Numerical Methods for Finance – Finite Differences

Christoph Reisinger
christoph.reisinger@maths.ox.ac.uk

Michaelmas Term 2015
(version 07/12/15)

Chapter 1

Introduction

1.1 Mathematical models for financial derivatives

In the simplest setting, consider a contingent claim on a risky asset S, whose value S_T at time $t = T$ is unknow. The holder of this derivative recieves a payoff $g(S_T)$ at expiry T (*European option*).

Generally, valuation of such a contract requires modelling assumptions on the underlying asset.

The class of models we will study in the first instance is one where the option price is the *expected value* of the stock under a process of the form

$$\mathrm{d}S = rS\,\mathrm{d}t + a(S,t)\,\mathrm{d}W, \tag{1.1}$$

where rS is the return on a risk free investment and $\mathrm{d}W$ a Brownian increment. Future cashflows have to be discounted back to today, with the risk-free interest rate, that is

$$u(S,t) = \mathbb{E}\left(\mathrm{e}^{-r(T-t)} g(S_T) | S_t = S\right).$$

The partial derivatives, which are important trading parameters,

$$\frac{\partial u}{\partial S}, \frac{\partial^2 u}{\partial S^2}, \frac{\partial u}{\partial t}$$

are related by the *Feynman-Kac PDE*

$$\frac{\partial u}{\partial t} + \frac{1}{2}a^2(S,t)\frac{\partial^2 u}{\partial S^2} + rS\frac{\partial u}{\partial S} - ru = 0.$$

This PDE is complemented with the terminal condition

$$u(S,T) = g(S),$$

which states that the payoff at $t = T$ is g.

Examples for more complex derivatives include those that can be *exercised* prior to expiry (*American options*), or that depend on values of the stock at points in time throughout the lifetime of the contract (*path-dependent options*).

1.2 Workflow in computational finance

1. Identify a suitable model for the underlying asset.
2. Find fast and accurate approximations to simple derivative contracts.
3. Invert these relations to imply model parameters from quoted market prices.
4. Develop a numerical scheme suitable for pricing more complex contracts.
5. Test the numerical scheme for the simple products where the solution is known.
6. Stress-test the model and pricing method under different market scenarios.
7. Compute hedge parameters.
8. Use the above setup to compute the (unknown!) prices of complex derivatives.

1.3 Types of calculation

Solution methods often fall into one of the following main categories:

- semi-analytic, if distributions are known analytically or semi-analytically, e.g. via their characteristic function;
- estimation of the expected pay-off by simulation;
- numerical approximation of a corresonding PDE or deterministic integral.

If the terminal probability density is known, the (discounted) expectation can be evaluated as an integral

$$u(S,0) = \int_0^\infty \mathrm{e}^{-rT} g(S') p(S,0;S',T)\,\mathrm{d}S', \tag{1.2}$$

where $p(S,0;S',T)$ is the transition probability to go from S at $t=0$ to S' at $t=T$.

In some cases, like the Black-Scholes problem, integration can be carried out analytically, which results in closed-form solutions. If this is not the case, approximation of (1.2) via a *quadrature rule*, e.g. the trapezoidal rule, gives

$$u_N(S,0) = \frac{1}{N}\sum_{k=0}^{N} \alpha_k \mathrm{e}^{-rT} g(s_k) p(S,0;s_k,T)$$

with integration points s_k, $k=0,\ldots,N$, and weights α_k. Note that if the pay-off and density have infinite support this also means truncating the integration interval.

In a *Monte Carlo* method, the expectation is approximated by the average over a (finite) random sample $S^{(i)}$, $i=1,\ldots,N$, drawn independently from the distribution with density $p(S,0;S',T)$,

$$u_N(S,0) = \frac{1}{N}\mathrm{e}^{-rT}\sum_{i=1}^{N} g(S^{(i)}).$$

If the terminal distribution is not known, but given as the solution of a stochastic differential equation (1.1), one can still approximate the paths via a Monte Carlo method, e. g. the *Euler scheme*

$$S_{k+1}^{(i)} - S_k^{(i)} = rS_k^{(i)}\tau + a(S_k^{(i)}, t)\sqrt{\tau}X_k^{(i)}$$

where $X_k^{(i)} \sim N(0,1)$ is a standard normal, τ the timestep.

If we choose to solve the equivalent PDE analytically or numerically, we are faced with a final value problem: the value of the option at expiry is known to be $g(S)$, the PDE is solved backwards from expiry.

1.4 The main methods compared

- The solution to the PDE is the value function for all possible values of the underlying. This makes it straightforward to calculate sensitivities with respect to the underlying.
- The downside is that we need to calculate the solution for a large number of values for the underlying. This raises severe complexity issues in higher dimensions.
- Monte Carlo path simulation with the current spot price as starting point yields the option price for a single underlying value. Accurate and stable sensitivity calculation is more involved.
- Monte Carlo is based on forward simulation and allows us to 'look back' on the path. This is useful when the option depends on the whole path of the stock, e. g. *Asian options* on the average of the price over the lifetime of the option.
- In contrast to this, PDE methods are based on backward induction. This is advantageous when the option contract allows early exercise (*American options*), as we can 'look forward' and take the optimal decision over all different scenarios.

Chapter 2

From random walks to finite differences and back

This introductory chapter is concerned with random walks which move between a discrete set of points, it studies their transition probabilities and quantities derived from those, for instance the expected value of a function of their final state. We will see that if we shrink the distance between points and compensate by moving at shorter time intervals, the process approaches *Brownian motion*, a stochastic process in "continuous time". In this limit, its transition density is governed by the *heat equation*, a partial differential equation (PDE).

We then turn the objective around and pose questions of the kind: given this PDE, can we use the discrete model to devise a computationally tractable method to approximate the solution to the PDE; what determines the accuracy of this approximation; which properties of the continuous model are preserved by its discrete approximation. These questions form the backbone of numerical analysis and will be a central topic of the first part of this book. This perspective is relevant in financial engineering because models are usually elegantly formulated in continuous time, without having first looked at a discrete version. It should be added that even in cases where there is a finacially meaningful discrete process in the backgound, there are often more accurate discretisation techniques which give faster solutions to the PDE than the discrete model which may have motivated the PDE in the first place. A probabilistic interpretation of numerical methods can non-the-less give valuable insights in the properties of numerical schemes.

2.1 Random walks and the heat equation

2.1.1 A symmetric random walk

Consider a process (a "marker") which performs a random walk on the real line, and whose position X_t at time t evolves as follows. At time $t_0 = 0$, it starts off at $X_0 = 0$. At equally spaced points in time t_m, taken out of $\{m\Delta t : m \in \mathbb{N}\}$ with intervals $\Delta t > 0$, the marker moves left or right an amount $\Delta x > 0$ with equal probability $0 \leq p \leq 1/2$, or stays put with probability $1 - 2p$. We assume that each move is independent of previous moves. So if we

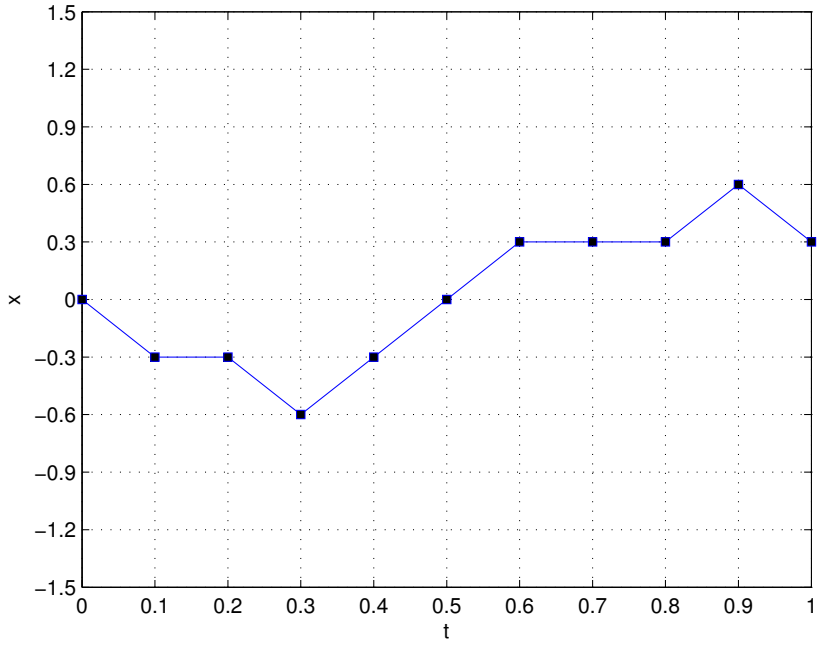

Figure 2.1: A realisation of X_t for $t \in [0,1]$, $\Delta x = 0.3$, $\Delta t = 0.1$, $p = 1/3$.

define

$$\xi_m = \begin{cases} 1 & \text{with probability } p \\ -1 & \text{with probability } p \\ 0 & \text{with probability } 1-2p \end{cases} \tag{2.1}$$

the independent directions of travel, the posistion at t_m is

$$X_{t_m+\Delta t} = X_{t_m} + \xi_m \Delta x$$

and lies on a grid $\{n\Delta x : n \in \mathbb{Z}\}$. In fact, the range of possible positions is restricted by

$$X_{m\Delta t} = \sum_{k=0}^{m-1} \xi_k \Delta x \in \{-m\Delta x, \ldots, 0, \ldots, m\Delta x\}. \tag{2.2}$$

Between these times, we assume the marker moves uniformly to its new position,

$$X_t = X_{t_m} + \frac{t - t_m}{\Delta t}(X_{t_m+\Delta t} - X_{t_m}), \quad t \in [t_m, t_{m+1}].$$

Fig. 2.1 shows a possible realisation of such a path.

As $\mathbb{E}(\xi_m) = 0$, from (2.2) follows

$$\mathbb{E}(X_m) = 0.$$

As the ξ_m are independent and identically distributed (i.i.d.),

$$Var(X_{m\Delta t}) = m\Delta x^2 Var(\xi_0) = m\,\Delta x^2\, 2p. \tag{2.3}$$

The variance of the process at a time t is intuitive determined by the number of steps m, the size of a step Δx and the probability $2p$ of a non-zero move. We "standardize" the process to have unit variance per unit time,

$$Var(X_t) = t, \quad t/\Delta t \in \mathbb{N}.$$

This is the case exactly if $Var(X_{\Delta t}) = \Delta t$, which together with (2.3) requires the relation

$$p = \frac{1}{2}\frac{\Delta t}{\Delta x^2}. \tag{2.4}$$

We need $p \leq 1/2$ for (2.1) to make sense, which restricts the range of allowable Δt and Δx^2, specifically $\Delta t \leq \Delta x^2$. This is reflection of the fact that if the number of timesteps is too small relative to the (square of) the grid size, we cannot achieve a prescribed variance. If we do have (more than) enough timesteps, we can adjust the probability of moves to ensure the variance is exactly matched.

2.1.2 Transition probabilities

We now move on to find the probability that the marker is found at a specific point at a given time.

The process starts at $X_0 = 0$, such that

$$\mathbb{P}(X_0 = 0) = 1, \tag{2.5}$$

while at time t, from (2.2),

$$\mathbb{P}(X_{m\Delta t} < -m\Delta x) = \mathbb{P}(X_{m\Delta t} > m\Delta x) = 0. \tag{2.6}$$

Denote by U_n^m the probability that the marker is at position $x_n = n\Delta x$ at time $t_m = m\Delta t$,

$$U_n^m = \mathbb{P}(X_{t_m} = x_n).$$

This discrete probability density can be computed directly by considering all paths leading to a certain point, and the probability of following each of these paths. This is left as an exercise (Exercise 1) and we follow a different tack here.

By partitioning with respect to the position at the current time t_m,

$$\mathbb{P}(X_{t_{m+1}} = x_n | X_{t_m} = x_k) = \begin{cases} p & n = k \pm 1, \\ 1 - 2p & n = k, \end{cases}$$

and therefore

$$\begin{aligned} U_n^{m+1} &= \sum_{k=-\infty}^{\infty} \mathbb{P}(X_{t_{m+1}} = x_n | X_{t_m} = x_k)\, \mathbb{P}(X_{t_m} = x_k) \\ &= p \cdot U_{n+1}^m + (1-2p) \cdot U_k^m + p \cdot U_{n-1}^m. \end{aligned} \tag{2.7}$$

This defines an induction by which all values U_n^{m+1}, $n \in \mathbb{Z}$, are defined by – and can be explicitly calculated from – the values of U_n^m. The initial condition for this induction is seen from (2.5) as

$$U_n^0 = \delta_{n0} = \begin{cases} 1 & n = 0, \\ 0 & n \neq 0. \end{cases} \tag{2.8}$$

Because of (2.6), only finitely many U_n^m are non-zero for fixed m and need to be computed.

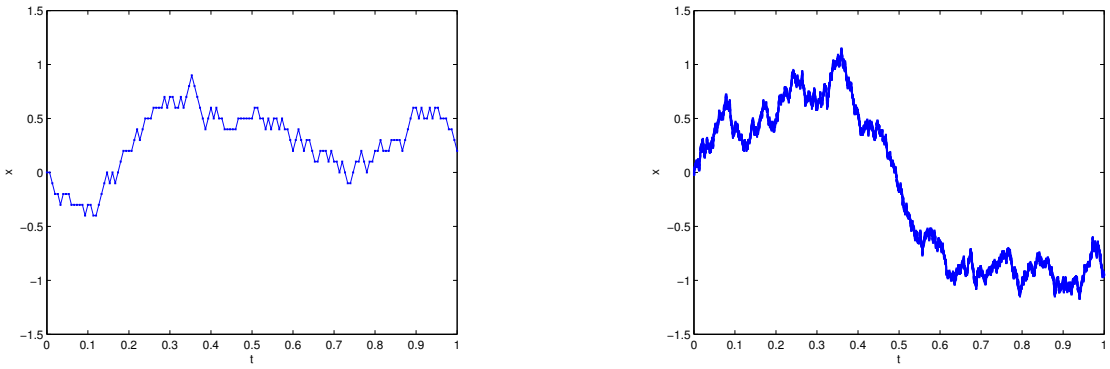

Figure 2.2: Realisation of X_t for $t \in [0,1]$, $\Delta x = 0.1$ and $\Delta x = 0.01$ respectively, $p = 1/3$ and $\Delta t = 2p\Delta x^2$ ensuring unit variance.

2.1.3 The continuous-time limit

Now fix t and let $m \to \infty$ with $\Delta t = t/m$, $0 < p \leq 1$ fixed and $\Delta x = \sqrt{\Delta t/2p}$ as in (2.4). Examples of paths are plotted in Fig. 2.2. We make the dependence on Δt explicit by writing $X_t = X_t^{\Delta t}$. By the assumption that ξ_k are i.i.d. and such that $Var(X_t^{\Delta t}) = t$, it follows from (2.2) by virtue of the Central Limit Theorem (see eg [Grimmet and Stirzaker, 2001]) that

$$X_t^{\Delta t} \to W_t \sim N(0,t).$$

Here $N(0,t)$ denotes the normal distribution with mean 0 and variance t. The density of W_t is therefore given by the normal probability density function

$$u(x,t) = \frac{1}{\sqrt{2\pi t}} \mathrm{e}^{-x^2/2t}. \tag{2.9}$$

Convergence is in distribution, [Grimmet and Stirzaker, 2001].

It can indeed be shown that the limiting stochastic process W_t has the following intuitive properties:

- $W_0 = 0$;
- for any finite set of times $0 \leq t_1 < t_2 \cdots < t_m$,

  $$W_{t_2} - W_{t_1},\ W_{t_3} - W_{t_2},\ \ldots, W_{t_m} - W_{t_{m-1}}$$

  are independent;
- for any $0 \leq s \leq t$,

  $$W_t - W_s \sim N(0, t-s);$$

- W_t is continuous in t with probability 1.

These are the defining properties of *standard Brownian motion*. See any of [Steele, 2001, Shreve, 2004].

We now move on to investigate whether there is a continuous-time limit to (2.7) which the continuous probability density (2.9) must satisfy. Inserting (2.4) in (2.7), and rearranging,

$$\frac{U_n^{m+1} - U_n^m}{\Delta t} = \frac{1}{2}\frac{U_{n+1}^m - 2U_n^m + U_{n-1}^m}{\Delta x^2}. \tag{2.10}$$

Motivated by (2.9), we assume that for small Δt the density can be approximated by the smooth function u. We take into account the appropriate scaling with Δx and set

$$u_n^m = \frac{1}{\Delta x} U_n^m,$$

such that u_n^m becomes interpretable as

$$u_n^m \, \Delta x \approx \int_{x_n - \Delta x/2}^{x_n + \Delta x/2} u(x, t_m) \, \mathrm{d}x.$$

Note that u_n^m also satisfies (2.10) because of linearity. By Taylor expansion, for smooth enough u,

$$u(x, t + \Delta t) = u(x,t) + \Delta t u_t(x,t) + o(\Delta t), \tag{2.11}$$

$$u(x \pm \Delta x, t) = u(x,t) \pm \Delta x u_x(x,t) + \frac{1}{2}\Delta x^2 u_{xx}(x,t) + o(\Delta x^2). \tag{2.12}$$

We use order notation for a small parameter h,

$$f(h) = O(g(h)) \quad :\Leftrightarrow \quad \limsup_{h \to 0} \frac{f(h)}{g(h)} < \infty$$

(f goes to 0 "at least as fast as" g), and

$$f(h) = o(g(h)) \quad :\Leftrightarrow \quad \limsup_{h \to 0} \frac{f(h)}{g(h)} = 0$$

(f goes to 0 "faster than" g), where $\limsup_{h\to 0} g(h) = \lim_{h \to 0} \sup_{0 < k \leq h} g(k)$.

From (2.11) and (2.12), by insertion,

$$\delta_t^+ u = \frac{u(x, t + \Delta t) - u(x,t)}{\Delta t} \to \frac{\partial u}{\partial t} \qquad \text{as } \Delta t \to 0,$$

$$\delta_x^2 u = \frac{u(x + \Delta x, t) - 2u(x,t) + u(x - \Delta x, t)}{\Delta x^2} \to \frac{\partial^2 u}{\partial x^2} \qquad \text{as } \Delta x \to 0.$$

We conjecture that u_n^m can be approximated by $u(x_n, t_m)$, then identifying terms with (2.10), letting again $\Delta x, \Delta t \to 0$, one approaches formally

$$\frac{\partial u}{\partial t} = \frac{1}{2}\frac{\partial^2 u}{\partial x^2}. \tag{2.13}$$

So in some sense the discrete inductive formula approaches the heat equation. We will make this notion more precise in the next section. For now, one verifies easily by insertion that (2.9) indeed solves (2.13). Taking $t \to 0$ in (2.9),

$$u(x, 0) = \delta(x), \tag{2.14}$$

the *Dirac delta distribution*. The marker starts from 0 with probability 1. One also checks easily that

$$u_0(x) = \left\{ \begin{array}{ll} \frac{1}{\Delta x} & x \in [-\Delta x/2, \Delta x/2] \\ 0 & \text{else} \end{array} \right\} \longrightarrow \delta(x) \qquad \text{for } \Delta x \to 0,$$

confirming $u_n^0 = U_n^0/\Delta x = \delta_{n0}/\Delta x$ as a reasonable approximation. We summarize this as

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} = \frac{1}{2}\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\Delta x^2}, \quad m \geq 0 \tag{2.15}$$

$$u_n^0 = \frac{1}{\Delta x}\delta_{n0} = \left\{ \begin{array}{ll} \Delta x^{-1} & n = 0, \\ 0 & \text{otherwise.} \end{array} \right. \tag{2.16}$$

This scheme can be used as an implementable numerical method to approxiamte the solution to the heaat equation, as detailed in the next section.

2.2 The explicit (forward) Euler scheme

2.2.1 Definition of the scheme

We saw in 2.1.3 that in the "continuous-time limit" the symmetric random walk approaches Brownian motion, whose transition probability, the normal distribution, solves the heat equation. Conversely, retracing the steps, the solution to the heat equation approximately satisfies the discrete inductive scheme (2.15). More precisely, if u solves the heat equation, one sees

$$\frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} = \frac{1}{2}\frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} + o(1) \tag{2.17}$$

where $o(1)$ is a term that goes to zero if Δt and Δx do. In fact, taking the Taylor expansions further – by one order in (2.11), and two in (2.12) – the term can be seen to be of order $O(\Delta t, \Delta x^2)$ if u is sufficiently smooth to allow this analysis. This opens up the opportunity of using (2.15) as a numerical scheme to approximate the solution to the heat equation.

Let u_n^m denote the numerical approximation to the solution at x_n, t_m, where $x_n = n\Delta x$, $n \in \mathbb{Z}$, and $t_m = m\Delta t$, $m = 0, \ldots, M$ with $M\Delta t = T$. Motivated by (2.17), one defines the *explicit Euler scheme* as

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} = \frac{1}{2}\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\Delta x^2}, \tag{2.18}$$

which we write shorthand as

$$\delta_t^+ u_n^m = \delta_x^2 u_n^m.$$

We call this a *finite difference scheme*, because the (partial) derivatives of the heat equation have been replaced by "finite differences" in the coordinate directions. This step allows the computation of approximations to the PDE solution by computation of a finite number of values, which, even if the number of equations can be large if a fine grid is used, is implemetable on a computer.

Rewriting (2.18) as

$$u_n^{m+1} = pu_{n+1}^m + (1 - 2p)u_n^m + pu_{n-1}^m \tag{2.19}$$

with $2p = \Delta t/\Delta x^2$, we see why the scheme is called *explicit*: the value at a desired time, say t_{m+1}, can be explicitly computed from values at time t_m using (2.18), and hence inductively from initial data at t_0.

To define the scheme fully, we have to specify inital conditions for u_n^0, most naturally as in (2.16).

2.2.2 Key properties

If $2p \in [0,1]$, ie

$$\Delta t \leq \Delta x^2,$$

u_n^{m+1} defined by (2.19) is a weighted average of u_{n+1}^m, u_n^m, u_{n-1}^m, the solution in an immediate neighbourhood at the previous timestep. It follows

$$\min_{k\in\mathbb{Z}} u_k^m \leq \min\{u_{n-1}^m, u_n^m, u_{n+1}^m\} \leq \underbrace{pu_{n+1}^m + (1-2p)u_n^m + pu_{n-1}^m}_{u_n^{m+1}} \leq \max\{u_{n-1}^m, u_n^m, u_{n+1}^m\} \leq \max_{k\in\mathbb{Z}} u_k^m.$$

Inductively, for all m

$$\min_{k\in\mathbb{Z}} u_k^0 \leq u_n^m \leq \max_{k\in\mathbb{Z}} u_k^0$$

We can say the numerical solution satisfies a *discrete maximum principle*. This is the discrete analogue of the maximum principle obeyed by the heat equation and other parabolic initial-value problems. The solution remains bounded from below and above, which is a measure of stability of the scheme as a numerical method.

Moreover, the solution is *monotone*, i.e. if $u^m = (\ldots, u_{-1}^m, u_0^m, u_1^m \ldots)$ and v^m are finite difference solutions to the initial conditions u^0 and v^0 with $u^0 \geq v^0$, then $u^m \geq v^m$ for all m. In particular, $u^m \geq 0$ if $u^0 \geq 0$. Lastly,

$$\sum_{n\in\mathbb{Z}} u_n^{m+1} = \sum_{n\in\mathbb{Z}} pu_{n+1}^m + (1-2p)u_n^m + pu_{n-1}^m = \sum_{n\in\mathbb{Z}} u_n^m,$$

so if u^0 has the properties of a discrete probability density – non-negativity and correct scaling – then so too has u^m for all m.

2.2.3 Nodes and trees

Note from the construction that $u_n^m = 0$ for $n < -m$ and $n > m$: the underlying random walk stays in a range $-m\Delta t \leq x \leq m\Delta t$ with probability one. Therefore, it is sufficient to compute u_n^m in the range $n \in \{-m, \ldots, m\} \subset \{-M, \ldots, M\}$. This motivates the name *trinomial tree*: every parent *node* (x_n, t_m) has three children nodes $(x_{n-1}, t_{m+1}), (x_n, t_{m+1}), (x_{n+1}, t_{m+1})$, the states the walk can move to in the next timestep. The *root* of the tree is $(0,0)$. For $p = 1/2$, the probability for a move from (x_n, t_m) to (x_n, t_{m-1}), i.e. not moving, is zero. The tree collapses to a *binomial tree* with only two possible children $(x_{n-1}, t_{m+1}), (x_{n+1}, t_{m+1})$.

The *support* of u^m, i.e. the range of gridpoints x_n, where u^m has non-zero value, is $[-m\Delta x, m\Delta x]$ where $m\Delta t = t$ and

$$m\Delta x = m\sqrt{\Delta t/2p} = \sqrt{m/2p}\sqrt{t}. \tag{2.20}$$

This is in contrast to the continuous problem where the marker can move to any position in infinitesimally small time, albeit with low probability if that position is far away. The

continuous model therefore has infinite *speed of propagation*, whereas the discrete model has finite speed of propagation. The underlying Brownian motion is normally distributed with variance t and standard deviation $\sqrt{t}$. The relevant range of points which the continuous process is likely to visit should therefore increase proportional to $\sqrt{t}$. This is also seen from the scaling invariance of the solution to the heat equation in (2.9), $u(x,t) = 1/\sqrt{t}\, u(x/\sqrt{t}, 1)$.

Either way, the range of x with non-negligible $u(x,t)$ for a given t depends on $\sqrt{t}$ alone, certainly not on any parameters of the discretisation. For small timesteps Δt, the range imposed by (2.20) is unnecessarily large if an approximate solution is the goal. From a practical perspective, we want to keep the number of nodes to a minimum for computational efficiency.

For now, choose a value x_{max} large enough, as measured in units of standard deviations, to ensure that the approximation $u(-x_{max}, t) = u(x_{max}, t) = 0$ is justified for all $t \leq T$ where T is the time horizon of interest.

For the heat equation, $u(-x_{max}, 1) = u(x_{max}, 1) = \phi(x_{max})$ where ϕ is the standard normal density, e.g. $\phi(5) = 1.4867e-06$. Hence set in the following $x_{max} = 5\sqrt{T}$, and

$$u^m_{-N} = u^m_N = 0 \tag{2.21}$$

as boundary condition. Then (2.19) is well-defined for all m and $-N+1 \leq n \leq N-1$.

2.2.4 Implementation and tests

A practically extremely useful feature of the explicit scheme is that values for u^{m+1}_n are explicitly computable from the range of u^m_n's by (2.19) and (2.21). Starting from the initial condition (2.16), the solution for all m is therefore found inductively. This is sketched as pseudocode in Algorithm 1.

A few comments:

- When implementing, if we are only interested in the solution at $t = T$, i.e. $m = M$, there is no need to store the solution for all m and therefore we omit the superscript. At a given timestep, only two vectors are needed, denoted u for u^m and w for u^{m+1} in the algorithm.

- Most programming languages start indexing at 0 or 1. Therefore, we now shift indices by $N+1$ and use as solution vector, omitting m, $u = (u_1, u_2, \ldots, u_{2N}, u_{2N+1})$, corresponding to nodes $(-x_{max}, -x_{max} + \Delta x, \ldots, -\Delta x, 0, \Delta x, \ldots, x_{max} - \Delta x, x_{max})$, where $\Delta x = x_{max}/N$. That is u_n is thought to approximate $u((n-N-1)\Delta x, m\Delta t)$ rather than $u(n\Delta x, m\Delta t)$ as earlier.

- In lines 5-8, the initial condition is set according to (2.16) with the indices shifted as discussed above. Node $N+1$ is halfway between indices 1 and $2N+1$ and therefore $x_{N+1} = 0$.

- The main body is in lines 9-15, where the outer loop is over all timesteps, the inner loop implements (2.19) for all inner grid points, and line 10 incorporates (2.21).

In principle, Δt and Δx can now be chosen independently, however only for $p = \Delta t/2\Delta x^2 \leq 1/2$ the result is meaningful. (In the limiting case $p = 1/2$, the trinomial tree reduces to a binomial tree.) We therefore focus on the case that $\Delta t, \Delta x \to 0$ with $2p = \Delta t/\Delta x^2$ fixed and less than 1.

Algorithm 1 Explicit Euler scheme

$T \leftarrow 1$, $x_{max} \leftarrow 5\sqrt{T}$
$N \leftarrow 16$, $M \leftarrow 16$
$\Delta t \leftarrow T/M$, $\Delta x \leftarrow x_{max}/N$
$p = \Delta t/2\Delta x^2$
for $n \leftarrow 1$, $2N+1$ **do**
    $u_n \leftarrow 0$
end for
$u_{N+1} \leftarrow 1/\Delta x$
for $m \leftarrow 1$, M **do**
    $w_1 \leftarrow 0$, $w_{2N+1} \leftarrow 0$
    for $n \leftarrow 2$, $2N$ **do**
        $w_n \leftarrow p\,u_{n-1} + (1-2p)\,u_n + p\,u_{n+1}$
    end for
    $u = w$
end for

2.3 The implicit (backward) Euler scheme

A major drawback of the explicit scheme is the restriction on the timestep to obtain an *accurate* and *stable* solution. In practice, this means that a very large number of timesteps is needed, resulting in large computing times. We look at accuracy and stability in turn.

To develop an intuition for the stability of the scheme, rewrite (2.19) as

$$u_n^{m+1} = u_n^m + \frac{\Delta t}{\Delta x^2}\left(\frac{u_{n+1}^m + u_{n-1}^m}{2} - u_n^m\right).$$

If the solution at t_m is locally *convex* at x_n, such that the value u_n^m is smaller than the average of the neighbouring values, the increment $u_n^{m+1} - u_n^m$ is positive and the solution pulled up towards the average. Conversely, if the solution at t_m is locally *concave* at x_n, such that the value u_n^m is larger than the average of the neighbouring values, it is pulled down towards the average. Note this is basically the mechanism underlying the maximum/minimum principle of parabolic PDEs. However, if the timestep is chosen too large, the value overshoots the average and, in the extreme, $u_n^{m+1} \to \pm\infty$ as $\Delta t \to \infty$. This happens because the instantaneous time change is extapolated too far into the future, neglecting the two-way interplay of the change at x_n with the change of neighbouring values. To allow larger timesteps, we need to "couple" the equation for u_n^{m+1} somehow with those for u_{n-1}^{m+1}, u_{n+1}^{m+1} etc.

As concerns accuracy, we have seen that (2.17) defines an approximation to the heat equation which is accurate up to order $O(\Delta t, \Delta x^2)$. It is more accurate in the x-direction than in t. A simple high-level argument why the finite x difference is of second order accurate is the symmetry in $x \pm \Delta x$. There is no such symmetry in t in (2.17). This gives the idea for the following scheme.

2.3.1 Definition of the scheme

The mirror image in time of the scheme (2.18) is

$$\delta_t^- u_n^m = \frac{u_n^m - u_n^{m-1}}{\Delta t} = \frac{1}{2}\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\Delta x^2} = \delta_x^2 u_n^m, \tag{2.22}$$

where the right-hand side is evaluated at new time t_m instead of t_{m-1}.

Taylor expansion shows again that

$$\frac{u(x,t) - u(x,t-\Delta t)}{\Delta t} = \frac{1}{2}\frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2} + O(\Delta t, \Delta x^2),$$

in analogy to the explicit scheme.

Now write (2.22) as

$$-pu_{n+1}^m + (1+2p)u_n^m - pu_{n-1}^m = u_n^{m-1} \tag{2.23}$$

with $2p = \Delta t/\Delta x^2$ as before. We understand (2.23) again to define $u^m = (u_n^m)_{-\infty<n<\infty}$ in terms of u^{m-1}, such that any u^m is determined by forward induction from an initial condition u^0. In contrast to the explicit Euler scheme, however, a single u_n^m can no longer be explicitly computed from u^{m-1}, but is implicitly defined via a set of equations. Hence the name *implicit* scheme.

Going back to the large timestep behaviour, which was the crux of the explicit Euler scheme, if we let $\Delta t \to \infty$, i.e. $p \to \infty$ in (2.23), one formally gets $u_n^m \to (u_{n+1}^m + u_{n-1}^m)/2$ as desired. Note that p here no longer has any interpretation of a transition probability of a random walk, but is merely a parameter of the discretisation, so it is not unnatural to let it increase. There is no overshoot and the scheme is expected to be stable. We analyse this more carefully now.

2.3.2 Key properties

As for any numerical method, a crucial question is whether the numerical solution preserves properties of the solution to the original equation. In particular, we can ask here whether the solution can be seen as a discrete probability density. Recall this was the case for the explicit scheme, in fact it was the original motivation in its derivation.

Assuming boundedness, the solution u^1 after one step of the implicit Euler scheme is found (see Exercise 2) to be

$$u_n^1 = c\, z^{|n|}, \tag{2.24}$$

where $z = \alpha - \sqrt{\alpha^2 - 1} \in (0,1)$, $\alpha = (2p+1)/(2p) \in (1,\infty)$, $c\,\Delta x = (1-2p)/(1+2p)$. (**Warning**: This is obviously wrong as c would be negative for $p > 1/2$ – need to re-compute.) One verifies that

$$\Delta x \sum_{n\in\mathbb{Z}} u_n^1 = 1,$$

and clearly $u_n^1 > 0$. This makes $u_n^1 \Delta x$ interpretable as transition probability from $x_0 = 0$ at $t_0 = 0$ to x_n at time t_1.

So if we define an i.i.d. sequence (ξ_m) such that

$$\xi_m = n \text{ with probability } u_n^1 \Delta x, \tag{2.25}$$

instead of (2.1), a similar construction to Section 2.1.1 is possible. Note that u_n^1 is strictly positive for all n, such that there is a positive probability for the random walk to move to any point within a single timestep. This is characteristic for implicit schemes, and contrasts the behaviour of the explicit scheme. We can use this argument to construct u_n^m inductively and conclude that it indeed has the properties of a probability density.

There is another quick way to see that the scheme preserves non-negativity. Assume the opposite and that m is the first time index where the solution goes below zero somewhere. Then it follows from $u_n^m \to 0$ for $n \to \pm\infty$ that a negative minimum over all points is attained at a certain index k, $u_k^m = \min_{n\in\mathbb{Z}} u_n^m$. At this point,

$$u_k^m \leq \frac{1}{2}(u_{k+1}^m + u_{k-1}^m)$$

and hence

$$u_k^{m-1} = -pu_{k+1}^m + (1+2p)u_k^m - pu_{k-1}^m \leq u_k^m \tag{2.26}$$

and from $u_n^{m-1} \geq 0$ follows a contradiction, hence the solution must be non-negative.

Moreover, we get by similar reasoning a discrete minimum/maximum principle, i.e. for all $m \geq 0$, $n \in \mathbb{Z}$,

$$\min_{k\in\mathbb{Z}} u_n^0 \leq u_n^m \leq \max_{k\in\mathbb{Z}} u_n^0.$$

This implies further that there is a unique solution for all m. We can also deduce monotonicity in the initial data as in the explicit case, particularly $u^m \geq v^m$ for two solutions u^m and v^m with $u^0 \geq v^0$. By summing (2.23) over n, one gets again

$$\sum_{n\in\mathbb{Z}} u_n^m = \sum_{n\in\mathbb{Z}} u_n^{m-1}.$$

In particular, the properties of a discrete probability density are preserved.

2.3.3 Boundary conditions

In contrast to the explicit method, the grid value at any point in the implicit scheme has an influence on all other values in the next timestep. Equation (2.23) constitutes an infinite system which has to be solved simultaneously. Unless a closed-form solution to (2.23) is available, one has to restrict the sytem to finitely many unknowns to make it computationally feasible, which is done again by setting up approximate boundary conditions. We argued earlier that $u(x_{max}, t) = 0$ for large enough x_{max}, and similar for $-x_{max}$, such that we can restrict the system of equations (2.23), at least approximately, to a range $-N+1 \leq n \leq N-1$ by setting

$$u_{-N}^m = u_N^m = 0. \tag{2.27}$$

Equations (2.23) and (2.27) give a linear system for $u^m = (u_{-N}^m, \ldots, u_0^m, \ldots u_N^m)$,

$$Ku^m = u^{m-1}, \tag{2.28}$$

where

$$K = \begin{pmatrix} 1 & 0 & 0 & & & 0 & 0 \\ -p & 1+2p & -p & & \ldots & 0 & 0 \\ 0 & -p & 1+2p & -p & \ldots & 0 & 0 \\ 0 & & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & & \ldots & -p & 1+2p & -p & 0 \\ 0 & & \ldots & 0 & -p & 1+2p & -p \\ 0 & 0 & & & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(2N+1)\times(2N+1)}.$$

The first and last line determine the boundary values. The linear system (2.28) has to be solved in each timestep, replacing the explicit calculations of the previous scheme.

2.3.4 Implementation and tests

The implicit Euler algorithm follows similar lines to the explicit one, with the modification that a linear system has to be set up, and solved in every timestep. We explain this first, then sketch the overall algorithm.

Solving the linear system

The system (2.28) is of the general form

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \dots & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}, \tag{2.29}$$

where $n = 2N + 1$ for simplicity. For future reference, denote the system matrix by

$$K = \text{tridiag}(a, b, c, n), \tag{2.30}$$

n being the dimension of the square matrix $K \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ the diagonal, and a and c left and right of the diagonal respectively. Note that a_1 and c_n are not needed but we define $a, c \in \mathbb{R}^n$ for ease of notation.

The system matrix K is *sparse*, meaning that per line there is a given number – in this case three – of non-zero elements. This is independent of the size of the matrix and does not increase if the number N of grid points increases. Loosely speaking, this comes from the fact that differentiation is a "local" operation, so a discretisation of a differential equation should not have to take into account grid values some distance away.

More specifically, K is a *tridiagonal* matrix: it has non-zero elements only in the main diagonal and for neighbouring entries. This can be taken advantage of when solving the system. An important consequence is that the number of calculations, and hence the computational time required for the solution of the system is proportional to the number of unknowns. This is referred to as an $O(n)$ algorithm. The overall computational time of the implicit scheme is therefore not notably larger than for the simpler explicit scheme.

One easy way to solve such a linear system is by an adaptation of Gaußian elimination, known as the Thomas algorithm. It exploits the tridiagonal structure of the matrix to solve the system in $O(n)$ operations.

Although it is sometimes useful to write $u^m = K^{-1}u^{m-1}$, there are almost no circumstances under which it is advisable to invert these matrices directly in practice. To see why,

consider the discretisation matrix K and its inverse K^{-1}, for $x_{max} = 3$, $N = 3$, $M = 1$:

$$K = \begin{pmatrix} 2 & -0.5 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 2 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 2 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 2 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 2 & -0.5 & 0 \\ 0 & 0 & 0 & 0 & -0.5 & 2 & -0.5 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 2 \end{pmatrix} \tag{2.31}$$

$$K^{-1} = \begin{pmatrix} 0.5359 & 0.1436 & 0.0385 & 0.0103 & 0.0028 & 0.0007 & 0.0002 \\ 0.1436 & 0.5744 & 0.1539 & 0.0412 & 0.0110 & 0.0029 & 0.0007 \\ 0.0385 & 0.1539 & 0.5771 & 0.1546 & 0.0414 & 0.0110 & 0.0028 \\ 0.0103 & 0.0412 & 0.1546 & 0.5773 & 0.1546 & 0.0412 & 0.0103 \\ 0.0028 & 0.0110 & 0.0414 & 0.1546 & 0.5771 & 0.1539 & 0.0385 \\ 0.0007 & 0.0029 & 0.0110 & 0.0412 & 0.1539 & 0.5744 & 0.1436 \\ 0.0002 & 0.0007 & 0.0028 & 0.0103 & 0.0385 & 0.1436 & 0.5359 \end{pmatrix} \tag{2.32}$$

The inverse of a sparse (tridiagonal) matrix is normally non-sparse, with $O(n^2)$ non-zero elements. In each timestep of the implicit Euler scheme, we would multiply this matrix with a vector. The complexity of this, i.e. the number of operations necessary, is also $O(n^2)$. The complexity of the inversion itself is $O(n^3)$, but has to be performed once in the set-up phase only. This compares to $O(n)$ per timestep when solving the system.

The implicit Euler algorithm

The full algorithm of the implicit Euler scheme is sketched in Algorithm 2. Lines 5-13 define the initial condition and system matrix, where we stick to the format from (2.28), (2.30). Line 16 replaces lines 11 to 13 of Algorithm 1.

Algorithm 2 Implicit Euler scheme

1: $T \leftarrow 1$, $x_{max} \leftarrow 5\sqrt{T}$
2: $N \leftarrow 16$, $M \leftarrow 16$
3: $\Delta t \leftarrow T/M$, $\Delta x \leftarrow x_{max}/N$
4: $\lambda = \Delta t/\Delta x^2$
5: $u_1 \leftarrow 0$
6: $a_1 \leftarrow 0$, $b_1 \leftarrow 1$, $c_1 \leftarrow 0$
7: **for** $n \leftarrow 2$, $2N - 1$ **do**
8: $\quad u_n \leftarrow 0$
9: $\quad a_n \leftarrow -\lambda/2$, $b_n \leftarrow 1 + \lambda$, $c_n \leftarrow -\lambda/2$
10: **end for**
11: $u_{2N+1} \leftarrow 0$
12: $a_{2N+1} \leftarrow 0$, $b_{2N+1} \leftarrow 1$, $c_{2N+1} \leftarrow 0$
13: $u_{N+1} \leftarrow 1/\Delta x$
14: $K = \text{tridiag}(a, b, c, 2N + 1)$
15: **for** $m \leftarrow 1$, M **do**
16: $\quad$ solve $Kv = u$ for v, set $u \leftarrow v$
17: **end for**

2.4 Difference schemes as Markov chains

Recall that the explicit Euler scheme was first derived as describing the transition density of a discrete-time random walk on a grid $\Delta x\mathbb{Z}$. It is a natural question to ask whether the implicit scheme has a similar interpretation, ie is there a random walk X_t on the state space $\Delta x\mathbb{Z}$, such that the associated transition density $U_n^m = \mathbb{P}(X_{m\Delta t} = n\Delta x)$ satisfies the implicit Euler scheme?

We have already shown in (2.3.2) that the solution to the implicit Euler scheme has the making of a probability density – non-negativity and "preserves probability",

$$\sum_{n=-\infty}^{\infty} u_n^m = \sum_{n=-\infty}^{\infty} u_n^{m-1}. \tag{2.33}$$

A prime candidate for a random walk is therefore defined by the one-step transition densities,

$$X_{t_m+\Delta t} = X_{t_m} + \xi_m \Delta x,$$

where

$$\mathbb{P}(\xi_m = n) := U_n^1 := u_n^1 \Delta x$$

with $u_n^1 = c\Delta x z^{|n|}$ as defined in (2.24), and all ξ_m independent.

The process X is a *Markov chain* because it satisfies the *Markov property*

$$\mathbb{P}(X_{t_m} = y_m | X_{t_0} = y_0, \ldots, X_{t_{m-1}} = y_{m-1}) = \mathbb{P}(X_{t_m} = y_m | X_{t_{m-1}} = y_{m-1}).$$

See e.g. [Grimmet and Stirzaker, 2001], pp214, for more details. We can think of the process having an infinite transition matrix

$$P_{nk} = \mathbb{P}(X_{t_m} = x_k | X_{t_{m-1}} = x_n) = \mathbb{P}(X_{t_m} = x_{k-n} | X_{t_{m-1}} = 0) = P_{0,k-n} = U_{k-n}^1.$$

For the implicit scheme on a finite grid, where we approximate $U_N^m = U_{-N}^m = 0$ for sufficiently large N, (2.33) is no longer exactly given, but instead

$$\sum_{n=-N}^{N} U_n^m = \sum_{n=-N}^{N} U_n^{m-1} - p\,(U_{-N+1}^m + U_{N-1}^m) < \sum_{n=-N}^{N} U_n^{m-1}.$$

The boundaries are absorbing and the probability of staying in the interval $(-N\Delta x, N\Delta x)$ goes down by the probability $pU_{\pm(N-1)}^m$ of moving out to $\pm N\Delta x$ from $\pm(N-1)\Delta x$. We could account for this by introducing "absorbed" states into the Markov chain.

We try a different tack and see if we can modify the boundary conditions at $\pm N\Delta x$ in order to preserve probability. For reasons that will become apparent shortly it is useful to define the vector $(1) = (1, \ldots, 1) \in \mathbb{R}^{2N+1}$. With this notation, we want

$$1 = \sum_{n\in\mathbb{Z}} U_n^m = (1)'U^m = (1)'(K^{-1}U^{m-1}) = ((1)'K^{-1})U^{m-1},$$

but because $(1)'U^{m-1} = 1$, it is sufficient that

$$(1)'K^{-1} = (1)' \quad \Leftrightarrow \quad (1)' = (1)'K,$$

that is to say all columns sum up to one,

$$\sum_{k=-N}^{N} K_{kn} = 1 \quad \forall n. \tag{2.34}$$

Taken the coefficients for interior points as given, this is only possible if

$$\begin{aligned} K_{N,N} &= 1+p \\ K_{N,N-1} &= -p \end{aligned}$$

and zero elsewhere in the last line, and similarly for $-N$. The corresponding equation is

$$U_N^m = U_N^{m-1} - p(U_N^m - U_{N-1}^m) \quad \Leftrightarrow \quad \frac{U_N^m - U_N^{m-1}}{\Delta t} = \frac{1}{2}\frac{U_N^m - U_{N-1}^m}{\Delta x^2},$$

which is consistent with

$$\frac{U_N^m - U_N^{m-1}}{\Delta t} = \frac{1}{2}\frac{U_{N+1}^m - 2U_N^m + U_{N-1}^m}{\Delta x^2}$$

only if

$$\frac{U_{N+1}^m - U_N^m}{\Delta x} = 0.$$

This is the discrete counterpart of a *Neumann boundary condition*

$$\frac{\partial u}{\partial x}(x_{max}, t) = 0$$

for the heat equation. It is left as an exercise to check that the heat equation with Neumann boundary conditions at the upper and lower boundary preserves the integral of the solution.

Back to the discretised equations, we already know that for any $u \geq 0$ the solution to $Kv = u$ has $v = K^{-1}u \geq 0$, and since this is true for all u,

$$P_{nk} = (K^{-1})_{nk} \geq 0. \tag{2.35}$$

It follows that

$$P = K^{-1}$$

is a so-called *stochastic matrix* defined through properties (2.34) and (2.35). The *transition matrix* P therefore defines a *Markov chain*.

In an appropriate limit $\Delta t, \Delta x \to 0$, the Markov chain approaches Brownian motion. It is an interesting question to ask what happens if we consider continuous time ($\Delta t \to 0$) but keep the states x_n fixed. (We leave the Δx limit for fixed Δt to Exercise 6, Section 4.4.)

The limit will have the form $U_n(t)$, where $n \in \mathbb{Z}$ and $t \geq 0$. Neither the explicit nor the implicit scheme have difficulties for small Δt, and letting $t \to 0$ in either one formally gets

$$\frac{\mathrm{d}}{\mathrm{d}t}U_n(t) = \frac{1}{2}\frac{U_{n-1}(t) - 2U_n(t) + U_{n+1}(t)}{\Delta x^2}, \tag{2.36}$$

with initial condition

$$U_n(0) = \delta_{n0}.$$

In matrix form,

$$\frac{\mathrm{d}}{\mathrm{d}t}U(t) = AU(t) \tag{2.37}$$

with $A = 1/(2\Delta x^2)\text{tridiag}(1, -2, 1, 2N+1)$. Writing the explicit Euler scheme as

$$\frac{U^{m+1} - U^m}{\Delta t} = AU^m \quad \Leftrightarrow \quad U^{m+1} = (I + \Delta t A)U^m$$

gives

$$U^m = (I + \Delta t A)^m U^0 = (I + \Delta t A)^{t/\Delta t} U^0 \;\; \to \;\; U(t) = \mathrm{e}^{At}U^0 \quad \text{for } \Delta t \to 0.$$

A similar derivation using the implicit scheme, of course, gives the same result. This defines a continuous-time Markov process X_t on the state space $\mathbb{Z}$ with transition probabilities

$$p_{kn}(t) = \mathbb{P}(X_t = n | X_0 = k) = U_{n-k}(t).$$

The matrix

$$P_t = \mathrm{e}^{At}$$

satisfies the forward equation

$$\frac{\mathrm{d}}{\mathrm{d}t}P_t = P_t A.$$

The process X_t jumps between states at exponentially distributed times, specifically the time of the first jump, $\tau = \inf\{t \geq 0 : \; X_t \neq 0\}$, is exponentially distributed with parameter $-A_{00} = 1/\Delta x^2$. The chain jumps to ± 1 with equal probability $1/2$. An analogous statement is true for the following inter-jump times.

2.5 Exercises

1. Given is the setup from Sections 2.1.1 and 2.1.2.

   (a) Show that for $m \geq 0$, $0 \leq n \leq m$,

   $$U_n^m = U_{-n}^m = \sum_{k=0}^{\lfloor \frac{m-n}{2} \rfloor} \frac{m! p^{n+2k}(1-2p)^{m-n-2k}}{(n+k)!k!(m-n-2k)!}. \tag{2.38}$$

   (b) By insertion, show that (2.38) solves the recursion (2.7) and initial condition (2.8).

   (c) Show directly that in the limit $\Delta x \to 0$, $\Delta t = 2p\Delta x^2$,

   $$\frac{1}{\Delta x}U_n^m \to \frac{1}{\sqrt{2\pi t}}\mathrm{e}^{-\frac{x^2}{2t}},$$

   where $t = m\Delta t$ and $x = n\Delta x$ fixed.

2. Given is the setup from Section 2.3.1.

(a) For $m = 1$, the scheme takes the special form

$$\begin{aligned} -pu^1_{-1} + (1+2p)u^1_0 - pu^1_1 &= \Delta x^{-1}, & (2.39)\\ -pu^1_{n-1} + (1+2p)u^1_n - pu^1_{n+1} &= 0, \quad n \neq 0. & (2.40) \end{aligned}$$

Show that the general solution to (2.40) for $n > 0$ is of the form

$$u^1_n = c_- z^n_- + c_+ z^n_+,$$

where you have to determine z_- and z_+. Use (2.39) to find the (unique) bounded symmetric solution. Verify that

$$\Delta x \sum_{n\in\mathbb{Z}} u^1_n = 1.$$

(b) Show that

$$u^m_n = \sum_{k\in Z} u^{m-1}_k u^1_{n-k} \Delta x$$

defines a solution to the implicit scheme. What is the counterpart for continuous x and discrete time?

(c) Explicitly find u^2_n for all n. Does this lead to a computationally efficient method?

3. Consider the implicit Euler finite difference scheme for the heat equation with Dirac initial data, for $x \in [-5, 5]$, $t \in [0, 1]$, with zero boundary conditions at -5 and 5. Use a grid $(-N\Delta x, \ldots, -\Delta x, 0, \Delta x, \ldots, N\Delta x)$, where $N\Delta x = 5$, and M timesteps, $\Delta t = 1/M$.

(a) Show that for $\Delta t \to 0$, and Δx *fixed*, the finite difference scheme approaches a system of ODEs

$$\begin{aligned} \frac{\partial u}{\partial t} &= Au & (2.41)\\ u(0) &= u^0 & (2.42) \end{aligned}$$

where $u(t) = (u_1(t), \ldots, u_{2N+1}(t)) \in \mathbb{R}^N$, $u^0 \in \mathbb{R}^N$, $A \in \mathbb{R}^{N\times N}$. State A and u^0. Explain why this is often called the *method of lines*.

(b) Show that a solution to (2.41), (2.42) is given by $u(t) = \mathrm{e}^{A\,t}\,u^0$. Hence compute the solution to this semi-discrete problem (x discrete, t continuous) in MATLAB for $t = 1$, $N = 10$ and plot over x together with the exact solution.

(c) Compute the error for $x = 0$, $t = 1$, for $N = 10, 20, 40, 80, 160, 320$. What do you observe?

4. (a) Implement the explicit as well as the implicit Euler finite difference scheme for the heat equation with Dirac initial data, for $x \in [-5, 5]$, $t \in [0, 1]$, with zero boundary conditions at -5 and 5. Use a grid $(-N\Delta x, \ldots, -\Delta x, 0, \Delta x, \ldots, N\Delta x)$, where $N\Delta x = 5$, and M timesteps, $\Delta t = 1/M$.

(b) Table the difference between

i. the discrete solutions and the analytical one,

ii. the discrete solutions and the semi-discrete solution from 3. (with the same number of grid points),

at $x = 0$, $t = 1$, for both schemes for $N = 10, 20, 40, 80, 160, 320$, M =10, 20, 40, 80, 160, 320, 640, 1280, 2560. Discuss the different patterns.

Chapter 3

Extensions to the basic scheme

3.1 Random walks with drift

We return to the set-up of 2.1, but now want to include the possibility that the position of the random walk at a time t, X_t, has a given non-zero expectation. For simplicity, consider first the case where the marker drifts, on average, with constant rate μ, such that

$$\mathbb{E}(X_{t+\Delta t} - X_t) = \mu \Delta t. \tag{3.1}$$

We will look at two ways to achieve this: one of making the walk asymmetric by attaching a higher probability to up- or down-moves, hence creating a bias; and one of making the nodes drift themselves. Finally, we bring the two together by a scheme that is motivated by the latter approach, but, by projeting the moves onto a fixed lattice, results in a scheme similar to the former one.

3.1.1 Biased moves

First, we define a random walk again by

$$X_{t_m+\Delta t} = X_{t_m} + Z_m \Delta x,$$

with $X_0 = 0$, $\Delta x > 0$, Z_m i.i.d., as in 2.1, but now allowing unequal probability for the direction of moves,

$$\xi_m = \begin{cases} 1 & \text{with probability } p, \\ -1 & \text{with probability } q, \\ 0 & \text{with probability } 1-p-q, \end{cases}$$

where $0 \leq p, q$ and $p + q \leq 1$. To match (3.1), one needs

$$\mathbb{E}(X_{\Delta t}) = \mu \Delta t. \tag{3.2}$$

The variance is again (see 2.1.1) normalised by

$$Var(X_{\Delta t}) = \mathbb{E}(X_{\Delta t}^2) - \mathbb{E}(X_{\Delta t})^2 = \Delta t. \tag{3.3}$$

It is left as an exercise to work out the details of the following steps. The above two conditions define p and q as

$$p = \frac{1}{2}\frac{\Delta t}{\Delta x^2} + \frac{1}{2}\mu\frac{\Delta t}{\Delta x} + \frac{1}{2}\mu^2\frac{\Delta t^2}{\Delta x^2}, \quad q = p - \mu\frac{\Delta t}{\Delta x}. \tag{3.4}$$

Note that $0 \leq p, q$ and $p + q \leq 1$ only if

$$\begin{aligned} |\mu|\Delta x + \mu^2\Delta t &\leq 1 \text{ and} \\ \frac{\Delta t}{\Delta x^2} + \mu^2\frac{\Delta t^2}{\Delta x^2} &\leq 1, \end{aligned}$$

respectively. The first condition is not critival because Δt and Δx are small, the second one is not much more critical than in the symmetric case $\mu = 0$ because if $\Delta t \leq \Delta x^2$, and Δx small, then the second term is also small.

The inductive formula for the probability density $u_n^m \Delta x = \mathbb{P}(X_{m\Delta t} = n\Delta x)$ (see 2.1.2 and 2.1.3) then reads

$$u_n^{m+1} = pu_{n-1}^m + (1 - p - q)u_n^m + qu_{n+1}^m, \tag{3.5}$$

with its continuous limit the drift-diffusion equation

$$\frac{\partial u}{\partial t} + \mu\frac{\partial u}{\partial x} = \frac{1}{2}\frac{\partial^2 u}{\partial x^2}. \tag{3.6}$$

The inital density is again $u(x, 0) = \delta(x)$ and the solution to (3.6) is then given by

$$u_\mu(x, t) = \frac{1}{\sqrt{2\pi t}}\mathrm{e}^{-(x-\mu t)^2/2t}, \tag{3.7}$$

which is seen to be a shift of the drift-less solution u to the heat equation, $u_\mu(x, t) = u(x - \mu t, t)$.

Inserting p and q from (3.4) into (3.5), an explicit finite difference scheme for (3.6) is given by

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} + \mu\frac{u_{n+1}^m - u_{n-1}^m}{2\Delta x} = \frac{1}{2}(1 + \Delta t\mu^2)\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\Delta x^2}. \tag{3.8}$$

Neglecting the Δt term (why is this justified?), this slightly simplifies to

$$\delta_t^+ u_n^m + \mu\delta_x u_n^m = \delta_x^2 u_n^m \tag{3.9}$$

The second term

$$\delta_x u_n^m = \frac{u_{n+1}^m - u_{n-1}^m}{2\Delta x}$$

discretises the first spatial derivative and is a *central first difference*, which is of second order accurate in Δx.

The scheme (3.9) defines a random walk if

$$|\mu|\Delta x \leq 1$$

in addition to the standard condition on $\Delta t \leq \Delta x^2$. A more subtle point is that even if Δt and Δx are chosen in such a way, this random walk will *not* satisfy (3.2) and (3.3). This does not preclude convergence of the finite difference solution from (3.9) to the exact solution of (3.6). In fact, many of the numerical schemes we will look at later will not exactly track the moments of the underlying random walks, or have an underlying random walk in the first place. What is relevant is that the scheme approximates the PDE in the limit of small Δt and Δx, in a sense to be made precise later.

3.1.2 Moving trees and grids

As an alternative to the previous approach, a deterministic drift can be incorporated in a shift of the nodes of the tree over time. This leads to

$$X_{t_m+\Delta t} = X_{t_m} + \mu\Delta t + \xi_m \Delta x,$$

where as originally $0 \leq p = q \leq 1/2$. The moves around the mean are symmetric. At time $t = t_m = m\Delta t$, the marker moves from $X_t = x$ to $X_{t+\Delta t} = x + \mu\Delta t + \Delta x$ with probability p, to $x + \mu\Delta t - \Delta x$ with equal probability p, and to $x + \mu\Delta t$ with probability $1 - 2p$.

Defining $u_n^m \Delta x = \mathbb{P}(X_{m\Delta t} = n\Delta x + \mu m\Delta t)$,

$$u_n^{m+1} = pu_{n-1}^m + (1-2p)u_n^m + pu_{n+1}^m. \tag{3.10}$$

This finite difference scheme is the same as for the heat equation, as the drift is already incorporated in the coordinate system. One confirms by Taylor expansion that for $\Delta t, \Delta x \to 0$, $\Delta t/\Delta x^2 = 2p$, the limiting PDE is again (3.6), if u_n^m is seen as approximation to $u(n\Delta x + \mu m\Delta t, m\Delta t)$.

Another twist on this is to solve the heat equation *without* drift with finite differences on a *fixed* grid up to time $t = T$, but to evaluate the numerical solution at time T at the point $x - \mu T$ to get an approximation to $u_\mu(x,t)$ which solves (3.6). If $x - \mu T$ is not a grid point, one can take the closest neighbour or interpolate more accurately from its neighbours. This leads to so-called *Lagrangian* schemes discussed in the next section.

3.1.3 Lagrangian coordinates and upwind differencing

Following through the reasoning of the previous sections, we note that the solution to

$$\frac{\partial u}{\partial t} + \mu\frac{\partial u}{\partial x} = \frac{1}{2}\frac{\partial^2 u}{\partial x^2}$$

is the solution to the heat equation, shifted along a path

$$\xi(t) = x + \mu\,(t - t_m).$$

We have anchored the path to go through x at a time point t_m. Along this coordinate,

$$\frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t),t) = \frac{\partial u}{\partial t} + \mu\frac{\partial u}{\partial x},$$

so integrating in time leads to

$$u(x,t_m) - u(x-\mu\Delta t, t_{m-1}) = \int_{t_{m-1}}^{t_m} \frac{\mathrm{d}}{\mathrm{d}t}u(\xi(t),t)\,\mathrm{d}t = \frac{1}{2}\int_{t_{m-1}}^{t_m} \frac{\partial^2 u}{\partial x^2}(\xi(t),t)\,\mathrm{d}t.$$

Approximating the x-derivative with central differences, using shorthand

$$\delta_x^2 u(x,t) = \frac{u(x-\Delta x,t) - 2u(x,t) + u(x+\Delta x,t)}{\Delta x^2},$$

leads to

$$u(x,t_m) - u(x-\mu\Delta t, t_{m-1}) = \frac{1}{2}\int_{t_{m-1}}^{t_m} \left[\delta_x^2 u(\xi(t),t) + O(\Delta x^2)\right]\,\mathrm{d}t.$$

At this point, a decision has to be made about approximation of the integral. Note that $\xi(t)$ will not coincide with a grid point for most t, but we have the freedom to choose e.g. $\xi(t_m) = x$, and then can approximate the time integral by $\delta_x u(x, t_m)\Delta t$ with an error of $O(\Delta t^2)$. This is effectively the implicit Euler scheme and leads to

$$\frac{u(x,t_m) - u(x - \mu\Delta t, t_{m-1})}{\Delta t} = \frac{1}{2}\delta_x^2 u(x,t_m) + O(\Delta x^2) + O(\Delta t).$$

The problem with practical implementation of the scheme on a fixed grid is that if x, $x - \Delta x$, $x + \Delta x$ are gridpoints, then $x - \mu\Delta t$ still is not. But $u(x - \mu\Delta t, t_{m-1})$ can be reconstructed from grid values at t_{m-1} by interpolation, say Iu^{m-1}.

So if

$$-pu_{n-1}^m + (1+2p)u_n^m - pu_{n+1}^m = u_n^{m-1}$$

is the implicit scheme for the heat equation (i.e. without drift!), then

$$-pu_{n-1}^m + (1+2p)u_n^m - pu_{n+1}^m = Iu^{m-1}(x_n - \mu\Delta t)$$

is the corresponding scheme in Lagrangian coordinates for the equation with drift.

For the interpolation, use e.g. piecewise linear,

$$Iu^{m-1}(x) = \sum_n u_n^{m-1}\widehat{\Phi}_n(x)$$

with "hat functions" $\widehat{\Phi}_n$ defined as

$$\widehat{\Phi}_n = \begin{cases} 0 & x \le x_{n-1} \text{ or } x \ge x_{n+1}, \\ 1 + \frac{x - x_n}{\Delta x} & x_{n-1} \le x \le x_n, \\ 1 - \frac{x - x_n}{\Delta x} & x_n \le x \le x_{n+1}. \end{cases}$$

Then $Iu^{m-1}(x_n) = u_n^{m-1}$ for all n, and Iu^{m-1} is linear in intervals $[x_n, x_{n-1}]$.

The interpolation point $x_n - \mu\Delta t$ lies left of x_n if $\mu > 0$ and right of x_n if $\mu < 0$. It lies in $[x_{n-1}, x_{n+1}] = [x_n - \Delta x, x_n + \Delta x]$ if $\mu\Delta t \le \Delta x$. In this case, the resulting finite difference scheme is equivalent to

$$\begin{aligned} \frac{u_n^{m+1} - u_n^m}{\Delta t} + \mu\frac{u_n^m - u_{n-1}^m}{\Delta x} &= \frac{1}{2}\frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{\Delta x^2}, \\ \delta_t^+ u_n^m + \mu\delta_x^- u_n^m &= \frac{1}{2}\delta_x^2 u_n^m, \end{aligned}$$

if $\mu \ge 0$, and to

$$\begin{aligned} \frac{u_n^{m+1} - u_n^m}{\Delta t} + \mu\frac{u_{n+1}^m - u_n^m}{\Delta x} &= \frac{1}{2}\frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{\Delta x^2}, \\ \delta_t^+ u_n^m + \mu\delta_x^+ u_n^m &= \frac{1}{2}\delta_x^2 u_n^m, \end{aligned}$$

if $\mu \le 0$. It is interesting to note that the discretisation of the first derivative in this case uses one-sided differences δ_x^- and δ_x^+ respectively, taking into account the direction of the drift. Because of its physical interpretation of going up against the drift ("wind"), it is traditionally called *upwind* differencing.

For this piecewise linear interpolation, the overall error is of order $O(\Delta x)$ in spite of second order of the finite difference δ_x^2. This can be fixed by higher order interpolation, e.g. cubic splines, at the expense of losing monotonicity.

As an aside, we record the relations

$$\begin{aligned}\delta_x^2 &= \delta_x^+\delta_x^- = \delta_x^-\delta_x^+,\\ \delta_x &= \frac{1}{2}\left(\delta_x^+ + \delta_x^-\right).\end{aligned}$$

3.2 Weighted timestepping schemes and Crank-Nicolson

3.2.1 Combining schemes

We now switch the focus on improving the accuracy of the time discretisations. The explicit and implicit schemes are "one-sided" approxiamtions to the time-derivative from opposite directions, both resulting in first order accuracy. This begs the question whether a combination of the two might give better, possibly second order accurate solutions. We first try alternating the steps, which leaves two possibilities.

If the first step is explicit, and the second implicit,

$$\begin{aligned}\frac{u_n^m - u_n^{m-1}}{\Delta t} &= \frac{1}{2}\frac{u_{n+1}^{m-1} - 2u_n^{m-1} + u_{n-1}^{m-1}}{\Delta x^2},\\ \frac{u_n^{m+1} - u_n^m}{\Delta t} &= \frac{1}{2}\frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{\Delta x^2},\end{aligned}$$

the resulting scheme is

$$\delta_t u = \frac{u_n^{m+1} - u_n^{m-1}}{2\Delta t} = \frac{1}{4}\frac{u_{n+1}^{m-1} - 2u_n^{m-1} + u_{n-1}^{m-1}}{\Delta x^2} + \frac{1}{4}\frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{\Delta x^2}.$$

The scheme jumps level m altogether, so can leave it out and re-write it with half the step-size,

$$\frac{u_n^m - u_n^{m-1}}{\Delta t} = \frac{1}{4}\frac{u_{n+1}^{m-1} - 2u_n^{m-1} + u_{n-1}^{m-1}}{\Delta x^2} + \frac{1}{4}\frac{u_{n+1}^{m} - 2u_n^{m} + u_{n-1}^{m}}{\Delta x^2}. \tag{3.11}$$

This is the very popular *Crank-Nicolson scheme*. It is implicit. One would expect that at the very least that the scheme inherits the accuracy from the individual sub-steps, but given its symmetry we hope to get an improvement. It is also clear that if both sub-steps preserve the defining properties of a probability distribution, then the overall scheme will preserve them.

If we switch the order of explicit and implicit step, i.e. the first sub-step is implicit, and the second one explicit,

$$\begin{aligned}\frac{u_n^m - u_n^{m-1}}{\Delta t} &= \frac{1}{2}\frac{u_{n+1}^{m} - 2u_n^{m} + u_{n-1}^{m}}{\Delta x^2},\\ \frac{u_n^{m+1} - u_n^m}{\Delta t} &= \frac{1}{2}\frac{u_{n+1}^{m} - 2u_n^{m} + u_{n-1}^{m}}{\Delta x^2},\end{aligned}$$

the resulting scheme is

$$\frac{u_n^{m+1} - u_n^{m-1}}{2\Delta t} = \frac{1}{2}\frac{u_{n+1}^{m} - 2u_n^{m} + u_{n-1}^{m}}{\Delta x^2}. \tag{3.12}$$

Recall that m here is an odd number by the alternating definition of the scheme, and (3.12) alone does not define the solution. If instead we apply (3.12) in every timestep m, the resulting method is known as the *Leapfrog scheme*. Note that it is only defined for $m > 1$. In the first step, one has to provide an approximation to u^1 from the initial condition u^0. One possibility is to use the explicit Euler scheme. The scheme is then explicit. Unfortunately, we will see later that the scheme is always unstable.

We therefore focus the discussion now on the Crank-Nicolson scheme (3.11).

3.2.2 Method of lines and the θ-scheme

To bring some order into this growing zoo of timestepping schemes, we can think of the problem more systematically by taking as stepping stone the semi-discrete system

$$\frac{\mathrm{d}}{\mathrm{d}t}u_n(t) = \frac{1}{2}\frac{u_{n+1}(t) - 2u_n(t) + u_{n-1}(t)}{\Delta x^2} \tag{3.13}$$

where $n \in \mathbb{Z}$ and $u_n(0) = u_n^0$ a given initial value. We have seen in 2.4, specifically (2.36), that this is the limit of the explicit and implicit finite difference schemes if we let $\Delta t \to 0$. Coming from the continuous equations, we can think of it as letting time continuous, but discretising x. We track the solution continuously along lines (x_n, t), which leads to the name ("vertical") *method of lines*. (A "horizontal" method of lines would fix Δt and consider continuous x.) The structure of these equations is that of a coupled systems of ordinary differential equations. The explicit Euler scheme, implicit Euler scheme, Crank-Nicolson scheme etc are all *timestepping* schemes for this system of ODEs. For simplicity, we omit boundary conditions but one could truncate by setting $u_{-N}(t) = u_N(t) = 0$.

Integrating (3.13) over a time interval $[m\Delta t, (m+1)\Delta t]$,

$$u_n((m+1)\Delta t) - u_n(m\Delta t) = \frac{1}{2}\int_{m\Delta t}^{(m+1)\Delta t} \frac{1}{\Delta x^2}\left[u_{n+1}(t) - 2u_n(t) + u_{n-1}(t)\right]\,\mathrm{d}t.$$

Instead of approximating a derivative, we need to now approximate an integral. The left-sided rectangle rule,

$$\frac{1}{\Delta t}\int_a^b u(t)\,\mathrm{d}t = u(a) + O(\Delta t),$$

leads to the explicit scheme. The right-sided rectangle rule,

$$\frac{1}{\Delta t}\int_a^b u(t)\,\mathrm{d}t = u(b) + O(\Delta t),$$

gives the implicit scheme. The second order accuate trapezium rule,

$$\frac{1}{\Delta t}\int_a^b u(t)\,\mathrm{d}t = \frac{u(a)+u(b)}{2} + O(\Delta t^2),$$

motivates the *Crank-Nicolson* scheme.

Slightly more generally, we can embed these schemes in a family

$$\frac{u_n^m - u_n^{m-1}}{\Delta t} = \frac{\theta}{2}\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\Delta x^2} + \frac{(1-\theta)}{2}\frac{u_{n+1}^{m-1} - 2u_n^{m-1} + u_{n-1}^{m-1}}{\Delta x^2} \tag{3.14}$$

with a parameter $\theta \in [0,1]$. This *θ-scheme* has as its special cases, for

- $\theta = 0$ the explicit Euler scheme, for
- $\theta = 1/2$ the Crank-Nicolson scheme, and for
- $\theta = 1$ the implicit Euler scheme.

The explicit scheme has as prime virtue its explicitness and resulting implementational ease, the implicit scheme its unfaltering stability, and the Crank-Nicolson scheme's distinguishing feature is its second order accuracy. None of the schemes with differing θ have any of these advantages and are not of much practical relevance in their own right. We will see later that timesteps with varying θ can be used as sub-steps of useful combined schemes, in a similar spirit to Crank-Nicolson. Moreover, seeing these three examples as special cases of a bigger scheme allows as a unified analysis later on, and to some extent unified implementation.

3.2.3 Implementation and numerical tests

Rearrange the θ-scheme for the heat equation (3.14) into

$$-\frac{\theta\lambda}{2}u^m_{n+1} + (1+\theta\lambda)u^m_n - \frac{\theta\lambda}{2}u^m_{n-1} = \frac{(1-\theta)\lambda}{2}u^{m-1}_{n+1} + (1-(1-\theta)\lambda)u^{m-1}_n + \frac{(1-\theta)\lambda}{2}u^{m-1}_{n+1}$$

for $-N < n < N$, and add boundary conditions $u^m_{-N} = u^m_{-N} = 0$. If u^m_{-N} and u^m_{-N} are eliminated from the system, this reads in matrix-vector form

$$\underbrace{\begin{pmatrix} b & c & & \dots & 0 \\ a & b & c & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & a & b & c \\ 0 & \dots & 0 & a & b \end{pmatrix}}_{:=K_1} \begin{pmatrix} u^m_{-N+1} \\ u^m_{-N+2} \\ \vdots \\ u^m_{N-2} \\ u^m_{N-1} \end{pmatrix} = \underbrace{\begin{pmatrix} B & C & & \dots & 0 \\ A & B & C & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & A & B & C \\ 0 & \dots & 0 & A & B \end{pmatrix}}_{:=K_2} \begin{pmatrix} u^{m-1}_{-N+1} \\ u^{m-1}_{-N+2} \\ \vdots \\ u^{m-1}_{N-2} \\ u^{m-1}_{N-1} \end{pmatrix} \tag{3.15}$$

with

$$\begin{aligned} a &= -\theta\frac{\lambda}{2} \\ b &= 1+\theta\lambda \\ c &= -\theta\frac{\lambda}{2} \end{aligned}$$

and

$$\begin{aligned} A &= (1-\theta)\frac{\lambda}{2} \\ B &= 1-(1-\theta)\lambda \\ C &= (1-\theta)\frac{\lambda}{2} \end{aligned}$$

where $\lambda = \Delta t/\Delta x^2$. In each timestep, a linear system $K_1 u^m = K_2 u^{m-1}$ with tridiagonal K_1 has to be solved for u^m, similar to the implicit Euler method. This leads to the following algorithm.

3.3 Forward and backward equations

The upshot of the previous sections is that we can use finite difference schemes to approximate the transition densities of diffusion processes. The question arises if we can deduce further quantities of interest. An important example are expectations of functions of the process, not least because the majority of option pricing equations are of this form.

With the setup as previously, denote for a function G and fixed end time T

$$V_n^m = \mathbb{E}(G(X_T)|X_{t_m} = x_n).$$

One way to compute this is to use the density $U_n^m = \mathbb{P}(X_{t_m} = x_n)$, then

$$\begin{aligned} V_n^m &= \sum_{k\in\mathbb{Z}} G(x_k)\mathbb{P}(X_T = x_k|X_{t_m} = x_n) \\ &= \sum_{k\in\mathbb{Z}} G(x_k)U_{k-n}^{M-m}. \end{aligned} \tag{3.16}$$

Another approach is via

$$\begin{aligned} V_n^m &= \mathbb{E}(G(X_T)|X_{t_m} = x_n) = \sum_{k\in\mathbb{Z}} \mathbb{E}(G(X_T)|X_{t_{m+1}} = x_k) \cdot \mathbb{P}(X_{t_{m+1}} = x_k|X_{t_m} = x_n) \\ &= pV_{n-1}^{m+1} + (1-2p)V_n^{m+1} + pV_{n+1}^{m+1}. \end{aligned} \tag{3.17}$$

In fact, one gets the same formula by using the induction (2.7) for U_n^m into (3.16), which is left as an exercise. The key observation, and the main difference to the scheme for U_n^m is that we do not know any initial value U_n^0, but this is exactly what we want to compute. The one time at which we do know V_n^m is at $t_m = t_M = T$, because then

$$V_n^M = \mathbb{E}(G(X_T)|X_T = x_n) = G(x_n). \tag{3.18}$$

This leads to a backward induction for V_n^m, with m running back from $M-1$ to 0, using the terminal condition (3.18) at t_M.

Given the backward nature of the problem, the continuous-time limit is the *backward heat equation*

$$\frac{\partial v}{\partial t} + \frac{1}{2}\frac{\partial^2 v}{\partial x^2} = 0, \tag{3.19}$$

$$v(x,T) = G(x). \tag{3.20}$$

This follows from an analysis similar to the forward case.

It is well-known that the *backward* heat equation (3.19) with prescribed *initial* data is *ill-posed*, ie the solution does not depend continuously on the data. In the same way, the *forward* heat equation with prescribed *terminal* data is ill-posed. It is impossible to construct a probability distribution at an earlier time from the current distribution. The *backward* heat equation with prescribed *terminal* data however is *well-posed* in the same way the *forward* heat equation with prescribed *initial* data is well-posed. This is easily seen after noticing that time reversal $t \leftrightarrow T - t$ transforms the one into the other.

Remark 3.3.1. *Note that in spite of its resemblence of the implicit scheme for the heat equation, (3.17) is an explicit scheme in the present context. The implicit Euler scheme for the backward heat equation instead reads*

$$\frac{v_n^{m+1} - v_n^m}{\Delta t} + \frac{1}{2}\frac{v_{n+1}^m - 2v_n^m + v_{n-1}^m}{\Delta x^2} = 0,$$

where now v^m is implicitly defined in terms of v^{m+1}.

It is for this reason that the terms "forward" and "backward" Euler scheme are avoided throughout. For the (forward) heat equation, backward Euler is the implicit and therefore more stable scheme, whereas for the backward heat equation one would need the forward scheme. We therefore stick to the terminology "explicit" and "implicit" which is unambiguous.

Chapter 4

A framework for analysing difference schemes

The goal of this chapter is to develop general tools which allow us to assess discretisation schemes qualitatively and quantitatively. As an exact solution is usually unattainable, we look for strategies where the numerical solution can be made as accuurate as necessary. This gives rise to the concept of convergence. Convergence here is understood in the sense that by choosing the gridsize and timestep small enough, we can always ensure that the error measured against the exact solution of the underlying differential equation does not exceed a prescribed tolerance. This is not just of interest to the numerical analyst, but also decidedly important in practice, because it allows the financial engineer to disentangle the modelling error, i.e. the error incurred by using an inaccurate model, from the error of the numerical solution to the model equations. Even if we do not always have full control over the adequacy of a model, at least we can be sure that the numerical solution procedure does not add noticably to the error of the final result, and preserves relevant properties of the true solution adequately.

4.1 Time-stepping schemes and error propagation

For the vast majority of models in financial practice, the pricing equations are parabolic partial differential equations. They describe the evolution of a transition density or expectation forward or backward in time. The notation here is for a forward equation, but the same argument applies to backward equations, if we think of time as time-to-expiry, with expiry a given finite time-horizon.

Here, we assume an initial value at $t = 0$ to be known and are ultimately interested in the solution u at a time T. Numerical methods for these equations naturally have the structure of time-stepping schemes. In order to find an approximation to u, M discrete time steps $t_0 = 0, t_1 = \Delta t, t_2 = 2\Delta t, \ldots, t_M = M\Delta t = T$ are applied. As previously, we introduce u^m an approximation to u at time t_m. The idea is that we can approximate the equation by a more tractable one over a small time interval with sufficient accuracy.

A finite difference scheme of the form

$$u^m = Lu^{m-1}, \quad 1 \leq m \leq M,$$

with a linear operator (i.e. matrix) L takes us from an approximation at t_{m-1} to one at t_m.

The schemes introduced earlier can all be written in this form: For the implict scheme (2.28), for instance, define $L = K^{-1}$. Starting at an initial value u^0, the solution at t_M is then

$$u^M = L^{M-m}u^m = L^M u^0, \quad 0 \leq m \leq M.$$

The lowest curve on Fig. 4.1 gives an illustration of this for $M = 4$, the top curve being the true solution. We keep in the back of our mind that u usually has a continuous spatial dimension x as well, and that u^m is a vector of grid values approximating u at spatially distributed grid points. We suppress this in our notation. In this sense, we can think of Fig. 4.1 as a cross-section through the solution for fixed x, or some other derived quantity.

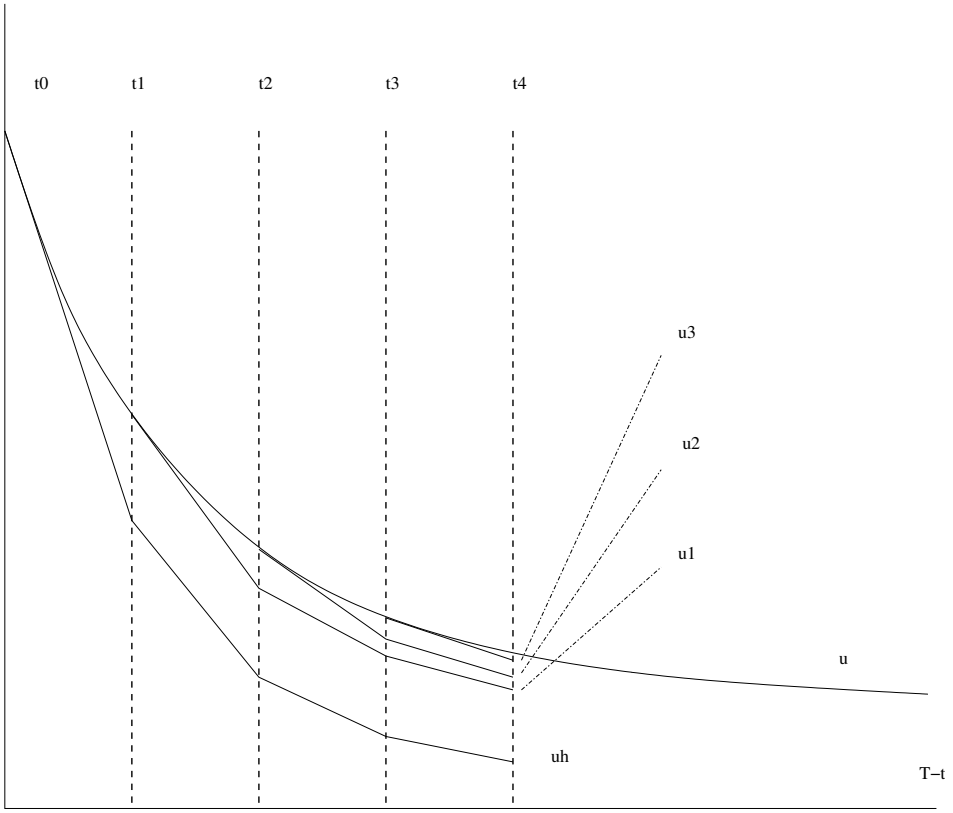

Figure 4.1: Propagation of discretisation errors for $M = 4$ timesteps.

At $t = t_0$, the solution is given by the known initial value $u(t_0)$. Stepping from $t = t_0$ to $t = t_1$, the difference scheme gives an approximation $u^1 = Lu^0 = Lu(t_0)$ to $u(t_1)$.[1] If we apply a second finite difference step at t_1, the error found at t_2 has two contributions, the error from approximating the model by a finite difference scheme in $[t_1, t_2]$, plus the error from departing at t_1 from the approximate solution u^1 rather than the exact one $u(t_1)$:

$$u^2 - u(t_2) = L\left(Lu(t_0) - u(t_1)\right) + \left(Lu(t_1) - u(t_2)\right).$$

[1]Note that $u(t_m)$ is a continuous function in the spatial direction, whereas u^m is a vector and L is a matrix acting on u^m. We understand $Lu(t_m)$ as first restricting $u(t_m)$ to the grid to obtain a grid vector, and then applying L.

Here $Lu(t_1)$ is the solution we would have obtained had we used as initial condition at t_1 the exact solution $u(t_1)$. This is somewhat hypothetical since we do not know $u(t)$ for $t > 0$, but useful for the analysis. Compare again to Fig. 4.1 for illustration.

More generally, $L^{m-j}u(t_j)$ for $M \geq m \geq j \geq 0$ is the hypothetical solution of a finite difference scheme starting from the exact solution $u(t_j)$ at t_j. Special boundary cases are

$$\begin{aligned} L^m u(t_0) &= u^m, \\ L^0 u(t_m) &= u(t_m). \end{aligned}$$

With this in mind,

$$u^M - u(T) = L^M(t_0) - L^0 u(t_M) = \sum_{m=1}^{M} L^{M-m} \left(Lu(t_{m-1}) - u(t_m) \right). \tag{4.1}$$

The error at time T is made up from the propagation of all local errors introduced before T. In each term (of the sum), the second factor measures how well the finite difference scheme tracks the continuous evolution over a single timestep. The first factor controls how these individual errors evolve over time. The art of designing a good numerical scheme lies in optimising the local approximation for small timesteps $\Delta t = T/M$, while keeping the error propagation under control when the total number of timesteps $M = T/\Delta t$ increases accordingly.

An added complexity is that a grid size Δx in the spatial direction has to be chosen suitably for every Δt, such that both go to zero following a functional relation $\Delta x = g(\Delta t)$. A good choice will balance the discretisation errors from the time- and space-discretisations, under the constraint that the scheme is stable. Take such a dependence as given as basis for the present analysis. (We will analyse this question for various finite difference schemes at length later.) We also make the dependence of L on Δt explicit now by writing $L(\Delta t)$.

From the above analysis, summarised in (4.1), we can deduce two key desiderata for a good numerical scheme:

1. *Consistency*: The error introduced in a single time step by replacing the original model with its finite difference approximation has to be small. Specifically, it has to go to zero faster than the stepsize,

$$\frac{1}{\Delta t} \left(L(\Delta t) u(t_{m-1}) - u(t_m) \right) \to 0 \text{ for } \Delta t \to 0. \tag{4.2}$$

   This is because the number of error sources is $M = T/\Delta t$, so for the accumulated error to vanish for $\Delta t \to 0$, it is necessary that each individual term goes to zero faster than Δt. Comparing with Fig. 4.1, this says intuitively that in the limit the approximation is tangential to the exact solution. Such a scheme is called *consistent* with the differential equation. The term in (4.2) is called *truncation error*.

2. *Stability*: Once a numerical error is introduced, which invariably happens, it is necessary that it does not blow up through iteration with the numerical scheme, if the number of time steps increases. This requires that

$$L(\Delta t)^M \text{ is bounded for } M \to \infty,\ \Delta t = T/M. \tag{4.3}$$

   Such a scheme is called *stable*.

From (4.1), we could potentially afford for L^M to increase as $\Delta t \downarrow 0$ if only the truncation error goes to zero fast enough. This may be true in an ideal world, but in practice other effects including boundary conditions, rounding errors etc introduce additional errors and (4.3) ensures that their influence on the final solution stays within bounds.

To make all these statements precise, we have to decide on a suitable measure for the error. The notions of convergence, boundedness etc are then to be interpreted in this sense. We analyse this a bit more formally in the next sections.

4.2 Convergence analysis of difference schemes

The pragmatic goal of a convergence analysis is to understand if and how a sufficiently "accurate" solution can be found by an appropriate choice of the numerical parameters. What exactly "accurate" means, depends on the computational question at hand.

If the desired result is a transition density u of a process X_t, there are various choices to measure the distance between u and a numerically computed density $\hat{u}$, for instance the Hellinger distance

$$d(u, \hat{u}) = \sqrt{\int_{\mathbb{R}} (\sqrt{u} - \sqrt{\hat{u}})^2 \, \mathrm{d}x}.$$

This will be difficult to come by through a numerical analysis of the PDE, and the probability density is usually used to compute some derived quantity.

We might ultimately be interested in the expectation of a function of the process at some time T, say $g(X_T)$,

$$\mathbb{E}[g(X_T)] = \int_{\mathbb{R}} g(x) p(x, T) \, \mathrm{d}x.$$

Think of X_t modelling a stock and g the payoff of an option. The error of the computation is

$$\text{error} = \int_{\mathbb{R}} g(x) \left(u(x, T) - \hat{u}(x, T)\right) \, \mathrm{d}x \leq \sqrt{\int_{\mathbb{R}} g(x)^2 \, \mathrm{d}x} \; \sqrt{\int_{\mathbb{R}} \left(u(x, T) - \hat{u}(x, T)\right)^2 \, \mathrm{d}x}.$$

Assuming the first factor is finite, e.g. for an option with bounded pay-off, a mean-square error norm would tell us how accurate the final result is.

Most commonly, the relevant result of the computation is an option price directly, by solving a Black-Scholes-type equation for instance. Typically we are most interested in the value function evaluated at the current time and the current spot price of the underlying asset. A pointwise error measure would then seem appropriate. It is however difficult to analyse the error at a single point in isolation, and we may want to reuse the solution to find the value of the solution at a later time, for a different value of the underlying process. A (usually pessimistic) upper bound which accounts for this the maximum norm.

We will develop a generic framework for the analysis first. Recall a one-step difference scheme in the form

$$u^m = L(\Delta t) u^{m-1}. \tag{4.4}$$

The igredients for the analysis are a vector norm $|\cdot|$, for the numerical solution u^m, and the associated operator/matrix norm $\|\cdot\|$ via

$$\|L\| = \max_{u, |u|=1} |Lu|,$$

for the discretisation matrix.

4.2.1 Stability

Definition 4.2.1. A scheme (4.4) is called *stable* in a norm $\|\cdot\|$ if

$$\|L(\Delta t)^m\| \leq C$$

for all $m \leq M$, $\Delta t = T/M$, and $C, T \geq 0$ are fixed.

Example 4.2.2 (Maximum stability of the θ-scheme)**.** *We have seen that discrete minimum/maximum principles,*

$$\min u^0 \leq u^m = L^m u^0 \leq \max u^0,$$

hold for the implicit scheme, and for the explicit scheme if $\Delta t \leq \Delta x^2$. Here the minimum/maximum are taken over the range of grid points, $\min u^m = \min_{n\in\mathbb{Z}} u_n^m$ etc. From

$$\begin{aligned} u^m &\leq \max u^0 \leq \max |u^0|, \\ -u^m &\leq -\min u^0 = \max -u^0 \leq \max |u^0|, \end{aligned}$$

follows

$$|L^m u^0| = |u^m| \leq \max |u^0|,$$

for all u^0 and therefore $\|L^m\| \leq 1$.

The explicit Euler scheme is conditionally stable *in the maximum norm, i.e. stable (only) if $\Delta t \leq \Delta x^2$. The implicit scheme is* unconditionally stable*.*

Now note that the θ-scheme can be seen as an explicit step of size $(1-\theta)\Delta t$, followed by an implicit step of size $\theta\Delta t$. The implict step is always stable, the explicit one if

$$(1-\theta)\Delta t \leq \Delta x^2. \tag{4.5}$$

There is a timestep constraint (4.5) for maximum norm stability of the θ-scheme for $\theta < 1$.

4.2.2 Truncation error and consistency

We cannot easily measure how well the finite difference solution satisfies the PDE, since it is only defined at discrete points. What we can measure is how well the solution to the PDE, evaluated at the grid points, satisfies the difference scheme. Notation is as in 4.1.

Definition 4.2.3 (Truncation error and consistency)**.** Let u be the solution to a PDE with time-coordinate t.

1. The *truncation error* of a one-step difference scheme of the form (4.4) is defined as

$$T(.,t) = \frac{1}{\Delta t}\left(u(.,t) - Lu(.,t-\Delta t)\right).$$

2. The scheme is called *consistent* with the PDE if

$$T \to 0$$

for $\Delta x, \Delta t \to 0$, where Δx is a spatial grid size related to Δt.

3. It is called *consistent of order* (p, q), if for sufficiently smooth u

$$T = O(\Delta x^p, \Delta t^q)$$

Consistency expresses that in the limit for vanishing grid size and timestep, the discrete equation approaches in a certain sense the continuous one. The consistency order measures the speed of this.

Example 4.2.4 (Consistency of the explicit and implicit scheme)**.** *The truncation error is easily calculated by Taylor expansion of*

$$\begin{aligned} u(x, t+\Delta t) &= u + \Delta t \frac{\partial u}{\partial t} + \tfrac{1}{2}\Delta t^2 \frac{\partial^2 u}{\partial t^2} + o(\Delta t^2), \\ u(x \pm \Delta x, t) &= u \pm \Delta x \frac{\partial u}{\partial x} + \tfrac{1}{2}\Delta x^2 \frac{\partial^2 u}{\partial x^2} \pm \frac{1}{6}\Delta x^3 \frac{\partial^3 u}{\partial x^3} + \frac{1}{24}\Delta x^4 \frac{\partial^4 u}{\partial x^4} + o(\Delta x^4), \end{aligned}$$

where arguments (x, t) of u and its derivatives are omitted on the right-hand side. In the case of the explicit scheme, directly from the definition of the truncation error,

$$\begin{aligned} T(x, t+\Delta t) &= \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t} - \frac{1}{2}\frac{u(x+\Delta x, t) - 2u(x,t) + u(x-\Delta x, t)}{\Delta x^2} \\ &= \frac{\partial u}{\partial t} - \frac{1}{2}\frac{\partial^2 u}{\partial x^2} + \frac{1}{2}\Delta t \frac{\partial^2 u}{\partial t^2} - \frac{1}{24}\Delta x^2 \frac{\partial^4 u}{\partial x^4} + o(\Delta t) + o(\Delta x^2) \qquad (4.6) \\ &= O(\Delta t) + O(\Delta x^2), \qquad (4.7) \end{aligned}$$

using in (4.6) that u is a solution to the heat equation. For the implicit scheme, is is helpful to write

$$\begin{aligned} L^{-1}T(x, t) &= \frac{u(x, t) - u(x, t-\Delta t)}{\Delta t} - \frac{1}{2}\frac{u(x+\Delta x, t) - 2u(x,t) + u(x-\Delta x, t)}{\Delta x^2} \\ &= \frac{\partial u}{\partial t} - \frac{1}{2}\frac{\partial^2 u}{\partial x^2} - \frac{1}{2}\Delta t \frac{\partial^2 u}{\partial t^2} - \frac{1}{24}\Delta x^2 \frac{\partial^4 u}{\partial x^4} + o(\Delta t) + o(\Delta x^2) \\ &= O(\Delta t) + O(\Delta x^2), \qquad (4.8) \end{aligned}$$

and now use the stability of L to deduce $T = O(\Delta t) + O(\Delta x^2)$. The explicit and implicit finite difference scheme with central differences are both consistent with the heat equation of order $p = 2$ in x and $q = 1$ in t.

Remark 4.2.5. *As concerns technique, we note from the implicit case that if a scheme is more naturally written as*

$$K_1 u^m = K_2 u^{m-1},$$

as is the case for the implicit scheme above – with $K_2 = I$, $L = K_1^{-1}$ – or more generally for the θ-scheme, it is sufficient to use Taylor expansion on the scheme in its simpler form, and then appeal to stability of the implicit part.

It is left as an exercise to show that the θ-scheme is of second order accurate in Δt if and only if $\theta = 1/2$, i.e. for the Crank-Nicolson scheme.

Another intersting observation is to be made from the analysis of the explicit scheme. Differentiating the heat equation,

$$\frac{\partial^2 u}{\partial t^2} = \frac{1}{2}\frac{\partial}{\partial t}\frac{\partial^2 u}{\partial x^2} = \frac{1}{4}\frac{\partial^4 u}{\partial x^4},$$

so by inspection of (4.6) one sees that the terms proportional to Δt and Δx^2 can be made to cancel by choosing $\Delta t/\Delta x^2 = 1/3$. The scheme is then of order $p = 2$, $q = 4$, at no extra cost. Note this choice is in the stability range $\Delta t \leq \Delta x^2$.

4.2.3 Consistency + stability = convergence

Theorem 4.2.6 (Lax Equivalence Theorem)**.** *A consistent scheme is convergent if and only if it is stable.*

Proof. We only show the direction implying convergence. For a consistent and stable scheme, it follows from (4.3) and (4.2) that

$$\begin{aligned}
|u(T) - u^M| &\leq \sum_{m=1}^{M} \|L^{M-m}\| |Lu(t_{m-1}) - u(t_m)| \\
&\leq M \max_{1\leq m\leq M} \|L^{M-m}\| \max_{1\leq m\leq M} |Lu(t_{m-1}) - u(t_m)| \\
&= T \max_{1\leq m\leq M} \|L^m\| \max_{1\leq m\leq M} \frac{1}{\Delta t} |Lu(t_{m-1}) - u(t_m)| \qquad (4.9) \\
&\to 0 \quad \text{for } \Delta t \to 0, \qquad (4.10)
\end{aligned}$$

i.e. the scheme is convergent.

□

Looking at the proof, one also sees that a a stable scheme that is consistent (of order (p, q)) is convergent (of order (p, q)).

A direct consequence is:

Corollary 4.2.7. *The explicit Euler schemes is conditionally convergent of order* $(2, 1)$*, i.e. provided* $\Delta t \leq \Delta x^2$*. The implicit Euler scheme is unconditionally convergent of order* $(2, 1)$*.*

The error can be made as small as required by reducing Δx and Δt, i.e. by increasing N and M. For stability of the explicit method, one has to choose $M \sim N^2$, whereas there is no such restriction for the fully implict scheme. This implies that if we keep doubling the number of grid points, we have to increase the number of timesteps by a factor of four each time. The error of the explicit and implicit schemes is $O(M^{-1}) + O(N^{-2})$. For both error terms to be of the same order of magnitude, it is also necessary that $M \sim N^2$. So even if there was no constraint on the stability of the method, it would be optimal to keep this relation between M and N. This puts severe constraints on the viability of the method. If we double N, we must take four times as many timesteps, and the overall computation time will be eight times as long. This is just to reduce the error by a factor of four.

For the Crank-Nicolson scheme, the error is $O(M^{-2}) + O(N^{-2})$, which would suggest and optimal choice of $M \sim N$, however (4.5) dictates $M \sim N^2$ for maximum norm stability. Experiments suggest that under certain circumstances, notably smooth initial data, we can get away without (4.5), hence improving the efficiency of the Crank-Nicolson method. We need a more refined analysis to explain this.

4.3 Spectral analysis and mean-square convergence

The previous analysis went some distance towards explaining the behaviour of finite difference schemes, however there are still at least two main open questions. The first concerns the unexplained empirical convergence of the Crank-Nicolson scheme for large timesteps in the case of smooth solutions. The second concerns the properties of solutions for non-smooth data, which are not covered by the analysis at all. While explicit and implicit schemes may have given optimism that the same criteria might hold as for the smooth case, this was not the case for the Crank-Nicolson scheme.

4.3.1 Von Neumann stability analysis

It was observed in numerical experiments earlier that instabilities resulted in the explosion of highly oscillatory modes. A (historically) very popular tack of analysis uses a more or less explicit representation of the finite difference solution, akin the classical Fourier series solution of the heat equation. This allows to investigate the dependence of amplification on the wave number. For the continuous as for the discretised form, the strategy it relies on linear constant coefficient equations to achieve a separation of variables with exponential/trigonometric simple solutions, and a superposition principle to construct more complex solutions which match initial data.

Recall the θ-scheme as

$$\frac{u_n^m - u_n^{m-1}}{\Delta t} = \frac{\theta}{2}\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\Delta x^2} + \frac{(1-\theta)}{2}\frac{u_{n+1}^{m-1} - 2u_n^{m-1} + u_{n-1}^{m-1}}{\Delta x^2}.$$

For simplicity look at the explicit scheme, $\theta = 0$, first, then

$$u_n^m = u_n^{m-1} + \frac{\lambda}{2}\left(u_{n-1}^{m-1} - 2u_n^{m-1} + u_{n+1}^{m-1}\right)$$

with $\lambda = \Delta t/\Delta x^2$. We ignore boundary conditions for now and consider the problem for $n \in \mathbb{Z}$.

We could run through a similar derivation as for the heat equation itself, but to cut to the chase guess solutions with a spatial variation of the form $\cos(kn)$, which we extend to

$$u_n^m = R_0^m \mathrm{e}^{ikn}$$

with $i = \sqrt{-1}$ to simplify the algebra with trigonometric functions. To get the original solution back simply take the real part.

From

$$\frac{1}{\Delta x^2}\left(u_{n+1}^m - 2u_n^m + u_{n-1}^m\right) = \frac{1}{\Delta x^2}\mathrm{e}^{ikn}\left(\mathrm{e}^{ik/2} - \mathrm{e}^{-ik/2}\right)^2 = -\frac{4}{\Delta x^2}\sin^2(k/2)\, u_n^m$$

one obtains

$$u_n^m = R_0(\Delta x, \Delta t; k) u_n^{m-1}$$

with

$$R_0(\Delta x, \Delta t; k) = 1 - 2\Delta t/\Delta x^2 \sin^2(k/2)$$

R is called the *symbol* of the method.

Similar to the continuous case, differencing has turned into a multiplication.

Indeed, for the θ-scheme one gets by similar reasoning (Exercise 4, Section 4.4)

$$u_n^m = R_\theta(\Delta x, \Delta t; k) u_n^{m-1}$$

with

$$R_\theta(\Delta x, \Delta t; k) = \frac{1 - 2(1-\theta)\Delta t/\Delta x^2 \sin^2(k/2)}{1 + 2\theta\Delta t/\Delta x^2 \sin^2(k/2)}. \tag{4.11}$$

If we carry on, this gives

$$u_n^M = R_\theta(\Delta x, \Delta t; k)^M u_n^0$$

We expect the method can generally only be stable, if

$$|R_\theta(\Delta x, \Delta t; k)|^M \leq C \quad \forall k \tag{4.12}$$

where C is independent of M, $\Delta t = T/M$, and Δx chosen as a function of Δt. A sufficient criterion is clearly $|R_\theta| \leq 1$. It is left as an exercise to show that (4.12) is true if and only if there is $c > 0$ such that

$$|R_\theta(\Delta x, \Delta t; k)| \leq 1 + c\Delta t.$$

Observe that always $R_\theta \leq 1$, and the difficulty comes from "overshooting" beyond -1 for large timesteps. Fig. 4.2 illustrates this for different values of θ and for increasing mesh rations $\lambda = \Delta t/\Delta x^2$.

Simple algebra shows that the explicit scheme is stable in this sense if

$$\Delta t \leq \Delta x^2$$

More generally, a sufficient criterion for the θ-scheme is

$$(1 - 2\theta)\Delta t \leq \Delta x^2. \tag{4.13}$$

This is true for all values of Δt, if and only if

$$\theta \geq \frac{1}{2},$$

then the method is unconditionally stable. The Crank-Nicolson method is the stable limiting case. For $\theta < 1/2$, there is a stability constraint on Δt. Note in how far (4.13) is an improvement to (4.5).

4.3.2 Mean-square stability

To see that this gives stability for "general" initial conditions, we can use a dicrete-continuous Fourier decomposition

$$u_n^0 = \frac{1}{2\pi\Delta x} \int_{-\pi}^{\pi} \widehat{u}^0(k) \mathrm{e}^{ink} \, \mathrm{d}k, \tag{4.14}$$

where

$$\widehat{u}^0(k) = \Delta x \sum_{n=-\infty}^{\infty} u_n^0 \mathrm{e}^{-ink}. \tag{4.15}$$

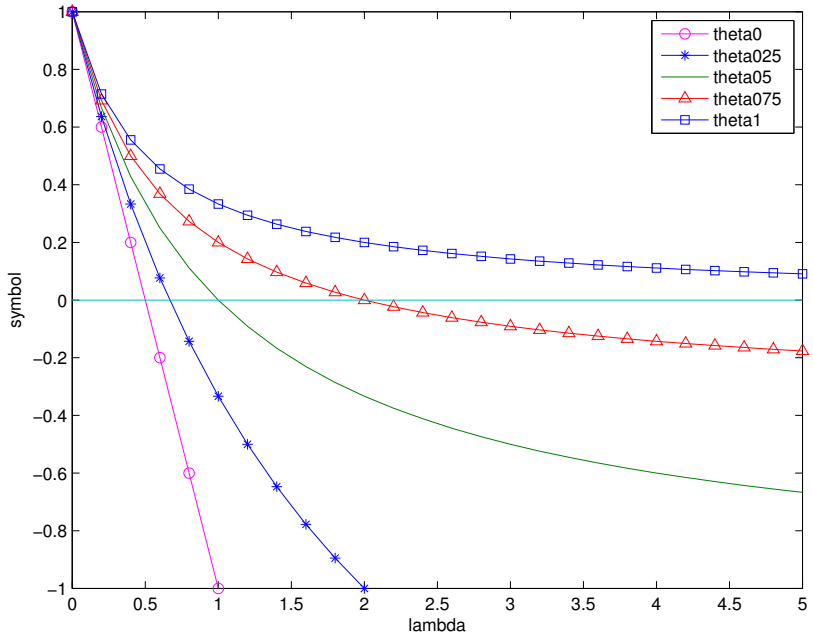

Figure 4.2: The amplification factor (4.11) for increasing mesh ratio $\lambda = \Delta t/\Delta x^2$ and different values of θ, $k = \pi$ fixed.

Then by linearity of the equation,

$$u_n^m = \int_{-\pi}^{\pi} R(\Delta x, \Delta t; k)^m \widehat{u}^0(k) \mathrm{e}^{ink} \,\mathrm{d}k. \tag{4.16}$$

The appropriate measure is a mean-square norm. For a vector $u = (u_n)_{-\infty<n<\infty}$, define

$$|u|_2 = \left(\Delta x \sum_{n=-\infty}^{\infty} u_n^2 \right)^{1/2}.$$

Then *Parseval's equality*

$$|u|_2^2 = \frac{1}{2\pi\Delta x} \int_{-\pi}^{\pi} |\widehat{u}(k)|^2 \,\mathrm{d}k$$

holds, which allows us to relate the norm of the solution to its Fourier modes,

$$\begin{aligned} |u^m|_2^2 &= \frac{1}{2\pi\Delta x} \int_{-\infty}^{\infty} |\widehat{u}^m(k)|^2 \,\mathrm{d}k \\ &= \frac{1}{2\pi\Delta x} \int_{-\infty}^{\infty} |R(\Delta x, \Delta t; k)\widehat{u}^{m-1}(k)|^2 \,\mathrm{d}k \\ &\leq \sup_k |R(\Delta x, \Delta t; k)|^2 |u^{m-1}|_2^2. \end{aligned}$$

So if $|R| \leq 1$,

$$|u^m|_2 \leq |u^0|_2$$

for $m \geq 0$.

Remark 4.3.1 (Analogy to the continuous problem)**.** *The Fourier (von Neumann) analysis of the discretisation is a semi-discrete version of the solution of the continuous equation via Fourier transform.*

$$u(x,t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \widehat{u}(k,t)\mathrm{e}^{ikx}\,\mathrm{d}k$$

where

$$\widehat{u}(k,t) = \mathrm{e}^{-\frac{1}{2}k^2 t}\widehat{u}(k,0)$$

Note that for the continuous problem with infinite range the spectrum is continuous (Fourier transform), whereas for the discrete infinite problem the spectrum is continuous but of finite range.

Note that this analysis requires that the initial condition is square integrable. In particular, it breaks down for Dirac initial data. We analyse this separately in the next chapter.

4.4 Exercises

1. Consider a solution $u : [0,T] \to \mathbb{R}$ to the scalar ODE

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= ru \\ u(0) &= 1 \end{aligned}$$

You can think of u as the money in a bank account with interest rate $r \geq 0$.

(a) Compute an approximate solution u^M at time T, for the explicit Euler method with M timesteps of length $\Delta t = T/M$, where

$$u^m = (1 + r\Delta t)u^{m-1}, \quad m > 0$$

and $u^0 = 1$. Hence show that $u^M \to \mathrm{e}^{rT}$ for $M \to \infty$ such that $T = M\Delta t$ fixed.

(b) Find bounds $c = c(T)$ and $C = C(T)$ such that

$$|(1 + rT/M)^m| \leq C \quad \forall m \leq M$$

and

$$\frac{1}{\Delta t}|(1 + r\Delta t)\mathrm{e}^{r(t-\Delta t)} - \mathrm{e}^{rt}| \leq c\,\Delta t \quad \forall t \leq T.$$

Using the Lax equivalence theorem, deduce convergence and show that a simple bound for the approximation error is given by

$$|u^M - u(T)| \leq \tfrac{1}{2}r^2\mathrm{e}^{2rT}\,\Delta t.$$

(c) Retracing the proof of the Lax equivalence theorem, by estimating carefully the propagation of truncation errors introduced at individual timesteps, show how you can improve the error bound for this specific example.

2. Show that the truncation error of the θ-scheme for the heat equation is of the form

$$(1-2\theta)O(\Delta t)+O(\Delta t^2)+O(\Delta x^2)$$

and deduce that the Crank-Nicolson scheme is of second order accurate.

Hint: Write the truncation error as

$$\delta_t^+ u(x,t)-\theta\tfrac{1}{2}\delta_x^2 u(x,t+\Delta t)-(1-\theta)\tfrac{1}{2}\delta_x^2 u(x,t)=\delta_t u(x,t)-\tfrac{1}{2}\delta_x^2\left\{\theta u(x,t+\Delta t)+(1-\theta)u(x,t)\right\}$$

where $\delta_t u(x,t)=(u(x,t+\Delta t)-u(x,t)/\Delta t$ *and* $\delta_x^2 u(x,t)=(u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t))/\Delta x^2$ *as usual. Expand* $u(x,t)$, $u(x,t+\Delta t)$, $u(x+\Delta x,t)$ *etc around* $(x,t+\theta\Delta t)$ *and use the heat equation at this point to find the truncation error.*

3. Show that if $\theta=\frac{1}{2}$ is replaced by $\theta=\frac{1}{2}-\frac{1}{12}\frac{\Delta x^2}{\Delta t}$ in the definition of the θ-scheme, the accuracy improves to $\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^4)$. Is the scheme stable, and in what sense? How should you choose Δt in relation to Δx to maximise efficiency?

4. (a) Show that the symbol of the θ-method for the heat equation is

$$R_\theta(\Delta t,\Delta x;k)=\frac{1-2(1-\theta)\lambda\sin^2(k/2)}{1+2\theta\lambda\sin^2(k/2)}$$

where $\lambda=\Delta t/\Delta x^2$.

(b) By expanding for small Δt, deduce the order of consistency of the explicit, implicit, and Crank-Nicolson schemes.

5. Given the symbol of the fully implicit method for the heat equation,

$$R_1(\Delta t,\Delta x;k)=\frac{1}{1+2\theta\lambda\sin^2(k/2)}$$

with $\lambda=\Delta t/\Delta x^2$, the finite difference solution can be written as

$$u_n^m=\int_{-\pi}^{\pi} R_1(\Delta x,\Delta t;k)^m\widehat{u}^0(k)\mathrm{e}^{ink}\,\mathrm{d}k, \tag{4.17}$$

where $\widehat{u}^0(k)$ is the Fourier transform of the discrete initial condition.

(a) Find $\widehat{u}^0(k)$ for Dirac initial data.

(b) Using calculus of residues, or otherwise, evaluate the integral in (4.17) to find u_n^m for all $m\geq 0$, $n\in\mathbb{Z}$.

(c) Compare your result to the one found in Section 2.5, Exercise 2., for $n=1$ and $n=2$.

6. Consider the semi-discrete implicit Euler scheme for the heat equation,

$$\frac{u^m-u^{m-1}}{\Delta t}=\frac{1}{2}\frac{\partial^2 u^m}{\partial x^2},\quad x\in\mathbb{R}, m\geq 1,$$

where time is discretised and the space coordinate left continuous (*horizontal method of lines*).

(a) Show that the first step with $u^0 = \delta$ has a solution of the form

$$u^1 = b \cdot \mathrm{e}^{-c|x|},$$

where you have to find b and c (compare Exercise 2, Section 2.5).

(b) By Fourier transform and calculus of residues, or otherwise, find the general solution of the semi-discrete implicit Euler scheme (compare Exercise 5 in this section). How smooth is the solution after m steps?

(c) Why would a semi-discrete explicit Euler scheme be problematic?

7. The Leapfrog scheme for the heat equation is defined as

$$\frac{u_n^{m+1} - u_n^{m-1}}{2\Delta t} = \frac{1}{2}\frac{u_{n-1}^m - 2u_n^m + u_{n+1}^m}{\Delta x^2},$$

where two initial data u_n^0, u_n^1 need to be given.

(a) Compute the truncation error at a point (x_n, t_m). Show that the scheme is consistent with the heat equation and determine the consistency order.

(b) Find all solutions of the form

$$u_n^m = R(\Delta x, \Delta t; k)^{M-m}\mathrm{e}^{ikn}$$

and give an expression for the general solution (i.e. for general u_n^0, u_n^1). Show that for any choice of u_n^1 which is consistent with the heat equation, the scheme is unstable in l_2.

8. The Du Fort and Frankel scheme for the heat equation is defined as

$$\frac{u_n^{m+1} - u_n^{m-1}}{2\Delta t} = \frac{1}{2}\frac{u_{n-1}^m - (u_n^{m-1} + u_n^{m+1}) + u_{n+1}^m}{\Delta x^2},$$

where two initial data u_n^0, u_n^1 need to be given.

(a) Find all solutions of the form

$$u_n^m = R(\Delta x, \Delta t; k)^{M-m}\mathrm{e}^{ikn}$$

and give an expression for the general solution (i.e. for general u_n^0, u_n^1). Deduce that the scheme is unconditionally stable in l_2. [2]

(b) Compute the truncation error at a point (x_n, t_m). Show that the scheme is consistent with the heat equation only if $\lim_{\Delta t, \Delta x \to 0} \Delta t/\Delta x = 0$. Which PDE is the scheme consistent with in the limit $\Delta t, \Delta x \to 0$ with $\Delta t/\Delta x = \lambda$ fixed?

(c) Explain briefly how you would choose Δt and Δx optimally. Compare the performance of the scheme to that of the explicit Euler method.

[2] See [Gordon, 1968] for maximum norm stability of the scheme.

9. This exercise offers a more abstract take on the question of convergence. Consider a "generic" equation

$$Au = b$$

where $u \in X$, $b \in Y$, X, Y linear spaces, $A \in L(X, Y)$ a linear operator from X to Y.

An approximation to this equation is written as

$$A_h u_h = b_h$$

where $u_h \in X_h$, $b_h \in Y_h$, X_h, Y_h normed linear spaces with norms $|\cdot|_{X_h}$ and $|\cdot|_{Y_h}$, and $A_h \in L(X_h, Y_h)$ a linear operator from X_h to Y_h.Assume A_h is invertible for all h, $A_h^{-1} \in L(Y_h, X_h)$ with standard operator norm $\|\cdot\|_h$.Define further operators $R_h : X \to X_h$ and $Q_h : Y \to Y_h$ and assume $b_h = Q_h b$.

Show that if

$$|Q_h A u - A_h R_h u|_{Y_h} \to 0 \quad \text{for } h \to 0$$

and if there exists a constant C with

$$\|A_h^{-1}\|_h \leq C,$$

then

$$|u_h - R_h u|_{X_h} \to 0 \quad \text{for } h \to 0.$$

Explain briefly how this relates to the convergence theory of timestepping schemes.

Chapter 5

Extensions to the basic analysis

5.1 Adding drift

Recall the advection-diffusion equation from 3.1,

$$\frac{\partial u}{\partial t} + \mu \frac{\partial u}{\partial x} = \frac{\sigma^2}{2} \frac{\partial^2 u}{\partial x^2}. \tag{5.1}$$

We focus our analysis on the case of constant μ and σ in the first instance. The solution is then the solution of the heat equation translated by an amount μt. Previously, we thought of time to be scaled such that the variance per unit time is one. We could achieve this here retrospectively by introducing a new timescale $\tau = \sigma^2 t$, as long as $\sigma \neq 0$, and write the equation in τ and x coordinates. However so as not to rule out the degenerate case $\sigma = 0$, resulting in the pure advection equation

$$\frac{\partial u}{\partial t} + \mu \frac{\partial u}{\partial x} = 0,$$

we keep σ in.

Various numerical schemes for (5.1) were introduced in 3.1. For simplicity, the analysis of the preceding sections was restricted to the heat equation though. The techniques developed there are equally applicable – with minor modifications – to the equation with drift. Indeed, we will see that

1. the order of the truncation error is additive, and therefore the analysis of local accuracy can be performed separately for each term;

2. stability is largely determined by the highest order term, here the second derivative and its discretisation. The character of the problem only changes if $\sigma = 0$, or in practice if it σ very small.

Consistency and stability lead to convergence by exactly the same mechanism studied earlier.

5.1.1 Accuracy of central differences vs upwinding

The main approximations used previously was a *central* difference, for which Taylor expansion shows

$$\frac{u(x + \Delta x, t) - u(x - \Delta x, t)}{2\Delta x} = \frac{\partial u}{\partial x}(x, t) + O(\Delta x^2),$$

i.e. second order spatial accuracy. In contrast, one-sided approximations as in

$$\begin{aligned}\frac{u(x+\Delta x,t)-u(x,t)}{\Delta x} &= \frac{\partial u}{\partial x}(x,t)+O(\Delta x),\\ \frac{u(x,t)-u(x-\Delta x,t)}{\Delta x} &= \frac{\partial u}{\partial x}(x,t)+O(\Delta x),\end{aligned}$$

are only of first order accurate. These are right- and left-sided approximations respectively. Given the nature of the equation and its solution (see 3.1), a more meaningful distinction for a particular drift μ in (5.1) is whether the finite difference is taken *in the direction of or against the drift*. For $\mu > 0$, the left-sided difference is against the drift and is called an *upwind* difference, whereas for $\mu < 0$ it is taken in the direction of the drift and is called a *downwind* difference. The opposite is the case for right-sided differences.

This leads to the definitions

$$\delta_x^- u = \operatorname{sign}(\mu)\frac{u(x,t)-u(x-\operatorname{sign}(\mu)\Delta x,t)}{\Delta x} \quad \text{(upwind difference)}, \tag{5.2}$$

$$\delta_x^+ u = \operatorname{sign}(\mu)\frac{u(x+\operatorname{sign}(\mu)\Delta x,t)-u(x,t)}{\Delta x} \quad \text{(downwind difference)}. \tag{5.3}$$

In a spirit similar to that of the θ-timestepping method, one can combine the two to

$$\begin{aligned}\mu\,\delta_x u &= \eta|\mu|\frac{u(x,t)-u(x-\operatorname{sign}(\mu)\Delta x,t)}{\Delta x}+(1-\eta)|\mu|\frac{u(x+\operatorname{sign}(\mu)\Delta x,t)-u(x,t)}{\Delta x}\\ &= \mu\,\eta\,\delta_x^- u+\mu\,(1-\eta)\,\delta_x^+ u\end{aligned}$$

with $\eta \in [0,1]$. $\eta = 0$ is downwinding, whereas $\eta = 1$ is upwinding, and $\eta = 0.5$ is the central difference scheme.

With

$$\delta_x^2 u(x,t) = \frac{u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)}{\Delta x^2} = \frac{\partial^2 u}{\partial x^2}+O(\Delta x^2)$$

the central second difference as previously,

$$\delta_t^- u = \frac{u(x,t)-u(x,t-\Delta t)}{\Delta t}$$

a time difference, Taylor expansion shows for a solution $u(x,t)$ to (5.1)

$$\begin{aligned}\delta_t^- u(x,t) &= \theta\left[\frac{\sigma^2}{2}\delta_x^2 u(x,t)-\mu\delta_x u(x,t)\right]+(1-\theta)\left[\frac{\sigma^2}{2}\delta_x^2 u(x,t-\Delta t)-\mu\delta_x u(x,t-\Delta t)\right]\\ &\quad+\underbrace{(1-2\theta)O(\Delta t)+O(\Delta t^2)+(1-2\eta)O(\Delta x)+O(\Delta x^2)}_{=R(x,t)},\end{aligned}$$

where the remainder term $R(x,t)$ is derived by Taylor expansion.

With the θ-η notation from above, this leads to a θ-timestepping finite difference scheme

$$\begin{aligned}\frac{u_n^m-u_n^{m-1}}{\Delta t} &= \theta\frac{1}{2}\frac{u_{n+1}^m-2u_n^m+u_{n-1}^m}{\Delta x^2}+(1-\theta)\frac{u_{n+1}^{m-1}-2u_n^{m-1}+u_{n-1}^{m-1}}{\Delta x^2} \qquad (5.4)\\ &- \theta\mu\frac{(1-\eta)u_{n+1}^m+(2\eta-1)u_n^m-\eta u_{n-1}^m}{\Delta x}-(1-\theta)\mu\frac{(1-\eta)u_{n+1}^{m-1}+(2\eta-1)u_n^{m-1}-\eta u_{n-1}^{m-1}}{\Delta x}.\end{aligned}$$

Proposition 5.1.1. *The scheme is of second order accurate in Δt for $\theta = 1/2$ (Crank-Nicolson), otherwise of first order; of second order in Δx for $\eta = 1/2$ (central first difference), otherwise of first order.*

5.1.2 Stability

In lose analogy with timestepping methods, upwinding corresponds to a backward difference, whereas a central spatial difference is similar to a time central difference. The expectation is therefore that upwinding will have best stability properties.

Assume $\mu > 0$. The case $\mu < 0$ is analogous, and $\mu = 0$ is trivial.

Example 5.1.2. *The explicit Euler scheme with central spatial differences,*

$$\frac{u_n^m - u_n^{m-1}}{\Delta t} + \mu \frac{u_{n+1}^{m-1} - u_{n-1}^{m-1}}{2\Delta x} = \frac{\sigma^2}{2} \frac{u_{n+1}^{m-1} - 2u_n^{m-1} + u_{n-1}^{m-1}}{\Delta x^2},$$

can be written as

$$u_n^m = \frac{\lambda}{2}\left(\sigma^2 - \Delta x\mu\right) u_{n+1}^{m-1} + (1 - \sigma^2\lambda)u_n^{m-1} + \frac{\lambda}{2}\left(\sigma^2 + \Delta x\mu\right) u_{n-1}^{m-1} \tag{5.5}$$

with $\lambda = \Delta t/\Delta x^2$.

By checking the signs of the coefficients, we read from (5.5) that the scheme is monotone and hence stable in the maximum norm if

$$\sigma^2\lambda \leq 1, \tag{5.6}$$
$$\Delta x\mu \leq \sigma^2. \tag{5.7}$$

The first condition (5.6) is identical to the purely diffusive case, except that here the variance is not normalised to one and explicitly taken into account. For the central difference scheme, there is an additional stability condition (5.7). This is not going to be critical as long as $\sigma \neq 0$, *because we want to let* $\Delta x \to 0$ *(or pick* Δx *small in practice). For* $\sigma = 0$ *(or, in practice, very small) this* is *problematic and the scheme is unstable.*

Example 5.1.3. *For* $\sigma = 0$, *i.e. the pure drift case, the explicit Euler scheme with upwind difference,*

$$\frac{u_n^m - u_n^{m-1}}{\Delta t} + \mu \frac{u_n^{m-1} - u_{n-1}^{m-1}}{\Delta x} = 0,$$

can be written as

$$u_n^m = (1 - \mu\lambda)u_n^{m-1} + \mu\lambda u_{n-1}^{m-1} \tag{5.8}$$

with $\lambda = \Delta t/\Delta x$. *The scheme is stable if*

$$\mu\lambda \leq 1.$$

This is not a very prohibitive condition because the scheme is of first order accurate in both Δt *and* Δx, *so choosing* $\Delta t \sim \Delta x$ *is appropriate. The condition says that the process does not drift more than one grid cell per timestep.*

The full finite difference scheme (5.4) can be re-written as

$$au_{n-1}^m + bu_n^m + cu_{n+1}^m = Au_{n-1}^{m-1} + Bu_n^{m-1} + Cu_{n+1}^{m-1}$$

with

$$\begin{aligned} a &= -\theta\frac{\lambda}{2}\left(\sigma^2 + \eta\Delta x\mu\right) \\ b &= 1 + \theta\lambda\left(\sigma^2 + (2\eta-1)\Delta x\mu\right) \\ c &= -\theta\frac{\lambda}{2}\left(\sigma^2 - (1-\eta)\Delta x\mu\right) \end{aligned}$$

and

$$\begin{aligned} A &= (1-\theta)\frac{\lambda}{2}\left(\sigma^2 + \eta\Delta x\mu\right) \\ B &= 1 - (1-\theta)\lambda\left(\sigma^2 + (2\eta-1)\Delta x\mu\right) \\ C &= (1-\theta)\frac{\lambda}{2}\left(\sigma^2 - (1-\eta)\Delta x\mu\right) \end{aligned}$$

We leave a more detailed stability analysis to 5.2.2 where it is somewhat better placed, and only anticipate the result. The essential requirement is that $A, B, C \geq 0$, which leads to the following stability conditions.

Proposition 5.1.4. *For $\mu \geq 0$, the θ-η scheme satisfies a discrete maximum principle if*

$$\begin{aligned} \lambda(1-\theta)\left(\sigma^2 + \Delta x\mu(2\eta-1)\right) &\leq 1, \\ (1-\eta)\mu\Delta x &\leq \sigma^2. \end{aligned}$$

For $\mu \leq 0$, replace μ by $-\mu$ and $1-\eta$ by η in the second inequality.

For $\theta = 1$ (fully implicit timestepping) and $\eta = 1$ (full upwinding) the scheme is unconditionally stable, in all other cases there are constraints on the timestep and grid size respectively. As we let $\Delta x \to 0$, the second condition will always be satisfied eventually if $\sigma > 0$. Similarly, the first condition is essentially a restricion on $\Delta t/\Delta x^2$, and the lower order term Δx does not alter this much for fine grids.

The von Neumann analysis of this discretisation is left as an exercise (Exercise 2). It produces no substantially different insights and the solution of the central difference scheme in Fourier space is now

$$\widehat{u}^m(k) = R(\Delta x, \Delta t; k)\widehat{u}^{m-1}(k) = R^m(\Delta x, \Delta t; k)\widehat{u}^0(k), \tag{5.9}$$

where the symbol, here given for the special case of central differencing

$$R(\Delta x, \Delta t; k) = \frac{1 - (1-\theta)[2\sigma^2\Delta t/\Delta x^2 \sin^2(k/2) + i\mu\Delta t/\Delta x \sin(k)]}{1 + \theta[2\sigma^2\Delta t/\Delta x^2 \sin^2(k/2) + i\mu\Delta t/\Delta x \sin(k)]}, \tag{5.10}$$

has an imaginary part. An interpretation of this is that the drift introduces a "phase shift". All schemes with $\theta \geq 1/2$ are still unconditionally stable in l_2, as was seen earlier for the heat equation. The stability conditions for $\theta < 1/2$ are also essentially identical to those for the heat equation for $\sigma > 0$, but become degenerate for $\sigma = 0$.

Remark 5.1.5. *On a final note regarding the stability of upwinding, it is instructive to write*

$$\frac{u_n^m - u_{n-1}^m}{\Delta x} = \frac{u_{n+1}^m - u_{n-1}^m}{2\Delta x} - \frac{1}{2}\Delta x\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\Delta x^2},$$

which shows that the upstream difference is a more accurate discretisation of a term

$$\frac{\partial u}{\partial x} - \frac{1}{2}\Delta x \frac{\partial^2 u}{\partial x^2}.$$

The "numerical diffusion" has a stabilising effect. This is best visible in numerical examples where the true solution is non-smooth, e.g. has discontinuities. Upwinding "smears out" out the solution.

5.2 Boundary value problems and the matrix analysis

In the previous examples and analysis, we considered random processes on infinite coordinate axes, resulting in PDEs on unbounded spatial domains, and an infinite number of coupled discretised equations in the case of implicit finite difference schemes. In practice, to make approximation schemes for this type of equations computationally tractable, we have to restrict the equations to a finite number. To "complete" the system, boundary conditions are required at the points where we truncate the coordinates. This is already present in the continuous system where an initial boundary value problem on an interval requires two boundary conditions, usually in the form of one at each end point. Since the solution is usually unknown (it is the objective of our computation!), we have to resort to approximations and the first question is how accurate these approximations are, and consequently what is the effect of these errors at the boundary on the solution in the interior. This is discussed in the next subsection.

A further question is if the introduction of boundary conditions has a noticable effect on the accuracy and stability of discretisation schemes. The answer is no and yes. Discrete maximum principles carry over easily to boundary value problems and are somewhat easier here because we have control over the boundary values via the imposed boundary conditions and do not have to consider e.g. asymptotic growth of the solution.

The spectral analysis has more differences between finite and infinite grids. On an infinite lattice, we saw that Fourier modes with arbitrary wave length retain their shape through the timesteps of a constant coefficient finite difference scheme, where they may be dampened or magnified. In other words, the infinite grid vector is the eigenvector of a linear operator (an infinite "matrix") defined by the discretisation, and the symbol is the corresponding eigenvalue. Fourier theory gives us a clear translation between grid space and frequency space. For a (finite) matrix system, there is a finite set of eigenvectors and a discrete spectrum of eigenvalues. In simple cases, we can still find these analytically, more generally we have to use approximations. More fundamentally, the question of what the eigenvalues tell us about the stability of the scheme are less clear-cut and we have to dig deeper into matrix analysis

5.2.1 Problem formulation and discretisation

An initial-boundary value problem (IBVP)

We consider the IBVP

$$\mathcal{L}u(x,t) = 0, \qquad x \in (\underline{x}, \overline{x}), t \in (0,T) \tag{5.11}$$

$$u(\underline{x}, t) = \underline{f}(t) \qquad t \in (0,T) \tag{5.12}$$

$$u(\overline{x}, t) = \overline{f}(t) \qquad t \in (0,T) \tag{5.13}$$

$$u(x,0) = g(x) \qquad x \in [\underline{x}, \overline{x}] \tag{5.14}$$

on an interval $(\underline{x}, \overline{x})$, with a (parabolic) differential operator

$$\mathcal{L}u = \frac{\partial u}{\partial t} + \mu \frac{\partial u}{\partial x} - \frac{1}{2}\sigma^2 \frac{\partial u}{\partial x}. \tag{5.15}$$

We do not necessarily assume that the coefficients are constant, but can have $\mu = \mu(x,t)$ and $\sigma = \sigma(x,t)$.

The boundary conditions are assumed of *Dirichlet* type, i.e. the value of the solution (as opposed to the derivative or other quantities) is given at the boundary points. As the solution is not known, we have to come up with approximate values. Often, an asymptotic approximation for large values of $-\underline{x}$ and $\overline{x}$ can be derived. For instance, if the initial condition g is localised (i.e. zero outside an interval), $u(x,t) \to 0$ for $x \to \pm\infty$ for all t and one can easily derive crude bounds on $u(\pm L, t)$ for large L. This allows us to choose L large enough such that $|u(\pm L, t)| \leq \epsilon$ for a desired accuracy ϵ.

The following stability result ensures that the error made by approximating the problem on the whole real line by one on a (large) finite interval is not larger than the error introduced at the boundaries.

Proposition 5.2.1. *If there are two solutions u and v, with boundary $\underline{f}_u$, $\overline{f}_u$, g_u and $\underline{f}_v$, $\overline{f}_v$, g_v respectively, then*

$$\max_{x,t} |u(x,t) - v(x,t)| \leq \max\left(\max_t |\underline{f}_u(t) - \underline{f}_v(t)|, \max_t |\overline{f}_u(t) - \overline{f}_v(t)|, \max_x |g_u(x) - g_v(x)| \right).$$

We defer the discussion of other boundary conditions to later.

Discretisation

Introducing a grid of N intervals of length $\Delta x = |\overline{x} - \underline{x}|/N$, M timesteps of length $\Delta t = T/M$, gives a discritised version of (5.11)–(5.14) as

$$Lu_n^m = 0, \qquad n \in \{1, \ldots, N-1\}, m \in \{0, \ldots, M-1\} \tag{5.16}$$

$$u_0^m = \underline{f}(t_m) \qquad m \in \{0, \ldots, M\} \tag{5.17}$$

$$u_N^m = \overline{f}(t_m) \qquad m \in \{0, \ldots, M\} \tag{5.18}$$

$$u_n^0 = g(x_n) \qquad n \in \{0, \ldots, N\} \tag{5.19}$$

where the finite difference operator at time $t_m = m\Delta t$ and grid point $x_n = n\Delta x$ reads

$$Lu_n^m = \delta_t^+ u_n^m - \left(\frac{1}{2}\sigma^2(x_n, t_{m+\theta})\delta_x^2 - \mu(x_n, t_{m+\theta})\delta_x \right)(\theta u_n^{m+1} + (1-\theta)u_n^m). \tag{5.20}$$

In the spirit of the θ-scheme, we have chosen to evaluate the time-dependent coefficients at time $t_{m+\theta} = (m+\theta)\Delta t = \theta t_{m+1} + (1-\theta)t_m$.

For $0 < n < N$, the discrete equations can be written as before,

$$a_n u_{n-1}^{m+1} + b_n u_n^{m+1} + c_n u_n^{m+1} = A_n u_{n-1}^m + B_n u_n^m + C_n u_n^m$$

where a_n, b_n, etc are given by

$$a_n = -\theta\frac{\lambda}{2}\left(\sigma^2(x_n, t_{m+\theta}) + \Delta x \mu(x_n, t_{m+\theta})\right) \tag{5.21}$$

$$b_n = 1 + \theta\lambda\sigma^2(x_n, t_{m+\theta}) \tag{5.22}$$

$$c_n = -\theta\frac{\lambda}{2}\left(\sigma^2(x_n, t_{m+\theta}) - \Delta x \mu(x_n, t_{m+\theta})\right) \tag{5.23}$$

and

$$A_n = (1-\theta)\frac{\lambda}{2}\left(\sigma^2(x_n,t_{m+\theta}) + \Delta x\mu(x_n,t_{m+\theta})\right) \tag{5.24}$$

$$B_n = 1-(1-\theta)\lambda\sigma^2(x_n,t_{m+\theta}) \tag{5.25}$$

$$C_n = (1-\theta)\frac{\lambda}{2}\left(\sigma^2(x_n,t_{m+\theta}) - \Delta x\mu(x_n,t_{m+\theta})\right) \tag{5.26}$$

We suppress the time-dependence of the factors to keep the notation simple.

Initial and boundary conditions are evaluated pointwise. The boundary values of Dirichlet type can be eliminated from the system to obtain

$$\underbrace{\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & a_{N-2} & b_{N-2} & c_{N-2} \\ 0 & \dots & 0 & a_{N-1} & b_{N-1} \end{pmatrix}}_{:=K_1} \begin{pmatrix} u_1^{m+1} \\ u_2^{m+1} \\ \vdots \\ u_{N-2}^{m+1} \\ u_{N-1}^{m+1} \end{pmatrix} = \underbrace{\begin{pmatrix} B_1 & C_1 & 0 & \dots & 0 \\ A_2 & B_2 & C_2 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & A_{N-2} & B_{N-2} & C_{N-2} \\ 0 & \dots & 0 & A_{N-1} & B_{N-1} \end{pmatrix}}_{:=K_0} \begin{pmatrix} u_1^{m} \\ u_2^{m} \\ \vdots \\ u_{N-2}^{m} \\ u_{N-1}^{m} \end{pmatrix} + d^m, \tag{5.27}$$

where

$$d^m = \begin{pmatrix} A_0u_0^m - a_0u_0^{m+1} \\ 0 \\ \vdots \\ 0 \\ A_Nu_N^m - a_nu_N^{m+1} \end{pmatrix}.$$

This is an $(N-1)\times(N-1)$ linear system to be solved at every timestep. We are less concerned with the practicalities of this at the moment than the error analysis.

Recap: Error propagation and convergence

Solving "symbolically",

$$u^{m+1} = K_1^{-1}K_0u^m + K_1^{-1}d^m = Ku^m + K_1^{-1}d^m,$$

where $K = K_1^{-1}K_0$. Recalling the definition of the truncation error as the remainder term when inserting the exact solution u in the finite difference scheme,

$$u(\cdot,t_{m+1}) = Ku(\cdot,t_m) + K_1^{-1}d^m + \Delta tT^m,$$

one gets a recursion for the error

$$e^{m+1} = u(\cdot,t_{m+1}) - u^{m+1} = Ke^m + \Delta tT^m.$$

We mention explicitly that because u_0^m and u_N^m were set equal to the Dirichlet boundary data, these terms drop out. The error at final time $T = M\Delta t$ is therefore

$$e^M = \frac{T}{M}\sum_{m=0}^{M-1} K^{M-m-1}T^m,$$

such that

$$|e^M| \leq T \max_{0\leq m\leq M} \|K^m\| \max_{0\leq m\leq M} |T^m|.$$

This is basically a rerun of the analysis in 4.1, but now u^m lives on a finite grid (i.e. in a finite-dimensional space) and we can think of K more concretely as a matrix.

Consistency (viz the truncation error) as a local quality is analysed precisely as before.

Stability can be expressed in terms of the matrix K as

$$\max_{0\leq m\leq M} \|K^m\| \leq C. \tag{5.28}$$

So far, we used matrices only notationally and as data-structures for the implementation of numerical schemes. In the next section, and especially 5.2.3, we will employ results from matrix analysis for the stability analysis of difference schemes.

5.2.2 Maximum principles and monotone matrices

We begin the stability analysis by revisiting the fully explicit and implicit schemes, but now with boundaries.

Example 5.2.2 (Explicit scheme)**.** *The explict Euler scheme for interior nodes $0 < n < N$ can be written as*

$$u_n^m = A_n u_{n-1}^{m-1} + B_n u_n^{m-1} + C_n u_{n+1}^{m-1}. \tag{5.29}$$

If

$$|A_n| + |B_n| + |C_n| \leq 1, \tag{5.30}$$

then for $0 < n < N$,

$$|u_n^m| \leq \max_{0\leq k\leq N} |u_k^{m-1}|. \tag{5.31}$$

Including boundaries,

$$\max_n |u_n^M| \leq \max\{\max_n |u_n^0|, \max_{m\leq M} |u_0^m|, \max_{m\leq M} |u_N^m|\}.$$

Remark 5.2.3. *In fact, for A_n, B_n, C_n from (5.24) to (5.26),*

$$A_n + B_n + C_n = 1, \tag{5.32}$$

and (5.32) is automatically given if

$$A_n, B_n, C_n \geq 0. \tag{5.33}$$

Conditions (5.33) provide a simple test for stability of the explicit scheme. Taking a step back and looking where (5.32) comes from, it says that the first and second finite differences of a constant are zero. This is necessary for consistency and therefore a generic property of finite difference schemes. NB: This gets lost in the presence of zero order terms.

More generally, we see from this example that a scheme of the form $u^m = Ku^{m-1}$ satisfies a discrete maximum principle, if

$$\|K\|_\infty \leq 1,$$

where

$$\|K\|_\infty = \sup_{|x|_\infty \leq 1} |Kx|_\infty = \max_i \sum_j |K_{ij}|$$

is the matrix (operator) norm associated with the maximum norm in $\mathbb{R}^{N-1}$.

Example 5.2.4 (Implicit scheme)**.** *The implict Euler scheme can be written for interior nodes $0 < n < N$ as*

$$a_n u_{n-1}^m + b_n u_n^m + c_n u_{n+1}^m = u_n^{m-1}. \tag{5.34}$$

Here want to conclude

$$\begin{aligned} |u_n^m| &= \frac{1}{|b_n|} \left| u_n^{m-1} - a_n u_{n-1}^m - c_n u_{n+1}^m \right| \\ &\leq \frac{|1 - a_n - c_n|}{|b_n|} \max\left(|u_n^{m-1}|, |u_{n-1}^m|, |u_{n+1}^m| \right) \\ &\leq \max\left(|u_n^{m-1}|, |u_{n-1}^m|, |u_{n+1}^m| \right), \end{aligned}$$

which is admissible if we assume that

$$a_n, c_n \leq 0 \tag{5.35}$$

and

$$a_n + b_n + c_n \geq 1 \quad \Leftrightarrow \quad 0 \leq \frac{1 - a_n - c_n}{b_n} \leq 1. \tag{5.36}$$

Then stability follows again inductively,

$$\max_n |u_n^M| \leq \max\{\max_n |u_n^0|, \max_{m \leq M} |u_0^m|, \max_{m \leq M} |u_N^m|\}.$$

Remark 5.2.5. *Similar to the previous example,*

$$a_n + b_n + c_n = 1, \tag{5.37}$$

implying (5.36) from (5.35). Again, this is generic for finite difference schemes of parabolic equations without zero-order term. The only conditions to check for maximum norm stability is then (5.35).

The matrix K_1 which defines the implicit scheme via $K_1 u^m = u^{m-1}$, has the following property.

Definition 5.2.6 (Diagonally dominant matrix)**.** A matrix is called *diagonally dominant* if

$$|K_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |K_{ij}| \quad \text{for} \quad 1 \leq i \leq n.$$

It is *strictly diagonally dominant* if the above inequality is strict.

Looking at the matrices in $K_1 = I + \Delta t A$, I is strictly diagonally dominant, A diagonally dominant but not strictly, and K_1 strictly diagonally dominant. A strictly diagonally dominant matrix is invertible and therefore K_1 is invertible, which we had tacidly assumed.

Another crucial ingredient to the derivation above was the "sign-condition" (5.35), which can easily be shown to guarantee monotonicity of the scheme. In particular, the solution u^m to $K_1 u^m = u^{m-1}$ with non-negative right-hand-side is non-negative. This is equivalent to K_1^{-1} being non-negative elementwise. This motivates the introduction of the following class of matrices, which we will return to later.

Definition 5.2.7 (M-matrix)**.** An $n \times n$ matrix $K = (K_{ij})_{1 \le i,j \le n}$ is called an *M-matrix* if it satisfies the following two conditions:

1. $K_{ij} \le 0$ for $i \neq j$, $1 \le i, j \le n$
2. K is invertible and $(K^{-1})_{ij} \ge 0$ for $1 \le i, j \le n$

An equivalent characterisation is given (see the remarkable Theorem 5.1.1 on page 129 in [Fiedler, 2008]) if instead of property 2. in Definition 5.2.7 one demands

$$\exists x \ge 0 : \quad Kx > 0. \tag{5.38}$$

This is sometimes easier to verify in practice, as one only has to find one particular x. In particular, by choosing $x = (1, \dots, 1)$ one finds that a strictly diagonally dominant matrix which has positive diagonal $K_{ii} > 0$, $1 \le i \le n$, and satisfies 1. in Definition 5.2.7, is an M-matrix. M-matrices are positive definite (and, by definition, invertible).

Consider now the matrix norm

$$\|K\|_\infty \equiv \max_{|x|_\infty = 1} |Kx|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |K_{ij}|$$

associated with the ∞-vector norm.

For an M-matrix it is then true that,

$$\|K^{-1}\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |(K^{-1})_{ij}|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} (K^{-1})_{ij}$$

and that

$$K(1, \dots, 1)' \ge (1, \dots, 1)' \qquad \Rightarrow \qquad K^{-1}(1, \dots, 1)' \le (1, \dots, 1)',$$

hence

$$\sum_{j=1}^{n} K_{ij} \ge 1 \text{ for all } i \qquad \Rightarrow \qquad \|K^{-1}\|_\infty \le 1.$$

It therefore follows that for an M-matrix with row sums greater than 1,

$$K u^{m+1} = u^m \qquad \Rightarrow \qquad \|u^{m+1}\|_\infty \le \|u^m\|_\infty.$$

The l_∞ norm often gives an overly pessimistic assessment of stability, and we therefore study the l_2 norm in comparison.

5.2.3 Eigenvalue analysis and mean-square convergence

Going back to (5.28), we concluded a numerical scheme is going to be stable, if the sequence K^j is bounded in some sense. There are two complementary viewpoints. We can think of Δt fixed and let the number M of timesteps go to infinity, i.e. moving the final time ahead. Then K is fixed and for a bounded solution it is necessary that

$$\|K^m\|^{1/m} \leq C^{1/m} \to 1, \quad m \to \infty.$$

The following notion is therefore useful here.

Definition 5.2.8 (and Lemma, Spectral radius)**.** The *spectral radius* ρ of an $n \times n$ matrix K, with eigenvalues (spectrum) $\lambda_1, \ldots, \lambda_n$, is defined by either of the following equivalent definitions:

1. $\rho(K) = \max_{i=1}^{n} |\lambda_i|$,

2. $\rho(K) = \lim_{j\to\infty} \|K^j\|^{1/j}$,

where $\|\cdot\|$ is an arbitrary matrix norm.

For the solution to be bounded in perpetuity, we need the spectral radius to be smaller or equal to one. This is only sensible to ask for if the solution to the underlying PDE itself does not grow unbounded in time.

In practice, we are usually more interested in the situation where we consider a fixed time horizon T, and want to improve the accuracy of a numerical approximation by increasing the number M of timesteps *within* this interval, hence letting $\Delta t = T/M \to 0$ for *fixed* final time T. In that case we want

$$\|K(\Delta t)^m\| \leq C \ \ \forall m \in \mathbb{N},$$

where Δt is linked to m via $\Delta t = t/m$. A *sufficient* condition for stability is then

$$\|K(\Delta t)\|^m \leq C \ \ \forall m \in \mathbb{N} \quad \Leftrightarrow \quad \|K\| \leq 1 + c\Delta t$$

for a constant c. The scheme is unconditionally stable if c is independent of Δx.

The question arises if this condition is *necessary*. A necessary condition is clearly $\rho(K) \leq 1 + c\Delta t$, otherwise components parallel to the eigenvector with the largest eigenvalue will blow up. To illustrate this, assume for the sake of the argument that the eigenvectors of K, say V_j with eigenvalues λ_j, form a basis of $\mathbb{R}^n$, then if $V = \sum_{j=1}^{k} \mu_j V_j$,

$$K^m V = \sum_{j=1}^{k} \lambda_j^m \mu_j V_j.$$

We have thus a necessary condition for stability expressed in the eigenvalues, and a sufficient one expressed in the norm.

Technical point: Norm vs eigenvalues

Only in special situations is the norm of a matrix K identical to the spectral radius. One important class of matrices for which this is true are *normal* matrices, defined by the property $KK' = K'K$, where the prime denotes the matrix transpose. Specifically this is true if K is *symmetric*, $K = K'$. For normal (symmetric) matrices, there is a *unitary (orthogonal) transformation* $U^*KU = D$ to diagonal form, with a *unitary* (orthogonal) matrix U, i.e. one with $U^* = \overline{U}' = U^{-1}$. D then contains the eigenvalues of K and $\|K\|_2 = \rho(K)$.

However, analysing only the eigenvalues may be misleading. The eigenvectors of the matrix may not be independent, i.e. the matrix not diagonalisable. In this scenario, the spectral radius may give no indication of the "size" of the matrix. (For instance, it can be zero for non-zero matrices.) To add a further complication, if we are looking to unconditional stability across different grids, the dimension of the vectors grows when Δx is refined along with Δt, so we are comparing solutions in different spaces. The basis transformation with the eigenvector matrix $V = (V_1, \ldots, V_n)$ – when it exists – takes place in different spaces and generally the norm of V will depend on the grid size Δx. Normal matrices are an exception because the unitary transformations have unit norm independent of the dimension.

The bad news is that the only normal matrices we will encounter are the disretisation matrices for the heat equation. These are in fact symmetric. The good news is that for many examples we will find, e.g., diffusion equations with drift and even with variable coefficients, the discretisation matrices are "almost" normal. In essence, the reason for this is that the leading order term comes from the highest derivative, the diffusion, and if the coefficients vary smoothly these can be seen as almost constant for fine grids.

Summing up, the eigenvalues of the discretisation matrix are going to be useful, if they describe the asymptotic behaviour of the scheme. In practice, they can often be estimated more easily than the l_2 norm itself. We will now investigate an example where the eigenvalues and eigenvectors are known explicitly.

Examples and applications

We start with a special case of the matrices in (5.27),

$$K = \mathrm{diag}(a, b, c, n) = \begin{pmatrix} b & c & & \ldots & 0 \\ a & b & c & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & a & b & c \\ 0 & \ldots & 0 & a & b \end{pmatrix}, \tag{5.39}$$

where the entries are constant along the diagonals. Matrices of this type are called *Töplitz* matrices. This particular structure of a tridiagonal Töplitz matrix arises for a finite difference discretisation of the differntial operator (5.15) with constant coefficients μ and σ.

Example 5.2.9. *The $n \times n$ matrix $K = \mathrm{diag}(a, b, c, n)$ has eigenvalues*

$$\lambda_k = b + 2\sqrt{ac}\cos(k\xi) \tag{5.40}$$

and corresponding eigenvectors

$$U_k = \left(r\sin(k\xi), r^2\sin(2k\xi)\ldots, r^n\sin(k\xi n)\right) \tag{5.41}$$

where $r = \sqrt{a/c}$ and $\xi = \pi/(n+1)$.

Note that K in (5.39) is *not* normal if $a \neq c$, i.e. if K is not symmetric. To see this, it is sufficient to calculate, for $n = 2$,

$$(KK')_{11} = b^2 + c^2 \neq a^2 + b^2 = (K'K)_{11},$$

and similarly for the second diagonal entry. Exercise 5 in 5.4 shows that this is only due to the boundaries (the 2×2 system has only "boundaries") and the discretisation of the constant coefficient PDE as such is "normal". This is no longer the case for variable coefficients.

Example 5.2.10. *Consider K_0 and K_1 from (5.27) with constant coefficients σ and μ. We analyse the spectral radius of the iteration matrix $K = K_1^{-1}K_0$ and want to show for the spectrum $\mathcal{S}$, i.e. the set of all eigenvalues,*

$$\mathcal{S}(K) \subset (-1, 1] \quad \textit{for} \quad \theta \geq 1/2. \tag{5.42}$$

Assume $\sigma^2 < \Delta x|\mu|$ such that $a, c < 0$, then the eigenvalues of K_1 are

$$\lambda_k = 1 + \theta\lambda\sigma^2(1 + \cos(k\xi)\sqrt{1 - \Delta x^2\mu^2/\sigma^4}) > 1. \tag{5.43}$$

From (5.43), $\mathcal{S}(K_1) \subset (1, \infty)$, follows from the inverse eigenvalue theorem $\mathcal{S}(K_1^{-1}) \subset (0, 1]$, which proves the statement (5.42) for the fully implicit scheme. We now show that (5.42) is also true for K from the θ-scheme for $\theta \geq 1/2$. Define A such $K_1 = I + \Delta t\theta A$, then

$$\begin{aligned} K_0 &= I - (1-\theta)\Delta tA = \frac{1}{\theta}I - \frac{1-\theta}{\theta}K_1, \\ K &= K_1^{-1}K_0 = \frac{1}{\theta}K_1^{-1} - \frac{1-\theta}{\theta}I. \end{aligned}$$

Now from the inverse eigenvalue theorem,

$$\mathcal{S}(K_1^{-1}) \subset (0, 1] \;\Rightarrow\; \mathcal{S}(K) = \frac{1}{\theta}\mathcal{S}(K_1^{-1} - (1-\theta)I) \subset (-\frac{1-\theta}{\theta}, 1] \subset [-1, 1].$$

Theorem 5.2.11. *Consider the IBVP (5.11) to (5.14) with constant coefficients $\sigma > 0$ and μ, such that the coefficients in the θ-central difference scheme (5.16) to (5.19), (5.27) with entries (5.21) to (5.26), are independent of n and m. Then the scheme is*

1. *unconditionally stable (convergent) in the $|\cdot|_2$ norm for $\theta \in [\frac{1}{2}, 1]$, which includes the Crank-Nicolson scheme $\theta = 1/2$;*
2. *conditionally stable (convergent) for $\theta \in [0, \frac{1}{2})$.*

Proof. We only cover in detail the case $\theta \in [\frac{1}{2}, 1]$ and leave the other case as an exercise. We have already shown in Example 5.2.10 that the eigenvalues of the matrix K are contained in $(-1, 1)$. We now deduce that $\|K(\Delta t)^M\|$ is bounded.

Assume Δx sufficiently small, such that $c < a < 0$. Denote by $\mathcal{S}(K)$ again the spectrum of a matrix K. We first observe that the eigenvectors of K_1 are identical to those of K_0, K_1^{-1} and consequently to those of $K = K_1^{-1}K_0$. This means that K is diagonalised by the matrix of eigenvectors U_j from Example 5.2.9 with $U_{ij} = (r^i \sin(ij\xi))$, $K = UDU^{-1}$. Writing $U = RQ$ with $R = \mathrm{diag}(r^i, n)$ and Q orthogonal,

$$V^m = K^m V^0 = UD^m U^{-1} V^0 = RQD^m Q' R^{-1} V^0. \tag{5.44}$$

From

$$r = \sqrt{\frac{1 + \Delta x \mu/\sigma^2}{1 - \Delta x \mu/\sigma^2}}, \tag{5.45}$$

Taylor expansion shows $1 - 3\Delta x|\mu|/\sigma^2 \le r \le 1 + 3\Delta x|\mu|/\sigma^2$ and $\exp(1 - 4|\mu|/\sigma^2) \le r^n \le \exp(1 + 4|\mu|/\sigma^2)$ for sufficiently small Δx, so

$$\|R\|, \|R^{-1}\| \le \exp(1 + 4|\mu|/\sigma^2) \le C.$$

The middle symmetric part QD^mQ' of the matrix product in (5.44) has norm $\max_k(|D_{kk}|) = \rho(K) \le 1$, so

$$|V^m|_2 \le C|V^0|_2.$$

□

Remark 5.2.12. *The technique adopted here, i.e. transformation of the original problem to a simpler (e.g. symmetric one) with a well-behaved matrix, is a useful tool to keep in mind. The reason the transformation is stable in the above example is that the matrix is almost symmetric, in the sense that the ratio r of the off-diagonals (5.45) is close to one. Similarly, for smoothly varying coefficients of the PDE, neighbouring diagonal entries are close. For an application to other non-symmetric problems, including ones with variable coefficients, see Exercises 6 and 7 in 5.4.*

In most practically relevant cases, the eigenvalues of the discretisation matrices are not known, because usually the coefficients vary with the coordinates and with time. The following result is very useful for estimating the size of the eigenvalues of a matrix in such cases.

Lemma 5.2.13 (Gershgoring circle theorem)**.** *Let $K \in \mathbb{R}^{n\times n}$ (or $\mathbb{C}^{n\times n}$), and define*

$$R_i = \sum_{j \ne i} |K_{ij}|.$$

1. *Each eigenvalue λ_j lies in at least one Gershgorin disk*

$$D_i = D(K_{ii}; R_i)$$

   in the complex plane, with centre K_{ii} and radius R_i.

2. *If a union of i discs is disjoint from the other disks, then this union contains exactly i eigenvalues.*

Remark 5.2.14. *It is probably clear by now that the finite-difference stability analysis of general initial-boundary value problems with variable coefficients is technically more involved than the simple von Neumann analysis. As a rule of thumb, the von Neumann stability analysis of a constant-coefficient problem of the same type normally gives a very good indication for stability of a scheme, and in practice suffices as a test.*

5.3 Non-smooth data and stronger stability concepts

The previous analysis of consistency and stability builds on regularity of the solution: smoothness of sufficient order is needed to compute the truncation error. The most important applications in finance lack this regularity, as a result of non-smooth data. Moreover, solution and error are measured in certain norms, so at the very least the data have to lie locally in L_∞ or L_2. For Dirac initial data, even this is not the case. Fortunately, the smoothing properties of the PDE usually mean that these effects are seen only at isolated points in time, and it is perfectly possible to compare the exact and numerical solutions at a given time away from the singular points.

A prime case in point is the initial-value problem for the heat equation on the real line, with solution

$$u(x,t) = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} u_0(y) \mathrm{e}^{-(x-y)^2/2t} \,\mathrm{d}y.$$

For reasonably well-behaved initial data u_0, the solution is infinitely smooth at any time $t > 0$. An interesting angle is given by the Fourier representation of u,

$$u(x,t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{u_0}(k) \mathrm{e}^{-\frac{1}{2}k^2 t} \mathrm{e}^{ikx} \,\mathrm{d}k, \tag{5.46}$$

where $\widehat{u_0}$ is the Fourier transform of u_0. The wave components e^{ikx} of wave number k are dampened by $\mathrm{e}^{-\frac{1}{2}k^2 t}$, i.e. decay exponentially in time and increasingly fast for high k. This corresponds to the intuition that diffusion smoothens out highly oscillatory components rapidly.

In comparison, for the θ-central difference scheme, the damping factor is

$$R_\theta(\Delta x, \Delta t; k) = \frac{1 - 2(1-\theta)\Delta t/\Delta x^2 \sin^2(k/2)}{1 + 2\theta \Delta t/\Delta x^2 \sin^2(k/2)}. \tag{5.47}$$

There is also time exponential decay by R_θ^m after m timesteps, if $|R_\theta(\Delta x, \Delta t; k)| < 1$. We now scrutinize this in more detail for the Crank-Nicolson scheme. Given a consistency order of two in both space and time, one would like to pick $\Delta t = O(\Delta x)$. It is therefore practically relevant to analyse the behaviour as $\Delta t/\Delta x^2 \to \infty$, and one gets

$$R_{\frac{1}{2}}(\Delta x, \Delta t; k) \to -1 \quad \text{for} \quad \Delta t/\Delta x^2 \to \infty,$$

such that high frequency components maintain their amplitude persistently for fine meshes.

So refining the stability condition $|R_\theta(\Delta x, \Delta t; k)| < 1$, we want this to be true uniformly for all k, i.e.

$$|R_\theta(\Delta x, \Delta t; k)| < q \leq 1 \quad \forall\, 0 < k_0 \leq |k| \leq \pi,\ \Delta x, \Delta t \geq 0, \tag{5.48}$$

with Δt chosen in relation to Δx and k_0 a small number. The low frequencies $|k| \leq k_0$ correspond to the smooth component of the solution. Rearranging (5.47), (5.48) is guaranteed for $(1-2\theta+\epsilon)\Delta t \leq \Delta x^2$ for some small ϵ. This is automatically satisfied for $\theta > 1/2$. However, notably for the Crank-Nicolson scheme $\theta = 1/2$, there is a restriction $\Delta t = O(\Delta x^2)$ which compromises the efficiency of the scheme.

For implicit Euler, $\theta = 1$,

$$R_1(\Delta x, \Delta t; k) \to 0 \quad \text{for} \quad \Delta t/\Delta x^2 \to \infty \tag{5.49}$$

such that "high" frequencies with $k\Delta x = O(1)$ are dampened rapidly over timesteps. This comes at the expense of reduced accuracy for low wave-numbers. The implicit scheme is in a sense too diffusive. In contrast, the Crank-Nicolson scheme is not diffusive enough, which gives unrealistic solutions for non-smooth data.

In the following, we analyse this more precisely and then propose modifications which address this problem. For many of the data encountered in financial engineering, the irregularities are localised and appear in the form of kinks or discontinuities of an otherwise smooth function, or as Dirac components. This allows us to study these effects in isolation, given the linearity of the equations, and we can think of decomposing the data into smooth components and specific singularities of the above type. We start with the most severe, Dirac distributions. We have seen that they come up naturally in models for probability densities.

5.3.1 Dirac initial data and θ-schemes

We focus on the IBVP

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{1}{2}\frac{\partial^2 u}{\partial x^2}, \quad x \in \mathbb{R},\ t > 0, \\ u(\cdot, 0) &= \delta. \end{aligned}$$

A Dirac distribution as initial data is not even square integrable and the previous analysis of the error uniformly in time breaks down. We will instead use exact formulae for the analytic and finite difference solutions to the heat equation at a fixed finite time, which are known in this specific case, and compare the two to compute the discretisation error. This is clearly a stylized problem but we argue that the qualitative behaviour is similar in more general situations: in the presence of drift, boundary conditions, and even variable coefficients. This is because accuracy and stability are determined *locally* and we can therefore imagine "freezing" the coefficients at a constant level in a neighbourhood of the point of interest. Similarly, the effect of boundary conditions can be analyzed separately to this question. Finally, stability is influenced predominantly by the highest order term in the equation, which here is the second order diffusion term.

So, for a one-step finite difference scheme to the heat equation, recall from (4.16) in 4.3.2 that there is a Fourier representation

$$u_n^m = \int_{-\pi}^{\pi} R(\Delta x, \Delta t; k)^m \widehat{u}^0(k) \mathrm{e}^{ink} \, \mathrm{d}k. \tag{5.50}$$

In particular, for

$$u_n^0 = \frac{1}{\Delta x}\delta_{n0},$$

a discrete approximation to the Dirac distribution as in (2.16), it follows from (4.15) that

$$\widehat{u}^0(k) = \Delta x \sum_{n=-\infty}^{\infty} u_n^0 \mathrm{e}^{-ink} = 1.$$

Note that $u^0 \in l_2$, somewhat breaking the analogy with the continuous problem where $u(\cdot, 0) = \delta \notin L_2$. From $|u^0|_2 = 1/\Delta x$ it is clear however that the limit $\Delta x \to 0$ is in some sense singular. The inversion formula (4.14) is easily verified here as

$$u_n^0 = \frac{1}{2\pi\Delta x}\int_{-\pi}^{\pi} \widehat{u}^0(k)\mathrm{e}^{ink}\,\mathrm{d}k = \frac{1}{2\pi\Delta x}\int_{-\pi}^{\pi} \mathrm{e}^{ink}\,\mathrm{d}k = \frac{1}{\Delta x}\delta_{n0}.$$

From (4.16),

$$u_n^m = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{\Delta x} R(\Delta x, \Delta t; k)^m \mathrm{e}^{ink}\,\mathrm{d}k = \frac{1}{2\pi}\int_{-\pi/\Delta x}^{\pi/\Delta x} \widehat{u}^m(k)\mathrm{e}^{ikx_n}\,\mathrm{d}k.$$

This compares to the known analytical solution in the form

$$u(x_n, t_m) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \widehat{u}(t_m, k)\mathrm{e}^{ikx_n}\,\mathrm{d}k = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{e}^{-\frac{1}{2}k^2 m\Delta t}\mathrm{e}^{ikx_n}\,\mathrm{d}k.$$

For the θ-scheme, specifically, from (4.11),

$$\widehat{u}^m(k\Delta x) = R_\theta(\Delta x, \Delta t; \Delta x k)^m = \left(\frac{1 - 2(1-\theta)\Delta t/\Delta x^2 \sin^2(k\Delta x/2)}{1 + 2\theta\Delta t/\Delta x^2 \sin^2(k\Delta x/2)}\right)^m, \tag{5.51}$$

which has to be measured against

$$\widehat{u}(t_m, k) = \exp(-\tfrac{1}{2}k^2 m\Delta t). \tag{5.52}$$

From $(1 + x/m)^m \to \exp(x)$ for $m \to \infty$, it looks reasonably promising that the difference between (5.51) and (5.52) can be computed up to a certain order of Δt and Δx by Taylor expansion, as long as $k\Delta x$ is "small". We look at this regime first. For "large" k, the analytical solution goes to 0 exponentially fast and we have to make sure the timestepping scheme dampens these components at least to some order in $\Delta t, \Delta x$ as well (see the introductory comments). This is analysed in the subsequent section.

Low wavenumber range

We Taylor-expand in Δt and Δx the expression

$$\begin{aligned}\log(\widehat{u}^m(k\Delta x)) &= m\log(1 - 2(1-\theta)\Delta t/\Delta x^2 \sin^2(k\Delta x/2)) - m\log(1 + 2\theta\Delta t/\Delta x^2 \sin^2(k\Delta x/2)) \\ &= -\frac{1}{2}\,t\,k^2 + \frac{1}{24}\,t\,\Delta x^2\,k^4 + \frac{1}{8}(\theta^2 - (1-\theta)^2)\,t\,\Delta t\,k^4 - \frac{1}{4}(\theta^3 + (1-\theta)^3)\,t\,\Delta t^2\,k^6 \\ &\quad + O(\Delta x^3 k^6) + O(\Delta t^3 k^8).\end{aligned}$$

As expected, the error is of second order in Δt exactly if $\theta = 1/2$, otherwise of first order. In the first case, we want to pick $\Delta t \sim \Delta x$ to balance leading order error terms in Δt and Δx^2, otherwise $\Delta t \sim \Delta x^2$. For the first remainder term to go to zero for $\Delta x \to 0$, we need

$k = o(\Delta x^{-1/2})$, for the second $k = o(\Delta t^{-3/8})$. Depending on whether $\Delta t \sim \Delta x^2$ or $\Delta t \sim \Delta x$, the first or second condition will be stringent.

Taking the exponential and Taylor-expanding again,

$$\begin{aligned}\widehat{u}^m(k\Delta x) &= \exp(-\tfrac{1}{2}k^2 t)\left(1 + \Delta t(2\theta-1)\widehat{a}_1^m(k) + \Delta t^2 \widehat{a}_2^m(k) + \Delta x^2 \widehat{a}_3^m(k)\right. \\ &\qquad \left. + O(\Delta x^3 k^6) + O(\Delta t^3 k^8)\right), \quad (5.53)\end{aligned}$$

with suitably defined $\widehat{a}_1^m$, $\widehat{a}_2^m$, $\widehat{a}_3^m$.

Medium to high wavenumber range

Now looking at the behaviour for larger k, the first observation is that R_θ is decreasing in k. We argued earlier in 4.3.2 that for u^0 in l_2, i.e. square summable initial data, with $|u^0|_2$ bounded independent of Δx, the condition

$$(1-2\theta)\frac{\Delta t}{\Delta x^2} \leq 1$$

guarantees stability and consequently convergence. The above prerequisites on u^0 are not met here. In this case, we have to make sure that high wavenumber components are dampened sufficiently. Now we look at a number of key cases more closely.

For the fully implicit scheme, it still holds that $0 < R < 1$ for all $k \neq 0$. Because this term comes up so often, we introduce $f(y) = \sin^2(y/2)/(y/2)^2$ with $y \in [-\pi, \pi]$ and then $f(y) \in [4/\pi^2, 1]$. This implies

$$\widehat{u}^m(k\Delta x) = (1 + \tfrac{1}{2}k^2\Delta t f(k\Delta x))^{-m} \to \exp(-\tfrac{1}{2}k^2 t f(k\Delta x)) \quad \text{for } m \to \infty, \qquad (5.54)$$

where $f(k\Delta x)$ is $O(1)$. In the range $\Delta x^{-p} \leq k \leq \pi/\Delta x$ for any $0 < p < 1$ the convergence to 0 is faster than Δx^r for any order $r > 0$.

An identical line of reasoning shows that for *any* θ, as long as $0 \leq R$, i.e. if $2(1-\theta)\Delta t/\Delta x^2 \leq 1$, the convergence in the large wave number range is exponential. This guarantees e.g. convergence of the Crank-Nicolson scheme for $\Delta t \leq \Delta x^2$.

If $R \geq 0$ is not ensured, then there is a zero of R where

$$\tfrac{1}{2}(1-\theta)\Delta t k^2 f(k\Delta x) = 1,$$

which appears at

$$k_0 = O(\Delta t^{-1/2}).$$

From the previous reasoning, we are covered for $k \leq k_0$. If we can ensure that

$$|R_\theta| \leq q < 1 \qquad (5.55)$$

for $k \geq k_0$, then surely in this range

$$|\widehat{u}^m(k)| \leq q^{-t/\Delta t} = o(\Delta t^r) = o(\Delta x^r)$$

for all $r > 0$.

The inequality (5.55) is given for $\theta > 1/2$ for all combinations of Δt and Δx, but *not* for the Crank-Nicolson scheme, $\theta = 1/2$, where $R_{1/2}(\Delta x, \Delta t; k) \to -1$ for $k \to \pm\pi$, $\Delta t/\Delta x^2 \to \pm\infty$.

Putting it all together

Summing up, if

$$1/2 < \theta \leq 1 \quad \text{or} \quad 2\Delta t \leq \Delta x^2 \text{ for } 0 \leq \theta \leq 1/2, \tag{5.56}$$

the approximation error is

$$u(x_n, t_m) - u_n^m = \frac{1}{2\pi} \int_{|k|>\pi/\Delta x} \mathrm{e}^{-\frac{1}{2}k^2 t_m} \mathrm{e}^{ikx_n} \,\mathrm{d}k + \frac{1}{2\pi} \int_{-\pi/\Delta x}^{\pi/\Delta x} (\widehat{u}(k, t_m) - \widehat{u}^m(k\Delta x)) \,\mathrm{e}^{ikx_n} \,\mathrm{d}k,$$

where

$$\Big| \int_{|k|>\pi/\Delta x} \mathrm{e}^{-\frac{1}{2}k^2 t_m} \mathrm{e}^{ikx_n} \,\mathrm{d}k \Big| \leq \int_{|k|>\pi/\Delta x} \mathrm{e}^{-\frac{1}{2}k^2 t_m} \,\mathrm{d}k = o(\Delta x^r)$$

for all $r > 0$ and

$$\int_{-\pi/\Delta x}^{\pi/\Delta x} (\widehat{u}(k, t_m) - \widehat{u}^m(k\Delta x)) \,\mathrm{e}^{ikx_n} \,\mathrm{d}k = \Delta t(2\theta - 1) a_1^m(x_n) + \Delta t^2 a_2^m(x_n) + \Delta x^2 a_3^m(x_n) + O(\Delta x^3) + O(\Delta t^3), \tag{5.57}$$

where

$$a_j^m(x) = \frac{1}{2\pi} \int_{-\pi/\Delta x}^{\pi/\Delta x} \widehat{a}_j^m(k) \mathrm{e}^{-\frac{1}{2}k^2 t_m} \mathrm{e}^{ikx} \,\mathrm{d}k.$$

In short,

$$u(x_n, t_m) - u_n^m = (1 - 2\theta) O(\Delta t) + O(\Delta t^2) + O(\Delta x^2), \tag{5.58}$$

provided the stronger "stability"[1] constraint (5.56) is satisfied.

In particular:

- As expected, the fully implicit scheme converges of first order in Δt and second order in Δx, unconditionally.
- Also, unsurprisingly, the explicit Euler scheme converges also of first order in Δt, but only if a timestep constraint $2\Delta t \leq \Delta x^2$ is fulfilled.
- A genuinely new phenomenon is encountered for the Crank-Nicolson scheme which is not unconditionally convergent in l_2 for Dirac initial data, even though it is unconditionally stable in the l_2 sense.

In the following, we will introduce notions of stability which classify exactly this behaviour. We will also present a few solutions to the practical problem of constructing a second order convergent scheme which works for non-smooth, especially Dirac, initial data.

[1] The term stability constraint is not entirely accurate, because it is incorrect to say that the scheme is unstable if these are violated, rather the data and the solution lie outside the class of functions where the standard stability analysis is applicable.

5.3.2 Stronger stability of time-stepping schemes

The last section provided ample evidence that the stability of time-stepping schemes is characterised by the amplification or damping of highly oscillatory components. These are best singled out by studying the equations in Fourier space. We have in mind in the following time-stepping schemes for the heat equation with drift, as in (5.1). For the θ-central difference scheme, for instance, Exercise 2 in 5.4 shows that on an infinite grid,

$$\widehat{u}^m(k) = R(\Delta x, \Delta t; k)\widehat{u}^{m-1}(k) = R^m(\Delta x, \Delta t; k)\widehat{u}^0(k), \tag{5.59}$$

where

$$R(\Delta x, \Delta t; k) = \frac{1 - (1-\theta)[2\sigma^2 \Delta t/\Delta x^2 \sin^2(k/2) + i\mu\Delta t/\Delta x \sin(k)]}{1 + \theta[2\sigma^2 \Delta t/\Delta x^2 \sin^2(k/2) + i\mu\Delta t/\Delta x \sin(k)]}$$

is the amplification facor or symbol. One identifies the lengthy expression in $-[\ldots]$ as the eigenvalues of the spatial finite differences. For the equation without drift, the symbol reduces to (5.47), which is purely real, but generally we get an imaginary (phase) component. The domain of stability in Fourier space has to encompass all possible values of k. The corresponding eigenvalues $-[\ldots]$ are then characterised by negative real part, where the size depends on Δt and Δx but can be large. This motivates to investigate

$$R(z) = \frac{1 + (1-\theta)z}{1 - \theta z},$$

where $z \in \mathbb{C}^- = \{z \in \mathbb{C} : \ \mathrm{Re} z < 0\}$.

For a finite grid, recall the θ-scheme in matrix form

$$(I - \theta A)u^{m+1} = (I + (1+\theta)A)u^m \quad \Rightarrow \quad u^{m+1} = R(A)u^m = Q^{-1}(A)P(A)u^m, \tag{5.60}$$

where P, Q, R are appropriately defined (polynomial or rational) functions of the matrix A. For diagonalisable $A = BDB^{-1}$, a matrix function can be defined as $R(A) = BR(D)B^{-1}$ where $R(D)$ is the diagonal matrix with entries $R(d_{ii})$, where d_{ii} are the diagonal entries of D. (To see why this makes sense, picture the terms in a Taylor series for $R(A)$.) Writing

$$B^{-1}u^m = R(D)B^{-1}u^{m-1} = R(D)^m B^{-1}u^0$$

shows that stability is determined by the symbol R acting on the eigenvalues of A. For the central difference scheme, again for the heat equation with drift, the eigenvalues of A as per above are easily seen to be $\lambda_k = -\Delta t/\Delta x^2(\sigma^2 + \cos(k\xi)\sqrt{\sigma^4 - \Delta x^2\mu^2})$ with $\mathrm{Re}\lambda_k < 0$. They are real if $\sigma^2 \geq |\mu|\Delta x$.

The following classical notion of stability arises naturally.

Definition 5.3.1. A scheme (5.59) is called *absolutely stable (A-stable)* if

$$|R(z)| < 1 \quad \forall z \in \mathbb{C}^- = \{z \in \mathbb{C} : \ \mathrm{Re} z < 0\}. \tag{5.61}$$

The θ-scheme is A-stable for $\theta \geq \frac{1}{2}$.

Strong A-stability – fractional step schemes

We saw earlier that Crank-Nicolson does not dampen high wavenumber oscillations sufficiently, and this creates problems for non-smooth data. A stronger stability concept, which captures the required uniform damping across the spectrum, is the following.

Definition 5.3.2. A scheme (5.59) is called *strongly A-stable*, if it is A-stable and

$$\lim_{\mathrm{Re}z\to-\infty} |R(z)| < 1. \tag{5.62}$$

The θ-method is strongly A-stable for $\theta > \frac{1}{2}$, but the Crank-Nicolson scheme as borderline case is not. Recall that the Crank-Nicolson scheme of stepsize Δt can be seen as a combination of an explicit step ($\theta = 0$) of size $\Delta t/2$, followed by a fully implicit step ($\theta = 1$) of size $\Delta t/2$. This symmetry provides second order consistency, but lies just on the verge of stability. A natural generalisation are schemes which combine two or more, say k steps of θ-schemes with potentially different θ_j and potentially different step-size Δt_j, into one macro-step. For the scheme to be consistent, it is necessary that

$$\sum_{j=1}^{k} \Delta t_j = \Delta t, \tag{5.63}$$

and one easily sees that this is also sufficient. This leaves $2k-1$ degrees of freedom to play with to achieve higher than first order accuracy and strong stability. An analysis similar to the one leading onto (5.53) shows that the additional condition

$$\sum_{j=1}^{d} (2\theta_j - 1)\Delta t_j = 0 \tag{5.64}$$

leads to second order accuracy for constant coefficient problems. This shows immediately that it is not possible to pick all $\theta_j > 1/2$. Any number of Crank-Nicolson sub-steps is of course second order accurate, but not strongly stable. For strong stability, one has to ensure (5.61) and (5.62). Strictly, this fractional step scheme is not consistent with our earlier definitions of a symbol for a one-step scheme, but it is obvious that the definition

$$R(z) = \prod_{j=1}^{k} R_{\theta_j}(\Delta t_j/\Delta t z)$$

provides the necessary generalisation. Assuming A-stability, which of course has to be checked (Exercise 8 in 5.4), strong A-stability is given if

$$\prod_{j=1}^{k} \frac{(1-\theta_j)}{\theta_j} < 1.$$

We search in the parameter range $0 \le \Delta t_j \le \Delta t$ and $0 \le \theta_j \le 1$. This still leaves room for various choices.

The following scheme with $k = 3$ substeps has been proposed by Glowinski [Glowinski, 1985], with parameters chosen according to Table 5.1. The inuitive explanation why the scheme is

Table 5.1: Parameters for a fractional step θ-scheme.

i	Δt_i	θ_i
1	$(1-\sqrt{2}/2)\Delta t$	$2-\sqrt{2}$
2	$(\sqrt{2}-1)\Delta t$	$\sqrt{2}-1$
3	$(1-\sqrt{2}/2)\Delta t$	$2-\sqrt{2}$

of second order accurate is to note the symmetry of the substeps

$$t_m \;\to\; t_m + \Delta t_1 \;\to\; t_{m+1} - \Delta t_1 \;\to\; t_{m+1}$$

(and their θ-parameters) around $t + \Delta t/2$. It is left to Exercise 8 in 5.4 to show in detail that the resulting scheme is strongly A-stable and of second order accurate.

As an added bonus, $\theta_1\Delta t_1 = \theta_2\Delta t_2 = \theta_3\Delta t_3$, and therefore the implicit discretisation matrices $I - \theta_j\Delta t_j A$ are identical in each sub-step and therefore only one matrix has to be computed. Practically, if an implementation of the θ-scheme is available, this modification can be added on at almost no extra implementation cost, and guarantees second-order convergence for non-smooth data.

L-stable schemes – fully implicit start-up

A different type of scheme is more directly taylored to the important situation where non-smoothness arises only from initial data. We then know that the diffusion equation provides for smoothness of the solution after infinitesimally small time. This is seen e.g. from the formula (5.46) which shows the decay of high wavenumbers which may be present in the initial data u_0. In the class of θ-schemes, only the fully implicit scheme $\theta = 1$ has the same asymptotic behaviour for small grid sizes, as we saw in (5.49). This motivates an even stronger notion of stability.

Definition 5.3.3. A method is called *L-stable*, if it is A-stable and

$$\lim_{\mathrm{Re}\, z \to -\infty} R(z) = 0.$$

The θ-method is L-stable only for $\theta = 1$, i. e. the fully implicit method.

The idea to combine the smoothing property of an L-stable scheme to deal with non-smooth initial data, with the accuracy of a higher order A-stable scheme to take over the smooth solution, was proposed by Rannacher [Rannacher, 1984]. The underlying principle is that a small fixed (!) number of lower order steps does not reduce the order of convergence, while it smoothens the solution enough to eliminate spurious oscillations.

The following variant, now often referred to as Rannacher start-up, has proved useful in practice. Using the Crank-Nicolson scheme as basis, the first say l timesteps are replaced by $2l$ fully implicit steps of half the stepsize. A detailed analysis for the advection-diffusion equation with Dirac initial data is found in [Carter and Giles, 2007] and [Giles and Carter, 2006].

Essentially similar to (5.51), but now accounting for the modified start-up phase, the Fourier transform of the finite difference solution can be written as

$$\widehat{u}^m(k) = R_{\frac{1}{2}}(\Delta x, \Delta t; k)^{m-l} R_1(\Delta x, \Delta t/2; k)^{2l} \widehat{u}^0(k), \tag{5.65}$$

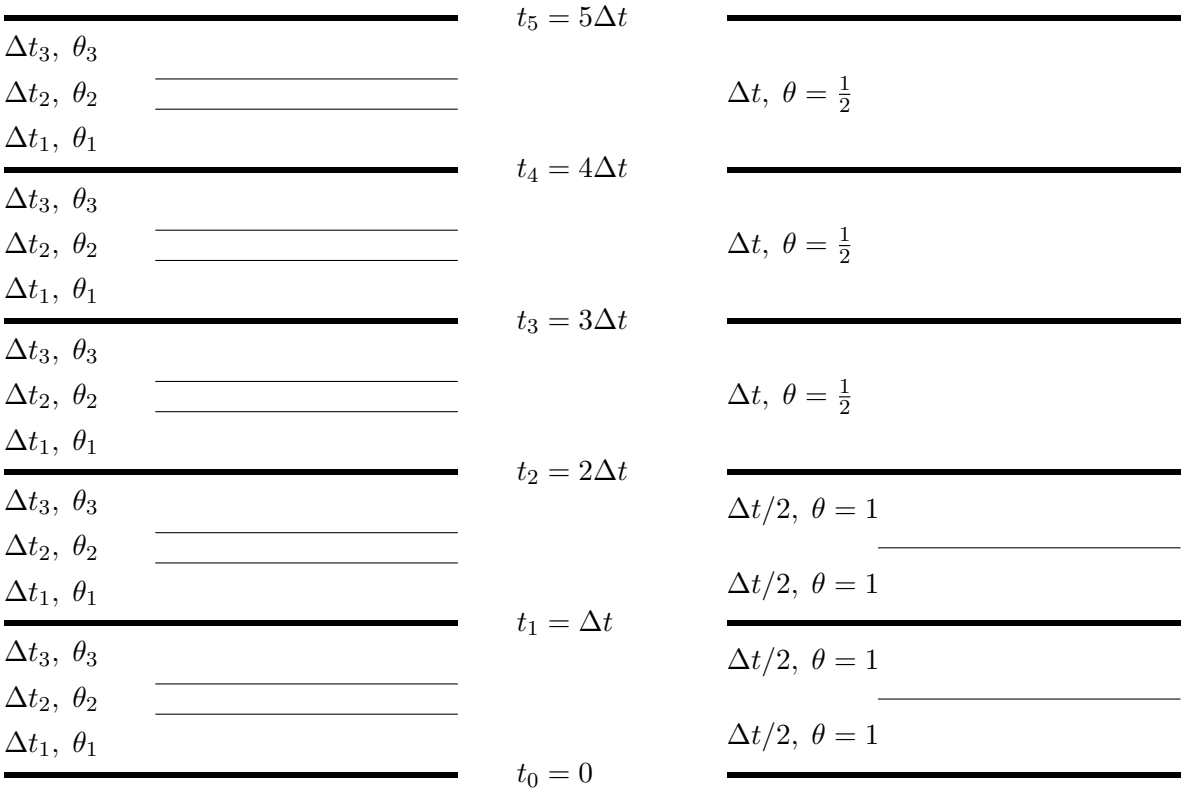

Figure 5.1: Schematic plot of the fractional-step θ-scheme (left) and Crank-Nicolson with "Rannacher start-up". In the first, *each* timestep is a sequence of three steps of θ-schemes with different θ_j and Δt_j, $j = 1, 2, 3$. In the latter, only the first four steps are implicit Euler steps ($\theta = 1$) of size $\Delta t/2$, all subsequent steps are Crank-Nicolson steps ($\theta = 1/2$) of size Δt.

where R_1 and $R_{1/2}$ are again the symbols of the fully implicit and Crank-Nicolson methods. Writing again

$$u_n^m = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{\Delta x} R_{\frac{1}{2}}(\Delta x, \Delta t; k)^{m-l} R_1(\Delta x, \Delta t/2; k)^{2l} \mathrm{e}^{ink}\, \mathrm{d}k,$$

a careful analysis as in [Giles and Carter, 2006] shows that

- for small k, $|k|/\Delta x < \Delta x^{-p}$, $0 < p < 1/3$, the difference between the analytical solution (5.52) and numerical solution (5.65) is $O(\Delta x^2) + O(\Delta t^2)$, by Taylor expansion;
- for medium k, $\Delta x^{-p} < |k|/\Delta x < \Delta x^{-q}$, $1/2 < q < 1$, analytical and numerical solution are negligible;
- for large k, $\Delta x^{-q} < |k|/\Delta x$, the analytical solution is still negligible, but at least $l = 2$ fully implicit steps are needed to dampen these high frequency components in the numerical solution to below order Δx^2. Recall this is the range where Crank-Nicolson runs into problems.

Indeed it turns out that this is the optimal choice and restores second order convergence for Dirac initial data. A practical advantage of this particular setup is that the system matrices for Crank-Nicolson, and implicit Euler of half the stepsize, are identical, so similarly to the fractional step scheme only one matrix has to be computed.

5.4 Exercises

1. Consider the advection diffusion with positive drift $\mu > 0$,

$$\frac{\partial u}{\partial t} = \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} - \mu\frac{\partial u}{\partial x}, \quad x \in \mathbb{R},\, t > 0. \tag{5.66}$$

   (a) Assume in the following $\mu \geq 0$ and, for a finite difference scheme with step size Δt and grid size Δx, $\mu\Delta t \leq \Delta x$. Give a geometric interpretation to these conditions.

   (b) Show that the implicit Euler scheme with upwinding difference for (5.66) is monotone with respect to the initial data, $u^m \geq 0$ for all $m \geq 0$ if $u^0 \geq 0$. Deduce that the scheme is stable in the maximum norm.

   (c) Define the implicit Euler discretisation in Lagrangian coordinates for (5.66), with a general interpolating function. Show that for piecewise linear interpolation this scheme is identical to the upwinding scheme, but with the modification that only the second order difference is implicit, whereas the (first order) upwinding difference is explicit (i.e. evaluated at the previous timestep). Discuss stability of the scheme.

2. Consider the θ-η-scheme (5.4) for the PDE (5.66). Using von Neumann analysis, assess the stability of the scheme. Verify that the symbol for the special case of the central difference scheme is given by (5.10). [*Hint: You may want to prove and use the identity* $(1-\eta)\exp(ik) + (2\eta - 1) - \eta\exp(-ik) = 2(2\eta - 1)\sin^2(k/2) + i\sin(k)$.]

3. Assume that a matrix $K \in \mathbb{R}^{n\times n}$ satisfies the following conditions:

i. $K_{ij} \leq 0, \quad 1 \leq i, j \leq n, \quad i \neq j;$
ii. $K_{ii} > \sum_{j \neq i} |K_{ij}|, \quad 1 \leq j \leq n.$

Show that:

(a) K is invertible;
(b) $K^{-1} \geq 0$ elementwise, which is equivalent to saying that $K^{-1}u \geq 0$ for all $u \in \mathbb{R}^n$ with $u \geq 0$, where again non-negativity is elementwise;
(c) if the stronger inequality

$$K_{ii} \geq 1 + \sum_{j \neq i} |K_{ij}|$$

holds for an index $1 \leq j \leq n$, then for the solution $u^1 \in \mathbb{R}^n$ of

$$Ku^1 = u^0,$$

for this index j

$$u_j^1 \leq \max\left(u_j^0, \max_{k \neq j} u_k^1\right).$$

(d) Discuss what the above properties mean for implicit finite difference schemes defined by the matrix K.

4. (a) Show that (5.40) and (5.41) are the eigenvalues and eigenvectors of (5.39).
   (b) Use Gershgorin's theorem (Lemma 5.2.13) to estimate the eigenvalues of the constant coefficient matrix (5.39) and compare with the exact values.

5. Consider an infinite version of the matrix K in (5.39), i.e. an operator $K : l_2 \to l_2$, $(Ku)_n = au_{n-1} + bu_n + cu_{n+1}$.
   (a) Find the adjoint operator K^* such that $(Ku, v) = (u, K^*v)$ where $(\cdot, \cdot)$ is the standard l_2 inner product. When is K self-adjoint?
   (b) Show that K is a *normal* operator, i.e. $KK^* = K^*K$.
   (c) Find the eigenvalues of K corresponding to "eigenvectors" e^{ikn}. Are these "eigenvectors" in l_2?
   (d) Show that the operator K for non-constant a_n, b_n, c_n is not normal.

Bonus: Explain why or why not the spectral theorem for operators in Hilbert spaces is applicable and how it relates to the above result.

6. Given is a tridiagonal matrix $A = \text{tridiag}(a_i, b_i, c_i, n)$, i.e. of the shape of the matrix K_1 from (5.27), with $0 < a_{i+1}/c_i$ for $i = 1, \ldots, n-1$.
   (a) Show that for a diagonal matrix $D = \text{diag}(d_i, n)$ such that

$$\frac{d_i^2}{d_{i+1}^2} = \frac{a_{i+1}}{c_i},$$

the *similarity transformation*

$$\widetilde{A} = DAD^{-1}$$

gives a symmetric, tridiagonal matrix $\widetilde{A} = \text{tridiag}(\tilde{a}_i, \tilde{b}_i, \tilde{c}_i, n)$ with $\tilde{b}_i = b_i, \tilde{a}_i = \tilde{c}_i = \sqrt{a_{i+1}c_i}$.

(b) Show that A and $\widetilde{A}$ have the same eigenvalues, $\mathcal{S}(A) = \mathcal{S}(\widetilde{A})$, and eigenvectors.

(c) Conclude that

$$\|A^m\|_2 \leq \frac{\max_i |d_i|}{\min_i |d_i|} \rho(\widetilde{A})^m. \tag{5.67}$$

7. Consider the IBVP

$$\frac{\partial u}{\partial t} + \mu(x,t)\frac{\partial u}{\partial x} = \frac{1}{2}\sigma^2(x,t)\frac{\partial^2 u}{\partial x^2}, \quad x \in (0,1), t > 0, \tag{5.68}$$

where σ and μ are differentiable functions of x and t with

$$0 < \underline{\mu} \leq \mu(x,t) \leq \overline{\mu}, \quad 0 < \underline{\sigma} \leq \sigma(x,t) \leq \overline{\sigma}, \quad x \in (0,1), t > 0. \tag{5.69}$$

(a) Define the fully implicit central difference scheme for (5.68).

(b) Use Exercise 6 to calculate a symmetric matrix $\widetilde{K}$ which is similar to the discretisation matrix K, by transformation with a diagonal matrix D.

(c) Use the Gerschgorin's theorem (Lemma 5.2.13) to find a lower bound for the eigenvalues of $\widetilde{K}$.

(d) Find upper and lower bounds for the diagonal elements of D.

(e) Using (5.67) and the previous two items, deduce that the implicit Euler scheme is stable in the l_2-norm.

(f) Explain briefly what you would expect to happen for the θ-scheme?

(g) Retrace the steps to find out if (5.69) is crucial here, with a view of extending the analysis to the case where the coefficients approach zero at the boundaries.

8. Consider the fractional-step θ-scheme from 5.3.2 for a finite difference solution of the heat equation on $(0,1)$.

(a) Write the finite difference solution for the heat equation in the form

$$\begin{aligned} u^m &= R(A)u^{m-1} \\ R(A) &= R_1(A) \cdot R_2(A)^2 \\ R_i(A) &= (I - \theta_i \Delta t_i A)^{-1}(I + (1-\theta_i)\Delta t_i A) \end{aligned}$$

and clearly define the matrix A.

(b) Give the spectrum of A and hence of $R(A)$.

(c) Show that $u(t) \in \mathbb{R}^n$ with

$$u(t) = \exp(At)u^0$$

is a solution to the semi-discrete problem

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= Au, \\ u(0) &= u^0. \end{aligned}$$

(d) By expanding $R(z)$ in terms of z for real scalar z (by abuse of notation, with I interpreted as 1) and comparison with the exponential e^z, show that the scheme is of second order accurate in Δt.

(e) Show that

$$|R(z)| \leq 1, \ \forall z \in \mathbb{R}^- \quad \text{and} \quad \lim_{z \to -\infty} R(z) < 1.$$

[Hint: Show separately that $|R_1(z)R_2(z)| \leq 1$ and $|R_2(z)| \leq 1$.]

9. (a) Based on your code for the θ-method, implement Crank-Nicolson with Rannacher start-up and the fractional-step θ-scheme for the heat equation with Dirac initial data, for $x \in [-5, 5]$, $t \in [0, 1]$, on a grid $(-N\Delta x, \ldots, -\Delta x, 0, \Delta x, \ldots, N\Delta x)$, where $N\Delta x = 5$, for M timesteps, $\Delta t = 1/M$, with zero boundary conditions at -5 and 5.

(b) For both schemes, table the difference between the discrete solution and the analytical one, at $x = 0$, $t = 1$, for $N = 10$, 20, 40, 80, 160, 320, $M =$10, 20, 40, 80, 160, 320.

(c) Repeat the study from 9b, but now in the interval $t \in [1, 2]$, with the exact solution to the heat equation at $t = 1$ as initial condition. Hence compare the error at $x = 0$, $t = 2$ between both schemes.

(d) From your results in 9b and 9c, plot the errors for $N = M = 10$, 20, 40,... over N in a log-log plot. Estimate the slope in this plot by linear regression and compare to the theoretically expected value.

Chapter 6

Finite difference pricing and hedging of European options

6.1 A model problem: Black-Scholes European put

6.1.1 Model specification and preliminaries

The classical model for discussing the finite difference pricing of European options is the Black-Scholes model. It is a good test case not because it is a realistic model in practice, but because it is *the* building block for more adequate models, it has many of the characteristic features of option pricing equations, with well-studied poperties and known closed-form solutions to compare the numerical solution against.

In the Black-Scholes model, the time t evolution of the risk-free value $V(S,t)$ of a European option on a stock with price S, follows the Black-Scholes PDE

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \quad S \in [0,\infty), t \in (0,T). \tag{6.1}$$

The option expires at some time $T > 0$. The model parameters σ and r are the volatility and prevailing interest rate. They are assumed constant and are non-negative. A typical set of parameters would be in the range $\sigma = 0.1, \ldots, 0.6$, $r = 0.01, \ldots, 0.08$, $T = 0.1, \ldots, 10$. The task is to find $V(S_0, 0)$, today's option value, where S_0 is the current value of the stock.

To do this, the PDE (6.1) is solved *backwards* in time, from a terminal condition at the expiry T of the contract, which is given by the payoff, say G, of the European option,

$$V(S,T) = G(S), \quad S \in [0,\infty). \tag{6.2}$$

The Black-Scholes equation (6.1) is the *Feynman-Kac* PDE to the stochastic process

$$\mathrm{d}S_t = rS_t\,\mathrm{d}t + \sigma S_t\,\mathrm{d}W_t, \tag{6.3}$$

which is of course geometric Brownian motion. Conversely, $V(S_t,t)$ is the discounted expectation of the payoff,

$$V(S_t,t) = \mathrm{e}^{-r(T-t)}\mathbb{E}(G(S_T)|S_t),$$

given the current stock price is S_t and where S follows (6.3).

For a European put option, for instance, the terminal condition (payoff) at expiry T is

$$V(S,T) = \max(K-S,0) \quad S \in [0,\infty), \tag{6.4}$$

where $K > 0$ is the exercise or strike price.

The Black-Scholes model is special in that it assumes, among other things, known constant volatility (the relative variance of returns). This makes it analytically solvable such that a new numerical scheme can be tested against the closed-from solutions before applying it to a more complex case with unknown solution. The Black-Scholes value of a European put is

$$P(S,t) = K\mathrm{e}^{-r(T-t)}N(-d_2) - SN(-d_1), \tag{6.5}$$

where

$$d_1 = \frac{\ln(S/K) + (r+\sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}, \tag{6.6}$$

$$d_2 = \frac{\ln(S/K) + (r-\sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}. \tag{6.7}$$

Aside its specific form and closed-form solution, the Black-Scholes model is representative for the general structure of pricing equations. When discussing numerical schemes for the Black-Scholes model, we will have in the back of our minds the slightly more general PDE

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)\frac{\partial^2 V}{\partial S^2} + \mu(S,t)\frac{\partial V}{\partial S} - r(S,t)V = 0, \tag{6.8}$$

which encompasses a wider range of models by suitable choice of the functions σ, μ, and r. By comparison, in the Black-Scholes model

$$\sigma(S,t) = \sigma S, \quad \mu(S,t) = rS, \quad r(S,t) = r,$$

with non-negative constants σ and r.

A key feature is that the coefficients $\sigma(S,t)$ and $\mu(S,t)$ degenerate for $S \to 0$, which is related to the process for S being bounded below by zero for positive starting value. This means $S = 0$ is a "natural" boundary for the PDE. At $S = 0$, the PDE reduces to the ODE

$$\frac{\partial V}{\partial t}(0,t) - rV(0,t) = 0, \tag{6.9}$$

$$V(0,T) = G(0). \tag{6.10}$$

This ODE can be solved independent of the PDE, with solution

$$V(0,t) = \mathrm{e}^{-r(T-t)}G(0).$$

No additional conditions for the PDE can (have to) be imposed at $S = 0$. We will call this a *natural boundary condition.*

We also need to demand an asymptotic condition for $S \to \infty$. Looking towards the computation, we restrict the positive axis to a finite interval $[0, S_{max}]$, and therefore need to set a boundary condition at $S = S_{max}$. If S_{max} is sufficiently large, the impact of this boundary value on the solution in the region of interest will be small. This becomes clear if we think of the boundary value as a payoff which is realised when S hits (the boundary point)

S_{max}. For large S_{max}, this event becomes unlikely and has little impact on the expected payoff, i.e. the solution $V(S,t)$, for relevant S.

For computational efficiency, however, S_{max} should not be unnecessarily large and one aims to restrict the computation to as small a region as possible. It then becomes a relevant question how to approximate the solution at S_{max} most accurately. If we could set the exact value of the solution to the unrestricted problem on $(0,\infty)$ as boundary condition at S_{max}, the solution to the IBVP would coincide with the former, but finding this solution is exactly the task of this computation. It is often possible to find an asymptotically accurate approximation to the solution, i.e. one that becomes more accurate for growing S, without solving the full PDE. The impact of this approximation on the overall solution is bounded by the error at the boundary, from the maximum principle for parabolic PDEs. In practice, the error in the region of interest is usually much smaller.

Take the example of the put. The put is unlikely to get *in-the-money* by expiry, if S is "large" today, such that the value $V(S,t) \to 0$ for $S \to \infty$. An asymptotic approximation is therefore $V(S_{max},t) = 0$. We focus on this example here and consider the implementation of different payoffs in Section 6.4.

A reasonable choice for "large" S_{max} could be guided by the current spot value plus a sufficiently large number of standard deviations. The variance scales with the expiry T. With common values of the volatility in the range of at most 50% p.a. (per year), a interest rate of around 5% and expiry in less than 20 years, picking $S_{max} = S_0 \exp(3\sqrt{T})$ should be good enough for practical applications. Then

$$\begin{aligned} \mathbb{P}(S_t > S_{max}) = \mathbb{P}(\log S_t > \log S_{max}) &= \mathbb{P}(\sigma W_t > \log(S_{max}/S_0) - (r - \tfrac{1}{2}\sigma^2)t) && (6.11)\\ &\leq \Phi(-5) && (6.12)\\ &\approx 2.87 \cdot 10^{-7}. && (6.13) \end{aligned}$$

A more refined calculation would use hitting probabilities via the running maximum. We will estimate the approximation error in detail in Section 6.4.

The resulting initial-boundary value problem is

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \qquad & S \in [0, S_{max}), t \in (0,T), && (6.14)\\ V(S_{max},t) = 0 \qquad & t \in (0,T], && (6.15)\\ V(S,T) = \max(K-S,0) \qquad & S \in [0, S_{max}]. && (6.16) \end{aligned}$$

It is interesting to note as an aside that the Black-Scholes PDE is invariant under rescaling of the S coordinate by a constant factor.

Remark 6.1.1 (Price vs. log-price)**.** *Due to the log-normality of the underlying distribution in the Black-Scholes model, the transformation $X = \log S$, $U(x,t) = V(S,t)$, leads to the constant coefficient PDE*

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 U}{\partial X^2} + (r - \sigma^2/2)\frac{\partial U}{\partial X} - rU = 0. \tag{6.17}$$

This is the Feynman-Kac PDE for a Brownian motion with variance σ^2 and drift $\mu = r - \sigma^2/2$. Further transformations, namely

1. *removing the drift via $x = X - \mu t$;*

2. *reverting the direction of time to measure* time-to-maturity*, and rescaling for unit variance per unit time,* $\tau = \sigma^2(T-t)$*;*

3. *inflating (i.e., undoing discounting of) the price,* $u = \exp(r(T-t))U$*,*

lead to the (forward) heat equation

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\frac{\partial^2 u}{\partial x^2}. \tag{6.18}$$

We briefly comment on the potential benefits of solving the PDE numerically in the transformed versus original coordinates.

Some discretisation schemes are taylored to constant coefficient PDEs or even more specifically to the heat equation. They may achieve better accuracy by exploiting this special form, see, e.g., the comments at the end of 4.2.2 or Exercise 3 in 4.4.

A computational disadvantage of going to log-price is that the transformation maps $S = 0$ *to* $X = -\infty$*, such that the numerical range needs to be truncated at some* X_{min}*, requiring an (additional) asymptotic boundary condition. Another side effect is that a uniform grid in* X *coordinates, say* $-N\Delta x, \ldots, -\Delta x, 0, \Delta x, \ldots, N\Delta x$*,* $x_n = -n\Delta X$*, maps to a very distorted grid in* S *coordinates via* $S_n = \exp(x_n)$*, with sparse points for large* S *and dense points for small* S*. This is likely to be advantageous for large* S*, and inefficient for small* S*, as in both regions the solution is usually nearly linear and can be resolved accurately on a coarse grid.*

The bottom line is that the majority of PDEs we want to solve numerially does not admit a transformation to the heat equation, and for those that do, a "semi-analytic" solution (i.e., closed-form modulo evaluation of an integral) is available and no numerical scheme needed. We therefore avoid making such simplifying transformations but solve the equation in "native" coordinates.

6.1.2 Discretisation

We focus in this section on the example of pricing a put option in the Black-Scholes model, as introduced in 6.1.1, i.e. we solve the IBVP (6.14) to (6.16). The key step is again to *discretise* both the solution and the equation, i.e. to approximate the solution by a finite set of unknowns, and to approximate the continuous equation by a finite-dimensional (linear) system of equations for these values.

Define a grid of $N+1$ nodes $S_0 = 0, S_1 = \Delta S, S_2 = 2\Delta S, \ldots, S_N = N\Delta S = S_{max}$, $\Delta S = S_{max}/N$, further M time steps $t_0 = 0, t_1 = \Delta t, t_2 = 2\Delta t, \ldots, t_M = M\Delta t = T$, $\Delta t = T/M$, at which we introduce a numerical approximation V_n^m to $V(n\Delta S, m\Delta t) = V(S_n, t_m)$. See Fig. 6.1.

The terminal condition (6.16) becomes

$$V_n^M = \max(K - S_n, 0) = \max(K - n\Delta S, 0). \tag{6.19}$$

The upper boundary condition is given by (6.15), so

$$V_N^m = 0.$$

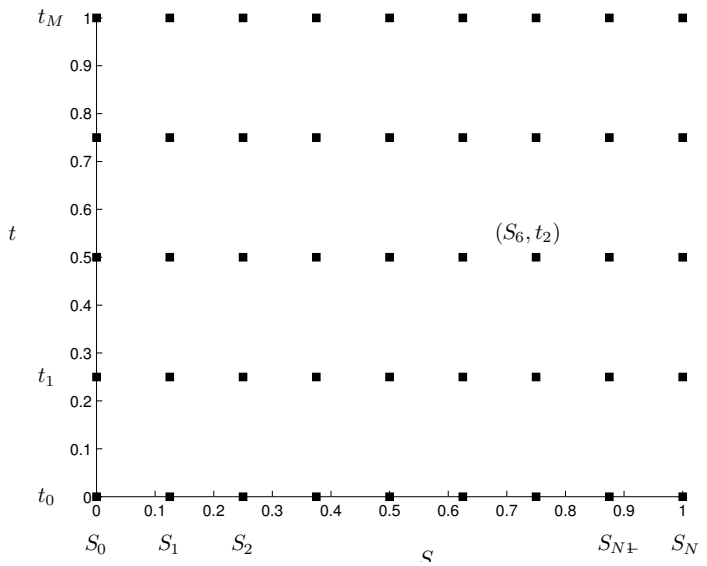

Figure 6.1: Grid, $N = 8$, $M = 4$.

Next, approximate derivatives in the PDE (6.14) by finite differences at grid points, assuming V is sufficiently smooth. We choose for the first derivative a *central difference*

$$\frac{\partial V}{\partial S}(S_n, t_m) = \frac{V(S_{n+1}, t_m) - V(S_{n-1}, t_m)}{2\Delta S} + O(\Delta S^2),$$

and for the second derivative the *second central difference*

$$\frac{\partial^2 V}{\partial S^2}(S_n, t_m) = \frac{V(S_{n+1}, t_m) - 2V(S_n, t_m) + V(S_{n-1}, t_m)}{\Delta S^2} + O(\Delta S^2).$$

For the time derivative, we will consider the *backward difference*

$$\frac{\partial V}{\partial t}(S_n, t_m) = \frac{V(S_n, t_m) - V(S_n, t_{m-1})}{\Delta t} + O(\Delta t)$$

and the *forward difference*

$$\frac{\partial V}{\partial t}(S_n, t_m) = \frac{V(S_n, t_{m+1}) - V(S_n, t_m)}{\Delta t} + O(\Delta t),$$

and combinations thereof. We study specific examples first.

Example 6.1.2 (Explicit Euler for Black-Scholes)**.** *The backward difference leads to the scheme*

$$\frac{V_n^m - V_n^{m-1}}{\Delta t} + \frac{1}{2}\sigma^2 \Delta S^2 n^2 \frac{V_{n+1}^m - 2V_n^m + V_{n-1}^m}{\Delta S^2} + r\Delta S n \frac{V_{n+1}^m - V_{n-1}^m}{2\Delta S} - rV_n^m = 0,$$

$m = 1, \ldots, M,\ n = 1, \ldots, N-1$. *Here V_n^{m-1} can be computed explicitly from the values V_{n-1}^m, V_n^m, V_{n+1}^m as*

$$V_n^{m-1} = A_n^m V_{n-1}^m + B_n^m V_n^m + C_n^m V_{n+1}^m, \tag{6.20}$$

where

$$A_n^m = \frac{1}{2}n^2\sigma^2\Delta t - \frac{1}{2}nr\Delta t, \tag{6.21}$$

$$B_n^m = 1 - n^2\sigma^2\Delta t - r\Delta t, \tag{6.22}$$

$$C_n^m = \frac{1}{2}n^2\sigma^2\Delta t + \frac{1}{2}nr\Delta t. \tag{6.23}$$

This is the explicit Euler *scheme. The backward difference is explicit for the backward equation, as the forward difference is explicit for the forward equation.*

At $n = 0$, the scheme naturally reduces to

$$V_0^{m-1} = (1 - r\Delta t)V_0^m, \tag{6.24}$$

and at the upper boundary we know

$$V_N^{m-1} = 0.$$

This procedure can be applied inductively in m, starting from V_n^M given by (6.19), to compute all values down to $m = 0$, i.e. $t = 0$. In the Black-Scholes case, with parameters constant over time, A_n^m, B_n^m and C_n^m do not actually depend on m, and therefore do not need to be recomputed in each timestep.

Example 6.1.3 (Implicit Euler for Black-Scholes)**.** *The forward difference leads to the scheme*

$$\frac{V_n^{m+1}-V_n^m}{\Delta t}+\frac{1}{2}\sigma^2\Delta S^2 n^2\frac{V_{n+1}^m-2V_n^m+V_{n-1}^m}{\Delta S^2}+r\Delta S n\frac{V_{n+1}^m-V_{n-1}^m}{2\Delta S}-rV_n^m=0,$$

$m=0,\ldots,M-1,\ n=1,\ldots,N-1$. *Here V_n^m is implicitly given by the system of equations*

$$a_n^m V_{n-1}^m+b_n^m V_n^m+c_n^m V_{n+1}^m=V_n^{m+1}, \tag{6.25}$$

where

$$\begin{aligned} a_n^m &= -\frac{1}{2}n^2\sigma^2\Delta t+\frac{1}{2}nr\Delta t,\\ b_n^m &= 1+n^2\sigma^2\Delta t+r\Delta t,\\ c_n^m &= -\frac{1}{2}n^2\sigma^2\Delta t-\frac{1}{2}nr\Delta t. \end{aligned}$$

This is the implicit Euler *scheme. The forward difference is implicit for the backward equation, as the backward difference is implicit for the forward equation. At $n=0$, the scheme naturally reduces to*

$$(1+r\Delta t)V_0^m=V_0^{m+1},$$

and at the upper boundary, we know

$$V_N^m=0.$$

In each time step, $m=M-1,\ldots,0$, a linear system for the unknowns V_0^m, V_1^m, …, V_{N-1}^m has to be solved.

Following these examples, we deduce the equivalent of the θ-timestepping scheme from earlier for the backward PDE

$$\frac{\partial V}{\partial t}+\frac{1}{2}\sigma^2(S,t)\frac{\partial^2 V}{\partial S^2}+\mu(S,t)\frac{\partial V}{\partial S}-r(S,t)V=0.$$

Recall the θ-timestepping scheme as interpreted as an explicit step of size $(1-\theta)\Delta t$, followed by a fully implicit step if size $\theta\Delta t$, with $\theta\in[0,1]$. One thus gets

$$\delta_t^- V_n^m+\left(\frac{1}{2}\sigma^2(S_n,t_{m-\theta})\delta_S^2+\mu(S_n,t_{m-\theta})\delta_S-r(S_n,t_{m-\theta})\right)(\theta V_n^{m-1}+(1-\theta)V_n^m)=0, \tag{6.26}$$

where δ_S and δ_S^2 are first and second central difference operators, e.g. $\delta_S V_n^m=(V_{n+1}^m-V_{n-1}^m)/2\Delta S$ etc, and the backward time difference $\delta_t^- V_n^m=(V_n^m-V_n^{m-1})/\Delta t$. This defines a scheme backwards in time, which can be written as

$$a_n^m V_{n-1}^{m-1}+b_n^m V_n^{m-1}+c_n^m V_{n+1}^{m-1}=A_n^m V_{n-1}^m+B_n^m V_n^m+C_n^m V_{n+1}^m, \tag{6.27}$$

where

$$a_n^m = -\frac{1}{2}\theta\Delta t\left(\sigma^2(S_n,t_{m-\theta})/\Delta S^2-\mu(S_n,t_{m-\theta})/\Delta S\right) \tag{6.28}$$

$$b_n^m = 1+\theta\Delta t\left(\sigma^2(S_n,t_{m-\theta})/\Delta S^2+r(S_n,t_{m-\theta})\right) \tag{6.29}$$

$$c_n^m = -\frac{1}{2}\theta\Delta t\left(\sigma^2(S_n,t_{m-\theta})/\Delta S^2+\mu(S_n,t_{m-\theta})/\Delta S\right) \tag{6.30}$$

and

$$A_n^m = \frac{1}{2}(1-\theta)\Delta t\left(\sigma^2(S_n,t_{m-\theta})/\Delta S^2 - \mu(S_n,t_{m-\theta})/\Delta S\right) \tag{6.31}$$

$$B_n^m = 1-(1-\theta)\Delta t\left(\sigma^2(S_n,t_{m-\theta})/\Delta S^2 + r(S_n,t_{m-\theta})\right) \tag{6.32}$$

$$C_n^m = \frac{1}{2}(1-\theta)\Delta t\left(\sigma^2(S_n,t_{m-\theta})/\Delta S^2 + \mu(S_n,t_{m-\theta})/\Delta S\right) \tag{6.33}$$

for $n = 1,\ldots,N-1$. Setting $\theta = 0$, one recovers the explicit method, with $\theta = 1$ the fully implicit method. The choice $\theta = 0.5$ is the second order accurate Crank-Nicolson method.

Boundary conditions depend on the problem at hand. For $n = 0$, in the Black-Scholes case, $a_n^m = c_n^m = A_n^m = C_n^m = 0$. The discretisation of the PDE naturally reduces to a discretisation of the ODE (6.9) and no numerical boundary conditions are necessary. For $n = N$, for the put, set $V_N^m = V_N^{m-1} = 0$ and eliminate them from the other equations, then the linear system takes the form

$$\underbrace{\begin{pmatrix} b_0^m & c_0^m & & \ldots & 0\\ a_1^m & b_1^m & c_1^m & \ldots & 0\\ 0 & \ddots & \ddots & \ddots & 0\\ 0 & \ldots & a_{N-2}^m & b_{N-2}^m & c_{N-2}^m\\ 0 & \ldots & 0 & a_{N-1}^m & b_{N-1}^m \end{pmatrix}}_{:=K_1(\theta)} \begin{pmatrix} V_0^{m-1}\\ V_1^{m-1}\\ \vdots\\ V_{N-2}^{m-1}\\ V_{N-1}^{m-1}\end{pmatrix} = \underbrace{\begin{pmatrix} B_0^m & C_0^m & & \ldots & 0\\ A_1^m & B_1^m & C_1^m & \ldots & 0\\ 0 & \ddots & \ddots & \ddots & 0\\ 0 & \ldots & A_{N-2}^m & B_{N-2}^m & C_{N-2}^m\\ 0 & \ldots & 0 & A_{N-1}^m & B_{N-1}^m \end{pmatrix}}_{:=K_0(\theta)} \begin{pmatrix} V_0^m\\ V_1^m\\ \vdots\\ V_{N-2}^m\\ V_{N-1}^m\end{pmatrix}. \tag{6.34}$$

This leads to Algorithm 3. The form of (6.34) is special because of degeneracy of the PDE

Algorithm 3 θ-method in European option pricing

1: Choose S_{max}, N, M, θ
2: $\Delta S = S_{max}/N$, $\Delta t = T/M$
3: $S \leftarrow (0,\Delta S,\ldots,(N-1)\Delta S) \in \mathbb{R}^N$
4: $V \leftarrow \text{payoff}(S) \in \mathbb{R}^N$
5: Set up matrices K_0 and $K_1 \in \mathbb{R}^{N\times N}$ as per (6.34) and (6.28)–(6.33)
6: **for** $m \leftarrow M, 1$ **do**
7: $\quad rhs \leftarrow K_0 V$
8: $\quad$ Solve $K_1 V = rhs$ for V
9: **end for**

coefficients at the lower boundary, and zero boundary conditions at the upper boundary.

Anticipating an example from section 6.4 on other payoffs, for a call option, one can envoce put-call parity (6.79) to deduce

$$V(S_{max},t) \approx S_{max} - K\mathrm{e}^{-r(T-t)},$$

where the approximation error is exactly identical to the put. The numerical boundary condition at S_N is then $V_N^m = S_N - K\mathrm{e}^{-r(T-t_m)}$. Proceding similar to the put leads to a system

$$K_1(\theta)V^{m-1} = K_0(\theta)V^m + F^m,$$

where the "boundary term" F_n^m is zero for $n = 0,1,\ldots,N-2$ and

$$F_{N-1}^m = -c_{N-1}V_N^{m-1} + C_{N-1}^m V_N^m.$$

The reason why there is no contribution from the $S = 0$ boundary is that $\mu(0,t) = \sigma(0,t) = 0$. A more detailed discussion follows in 6.4.

6.1.3 Numerical tests

We investigate the explicit scheme first, on the example of a Black-Scholes put. It is clearly a special case of Algorithm 3, but the main motivation for using the explict scheme is usually its ease of implementation, and for that reason we formulate the scheme very explicitly (pardon the pun) in Algorithm 4. Note that as the coefficients A_n^m, B_n^m, C_n^m do not depend on m

Algorithm 4 Explicit Euler scheme

$\sigma \leftarrow 0.4$, $r \leftarrow 0.05$
$T \leftarrow 1$, $K \leftarrow 0.25$
$S_{max} \leftarrow 1$
$N \leftarrow 16$, $M \leftarrow 16$
$\Delta t \leftarrow T/M$, $\Delta S \leftarrow S_{max}/N$
for $n \leftarrow 0$, N **do**
    $A_n \leftarrow \frac{1}{2}n^2\sigma^2\Delta t - \frac{1}{2}nr\Delta t$
    $B_n \leftarrow 1 - n^2\sigma^2\Delta t - r\Delta t$
    $C_n \leftarrow \frac{1}{2}n^2\sigma^2\Delta t + \frac{1}{2}nr\Delta t$
end for
for $n \leftarrow 0$, N **do**
    $V_n^M \leftarrow \max(K - n\Delta S, 0)$
end for
for $m \leftarrow M$, 1 **do**
    $V_0^{m-1} \leftarrow B_0V_0^m + C_0V_1^m$
    for $n \leftarrow 1$, $N-2$ **do**
        $V_n^{m-1} \leftarrow A_nV_{n-1}^m + B_nV_n^m + C_nV_{n+1}^m$
    end for
    $V_{N-1}^{m-1} \leftarrow A_{N-1}V_{N-2}^m + B_{N-1}V_{N-1}^m$
end for

for this example, because the coefficients in the Black-Scholes equation are time-independent, they need only be computed once. The boundary condition $V_N^m = 0$ is applied by leaving out the term $C_{N-1}V_N^m$ in the equation for V_{N-1}^{m-1}.

In Fig. 6.2, the numerical solution V_n^0 is compared to the analytically known Black-Scholes price $V(S,0)$ for $N = 16$ and $M = 16$.

To reduce the error, try and increase the number of time steps, say $M = 32$, still with $N = 16$. As shown in Fig. 6.3, the error does not decrease, as a matter of fact it becomes slightly larger. We therefore suspect that the error is mainly due to the space discretisation and increase the number of space steps as well, e. g. $N = 32$, $M = 32$. However, this goes terribly wrong and the numerical solution blows up.

To investigate this systematically, we study the error over grid refinements in Table 6.1.

Table 6.1 shows that reasonable solutions are obtained only provided that the number of time steps is sufficiently large, in particular increasing with the number of grid points. The

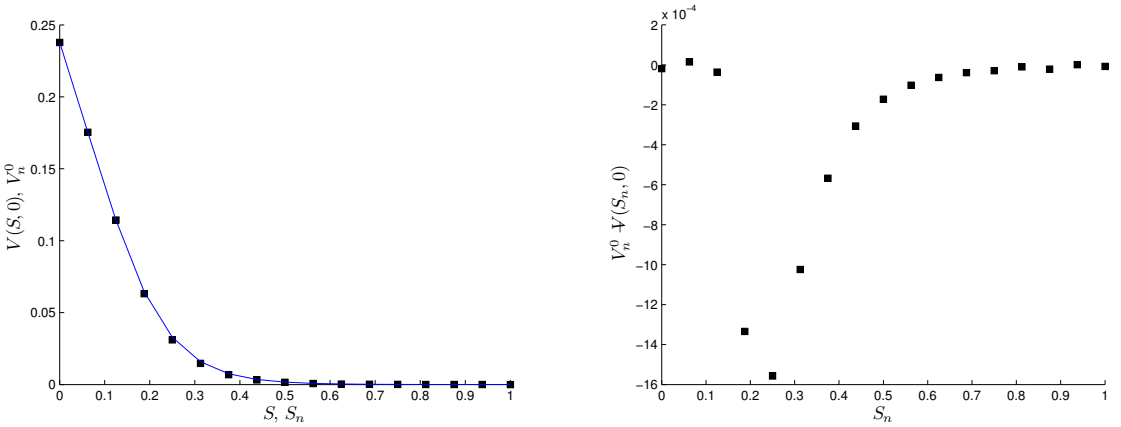

Figure 6.2: Left: Black-Scholes value of a put and finite difference approximation with 16 time steps and 16 grid intervals. Right: Error at the grid points. Parameters used are $\sigma = 0.4$, $r = 0.05$, $T = 1$, $K = 0.25$, $S_{max} = 1$.

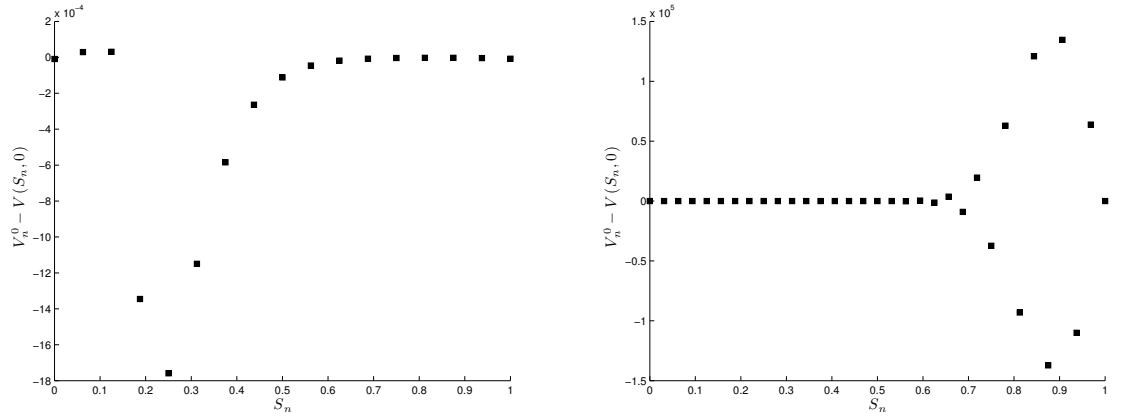

Figure 6.3: Finite difference error for $N = 16$, $M = 32$ (left) and $M = 32$, $N = 32$ (right).

behaviour therefore depends on the path of space/time refinement, which us referred to as *conditional stability*. Specifically, it appears two time refinements are required per space refinement.

If we look at the italic values in the "diagonal", obtained by two time refinements per refinement in the space coordinate, the error goes down by a factor of roughly four. The tabular indicates that we get second order convergence in space, first order convergence in time,

$$\epsilon = O(\Delta t) + O(\Delta S^2). \tag{6.35}$$

Along this "diagonal", $N^2/M = \Delta t/\Delta S^2 = 4$. It is clear that stability can only depend on the dimensionless variable $\sigma^2 \Delta t$. The explicit scheme appears stable as long as

$$\sigma^2 \frac{\Delta t}{\Delta S^2} < 0.64. \tag{6.36}$$

We now contrast this with the Crank-Nicolson scheme in Table 6.2.

$M\backslash N$	16	32	64	128	256	512
16	-1.5569e-03	-7.2855e-04	-1.8888e+07	-1.5937e+17	-5.0711e+26	-1.2048e+36
32	-1.7585e-03	-2.8995e-04	-3.3148e+09	-2.4715e+28	-3.2423e+47	-3.4126e+66
64	*-1.8596e-03*	-3.7393e-04	-7.2127e+13	-6.6809e+44	-4.0451e+80	-1.5998e+118
128	-1.9102e-03	-4.1590e-04	-7.3410e+30	-5.9686e+75	-1.0071e+135	-5.0670e+204
256	-1.9355e-03	*-4.3688e-04*	-6.9726e+54	-2.2570e+151	-3.7686e+238	-Inf
512	-1.9482e-03	-4.4736e-04	-1.0281e-04	NaN	NaN	NaN
1024	-1.9545e-03	-4.5261e-04	*-1.0789e-04*	NaN	NaN	NaN
2048	-1.9576e-03	-4.5523e-04	-1.1044e-04	6.8825e+140	NaN	NaN
4096	-1.9592e-03	-4.5654e-04	-1.1171e-04	*-2.6895e-05*	NaN	NaN
8192	-1.9600e-03	-4.5719e-04	-1.1235e-04	-2.7526e-05	NaN	NaN
16384	-1.9604e-03	-4.5752e-04	-1.1266e-04	-2.7842e-05	*-6.7188e-06*	NaN
32768	-1.9606e-03	-4.5768e-04	-1.1282e-04	-2.8000e-05	-6.8764e-06	NaN
65536	-1.9607e-03	-4.5777e-04	-1.1290e-04	-2.8079e-05	-6.9552e-06	*-1.6794e-06*

Table 6.1: Finite difference error for the explicit scheme, at-the-money, i. e. at $S = K$, for M time steps and N grid intervals. The values given in the table are $V_k^0 - V(K, 0)$, where k is the index for which $S_k = K$.

$M\backslash N$	16	32	64	128	256	512
16	-1.9534e-03	-4.5252e-04	*-1.0792e-04*	-5.0050e-05	-2.8698e-04	-5.0914e-04
32	-1.9590e-03	-4.5651e-04	-1.1171e-04	*-2.6906e-05*	-1.9418e-05	-1.4315e-04
64	-1.9603e-03	-4.5751e-04	-1.1266e-04	-2.7844e-05	*-6.7223e-06*	-8.2854e-06
128	-1.9607e-03	-4.5776e-04	-1.1290e-04	-2.8079e-05	-6.9559e-06	*-1.6804e-06*
256	-1.9608e-03	-4.5783e-04	-1.1296e-04	-2.8138e-05	-7.0144e-06	-1.7387e-06
512	-1.9608e-03	-4.5784e-04	-1.1298e-04	-2.8153e-05	-7.0291e-06	-1.7533e-06

Table 6.2: Finite difference error for the Crank-Nicolson scheme, at-the-money, i. e. at $S = K$, for M time steps and N grid intervals. The values given in the table are $V_k^0 - V(K, 0)$, where k is the index for which $S_k = K$.

If we look at the tabular line-wise, the error initially decays roughly by a facor of four from one column to the next, before it levels off and in fact increases slightly, but there is no sense of explosion. We see a similar effect looking down the columns. The Crank-Nicolson scheme appears of second order accurate in both ΔS and Δt,

$$\epsilon = O(\Delta t^2) + O(\Delta S^2), \tag{6.37}$$

and unconditionally stable. The smallest errors over the columns are highlighted in italic. This indicates the optimal refinement strategy as one where $N = 4M$.

6.1.4 Complexity considerations

The ultimate measure in judging the efficiency of a numerical scheme for a particular problem are the computational resources needed to solve the problem to a certain desired accuracy. In the wider context, issues like the implementation time, re-usability, availability of components etc will be relevant, but we here focus on computational time, specifically for the solution of the Black-Scholes PDE. Memory usage would be another factor but is typically less critical in this context.

Denote by the ϵ-*complexity* $C(\epsilon)$ the computational cost required to solve the problem to accuracy ϵ. We measure cost here as computation time. Clearly, the precise time will depend on factors like the implementation, computing architecture, and via the error on the parameters of the problem. As a useful crude measure it is sufficient to look at the *order* of $C(\epsilon)$ for small ϵ. This measure does not necessarily tell us which of two algorithms is superior for a given accuracy level, but assuming we want an accurate solution typically does give a good indication of the relative cost.

There are two ingredients to the ϵ-complexity. Firstly, the numerical parameters, here number of grid points N and time steps M, needed to solve the problem to accuracy ϵ. This has to come out of the numerical analysis of the method and assumes that we are able to determine N and M in such a way that the error is below ϵ. And secondly, the complexity of solving the problem for numerical parameters N and M.

The second question is usually easier to answer. By inspection of Algorithm 4, for the explicit Euler finite difference scheme, the number of simple operations is proportional to M via the time step loop and N via the loop over grid points within each timestep, $C \sim NM$.

From (6.35), we see that we want to choose $\Delta t \sim \Delta S^2$ for the explicit Euler scheme to balance error terms resulting from the time and asset discretisation. In fact, from (6.36), this is also necessary for stability. An optimal choice is therefore

$$M \sim N^2 \sim \epsilon^{-1},$$

so, inserting back in, for explicit Euler,

$$C_{EE}(\epsilon) = O(\epsilon^{-3/2}).$$

For Crank-Nicolson, inspecting Algorithm 3, the computational time is still proportional to NM if the linear system can be solved in $O(N)$ operations in each timesteps. The Thomas algorithm does this. The error is now given by (6.37) and we want to choose

$$M \sim N \sim \epsilon^{-1/2},$$

so, inserting again,

$$C_{CN}(\epsilon) = O(\epsilon^{-1}).$$

The Crank-Nicolson method is more efficient asymptotically because for roughly the same computational cost for identical grid parameters, higher accuracy is achieved. This means conversely that less timesteps are needed for the same accuracy which makes the method computationally cheaper.

6.2 Convergence analysis

In chapters 4 and 5 we analysed convergence of forward one-step schemes of the form

$$\begin{aligned} u^m &= Lu^{m-1}, \quad m > 1; && (6.38)\\ u^0 &\quad \text{given;} && (6.39)\\ \text{find } u^M &\quad \text{for some } M > 0, && (6.40) \end{aligned}$$

for forward PDEs. The fact that we are concerned with backward equations here, formally, merely changes the direction of travel,

$$\begin{aligned} V^{m-1} &= KV^m, \quad m > 1; && (6.41)\\ V^M &\quad \text{given for some } M > 0; && (6.42)\\ \text{find } V^0. & && (6.43) \end{aligned}$$

The backward PDE can be written as a forward PDE and vice versa by time reversal, $t \to T-t$, and a finite difference scheme for the forward equation thus turns into a finite difference scheme for the backward equation by counting timesteps backwards, $m \to M - m$.

The analysis in 4.1 and 4.2 reveals that convergence of finite difference schemes is a result of consistency and stability. To recap, the consistency analysis is based on the so-called truncation error of the scheme, which measures the closeness of the discretised and continuous equations. This translates immediately to the backward context. The same is true in principle for stability, the notion of boundedness of solutions, with the proviso that option pricing PDEs are routinely degenerate at boundaries which falls outside the previous framework. Maximum norm stability is analysed relatively easily because maximum principles only rely on a local comparison of the terms in the (discretised) PDE. This is more involved for l_2-stability, and we present a new technique, a so-called "energy method", and apply it to Black-Scholes-type PDEs.

6.2.1 Truncation error for Black-Scholes-type problems

Recall that the truncation error measures how well the solution to the PDE, evaluated at the grid points, satisfies the difference scheme. The notion of consistency expresses that in the limit for vanishing grid size and timestep the discrete equation approaches the continuous one. The consistency order measures the speed. We adapt the definition from the forward PDE for clarity.

Definition 6.2.1 (Truncation error for backward equations)**.** Let V be the solution to a PDE with time-coordinate t. The *truncation error* of a backward one-step difference scheme of the

form (6.41) is defined as

$$T(.,t) = \frac{1}{\Delta t}\left(u(.,t-\Delta t) - Ku(.,t)\right).$$

Based on this definition of the truncation order, the notions of consistency and consistency order are as per points 2. and 3. in Definition 4.2.3.

Example 6.2.2. *The explicit finite difference scheme with central differences is consistent of order $p = 2$, $q = 1$. This can be verified by Taylor expansion of*

$$\begin{aligned} V(S_n, t_{m-1}) &= V - \Delta t\frac{\partial V}{\partial t} + \tfrac{1}{2}\Delta t^2\frac{\partial^2 V}{\partial t^2} + o(\Delta t^2) \\ V(S_{n\pm1}, t_m) &= V \pm \Delta S\frac{\partial V}{\partial S} + \tfrac{1}{2}\Delta S^2\frac{\partial^2 V}{\partial S^2} \pm \frac{1}{6}\Delta S^3\frac{\partial^3 V}{\partial S^3} + \frac{1}{24}\Delta S^4\frac{\partial^4 V}{\partial S^4} + o(\Delta S^4), \end{aligned}$$

where arguments (S_n, t_m) of V and its derivatives are omitted. From the definition of the truncation error,

$$\begin{aligned} T(S_n,t_m) &= \frac{V(S_n,t_{m-1}) - V(S_n,t_m)}{\Delta t} - \frac{1}{2}\sigma^2(S_n,t_m)\frac{V(S_{n+1},t_m) - 2V(S_n,t_m) + V(S_{n-1},t_m)}{\Delta S^2} \\ &\quad -\mu(S_n,t_m)\frac{V(S_{n+1},t_m) - V(S_{n-1},t_m)}{2\Delta S} + r(S_n,t_m)V(S_n,t_m) \\ &= -\frac{\partial V}{\partial t} + \tfrac{1}{2}\Delta t\frac{\partial^2 V}{\partial t^2} + o(\Delta t) - \frac{1}{2}\sigma^2(S_n,t_m)\frac{\partial^2 V}{\partial S^2} - \frac{1}{24}\sigma^2(S_n,t_m)\Delta S^2\frac{\partial^4 V}{\partial S^4} + o(\Delta S^2) \\ &\quad -\mu(S_n,t_m)\frac{\partial V}{\partial S} - \frac{1}{6}\Delta S^2\mu(S_n,t_m)\frac{\partial^3 V}{\partial S^3} + o(\Delta S^2) + rV \\ &= -\frac{1}{24}\sigma^2(S_n,t_m)\Delta S^2\frac{\partial^4 V}{\partial S^4} - \frac{1}{6}\Delta S^2\mu(S_n,t_m)\frac{\partial^3 V}{\partial S^3} + \tfrac{1}{2}\Delta t\frac{\partial^2 V}{\partial t^2} + o(\Delta S^2) + o(\Delta t). \end{aligned} \tag{6.44}$$

Similarly, by Taylor expansion exactly as for the forward PDEs, one shows the following for the truncation error of the θ-scheme.

Proposition 6.2.3. *The θ-scheme with central differences is consistent of order 2 in ΔS and of order 1 in Δt, unless $\theta = 1/2$ (Crank-Nicolson), for which the order in Δt is 2.*

For a put in the Black-Scholes model, as the analytical solution is known explicitly, one can actually compute the truncation error from 6.44 for illustration. The individual terms and the overall truncation error are plotted for continuous argument S and $t = 0$ in Fig. 6.4 for the explicit Euler scheme.

The truncation error is largest in absolute terms around the strike. By closer inspection, the location of the maximum error is somewhat below the strike. To understand this, note that the PDE (and the numerical scheme, respectively) does not only propagate the solution, but equally the error. The drift present in the PDE has moved the peak towards the left. This means that for most of the time the truncation error around the strike is positive, and explains why the numerical solution in Table 6.1 is smaller than the exact one.

It is also observed in Table 6.1 that reducing the timestep for fixed gridsize increases the error, if marginally, which may seem counter-intuitive at first. Looking at the individual terms of the truncation error, one sees that the peak in the term proportional to Δt, and the total truncation error, have opposite signs, which explains why the truncation error can become larger when the timestep is reduced. For $\Delta t \to 0$, and ΔS fixed, the numerical solution converges to a semi-discrete one given by a system of ODEs in time.

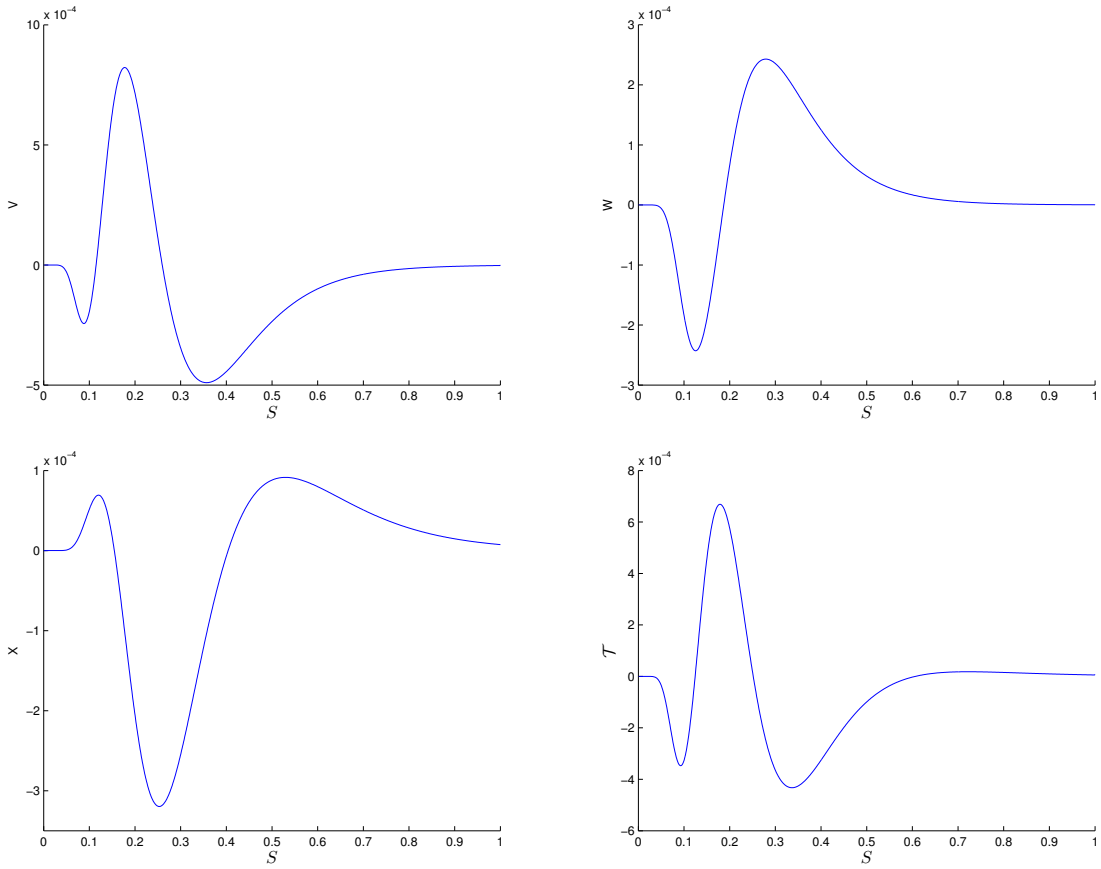

Figure 6.4: The truncation error for the explicit Euler central difference scheme for the Black-Scholes PDE, and its individual terms. Parameters are $\sigma = 0.4$, $r = 0.05$, $T = 1$, $t = 0$, $K = 0.25$.

6.2.2 Maximum norm stability for Black-Scholes-type problems

Recall the notion of stability from Definition 4.2.1, with the interpretation that perturbations of the solution stay within the order of magnitude of their initial value.

Example 6.2.4 (Explicit Euler with central differences)**.** *By analogy with the forward equation (see Example 5.2.2 and Remark 5.2.3), a scheme of the form (6.20) satisfies a discrete maximum principle if*

$$A_n, B_n, C_n \geq 0, \tag{6.45}$$
$$A_n + B_n + C_n \leq 1. \tag{6.46}$$

This follows by induction from

$$|V_n^{m-1}| \leq \max(|V_{n-1}^m|, |V_n^m|, |V_{n+1}^m|),$$

and therefore $|V^0| \leq |V^1| \leq \ldots |V^M|$*, where* $|\cdot|$ *the maximum norm.*

To check when (6.46) holds, first note that

$$A_n^m + B_n^m + C_n^m = 1 - r(S_n, t_m)\Delta t \leq 1$$

if $r \geq 0$*, which is generally the case. Now, for non-negativity of the coefficients, (6.45), it is necessary that*

$$B_n^m \geq 0 \quad \Leftrightarrow \quad \sigma^2(S_n, t_m)\Delta t/\Delta S^2 + r(S_n, t_m)\Delta t \leq 1 \tag{6.47}$$

for all $1 \leq n \leq N-1$*, and*

$$A_n^m, C_n^m \geq 0 \quad \Leftrightarrow \quad \Delta S|\mu(S_n, t_m)| \leq \sigma^2(S_n, t_m) \tag{6.48}$$

for $1 \leq n \leq N-1$*. The critical constraint is usually* $\sigma^2\Delta t/\Delta S^2 \leq 1$*, because* Δt *and* ΔS *and therefore the remaining terms in (6.47) and (6.48) are both small. But see the next example.*

Example 6.2.5 (Black-Scholes)**.** *Continuing the previous example, specifically, applying Example 6.2.4 to the Black-Scholes PDE, with coefficients as in (6.20), one needs to satisfy the stability conditions*

$$B_n^m \geq 0 \quad \Leftrightarrow \quad n^2\sigma^2\Delta t + r\Delta t \leq 1$$

for all $0 \leq n \leq N-1$*, and*

$$A_n^m \geq 0 \quad \Leftrightarrow \quad r \leq \sigma^2 n \tag{6.49}$$

for $n \geq 1$*. The critical cases are for large and small* n *respectively. The explicit Euler scheme is therefore stable, if*

$$M = T/\Delta T \geq T\left(\sigma^2(N-1)^2 + r\right) \tag{6.50}$$

and

$$r \leq \sigma^2. \tag{6.51}$$

Condition (6.50) is the expected timestep constraint for explicit schemes. For large N *it essentially requires that* $M \geq \sigma^2 T N^2$*, which confirms precisely the empirical results from Table 6.1. Condition (6.51) is new and arises from the way in which the coefficients degenerate towards the zero boundary.*

This can be addressed by upwinding as follows. For some k *large enough,* $r \leq \sigma^2 n$ *for* $n \geq k$*, so (6.49) is given for* $n \geq k$*. For* $n < k$*, upwinding can be used, resulting in slightly modified and always non-negative coefficients. As* k *is fixed and does not change when the total number* N *of grid points increases, the point* S_k *moves closer to* 0 *and this "local" upwinding has no adverse impact on the convergence order. This procedure is rarely necessary in practice though, even if (6.51) is violated.*

Example 6.2.6 (Implicit Euler with central differences)**.** *Observe for the coefficients in (6.25)*

$$a_n^m + b_n^m + c_n^m = 1 + r(S_n, t_m)\Delta t \geq 1, \tag{6.52}$$

if $r \geq 0$ *as usual. Assume now (6.48) holds again,*

$$\sigma^2(S_n, t_m) \geq \Delta S|\mu(S_n, t_m)| \tag{6.53}$$

everywhere, as needed for stability of central differences in the explicit scheme in Example 6.2.4. Then this implies

$$a_n^m, c_n^m \leq 0 \tag{6.54}$$

$$b_n^m \geq 1. \tag{6.55}$$

This guarantees convergence in the maximum norm, as seen for the forward equation in 5.2.2.

Recall, once again, that the θ-scheme is equivalent to a "fractional step" scheme with

1. an explicit Euler step of length $(1-\theta)\Delta t$,
2. followed by an implicit Euler step of length $\theta\Delta t$.

From this remark it follows that the θ-scheme will be stable if both substeps are stable, i.e. the coefficients A-C in (6.31)-(6.33) satisfy (6.45), and the coefficients a-c in (6.28)-(6.30) satisfy (6.52) and (6.54). This shows the following.

Proposition 6.2.7. *The θ-central difference scheme satisfies a discrete maximum principle, and is consequently stable in the maximum norm, if*

$$\sigma^2(S,t) \geq \Delta S \, |\mu(S,t)| \tag{6.56}$$

and

$$(1-\theta)\Delta t \left(\frac{\sigma^2(S,t)}{\Delta S^2} + r \right) \leq 1 \tag{6.57}$$

for all S and t.

The condition (6.57) can usually be satisfied by choosing ΔS small enough, provided $\sigma(S_n, t_m)$ is not degenerate. Indeed, the equation *is* degenerate for the Black-Scholes model, see Example 6.2.5, and the work-around explained there.

Similar to the heat equation, there appears to be a stability constraint on Δt if $\theta \neq 1$. Further analysis would reveal that such a condition is necessary for a maximum principle to apply, i.e. maximum norm stability with an amplification factor of $C = 1$ in Definition 4.2.1, but that the scheme is stable (however with $C > 1$) under weaker conditions, namely identical to those for l_2 stability. This is in line with numerical experiments, where all schemes with $\theta \geq 1/2$, including the Crank-Nicolson scheme (see Table 6.2), appear unconditionally stable. We refer for the analysis in the maximum norm to [Thomée, 1990] and sketch an analysis explaining this behaviour in the l_2 norm in 6.2.3.

6.2.3 Mean-square stability for Black-Scholes-type problems

Analysing l_2-stability for Black-Scholes problems becomes more involved than for the heat-equation. This is partly because of "non-normality", as discussed in 5.2.3 and especially Example 7 in 5.4, but more severely the degeneracy of the pricing equations at the boundary.

We follow [Achdou and Pironneau, 2005] here and consider in the following a model somewhat between (6.1) and (6.8) in generality, the so-called *local volatility* model

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{6.58}$$

We assume that σ stays within a certain range,

$$0 < \underline{\sigma} \leq \sigma(S,t) \leq \overline{\sigma} < \infty \tag{6.59}$$

for all S, t, so the model is in some sense close to Black-Scholes. This is especially crucial for large and small S, where (6.59) ensures that the diffusion coefficient σS is Lipschitz continuous.

We focus on the fully implicit Euler scheme of the form

$$K_1 V^{m-1} = (I + \Delta t A)V^{m-1} = V^m \tag{6.60}$$

with suitably defined matrices K_1 and A. The key to the analysis is an inequality of the form

$$V'AV \geq -C|V|^2 \quad \forall V \in \mathbb{R}^N, \tag{6.61}$$

where C is independent of ΔS and Δt. In [Achdou and Pironneau, 2005], the stronger inequality

$$V'AV \geq C_1|V|_w^2 - C_2|V|^2 \quad \forall V \in \mathbb{R}^N, \tag{6.62}$$

is derived, with a "weighted" norm

$$|V|_w^2 = \sum_{n=1}^{N} (S_n/\Delta S)^2 (V_n - V_{n-1})^2 + (S_N/\Delta S)^2 V_N^2.$$

We refer to [Achdou and Pironneau, 2005] for the proof of this, and can then deduce from (6.61) for the fully implicit scheme (6.60) that

$$\begin{aligned} |V^m||V^{m-1}| &\geq V^{m-1\prime}V^m = V^{m-1\prime}(I + \Delta t A)V^{m-1} \geq |V^{m-1}|^2 - \Delta t C|V^{m-1}|^2 \quad (6.63) \\ &= (1 - \Delta t C)|V^{m-1}|^2. \quad (6.64) \end{aligned}$$

Assume now that $\Delta t < 1/C$, which is not very restrictive because C is assumed independent of ΔS. Dividing both sides by $|V^{m-1}|$, by induction,

$$|V^m| \leq (1 - C\Delta t)^{-(M-m)}|V^M| \leq \mathrm{e}^{CT}|V^M|.$$

Properties of the type (6.62) are called Gårding inequalities. They are weaker than *coercivity*,

$$V'AV \geq C|V|^2 \quad \forall V \in \mathbb{R}^N,$$

with $C > 0$, which holds e.g. for the heat equation, but not generally for problems with drift or variable coefficients. Equation (6.62) expresses that A is "not too negative". Loosely speaking, in (6.61), the negative eigenvalues are bounded below independent of N.

Remark 6.2.8. *The bad news is that the technique in [Achdou and Pironneau, 2005] to prove (6.62) appears limited to problems of the type (6.58), with well-behaved volatility as in (6.59). This excludes other relevant examples like the CIR process, where the diffusion coefficient is not linear at 0, but e.g. a square-root. A convergence analysis of difference schemes for some of these more general models is found in [Sun et al., 2003].*

6.3 Stability of finite difference sensitivities

Sensitivities of the option price with respect to input parameters, so-called *Greeks*, are crucial trading (hedge) parameters. It is key that a computational method does not only give accurate prices, but can also produce accurate and stable sensitivities.

6.3.1 In- and out-of-model sensitivities

Important examples are the derivatives with respect to the underlying asset price, the *delta* and *gamma* of the option,

$$\Delta = \frac{\partial V}{\partial S}, \quad \Gamma = \frac{\partial^2 V}{\partial S^2},$$

and its time derivative, *theta*,

$$\Theta = \frac{\partial V}{\partial t}.$$

These are often linked by the pricing equation, e.g. in the Black-Scholes model

$$\Theta + \frac{1}{2}\sigma^2 S^2 \Gamma + rS\Delta - rV = 0,$$

so if we have the price and two sensitivities, the other can be computed directly from the pricing equation. When solvig the pricing PDE numerically, the value function is computed over a whole range of underlying prices and times, which lends itself to a sensitivity analysis in these variables.

A different type of sensitivities are those to model parameters, such as the *vega*

$$\mathcal{V}(S,t) = \frac{\partial V}{\partial \sigma}(S,t).$$

This is an "out of model" sensitivity, because the assumption underlying the Black-Scholes model is that the volatility σ is constant. Solving the pricing equation numerically only gives an approximation $\widehat{V}(\sigma)$ to the option value $V(\sigma)$ for a single input parameter σ. To obtain an approximation to the vega, one can compute

$$\widehat{\mathcal{V}} = \frac{\widehat{V}(\sigma + \Delta\sigma) - \widehat{V}(\sigma)}{\Delta\sigma}, \tag{6.65}$$

which involves pricing the option numerically for a second input parameter shifted by a small amount $\Delta\sigma$. This is a procedure referred to in practice as "bumping", and is really independent of the pricing method (e.g. finite differences) used to determine the option value for fixed input parameters. The total error incurred is a combination of the truncation error of the finite difference in the σ direction, and the error of the numerical method used to determine the prices, say ϵ,

$$\begin{aligned} \widehat{\mathcal{V}} &= \frac{V(\sigma + \Delta\sigma) - V(\sigma)}{\Delta\sigma} + \frac{(\widehat{V}(\sigma + \Delta\sigma) - V(\sigma + \Delta\sigma)) - (\widehat{V}(\sigma) - V(\sigma))}{\Delta\sigma} && (6.66) \\ &= \mathcal{V} + O(\Delta\sigma) + O(\epsilon/\Delta\sigma). && (6.67) \end{aligned}$$

The pricing error ϵ is divided by the small step $\Delta\sigma$ and hence magnified, so for *fixed* ϵ, choosing the optimal step size $\Delta\sigma$ is a trade-off between the two terms. This is slightly pessimistic because the numerical errors for parameters σ and $\sigma + \Delta\sigma$ will normally be related and will cancel out to some extent in the difference in the second term of (6.66). The first term can be reduced e.g. by the use of a central difference instead of the one-sided difference in σ direction, giving better accuracy $O(\Delta\sigma^2)$, which allows larger step sizes hence reducing the magnification of the pricing error in the second term.

We note in passing that as an alternative one could differentiate the PDE with respect to σ, and solve the resulting PDE for $\mathcal{V}$. This removes some of the aforementioned problems, but requires implementation of a separate PDE to be solved in conjunction with the original one. A largely equivalent approach is *algorithmic differentiation* of the pricing algorithm with respect to the input parameters [Griewank and Walther, 2008]. Tools for *automatic differentiation* are available, see e.g. [Research website on automatic differentiation,]. See

[Giles and Glasserman, 2006] for the application of these techniques to Monte Carlo sensitivities. Although the set-up of the pricing method is entirely different between Monte Carlo and finite differences, several issues concerning the computation of these "out of model" sensitivities stand outside the particular pricing method.

This is different for sensitivities to the underlying of the option, and time. We focus in the following on this first type of sensitivities, the derivatives of the value function with respect to its arguments S and t.

6.3.2 Finite difference sensitivities

In the finite difference method, one solves the pricing equation by approximating the derivatives by finite differences. An obvious route to Δ, Γ, and Θ is therefore to use these finite difference approximations, say V_n^m at point $S_n = n\Delta S$ and $t_m = m\Delta t$, applied to the computed finite difference solution,

$$\begin{aligned}
\Delta(S_n, t_m) &= \tfrac{\partial V}{\partial S}(S_n, t_m) \approx \Delta_n^m = \frac{V_{n+1}^m - V_{n-1}^m}{2\Delta S}, \\
\Gamma(S_n, t_m) &= \tfrac{\partial^2 V}{\partial S^2}(S_n, t_m) \approx \Gamma_n^m = \frac{V_{n+1}^m - 2V_n^m + V_{n-1}^m}{\Delta S^2}, \\
\Theta(S_n, t_m) &= \tfrac{\partial V}{\partial t}(S_n, t_m) \approx \Theta_n^m = \frac{V_n^{m+1} - V_n^{m-1}}{2\Delta t}.
\end{aligned}$$

We investigate the approximation "$\approx$" more quantitatively below. The boundary points $n = 0$ and $n = N$ would need a different treatment, e.g. by one-sided differences, but it is unlikely these are required in practice. Note also that for Θ we have used a central difference and that this is somewhat decoupled from the question whether a central time difference is a stable time marching scheme. If we are interested in the Θ at $t = 0$, we can perform one more timestep backward in time to $t = -\Delta t$ and still use a central difference.

The key empirical observation is that finite differences taken from the numerical solution do not necessarily have the same accuracy as the numerical solution V_n^m itself. Far from it, the error is potentially amplified through numerical differentiation. Specifically, if V_{n-1}^m, V_n^m and V_{n-1}^m are all accurate to say $O(\Delta S^2)$ at every point, we may find

$$\begin{aligned}
\Delta_n^m &= \frac{V_{n+1}^m - V_{n-1}^m}{2\Delta S} = \frac{V(S_n + \Delta S, t_m) + O(\Delta S^2) - V(S_n - \Delta S, t_m) + O(\Delta S^2)}{2\Delta S} \\
&= \frac{\partial V}{\partial S}(S_n, t_m) + O(\Delta S),
\end{aligned} \tag{6.68}$$

instead of $O(\Delta S^2)$ as expected from a central difference, and similarly

$$\Gamma_n^m = \frac{\partial^2 V}{\partial S^2}(S_n, t_m) + O(1).$$

That is to say the error is not reduced by refining the grid size, hence the finite difference gamma does not necessarily approximate the true gamma at all. This is already highlighted in [Shaw, 1998].

The way to get around this is to ensure that the error terms "$O(\Delta S^2)$" in (6.68) cancel out at neighbouring points S_{n-1}, S_n, S_{n+1} to some higher order in ΔS. This does happen if the (truncation) error is sufficiently smooth and is the case for smooth data. As we will see next, this is not the case in most relevant applications from derivative pricing.

Consider a European put. The discretised payoff/terminal condition is

$$V_n^M = (K - S_n)^+,$$

and we assume for simplicity that the grid has been chosen such that a grid point, say S_k, coincides with the strike, $S_k = K$. Then[1]

$$\Delta_n^M = \begin{cases} -1 & n < k \\ -\frac{1}{2} & n = k \\ 0 & n > k \end{cases} \tag{6.69}$$

and

$$\Gamma_n^M = \begin{cases} \frac{1}{\Delta S} & n = k \\ 0 & n \neq k \end{cases} . \tag{6.70}$$

This falls square into the numerical analysis of smoothing differences for non-smooth data in 5.3. In fact, differentiating the pricing PDE, e.g. in the Black-Scholes model

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{6.71}$$

gives equations for the sensitivities. For the Δ, after S differentiation,

$$\frac{\partial \Delta}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \Delta}{\partial S^2} + (r + \sigma^2)S\frac{\partial \Delta}{\partial S} = 0, \tag{6.72}$$

with terminal condition for the put

$$\Delta(S,T) = \begin{cases} -1 & \text{if } S < K \\ 0 & \text{else} \end{cases} .$$

By a further differentiation, for Γ,

$$\frac{\partial \Gamma}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \Gamma}{\partial S^2} + (r + 2\sigma^2)S\frac{\partial \Gamma}{\partial S} + (r + \sigma^2)\Gamma = 0, \tag{6.73}$$

where

$$\Gamma(S,T) = \delta(S - K),$$

the Dirac delta.

So, if we solve the orginal pricing equation and then apply finite differences to the finite difference solution to compute the Greeks, in essence what we are doing is numerically solve a PDE with more irregular final condition.

6.3.3 Stability analysis

We know from 5.3 that irregular data are characterised by slowly decaying Fourier transforms, and we need strong stability of timestepping schemes to get realistic solutions. A key indicator for stability were the eigenvalues of the iteration matrix $K = {K_1}^{-1}K_2$, denoted say by λ_j. For constant coefficient problems these are known in closed form, with their corresponding eigenvectors. This is no longer the case for the Black-Scholes PDE or other models, but we can still investigate the behaviour *numerically*. Fig. 6.5 shows the eigenvalues of K in descending order ($\lambda_{j+1} \leq \lambda_j$), and corresponding eigenvectors W_j.

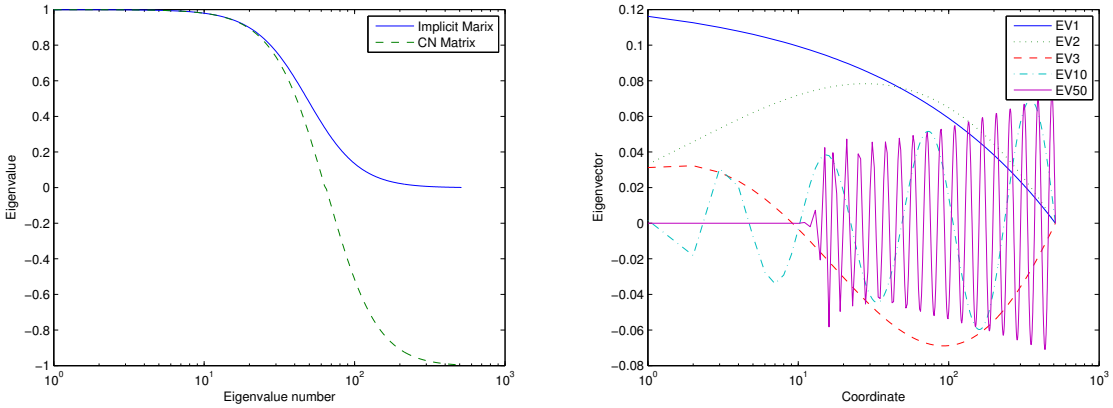

Figure 6.5: Left: Eigenvalues λ_k of the implicit and Crank-Nicolson matrix. Right: Eigenvectors to a few different eigenvalues λ_k, $k = 0, 1, 2, 9, 49$.

The spectrum is qualitatively very similar to the heat equation, see Fig. 6.5, with the eigenvalues smoothly decaying from 1 to 0 for the implicit scheme, and to -1 for the Crank-Nicolson scheme. The corresponding eigenvectors, in log coordinates, are still oscillatory for increasing eigenvalue number. Note the eigenvectors only depend on the space discretisation and are therefore identical for both timestepping schemes. The eigenvectors are no longer orthogonal, but given the remarkable similarity to the trigonometric eigenvectors of the heat equation, a transformation to the eigenvector basis is still very similar to taking the Fourier transform. So, if we introduce $W = (W^0, \dots, W^N)$, the matrix of eigenvectors of K, $\widehat{V}^m = W^{-1}V^m$ contains the coordinates of $V^m = (V_0^m, \dots, V_N^m)'$ in the eigenvector basis. The top plots of Fig. 6.6 show $\widehat{V}^M$ and $\widehat{V}^0$ for both the Crank-Nicolson and implicit Euler scheme.

Moving on to the Greeks, define similarly $\Delta^m = (\Delta_0^m, \dots, \Delta_N^m)'$, $\Gamma^m = (\Gamma_0^m, \dots, \Gamma_N^m)'$, and $\widehat{\Delta}^m = W^{-1}\Delta^m$, $\widehat{\Gamma}^m = W^{-1}\Gamma^m$. The remaining plots of Fig. 6.6 show $\widehat{\Delta}^M$ and $\widehat{\Delta}^0$, $\widehat{\Gamma}^M$ and $\widehat{\Gamma}^0$, the finite difference delta and gamma represented in the eigenvector basis. Direct computation shows that

$$\widehat{V}^0 = \Lambda^M \widehat{V}^M,$$

where $\Lambda = \mathrm{diag}(\lambda_j)$ the diagonal matrix of eigenvalues of K. Observe particularly at the bottom left plot of Fig. 6.6 the band of high frequency components which Crank-Nicolson fails to eliminate. This destroys the accuracy of the Gamma.

To obtain stable results, a smoothing timestepping scheme like the fractional-step θ-scheme or Crank-Nicolson with Rannacher start-up has to be used, as discussed in 5.3.2. The delta of a standard call or put is a step-function, its gamma a Dirac distribution. For a digital option, the delta is already a Dirac distribution and the gamma the derivative thereof.

[1]If the strike does not coincide with a point S_k, but lies between two S_k and S_{k+1}, one gets a different weighting of Δ_k^M, Δ_{k+1}^M etc. We return to the accurate grid approximation of non-smooth and singular functions more systematically in 6.4.

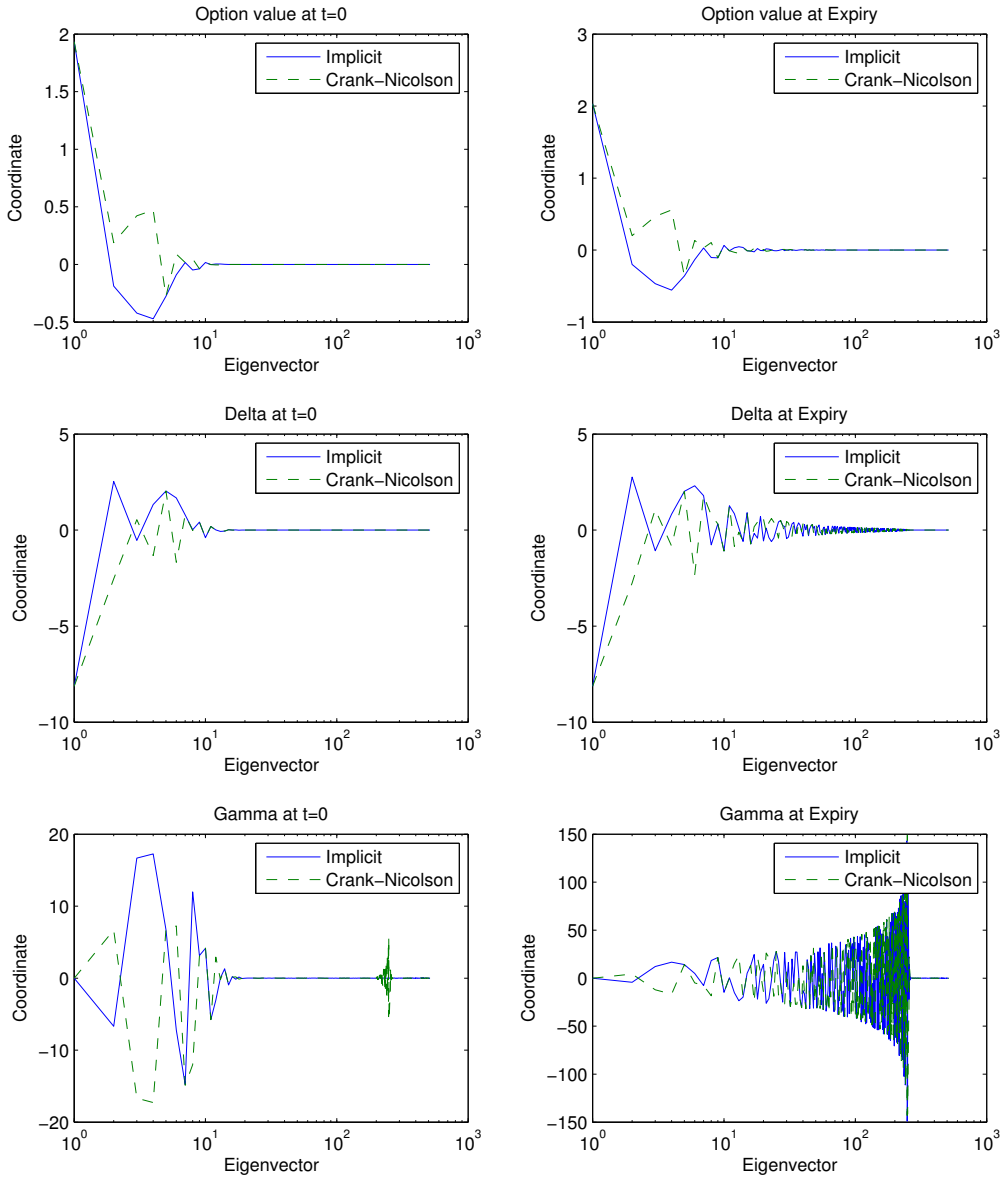

Figure 6.6: Finite difference value, delta, and gamma of a put option in the implicit and Crank-Nicolson scheme, in the eigenvector basis of the discretisation matrix.

We note without proof that strongly A-stable schemes like the fractional-step θ-scheme from 5.3.2 can still handle the situation of higher order derivatives of distributions. It is also shown in [Giles and Carter, 2006] that four fully implicit steps in the Rannacher start-up are still sufficient for a digital gamma. It is unlikely that higher derivative are needed in practice.

Remark 6.3.1. *The above PDEs (6.72) and (6.73) are Black-Scholes-type PDEs, but with slightly modified coefficients. We do not want to elaborate on this excessively as it usually causes no problems in practice, but this modification could conceivably have a bearing on the viability of numerical schemes if finite difference Greeks are required. Recall from 6.2.2 that for the Black-Scholes PDE, the central difference scheme imposed a constraint $\sigma^2 \geq r$ on the parameters for guaranteed maximum norm stability. This results from the type of degeneracy of the coefficients at the boundary, and expresses that the drift factor r has to be smaller than the diffusion factor σ^2. With r replaced by $r + \sigma^2$ in the Δ PDE, and $r + 2\sigma^2$ in the Γ case, this is* never *fulfilled. From the analysis in 6.2.2 it is clear that this problem only arises for nodes close to the $S = 0$ boundary, and if need be we could apply upwinding for these nodes only, which would not effect the second order convergence of the scheme overall. In practice this does not turn out to be necessary though.*

6.4 Implementing different payoffs

Different payoff functions can – conceptually – be incorporated very easily in the finite difference framework. What changes is the terminal condition (6.4) to the pricing PDE, and usually this means that asymptotic boundary conditions in the stock direction have to be adapted as well. So, mathematically speaking, this is a section about the accurate and stable approximation of boundary conditions.

Take our running example the Black-Scholes model. The three boundaries, $S = 0$, $S \to \infty$, and $t = T$, are very different in their nature.

1. At $S = 0$, the boundary values to the PDE solution come out naturally, as the PDE (6.1) reduces to the ODE (6.9), (6.10). All that changes is the terminal value $V(0, T) = G(0)$, the payoff for $S = 0$.

2. The boundary $S \to \infty$ has to be approximated at a large but finite value S_{max}. More often than not, we need to set approximate values for the solution, e.g. obtained from an asymptotic analysis of the PDE for large S. Ideally, one would like a "generic" boundary condition which works for all situations, or at least a large class. This has the practical advantage not to have to derive, implement, and possibly analyse different boundary conditions for each new problem. This leads to a discussion on derivative boundary conditions. We discuss this in 6.4.1.

3. At $t = T$, a terminal condition is defined by the payoff of the option. In the previous sections, the numerical solution at T is picked pointwise from the payoff, and this works fine for continuous payoff functions. In the presence of discontinuous, say digital option payoffs, or even Dirac data, say the gamma of a vanilla call at expiry, the accurate approximation on a grid is less trivial, and we analyse this in 6.4.2.

We are not going to say much about the implementation of boundary conditions as in item 1. If implemented diligently, the finite difference scheme for the PDE reduces to a

timestepping scheme for an ODE at the lowest node $S_0 = 0$. To exemplify this, setting $n = 0$ in (6.20) to (6.23) gives (6.24), and this serves as boundary condition for (6.20) for $n > 0$. It is worth pointing out, however, that the question of setting boundary conditions is related to the boundedness of the underlying stochastic process. Recall that the PDE (6.8) is the backward equation for a process of the form

$$\mathrm{d}S_t = \mu(S_t, t)\,\mathrm{d}t + \sigma(S_t, t)\,\mathrm{d}W_t. \tag{6.74}$$

Assume $\sigma(S, t) > 0$ for $S > 0$. For S_t to stay non-negative, it is intuitively necessary that

$$\sigma(0, t) = 0, \tag{6.75}$$

$$\mu(0, t) \geq 0, \tag{6.76}$$

where we assume that σ and μ are smooth. If the variance was non-zero, the process could surely diffuse beyond zero with positive probability, independent of the drift. Given that, the drift has to be non-negative or the process would drift into the negative range.

More precisely, and adapted to the question of boundary conditions, it is shown in [Sun et al., 2003] that the solution to the PDE is unique without boundary condition at $S = 0$, if

$$-\frac{1}{2}\frac{\partial \sigma^2}{\partial S}(0+, t) + \mu(0+, t) \geq 0,$$

where $0+$ indicates the limit coming from the positive side. This is clearly true for the Black-Scholes model. In other cases, this depends on the order by which the diffusion coefficient goes to zero. The first term is zero for say $\sigma = S^\alpha$ with $\alpha > 1/2$ and ∞ if $\alpha < 1/2$. In the limiting case of a square-root diffusion, as in the CIR model, the size of the drift becomes relevant. We refer to [Sun et al., 2003] for an analysis.

We focus in the following on the other boundary conditions 2. and 3. These depend largely on the payoff, and we have in mind as principal examples standard and digital calls and puts. Most other common payoffs are combinations of these. The payoff of a European call,

$$V(S, T) = \max(S - K, 0), \tag{6.77}$$

is continuous, with a discontinuity in the first derivative at the strike. It is asympotically linear for large S. Its value is in the Black-Scholes model given by the Black-Scholes formula

$$C(S, t) = SN(d_1) - K\mathrm{e}^{-r(T-t)}N(d_2), \tag{6.78}$$

which is related to the put value (6.5) by *put-call parity*,

$$C(S, t) - P(S, t) = S - K\mathrm{e}^{-r(T-t)}. \tag{6.79}$$

A binary *cash-or-nothing* call pays 1 if the stock is above the strike at expiry, and 0 otherwise. It has discontinuous payoff

$$V(S, T) = \begin{cases} 1 & \text{if } S > K, \\ 0 & \text{else.} \end{cases} \tag{6.80}$$

The Black-Scholes price of this option is

$$V(S, 0) = \mathrm{e}^{-rT}N(d_2), \tag{6.81}$$

with d_2 from (6.7) the same as in the Black-Scholes formula. Payoff and solution are asymptotically constant for large S. At expiry, the solution has a discontinuity at the strike, identical to the delta of the standard European call. The delta of the digital call at expiry is a Dirac distribution, and its gamma therefore the derivative of a Dirac distribution.

6.4.1 Asymptotic boundary condations

In this section, we revisit the question of truncating the compuational range at a large asset value, and the suitability of numerical boundary conditions. This aspect was first taken up in 5.2 with a focus on their impact on stability of numerical schemes. We extend this discussion here to a range of possible payoff functions, and discuss the accuracy of asymptotic values themselves.

Dirichlet conditions

Suitable approximations of the value of the option at the upper boundary of S depend on the specific contract. Put options are straightforward, because they are increasingly likely to expire worthless if the underlying spot price is high, and the value function goes to zero,

$$V(S,t) \to 0 \quad \text{for} \quad S \to \infty.$$

Numerically, we can approximate this by the boundary condition $V(S_{max},t) \approx 0$ for a large value S_{max}, take this point as largest grid point, S_N, and add $V_N^m = 0$ to finite difference equations for $n = 0, 1, \ldots, N-1$. From 5.2 we know that his approximation is stable, but how accurate is it? It is convenient to take a probabilistic interpretation of V as

$$V(S,t) = \mathbb{E}(\mathrm{e}^{-r(T-t)}(K - S_T)^+ | S_t = S) \leq K\,\mathbb{P}(S_T \leq K | S_t = S). \tag{6.82}$$

For the Black-Scholes model, S_T is log-normal, $\log(S_T/S_t) \sim N((\sigma^2/2 - r)(T-t), \sigma^2(T-t))$, and $\mathbb{P}(S_T \leq K | S_t = S)$ can be estimated by the cumulative standard normal, which shows that the put value $V(S,t)$ goes down very rapidly as $S \to 0$.

The analysis so far uses the specific shape of the payoffs, and the stock price distribution of the Black-Scholes model, but a similar argument can be applied to a wider situation by noting that the majority of option payoffs are asymptotically linear (affine, more precisely),

$$G(S) = aS - b \quad \text{for} \quad S > S_{min},$$

for some constants a and b. By conditioning on hitting the range $S < S_{min}$ at expiry, we can separate out the contribution of a potentially more complex payoff there, via

$$\begin{aligned} V(S,t) \;=\; & \mathbb{E}(B(t,T)(G(S_T) - (aS_T - b))|S_t = S, S_T \leq S_{min})\mathbb{P}(S_T \leq S_{min}|S_t = S) && (6.83)\\ & + \underbrace{\mathbb{E}(B(t,T)(aS_T - b)|S_t = S)}_{F(S,t)}, && (6.84) \end{aligned}$$

where $B(t,T)$ is the time t value of a bond expiring at T, say $B(t,T) = \exp(r(t-T))$ for constant interest rate r. The point is that F is independent of the model for the stock and explicitly computable, because the expectation is taken under the *risk-neutral* measure under which the discounted stock price is a martingale, $\mathbb{E}(B(t,T)S_T|S_t = S) = S$, and the remainder term is the bond price, so e.g. for constant interest rates

$$F(S,t) = aS - b\mathrm{e}^{-r(T-t)}.$$

The first term in (6.83) will be small for large S_{max} and gives an estimate

$$|V(S,t) - F(S,t)| \leq \max_{s \leq S_{min}} |G(s) - (as - b)|\;\mathbb{P}(S_T \leq S_{min}|S_t = S), \tag{6.85}$$

which justifies the numerical boundary condition

$$V_N^m = aS_N - b\mathrm{e}^{-r(T-t_m)}.$$

Put options, $a = b = 0$, vanilla calls, $a = 1$, $b = K$, and digital calls, $a = 0$, $b = 1$, are special cases.

Generic asymptotic conditions

Taking the argument around asymptotic linearity a step further, a slightly more generic boundary condition is based on the observation that often

$$\lim_{S\to\infty} \frac{\partial^2 V}{\partial S^2} = 0,$$

which can be approximated by

$$\frac{\partial^2 V}{\partial S^2}(S_{max}, t) = 0.$$

Approximated by a finite difference at $S_N = S_{max}$,

$$\frac{V_{N+1}^m - 2V_N^m + V_{N-1}^m}{\Delta S^2} = 0. \tag{6.86}$$

Given S_N is assumed the right-most grid point, we have introduced a "ghost point" S_{N+1} outside the computational range. One can obtain a second equation for the unknown value V_{N+1}^m from a finite difference discretisation of the PDE at S_N, e.g. from a one-step finite difference scheme in the form (6.27),

$$a_N^m V_{N-1}^{m-1} + b_N^m V_N^{m-1} + c_N^m V_{N+1}^{m-1} = A_N^m V_{N-1}^m + B_N^m V_N^m + C_N^m V_{N+1}^m,$$

and can then use (6.86) to eliminate V_{N+1}^m and get

$$\bar{a}_N^m V_{N-1}^{m-1} + \bar{b}_N^m V_N^{m-1} = \bar{A}_N^m V_{N-1}^m + \bar{B}_N^m V_N^m$$

with

$$\bar{a}_N^m = a_N^m - c_N^m, \quad \bar{b}_N^m = b_N^m + 2c_N^m, \quad \bar{A}_N = A_N^m - C_N^m, \quad \bar{B}_N^m = B_N^m + 2C_N^m.$$

The question of stability of these boundary conditions arises, and falls outside the analysis of 5.2. To make this a bit more concrete, consider the explicit Euler scheme for the Black-Scholes PDE from Example 6.1.2, evaluated at S_N and incorporating the boundary condition (6.86),

$$\partial_t^- V_N^m = -\left(\frac{1}{2}S_N^2\delta_S^2 + rS_N\delta_S - r\right)V_N^m = -(rS_N\delta_S^- - r)V_N^m,$$

which can be written as

$$V_N^{m-1} = \bar{A}_N V_{N-1}^m + \bar{B}_N V_N^m, \quad \bar{A}_N = -rS_N\Delta t/\Delta S < 0,\ \bar{A}_N = r(1 + S_N\Delta t/\Delta S) > 0. \tag{6.87}$$

The second derivative term disappears, by design, but the application of the boundary condition to the first derivative term implies that we end up using a one-sided difference in the "wrong" direction, down-winding instead of up-winding, resulting in a negative off-diagonal entry, which ultimately destroys monotonicity of the scheme. This does not rule out that the scheme might be stable nonetheless, and it does appear so in practice. An (informative but not entirely conclusive) analysis of this boundary condition for the Black-Scholes PDE using Fourier techniques is found on [Wincliff et al., 2004].

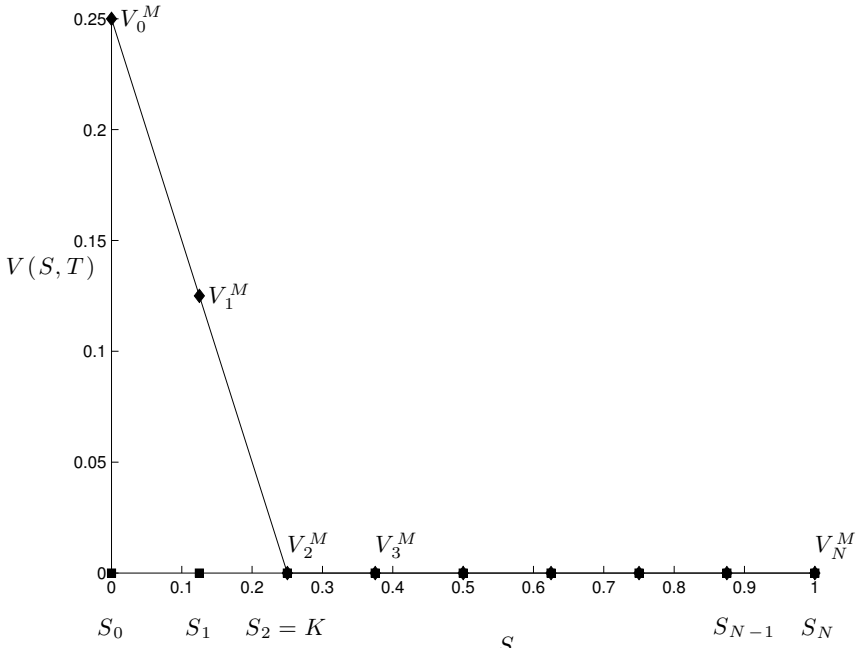

Figure 6.7: Pointwise approximation to the put payoff.

Remark 6.4.1. *The "linear boundary condition" is motivated by the goal to use a generic boundary condition which approximates the problem on the infinite axis well by one on a finite interval. This approximate boundary condition is necessary because the PDE on a finite interval is* not fully specified without boundary conditions*. An idea which sounds to good to be true is therefore the one proposed in [Tavella and Randall, 2000], to "apply the pricing equation itself as a boundary condition". Exercise 6 in 6.5 shows that – inevitably – specific boundary conditions on the solution* are *implied by this approach, although this is not transparent from the outset. There is also evidence in [Wincliff et al., 2004] that this way of applying boundary conditions results in unstable schemes.*

6.4.2 Payoff discretisation

So far, we paid little attention to the discretisation of the terminal condition. The obvious approach to evaluate the payoff at the grid points makes the approximation error at expiry zero – any deviation from that would have to be accounted for in the finite difference error analysis. Fig. 6.7 shows the representation of a standard put payoff on a grid. Shown is the special case where of the grid points coincides with the strike. In this case a piecewise linear interpolation is exact.

In the context of the convergence analysis, it is not clear though whether this is optimal. The finite difference error is determined by the truncation error, see e.g. (6.44), and we have so far tacidly assumed that the solution is sufficiently smooth for the truncation error to be well-defined. This is not the case even for standard call or put pay-offs. As we approach maturity, higher derivatives become singular around the strike. In the numerical examples

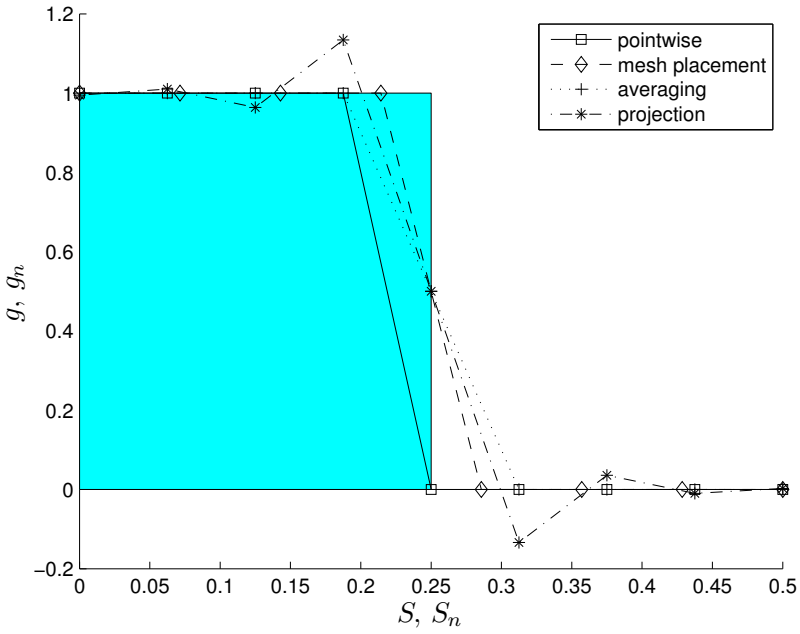

Figure 6.8: Various approximations to the digital payoff.

in 6.1.3, this has not reduced the convergence order, but it is worth investigating if a more carefully placed mesh might give more accurate solutions (see Exercise 7. in 6.5).

It is even less obvious how a discontinuous payoff – or even a Dirac delta – should be approximated optimally. Insightful discussions on this can be found in [Tavella and Randall, 2000] and [Pooley et al., 2003] and form the groundwork for the following sections. Fig. 6.8 shows a range of possibilities, on the example of a digital put, which will be discussed in turn in the following.

The good news is that we can restore second order grid convergence even for discontinuous payoffs or Dirac initial data, independent of where these singularities lie with respect to the grid. When we say "restore" we mean that a second order accurate discretisation is used, e.g. standard central differences, but a more naive payoff discretisation would result in reduced convergence order due to the non-smoothness of the payoff. However with small amendments, and the use of a strongly stable timestepping scheme as in 5.3.2, second order convergence is achieved.

Mesh placement

We first investigate the impact of the choice of grid location on the approximation, and have in mind especially step functions and Dirac deltas. In the case of the binary put as an example for a step function, a pointwise discretisation does not distinguish between options with strikes that lie within an interval $[S_k, S_{k+1})$. The difference in true value between these options will be proportional to the difference between their strikes, i.e. of order ΔS. If a grid point S_k coincides with the strike, the option value will be biased high or low by an amount

$O(\Delta S)$ if we choose $V_k^M = 1$ or $V_k^M = 0$ respectively. For the sake of symmetry, it would therefore seem advantageous to construct the grid such that the strike lies half-way between two mesh points, and indeed this is seen to restore second order grid convergence for central difference schemes.

A technical issue is the construction and then especially the refinement of such meshes. In fact, if we aim to keep the mesh uniform and the total size of the computational interval fixed, such a construction will generally not be possible. Even having constructed a particular mesh, we may find that the numerical solution is not accurate enough, or want to compare against that on a mesh with a smaller mesh size to get an indication of the discretisation error. A refined mesh has to be constructed so as to have the above symmetry around the strike. If we use e.g. bisection of the original mesh – the one where the strike was halfway between grid points – this will place a point at the strike upon refinement, and we have the same problem as previously. One way of getting around this is to modify the upper boundary S_{max} slightly, which is going to have a negligible impact on the solution if S_{max} was originally chosen large enough. To give a numerical example, if originally $K = 0.25S_{max}$, then for S_{max} replaced by $N/(N-2)S_{max}$, if N is the number of mesh intervals, all meshes with $N = 2^\nu$, $2 \leq \nu \in \mathbb{N}$, have the desired property that K lies midways between two grid points.

Already it shows that the optimised placement of mesh points can be cumbersome, and for more general payoffs with multiple jumps this can become intractable, let alone multidimensional payoffs and meshes.

Summarizing, while careful placement of grid points may give the most accurate results for simple examples, this advantage is often outweighed by the lack of generality and we focus in the following on discretisations which pick accurate grid representations on a *given* grid.

Mesh averaging

It is worth pausing and reflecting why the impact of mesh placement on convergence speed did not come up when studying the convergence of finite difference Greeks, which correspond to step function terminal data for the call and put delta, and Dirac data for the gammas (see 6.3). Assuming for a European call that $K = S_k + \theta\Delta S$ with $\theta \in [0, 1)$, a quick sketch shows that $V_{k-1}^M = V_k^M = 0$, $V_{k+1}^M = \Delta S(1-\theta)$, so

$$\Delta_k^M = 1 - \theta, \qquad \Delta_{k+1}^M = 2 - \theta, \tag{6.88}$$

$$\Gamma_k^M = \frac{\theta}{\Delta S}, \qquad \Gamma_{k+1}^M = \frac{1-\theta}{\Delta S}, \tag{6.89}$$

with the obvious constant values for Δ_n^M and Γ_n^M on either side of k and $k+1$. As the true delta and true gamma approach step functions and Dirac deltas respectively for European calls, this motivates taking the grid values in (6.88) and (6.89) to approximate step functions and Dirac terminal data. The weight of values is according to the distance of singularities to the nearest grid points, giving second order accurate finite difference solutions. As an example, for a digital put with strike at $K = S_k$, we would set $V_{k-1}^M = 1$, $V_k^M = 0.5$, $V_{k+1}^M = 0$ etc.

More generally, this suggests the following. Instead of setting $V_n^M = V(S_n, T) = g(S_n)$, the payoff is smoothened by taking the average

$$V_n^M = \frac{1}{\Delta S} \int_{S_n - \frac{1}{2}\Delta S}^{S_n + \frac{1}{2}\Delta S} V(S, T)\, \mathrm{d}S \tag{6.90}$$

over a symmetric interval of width ΔS around S_n. This coincides with the above approach for step functions, does not capture the case of Dirac data though, where a simple average cannot distinguish between locations of δs within the range of integration. We need some sort of weighted average which takes into account distance. The average (6.90) is seen to have a uniform weight function if written as

$$V_n^M = \int_{-\infty}^{\infty} \frac{1}{\Delta S} \Phi_n(S) g(S) \, \mathrm{d}S, \tag{6.91}$$

where $\Phi_n = \overline{\Phi}_n$ with

$$\overline{\Phi}_n(S) = \begin{cases} 1 & S \in [S_n - \Delta S/2, S_n + \Delta S/2), \\ 0 & \text{otherwise.} \end{cases} \tag{6.92}$$

For higher order accuracy, try instead "hat functions"

$$\widehat{\Phi}_n(S) = \begin{cases} 0 & 0 \le S \le S_{n-1} \\ \frac{S - S_{n-1}}{\Delta S} & S_{n-1} \le S \le S_n \\ \frac{S_{n+1} - S}{\Delta S} & S_n \le S \le S_{n+1} \\ 0 & S_{n+1} \le S \le S_{max} \end{cases} \tag{6.93}$$

for $0 < n < N$, and

$$\widehat{\Phi}_0(S) = \begin{cases} \frac{S_1 - S}{\Delta S} & 0 \le S \le S_1 \\ 0 & S_1 \le S \le S_{max} \end{cases},$$

and similarly for $n = N$. This is easily seen to lead to the same approximation V_n^M of Dirac data as by the finite difference Gamma (6.89) from above, and restores second order accurace in ΔS.

In the above examples, and many other typical payoff functions, the integration can be carried out analytically. A numerical quadrature rule gives a more generic tool in principle, however will run into problems for general non-smooth payoffs (the nodes of the quadrature formula will not take into account the location of the discontinuity).

Remark 6.4.2. *The approximation of non-smooth functions by smooth ones is a generally useful technique in mathematics, both for developing theory and in applications. One way to achieve this is by convolution of a non-smooth function by a smooth function of compact support, called a* mollifier*, akin (6.91) where Φ_n is chosen infinitely differentiable. As $\Delta S \to 0$, $V_n^M \to g(S_n)$, so one also speaks of approximations to the identity.*

Payoff projection

In the previous section, we have chosen discrete terminal data as either a pointwise reconstruction or average over a neighbourhood. Both are local operations.

We now take the wider view that we are ultimately trying to approximate the PDE solution by a "grid function". Under grid function we understand a function defined on $[0, S_{max}]$ but chosen so it can be represented by a finite-dimensional vector $V^M = (V_0^M, \ldots, V_N^M)'$. A way to do this is by interpolation from grid values, so define a space of the same dimension as the number of grid points, $N + 1$,

$$\mathcal{S} = \mathrm{span}\left\{\Phi_n : \; n = 0, \ldots, N\right\},$$

where "span" is the set of all linear combinations spanned by the basis functions Φ_n, which can e.g. be the piecewise constant functions $\overline{\Phi}$ or the piecewise linear ones $\widehat{\Phi}$ from above.

We seek to find the element g^s of $\mathcal{S}$ which is closest to g, and in keeping with the previous analysis, we may envisage measuring closeness in L_2 or L_∞. There is no hope of approximating discontinuous data in L_∞, hence we go with L_2,

$$g^s = \text{argmin}_{v\in\mathcal{S}}\|g - v\|_2.$$

(It is fair to observe that we cannot hope to approximate Dirac data in L_2 either, but we will still be able to make sense of this below.) As the two-norm is induced by an inner product $\langle .,. \rangle$,

$$\|v\|_2^2 = \langle v, v\rangle, \quad \langle u, v\rangle = \int_0^{S_{max}} u(S)v(S)\,\mathrm{d}S \qquad \forall\, u, v \in L_2(0, S_{max}),$$

g^s is the orthogonal projection of g onto $\mathcal{S}$, such that

$$\langle g - g^s, v\rangle = 0 \;\; \forall v \in \mathcal{S} \quad \Leftrightarrow \quad \langle g^s, \Phi_n\rangle = \langle g, \Phi_n\rangle \;\; \forall n = 0, \ldots, N.$$

Writing $g^s = \sum_{n=0}^N g_n\Phi_n$, gives the linear system

$$\begin{pmatrix} \langle\Phi_0,\Phi_0\rangle & \ldots & \langle\Phi_N,\Phi_0\rangle \\ \vdots & \ddots & \vdots \\ \langle\Phi_0,\Phi_N\rangle & \ldots & \langle\Phi_N,\Phi_N\rangle \end{pmatrix} \begin{pmatrix} g_0 \\ \vdots \\ g_N \end{pmatrix} = \begin{pmatrix} \langle g,\Phi_0\rangle \\ \vdots \\ \langle g,\Phi_N\rangle \end{pmatrix}. \tag{6.94}$$

The matrix elements can usually be calculated analytically if the Φ_n are chosen as simple functions, the right-hand side generally through quadrature.

The simplest non-trivial class of L_2 functions are piecewise constant ones as per (6.92), therefore define

$$\overline{\mathcal{S}} = \text{span}\left\{\overline{\Phi}_n :\; n = 0, \ldots, N\right\}$$

Then $\langle\Phi_i, \Phi_j\rangle = \Delta S\delta_{ij}$ and

$$\langle g, \Phi_j\rangle = \int_{S_n-\frac{1}{2}\Delta S}^{S_n+\frac{1}{2}\Delta S} g(S)\,\mathrm{d}S$$

so we recover the basic averaging method from above.

More accurate candidate grid functions are piecewise linear ones, therefore define

$$\widehat{\mathcal{S}} = \text{span}\left\{\widehat{\Phi}_n :\; n = 0, \ldots, N\right\}$$

the space of *linear splines* on the grid, with $\widehat{\Phi}_n$ as in (6.93).

The entries $\langle\widehat{\Phi}_i, \widehat{\Phi}_j\rangle$ in (6.94) are still easy to compute and give a (symmetric, positive definite) tridiagonal linear system for the basis coefficients of g^s, which can be solved with e.g. the Thomas algorithm. Simple integration gives

$$\begin{aligned} \langle\widehat{\Phi}_0, \widehat{\Phi}_0\rangle = \langle\widehat{\Phi}_N, \widehat{\Phi}_N\rangle &= \frac{1}{3}\Delta S \\ \langle\widehat{\Phi}_n, \widehat{\Phi}_n\rangle &= \frac{2}{3}\Delta S \quad n = 1, \ldots, N-1 \\ \langle\widehat{\Phi}_n, \widehat{\Phi}_{n+1}\rangle &= \frac{1}{6}\Delta S \quad n = 0, \ldots, N-1 \\ \langle\widehat{\Phi}_n, \widehat{\Phi}_j\rangle &= 0 \qquad \text{else} \end{aligned}$$

and the right-hand-side $\langle g, \widehat{\Phi}_i \rangle$ is reminiscent of the weighted average using a hat weight function as in (6.91) with Φ_n from (6.93). In fact, we essentially (i.e. with small modification at the boundary points) recover this earlier method by replacing the system matrix by a diagonal matrix with diagonal elements $D_i = \sum_j \langle \widehat{\Phi}_i, \widehat{\Phi}_j \rangle$. This is interpretable as applying a crude quadrature rule to the integration.

It turns out that both discontinuous and Dirac terminal data are represented sufficiently accurately on the grid to restore second order grid convergence.

Remark 6.4.3. *For discontinuous payoffs, convergence of this projection to the exact payoff is in L_2, not in C. The projection has features similar to the Gibbs phenomenon observed in Fourier series of discontinuous functions (see Fig. 6.8). This does not spoil the finite difference solution at a given time before expiry, as long as strongly stable timestepping schemes are used, in which case the highly oscillatory components are dampened rapidly.*

6.5 Exercises

1. For the Black-Scholes PDE, calculate analytically the truncation error of the implicit Euler central difference scheme. What behaviour of the discretisation error do you predict as
   (a) $\Delta t \to 0$, ΔS fixed;
   (b) $\Delta S \to 0$, Δt fixed;
   (c) $\Delta t, \Delta S \to 0$, $\Delta t / \Delta S$ fixed;
   (d) $\Delta t, \Delta S \to 0$, $\Delta t / \Delta S^2$ fixed.

2. The Black-Scholes equation in log-price is given by
$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 V}{\partial X^2} + (r - \tfrac{1}{2}\sigma^2)\frac{\partial V}{\partial X} - rV = 0, \quad X \in \mathbb{R}, t \in (0, T). \tag{6.95}$$
   (a) Write down the explicit finite difference scheme for this equation with M timesteps and on an infinite grid with spacing ΔX. Denote V_n^m the numerical approximation at $t = m\Delta t$, $X = n\Delta x$, $n \in \mathbb{Z}$, $m = 0, \ldots, M$.
   (b) Show that for a terminal condition of the form
$$V_n^M = \mathrm{e}^{ikn},$$
   the solution is of the form
$$V_n^m = R_0(\Delta x, \Delta t; k)^{M-m} V_n^M$$
   and find R_0. *[Hint: Prove and use that* $\mathrm{e}^{ik(n+1)} - \mathrm{e}^{ik(n-1)} = 2i\mathrm{e}^{ikn} \sin k$*.]*
   (c) Give sufficient conditions for stability of the scheme. Are these necessary?
   (d) Discuss, without proof, what this result indicates for l_2 stability of explicit finite differences for the Black-Scholes PDE in original coordinates.

3. Consider a European put option with parameters $\sigma = 0.4$, $r = 0.05$, $K = 0.25$, $T = 1$. Set $S_{max} = 1$. Implement the θ-method for a PDE of the form (6.8), and specify the coefficients μ, σ and r to compute numerical approximations to the Black-Scholes price of the put, solving

(a) the Black-Scholes PDE in S and t;

(b) the Black-Scholes PDE in log-price X and t as per (6.95);

(c) the heat equation in x and τ as in (6.18).

The terminal condition in cases 3b. and 3c. is the payoff in transformed coordinates. Use suitable upper and lower bounds for the computational domain (e.g. $X_{min} = \log(K^2/S_{max})$ etc) and asymptotic boundary conditions as appropriate.

Compare the l_∞ error (measured against the known analytical solution) of the numerical solutions, for an increasing number of grid points, say $N = 100$, 200, 400, 800, identical for all three approaches. Which method is most accurate?

4. Consider a European put with identical parameters to Exercise 3. For Crank-Nicolson with and without Rannacher start-up, compute a finite difference solution with $N = 512$ grid intervals and $M = 64$ timesteps. Plot V_n^M and V_n^0 versus S_n for $n = 0, \ldots, N$. From these finite difference solutions, compute sensitivities

$$\Delta_n^m = \frac{V_{n+1}^m - V_{n-1}^m}{2\Delta S}, \quad \Gamma_n^m = \frac{V_{n+1}^m - 2V_n^m + V_{n-1}^m}{\Delta S^2}$$

for $m = 0$ and $m = M$, and plot them over S_n, $n = 0, \ldots, N$ (using appropriate modifications for $n = 0, N$, e. g. $\Delta_0^m = -1$ etc or appropriate one-sided differences).

5. Write the Crank-Nicolson scheme with Rannacher start-up as

$$V^m = K_{1/2,\Delta t}^{M-m-r} K_{1,\Delta t/2}^{2r} V^M$$

where you have to define $K_{\theta,\Delta t}$.

(a) Show that the eigenvectors of $K_{\theta,\Delta t}$ are identical for all θ, Δt, and find the eigenvalues.

(b) If we introduce $W = (W_0, \ldots, W_N)$, the matrix of eigenvectors of K, $\widehat{V}^m = W^{-1}V^m$ contains the coordinates of $V^m = (V_0^m, \ldots, V_N^m)'$ in the eigenvector basis. Show that for $\Lambda_{\theta,\Delta t}$ the diagonal matrix of eigenvalues of $K_{\theta,\Delta t}$,

$$\widehat{V}^0 = \Lambda_{1/2,\Delta t}^{M-r} \Lambda_{1,\Delta t/2}^{2r} \widehat{V}^M.$$

(c) Plot $\widehat{V}^M$, and $\widehat{V}^0$ for $r = 0, 1, 2$.

(d) Similarly, define $\Delta^m = (\Delta_0^m, \ldots, \Delta_N^m)'$, $\Gamma^m = (\Gamma_0^m, \ldots, \Gamma_N^m)'$, and $\widehat{\Delta}^m = W^{-1}\Delta^m$, $\widehat{\Gamma}^m = W^{-1}\Gamma^m$. Compute $\widehat{\Delta}^M$ and $\widehat{\Delta}^0$, $\widehat{\Gamma}^M$, and $\widehat{\Gamma}^0$, the finite difference delta and gamma represented in the eigenvector basis, and plot them, again for $r = 0, 1, 2$. Interpret the result.

6. This exercise analyses ways of applying asymptotic boundary conditions, motivated by the following suggestion in [Tavella and Randall, 2000]. Consider an explicit discretisation

$$\delta_t^- V_n^m + DV_n^m = 0, \quad n < N,$$

where D contains the discretised spatial derivatives, e.g. for central differences for Black-Scholes $D = \sigma^2 S_n^2 \delta_S^2 + rS_n\delta_S - r$. At the upper boundary, $n = N$, we use instead

$$\delta_t^- V_N^m + DV_{N-1}^m = 0, \tag{6.96}$$

i.e. the spatial finite differences are shifted one mesh point to the left. The motivation for this is that the equation (6.96) only uses values V_{N-2}^m, V_{N-1}^m and V_N^m, hence "closes" the system of equations for $n = 0, 1, \ldots, N-1$.

(a) For the Black-Scholes PDE in log-price, (6.95), write down the explicit Euler central difference scheme at a general interior grid point n, specifically at point $N-1$, and the equation at N using the boundary condition (6.96). Letting $\Delta S \to 0$ and comparing the equations at $N-1$ and N, show that this scheme is consistent with the boundary condition

$$\frac{\partial V}{\partial S}(S_N, t_m) = 0.$$

(b) Repeat the above analysis, but now with the Black-Scholes PDE in *original* coordinates. What is the hidden boundary condition now?

7. For this numerical experiment, consider a standard put payoff

$$G(S) = \max(K - S, 0),$$

and a binary payoff

$$G(S) = \begin{cases} 1 & \text{if } S < K \\ 0 & \text{else} \end{cases}$$

Data are as in Exercise 3. The exact time 0 Black-Scholes prices of options with these payoffs can be derived from (6.78) and (6.81) via put-call parity (6.79) and similarly for the digitals.

Study a fixed mesh with $N = 256$ grid intervals, $\Delta S = 1/N$, grid points S_n, $n = 0, \ldots, N$. Pick strikes $K = 0.25$ (i. e. coinciding with grid point S_{64}), $K = 0.25 + \Delta S/2$ (symmetrically between two grid points), $K = 0.25 + (\pi - 3)\Delta S$ (neither of the above).

Use Crank-Nicolson timestepping with Rannacher start-up and $M = 64$ steps. The solution $V(S, 0)$ at $S = S_{64} = 0.25$ can be approximated by V_{64}^0, the finite difference solution at this point. Also consider the l_∞ error.

(a) Use as discrete terminal condition

$$V_n^M = V(S_n, T) = G(S_n)$$

where G is one of the two payoffs above. Compare the finite difference solution V_{64}^0 to the Black-Scholes price $V(S_{64}, 0) = V(0.25, 0)$, for the standard put and the binary put.

(b) Calculate analytically an averaged terminal condition

$$V_n^M = \frac{1}{\Delta S} \int_{S_n - \frac{1}{2}\Delta S}^{S_n + \frac{1}{2}\Delta S} V(S, T)\, \mathrm{d}S = \frac{1}{\Delta S} \int_{S_n - \frac{1}{2}\Delta S}^{S_n + \frac{1}{2}\Delta S} G(S)\, \mathrm{d}S,$$

again for standard and binary put, and implement this as terminal condition for the finite difference scheme. Recompute finite difference solutions and errors, and compare to the ones from above.

(c) Solve the same problem with L_2 projection of the payoff onto a basis of hat functions. Plot and discuss the numerical solutions V^M, V^{M-1}, V^{M-2}, i.e. before, during, and after Rannacher start-up.

Bibliography

[Achdou and Pironneau, 2005] Achdou, Y. and Pironneau, O. (2005). *Computational Methods for Option Pricing*. Frontiers in Applied Mathematics. SIAM.

[Carter and Giles, 2007] Carter, R. and Giles, M. B. (2007). Sharp error estimates for discretizations of the 1D convection-diffusion equation with Dirac initial data. *IMA Journal of Numerical Analysis*, 27:406–425.

[Crank and Nicolson, 1947] Crank, J. and Nicolson, P. (1947). A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. *Proc. Camb. Phil. Soc.*, 43:50–67.

[Evans, 2004] Evans, L. C. (2004). *Partial Differential Equations*. AMS.

[Fiedler, 2008] Fiedler, M. (2008). *Special Matrices and Their Applications in Numerical Mathematics*. Dover, second edition edition.

[Forsyth and Vetzal, 2002] Forsyth, P. and Vetzal, K. (2002). Quadratic convergence of a penalty method for valuing American options. *SIAM J. Sci. Comp.*, 23:2095–2122.

[Forsyth et al., 2002] Forsyth, P., Vetzal, K., and Zvan, R. (2002). Convergence of lattice and PDE methods for valuing path dependent options using interpolation. *Review of Derivatives Research*, 5:273–314.

[Giles and Glasserman, 2006] Giles, M. and Glasserman, P. (2006). Smoking adjoints: fast monte carlo greeks. *RISK*.

[Giles and Carter, 2006] Giles, M. B. and Carter, R. (2006). Convergence analysis of Crank-Nicolson and Rannacher time-marching. *Journal of Computational Finance*, 9(4):89–112.

[Glowinski, 1985] Glowinski, R. (1985). Viscous flow simulation by finite element methods and related numerical techniques. In Murman, E. and Abarbanel, S., editors, *Progress and Supercomputing in Computational Fluid Dynamics*, pages 173–210. Birkhäuser, Boston, MA.

[Gordon, 1968] Gordon, P. (1968). A note on a maximum principle for the Du Fort-Frankel difference equation. *Math. Comp.*, 22:437–439.

[Griewank and Walther, 2008] Griewank, A. and Walther, A. (2008). *Evaluating Derivatives – Principles and Techniques of Algorithmic*. SIAM.

[Grimmet and Stirzaker, 2001] Grimmet, G. and Stirzaker, D. (2001). *Probability and Random Processes*. OUP.

[Howison, 2007] Howison, S. (2007). A matched asymptotic expansions approach to continuity corrections for discretely sampled options. Part 2: Bermudan options. *Applied Mathematical Finance*, 14:91–104.

[Howison and Steinberg, 2007] Howison, S. and Steinberg, M. (2007). A matched asymptotic expansions approach to continuity corrections for discretely sampled options. Part 1: Barrier options. *Applied Mathematical Finance*, 14:63–89.

[Larsson and Thomée, 2005] Larsson, S. and Thomée, V. (2005). *Partial Differential Equations with Numerical Methods*. Springer.

[Morton and Mayers, 2005] Morton, K. W. and Mayers, D. F. (2005). *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, second edition.

[Ockendon et al., 2003] Ockendon, J., Howison, S., Lacey, A., and Movchan, A. (2003). *Applied Partial Differential Equations*. OUP.

[Pooley et al., 2003] Pooley, D. M., Vetzal, K. R., and Forsyth, P. A. (2003). Convergence remedies for non-smooth payoffs in option pricing. *J. Comp. Fin.*

[Rannacher, 1984] Rannacher, R. (1984). Finite element solution of diffusion problems with irregular data. *Numerische Mathematik*.

[Research website on automatic differentiation,] Research website on automatic differentiation. `www.autodiff.org`.

[Richtmyer and Morton, 1994] Richtmyer, R. D. and Morton, K. W. (1994). *Difference Methods for Initial-Value Problems*. Krieger.

[Seydel, 2006] Seydel, R. (2006). *Tools for Computational Finance*. Springer.

[Shaw, 1998] Shaw, W. T. (1998). *Modelling Financial Derivatives with MATHEMATICA®*. CUP.

[Shreve, 2004] Shreve, S. E. (2004). *Stochastic Calculus for Finance II, Continuous-time Models*. Springer.

[Smith, 1985] Smith, J. D. (1985). *Numerical Solution of Partial Differential Equations – Finite Difference Methods*. Oxford Applied Mathematics and Computing Series. Oxford University Press, Oxford, third edition.

[Steele, 2001] Steele, J. M. (2001). *Stochastic Calculus and Financial Applications*. Springer.

[Sun et al., 2003] Sun, Z.-Z., Yan, N.-N., and Zhu, Y.-L. (2003). Convergence of second-order difference schemes and extrapolation algorithm for degenerate linear parabolic equations in finance. privately sent by Y-L Z from UNC at Charlotte.

[Tavella and Randall, 2000] Tavella, D. and Randall, C. (2000). *Pricing Financial Instruments: The Finite Difference Method*. Wiley.

[Taylor, 1996a] Taylor, E. (1996a). *Partial Differential Equations I: Basic Theory*. Springer.

[Taylor, 1996b] Taylor, E. (1996b). *Partial Differential Equations II: Qualitative Studies of Linear Equations.* Springer.

[Taylor, 1996c] Taylor, E. (1996c). *Partial Differential Equations III: Nonlinear Equations.* Springer.

[Thomée, 1990] Thomée, V. (1990). Finite difference methods for linear parabolic equations. In Ciarlet, P. and Lions, J., editors, *Handbook of Numerical Analysis*, volume 1. North–Holland.

[Wade et al., 2005] Wade, B., Khaliq, A., Siddique, M., and Yousuf, M. (2005). Smoothing with positivity-preserving pad'e schemes for parabolic equations. *Numerical Methods for Partial Differential Equations*, 21(3):553–573.

[Williams, 1980] Williams, W. E. (1980). *Partial Differential Equations.* OUP.

[Wilmott et al., 1995] Wilmott, P., Howison, S., and Dewynne, J. (1995). *The Mathematics of Financial Derivatives: A Student Introduction.* CUP.

[Wincliff et al., 2004] Wincliff, H., Forsyth, P., and Vetzal, K. (2004). Analysis of the stability of the linear boundary condition for the Black-Scholes equation. *J. Comp. Fin.*, 8(1):65–92.