To: **Computational Finance Course Participants**
Fr: **Frederik Kryger-Baggesen**
St: **Week 2**
Dt: **03 May 2024**

Agenda

Class 03 May 2024 13:15-15 in room on 4^{th} floor in Math building

Bring your Windows laptop.

The backward and forward pde and their associated discretized theta scheme.

The need for matrix algebra for banded – and especially tridiagonal – matrices

Implementation of some Numerical Recipes algorithms

To do list

Install latest Visual Studio C++ 2022 community version on your laptop. Include the module 'Desktop development with C++'.

Download the Week 2 solution from GitHub: https://github.com/brnohu.

Verify the solution compiles and that you can access the debugger from Excel.

Implement today's code in the solution file from week2, i.e. not your own from week1

Implement the kMatrixAlgebra::banmul() and kMatrixAlgebra::tridag() functions in the kMatrixAlgebra namespace using the predefined declarations. The algorithms can be found in the attached from 'Numerical Recipes'.

Expose both functions into two new Excel function xBanmul() and xTridag() in the xlExport.cpp in either CompFin_32 or CompFin_64.

Create a spreadsheet where you call your exposed implementation. Verify that output is correct and step through your functions in the debugger.

If time allows, implement the functions kFiniteDifference::dx(), kFiniteDifference::dxx() and kFd1d::calcAx()

Notes

Pde

$$0 = \partial_t f + Af, \qquad A = -r + \mu f_s + \frac{1}{2}\sigma^2 f_{ss}$$

Backward theta 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 theta scheme

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

Where ‘ denotes transpose, I denotes the identity matrix, $\bar{A}$ denotes the discretized approximation to A and $0 \leq \theta \leq 1$ is a constant.

Note that the backward scheme is solved for $f(t_h)$ given the values of $f(t_{h+1})$, while the forward scheme is solved for $p(t_{h+1})$ given the values of $p(t_h)$, i.e. we run backward and forward in time respectively.