Université Paris-Dauphine PSL

MSc 203 - Financial Markets

C++ Project
Finite Difference Method Pricer

Author:
Clémence RIBUOT-SOUPISON
Meriem HRITTANE
Alexandre PUCH

Teacher:
Johan Mabille

2020 - 2021

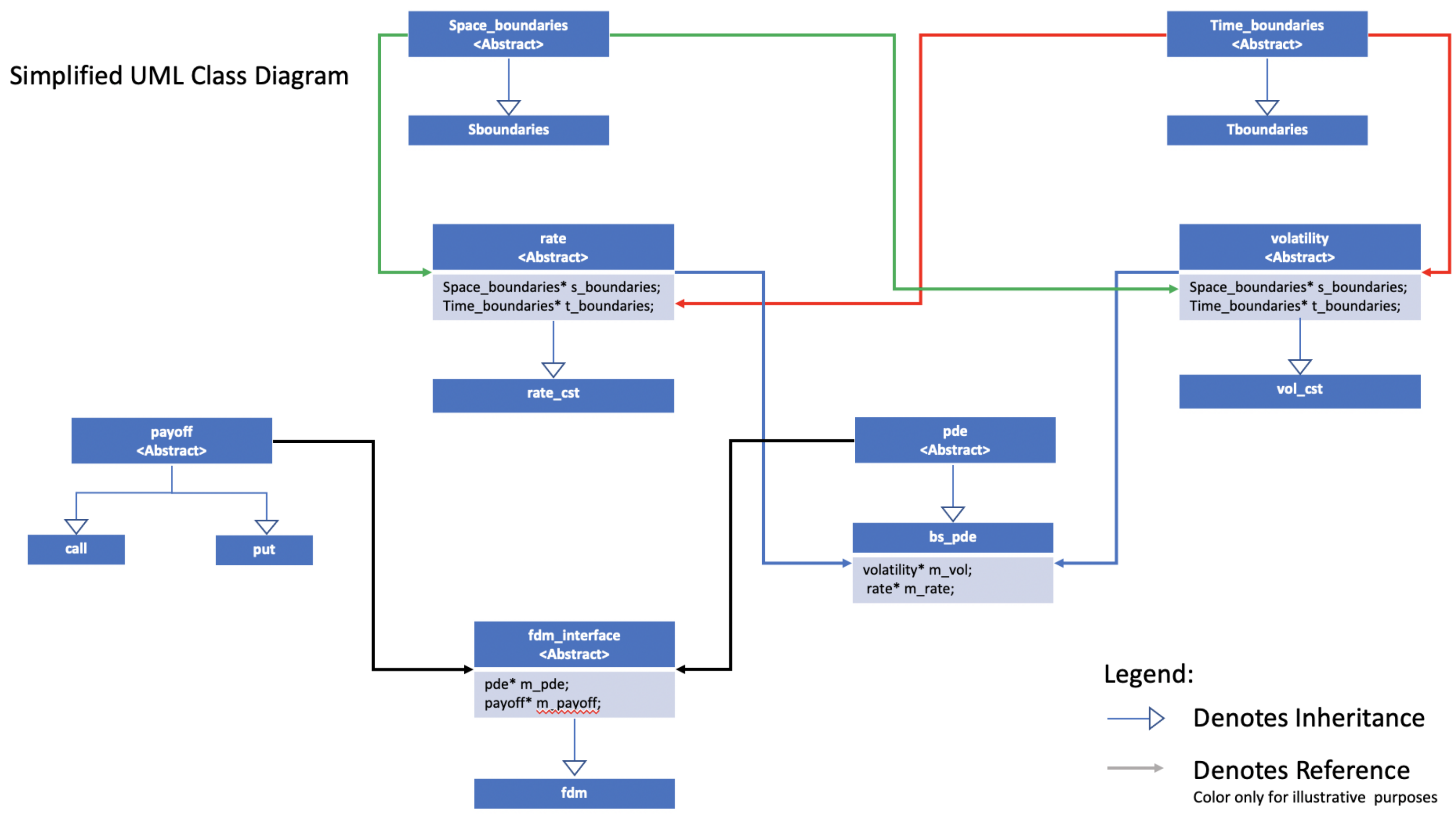
Contents

Chapter 1

Architecture

To launch the pricer, the user has to put the input parameters in the "global.hpp" file. We advise to use a small dx, i.e. a small space mesh, for the code to run smoothly.

The architecture of our code is on the following page.


Simplified UML Class Diagram
Space_boundaries
<Abstract>
Sboundaries
Time_boundaries
<Abstract>
Tboundaries
rate
<Abstract>
Space_boundaries* s_boundaries;
Time_boundaries* t_boundaries;
rate_cst
volatility
<Abstract>
Space_boundaries* s_boundaries;
Time_boundaries* t_boundaries;
vol_cst
payoff
<Abstract>
call
put
pde
<Abstract>
bs_pde
volatility* m_vol;
rate* m_rate;
fdm_interface
<Abstract>
pde* m_pde;
payoff* m_payoff;
fdm
Legend:
Denotes Inheritance
Denotes Reference
Color only for illustrative purposes

Chapter 2

Mathematics behind the pricer

With the theta scheme, the Black-Scholes equation can be written as, $\forall n \in [\![0, T]\!]$, and $\forall i \in [\![0, N]\!]$:

$$\frac{f_i^{n+1} - f_i^n}{dt} = \theta L_i^n + (1-\theta) L_i^{n+1} \tag{2.1}$$

With:

$$L_i^n = \frac{1}{2}\sigma^2 \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{dx^2} + (\frac{1}{2}\sigma^2 - r)\frac{f_{i+1}^n - f_{i-1}^n}{2dx} + rf_i^n \tag{2.2}$$

We set:

$$\alpha = \frac{1}{2}\sigma^2 \frac{1}{dx^2} \tag{2.3}$$

and

$$\beta = (\frac{1}{2}\sigma^2 - r)\frac{1}{2dx} \tag{2.4}$$

We thus have $\forall n \in [\![0, T]\!]$, and $\forall i \in [\![1, N]\!]$:

$$L_i^n = \alpha(f_{i+1}^n - 2f_i^n + f_{i-1}^n) + \beta(f_{i+1}^n - f_{i-1}^n) + rf_i^n \tag{2.5}$$

i.e.

$$L_i^n = f_{i-1}^n(\alpha - \beta) + f_i^n(r - 2\alpha)f_{i+1}^n(\alpha + \beta) \tag{2.6}$$

$$\text{By setting: } \begin{cases} \text{A} = \text{dt } (\alpha - \beta) \\ \text{B} = \text{dt (r - 2}\alpha) \\ \text{C} = \text{dt}(\alpha + \beta) \end{cases}$$

So, Eq. 2 is equivalent to, $\forall n \in [\![0, T]\!]$, and $\forall i \in [\![1, N]\!]$:

$$-(1-\theta)Af_{i-1}^{n+1} + (1-(1-\theta)B)f_i^{n+1} - C(1-\theta)f_{i+1}^{n+1} = A\theta f_{i-1}^n + (1+)f_i^n - C\theta f_{i+1}^n \tag{2.7}$$

We then set for simplification purposes:

$$\begin{cases} \text{a}_1 = -(1-\theta)A \\ \text{a}_2 = -(1-(1-\theta))B \\ \text{a}_3 = -(1-\theta)C \\ \text{b}_1 = \theta A \\ \text{b}_2 = (1+\theta)B \\ \text{b}_3 = \theta C \end{cases}$$

We thus get the following the matrix system (N+1,N+1):

$$\begin{bmatrix} 1 & 0 & 0 & ... & ... & ... & ... \\ a_1 & a_2 & a_3 & 0 & ... & ... & ... \\ 0 & a_1 & a_2 & a_3 & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & a_1 & a_2 & a_3 & 0 \\ ... & ... & ... & ... & a_1 & a_2 & a_3 \\ ... & ... & ... & ... & ... & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0^{n+1} \\ f_1^{n+1} \\ f_2^{n+1} \\ ... \\ f_{N-2}^{n+1} \\ f_{N-1}^{n+1} \\ f_N^{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & ... & ... & ... & ... \\ b_1 & b_2 & b_3 & 0 & ... & ... & ... \\ 0 & b_1 & b_2 & b_3 & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & b_1 & b_2 & b_3 & 0 \\ ... & ... & ... & ... & b_1 & b_2 & b_3 \\ ... & ... & ... & ... & ... & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0^n \\ f_1^n \\ f_2^n \\ ... \\ f_{N-2}^n \\ f_{N-1}^n \\ f_N^n \end{bmatrix} \tag{2.8}$$

We can express this system as a (N-1,N-1) system:

$$\begin{bmatrix} b_1 & b_2 & b_3 & 0 & ... & ... & ... \\ 0 & b_1 & b_2 & b_3 & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & b_1 & b_2 & b_3 & 0 \\ ... & ... & ... & ... & b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} f_1^n \\ f_2^n \\ ... \\ f_{N-2}^n \\ f_{N-1}^n \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & 0 & ... & ... & ... \\ 0 & a_1 & a_2 & a_3 & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & a_1 & a_2 & a_3 & 0 \\ ... & ... & ... & ... & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} f_1^{n+1} \\ f_2^{n+1} \\ ... \\ f_{N-2}^{n+1} \\ f_{N-1}^{n+1} \end{bmatrix} + \begin{bmatrix} a_1(1-b_1e^{-rdt})f_0^{n+1} \\ 0 \\ ... \\ 0 \\ a_3(1-b_3e^{-rdt})f_N^{n+1} \end{bmatrix} \tag{2.9}$$

We thus have:

$$M^R F_t = D_{t+1} \tag{2.10}$$

With M^R a triangualr matrix we inverse through the Thomas algorithm.
For each step, we go back in time through the algorithm until we are at time 0 where we will get the price of the instrument.