Computational Finance PDE

Copenhagen University
May 2024

Frederik Kryger-Baggesen
Saxo Bank, Copenhagen
frek@saxobank.com

Outline

- PDEs in finance.
- Backward solution.
- Forward solution.
- Let’s get practical.

Material

- Andreasen, J (2011): “Finite Difference Methods for Financial Problems.” *PhD Course Copenhagen University*.

- Andreasen, J (2022): “Catch-Up.” Forthcoming Wilmott.

- Andreasen, J, B Huge and F Kryger-Baggesen (2022): *https://github.com/brnohu*.

PDE's in Finance

- Let

$$ds = \mu(t,s)dt + \sigma(t,s)dW \tag{1}$$

$$db/b = r(t,s)dt$$

- We are interested in computing

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

- … which by Ito's lemma and Feynman-Kac is the solution to the *backward* PDE

$$0 = f_t + Af \quad ,A = -r + \mu\,\partial_s + \frac{1}{2}\sigma^2\,\partial_{ss} \tag{3}$$

- We are particularly interested in the pricing of derivatives in which case we can think of *f* as the value of some derivative and the expectation taken under the risk-neutral measure

- When *f* solves (3), the Fokker-Planck equation tells us

$$f(0, s(0)) = \int f(T, s) p(T, s) ds \tag{4}$$

- … where p are the discounted transition probabilities and solves the *forward* PDE

$$0 = -p_t + A^* p \ , A^* = -r - \mu\, \partial_s + \frac{1}{2} \partial_{ss} \sigma^2 \ , p(0, s) = \delta(s - s(0)) \tag{5}$$

- A^* is denoted the *adjoint* operator.

- The backward PDE is solved backwards in time: from terminal value to current value.

- The forward PDE is solved forwards in time: from current density of current state to density of future state.

- The PDE's are continuously consistent – but we want consistent finite difference schemes!

- One can possibly somewhat understand this intellectually but the way to really grasp it is to actually do it!

Backward solution

- We discretize in space at first => differential operators become difference operators

- Let $\bar{A}$ be the discretized version of A, i.e.

$$\bar{A} = -r + \mu\delta_s + \frac{1}{2}\sigma^2\delta_{ss} \qquad (6)$$

- … where δ is our finite difference operator, i.e. the discrete approx. to the differential operator

- Leads to the *tridiagonal* matrix ODE

$$0 = f_t + \bar{A}f \qquad (7)$$

- … which has the solution

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

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

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

- where

$$f(t_h) = (f(t_h, s_0), \ldots, f(t_h, s_{n-1}))' \quad (10)$$

- … is a vector of solution values.

- We solve the system (9) backward in time:

$$f(t_h) \underbrace{\leftarrow}_{\substack{tridiagonal\ matrix \\ inversion}} f(t_{h+1/2}) \underbrace{\leftarrow}_{\substack{tridiagonal\ matrix \\ multiplication}} f(t_{h+1})$$

Forward Solution

- Looking at the matrix

$$H = [I - \theta \Delta t \bar{A}]^{-1}[I + (1-\theta)\Delta t \bar{A}] \qquad (11)$$

- … we observe that the Theta scheme (9) is a series of vector-matrix multiplications.

- By letting $p(t_0, s_i) = 1_{s_{i=s(t_0)}}$, i.e. a vector of zeros expect at initial spot, we can express the time $t_0$value of f as

$$f\big(t_0, s(t_0)\big) = p(t_0)'\{H(t_0) \cdot \ldots \cdot H(t_m)\} f(t_m) \qquad (12)$$

- We can simply reverse the order by transposing (12)

$$f\left(t_0, s(t_0)\right) = f(t_m)'\{H(t_m)' \cdot \ldots \cdot H(t_0)'\}p(t_0) \qquad (13)$$

- … which leads to our forward scheme

$$[I - \theta\Delta t\bar{A}]'p\left(t_{h+\frac{1}{2}}\right) = p(t_h) \qquad (14)$$

$$p(t_{h+1}) = [I + (1-\theta)\Delta t\bar{A}]'p\left(t_{h+\frac{1}{2}}\right)$$

- We solve (14) forward in time

$$p(t_h) \underbrace{\rightarrow}_{\substack{tridiagonal\ matrix\\ inversion}} p(t_{h+1/2}) \underbrace{\rightarrow}_{\substack{tridiagonal\ matrix\\ multiplication}} p(t_{h+1})$$

Overview

- Backward scheme

$$f\left(t_{h+\frac{1}{2}}\right) = [I + (1-\theta)\Delta t\bar{A}]f(t_{h+1})$$
$$[I - \theta\Delta t\bar{A}]f(t_h) = f\left(t_{h+\frac{1}{2}}\right)$$

- Forward scheme

$$[I - \theta\Delta t\bar{A}]'p\left(t_{h+\frac{1}{2}}\right) = p(t_h)$$
$$p(t_{h+1}) = [I + (1-\theta)\Delta t\bar{A}]'p\left(t_{h+\frac{1}{2}}\right)$$

- … yield exact same result

Practical Steps

- For this to work we need (at least) 6 things:

1. A representation of a tridiagonal matrix: kMatrix(n,3).
2. A way of multiplying tridiagonal matrix with vector: banmul().
3. A way solving tridiagonal matrix system: tridag().
4. Routines for constructing operators: dx(), dxx().
5. Routine for constructing the matrix $[1+\Delta t\bar{A}]$ and its transpose: calcAx().
6. Put it all together: rollBwd(), rollFwd().

- Today we will focus on the first three steps.

Notes

- In this course vectors (and matrixes) are always zero offset and in increasing order of the state: x[0]<x[1]<x[2]<…

- So vectors and matrixes need to be envisioned as axis and coordinates systems.

$$here \quad \uparrow \begin{bmatrix} x_{n-1} \\ \vdots \\ x_2 \\ x_1 \\ x_0 \end{bmatrix} \qquad vs \quad conventional \quad \downarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$