Computational Finance Intro

Copenhagen University
April 2024

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

Outline

- Purpose of course.
- Structure of course.
- C++ and Excel.
- Let’s get practical.

Material

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

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

Purpose of Course

- Practical introduction to the fascinating world of computational finance.
- I.e. how do we get math into computers.
- Aim to produce something quite close to production quality.
- Focus on model evergreens rather than the latest fashion.
- Focus more on code structure and code hygiene.
- The latter is typically not something you see at universities.
- But it is super important in practice.

- And there are some super useful and cheap tricks.
- Aim to have something that you can take offset in for use in your later life.
- … as a graduate student or when working in a financial institution.

Structure of Course

- We will spend very little time on lecturing.

- We will attempt to get jiggy with code and numbers as soon as possible.

- Plan is

    - XLL, C++, Black-Scholes as an example.

    - kFd1d finite difference solver.

    - Application of the above to do something relevant: Black-Scholes PDE, Vasicek PDE, American options, Barrier options, convergence, etc.

- You write a report in groups of up to 3 – oral defense individually

- We might make and change plans as we go.
- Course material will be on GitHub.
- Teachers are Frederik and Brian
- We will take turns on who is lecturing and TA’ing.

C++ and Excel

- The industry standard is to use Excel as an interface on top of a C++ add-in.
- That is Excel is essentially a GUI and all significant calculations are taking place in the C++ coded XLL.
- When used correctly, i.e. sparsely and without buttons and macros, Excel is excellent.
- And C++ is one of the most versatile and efficient computer language in existence.
- It is also one of the most actively researched languages and has been so for 30-40 years.

- Don’t expect this standard to change.
- Though Python, Linux etc is seeing increased usage.

Let’s Get Practical

- Install latest Visual Studio C++ 2022 community version on your laptop. Include the module ‘Desktop development with C++’.
- Download/Clone the git repository ‘CompFin’ from GitHub:
  https://github.com/brnohu/CompFin
- Open Week1/xladdin/CompFin.sln and follow Week1/slnSetup.pdf
- Run the spreadsheet bachelier.xls through the debugger.
- Check the function xBachelierCall () against handmade calculations in a spreadsheet.
- Test the function xBachelierImplied().

- Write a new class kBlack that has the same methods as kBachelier.

- Make xll functions xBlackCall() and xBlackImplied() that expose kBlack::call() and kBlack::implied() respectively.

- Test the functions xBlackCall() and xBlackImplied().

Notes

Bachelier formula:

$$c=(s-k)\Phi(x)+v\sqrt{T}\phi(x) \quad ,x=\frac{s-k}{v\sqrt{T}}$$

Black formula:

$$C=S\Phi(x_{+})-K\Phi(x_{-}) \quad ,x_{\pm}=\frac{\ln(S/K)}{v\sqrt{T}}\pm\frac{1}{2}v\sqrt{T}$$