Finite Difference Methods for Financial Problems

Part 1: The Theta Scheme

Kwant Skool
Copenhagen
March 2011

Jesper Andreasen
Danske Markets
kwant.daddy@danskebank.com

Outline

- Numerical integration versus Monte-Carlo simulation.
- The backward and forward PDEs.
- Time discretisation: the theta method.
- Solving tridiagonal matrix systems.
- Properties of the theta method.
- Finite difference approximation in the spatial domain.
- Boundary conditions.
- Discrete forward equations.
- Conclusion.

Numerical Integration

- Generally we are after solving integration problems

$$V = \int_{\mathbb{R}^n} \underbrace{g(x)}_{payoff} \cdot \underbrace{p(x)}_{density} dx = \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} g(x_1,\ldots,x_n) p(x_1,\ldots,x_n) dx_1 \ldots dx_n \tag{1}$$

- Two ways of doing this numerically.

- Monte-Carlo

$$\begin{aligned} &V_{MC} = M^{-1} \sum_{i=1}^{M} g(x_i) + \underbrace{O(M^{-1/2})}_{error} \quad ,x_i \sim p\,, x_i \; independent \\ &error \sim M^{-1/2} \equiv e_{MC} \\ &work \sim M = e_{MC}^{-2} \end{aligned} \tag{2}$$

- Numerical integration (Trapezoidal):

$$
\begin{aligned}
V_I &= \sum_{i=1}^{m_1} \cdots \sum_{i=1}^{m_n} g(i_1\Delta x_1,\ldots,i_n\Delta x_n)\, p(i_1\Delta x_1,\ldots,i_n\Delta x_n)\Delta x_1 \cdot \ldots \cdot \Delta x_n + \underbrace{O(\sum_i \Delta x_i^2)}_{error} \\
&\sim \sum_{i_1=1}^{m} \cdots \sum_{i_{1_n}=1}^{m} g(i_1\Delta x_1,\ldots,i_n\Delta x_n)\, p(i_1\Delta x_1,\ldots,i_n\Delta x_n)\Delta x_1 \cdot \ldots \cdot \Delta x_n + \underbrace{O(n/m^2)}_{error} \qquad (3)
\end{aligned}
$$

$$
error \sim n/m^2 \equiv e_{NI}
$$

$$
work \sim m^n = n^{n/2} e_{NI}^{-n/2}
$$

- We see that the required work is lower with NI than with MC for problems of dimension $n<4$, the two methods are equivalent for $n=4$, and MC wins for $n>4$.

- FD and NI are not exactly the same but approximately.

- The fixed cost of NI/FD is, however, generally higher, so we get the rule of thump

  - Finite difference for problems $n \le 3$.

  - Monte Carlo for problems $n \ge 4$.

- This is a general rule not an absolute.

- In practice MC is used for many problems with $n \le 3$, mainly for ease convenience of implementation.

- On the other hand, there are banks that solve $n = 4$ problems with FD.

- Sobol sequences (non-independent sampling) aim to achieve $O((\ln M)^n / M)$ asymptotic convergence. But this can't really be proven mathematically (or experimentally) and experience suggests that this behaviour only occurs very asymptotically, i.e. for very large M . If true the switch point between MC and FD would be $n \le 2$ for FD.

- Exercise 1: Find the error terms in (2) and (3).

- Exercise 2: Set $e = 0.01$ and compute the amount of required work in (2) and (3) for $n = 1, 2, 3, 4, 5$.

The Backward and Forward PDEs

- Suppose we wish to compute

$$V(t) = E_t\left[\frac{V(T)}{B(T)/B(t)}\right] \quad , B(t) = e^{\int_0^t r(u)du} \tag{4}$$

- …and suppose that we can limit the information of the conditional expectation to the value of a one-dimensional Markov process

$$\begin{aligned} V(t) &= E_t\left[\frac{V(T)}{B(T)/B(t)}\right] = E\left[\frac{V(T)}{B(T)/B(t)} \,\middle|\, x(t)\right] \\ dx(t) &= \mu(t, x(t))dt + \sigma(t, x(t))dW(t) \end{aligned} \tag{5}$$

- Then $V = V(t, x(t))$ and as V/B is a Martingale then

$$0 = E_t\left[d\frac{V}{B}\right] = \frac{1}{B}\left[-rV + V_t + \mu V_x + \frac{1}{2}\sigma^2 V_{xx}\right]dt \tag{6}$$

- From this we get the backward PDE

$$0=\frac{\partial V}{\partial t}+DV \quad ,D=[-r+\mu\frac{\partial}{\partial x}+\frac{1}{2}\sigma^2\frac{\partial^2}{\partial x^2}] \tag{7}$$

- We also have the forward PDE

$$0=-\frac{\partial q}{\partial t}+D^*q \quad ,D^*=[-r-\frac{\partial}{\partial x}\mu+\frac{1}{2}\frac{\partial^2}{\partial x^2}\sigma^2] \quad ,q(0,x)=\delta(x-x(0)) \tag{8}$$

- Which can be solved and used for calculating European option values

$$V(0,x(0))=\int V(t,x)q(t,x)dx \tag{9}$$

- We will mainly work on the numerical solution of the backward PDE.

- It will later turn out that finite difference solution of backward PDE give rise to *discrete* forward equations.

- This PDE (7) is valid on the domain for the stochastic process and subject to boundary conditions that define the claim.

- If we are considering a European call option on x with r,μ,σ constants then we have the domain $[0,T]\times\mathbb{R}$ with the boundary conditions

$$\begin{aligned} &V(T,x)=(x-k)^+ \\ &\lim_{x\to\pm\infty}[V(t,x)-(x-k)^+]=0 \quad ,0\le t\le T \end{aligned} \tag{10}$$

- Here, we have defined the spatial boundary conditions in terms of asymptotic *value*.

- The alternative is to specify in terms of derivatives. For example

$$\begin{aligned} &V_x(t,-\infty)=0 \quad ,V_x(t,+\infty)=1 \\ &or \\ &V_{xx}(t,-\infty)=V_{xx}(t,+\infty)=0 \end{aligned} \tag{11}$$

- Exercise: Go through the steps (4-7) for an up-and-out call option in the Black-Scholes model. Identify $V(T)$, the domain and the boundary conditions.

- My job here is to convince you that D can be approximated by a matrix *and* that the value is a vector

$$D = \begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \dots & 0 & a_n & b_n \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$$v(t) = \begin{bmatrix} v(t, x_1) \\ \vdots \\ v(t, x_n) \end{bmatrix} \in \mathbb{R}^n \quad , x_i = x_1 + (i-1)\Delta x \quad , i = 2, \dots, n \tag{12}$$

- So (7) is now a matrix equation

$$0 = \frac{\partial v}{\partial t} + \bar{D} v \tag{13}$$

The Theta Method

- Discretising the time derivative and the matrix vector product by

$$\frac{\partial v}{\partial t} \approx \frac{1}{\Delta t}[v(t_{h+1}) - v(t_h)] \quad , t_h = h \cdot \Delta t$$
$$\bar{D}v \approx \bar{D}[\theta v(t_h) + (1-\theta) v(t_{h+1})] \quad , \theta \in [0,1] \qquad (14)$$

- ...yields the matrix equation

$$[I - \theta \Delta t \bar{D}] v(t_h) = [I + (1-\theta) \Delta t \bar{D}] v(t_{h+1}) \qquad (15)$$

- …where $I \in \mathbb{R}^{n \times n}$ is the identity matrix.

- The matrix equation (15) has to be solved for the vector $v(t_h)$ given the vector of values $v(t_{h+1})$.

- …step by step backwards from the terminal boundary condition until we have the solution for all discrete points (t_h, x_i) and particularly the values at time 0.

- On the way backwards additional boundary conditions may apply.

Tridiagonal Matrix Solution

- Solving tri-diagonal systems, NRC::tridag():

```
void
kMatrixAlgebra::tridag(
      int rows, const kValArray<double>& a, const kValArray<double>& b, const kValArray<double>& c,
      const kValArray<double>& r,kValArray<double>& x,kValArray<double>& ws)
{
      int j;
      double bet;

      bet  = 1.0/b(0);
      x(0) = r(0)*bet;

      //    decomposition and forward substitution
      for(j=1;j<rows;++j)
      {
            ws(j) = c(j-1)*bet;
            bet   = 1.0/(b(j) - a(j)*ws(j));
            x(j)  = (r(j) - a(j)*x(j-1))*bet;
      }
      //    backsubstitution
      for(j=rows-2;j>=0;--j)
      {
            x(j)  -= ws(j+1)*x(j+1);
      }
}
```

- Very easy to code and work load is $O(n)$.

Properties of the Theta Method

- The explicit method, $\theta = 0$:

$$v(t_h) = [I + \Delta t \bar{D}] v(t_{h+1}) \tag{14}$$

corresponds to a trinomial tree and has accuracy $O(\Delta t + \Delta x^2)$ -- but is only stable if Δt is chosen sufficiently small to guarantee that $I + \Delta t D \geq 0$, i.e. all elements are non-negative.

- The fully implicit method, $\theta = 1$:

$$[I - \Delta t \bar{D}] v(t_h) = v(t_{h+1}) \tag{15}$$

has accuracy $O(\Delta t + \Delta x^2)$ and is super unconditionally stable for any choice of $(\Delta t, \Delta x)$.

- The Crank-Nicolson method, $\theta = 1/2$:

$$[I - \frac{1}{2}\Delta t \bar{D}]v(t_h) = [I + \frac{1}{2}\Delta t \bar{D}]v(t_{h+1}) \tag{16}$$

  has accuracy $O(\Delta t^2 + \Delta x^2)$ and is unconditionally stable for any $(\Delta t, \Delta x)$ but not super stable as the fully implicit method.

- Note that the CN can be seen as one half step of explicit solution followed by one half-step of implicit solution.

- Both the fully implicit and the CN converge to the correct solution.

- However, the trouble is that the CN has a tendency of ringing around discontinuous boundary conditions. But the ringing dies out as you move away in time and space from the discontinuity.

- The ringing of the CN method has created a lot of controversy and debate.

- The reality is that the ringing of the CN is not a practical problem if you use the CN method correctly, i.e, you use sufficiently wide grids and a sufficient number of time steps.

- The $O(\Delta t^2)$ accuracy is a real benefit particularly for interest rate models. 100 versus 10,000 time steps to achieve the same accuracy.

- So Crank-Nicolson will (nearly) always be the weapon of choice for the pro.

Finite Differences in the Spatial Domain

- We have

$$D = [-r + \mu \frac{\partial}{\partial x} + \frac{1}{2}\sigma^2 \frac{\partial^2}{\partial x^2}] \tag{17}$$

- To find $\bar{D}$ we make the approximations

$$\begin{aligned} D &\approx -r + \mu\delta_x + \frac{1}{2}\sigma^2\delta_{xx} \\ \delta_x f(x) &= \frac{1}{2\Delta x}[-f(x-\Delta x) + f(x+\Delta x)] \\ \delta_{xx} f(x) &= \frac{1}{\Delta x^2}[f(x-\Delta x) - 2f(x) + f(x+\Delta x)] \end{aligned} \tag{18}$$

- Then we have

$$\bar{D} = -I(r) + I(\mu)\Delta_x + \frac{1}{2}I(\sigma^2)\Delta_{xx} \tag{19}$$

…where

$$I(k)=\begin{bmatrix} k_1 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & k_{n-1} & 0 \\ 0 & \dots & 0 & 0 & k_n \end{bmatrix} \in \mathbb{R}^{n\times n} \quad ,\Delta_x = \frac{1}{2\Delta x}\begin{bmatrix} -2 & 2 & 0 & \dots & 0 \\ -1 & 0 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 0 & 1 \\ 0 & \dots & 0 & -2 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n}, \Delta_{xx} = \frac{1}{\Delta x^2}\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}$$

(20)

- These matrices are all tridiagonal. So $\bar{D}$ is tridiagonal.
- Specifically, the total tridiagonal matrix is

$$\bar{D} = \begin{bmatrix} -r_0 - \frac{\mu_0}{\Delta x} & \frac{\mu_0}{\Delta x} & & & & & \\ -\frac{\mu_1}{2\Delta x} + \frac{\sigma_1^2}{2\Delta x^2} & -r_1 - \frac{\sigma_1^2}{\Delta x^2} & \frac{\mu_1}{2\Delta x} + \frac{\sigma_1^2}{2\Delta x^2} & & & & \\ & -\frac{\mu_2}{2\Delta x} + \frac{\sigma_2^2}{2\Delta x^2} & -r_2 - \frac{\sigma_2^2}{\Delta x^2} & \frac{\mu_2}{2\Delta x} + \frac{\sigma_2^2}{2\Delta x^2} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -\frac{\mu_{n-2}}{2\Delta x} + \frac{\sigma_{n-2}^2}{2\Delta x^2} & -r_{n-2} - \frac{\sigma_{n-2}^2}{\Delta x^2} & \frac{\mu_{n-2}}{2\Delta x} + \frac{\sigma_{n-2}^2}{2\Delta x^2} & \\ & & & & -\frac{\mu_{n-1}}{2\Delta x} + \frac{\sigma_{n-1}^2}{2\Delta x^2} & -r_{n-1} - \frac{\sigma_{n-1}^2}{\Delta x^2} & \frac{\mu_{n-1}}{2\Delta x} + \frac{\sigma_{n-1}^2}{2\Delta x^2} \\ & & & & & -\frac{\mu_n}{\Delta x} & -r_n + \frac{\mu_n}{\Delta x} \end{bmatrix} \qquad (21)$$

- Exercise: What boundary conditions at the edges have been used to obtain (21)?

Boundary Conditions

- In the above we used the second derivative boundary condition:

$$V_{xx}(x_1) = V_{xx}(x_n) = 0 \tag{22}$$

- …combined with up/down differencing for the first derivative.
- This corresponds to the volatility being zero at the edges (absorption) and/or the pay-off being asymptotically linear.
- Occasionally we use at the grid edges:
  - Absorption: $V_x = V_{xx} = 0$.
  - Reflection: $V_{xx} = 0 \quad , \mu(x = x_1) > 0 \quad , \mu(x = x_n) < 0$
  - Log-linearity: $V_{xx} - V_x = 0$.

- For problems where the domain is *open*, the exact specification of the boundary conditions at the edges of the grid should matter very little.

- If not, it is a sign that your grid is not wide enough.

- Exercise: "All pay-offs in finance are asymptotically linear." Do you agree?

- Exercise: Find the accuracy of the approximations in (18).

- Exercise: Give examples of problems with open and closed domains.

- Exercise: How does one determine whether a problem has an open or closed domain?

Forward Equations – NEW!

- We can derive discrete forward equations for the theta scheme:

$$\begin{aligned} v(t_{h+1/2}) &= [I + (1-\theta)\Delta t\bar{D}]v(t_{h+1}) \\ [I - \theta\Delta t\bar{D}]v(t_h) &= v(t_{h+1/2}) \end{aligned} \qquad (23)$$

- Here we have written the scheme in split form: first we do an explicit step over half the time step, then we do an implicit step over the second half step.

- We have

$$v(t_h) = [I - \theta\Delta t\bar{D}]^{-1}[I + (1-\theta)\Delta t\bar{D}]v(t_{h+1}) \qquad (24)$$

- Hence, if we multiply (24) by a function (vector) $p(t_h)$, we get

$$p(t_h)'v(t_h) = p(t_h)'[I - \theta\Delta t\bar{D}]^{-1}[I + (1-\theta)\Delta t\bar{D}]v(t_{h+1}) \qquad (25)$$

- From this we conclude that the function that solves

$$\begin{aligned} p(t_h) &= 1_{i=j} \\ [I-\theta\Delta t\bar{D}]'\,p(t_{h+1/2}) &= p(t_h) \\ p(t_{h+1}) &= [I+(1-\theta)\Delta t\bar{D}]'\,p(t_{h+1/2}) \end{aligned} \tag{26}$$

- …is the fundamental solution (Green's function) for our theta scheme (23), in the sense that

$$v(t_h, x_j) = p(t_{h+1})'v(t_{h+1}) \tag{27}$$

- The forward equation (26) runs forward from t_h to t_{h+1}.

- The backward equation (23) runs backward from t_{h+1} to t_h.

- The two equations are completely consistent.

- Note that in the backward scheme we first go explicit then implicit.

- For the forward scheme we the order is reversed and we go implicit and then explicit in the next half time step.

- For $q = p/\Delta x$, the limit, $\Delta x \to 0$, of (26) is the usual continuous forward equation

$$\begin{aligned} &0 = -\frac{\partial q}{\partial t} + D^* q \quad , q(0,x) = \delta(x - x(0)) \\ &D^* q = -rq - \frac{\partial}{\partial x}[\mu q] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[\sigma^2 q] \end{aligned} \tag{28}$$

- Viola! A new derivation of the continuous forward equation.

- Note that direct finite difference solution of (28) would *not* give the same scheme as (26).

- For some processes, sqrt() for example, the boundary conditions at the edges of the grid for direct discretisation of (26) are not trivial.

- Our forward scheme (26) comes directly from the backward scheme (22) and it is therefore not necessary to apply special boundary conditions at the edges of the grid.

- The standard absorption/reflection boundary conditions from the backward scheme can be used.

- In the following we will use this for creating all sorts of examples for visualising accuracy and stability.

- Exercise: Prove (28).

Conclusion

- FD will always be more efficient than MC for low dimensional problems.
- The principles behind FD are very simple.
- Coding FD is easy. In fact, a lot easier than trees.
- FD methods provides a very simple and consistent way of getting to forward equations.
- Backward schemes have associated a dual fully consistent forward scheme – that is as straightforward to solve as the backward scheme.
- Questions?