*. Here, we simply imagine that the value of g at a grid point x_j represents the average value of the function over the interval $[x_j - (x_j - x_{j-1})/2, x_j + (x_{j+1} - x_j)/2]$. In setting the terminal boundary value $V(T, x_j)$ we thus write

$$V(T, x_j) = \frac{1}{(x_{j+1} - x_{j-1})/2} \int_{x_j - (x_j - x_{j-1})/2}^{x_j + (x_{j+1} - x_j)/2} g(x) dx. \quad (2.32)$$

We note that this implies that $V(T, x_j) \neq g(x_j)$, unless g is linear in x . The application of continuity correction to parabolic PDE solvers was first proposed in Kreiss et al. [1970].

2.5.3 Grid Shifting

Consider the effect of using (2.32) on a *digital call option*, $g(x) = 1_{\{x > H\}}$, where the level H (the digital strike) is located between nodes x_k and

x_{k+1} . For nodes x_j , $j > k + 1$, clearly $V(T, x_j) = 1$; for nodes x_j , $j < k$, $V(T, x_j) = 0$. The smoothing algorithm will have effect only at x_k or x_{k+1} , and will set either $V(T, x_k)$ or $V(T, x_{k+1})$ to a value somewhere between 0 and 1, depending on which of x_k or x_{k+1} is closest to H . If H happens to be exactly midway between x_k or x_{k+1} , the continuity correction is seen to have no effect whatsoever.

The digital option example above gives rise to a method listed in Tavella and Randall [2000] (see also Cheuk and Vorst [1996]). Here, we simply arrange the spatial grid such that the x -values where the payoff (or its derivatives) is discontinuous are exactly midway between grid nodes. If necessary, we can use a scheme with non-equidistant grid spacing to accomplish this (see Section 2.4). Our example above shows that aligning the grid in this way will, in a loose sense, make the payoff smooth.

For digital options, the grid shifting technique can be very efficient, and such “locking” of the location of strikes and barriers relative to the spatial grid can often reduce odd-even effects even better than the continuity correction discussed earlier. To demonstrate, consider the concrete task of using a finite difference grid to price a digital call option on a stock S in the Black-Scholes model. In this case, we conveniently have a theoretical option price to compare against, since it is easily shown that the time 0 value $V(0)$ must be

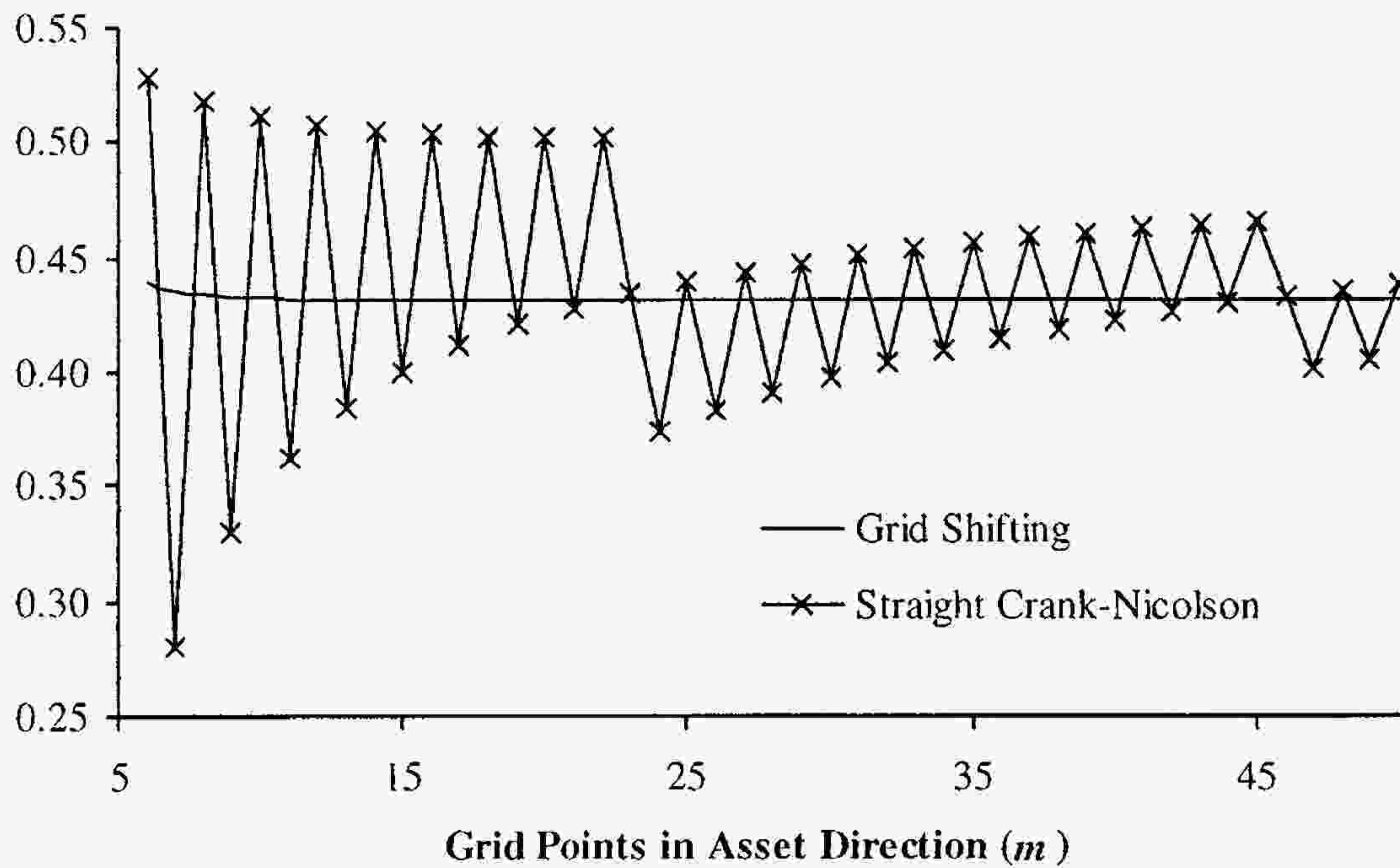
$$V(0) = e^{-rT} Q(S(T) > H) = e^{-rT} \Phi \left(\frac{\ln(S(0)/H) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} \right). \quad (2.33)$$

For our numerical work, we discretize the asset equidistantly in log-space (i.e., we work with the PDE (2.3)) and determine the spatial grid boundaries by probabilistic means using a multiplier of $\alpha = 4.5$, see Section 2.1. Spatial boundary conditions are $\partial V / \partial x = \partial^2 V / \partial x^2$, implemented as described in Section 2.2.2. In one experiment, we apply a straight Crank-Nicolson approach, with no attempt to regularize the payoff condition. In a second experiment, we combine Crank-Nicolson with Rannacher stepping and also nudge the entire spatial grid upwards until the log-barrier $\ln(H)$ is located exactly half-way between two spatial grid points. Numerical results are shown in Figure 2.1.

As Figure 2.1 shows, a naive Crank-Nicolson implementation is plagued by severe odd-even effects and very slow convergence — 100’s of spatial steps appear to be necessary before acceptable levels of the option price are reached. On the other hand, grid shifting combined with Rannacher stepping results in a perfectly smooth¹¹ convergence profile, and 5-digit price precision is here reached in less than 30 steps.

¹¹It can be verified that the convergence order in m is, as expected, close to 2 in this experiment.

Fig. 2.1. 3 Year Digital Option Price



Notes: Finite difference estimates for the Black-Scholes price of a 3 year digital option with a strike of $H = 100$. The initial asset price is $S(0) = 100$, the interest rate is $r = 0$, and the volatility is $\sigma = 20\%$. Time stepping is performed with an equidistant grid containing $n = 50$ points. Spatial discretization in log-space is equidistant, as described in the main text; the number of grid points (m) is as listed on the x -axis of the figure. The “Straight Crank-Nicolson” graph shows the convergence profile for a pure Crank-Nicolson finite difference grid. The “Grid Shifting” graph shows the convergence profile for a Crank-Nicolson finite difference grid with Rannacher stepping and a shift of the spatial grid to center $\ln(H)$ midway between two grid points. From (2.33), the theoretical value of the option is 0.4312451.

2.6 Convection-Dominated PDEs

Recall from Section 2.3 that stability of the explicit finite difference scheme requires that (omitting grid subscripts on μ and σ)

$$\frac{2\Delta_x^2}{\Delta_t} \sigma^2 \geq \sigma^4 + \mu^2 \Delta_x^2 + |\mu^2 \Delta_x^2 - \sigma^4|.$$

As discussed, this condition can be violated if Δ_t is too large relative to Δ_x . However, for fixed Δ_t and Δ_x we notice that instability can also be triggered if the absolute value of the drift μ is raised to be sufficiently large relative to the diffusion coefficient σ .

While theta schemes with $\theta \geq 1/2$ are always stable, large drifts in the PDE can nevertheless cause spurious oscillations and an overall deterioration in numerical performance of these schemes. PDEs for which this effect

occurs are said to be *convection-dominated*. To quantify matters, assume for simplicity that the finite difference grid is equidistant in the x -direction, and consider the matrix \mathbf{A} in (2.11) with tri-diagonal coefficients c , u , and l given by (2.7)–(2.9). As discussed in e.g. d’Halluin et al. [2005], spurious oscillations can occur when, for some t and some j , either $u_j(t) < 0$ or $l_j(t) < 0$. From (2.8) and (2.9), to avoid spurious oscillations we would thus need

$$\sigma(t, x_j)^2 \geq |\mu(t, x_j)|\Delta_x. \quad (2.34)$$

Intuitively, in convection-dominated systems, the central difference coefficient δ_x and δ_{xx} used to discretize the PDE can no longer fully contain the large expected up- or downward trend of the underlying process for x ; as a result, spurious oscillations can occur.

2.6.1 Upwinding

There are a number of well-established techniques to deal with convection-dominated PDEs. First, we can obviously attempt to lower Δ_x such that (2.34) is satisfied. This, however, may not be practical from a computational standpoint (and may require that Δ_t is lowered as well to avoid spurious oscillations originating from the time-stepping scheme). An alternative is to modify the first-order discrete operator δ_x such that it points in the direction of the large absolute drift. For instance, we can simply elect to use a suitably oriented one-sided difference, rather than a central difference, whenever (2.34) is violated. This procedure is known as *upstream differencing* or *upwinding*. To formalize the idea, introduce a new first-order difference operator δ_x^* given as

$$\delta_x^* V(t, x_j) = \begin{cases} \frac{1}{2} (V(t, x_{j+1}) - V(t, x_{j-1})) \Delta_x^{-1}, & |\mu(t, x_j)|\Delta_x \leq \sigma(t, x_j)^2, \\ (V(t, x_j) - V(t, x_{j-1})) \Delta_x^{-1}, & \mu(t, x_j)\Delta_x < -\sigma(t, x_j)^2, \\ (V(t, x_{j+1}) - V(t, x_j)) \Delta_x^{-1}, & \mu(t, x_j)\Delta_x > \sigma(t, x_j)^2. \end{cases}$$

Using δ_x^* instead of δ_x modifies the matrix \mathbf{A} in (2.11). Specifically, if $\mu(t, x_j)\Delta_x < -\sigma(t, x_j)^2$ we replace (2.7)–(2.9) with:

$$c_j(t) = \mu(t, x_j)\Delta_x^{-1} - \sigma(t, x_j)^2 \Delta_x^{-2} - r(t, x_j), \quad (2.35)$$

$$u_j(t) = \frac{1}{2}\sigma(t, x_j)^2 \Delta_x^{-2}, \quad (2.36)$$

$$l_j(t) = -\mu(t, x_j)\Delta_x^{-1} + \frac{1}{2}\sigma(t, x_j)^2 \Delta_x^{-2}. \quad (2.37)$$

And when $\mu(t, x_j)\Delta_x > \sigma(t, x_j)^2$, we use

$$c_j(t) = -\mu(t, x_j)\Delta_x^{-1} - \sigma(t, x_j)^2 \Delta_x^{-2} - r(t, x_j), \quad (2.38)$$

$$u_j(t) = \mu(t, x_j)\Delta_x^{-1} + \frac{1}{2}\sigma(t, x_j)^2 \Delta_x^{-2}, \quad (2.39)$$

$$l_j(t) = \frac{1}{2}\sigma(t, x_j)^2 \Delta_x^{-2}. \quad (2.40)$$

For non-equidistant grids, a similar modification to (2.29)–(2.31) is required. We omit the straightforward details.

Let us try to gain some further understanding of the upwind algorithm. Comparison of (2.35)–(2.40) with (2.7)–(2.9), shows that upwinding amounts to using a regular central difference operator δ_x on a PDE with a diffusion coefficient modified to be $\sigma(t, x) + \sqrt{|\mu(t, x)|\Delta_x}$. The numerical scheme in effect introduces enough artificial diffusion into the PDE to satisfy (2.34). Doing so, however, comes at a cost: the convergence order of the scheme will be reduced to $O(\Delta_x)$ if one-sided differencing ends up being activated in a significant part of the grid. We note that higher-order upwinding schemes are possible if the finite difference operator δ_x^* is allowed to act on more than three neighboring points. For such schemes, the matrix \mathbf{A} will no longer be tri-diagonal.

2.6.2 Other Techniques

As discussed earlier, upwinding amounts to adding numerical diffusion at nodes where the scheme is convection dominated. Alternatively, we can increase $\sigma(t, x)$ directly, to $\sigma(t, x) + \varepsilon$ where ε is chosen to be large enough for the scheme to satisfy (2.34). By solving the resulting PDE for different values of ε , it may be possible to determine how the error associated with ε scales in ε . This, in turn, will allow us to extrapolate to the limit $\varepsilon = 0$. See p. 135 of Tavella and Randall [2000] for an example.

The upwinding scheme presented in Section 2.6.1 switches abruptly from central differencing to one-sided differencing when the condition (2.34) is violated. In some schemes, the switch from central to one-sided differencing is made smooth by using a weighted average of a one-sided and a central difference operator. The weight on the central difference is close to one when $\sigma(t, x)^2 \gg |\mu(t, x)|\Delta_x$, but decreases smoothly to zero as $\sigma(t, x)^2/|\mu(t, x)|$ tends to zero. While it is unclear whether a smooth transition to upwinding is truly important (the convergence order is typically not improved over straight upwinding), Duffy [2000] suggests that the class of exponentially fitted schemes (see Duffy [2000] and Stoyan [1979]) may be quite robust in derivatives pricing applications.

In some finance applications, multi-dimensional PDEs might arise where $\sigma(t, x) = 0$ for one of the underlying variables; see for instance Section 2.7.5. While upwinding techniques still apply here, we note that specialized methods exist with better ($O(\Delta_x^2)$) convergence, should they become necessary. See, for instance, d'Halluin et al. [2005] for details on the so-called *semi-Lagrangian* methods.

2.7 Option Examples

In our discussion so far, we have assumed that options are characterized by a single terminal payoff function $g(x)$ and a set of spatial boundary

While the question of how to exercise optimally on a chooser cap may appear quite complex, it is surprisingly easy to implement in a finite difference setting by combining techniques from Sections 2.7.4 and 2.7.5 above. The key to the method is to introduce an additional state variable I to keep track of how many exercise opportunities are left. Assume that all interest rates are functions of a Markov state variable $x(\cdot)$, and let therefore $V(t, x, I)$ denote the value of the chooser cap at time t , given $x(t) = x$ and given that there are still I exercise opportunities left. Notice that the variable I can only take $l + 1$ distinct values: $0, 1, \dots, l$; notice also that $V(t, x, 0) = 0$ for all t and x , since $I = 0$ corresponds to the situation where there are no exercise opportunities left. Additionally, at the terminal time T_L we clearly have

$$V(T_L, x, I) = (S(T_L, x) - K)^+, \quad I = 1, 2, \dots, l, \quad (2.51)$$

where we have written $S(T_L) = S(T_L, x)$ to emphasize the deterministic dependence of S on the state variable x .

For given dynamics of $x(t)$, starting with the terminal conditions in (2.51), we may roll the l different value functions $V(\cdot, x, I)$, $I = 1, 2, \dots, l$, back through time in standard finite difference manner. At each time T_i , $i = 1, \dots, L - 1$, jump conditions similar to (2.45) must be applied, for all $I = 1, 2, \dots, l$:

$$V(T_i^-, x, I) = \max(V(T_i^+, x, I), V(T_i^+, x, I - 1) + (S(T_i, x) - K)^+).$$

Notice that these conditions simply express that exercise is optimal only if the exercise value (the cap payout plus the value of a chooser cap with one less exercise opportunity) exceeds the hold value (the non-exercised chooser cap). Once we have rolled all the way back to $t = 0$, the chooser cap value at time $t = 0$ may be identified as $V(0) = V(0, x(0), l)$.

We should note that the “chooser” or “flexi” feature can be added to securities other than caps (and floors). For instance, in Section 19.5 we study the so-called *flexi-swap*, another security with multiple embedded exercise rights.

2.8 Special Issues

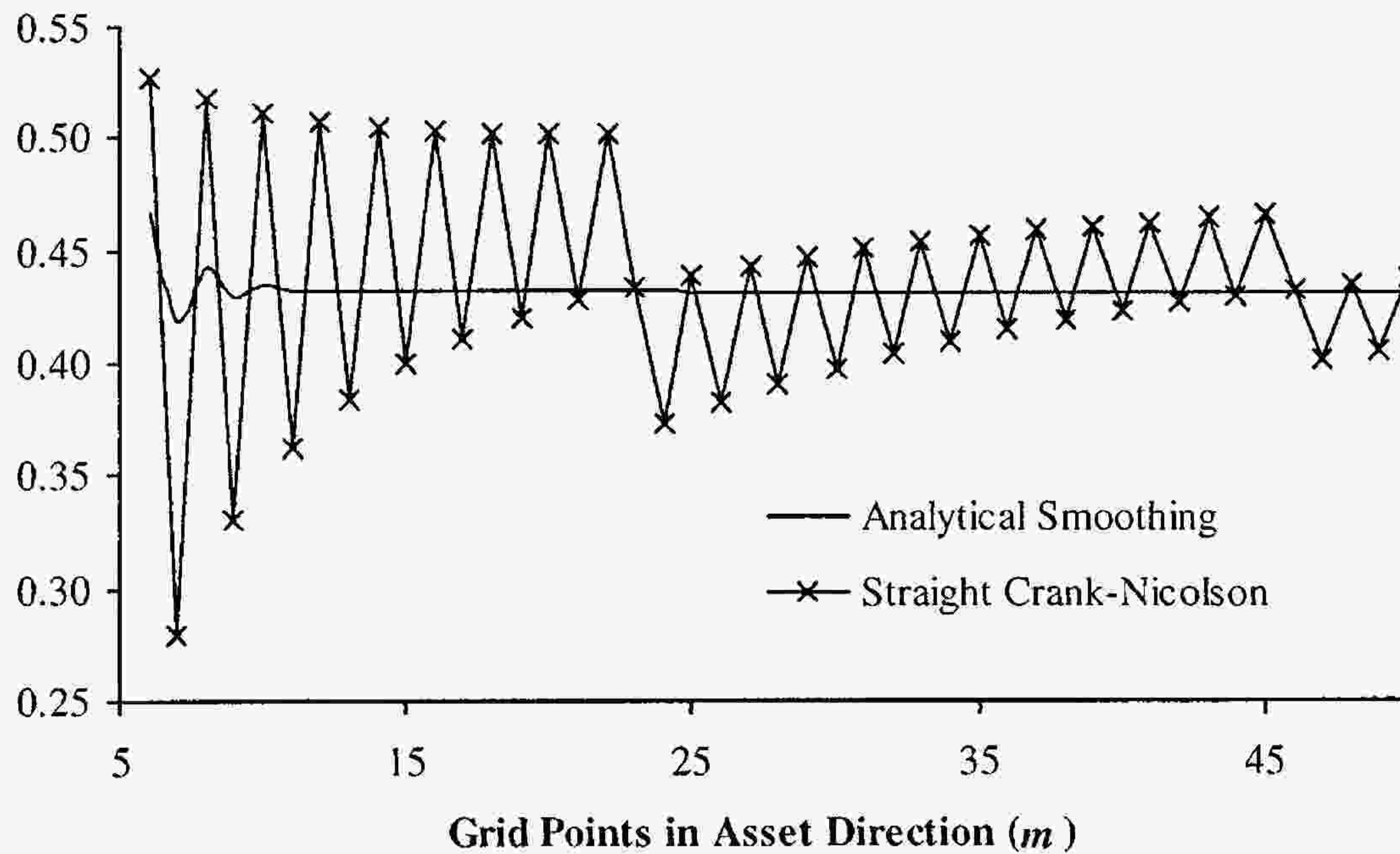
In this section, we briefly show a few techniques that may come in handy for certain applications.

2.8.1 Mesh Refinements for Multiple Events

As discussed in Section 2.1, the domain of the state variable x is often determined as an exact or approximate confidence interval for the random variable $x(T)$, where T is the final time of interest for a particular valuation problem we want to solve. Given the number of desired spatial steps in the

effect is discussed in more detail in Section 23.2.4 and is also demonstrated below, in Figure 2.2, where we have continued our investigation of the 3 year digital option considered earlier in Section 2.5.3. Since the model used in Figure 2.2 is ordinary Black-Scholes and $g(x) = 1_{\{x>H\}}$, the integrals in (2.55) can here be computed in closed form from (2.33).

Fig. 2.2. 3 Year Digital Option Price



Notes: Finite difference estimates for the Black-Scholes price of a 3 year digital option. All contract and model parameters are as in Figure 2.1. Time stepping is performed with an equidistant grid containing $n = 50$ points. Spatial discretization in log-space is equidistant, as described in the main text; the number of grid points (m) is as listed on the x -axis of the figure. The “Straight Crank-Nicolson” graph shows the convergence profile for a pure Crank-Nicolson finite difference grid. The “Analytical Smoothing” graph shows the convergence profile for a Crank-Nicolson finite difference grid starting at $T^* = 2.5$ years, with the terminal boundary condition set equal to a 0.5 year digital option price (as in (2.55)). The theoretical value of the option is 0.4312451.

In principle, we could continue rolling back from T^* (through, possibly, jump conditions at earlier times) by performing convolutions, rather than solving finite difference grids. In practice, this rarely leads to improvements over a finite difference grid, unless the densities and payoffs are quite simple¹⁸. Moreover, in many cases we may not have *exact* Arrow-Debreu

¹⁸For simple densities (especially Gaussian), special-purpose methods exist to compute convolutions rapidly, typically involving payoff approximations through piecewise polynomials or other simple functions. We do not cover these methods in

prices, only approximate ones based on, say, a small-time expansion (see, e.g., Section 13.1.9.1). In this case, a one-time convolution may be safe — especially if $T - T^*$ is small — whereas repeated convolutions may lead to unacceptable biases.

2.8.3 Analytics at the First Time Step

The idea in Section 2.8.2 of replacing the finite difference stepping with analytical integration is even easier to apply over the *first*, rather than the *last*, time step. Suppose T^* is the first “interesting” time for a given derivative security, i.e. there might be a jump condition at time T^* but none over the time interval $[0, T^*]$. Then, rather than stepping the finite difference scheme from T^* to 0, we can perform a *single* integration to calculate the value $V(0, x(0))$ of the derivative at time zero from the discretized values $\{V(T^*, x_j)\}_{j=0}^{m+1}$ of the derivative at time T^* (using the same notations as in Section 2.8.2),

$$V(0, x(0)) = \int_{\mathbb{R}} G(0, x(0); T^*, y) \tilde{V}(T^*, y) dy,$$

where $\tilde{V}(T^*, y)$ is interpolated (using cubic splines, say) from the values $\{V(T^*, x_j)\}_{j=0}^{m+1}$ on the grid. If the integral is computed numerically — as is most often the case — the numerical cost is often comparable with that of the finite difference stepping because only one value $V(0, x(0))$ is required at time 0, not the whole slice.

While there are typically no numerical cost savings that arise from using integration over the first time step, there are accuracy and stability considerations that favor this approach. We have already seen in Section 2.8.1 that the standard discretization of a PDE often leads to insufficient fidelity in resolving any features of the payoff that are close to today, and numerical integration can be of considerable help in this regard. Moreover, as we discuss in much detail later in Chapter 23, an integration scheme typically allows us to treat discontinuities in the value $V(T^*, x)$ arising from the jump condition at time T^* explicitly. If the discontinuity is introduced at the value of the state variable x^* , then the integration scheme can (and should) explicitly take this information into account. For example we would write

$$\begin{aligned} V(0, x(0)) &= \int_{-\infty}^{x^*} G(0, x(0); T^*, y) \tilde{V}^-(T^*, y) dy \\ &\quad + \int_{x^*}^{\infty} G(0, x(0); T^*, y) \tilde{V}^+(T^*, y) dy \end{aligned}$$

and calculate $\tilde{V}^-(T^*, y)$ by interpolating the grid values in the time interval $(-\infty, x^*)$, and $\tilde{V}^+(T^*, y)$ by interpolating the grid values in (x^*, ∞) , separately¹⁹.

The usefulness of the method is only limited by the availability of the closed-form expression for the time 0 Arrow-Debreu prices $G(0, x(0); T^*, \cdot)$. For some models this is not an issue; for most others, sufficiently close approximations could be obtained in a small-time limit (see e.g. Section 13.1.9.1 for a typical approach) that can be useful for times T^* that are not too large. By a change of measure, we see that

$$\begin{aligned} V(0, x(0)) &= \mathbb{E} \left(e^{-\int_0^{T^*} r(s) ds} V(T^*, x(T^*)) \right) \\ &= P(0, T^*) \mathbb{E}^{T^*} (V(T^*, x(T^*))), \end{aligned}$$

where \mathbb{E}^{T^*} is the expected value operator under the T^* -forward measure Q^{T^*} ; so we really only need the expression for the *density* (rather than Arrow-Debreu security prices) of $x(T^*)$ under Q^{T^*} , either exact or approximate.

Finally, we note that while the integration over the first time step can be seen to offer similar advantages to those of the methods in Section 2.8.1, the two approaches are not substitutes for each other, but are complementary. We typically recommend using direct integration over the time step $[0, T^*]$, where T^* is the smaller of the time of the first jump condition or the limit of applicability of the approximation to the density of $x(T^*)$, and then (if needed) use the methods in Section 2.8.1 over the time interval $[T^*, T]$, with T being the final maturity of the option in question.

2.9 Multi-Dimensional PDEs: Problem Formulation

We now turn our attention to the numerical solution of multi-dimensional terminal value problems. Let the spatial variable x be p -dimensional, $x = (x_1, \dots, x_p)^\top$, and consider the PDE

$$\frac{\partial V}{\partial t} + \sum_{h=1}^p \mu_h(t, x) \frac{\partial V}{\partial x_h} + \frac{1}{2} \sum_{h=1}^p \sum_{l=1}^p s_{h,l}(t, x) \frac{\partial^2 V}{\partial x_h \partial x_l} - r(t, x) = 0, \quad (2.56)$$

where $s_{h,h}(t, x) \geq 0$ and $s_{h,l}(t, x) = s_{l,h}(t, x)$ for $h, l = 1, \dots, p$. The PDE is assumed subject to the terminal value condition $V(T, x) = g(x)$, $g : \mathbb{R}^p \rightarrow \mathbb{R}$.

From the results in Chapter 1, we recognize that the PDE provides the solution to the expectation

¹⁹One of the functions $\tilde{V}^-(T^*, y), \tilde{V}^+(T^*, y)$ is often known analytically and for all values of y (rather than sampled on the grid); this is for instance the case for the Bermudan options of Section 2.7.4. The integration algorithm should obviously take advantage of this.

$$V(t, x) = \mathbb{E}_t \left(e^{-\int_0^T r(u, x) du} g(x(T)) | x(t) = x \right),$$

where the components of $x(t)$ satisfy risk-neutral SDEs of the type

$$dx_h(t) = \mu_h(t, x(t)) dt + \sigma_h(t, x(t)) dW(t), \quad h = 1, \dots, p. \quad (2.57)$$

Here $W(t)$ is a d -dimensional Brownian motion, $\mu_h : [0, T] \times \mathbb{R}^p \rightarrow \mathbb{R}$, $h = 1, \dots, p$, are (scalar) drifts, and $\sigma_h : [0, T] \times \mathbb{R}^p \rightarrow \mathbb{R}^{1 \times d}$, $h = 1, \dots, p$, are d -dimensional (row vector) diffusion coefficients. The PDE coefficients $s_{h,l}$ in (2.56) represent the instantaneous covariance matrix for the components of $x(\cdot)$, i.e., $s_{h,l}(t, x) = \sigma_h(t, x)\sigma_l(t, x)^\top$. We assume enough regularity on μ_h , σ_h , r , and g to ensure that (2.56) has a unique solution.

For the purpose of solving (2.56) numerically, we assume that the PDE is to be solved on a (finite) spatial domain in x , $x \in [\underline{M}_1, \bar{M}_1] \times \dots \times [\underline{M}_p, \bar{M}_p]$, where $\underline{M}_h, \bar{M}_h$, $h = 1, \dots, p$, are constants either dictated by the contract at hand (barrier options) or found by a suitable probabilistic truncation (see Section 2.1).

2.10 Two-Dimensional PDE with No Mixed Derivatives

To illustrate the construction of finite difference discretization of (2.56), we start out with the simple case where $p = d = 2$ and there are no mixed partial derivatives in the PDE: $s_{1,2}(t, x) = s_{2,1}(t, x) = 0$ for all t and x . Probabilistically, the absence of mixed derivatives corresponds to the case where the stochastic process increments $dx_1(t)$ and $dx_2(t)$ are independent. Defining $\gamma_h(t, x)^2 = s_{h,h}(t, x)$, $h = 1, 2$, the PDE to be solved now becomes

$$\frac{\partial V}{\partial t} + (\mathcal{L}_1 + \mathcal{L}_2) V = 0, \quad (2.58)$$

where

$$\mathcal{L}_h \triangleq \mu_h(t, x) \frac{\partial}{\partial x_h} + \frac{1}{2} \gamma_h(t, x)^2 \frac{\partial^2}{\partial x_h^2} - \frac{1}{2} r(t, x), \quad h = 1, 2.$$

Notice that we have divided the term $r(t, x)$ into equal pieces in \mathcal{L}_1 and \mathcal{L}_2 .

To discretize (2.58) in x , introduce grids $x_1 \in \{x_1^{j_1}\}_{j_1=0}^{m_1+1}$ and $x_2 \in \{x_2^{j_2}\}_{j_2=0}^{m_2+1}$. To simplify notation, assume these grids are equidistant such that $x_1^{j_1} = \underline{M}_1 + j_1 \Delta_1$ and $x_2^{j_2} = \underline{M}_2 + j_2 \Delta_2$. Let $V_{j_1, j_2}(t) \triangleq V(t, x_1^{j_1}, x_2^{j_2})$. We define discrete central difference operators as before

$$\begin{aligned} \delta_{x_1} V_{j_1, j_2}(t) &= \frac{V_{j_1+1, j_2}(t) - V_{j_1-1, j_2}(t)}{2\Delta_1}, \\ \delta_{x_2} V_{j_1, j_2}(t) &= \frac{V_{j_1, j_2+1}(t) - V_{j_1, j_2-1}(t)}{2\Delta_2}, \end{aligned}$$

and

$$\delta_{x_1 x_1} V_{j_1, j_2}(t) = \frac{V_{j_1+1, j_2}(t) - 2V_{j_1, j_2}(t) + V_{j_1-1, j_2}(t)}{\Delta_1^2},$$

$$\delta_{x_2 x_2} V_{j_1, j_2}(t) = \frac{V_{j_1, j_2+1}(t) - 2V_{j_1, j_2}(t) + V_{j_1, j_2-1}(t)}{\Delta_2^2}.$$

These operators, in turn, give rise to the discrete operators

$$\widehat{\mathcal{L}}_h \triangleq \mu_h(t, x)\delta_{x_h} + \frac{1}{2}\gamma_h(t, x)^2\delta_{x_h x_h} - \frac{1}{2}r(t, x), \quad h = 1, 2,$$

where x is constrained to take values in the spatial grid. A Taylor expansion shows that this operator is second-order accurate (compare to Lemma 2.2.1),

$$(\mathcal{L}_1 + \mathcal{L}_2)V(t, x) = (\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2)V(t, x) + O(\Delta_1^2 + \Delta_2^2).$$

2.10.1 Theta Method

Turning to a theta-style time discretization, consider first proceeding exactly as in Section 2.2.3. Assuming equidistant time spacing Δ_t , we get for the period $[t_i, t_{i+1}]$,

$$\left(1 - \theta\Delta_t (\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2)\right) V_{j_1, j_2}(t_i) = \left(1 + (1 - \theta)\Delta_t (\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2)\right) V_{j_1, j_2}(t_{i+1}) + e_i^{i+1},$$

where

$$e_i^{i+1} = O\left(\Delta_t \left(\Delta_1^2 + \Delta_2^2 + 1_{\{\theta \neq \frac{1}{2}\}}\Delta_t + \Delta_t^2\right)\right),$$

and where it is understood that $\widehat{\mathcal{L}}_1$ and $\widehat{\mathcal{L}}_2$ are to be evaluated at $(t, x) = (t_i^{i+1}(\theta), x_1^{j_1}, x_2^{j_2})$ with $t_i^{i+1}(\theta)$ defined as in (2.14). If $\widehat{V}_{j_1, j_2}(t) \triangleq \widehat{V}(t, x_1^{j_1}, x_2^{j_2})$ is a finite difference approximation to $V_{j_1, j_2}(t)$, we thus get the scheme

$$\left(1 - \theta\Delta_t (\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2)\right) \widehat{V}_{j_1, j_2}(t_i) = \left(1 + (1 - \theta)\Delta_t (\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2)\right) \widehat{V}_{j_1, j_2}(t_{i+1}), \quad (2.59)$$

to be solved for the $m_1 m_2$ interior points $\widehat{V}_{j_1, j_2}(t_i)$, $j_1 = 1, \dots, m_1$, $j_2 = 1, \dots, m_2$, given the values of $\widehat{V}_{j_1, j_2}(t_{i+1})$, and given appropriate boundary conditions at $j_1 = 0$, $j_1 = m_1 + 1$, $j_2 = 0$, and $j_2 = m_2 + 1$.

The scheme (2.59) represents a system of linear equations in $m_1 m_2$ unknowns $\{\widehat{V}_{j_1, j_2}(t_i)\}$. When written out as a matrix equation (which requires us to arrange the various $\widehat{V}_{j_1, j_2}(t_i)$ in some order in a $(m_1 m_2)$ -dimensional vector), the matrix to be inverted is sparse but, unfortunately, no longer tri-diagonal. Solution of the system of equations by standard methods (e.g., Gauss-Jordan elimination or LU decomposition) is out of the question due

to the size of the matrix²⁰. We can proceed in two ways: either we use a specialized sparse-matrix solver; or we attempt to redo the discretization (2.59) to make it computationally efficient. We personally prefer the second approach and shall outline one method in the next section. As for the first approach, we simply note that a good iterative sparse solver should be able to solve (2.59) in order $O((m_1 m_2)^{5/4})$ operations. See Saad [2003] for concrete algorithms.

2.10.2 The Alternating Direction Implicit (ADI) Method

The ADI method is an example of a so-called *operator splitting* method, where the simultaneous application of two operators (here $\widehat{\mathcal{L}}_1$ and $\widehat{\mathcal{L}}_2$) is split into two *sequential* operator applications. To illustrate the idea, set $\theta = \frac{1}{2}$ (Crank-Nicolson scheme) in (2.59) and approximate

$$\left(1 - \frac{1}{2}\Delta_t (\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2)\right) \approx \left(1 - \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_1\right) \left(1 - \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_2\right), \quad (2.60)$$

$$\left(1 + \frac{1}{2}\Delta_t (\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2)\right) \approx \left(1 + \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_1\right) \left(1 + \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_2\right). \quad (2.61)$$

It is easy to see²¹ (and to verify, by a Taylor expansion) that the operators on the right-hand sides of these approximations have the same order truncation error as do the left-hand sides, namely $O(\Delta_t(\Delta_1^2 + \Delta_2^2 + \Delta_t^2))$. To the order of our original scheme, no accuracy is gained or lost in using the right-hand sides of (2.60)–(2.61). What is gained, however, is a considerable improvement in computational efficiency, originating in the fact that the resulting scheme

$$\begin{aligned} & \left(1 - \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_1\right) \left(1 - \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_2\right) \widehat{V}_{j_1,j_2}(t_i) \\ &= \left(1 + \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_1\right) \left(1 + \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_2\right) \widehat{V}_{j_1,j_2}(t_{i+1}) \end{aligned} \quad (2.62)$$

can be *split* into the system

$$\left(1 - \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_1\right) U_{j_1,j_2} = \left(1 + \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_2\right) \widehat{V}_{j_1,j_2}(t_{i+1}), \quad (2.63)$$

$$\left(1 - \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_2\right) \widehat{V}_{j_1,j_2}(t_i) = \left(1 + \frac{1}{2}\Delta_t \widehat{\mathcal{L}}_1\right) U_{j_1,j_2}, \quad (2.64)$$

²⁰Recall that the solution of a general linear system with $m_1 m_2$ unknowns is an $O(m_1^2 m_2^2)$ operation. For, say, m_1 and m_2 in the order of 100, this would involve around 1,000,000 times more work than what is required for a one-dimensional (tri-diagonal) scheme ($O(m)$).

²¹To those versed in operator notation, we notice that the right- and left-hand sides both approximate, to identical order, $\exp(\pm 0.5\Delta_t(\widehat{\mathcal{L}}_1 + \widehat{\mathcal{L}}_2))$.

where we have introduced an *intermediate value* U_{j_1, j_2} . The advantage of this decomposition is the fact that in each of (2.63) and (2.64), there is only one operator on the left-hand side, leading to simple tri-diagonal equation systems. To formalize this, first define

$$\mathbf{U}_1^{j_2} = (U_{1,j_2}, U_{2,j_2}, \dots, U_{m_1,j_2})^\top.$$

Then, for a fixed value of j_2 we can write for the first step

$$\left(\mathbf{I} - \frac{1}{2} \Delta_t \mathbf{A}_1^{j_2} \left(\frac{t_{i+1} + t_i}{2} \right) \right) \mathbf{U}_1^{j_2} = \mathbf{M}_2^{j_2} \left(\frac{t_{i+1} + t_i}{2} \right), \quad (2.65)$$

where $\mathbf{A}_1^{j_2}$ is an $(m_1 \times m_1)$ -dimensional tri-diagonal matrix of the same form as (2.11) (to get $\mathbf{A}_1^{j_2}$, basically freeze $x_2 = x_2^{j_2}$ and substitute μ_1 and γ_1 for μ and σ in the definition of the one-dimensional matrix \mathbf{A}). The m_1 -dimensional vector $\mathbf{M}_2^{j_2}$ has components $M_{2,j_1}^{j_2}, j_1 = 1, \dots, m_1$, given by

$$\begin{aligned} M_{2,j_1}^{j_2} \left(\frac{t_{i+1} + t_i}{2} \right) &= \left(1 + \frac{1}{2} \Delta_t \hat{\mathcal{L}}_2 \right) \hat{V}_{j_1, j_2}(t_{i+1}) \\ &= \frac{1}{2} \varsigma_{j_1, j_2}^- \hat{V}_{j_1, j_2-1}(t_{i+1}) + \frac{1}{2} \varsigma_{j_1, j_2}^+ \hat{V}_{j_1, j_2+1}(t_{i+1}) \\ &\quad + \left(1 - \frac{1}{2} \varsigma_{j_1, j_2} \right) \hat{V}_{j_1, j_2}(t_{i+1}), \end{aligned} \quad (2.66)$$

where we have defined

$$\begin{aligned} \varsigma_{j_1, j_2}^\pm &\triangleq \frac{\Delta_t}{2\Delta_2^2} \left(\gamma_2 \left(\frac{t_{i+1} + t_i}{2}, x_1^{j_1}, x_2^{j_2} \right)^2 \pm \Delta_2 \mu_2 \left(\frac{t_{i+1} + t_i}{2}, x_1^{j_1}, x_2^{j_2} \right) \right), \\ \varsigma_{j_1, j_2} &\triangleq \frac{\Delta_t}{\Delta_2^2} \left(\gamma_2 \left(\frac{t_{i+1} + t_i}{2}, x_1^{j_1}, x_2^{j_2} \right)^2 + \frac{1}{2} \Delta_2^2 r \left(\frac{t_{i+1} + t_i}{2}, x_1^{j_1}, x_2^{j_2} \right) \right). \end{aligned}$$

For known values of $\hat{V}(t_{i+1})$, (2.65) defines a simple tri-diagonal equation system which can be solved for $\mathbf{U}_1^{j_2}$ in $O(m_1)$ operations. Repeating the procedure above for $j_2 = 1, \dots, m_2$ allows us to find U_{j_1, j_2} for all $j_1 = 1, \dots, m_1, j_2 = 1, \dots, m_2$, at a total computational cost of $O(m_1 m_2)$.

Turning to the second step of (2.63)–(2.64), we first fix j_1 and define

$$\hat{\mathbf{V}}_2^{j_1}(t) = (\hat{V}_{j_1, 1}(t), \hat{V}_{j_1, 2}(t), \dots, \hat{V}_{j_1, m_2}(t))^\top.$$

In the same fashion as earlier, we can then write

$$\left(\mathbf{I} - \frac{1}{2} \Delta_t \mathbf{A}_2^{j_1} \left(\frac{t_{i+1} + t_i}{2} \right) \right) \hat{\mathbf{V}}_2^{j_1}(t_i) = \mathbf{M}_1^{j_1} \left(\frac{t_{i+1} + t_i}{2} \right), \quad (2.67)$$

where $\mathbf{A}_2^{j_1}$ is an $(m_2 \times m_2)$ -dimensional tri-diagonal matrix and where the right-hand side vector now has components

$$M_{1,j_2}^{j_1} \left(\frac{t_{i+1} + t_i}{2} \right) = \left(1 + \frac{1}{2} \Delta_t \hat{\mathcal{L}}_1 \right) U_{j_1,j_2}, \quad j_2 = 1, \dots, m_2.$$

For brevity we omit writing out the $M_{1,j_2}^{j_1}$ (which will be similar to (2.66)), but just notice that the right-hand side of (2.67) is known after the first step of the ADI algorithm (above) is complete. For a given value of j_1 , we can solve the tri-diagonal system (2.67) for $\hat{V}_2^{j_1}(t_i)$ in $O(m_2)$ operations. Looping over all m_1 different values of j_1 , the full matrix of time t_i values $\hat{V}_{j_1,j_2}(t_i)$, $j_1 = 1, \dots, m_1$, $j_2 = 1, \dots, m_2$, can then be found at a total computational cost of $O(m_1 m_2)$.

The scheme outlined above is known as the *Peaceman-Rachford* scheme. As is the case for all ADI schemes, the scheme works by alternating the directions that are treated fully implicitly in the finite difference grid: in the first step, the x_1 -direction is fully implicit and the x_2 -direction is fully explicit, and in the second step the order is reversed. In effect, both spatial variables end up being discretized “semi-implicitly”, i.e. similar to a Crank-Nicolson scheme, resulting in convergence order is $O(\Delta_1^2 + \Delta_2^2 + \Delta_t^2)$. We emphasize, however, that whereas a direct application of the Crank-Nicolson scheme will involve (if an efficient sparse-matrix solver is used) a computational cost of $O((m_1 m_2)^{5/4})$ per time step, the computational cost of the Peaceman-Rachford ADI scheme is only $O(m_1 m_2)$. A (tedious) von Neumann analysis reveals that the scheme is A -stable, but, like the Crank-Nicolson scheme, not strongly A -stable.

While the Peaceman-Rachford scheme is a classical example of an ADI scheme, there are many others. For instance, consider a theta-version of the *Douglas-Rachford* scheme:

$$(1 - \theta \Delta_t \hat{\mathcal{L}}_1) U_{j_1,j_2} = (1 + (1 - \theta) \Delta_t \hat{\mathcal{L}}_1 + \Delta_t \hat{\mathcal{L}}_2) \hat{V}_{j_1,j_2}(t_{i+1}), \quad (2.68)$$

$$(1 - \theta \Delta_t \hat{\mathcal{L}}_2) \hat{V}_{j_1,j_2}(t_i) = U_{j_1,j_2} - \theta \Delta_t \hat{\mathcal{L}}_2 \hat{V}_{j_1,j_2}(t_{i+1}), \quad (2.69)$$

where we understand that in $\hat{\mathcal{L}}_1$ and $\hat{\mathcal{L}}_2$ the PDE coefficients are to be evaluated at time $t_i^{i+1}(\theta)$. Again, notice how the scheme consists of two steps, each involving the solution of tri-diagonal sets of equations along only one of the x_1 - or x_2 -directions. The computational cost thus remains at $O(m_1 m_2)$. It can be shown that the convergence order of this scheme is $O(\Delta_1^2 + \Delta_2^2 + 1_{\{\theta \neq \frac{1}{2}\}} \Delta_t + \Delta_t^2)$ and it is A -stable for $\theta \geq \frac{1}{2}$, and strongly A -stable for $\theta > \frac{1}{2}$. By elimination of U_{j_1,j_2} we note that the unsplit version of the Douglas-Rachford scheme is

$$\begin{aligned} & (1 - \theta \Delta_t \hat{\mathcal{L}}_1) (1 - \theta \Delta_t \hat{\mathcal{L}}_2) \hat{V}_{j_1,j_2}(t_i) \\ &= ((1 - \theta \Delta_t \hat{\mathcal{L}}_1) (1 - \theta \Delta_t \hat{\mathcal{L}}_2) + \Delta_t \hat{\mathcal{L}}_1 + \Delta_t \hat{\mathcal{L}}_2) \hat{V}_{j_1,j_2}(t_{i+1}). \end{aligned}$$

It is not difficult to see that this approximates (2.59) to second order.

2.10.3 Boundary Conditions and Other Issues

The fact that ADI schemes reduce to solving sequences of matrix systems identical to the ones arising in the one-dimensional case is convenient, in the sense that many of the issues we have encountered for one-dimensional finite difference grids (oscillations, stability, convection dominance, etc.) and their remedies (smoothing, non-equidistant discretization, upwinding, etc.) carry over to the ADI setting with only minor modifications. Consider for instance the issue of applying spatial boundary conditions along the edges of the (x_1, x_2) domain, which we have so far not discussed. As for the one-dimensional PDEs, the most convenient way to express such boundary conditions is typically by imposing conditions on derivatives, like $\partial^2 V(t, x_1^0, x_2^{j_2}) / \partial x_1^2 = \partial V(t, x_1^0, x_2^{j_2}) / \partial x_1$ and so forth. For the Peaceman-Rachford scheme, say, such conditions can be incorporated directly into (2.65) and (2.67) by altering the matrices $A_1^{j_2}$ and $A_2^{j_1}$, as well as the boundary elements of $M_1^{j_1}$ and $M_2^{j_2}$, in the manner outlined in Section 2.2.1. If instead we wish to impose Dirichlet boundary conditions, we need to add corrective terms to the tri-diagonal systems, as in (2.19). To complete the first part of the split scheme, this then requires us to establish what boundary terms are needed for the intermediate quantity U_{j_1, j_2} , i.e. we must define $U_{j_1, 0}$ and U_{j_1, m_2+1} for $j_1 = 1, \dots, m_1$, as well as U_{0, j_2} and U_{m_1+1, j_2} for $j_2 = 1, \dots, m_2$. While U_{j_1, j_2} is a purely mathematic construct, sometimes it is adequate to think of U_{j_1, j_2} as a proxy for V_{j_1, j_2} evaluated at $t_i^{i+1}(\theta)$, which obviously makes determination of boundary conditions straightforward. For maximum precision, however, we should use the ADI equations themselves to express the boundary conditions of U directly in terms of boundary conditions for $V(t_i)$ and $V(t_{i+1})$. Here, the Douglas-Rachford scheme is particularly easy to deal with, as a rearrangement of (2.69) directly relates U_{j_1, j_2} to $\hat{V}_{j_1, j_2}(t_i)$ and $\hat{V}_{j_1, j_2}(t_{i+1})$,

$$U_{j_1, j_2} = (1 - \theta \Delta_t \hat{\mathcal{L}}_2) \hat{V}_{j_1, j_2}(t_i) + \theta \Delta_t \hat{\mathcal{L}}_2 \hat{V}_{j_1, j_2}(t_{i+1}).$$

The Peaceman-Rachford scheme requires some further manipulations to express U in terms of $V(t_i)$ and $V(t_{i+1})$; see Mitchell and Griffiths [1980] for the details.

2.11 Two-Dimensional PDE with Mixed Derivatives

Consider now the case where the 2-dimensional PDE (2.58) has a mixed partial derivative,

$$\frac{\partial V}{\partial t} + (\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_{1,2}) V = 0, \quad (2.70)$$

where \mathcal{L}_1 and \mathcal{L}_2 are as in (2.58), and where

$$\mathcal{L}_{1,2} = s_{1,2}(t, x) \frac{\partial^2}{\partial x_1 \partial x_2} \triangleq \rho(t, x) \gamma_1(t, x) \gamma_2(t, x) \frac{\partial^2}{\partial x_1 \partial x_2}. \quad (2.71)$$

The quantity $\rho(t, x)$ is the instantaneous correlation between the processes $x_1(t)$ and $x_2(t)$ in (2.57), i.e. $\rho(t, x) \in [-1, 1]$.

The presence of $\mathcal{L}_{1,2}$ prevents a direct application of the ADI methods in Section 2.10.2, since the mixed operator $\mathcal{L}_{1,2}$ is not amenable to operator splitting. We shall demonstrate two ways to overcome this problem: a) orthogonalization of the PDE; and b) predictor-corrector schemes.

2.11.1 Orthogonalization of the PDE

The idea here is to introduce new variables $y_1(t, x_1, x_2)$ and $y_2(t, x_1, x_2)$ such that the PDE loses its mixed derivative term when stated in terms of these variables. To demonstrate this idea, assume first that $\rho(t, x)$, $\gamma_1(t, x)$, and $\gamma_2(t, x)$ are all functions of time only and independent of x . Then define, say,

$$y_1(t, x_1, x_2) = x_1, \quad (2.72)$$

$$y_2(t, x_1, x_2) = -\rho(t) \frac{\gamma_2(t)}{\gamma_1(t)} x_1 + x_2 \triangleq a(t)x_1 + x_2, \quad (2.73)$$

where we must assume that $\gamma_1(t) \neq 0$ for all t .

Lemma 2.11.1. Consider the PDE (2.70) subject to the terminal value condition $V(T, x) = g(x)$. Define $y = (y_1, y_2)^\top$ and $v(t, y) = V(t, x)$. With the variable change defined in (2.72)–(2.73), v satisfies

$$\begin{aligned} \frac{\partial v}{\partial t} + \mu_1^y(t, y) \frac{\partial v}{\partial y_1} + \mu_2^y(t, y) \frac{\partial v}{\partial y_2} + \frac{1}{2} \gamma_1(t)^2 \frac{\partial^2 v}{\partial y_1^2} \\ + \frac{1}{2} (1 - \rho(t)^2) \gamma_2(t)^2 \frac{\partial^2 v}{\partial y_2^2} - r(t, y_1, y_2 - a(t)y_1) = 0, \end{aligned} \quad (2.74)$$

where

$$\mu_1^y(t, y) \triangleq \mu_1(t, x_1, x_2) = \mu_1(t, y_1, y_2 - a(t)y_1), \quad (2.75)$$

$$\begin{aligned} \mu_2^y(t, y) &\triangleq \frac{da(t)}{dt} x_1 + a(t) \mu_1(t, x_1, x_2) + \mu_2(t, x_1, x_2) \\ &= \frac{da(t)}{dt} y_1 + a(t) \mu_1^y(t, y) + \mu_2(t, y_1, y_2 - a(t)y_1). \end{aligned} \quad (2.76)$$

The equation (2.74) is subject to the terminal value condition $v(T, y_1, y_2) = g(x_1, x_2) = g(y_1, y_2 - a(T)y_1)$.

Proof. While the result can be established by the usual mechanics of ordinary calculus, we will take the opportunity to show how stochastic calculus can

also conveniently prove results of this type. Going back to the processes underlying the PDE (see (2.57)), we write

$$dx_1(t) = \mu_1(t, x) dt + \gamma_1(t) dW_1(t), \quad (2.77)$$

$$dx_2(t) = \mu_2(t, x) dt + \gamma_2(t) \left(\rho(t) dW_1(t) + \sqrt{1 - \rho(t)^2} dW_2(t) \right), \quad (2.78)$$

for independent scalar Brownian motions $W_1(t)$ and $W_2(t)$; this is easily seen to generate the correct correlation $\rho(t)$ between x_1 and x_2 . An application of Ito's lemma then shows that the processes for y_1 and y_2 are

$$\begin{aligned} dy_1(t) &= dx_1(t) = \mu_1(t, x) dt + \gamma_1(t) dW_1(t), \\ dy_2(t) &= \frac{da(t)}{dt} x_1(t) dt + a(t) \mu_1(t, x) dt + a(t) \gamma_1(t) dW_1(t) \\ &\quad + \mu_2(t, x) dt + \gamma_2(t) \left(\rho(t) dW_1(t) + \sqrt{1 - \rho(t)^2} dW_2(t) \right) \\ &= \left(\frac{da(t)}{dt} x_1(t) + a(t) \mu_1(t, x) + \mu_2(t, x) \right) dt \\ &\quad + \gamma_2(t) \sqrt{1 - \rho(t)^2} dW_2(t). \end{aligned}$$

With the definitions (2.75)–(2.76), this becomes simply

$$dy_1(t) = \mu_1^y(t, y(t)) dt + \gamma_1(t) dW_1(t), \quad (2.79)$$

$$dy_2(t) = \mu_2^y(t, y(t)) dt + \gamma_2(t) \sqrt{1 - \rho(t)^2} dW_2(t). \quad (2.80)$$

Equations (2.79)–(2.80) define a Markov SDE in $y_1(t)$ and $y_2(t)$ where, importantly, the Brownian motions on $y_1(t)$ and $y_2(t)$ are now *independent*. Writing $V(t, x) = v(t, y)$, it then follows immediately from the backward Kolmogorov equation (see Section 1.8) that v satisfies the PDE (2.74). \square

Through the chosen transformation (2.72)–(2.73), our original PDE has now been put into a form where we can immediately apply the ADI schemes outlined in Section 2.10.2.

In performing the orthogonalization of the PDE in Lemma 2.11.1 we relied on $\rho(t, x)$, $\gamma_1(t, x)$, and $\gamma_2(t, x)$ all being independent of x . This can often be relaxed. Consider for instance the case where $\rho(t, x) = \rho(t)$, $\gamma_1(t, x) = \gamma_1(t, x_1)$, and $\gamma_2(t, x) = \gamma_2(t, x_2)$; here the correlation ρ is still assumed deterministic, but we now allow for some (though not full) x -dependence in γ_1 and γ_2 . Assuming that $\gamma_1(t, x_1) > 0$ and $\gamma_2(t, x_2) > 0$ we can introduce new variables

$$z_1(t, x_1) = \int \frac{1}{\gamma_1(t, x_1)} dx_1, \quad (2.81)$$

$$z_2(t, x_2) = \int \frac{1}{\gamma_2(t, x_2)} dx_2. \quad (2.82)$$

Applying Ito's lemma to (2.77)–(2.78) we see that

$$dz_1(t, x_1) = \left(- \int \frac{\partial \gamma_1(t, x_1)}{\partial t} \frac{1}{\gamma_1(t, x_1)^2} dx_1 + \frac{\mu_1(t, x)}{\gamma_1(t, x_1)} - \frac{1}{2} \frac{\partial \gamma_1(t, x_1)}{\partial x_1} \right) dt + dW_1(t) \quad (2.83)$$

and

$$dz_2(t, x_2) = \left(- \int \frac{\partial \gamma_2(t, x_2)}{\partial t} \frac{1}{\gamma_2(t, x_2)^2} dx_2 + \frac{\mu_2(t, x)}{\gamma_2(t, x_2)} - \frac{1}{2} \frac{\partial \gamma_2(t, x_2)}{\partial x_2} \right) dt + \rho(t) dW_1(t) + \sqrt{1 - \rho(t)^2} dW_2(t). \quad (2.84)$$

As we assumed that $\gamma_1(t, x_1) > 0$ and $\gamma_2(t, x_2) > 0$, the functions z_1 and z_2 are increasing in x_1 and x_2 , respectively, and are thereby invertible. As such, we can rewrite (2.83)–(2.84) in the more appealing form

$$\begin{aligned} dz_1(t, x_1) &= \mu_1^z(t, z_1, z_2) dt + dW_1(t), \\ dz_2(t, x_1) &= \mu_2^z(t, z_1, z_2) dt + \rho(t) dW_1(t) + \sqrt{1 - \rho(t)^2} dW_2(t). \end{aligned}$$

Through the transformation (2.81)–(2.82), we have reduced our original system to one where the coefficients on $W_1(t)$ and $W_2(t)$ are no longer state-dependent, similar to the case that lead to Lemma 2.11.1. We can now proceed with another variable transformation, as in (2.72)–(2.73), to orthogonalize the system and prepare it for an application of the ADI method.

While the orthogonalization method outlined here can be very effective on a range of practical problems, it suffers from a few drawbacks. Most obviously, the method is not completely general and requires a certain structure on the parameters of the PDE. Another drawback is that the introduction of a time-dependent transformation on one or more variables (Lemma 2.11.1) often makes the alignment of the finite difference grid along (time-independent) critical level points in x -space impossible. Also, the introduction of terms like $y_1 da(t)/dt$ in the drift of y_2 (see (2.76)) can be problematic, particularly if the functions $\gamma_1(t)$ and $\gamma_2(t)$ are not smooth. For instance, it is not unlikely that $y_1 da(t)/dt$ will locally be of such magnitude that upwinding will be necessary to prevent oscillations; see Section 2.6.1. Further, we note that inversion of the transformations (2.81)–(2.82) will not always be possible to perform analytically and may require numerical (root-search) work, complicating the scheme and potentially slowing it down. Finally, as we shall highlight in future chapters, maintaining the “continuity” of a numerical scheme with respect to input parameters is of critical importance for the smoothness of risk sensitivities. Such continuity is difficult to ensure if complicated transformations are applied to model variables. So, in the end, we recommend formulating the PDEs in terms of financially meaningful variables, avoiding excessive transformations, and relying on methods such as developed in the next section when dealing with mixed derivatives and other numerical complications.

2.11.2 Predictor-Corrector Scheme

In this section we shall consider a completely general method for handling mixed derivatives in two-dimensional PDEs. While a bit slower than the method outlined in Section 2.11.1, it does not involve any variable transformations and, by extension, does not suffer from the drawbacks associated with such transformations. As a first step, consider the discretization of the mixed derivative $\partial^2 V / \partial x_1 \partial x_2$. There are a few possibilities (see Mitchell and Griffiths [1980]), but we shall just use

$$\begin{aligned} \delta_{x_1 x_2} V_{j_1, j_2}(t) &= \delta_{x_1} \delta_{x_2} V_{j_1, j_2}(t) \\ &= \frac{V_{j_1+1, j_2+1}(t) - V_{j_1+1, j_2-1}(t) - V_{j_1-1, j_2+1}(t) + V_{j_1-1, j_2-1}(t)}{4\Delta_1 \Delta_2}. \end{aligned} \quad (2.85)$$

Extensions to non-equidistant grids follow directly from (2.27) and the relation $\delta_{x_1 x_2} V_{j_1, j_2}(t) = \delta_{x_1} \delta_{x_2} V_{j_1, j_2}(t)$. As we have not encountered mixed difference operators before, for completeness we show the following lemma.

Lemma 2.11.2. *For the discrete operator (2.85) we have*

$$\delta_{x_1 x_2} V_{j_1, j_2}(t) = \frac{\partial^2 V(t, x_1^{j_1}, x_2^{j_2})}{\partial x_1 \partial x_2} + O(\Delta_1^2 + \Delta_2^2).$$

Proof. A Taylor expansion of $V(t, x)$ around the point $x = (x_1^{j_1}, x_2^{j_2})^\top$ gives

$$\begin{aligned} V_{j_1+1, j_2 \pm 1}(t) &= V_{j_1, j_2}(t) + \Delta_1 \frac{\partial V}{\partial x_1} \pm \Delta_2 \frac{\partial V}{\partial x_2} + \frac{1}{2} \Delta_1^2 \frac{\partial^2 V}{\partial x_1^2} + \frac{1}{2} \Delta_2^2 \frac{\partial^2 V}{\partial x_2^2} \\ &\quad \pm \Delta_1 \Delta_2 \frac{\partial^2 V}{\partial x_1 \partial x_2} + \frac{1}{6} \Delta_1^3 \frac{\partial^3 V}{\partial x_1^3} \pm \frac{1}{6} \Delta_2^3 \frac{\partial^3 V}{\partial x_2^3} \\ &\quad + \frac{1}{2} \Delta_1 \Delta_2^2 \frac{\partial^3 V}{\partial x_1 \partial x_2^2} \pm \frac{1}{2} \Delta_1^2 \Delta_2 \frac{\partial^3 V}{\partial x_1^2 \partial x_2} + \dots, \\ V_{j_1-1, j_2 \pm 1}(t) &= V_{j_1, j_2}(t) - \Delta_1 \frac{\partial V}{\partial x_1} \pm \Delta_2 \frac{\partial V}{\partial x_2} + \frac{1}{2} \Delta_1^2 \frac{\partial^2 V}{\partial x_1^2} + \frac{1}{2} \Delta_2^2 \frac{\partial^2 V}{\partial x_2^2} \\ &\quad \mp \Delta_1 \Delta_2 \frac{\partial^2 V}{\partial x_1 \partial x_2} - \frac{1}{6} \Delta_1^3 \frac{\partial^3 V}{\partial x_1^3} \pm \frac{1}{6} \Delta_2^3 \frac{\partial^3 V}{\partial x_2^3} \\ &\quad - \frac{1}{2} \Delta_1 \Delta_2^2 \frac{\partial^3 V}{\partial x_1 \partial x_2^2} \pm \frac{1}{2} \Delta_1^2 \Delta_2 \frac{\partial^3 V}{\partial x_1^2 \partial x_2} + \dots. \end{aligned}$$

A little thought then shows that

$$\begin{aligned} V_{j_1+1, j_2+1}(t) - V_{j_1+1, j_2-1}(t) - V_{j_1-1, j_2+1}(t) + V_{j_1-1, j_2-1}(t) \\ = 4\Delta_1 \Delta_2 \frac{\partial^2 V}{\partial x_1 \partial x_2} + O(\Delta_1^3 \Delta_2 + \Delta_1 \Delta_2^3), \end{aligned}$$

as error terms of order Δ_1^4 , Δ_2^4 , and $\Delta_1^2 \Delta_2^2$ will cancel. The result follows.

□

Equipped with (2.85), we can approximate the operator $\mathcal{L}_{1,2}$ in (2.71) as

$$\widehat{\mathcal{L}}_{1,2} V_{j_1,j_2}(t) \triangleq \rho(t, x_1^{j_1}, x_2^{j_2}) \gamma_1(t, x_1^{j_1}, x_2^{j_2}) \gamma_2(t, x_1^{j_1}, x_2^{j_2}) \delta_{x_1 x_2} V_{j_1,j_2}(t),$$

which is accurate to order $O(\Delta_1^2 + \Delta_2^2)$. The first easy way to modify our ADI scheme to incorporate $\widehat{\mathcal{L}}_{1,2}$ is to treat the mixed derivative fully explicitly. In the Douglas-Rachford scheme (2.68)–(2.69), for instance, we thus modify the right-hand side of the first step as follows:

$$(1 - \theta \Delta_t \widehat{\mathcal{L}}_1) U_{j_1,j_2} = (1 + (1 - \theta) \Delta_t \widehat{\mathcal{L}}_1 + \Delta_t \widehat{\mathcal{L}}_2 + \Delta_t \widehat{\mathcal{L}}_{1,2}) \widehat{V}_{j_1,j_2}(t_{i+1}), \quad (2.86)$$

$$(1 - \theta \Delta_t \widehat{\mathcal{L}}_2) \widehat{V}_{j_1,j_2}(t_i) = U_{j_1,j_2} - \theta \Delta_t \widehat{\mathcal{L}}_2 \widehat{V}_{j_1,j_2}(t_{i+1}). \quad (2.87)$$

The addition of $\widehat{\mathcal{L}}_{1,2}$ this way clearly preserves the ADI structure of the scheme which will continue to involve only sequences of tri-diagonal linear equations. However, having, in effect, only a one-sided time-differencing of the mixed derivative term will lower the convergence order of the time step to $O(\Delta_t)$, irrespective of the choice of θ .

To change the time at which the mixed operator $\widehat{\mathcal{L}}_{1,2}$ is evaluated, consider using a *predictor-corrector* scheme, where the results of (2.86)–(2.87) are re-used in a one-time²² iteration. Specifically, we write, for some $\lambda \in [0, 1]$,

Predictor:

$$(1 - \theta \Delta_t \widehat{\mathcal{L}}_1) U_{j_1,j_2}^{(1)} = (1 + (1 - \theta) \Delta_t \widehat{\mathcal{L}}_1 + \Delta_t \widehat{\mathcal{L}}_2 + \Delta_t \widehat{\mathcal{L}}_{1,2}) \widehat{V}_{j_1,j_2}(t_{i+1}), \quad (2.88)$$

$$(1 - \theta \Delta_t \widehat{\mathcal{L}}_2) U_{j_1,j_2}^{(2)} = U_{j_1,j_2}^{(1)} - \theta \Delta_t \widehat{\mathcal{L}}_2 \widehat{V}_{j_1,j_2}(t_{i+1}). \quad (2.89)$$

Corrector:

$$(1 - \theta \Delta_t \widehat{\mathcal{L}}_1) Z_{j_1,j_2}^{(1)} = (1 + (1 - \theta) \Delta_t \widehat{\mathcal{L}}_1 + \Delta_t \widehat{\mathcal{L}}_2 + (1 - \lambda) \Delta_t \widehat{\mathcal{L}}_{1,2}) \widehat{V}_{j_1,j_2}(t_{i+1}) + \lambda \Delta_t \widehat{\mathcal{L}}_{1,2} U_{j_1,j_2}^{(2)}, \quad (2.90)$$

$$(1 - \theta \Delta_t \widehat{\mathcal{L}}_2) \widehat{V}_{j_1,j_2}(t_i) = Z_{j_1,j_2}^{(1)} - \theta \Delta_t \widehat{\mathcal{L}}_2 \widehat{V}_{j_1,j_2}(t_{i+1}). \quad (2.91)$$

²²We can run the iteration more than once if desired, but a single iteration will normally suffice.

Notice how the Douglas-Rachford scheme is first run once, in (2.88)–(2.89), to yield a first guess (a “predictor”), $U_{j_1,j_2}^{(2)}$, for the time t_i value $V_{j_1,j_2}(t_i)$. In a second run of the Douglas-Rachford scheme, in (2.90)–(2.91), this guess is used as a “corrector” to affect the time at which $\hat{\mathcal{L}}_{1,2}$ is evaluated, by applying this operator to $(1 - \lambda)\hat{V}_{j_1,j_2}(t_{i+1}) + \lambda U_{j_1,j_2}^{(2)}$; when $\lambda = \frac{1}{2}$ we effectively center the time-differencing of the mixed term. The scheme now relies on three intermediate variables, $U_{j_1,j_2}^{(1)}$, $U_{j_1,j_2}^{(2)}$, and $Z_{j_1,j_2}^{(1)}$.

The combined predictor-corrector scheme above (in a slightly less general form, with $\Delta_1 = \Delta_2$) was suggested by Craig and Sneyd [1988]. It can be shown that the scheme has convergence order

$$O\left((\Delta_1 + \Delta_2)^2 + 1_{\{\theta \neq \frac{1}{2}\}}\Delta_t + 1_{\{\lambda \neq \frac{1}{2}\}}\Delta_t + \Delta_t^2\right),$$

so second order convergence in the time domain is still achievable by setting $\theta = \lambda = \frac{1}{2}$. The scheme will be A -stable for $\theta \geq \frac{1}{2}$ and $\frac{1}{2} \leq \lambda \leq \theta$. The computational cost of the predictor-corrector is clearly still $O(m_1 m_2)$ per time step, as both the predictor and corrector schemes have $O(m_1 m_2)$ cost per time-step. Even though the standard Douglas-Rachford scheme is effectively run twice, we should point out that when intelligently implemented, (2.88)–(2.91) is typically only about 30-40% slower than the Douglas-Rachford scheme, as a number of results from the predictor step can be cached and reused in the corrector step.

As for the standard ADI grids, extensions to non-equidistant grids are straightforward using the techniques in Section 2.4. Boundary conditions in the x -domain are imposed along the lines outlined in Section 2.10.3.

2.12 PDEs of Arbitrary Order

We now turn our attention back to the general p -dimensional PDE (2.56). To prepare for a numerical scheme, let us rewrite the PDE as follows:

$$\frac{\partial V}{\partial t} + \sum_{h=1}^p \mathcal{L}_h V + \sum_{h=1}^p \sum_{l=h+1}^p \mathcal{L}_{h,l} V = 0, \quad (2.92)$$

where

$$\begin{aligned} \mathcal{L}_h &= \mu_h(t, x) \frac{\partial}{\partial x_h} + \frac{1}{2} s_{h,h}(t, x) \frac{\partial^2}{\partial x_h^2} - p^{-1} r(t, x), \\ \mathcal{L}_{h,l} &= s_{h,l}(t, x) \frac{\partial^2}{\partial x_h \partial x_l}. \end{aligned}$$

The method we present here for solution of (2.92) is a p -dimensional version of the predictor-corrector scheme outlined above. The extension

is straightforward and we simply list it here without further discussion; see Craig and Sneyd [1988] for additional background. To simplify notation, we have omitted sub-indices everywhere (i.e., $\widehat{V}(t_i)$ is used instead of $\widehat{V}_{j_1, j_2, \dots, j_p}(t_i)$).

Predictor:

$$\begin{aligned} & \left(1 - \theta \Delta_t \widehat{\mathcal{L}}_1\right) U^{(1)} \\ &= \Delta_t \left(\Delta_t^{-1} + (1 - \theta) \widehat{\mathcal{L}}_1 + \sum_{h=2}^p \widehat{\mathcal{L}}_h + \sum_{h=1}^p \sum_{l=h+1}^p \widehat{\mathcal{L}}_{h,l} \right) \widehat{V}(t_{i+1}), \\ & \left(1 - \theta \Delta_t \widehat{\mathcal{L}}_2\right) U^{(2)} = U^{(1)} - \theta \Delta_t \widehat{\mathcal{L}}_2 \widehat{V}(t_{i+1}), \\ & \quad \vdots \\ & \left(1 - \theta \Delta_t \widehat{\mathcal{L}}_p\right) U^{(p)} = U^{(p-1)} - \theta \Delta_t \widehat{\mathcal{L}}_p \widehat{V}(t_{i+1}). \end{aligned}$$

Corrector:

$$\begin{aligned} & \left(1 - \theta \Delta_t \widehat{\mathcal{L}}_1\right) Z^{(1)} \\ &= \Delta_t \left(\Delta_t^{-1} + (1 - \theta) \widehat{\mathcal{L}}_1 + \sum_{h=2}^p \widehat{\mathcal{L}}_h \right. \\ & \quad \left. + (1 - \lambda) \sum_{h=1}^p \sum_{l=h+1}^p \widehat{\mathcal{L}}_{h,l} \right) \widehat{V}(t_{i+1}) + \lambda \Delta_t \sum_{h=1}^p \sum_{l=h+1}^p \widehat{\mathcal{L}}_{h,l} U^{(p)}, \\ & \left(1 - \theta \Delta_t \widehat{\mathcal{L}}_2\right) Z^{(2)} = Z^{(1)} - \theta \Delta_t \widehat{\mathcal{L}}_2 \widehat{V}(t_{i+1}), \\ & \quad \vdots \\ & \left(1 - \theta \Delta_t \widehat{\mathcal{L}}_p\right) \widehat{V}(t_i) = Z^{(p-1)} - \theta \Delta_t \widehat{\mathcal{L}}_p \widehat{V}(t_{i+1}). \end{aligned}$$

With m_h points in the x_h -direction, $h = 1, \dots, p$, the computational cost of the predictor-corrector scheme is $O(\prod_{h=1}^p m_h)$. For $p \leq 3$, sufficient conditions for A -stability are $\theta \geq \frac{1}{2}$ and $\frac{1}{2} \leq \lambda \leq \theta$. For $p \geq 4$, sufficient conditions are $\theta \leq \frac{1}{2}$ and

$$\frac{1}{2} \leq \lambda \leq \frac{p^{p-1}}{(p-1)^p} \theta.$$

See Craig and Sneyd [1988] for a proof. Convergence is similar to the two-dimensional case.

As a final comment, let us note that as dimensionality increases, the computational complexity of an iterative sparse solver will start approaching that of ADI. Specifically, for a p -dimensional problem, the complexity of the former is $O(m_{\text{total}})$ and for the latter $O(m_{\text{total}}^{(2p+1)/2p})$, with $m_{\text{total}} = m_1 \cdot m_2 \cdot \dots \cdot m_p$.