Computational Finance PDE

Copenhagen University
December 2022

Jesper Andreasen
Saxo Bank, Copenhagen
kwant.daddy@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

$$ds = \mu(t,s)dt + \sigma(t,s)dW$$
$$db/b = r(t,s)dt \qquad (1)$$

- then the expectation

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

- … is the solution to the *backward* PDE

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

- … and

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

- … where p solves the *forward* PDE

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

- 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.

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

Backward Finite Difference Solution

- Theta scheme:

$$f(t_{h+1/2}) = [1+(1-\theta)\Delta t\bar{A}]\, f(t_{h+1})$$

$$[1-\theta\Delta t\bar{A}]\, f(t_h) = f(t_{h+1/2}) \tag{6}$$

- where

$$f(t_h) = (f(t_h, s_0), \ldots, f(t_h, s_{n-1}))' \tag{7}$$

- … is a vector of solution values.

- … and $\bar{A}$ is a *finite difference* approximation to A

$$\bar{A} = -r + \mu\delta_s + \frac{1}{2}\sigma^2\delta_{ss} \tag{8}$$

- … can be represented as a *tridiagonal* matrix.

- We solve the system (6) backward in time:

$$f(t_h) \underbrace{\leftarrow}_{\substack{\text{tridiagonal matrix} \\ \text{inversion}}} f(t_{h+1/2}) \underbrace{\leftarrow}_{\substack{\text{tridiagonal matrix} \\ \text{multiplication}}} f(t_{h+1}) \tag{9}$$

Forward Finite Difference Solution

- Multiply the (6a-b) from the left by vectors $p(t_h), p(t_{h+1/2})$:

$$p(t_h)' f(t_h) = p(t_h)'[1-\theta\Delta t\bar{A}]^{-1} f(t_{h+1/2})$$
$$p(t_{h+1/2})' f(t_{h+1/2}) = p(t_{h+1/2})'[1+(1-\theta)\Delta t\bar{A}] f(t_{h+1}) \qquad (10)$$

- … and thereby

$$[1-\theta\Delta t\bar{A}']p(t_{h+1/2}) = p(t_h)$$
$$p(t_{h+1}) = [1+(1-\theta)\Delta t\bar{A}']p(t_{h+1/2}) \qquad (11)$$

- So the adjoint operator is simply the transpose: $\bar{A}^* = \bar{A}'$.

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.

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