Computational Finance PDE

Copenhagen University
December 2022

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

Outline

- Recap
- The finite difference operators
- 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*.

Recap

- Let

$$
\begin{aligned}
ds &= \mu(t,s)dt + \sigma(t,s)dW \\
db/b &= r(t,s)dt
\end{aligned}
\tag{1}
$$

- then the expectation

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

- … 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

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

- Backward theta scheme:

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

- where

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

- … is a vector of solution values.
- … and $\bar{A}$ is a *finite difference* approximation to A

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

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

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

The finite difference operators

- First and second order $n \times n$ matrix operator δ_s and δ_{ss}

- … meaning

$$\delta_s f(t_h) \approx \left(f_s(t_h, s_0), \dots, f_s(t_h, s_{n-1})\right)' \qquad (10)$$
$$\delta_{ss} f(t_h) \approx \left(f_{ss}(t_h, s_0), \dots, f_{ss}(t_h, s_{n-1})\right)'$$

- Use finite differences to estimate

- First order we use weighted average of upward and downward differencing.

- Upward first order finite difference

$$\left(\delta_s^+ f(t_h)\right)_i = \frac{f(s_{i+1}) - f(s_i)}{s_{i+1} - s_i} \qquad (11)$$

- … so upward $n \times 3$ finite difference operator

$$(\delta_s^+)_i = \left[0, \frac{-1}{\mathrm{s_{i+1}} - \mathrm{s_i}}, \frac{1}{\mathrm{s_{i+1}} - \mathrm{s_i}}\right] \qquad , 0 \leq i < n - 1 \qquad (12)$$

- Downward first order finite difference

$$\left(\delta_s^- f(t_h)\right)_i = \frac{f(s_i)-f(s_{i-1})}{s_i-s_{i-1}} \qquad (13)$$

- … so downward $n \times 3$ finite difference operator

$$(\delta_s^-)_i = \left[\frac{-1}{s_i-s_{i-1}}, \frac{1}{s_i-s_{i-1}}, 0\right] \qquad , 0 < i \ \leq n-1 \qquad (14)$$

- Central first order finite difference

$$\left(\delta_s f(t_h)\right)_i = \frac{s_{i+1}-s_i}{s_{i+1}-s_{i-1}}\delta_s^- f(s_i) + \frac{s_i-s_{i-1}}{s_{i+1}-s_{i-1}}\delta_s^+ f(s_i) \qquad (15)$$

- … so central $n \times 3$ finite difference operator

$$(\delta_s)_i = \frac{1}{s_{i+1}-s_{i-1}}\left[-\frac{s_{i+1}-s_i}{s_i-s_{i-1}}, \frac{s_{i+1}-s_i}{s_i-s_{i-1}} - \frac{s_i-s_{i-1}}{s_{i+1}-s_i}, \frac{s_i-s_{i-1}}{s_{i+1}-s_i}\right], \qquad 0 < i < n-1 \;(16)$$

- Second order difference operator

$$\left(\delta_{ss} f(t_h)\right)_i = 2\,\frac{\left(\delta_s^+ f(t_h)\right)_i - \left(\delta_s^- f(t_h)\right)_i}{s_{i+1} - s_{i-1}} \qquad (17)$$

- … so second order $n \times 3$ difference operator

$$(\delta_{ss})_i = \frac{2}{s_{i+1} - s_{i-1}} \left[\frac{1}{s_i - s_{i-1}}, \left(\frac{-1}{s_i - s_{i-1}} + \frac{-1}{s_{i+1} - s_i}\right), \frac{1}{s_{i+1} - s_i}\right] \quad , 0 < i < n - 1 \quad (18)$$

- Difference operators δ_s and δ_{ss} are tridiagonal matrices

- So $\bar{A} = -r + \mu\delta_s + \frac{1}{2}\sigma^2\delta_{ss}$ is tridiagonal and so is $I + \Delta t \bar{A}$

Let’s get practical

- Implement kFiniteDifference::dx() and kFiniteDifference::dxx() such that they construct the finite difference operators
- Implement kFd1d::calcAx() to construct the $I + \Delta t \bar{A}$ matrix
- Test your implementations via xFd1d()