Finite Difference Methods for Financial Problems

Part 3: How-to-do

Kwant Skool
Copenhagen
March 2011

Jesper Andreasen
Danske Markets
kwant.daddy@danskebank.com

Outline

- Transforms.
- Non-uniform spacing.
- Grid dimensioning.
- Barriers.
- Placing points on the grid.
- American options.
- Stability and convergence in practice.
- Conclusion.

Transforms

- The first question is what PDE to solve?
- This one

$$0=\frac{\partial V}{\partial t}+[-r+\mu S\frac{\partial}{\partial S}+\frac{1}{2}\sigma^2S^2\frac{\partial^2}{\partial S^2}]V \qquad (1)$$

- …or this one?

$$0=\frac{\partial G}{\partial t}+\frac{1}{2}\frac{\partial^2}{\partial w^2}G \quad ,G=e^{rt}V \quad ,S=e^{(\mu-1/2\sigma^2)t+\sigma w} \qquad (2)$$

- General answer: If parameters were constant and there were no other considerations then (2).
- The more you can make the PDE look the higher accuracy you can get.
- Exercise: Verify that the (1) and (2) are equivalent.

- Exercise: find and compare the (uniform) S, x grids to use for a 10y option under assumption of 20% volatility.

- However, often you want to include specific points on the grid that you're solving so you want control over s.

- Also we typically have time dependent parameters that prevent direct translation between prices and Brownian motions.

- Compromise:

$$0 = \frac{\partial V}{\partial t} + [-r + \mu \frac{\partial}{\partial x} + \frac{1}{2}\sigma^2(\frac{\partial^2}{\partial x^2} - \frac{\partial}{\partial x})]V \quad , S = e^x \tag{3}$$

- The split in (3) is kept this way because it allows us to naturally split edge boundary conditions and upwinding [I rarely use upwinding].

- We may for example set

$$0 = \frac{\partial V}{\partial t} + [-r + \mu(1_{\mu>0}\delta_x^+ + 1_{\mu<0}\delta_x^-) + \frac{1}{2}\sigma^2(\delta_{xx} - \delta_x)]V$$
$$(\delta_{xx} - \delta_x)V(t, x_1) = (\delta_{xx} - \delta_x)V(t, x_n) = 0 \qquad (4)$$

- Another example is the Ornstein-Uhlenbeck process

$$dx = -\kappa x dt + \sigma dW \tag{5}$$

- The solution is

$$e^{\kappa t} x(t) = x(0) + \int_0^t e^{\kappa u} \sigma dW(u) \tag{6}$$

- Which *could* be transformed to the PDE (Smith (1994) – so that he could get it into a binomial tree -- hooray):

$$0 = G_v + \frac{1}{2} G_{yy} \quad , v(t) = \int_0^t e^{2\kappa u} \sigma^2 du \quad , x(v) = e^{-\kappa t(v)} (x(0) + y) \tag{7}$$

- This gives rise to a exponential time spacing which can be done but does not really seem natural for the problem => deterioration of accuracy.

- Instead just directly solve

$$0 = \frac{\partial V}{\partial t} + [-\kappa x \frac{\partial}{\partial x} + \frac{1}{2}\sigma^2 \frac{\partial^2}{\partial x^2}]V \qquad (8)$$

- Exercise: Derive (7) and find $t(v)$ in (7).

- Exercise: What would be good boundary conditions for (8)?

- For the square root process:

$$dx = \kappa(\theta - x)dt + \sigma\sqrt{x}dW \tag{9}$$

- …it would be tempting to try to transform away the square-root term in the diffusion by using the transform $y = \sqrt{x}$:

$$dy = \frac{1}{2}[(\kappa\theta - \sigma^2/4)y^{-1} - \kappa y]dt + \frac{1}{2}\sigma dW \tag{10}$$

- But the resulting PDE has a nasty singularity at $x = y = 0$.

- A way of circumventing this is to use a grid spacing with $x_i = O(i^2)$.

Non-Uniform Grids

- Using Taylor expansion

$$\frac{1}{\Delta x_u}(f(x+\Delta x_u)-f(x))=f_x(x)+\Delta x_u\frac{1}{2}f_{xx}(x)+\frac{1}{6}\Delta x_u^2 f_{xxx}(x)+O(\Delta x_u^3)$$
$$\frac{1}{\Delta x_l}(f(x-\Delta x_l)-f(x))=-f_x(x)+\Delta x_l\frac{1}{2}f_{xx}(x)-\frac{1}{6}\Delta x_l^2 f_{xxx}(x)+O(\Delta x_l^3) \qquad (11)$$

- …lead to the following difference operators for non-uniform grids

$$f_x(x)=\frac{1}{\Delta x_l+\Delta x_u}[\frac{\Delta x_l}{\Delta x_u}(f(x+\Delta x_u)-f(x))+\frac{\Delta x_u}{\Delta x_l}(f(x)-f(x-\Delta x_l))]+O(\Delta x^2)$$
$$f_{xx}(x)=\frac{2}{\Delta x_l+\Delta x_u}[\frac{1}{\Delta x_u}(f(x+\Delta x_u)-f(x))-\frac{1}{\Delta x_l}(f(x)-f(x-\Delta x_l))]+\frac{(\Delta x_u^2-\Delta x_l^2)}{3(\Delta x_l+\Delta x_u)}f_{xxx}(x)+O(\Delta x^2)$$
(12)

- So using non-uniform grids lead to a fall in accuracy but this is minimised if the grids are only *slowly* changing in grid spacing, ie $\Delta x_u-\Delta x_l$ is small.

- Exercise: How can you ensure slowly changing grid spacing?
- Exercise: What not to do.

Grid Dimensioning

- Grids should always be dimensioned according to some measure of dispersion of the underlying process.

- The *general* rule is

$$\begin{aligned} x_1 &= x(0) - 5v \\ x_n &= x(0) + 5v \\ v &= \left(\int_0^T E[dx(u)^2] \right)^{1/2} \end{aligned} \qquad (13)$$

- Note that we use the quadratic variation and not the variance of the process for dimensioning the grid.

- The number 5 here refers to the fact that this is a good width for integration of the normal distribution because $\Phi(-5) \approx 2 \cdot 10^{-7}$.

- For Black-Scholes we have $v = \sigma\sqrt{T}$ but this is also the case for the Ornstein-Uhlenbeck.

- For the sqrt process example we would, most conveniently, use something like $v \approx \sigma\sqrt{\bar{\theta}T}$.
- The reason for using quadratic variation and not variance is that for low to medium mean-reversion and long trades, we have empirically found that the grid simply isn't wide enough if we just use variance.
- For stochastic volatility we generally use wider grids, around $x(0) \pm 10v$.
- The easiest way to find this is simply to experiment: keep Δx fixed and widen the grid until we get convergence.
- The number of spatial points should be roughly the same for the same model independent maturity and parameters.
- This is also the case for time points. Irrespective of maturity the number of time steps should be the same.
- The reason for this is that the PDE

$$0 = V_t + \frac{1}{2}\sigma^2 V_{xx} \qquad (14)$$

- …is homogeneous in all parameters.
- We should therefore expect to see the same *relative* accuracy, irrespective of maturity, volatility, etc.
- Often heard: “you need to use fully implicit method to prevent ringing when you’re close to expiry and strike.”
- The argument is wrong, you simply just need to use the same number of time steps no matter whether the option is 1d or 30y to maturity.
- For CN methods the number of time steps should be of the same order as the number of spatial steps.
- However, most often the error in the spatial domain is (much) larger than the error in the time domain.

- Generally, I recommend number of spatial points to be 2-4 times bigger than the number of time points – for CN grids, at least.

- Which squares well with the results what we had for the ringing in CN.

- Exercise: What not to do?

Barriers

- Continuously observed barriers need to be on the grid and combined with an absorption condition otherwise you will effectively be pricing a discretely observed barrier option.

- The difference between a barrier option that knocks out continuously or daily can be quite large.

- Absorption is enforced by setting

$$\begin{aligned} &\mu(x=B)=\sigma(x=B)=0 \\ &V(t_h, x=B)=R=rebate \end{aligned} \tag{15}$$

- …where t_h are our time points for our FD solution.

- The alternative, and what most often is described in textbooks, is to directly enforce $V=R$ at the barrier and explicitly include this at the edges of the tridiagonal solution.

- The two methods are equivalent, the first is just a bit easier to implement, particularly in conjunction with in-barriers and scripting languages.

- Discretely observed barriers and digital strikes should be placed in the middle of two grid points. In addition to that, no extra care is needed.

- It is of course convenient to place the initial spot on the grid but it may mess up the grid geometry if you are close to a barrier level. So if there are barriers use instead a low order polynomial, order 1 to 3, to interpolate for the initial value after the FD grid has been solved.

- For polynomial interpolation, NRC::polint() can be used.

- Exercise: Ideally, where should the strike of a standard call sit in the grid?

Placing Points on the Grid

- We want to place grid points so that the change in grid spacing is minimal.
- Spline idea from Domingo & Tavella.
- We want to put the points $y_1, y_2, \ldots, y_k$ on the grid on n points.
- First construct a spline $s(i)$ by setting the knot points

$$s(\text{int}((y_j - y_1)/(y_k - y_1)\cdot(n-1))) = y_j \quad , j = 1, \ldots, k \tag{13}$$

- Next, set the grid points according to

$$x_i = s(i-1) \quad , i = 1, \ldots, n \tag{14}$$

- If we want a point y_j to sit in the middle of two grid points, then replace (13) by

$$s(1/2+\text{int}((y_j - y_1)/(y_k - y_1)\cdot(n-1)) = y_j \qquad (15)$$

- …for that point and set the remaining points as in (14).
- Natural splines can be implemented using NRC::spline() and NRC::splint().

American Options

- Suppose we want to solve the American option pricing problem

$$
\begin{aligned}
&0 = \frac{\partial V}{\partial t} + [-r + \mu \frac{\partial}{\partial x} + \frac{1}{2}\sigma^2 (\frac{\partial^2}{\partial x^2} - \frac{\partial}{\partial x})]V \\
&V(t,x) \geq (e^x - K)^+ \quad , t \leq T \\
&V(T,x) = (e^x - K)^+
\end{aligned}
\tag{16}
$$

- The easiest way to handle this problem is to replace the inequality condition with

$$
V(t_h, x) = \max(V(t_h +, x), e^x - K) \tag{17}
$$

- This corresponds to pricing a Bermuda option with exercise on the FD dates and does not incorporate the continuous smooth pasting condition at the optimal barrier:

$$
\begin{aligned}
&V_x(t, x = x^*(t)) = e^x \\
&x^*(t) = \inf_x \{x : V(t,x) = e^x - K\}
\end{aligned}
\tag{18}
$$

- Formally, this implies that the solution is only $O(\Delta t)$ accurate even though CN is used.

- This has lead many authors to suggest the use of so-called PSOR [Piecewise Successive Over Relaxation] methods that iterate the solution to achieve (18) everywhere and by this ensure $O(\Delta t^2)$ accuracy.

- However, as the European option component in American options tends to be large, the practical benefit of this is very small and does not really outweigh the extra computational cost.

- Hansen & Østerby (1997) suggest an alternative where they over each time step iterate over the length of the time step to make sure that the optimal exercise point x^*, lies on the x grid. By doing so they obtain second order convergence in time: $O(\Delta t^2)$.

Convergence

- For a Crank-Nicolson grid, we have formal error descriptions along the line of

$$v(\Delta t,\Delta x)=v(0,0)+O(\Delta t^2+\Delta x^2)=v(0,0)+k_t\Delta t^2+k_x\Delta x^2 \tag{19}$$

- This can be used to deduce relations along the line of

$$\ln|v(\Delta t,\Delta x)-v(0,\Delta x)|=\ln k_t+2\ln\Delta t$$
$$\ln|v(\Delta t,\Delta x)-v(\Delta t,0)|=\ln k_x+2\ln\Delta x \tag{20}$$

- Note here that I attempt to separate the convergence in each dimension.

- So if we plot log-errors versus log-resolution we should see straight lines. The slope of these lines should be approximately 2.

- We can also regress (19) to find the coefficients but in general the log-log plot is more effective.

- Exercise: Use a log-log plot to justify the $O(\Delta t)$ and $O(\Delta t^2)$ convergence of the fully implicit and the Crank-Nicolson methods.

Conclusion

- The most important points in implementation of the FD methods:
    - What PDE to solve, i.e. what transform to apply?
    - How wide should the FD grid be?
    - How many t and x points to use?
- We have provided rules of thumb on what to do.
- Also we have provided tools for handling barriers and placing points on the grid.
- Questions?