Finite difference schemes with exact recovery of vanilla option prices

Peter Austing shows how to set up finite difference solvers to exactly recover the prices of all vanilla options on the grid. The approach leads to a specific discretisation of Dupire's formula. It can be applied to local stochastic volatility and interest rate hybrid models and eliminates many of the convergence and stability issues that arise in the partial differential equation approach to exotics pricing

Finite difference schemes are widely used in quantitative finance to solve the partial differential equations (PDEs) that arise when pricing exotic options. The approach is to first discretise the PDE on a grid with spot and time axes, and then to solve backwards from the expiry time to the valuation time, and finally to interpolate the solution on the grid to the current spot rate. In more complex cases, there may be additional factors, represented as additional grid axes, for stochastic volatility or stochastic interest rates and dividends.

In the standard case, we are given an implied volatility surface, and we set up the PDE to match vanilla prices. The most fundamental way to achieve this is with the Dupire (1994) local volatility model. This can then be extended to local stochastic volatility (LSV) models, or to models including stochastic rates, with a new local volatility correction calculated using a forward PDE sweep.

Over the years, much expertise has been gained in the finance industry in solving these equations numerically. The main considerations are the order of convergence in both the time dimension and the spot and other space dimensions, the stability of the finite difference scheme, and the stability of second-order Greeks such as gamma. All of these must be balanced against the computation time required. No doubt each institution has found slightly different rules of thumb that work, but, broadly speaking, industry best practice can be summarised as:

(1) Use a number (say 4) of fully implicit steps after any discontinuities in the payout.
(2) Use Crank-Nicolson stepping (1D) or Craig-Sneyd stepping (2D and 3D) away from discontinuities.
(3) Work in log-spot space.
(4) Transform the Dupire formula into implied volatility (rather than vanilla call option) space, as in Gatheral (2006), before discretising.
(5) Discretise the time derivative in Dupire's local volatility formula on the time grid, but do the spot derivatives analytically.
(6) Place strikes and discrete barriers midway between spot grid points, and place continuous barriers exactly on the grid.
(7) Do not place the current spot on the grid.
(8) Keep the grid fixed when bumping for risk.
(9) Use a smooth interpolation scheme (eg, cubic) to recover the price at the current spot from the solution vector.

This and similar protocols have allowed investment banks to solve PDEs with sufficient accuracy in most situations, but the risk stability and price convergence are always a play-off against computation time, and it is hard to fully eliminate occasional risk explosions or convergence failure for finely defined payouts.

When one tries to look into the underlying reason for each of the above rules, one quickly finds more questions than answers. A secondary purpose of this article is to propose a new set of rules with the reasoning behind them made explicit.

Our results are set in the context of two key works: Andreasen & Huge (2011a,b). A core principle of the former is that the adjoint of a discretised backward pricing equation is a discretisation of the forward equation (see also Huge (2018)). The authors use this to set up a Monte Carlo simulation with discretisation exactly matching an original finite difference scheme. We shall depend on the principle of adjointness throughout our article. In Andreasen & Huge (2011b), the authors use a large stride implicit finite difference scheme with a numerical solution to interpolate missing implied volatilities on a sparse grid. In our case, the grid is not sparse, meaning our scheme cannot be used for interpolation. However, that allows us to solve the system analytically.

In the sections below, we develop finite difference schemes that recover vanilla option prices exactly. This allows us to eliminate the key convergence and stability issues that have made the field of exotic options pricing so challenging to date.

Discretisation

We are going to solve a PDE on a grid. The spatial grid $\{x_1 < x_2 < \cdots < x_n\}$ will be used to model the spot rate. We do not need the x_i to be evenly spaced. To keep track of boundary conditions, we insist on choosing $x_1 = 0$ and x_n to be large. The remaining points could be uniform in spot space, or in log-spot space, or they could be non-uniform, perhaps concentrating points around trade features of interest. Similarly, we discretise time on a grid $\{t^0 < t^1 < \cdots < t^T\}$, where t^0 is the valuation time, and t^T is the future expiry time of the option. We will use lower indexes for space and upper indexes for time.

We assume that we are given a set of call option prices $\{c_i^\tau,\ i = 1, \ldots, n-1\}$, where the call option with price c_i^τ has strike x_i and expiry time t^τ. In practice, these will be valuations from an interpolated implied volatility surface $\sigma(K, t)$. For the boundaries, we set:

$$c_i^\tau = 0, \quad i = n \tag{1}$$

$$c_i^\tau = df^\tau(F^\tau - x_i) \equiv df^\tau F^\tau, \quad i = 1 \tag{2}$$

where df^τ is the discount factor and F^τ is the forward to time t^τ. Finally, we define discounted risk-neutral probabilities p_i^τ so that:

$$c_i^\tau = \sum_{j=1}^{n} p_j^\tau (x_j - x_i)_+, \quad i = 1, \ldots, n-1 \tag{3}$$

By direct calculation, it is easy to see that:

$$c_i^\tau - c_{i-1}^\tau = -(x_i - x_{i-1}) \sum_{j=i}^{n} p_j \tag{4}$$

and therefore:

$$p_i^\tau = \frac{c_{i+1}^\tau - c_i^\tau}{x_{i+1} - x_i} - \frac{c_i^\tau - c_{i-1}^\tau}{x_i - x_{i-1}}, \quad i = 2, \dots, n-1 \tag{5}$$

$$p_n^\tau = \frac{c_{n-1}^\tau}{x_n - x_{n-1}} \tag{6}$$

This is familiar as a discretisation of the Breeden-Litzenberger formula.

The value of p_1 is irrelevant to satisfying the above equations. Meanwhile, we can make p_n as small as we like by taking x_n large enough. Mathematically, we work in the formal limit of $x_n \to \infty$ and $p_n = 0$. However, on a computer, x_n does not need to be particularly large to achieve full machine precision, and it is much more convenient to work with a large finite x_n than to keep track of the terms arising when $x_n \to \infty$.

We write L for the linear operator defined by the relation (5) together with the outer boundary conditions $L_{1j} = L_{nj} = 0$, $j = 1, \dots, n$. It is a tridiagonal matrix, and it is a discretisation of the second derivative operator. We constructed it as the inverse of the relation (3), and therefore we have:

$$p^\tau = Lc^\tau \tag{7}$$

$$c^\tau = L^{-1} p^\tau \tag{8}$$

where:

$$(L^{-1})_{ij} = (x_i - x_j)_+ \tag{9}$$

is an upper triangular matrix.

Although we are using the notation L^{-1}, it is important to note that the above relations hold only for our particular vector space in which $c_1 = 0$ and $p_n = 0$. There is, however, an additional interesting relation between L and L^{-1}:

$$(L^{-1} L^\dagger)_{ij} = (\text{diag}\{0, 1, \dots, 1, 0\})_{ij} \tag{10}$$

which holds in the sense of true matrix multiplication. Here, we use $\dagger$ to represent matrix transpose, to avoid confusion with terminal expiry times T.

Local volatility

We begin with the stochastic differential equation (SDE):

$$\frac{\mathrm{d}S}{S} = \mu \,\mathrm{d}t + \sigma(S, t) \,\mathrm{d}W \tag{11}$$

Let us consider how to discretise the forward Kolmogorov equation:

$$\frac{\partial q}{\partial t} = -\frac{\partial}{\partial S} S\mu q + \frac{1}{2} \frac{\partial^2}{\partial S^2} S^2 \sigma^2(S, t) q - rq \tag{12}$$

on the strike grid $\{x_i\}$, where $q(S, t)$ is the discounted risk-neutral probability density function. To do so, we define $\Delta x_i = \frac{1}{2}(x_{i+1} - x_{i-1})$ and the diagonal matrix:

$$U_{ii} = \text{diag}((\Delta x_i)^{-1}), \quad 2 \leqslant i \leqslant n-1 \tag{13}$$

$$U_{ii} = 0, \quad \text{otherwise} \tag{14}$$

Then we discretise q by replacing it with p, where p are the probabilities defined in the previous section.[1]

Later, when we discretise the backward pricing equation, we will define the matrix D by $(Dy)_i = y_{i+1} - y_{i-1}$ for $i = 2, \dots, n-1$, and $(Dy)_1 = (Dy)_n = 0$. Then, in our scheme, $\partial/\partial S$ is replaced by UD. Similarly, $\partial^2/\partial S^2 \to UL$. We represent S with the matrix $X = \text{diag}(x_i)$ and σ^2 with diagonal matrix V.

As the forward equation is the adjoint of the backward equation, we discretise (12) by replacing $-\partial/\partial S \to D^\dagger U$ and $\partial^2/\partial S^2 \to L^\dagger U$, where $\dagger$ means matrix transpose.

Initially, we will use a fully implicit scheme; so, the discretised PDE becomes:

$$\frac{(p^{\tau+1} - p^\tau)}{\Delta t} = D^\dagger X \mu^{\tau+1} U p^{\tau+1} + \tfrac{1}{2} L^\dagger X^2 V U p^{\tau+1} - r^{\tau+1} p^{\tau+1} \tag{15}$$

For ease of notation, we will drop the τ indexes on the drift μ and interest rate r for the moment. Our task is to find the local volatilities V such that vanilla options are exactly repriced.

To do so, we multiply by L^{-1} and use the relations $p = Lc$ and $c = L^{-1} p$ together with (10), and obtain:

$$\frac{(c^{\tau+1} - c^\tau)_i}{\Delta t} = (L^{-1} D^\dagger X \mu U L c^{\tau+1} + \tfrac{1}{2} X^2 V U L c^{\tau+1} - r c^{\tau+1})_i, \quad 2 \leqslant i \leqslant n-1 \tag{16}$$

This formula is expensive to compute because L^{-1} is triangular and so requires $\mathcal{O}(n^2)$ operations. To deal with this, we define a new tridiagonal matrix J:

$$J_{i,i-1} = -\frac{x_i(x_{i+1} - x_i)}{(x_{i+1} - x_{i-1})(x_i - x_{i-1})} \tag{17}$$

$$J_{i,i} = \frac{x_i(x_{i+1} - x_i) - x_{i+1}(x_i - x_{i-1})}{(x_{i+1} - x_i)(x_i - x_{i-1})} \tag{18}$$

$$J_{i,i+1} = \frac{x_i(x_i - x_{i-1})}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} \tag{19}$$

$$J_{ij} = 0, \quad \text{otherwise} \tag{20}$$

With some algebra, one can verify:

$$-LJ = L(L^{-1} D^\dagger X U L) \tag{21}$$

in the limit of $x_n \to \infty$. This is enough to allow us to replace $L^{-1} D^\dagger X U L$ with $-J$.

Then our formula becomes:

$$\frac{(c^{\tau+1} - c^\tau)}{\Delta t} = -\mu J c^{\tau+1} + \tfrac{1}{2} X^2 V U L c^{\tau+1} - r c^{\tau+1} \tag{22}$$

which we can rearrange to find the local volatilities $(\sigma_i^\tau)^2$, which are the diagonal elements of the matrix V:

$$(\sigma_i^\tau)^2 = \frac{(c_i^{\tau+1} - c_i^\tau)/\Delta t + (\mu J + r) c_i^{\tau+1}}{\frac{1}{2}(x_i)^2 (U L c^{\tau+1})_i}, \quad 2 \leqslant i \leqslant n-1 \tag{23}$$

$$(\sigma_i^\tau)^2 = 0, \quad \text{otherwise} \tag{24}$$

[1] *Strictly speaking, we have multiplied both sides of the equation by* $\mathrm{d}S$ *and discretised* $q\,\mathrm{d}S$ *with* p.

The matrix J can be thought of as an unusual discretisation of the identity operator. Its appearance in (23), together with the precise definitions of L and U (and of r and μ that will follow), is the key mathematical result of this article.

Having set up our discretisation scheme and our formula for the local volatilities in this way, vanilla options are guaranteed to be priced correctly. To see this, we can rewrite the implicit scheme for the forward equation as:

$$p^{\tau+1} = (1 - \Delta t A^{\tau})^{-1} p^{\tau} \tag{25}$$

where $A = \mu D^{\dagger} X U + \frac{1}{2} L^{\dagger} X^2 V U - r$.

Solving this forwards tells us the discounted probability distribution at the expiry time T is:[2]

$$p^T = \left[\prod_{\tau=0}^{T-1} (1 - \Delta t A^{\tau})^{-1}\right] p^0 \tag{26}$$

By construction, the probability distribution p^T correctly values vanilla call options whenever the strike is one of the grid points. Let us write the discretised payout of such a call option as a vector q:

$$q_i = (x_i - K)_+ \tag{27}$$

where the strike K is equal to one of the spot grid points x_j. Then, denoting transpose by †, we know that the PV of the option is given by:

$$c_j^T = (p^T)^{\dagger} q \tag{28}$$

$$= (p^0)^{\dagger}\left[\prod_{\tau=0}^{T-1} (1 - \Delta t (A^{\tau})^{\dagger})^{-1}\right] q \tag{29}$$

Equation (29) is exactly the backward pricing equation, giving the solution vector:

$$\left[\prod_{\tau=0}^{T-1} (1 - \Delta t (A^{\tau})^{\dagger})^{-1}\right] q \tag{30}$$

The scalar product of the solution with $(p^0)^{\dagger}$ is a very specific choice for interpolating the solution onto the initial spot. To understand this, we note that if the initial spot happens to be on the grid, then p^0 simplifies to a Kronecker delta distribution.

If we write q^{τ} for the solution vector at grid time τ, then (30) can be rewritten as:

$$\frac{q^{\tau-1} - q^{\tau}}{\Delta t} = (A^{\tau})^{\dagger} q^{\tau-1}$$

$$= (\mu^{\tau} X U D + \tfrac{1}{2} V^{\tau} X^2 U L - r^{\tau}) q^{\tau-1} \tag{31}$$

and we see that our backward scheme is also an implicit scheme. This is important. An implicit forward scheme has an equivalent adjoint backward scheme that is also implicit.

We have shown that our choices of discretisation for the differential operator and for Dupire's local volatility formula ensure that any call option with strike equal to one of the grid points is repriced exactly.

Cash and forward contracts

We have, up to now, been slightly cavalier with the discretisation of the drift μ and the domestic interest rate r. Let us consider first a cash payment at time τ. This is represented on the solution grid as a vector of all ones $q = (1, \ldots, 1)^{\dagger}$. As both D and L annihilate q, our discretised pricing equation (31) becomes:

$$\frac{q^{\tau-1} - q^{\tau}}{\Delta t} = -r^{\tau} q^{\tau-1} \tag{32}$$

Since $q^{\tau-1}$ is equal to q^{τ} discounted back by discount factor $df^{\tau-1,\tau}$, this tells us:

$$r^{\tau} = \frac{(df^{\tau-1,\tau})^{-1} - 1}{\Delta t} \tag{33}$$

that is, the interest rate to use between grid points $\tau - 1$ and τ in our discretisation is the market simple compound rate. When the interest rate is discretised this way, we are guaranteed to exactly reprice cashflows that happen at the grid times.[3]

If we price instead a simple strikeless forward contract, the solution vector at τ is $q = (x_1, \ldots, x_n)^{\dagger}$. This is annihilated by L but satisfies $XUDq = q$. We represent our discretised drift μ^{τ} as the difference between the dividend rate d^{τ} and the domestic interest rate r^{τ}. Then the discretised pricing equation becomes:

$$\frac{q^{\tau-1} - q^{\tau}}{\Delta t} = -d^{\tau} q^{\tau-1} \tag{34}$$

so that:

$$d^{\tau} = \frac{(df_{\text{div}}^{\tau-1,\tau})^{-1} - 1}{\Delta t} \tag{35}$$

where df_{div} is the dividend yield discount factor between $\tau - 1$ and τ. Thus, the correct discretisation of the dividend rate is, again, the simple compounding rate between the grid points.

With these discretisations of the interest rate and dividend yield, we are guaranteed that cashflows and forward contracts are exactly repriced. If we were using a semi-implicit scheme, such as Crank-Nicolson, rather than a fully implicit one, (33) and (35) would be modified in the obvious way.

Conservation of probability

We can rewrite the discretised forward equation (25) as:

$$p^{\tau} = [1 - \Delta t(\mu D^{\dagger} X U + \tfrac{1}{2} L^{\dagger} X^2 V U - r)] p^{\tau+1} \tag{36}$$

and denote the total discounted probability at τ by $\omega^{\tau} = \sum_i p_i^{\tau}$. Then, since the columns of both operators $L^{\dagger}$ and $D^{\dagger}$ sum to zero, we have:

$$\omega^{\tau} = (1 + r^{\tau} \Delta t)\omega^{\tau+1} \tag{37}$$

Equation (5) tells us $\omega^0 = 1$, and as r^{τ} is the simple compounding interest rate from τ to $\tau + 1$, we see that the total probability resulting from the forward equation is conserved and is equal to 1.

[2] *As A^{τ} is tridiagonal, the matrix inverse can be computed in $\mathcal{O}(n)$ time in the usual way using the Thomas algorithm.*

[3] *In practice, when there is a two-day spot settlement and cash payment lag, we use the old quant's trick of applying discount factors $D_f^{\tau+2\ days}$ at grid point τ so we model the true settled spot on the grid.*

Notes on stability

We chose to exhibit our discretisation using an implicit scheme, but it is straightforward to adapt it to an explicit scheme or Crank-Nicolson stepping. In the explicit case, the local volatility formula (23) would be replaced by:

$$(\sigma_i^\tau)^2 = \frac{(c_i^{\tau+1} - c_i^\tau)/\Delta t + (\mu J + r)c_i^\tau}{\frac{1}{2}(x_i)^2(ULc^\tau)_i} \tag{38}$$

In the denominator, we now have probabilities $p^\tau = Lc^\tau$ instead of $p^{\tau+1}$. This causes the local volatilities to blow up if the time step Δt is too large. In that case, we would be trying to fit to non-zero values of $c_i^{\tau+1}$ when the probability p_i^τ is extremely close to zero. The problem goes away with our implicit formula, as non-zero values of $c_i^{\tau+1} - c_i^\tau$ in the numerator are always associated with non-zero values of $p_i^{\tau+1}$ in the denominator.

It is beyond the scope of this article to provide a full stability analysis for the numerical scheme, but it is nonetheless interesting to make the following observation. Irrespective of the stepping scheme, if the local volatilities are real and finite, we can be sure the scheme is stable. By construction, each finite difference step maps a vector of probabilities onto the next vector of probabilities, and there is no way that introducing noise into the solution can cause this to blow up. This is a pleasing feature of our approach. In the traditional approach, stability analysis is done assuming constant PDE coefficients, so we have to rely on experience when we assume an implicit scheme is stable. By contrast, in our new approach, the question of stability is transferred to the calculation of the local volatilities, and we can conjecture that the implicit version is always stable when there is no arbitrage in the surface.

Since our approach gives exact repricing of vanilla options for all grid strikes and expiries, we see no need of semi-explicit schemes and therefore recommend implicit ones. Future authors may establish whether using Crank-Nicolson leads to better convergence for barrier or American options.

Numerical results and implementation notes

In spite of the fiddly algebra that led us to the discretisation formulas, applying them in practice is straightforward. It is, however, essential to set up the matrix operators exactly as they are set out in the earlier sections.

To proceed, we first calculate the interest and dividend rates using (33) and (35). We then calculate the local variance matrices V^τ using (23). When calculating the probabilities in the denominator of (23) from an implied volatility surface, it is best to choose out-of-the-money call or put options. Finally, step the solution backwards with (31), using the Thomas algorithm to perform the matrix inversion.

If there is arbitrage in the volatility surface, it will be manifested by the numerator or denominator of the local volatility formula (23) being non-positive. An advantage of our scheme is that the impact of fixing arbitrage can be confined to regions we care about less. For example, one could decide that three-month at-the-money options are most liquid and work forwards, backwards and sideways through the grid from that point, adjusting prices enough to fix any arbitrage. Once this is done with (presumably) a small number of modifications to option prices, the scheme is guaranteed to return those prices. It is essential to perform this step if the volatility surface is bad, as the Thomas algorithm can be unstable if negative local variances are encountered.

Figure 1 shows the solution of a simple Black-Scholes price on a 100 × 100 grid using the standard discretisation outlined in the introduction with constant Black-Scholes coefficients, versus our new approach. While the error is as high as 5×10^{-5} (0.5 basis points) in the standard approach, it is at worst 10^{-14} with the new discretisation.

However, to fully demonstrate the power of the algorithm, figure 2 shows the result and error when we generate option prices with some quite aggressive stochastic alpha, beta, rho (SABR) parameters. In this example, the spot grid was generated with 18 points using a random number generator, and we additionally set $x_1 = 0$ and $x_{20} = 10^{10}$. In spite of the random grid and extreme SABR volatility of volatility (vol-of-vol), convergence is achieved to machine precision. It is inconceivable that this problem could be tackled on such a coarse grid with the methods available before now. We have observed similar results for a wide variety of grid shapes and densities.

Simple local stochastic volatility

In the remaining sections, we sketch out how to apply the method to local stochastic volatility (LSV) models, as in Jex (1999) and Lipton (2002). Wyns & in 't Hout (2018) use the adjointness principle to set up a discretised local stochastic volatility model with prices matching a given local volatility discretisation. Our approach is similar, but as we have managed to write down a discretisation of Dupire's formula such that vanilla options are exactly repriced, the LSV also reprices vanillas.

As a warm up, we consider the simplest possible 'stochastic' volatility model:

$$\frac{\mathrm{d}S}{S} = \mu\,\mathrm{d}t + \sigma U(S,t)\,\mathrm{d}W \tag{39}$$

$$\sigma = \begin{cases} \sigma_+ & \text{probability} = \frac{1}{2} \\ \sigma_- & \text{probability} = \frac{1}{2} \end{cases} \tag{40}$$

That is, at time 0, we toss a coin to decide if we are in a high volatility state σ_+ or a low volatility state σ_-. Our task is to compute the local volatility correction function $U(S,t)$ on the finite difference grid in such a way that we exactly reprice vanilla options. To do so, we will perform a forward induction à la Jamshidian (1991).

At time step τ, the probability distribution has two components, p_+^τ and p_-^τ, and our (implicit) PDE scheme for the probabilities becomes:

$$\begin{aligned}\frac{p_+^{\tau+1} - p_+^\tau}{\Delta t} &= -\mu DXp_+^{\tau+1} + \tfrac{1}{2}(\sigma_+)^2 LV^\tau p_+^{\tau+1} - rp_+^{\tau+1} \\ &:= A_+ p_+^{\tau+1}\end{aligned} \tag{41}$$

$$\begin{aligned}\frac{p_-^{\tau+1} - p_-^\tau}{\Delta t} &= -\mu DXp_-^{\tau+1} + \tfrac{1}{2}(\sigma_-)^2 LV^\tau p_-^{\tau+1} - rp_-^{\tau+1} \\ &:= A_- p_-^{\tau+1}\end{aligned} \tag{42}$$

where:

$$V_{ii}^\tau = (x_i)^2(U_i^\tau)^2 \tag{43}$$

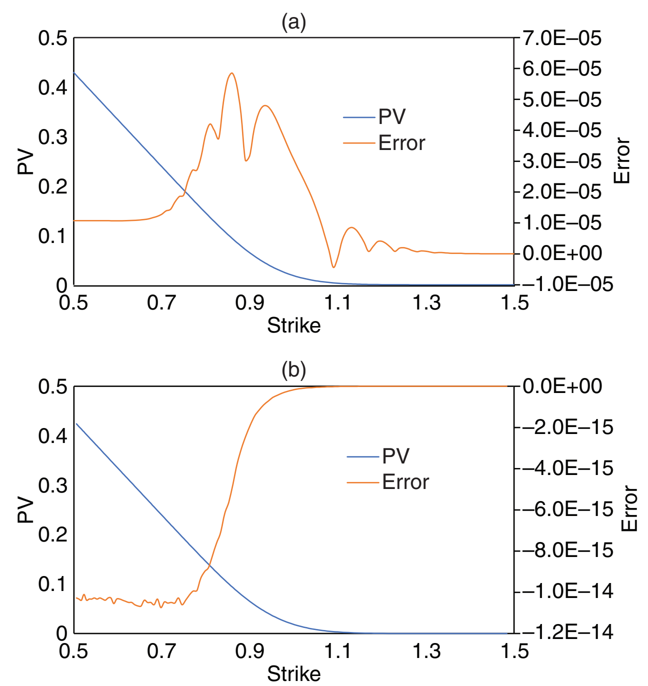
We assume inductively that we have already computed p_+^τ and p_-^τ. We wish to calculate A^τ in such a way that stepping to $p_+^{\tau+1}$ and $p_-^{\tau+1}$ yields exact vanilla prices. To achieve this, we require that $p^\tau = \frac{1}{2}(p_+^\tau + p_-^\tau)$ satisfies $L^{-1}p^\tau = c^\tau$ for every τ. Then, adding (41) and (42) and applying L^{-1} and the relations from the previous section tells us:

$$(U_i^\tau)^2 = \frac{(c_i^{\tau+1} - c_i^\tau)/\Delta t + (\mu Jc^{\tau+1} + rc^{\tau+1})_i}{\frac{1}{4}(x_i)^2(\sigma_+^2 p_+^{\tau+1} + \sigma_-^2 p_-^{\tau+1})_i} \tag{44}$$

This can be rewritten in the form:

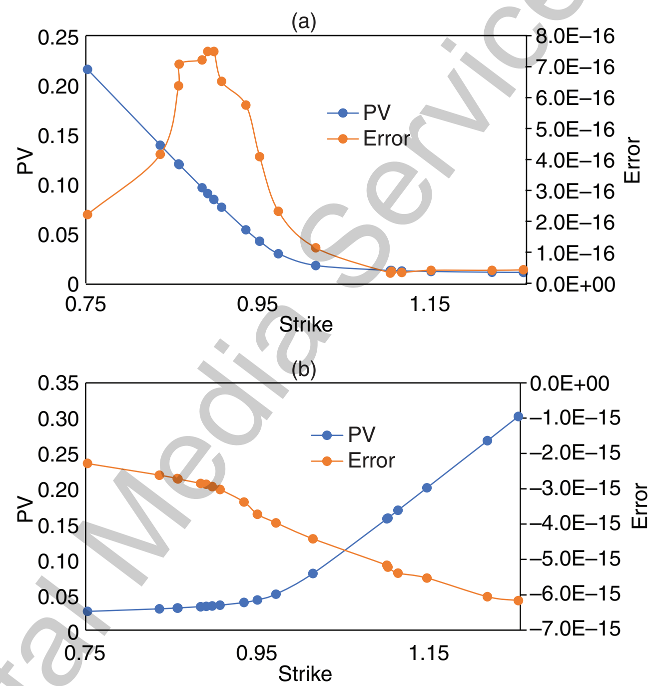
$$(U_i^\tau)^2 = \frac{(\sigma_i^\tau)^2}{\frac{1}{2}(\sigma_+^2 p_+^{\tau+1} + \sigma_-^2 p_-^{\tau+1})_i / p_i^{\tau+1}} \tag{45}$$

1 Numerical solution for call options generated with spot reference 1.0 and Black-Scholes volatility 10%



(a) Standard discretisation. (b) Our discretisation. The expiry time is one year. The dividend yield is 10% and the interest rate is 5%. The time grid has 100 points evenly spaced, and the spot grid has 100 points between 0.5 and 1.5 evenly spaced in log space

2 Our discretisation for options generated with spot reference 1.0, SABR initial volatility 10%, correlation −50% and vol-of-vol 200%



(a) Call option. (b) Put option. The expiry time is one year. The dividend yield is 10% and the interest rate is 5%. The time grid has 20 points evenly spaced, and the spot grid has 18 points randomly generated, plus the two boundaries

which is our discretisation of the usual forward induction formula:

$$U(S,t)^2 = \frac{\sigma_{\text{Dupire}}(S,t)^2}{E[\sigma_t^2 \mid S]} \tag{46}$$

that follows from Gyöngy's theorem.

In the LSV case, we are not quite so fortunate as in the pure Dupire case, because $p_{\pm}^{\tau+1}$ are unknown at time τ. The easiest approach is to solve numerically for U using the fixed point method. One can first step forward using the previous U, then calculate the new U, and then iterate. We have found that three iterations are ample.

For backward pricing, we take the adjoint of equations (41) and (42):

$$\text{PV} = (p^0)^\dagger \left[\frac{1}{2} \prod_\tau (1 - \Delta t (A_+^\tau)^\dagger)^{-1} + \frac{1}{2} \prod_\tau (1 - \Delta t (A_-^\tau)^\dagger)^{-1} \right] q^T \tag{47}$$

and by construction this reprices vanilla options on the grid exactly.

Full local stochastic volatility

To further illustrate our approach, let us consider a model with a full stochastic volatility factor this time:

$$\frac{\mathrm{d}S}{S} = \mu \,\mathrm{d}t + \sigma U(S,t)\,\mathrm{d}W \tag{48}$$

$$\mathrm{d}\sigma = \alpha(\sigma)\,\mathrm{d}t + \nu(\sigma)\,\mathrm{d}Z \tag{49}$$

$$\mathrm{d}W\,\mathrm{d}Z = \rho\,\mathrm{d}t \tag{50}$$

To model this on a grid, we introduce an extra grid dimension $\{\sigma_i\}$ for the stochastic volatility σ. Our grid terminal probability distributions are now matrices $(p^\tau)_{ij}$ with index i corresponding to the x_i (spot) dimension, and j corresponding to the y_j (volatility) dimension.

To discretise this, we define:

$$A_x = -\mu D_x X + \tfrac{1}{2} L_x V^\tau - r \tag{51}$$

where D_x and L_x are our usual discretisations of the $\partial/\partial x$ and $\partial^2/\partial x^2$ operators and act only on i indexes. $X = \operatorname{diag}(x_i)$ and, again, only acts on i indexes. The operator V^τ is defined by:

$$V^\tau = X^2 \Sigma^2 (U^\tau)^2 \tag{52}$$

where $\Sigma = \operatorname{diag}(\sigma_j)$ acts only on j indexes, and $U^\tau = \operatorname{diag}(U_i^\tau)$ is the discretised local volatility correction and acts only on i indexes.

Similarly, we discretise the $\partial/\partial\sigma$, $\partial^2/\partial\sigma^2$ and $\partial^2/\partial\sigma\partial x$ operators as D_σ, $L_{\sigma\sigma}$ and $L_{\sigma x}$, respectively, each acting only on j indexes, taking care that the boundary conditions on σ ensure:

$$\sum_j (D_\sigma p)_{ij} = \sum_j (L_{\sigma\sigma} p)_{ij} = \sum_j (L_{\sigma x} p)_{ij} = 0 \tag{53}$$

Then our (implicit) forward scheme becomes:

$$\frac{p^{\tau+1} - p^\tau}{\Delta t} = A_x p^{\tau+1} + B_\sigma p^{\tau+1} + C_{\sigma x} p^{\tau+1} \tag{54}$$

where:

$$B_\sigma = -D_\sigma \alpha + \tfrac{1}{2} L_{\sigma\sigma} \nu \quad (55)$$

$$C_{\sigma x} = L_{\sigma x} \Sigma U \nu \quad (56)$$

and α and ν are discretised as matrices (acting on j indexes).

Let us assume (inductively) that we have calculated the probabilities all the way up to p^τ. Then we can sum (54) over j, and the boundary conditions on σ ensure for us that the B and C terms sum to zero. We are left with what is, by now, a familiar formula:

$$\sum_j \frac{p^{\tau+1} - p^\tau}{\Delta t} = \sum_j A_x p^\tau \quad (57)$$

where $\sum_j p^\tau_{ij} = Lc^\tau$. Extracting out the local volatility corrections in the same way as before gives us:

$$(U_i^\tau)^2 = \frac{(\sigma_i^\tau)^2 p_i^{\tau+1}}{\sum_j \sigma_j^2 p_{ij}^{\tau+1}} \quad (58)$$

Here, σ_i^τ are our discretisation of Dupire's formula from (23), while σ_j are the stochastic volatility grid values. As before, since $p_{ij}^{\tau+1}$ are unknown until we complete the step, we can use the fixed point method with, say, three iterations to compute the grid local volatility corrections U_i^τ.

Conclusion

We have shown how to construct finite difference schemes that discretise common PDE pricing equations and recover vanilla option prices exactly for expiries and strikes on the grid points. In a sense, one can think of the method as unifying the continuous PDE approach of Dupire (1994) with the binomial and trinomial tree approaches of Derman & Kani (1994) and Rubinstein (1994). In addition to local volatility, local stochastic volatility and interest rate hybrid models, as in Piterbarg (2006), are in scope.

Our approach offers a number of significant advantages over the standard methods currently available. First, as vanilla prices are recovered exactly, the issue of convergence is resolved, and the grid density can be significantly reduced, with corresponding performance and stability gains. This also ensures that contracts with fine payout features resolve correctly, since one can simply ensure that all relevant strikes sit on the grid. Similarly, vanilla prices are exactly recovered when one is far from a barrier (a common bugbear of options traders).

Second, complex stepping schemes can be eliminated. As vanilla prices are recovered exactly, the issue of the order of time convergence is less relevant. It remains to be seen whether continuous barrier options and American options converge faster with the semi-implicit version of our scheme.

Third, as vanilla prices are always recovered, very coarse grids can be used for stochastic volatility and stochastic rates and dividends, allowing higher factor models to be feasible.

Fourth, as vanilla prices are exact, we can be sure that we will get good second-order Greeks, as long as we fix the grid when bumping. In particular, the gamma of vanilla options will be exactly equal to the bump and revalue gamma from the Black-Scholes formula. It is pleasing that this provides an explanation for the improvements seen by fixing the grid when using the traditional discretisation approach.

Finally, although fully implicit and Crank-Nicolson schemes are often described as unconditionally stable, in reality the analyses typically apply to the case of constant volatility. One has no way to be sure the scheme is stable with a particular set of market data encountered. Our approach makes it much easier to study stability, and it appears to be stable as long as the implied volatility surface does not contain arbitrage.

Having understood these points, we are now in a position to propose replacements for the industry best practice rules of thumb for finite difference pricing:

(1) Discretise Dupire's formula using (23).
(2) Ensure relevant vanilla strikes are placed on grid points.
(3) Place digital strikes midway between grid points (so they are priced as the spread of the two adjacent vanillas).
(4) Place more grid points in regions where the payout is non-linear.
(5) Use implicit stepping.
(6) Apply $(p^0)^\dagger$ to convert the solution vector at time 0 into the price; do not use an interpolator.
(7) Keep the grid fixed when bumping for risk; do not place the current spot on the grid. ■

Peter Austing is a quantitative strategist at Eisler Capital based in London. The views expressed in this article are the author's own and do not necessarily represent the views of his employer. Email: p.austing@runbox.com.

REFERENCES

Andreasen J and BN Huge, 2011a
Finite difference based calibration and simulation
Risk July, pages 62–67

Andreasen J and BN Huge, 2011b
Volatility interpolation
Risk March, pages 76–79

Derman E and I Kani, 1994
Riding on a smile
Risk February, pages 32–39

Dupire B, 1994
Pricing with a smile
Risk January, available at www.risk.net/derivatives/equity-derivatives/1500211/pricing-with-a-smile

Gatheral J, 2006
The Volatility Surface: A Practitioner's Guide
Wiley Finance

Huge B, 2018
A note on the Fokker-Planck equation
Working Paper, ResearchGate

Jamshidian F, 1991
Forward induction and construction of yield curve diffusion models
Journal of Fixed Income 1(1), pages 62–74

Jex M, 1999
Pricing exotics under the smile
Risk November, pages 72–75

Lipton A, 2002
The vol smile problem
Risk February, pages 61–65

Piterbarg V, 2006
Smiling hybrids
Risk May, pages 66–71

Rubinstein M, 1994
Implied binomial trees
Journal of Finance 49(3), pages 771–818

Wyns M and in 't Hout KJ, 2018
An adjoint method for the exact calibration of stochastic local volatility models
Journal of Computational Science 24, pages 182–194