## 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

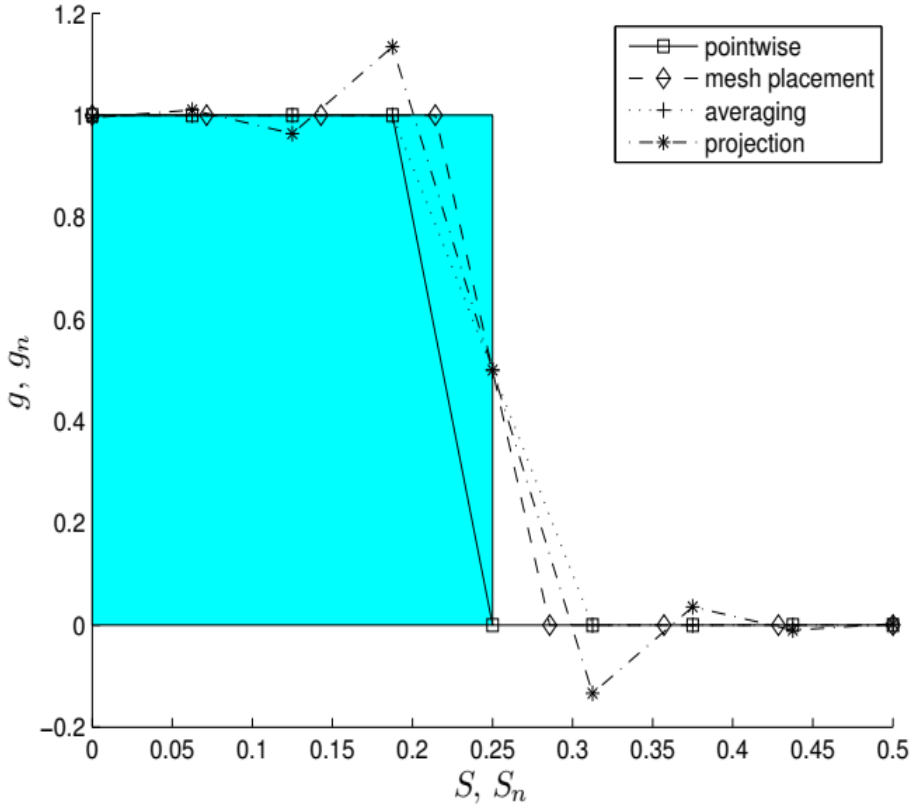


Figure 6.8: Various approximations to the digital payoff.

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

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 $\Delta S$. 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 $V^M_{k-1} = V^M_k = 0$, $V^M_{k+1} = \Delta S(1-\theta)$, so

$$\Delta^M_k = 1 - \theta, \qquad \Delta^M_{k+1} = 2 - \theta, \tag{6.88}$$

$$\Gamma^M_k = \frac{\theta}{\Delta S}, \qquad \Gamma^M_{k+1} = \frac{1-\theta}{\Delta S}, \tag{6.89}$$

with the obvious constant values for $\Delta^M_n$ and $\Gamma^M_n$ 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 $V^M_{k-1} = 1$, $V^M_k = 0.5$, $V^M_{k+1} = 0$ etc.

More generally, this suggests the following. Instead of setting $V^M_n = V(S_n, T) = g(S_n)$, the payoff is smoothened by taking the average

$$V^M_n = \frac{1}{\Delta S} \int_{S_n - \frac{1}{2}\Delta S}^{S_n + \frac{1}{2}\Delta S} V(S, T)\, \mathrm{d}S \tag{6.90}$$

over a symmetric interval of width $\Delta S$ 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 $\delta$s 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 $\Delta S$.

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 $\Phi_n$ 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, $N + 1$,

$$\mathcal{S} = \operatorname{span} \{\Phi_n : \ n = 0, \ldots, N\},$$

where "span" is the set of all linear combinations spanned by the basis functions $\Phi_n$, 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 $L_\infty$. There is no hope of approximating discontinuous data in $L_\infty$, 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 $\Phi_n$ 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 $\Phi_n$ 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.*

# Exercise

This exercise analyses ways of applying asymptotic boundary conditions, motivated by the following suggestion in [Tavella and Randall, 2000]. Consider an explicit discretisation

$$\delta_t^- V_n^m + D V_n^m = 0, \quad n < N,$$

where $D$ contains the discretised spatial derivatives, e.g. for central differences for Black-Scholes $D = \sigma^2 S_n^2 \delta_S^2 + r S_n \delta_S - r$. At the upper boundary, $n = N$, we use instead

$$\delta_t^- V_N^m + D V_{N-1}^m = 0, \tag{6.96}$$

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?