

Fun with Finite Difference

WBS
London
May 2023

Jesper Andreasen
Saxo Bank, Copenhagen
kwant.daddy@saxobank.com

Outline

- Continuous time and state: expectation and backward equation.
- Fokker-Planck and Dupire forward equations.
- Discrete time and state: Operators as matrices.
- Backward finite difference equation: the Theta scheme.
- Discrete duality: Fokker-Planck and Dupire forward finite difference equations.
- Accuracy.

- Stability.
- The finite difference cheat sheet.
- Coding the Theta solver.

References

- Andreassen, J (2011): “Finite Difference Methods for Financial Problems.” <https://github.com/brnohu/CompFin>.
- Andreassen, J (2020): “Local Volatility in Multi Dimensions.” *QuantMinds*.
- Andreassen, J (2023): “Catch Up.” *Wilmott*.
- Andreassen, J (2023): “Fun with Finite Difference.” *Wilmott*.
- Andreassen, J and B Huge (2011a): “Volatility Interpolation.” *Risk*.
- Andreassen, J and B Huge (2011b): “Random Grids.” *Risk*.

- Dupire, B (1994): “Pricing with a Smile.” *Risk*.

Continuous Time

- Assume that we have some Markov state variable that follows an SDE of the form

$$dx = \mu(t, x)dt + \sigma(t, x)dW \quad (1)$$

- ... and we wish to compute

$$f(t, x(t)) = E_t[e^{-\int_t^T r(u, x(u))du} f(T, x(T))] \quad (2)$$

- Then by Ito's lemmas f solves the backward PDE

$$0 = \partial_t f + Af \quad , A = -r + \mu \partial_x + \frac{1}{2} \sigma^2 \partial_{xx} \quad (3)$$

- Of particular interest is the pricing of derivatives in which case you can think of f as the value of some derivative and this case the expectation is taken under the risk-neutral measure.
- In this course we are considering how to numerically solve the expectation problem (2) by crunching the PDE (3) on a computer by use of finite difference methods.
- Before we hit the coding we'll spend a bit of time on the equations.
- Specifically, where the finite difference methods come from.

Forward Equations

- Let

$$p(t,x) = E[e^{-\int_0^t r(u)du} \delta(x(t)-x)]$$

$$c(t,x) = E[e^{-\int_0^t r(u)du} (x(t)-x)^+]$$

- ... be the initial Arrow-Debreu and European option prices respectively so that

$$f(0, x(0)) = \int_{-\infty}^{\infty} f(t, x) p(t, x) dx$$

$$p(t, x) = \partial_{xx} c(t, x)$$

- The Arrow-Debreu and European option prices satisfy the forward PDEs

$$0 = -\partial_t p + A^* p, \quad A^* = -r - \partial_x \mu + \frac{1}{2} \partial_{xx} \sigma^2$$

$$0 = -\partial_t c + A c, \quad \mu = 0$$

- The forward equations run forward in expiry and strike, rather than backwards in time and spot. Draw.
- We will show that these forward equations also exist in finite difference land.

- In fact, they are much easier to derive there.

Finite Difference Solution

- Consider the backward PDE

$$0 = f_t + Af \quad , A = -r + \mu \partial_x + \frac{1}{2} \sigma^2 \partial_{xx} \quad (1)$$

- Suppose we work on a discrete grid of states $x_0 < \dots < x_{n-1}$.
- Introduce the discrete difference operators

$$\begin{aligned} \delta_x^+ f(x_i) &= \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad , 0 \leq i < n-1 \\ \delta_x^- f(x_i) &= \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \quad , 0 < i \leq n-1 \end{aligned} \quad (2)$$

- ... and their weighted average and differences

$$\delta_x g(x_i) = (1-\lambda)\delta_x^- g(x_i) + \lambda\delta_x^+ g(x_i) \quad , \lambda = \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}} \quad (3)$$

$$\delta_{xx} g(x_i) = 2 \frac{(\delta_x^+ - \delta_x^-)g(x_i)}{x_{i+1} - x_{i-1}} \quad , 0 < i < n-1$$

- We can now discretize the PDE (1) by central differences for the drift

$$\bar{A} = -r + \mu\delta_x + \frac{1}{2}\sigma^2\delta_{xx} \quad (4)$$

- Or we can use dynamic up- and down-winding for the drift:

$$\bar{A} = -r + \mu^+\delta_x^+ - \mu^-\delta_x^- + \frac{1}{2}\sigma^2\delta_{xx} \quad (5)$$

- In this case, dependent on the sign of the drift we'll go either upwards or downwards.
- If we set absorption on the boundaries of the grid, i.e. $\sigma = \mu = 0$ for $i=0, n-1$, then the PDE can be approximated as a *tridiagonal* matrix ODE

$$0 = f_t + \bar{A}f \tag{6}$$

- Tridiagonal means that the matrix \bar{A} only has non-zero values on the diagonal and just around the diagonal

$$\bar{A}_{ij} = 0 \quad ; i=0, n-1 \quad ; j < i-1, i+1 < j \tag{7}$$

>>> Drawing: tridag matrix <<<

Theta Scheme

- The solution to the matrix ODE (6) is

$$f(t_h) = e^{\Delta t \bar{A}} f(t_{h+1}) = \left(\sum_{k=0}^{\infty} \frac{(\Delta t \bar{A})^k}{k!} \right) f(t_{h+1}) \quad , \Delta t = t_{h+1} - t_h \quad (8)$$

- We can discretize (8) in time using the Theta scheme

$$\begin{aligned} f(t_{h+1/2}) &= [I + (1-\theta)\Delta t \bar{A}] f(t_{h+1}) \\ [I - \theta\Delta t \bar{A}] f(t_h) &= f(t_{h+1/2}) \end{aligned} \quad (9)$$

>>> Drawing: theta scheme <<<

- Different value of theta
 - $\theta=0$: Explicit
 - $\theta=1$: Implicit
 - $\theta=1/2$: Crank-Nicolson
- We have now turned the PDE problem into a recursive matrix-vector system.

Arrow Debreu Prices

- The matrix

$$H = [I - \theta \Delta t \bar{A}]^{-1} [I + (1 - \theta) \Delta t \bar{A}]$$

- ... is a matrix of “discounted transition probabilities” or Arrow-Debreu prices in the sense that H_{ij} is the (t_h, x_i) value of *one* paid if we hit $x(t_{h+1}) = x_j$ in the next time step:

$$H_{ij} = E[e^{-\int_{t_h}^{t_{h+1}} r(u) du} 1_{x(t_{h+1})=x_j} | x(t_h) = x_i] \stackrel{r=0}{=} \Pr[x(t_{h+1}) = x_j | x(t_h) = x_i]$$

- These need not be non-negative.

- But if they are, the discrete finite difference solution will produce arbitrage free option process.

Forward Finite Difference

- Equation (9) is solved backwards in discrete time steps:

$$t_h \leftarrow t_{h+1/2} \leftarrow t_{h+1}$$

- Specifically, we have the sequence of backwards matrix-vector multiplications

$$f(t_0, x(t_0)) = p(t_0)' \{H(t_0) \cdot \dots \cdot H(t_m)\} f(t_m)$$

$$bwd: \leftarrow \leftarrow \dots \leftarrow \leftarrow \tag{10}$$

$$fwd: \Rightarrow \Rightarrow \dots \Rightarrow \Rightarrow$$

- ... where $p(t_0, x_i) = 1_{x_i = x(t_0)}$ is a vector of zero's except in one place – at the initial spot.
- We can run these operations in reverse, i.e. from left to right and thereby forward in time.
- Specifically,

$$\begin{aligned} [I - \theta \Delta t \bar{A}]' p(t_{h+1/2}) &= p(t_h) \\ p(t_{h+1}) &= [I + (1 - \theta) \Delta t \bar{A}]' p(t_{h+1/2}) \end{aligned} \tag{11}$$

- We then have

$$f(t_0, x(t_0)) = \sum_{i=0}^{n-1} p(t_m, x_i) f(t_m, x_i) = p(t_m)' f(t_m) \quad (12)$$

- We thus have that (11) is the *discrete Fokker-Planck* equation for the *backward* system (9).

Dupire Equation

- For European call options we have

$$c(t_h, k) = \sum_{i=0}^{n-1} p(t_h, x_i) (x_i - k)^+ \quad (13)$$

- For the case of $\mu=0$, it can be shown that the European option prices solve

$$\begin{aligned} [I - \theta \Delta t \bar{A}] c(t_{h+1/2}) &= c(t_h) \\ c(t_{h+1}) &= [I + (1 - \theta) \Delta t \bar{A}] c(t_{h+1/2}) \end{aligned} \quad (14)$$

- ... with initial boundary condition $c(t_0, k) = (x(t_0) - k)^+$.
- This is the discrete Dupire equation.

Finite Difference Duality

- Backward scheme

$$\begin{aligned} f(t_{h+1/2}) &= [I + (1-\theta)\Delta t \bar{A}] f(t_{h+1}) \\ [I - \theta\Delta t \bar{A}] f(t_h) &= f(t_{h+1/2}) \end{aligned} \tag{15}$$

- Forward scheme

$$\begin{aligned} [I - \theta\Delta t \bar{A}]' p(t_{h+1/2}) &= p(t_h) \\ p(t_{h+1}) &= [I + (1-\theta)\Delta t \bar{A}]' p(t_{h+1/2}) \end{aligned} \tag{16}$$

- Dupire equation ($\mu=0$)

$$\begin{aligned}
[I - \theta \Delta t \bar{A}]c(t_{h+1/2}) &= c(t_h) \\
c(t_{h+1}) &= [I + (1 - \theta) \Delta t \bar{A}]c(t_{h+1/2})
\end{aligned}
\tag{17}$$

- ... are all mutually *discretely* consistent.
- In the sense that the prices they produce will be the same no matter which method you use.
- You can for example calibrate to European options using the forward equation and reprice perfectly using the backward equation.

Accuracy

- By Taylor expansion of $f(t_h)=e^{\Delta t A} f(t_{h+1})$ we get the accuracies:
- Explicit $\theta=0$: $O(\Delta t + \Delta x^2)$
- Implicit $\theta=1$: $O(\Delta t + \Delta x^2)$
- Crank-Nicolson $\theta=1/2$: $O(\Delta t^2 + \Delta x^2)$
- Dropping to $O(\Delta x)$ if up- and down-winding is used or if the grid spacing is strongly varying.

Stability: von Neumann

- Previous accuracy considerations are only local.
- For convergence we additionally need stability in the sense that $\|f\|$ is bounded as $\Delta t, \Delta x \rightarrow 0$.
- Von Neumann analysis: for constant grid and parameters look for eigen solutions of the form

$$e(t_h, x_i) = g^{-h} e^{ikx_i} \quad , l = \sqrt{-1}, k \in \mathbb{R}, g \in \mathbb{C}$$

- If $|g| \leq 1 + O(\Delta t)$ for all k then we say that the scheme is von Neumann stable.

- For the constant parameter case von Neumann stability is sufficient to conclude that the scheme converges to the right result as $\Delta t, \Delta x \rightarrow 0$.
- For the general case, with general parameters and spacing it isn't proven to be sufficient for convergence.
- There are, however, to my knowledge, no counter examples.
- For the Theta scheme we have for constant parameters and grid spacing

$$g = \frac{[I + (1 - \theta)\Delta t \bar{D}](e^{ikx})}{[I - \theta\Delta t \bar{D}](e^{ikx})} = \frac{1 - (1 - \theta)\Delta t[r - \iota(\mu / \Delta x)\sin(k\Delta x) + 2(\sigma / \Delta x)^2 \sin(k\Delta x / 2)^2]}{1 + \theta\Delta t[r - \iota(\mu / \Delta x)\sin(k\Delta x) + 2(\sigma / \Delta x)^2 \sin(k\Delta x / 2)^2]}$$

Stability: Positivity

- Another related property is whether the transition probabilities, i.e. the elements of H are non-negative.
- I.e. whether all discrete option prices are consistent with absence of arbitrage.
- Explicit scheme, $\theta=0$, is vN stable iff $H \geq 0$, i.e. $\Delta t \leq O(\Delta x^2)$.
- Implicit scheme, $\theta=1$, is always vN stable but $H \geq 0$ only if up- and down-winding is used.
- CN scheme, $\theta=1/2$, is always vN stable but $H \geq 0$ only if up-and-down winding is used and the explicit condition $\Delta t \leq O(\Delta x^2)$ is met.

- CN scheme exhibits a (vaguely) oscillatory convergence pattern.
- However, CN converges way quicker $O(\Delta t^2)$ than the other methods and this is the method of choice if model parameters are given and we need to compute option prices as accurate as possible.
- However, for local volatility this is not really the task.
- Here, matching observed prices or producing arbitrage free option prices is the objective.
- In this case, the implicit scheme is the winner because it ensures $H \geq 0$.

>>> Live Example <<<

The Finite Difference Cheat Sheet

Schemes	Explicit $\theta = 0$	Implicit $\theta = 1$	Crank-Nicolson $\theta = 1/2$
PDE	$0 = f_t + Af \quad , A = -r + \mu \partial_x + \frac{1}{2} \sigma^2 \partial_{xx}$		
FD central	$\bar{A} = -r + \mu \delta_x + \frac{1}{2} \sigma^2 \delta_{xx}$		
FD winding	$\bar{A} = -r + \mu^+ \delta_x^+ + \mu^- \delta_x^- + \frac{1}{2} \sigma^2 \delta_{xx}$		
Boundaries	Absorption: $\mu = \sigma = 0$ or reflection: $\mu_0 > 0, \mu_{n-1} < 0, \sigma = 0$		
Backward	$f(t_h) = [I - \theta \Delta t \bar{A}]^{-1} [I + (1 - \theta) \Delta t \bar{A}] f(t_{h+1})$		
Forward	$p(t_{h+1}) = [I + (1 - \theta) \Delta t \bar{A}] [I - \theta \Delta t \bar{A}]^{-1} p(t_h)$		
Dupire	$c(t_{h+1}) = [I + (1 - \theta) \Delta t \bar{A}] [I - \theta \Delta t \bar{A}]^{-1} c(t_h) \quad , \mu = 0$		
Grid width	$\pm 5 \cdot (\int_0^T \sigma(u, x(0))^2 du)^{1/2}$		
Transform	PDE or grid spacing: $y = \int_{x_0}^x \sigma(a)^{-1} da$		
Vanilla strikes	Mid between grid points or $O(\Delta x^2)$ error		
Digitals	Mid between grid points or $O(\Delta x)$ error		
Cont barriers	On grid and absorption or $O(\Delta t^{1/2})$ error		
Von Neumann	$\Delta t \leq O(\Delta x^2)$	Always	Always
$p \geq 0, \mu = 0$	$\Delta t \leq O(\Delta x^2)$	Always	$\Delta t \leq O(\Delta x^2)$
$p \geq 0, \mu \neq 0$	$\Delta t \leq O(\Delta x^2)$	With winding	With winding and $\Delta t \leq O(\Delta x^2)$

Accuracy central	$O(\Delta t + \Delta x^2)$	$O(\Delta t + \Delta x^2)$	$O(\Delta t^2 + \Delta x^2)$
Accuracy winding	$O(\Delta t + \Delta x)$	$O(\Delta t + \Delta x)$	$O(\Delta t^2 + \Delta x)$
Time/spatial steps	4	0.5	0.5
Models	Brownian motion	Non-parametric	Parametric
CPU/time step (n=200)	3e-6s	4e-6s	7e-6s

Coding the Theta Solver

- The theta solver consists of the components:
 - `kFiniteDifference::dx()` and `::dxx()` construct the difference operators in compact matrix form.
 - `kFd1d::calcAx()` constructs the matrix $I + \Delta t \bar{A}$ in compact matrix form.
 - `kMatrixAlgebra::banmul()` and `::tridag()` do band diagonal matrix multiplication and tridiagonal matrix-vector solution.
 - `kFd1d::rollBwd()` implements the backward roll by calling the above routines.
 - `kMatrixAlgebra::transpose()` transposes a band-diagonal matrix.

- `kFd1d::rollFwd()` implements the forward roll.
- All in all in the hood of a few hundred lines of code.
- This is what we are going to implement in this course.