École nationale des ponts et chaussées

EDS SIMULATION

Efficient Simulation of the Heston Stochastic Volatility Model

Done by :

Yassir EL ORFI
Chadi NFISSI

Under the direction of:

Mr.Benoit VANDEVELDE

[0]Pdf here

Contents

The work entrusted us by our professor is using two methods to price options based on different models, since we are a two person group, each person developed a method hence the project containing two main folders; each folder is a method with its own Main.cpp implementation, we apologizing heavily for not optimizing the script. We sincerely would like to thanks our dear Professor Mr Benoit for all the efforts he had shown us during the entire year.

This paper sole reason is to compare the results of quadratic exponential scheme with analytical calculus of calls, we will proceed in a orderly fashion:

- Build up to QE scheme: presentation of past schemes
- Alternative: Analytical solution
- Comparison

1 Premilinary: Brownien Mouvement

1.1 Box Muller:

First, for the generation of the standardized random variable, we proceed as follows:

- Change the scale $u \to \tilde{u} = \frac{u}{\textbf{RAND MAX}}$ such that $\tilde{u} \in [0, 1]$.
- Generate two uniformly distributed variables in: $[0, 1]$.
  - $u_1 = \frac{\textbf{rand}}{\textbf{RAND MAX}}$
  - $-u_2 = \frac{\textbf{rand}}{\textbf{RAND MAX}}$
- Generate G such that: $G = \sqrt{-2logu_1}sin(2\pi u_2) \to \mathcal{N}(0, 1)$
- Then, we generate the Brownian motion by following the algorithm:
- Choose a time interval $\Delta_t, t_0 = 0, n = floor(\frac{T}{\Delta_t})$ For j going from 0 to n, $t_j = t_{j-1} + \Delta_t$, generate the Gaussian variable $\mathcal{N}_j$. $W_{tj} = W_{tj-1} + \mathcal{N}_j * \sqrt{\Delta_t}$, and increment j.

1.2 Results:

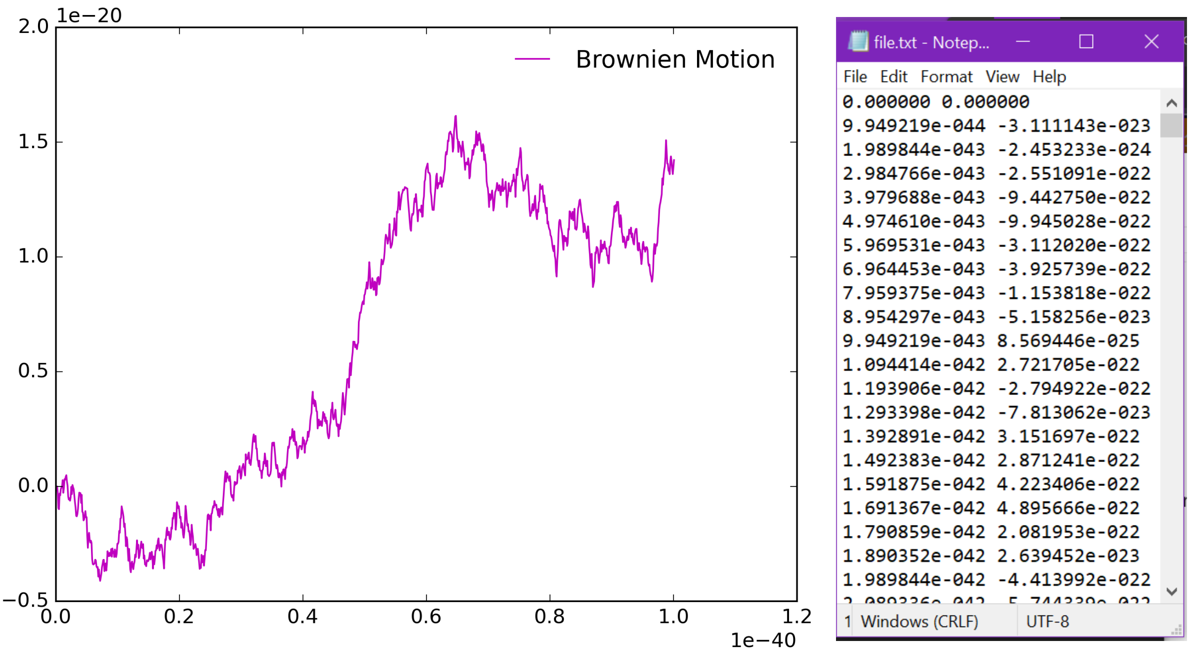

Figure 1: Brownien Motion

1.3 La formule close de Black-Scholes:

The Black and Scholes Call is priced as follows:

$$Call_{BS} = S\mathcal{N}(d_1) - K\exp(-r\theta)\mathcal{N}(d_2)$$

Avec: $d_1 = \frac{log(\frac{S}{K})+(r+\frac{\sigma^2}{2})\theta}{\sigma\sqrt{\theta}}$, $d_2 = d_1 - \sigma\sqrt{\theta}$, S is the underlying asset price, and K the option's Strike.

Likewise we obtain the Put formula, or we can use the Call Put Parity

2 Chapter 1: Schemes

2.1 Euler & Milstein discretisation

The Euler scheme and the Milstein scheme are both numerical methods for solving stochastic differential equations (SDEs).

The Euler scheme is a first-order method, meaning that it approximates the solution to the SDE by discretizing the time axis and updating the state of the system at each time step based on a local linearization of the system dynamics.

The Milstein scheme is a second-order method, meaning that it builds on the Euler scheme by incorporating the derivative of the drift term in the SDE. This allows the Milstein scheme to more accurately capture the dynamics of the system, especially when the drift term is nonlinear.

2.1.1 Formula:

For every SDE like :

$$dX_t^n = b(X_t^n, t)dt + \sigma(X_t^n, t)dW_t$$

Euler scheme is as follows :

$$X^n_{(k+1)\Delta_t} = X^n_{k\Delta_t} + b(X^n_{k\Delta_t})\Delta_t + \sigma(X^n_{k\Delta_t})(W_{(k+1)\Delta_t} - W_{k\Delta_t})$$
$$X^n_{(k+1)\Delta_t} = X^n_{k\Delta_t}(1 + r\Delta_t + \sigma\Delta W_k)$$

. As k is the iteration step.
And Milstein scheme:

$$X^n_{(k+1)\Delta_t} = X^n_{k\Delta_t} + (b(X^n_{k\Delta_t})\frac{1}{2}\sigma(X^n_{k\Delta_t})\sigma(X^n_{k\Delta_t}))\Delta_t + \sigma(X^n_{k\Delta_t})(W_{(k+1)\Delta_t}W - k\Delta_t) +$$

$$\frac{1}{2}\sigma'(X^n_{k\Delta t})\sigma(X^n_{k\Delta t})(W_{(k+1)\Delta t}W_{k\Delta t})^2$$

$$X^n_{(k+1)\Delta t} = X^n_{k\Delta t}(1 + (r - \frac{1}{2}\sigma^2)\Delta t + \sigma\Delta W_k + \frac{1}{2}\sigma^2(\Delta W_k)^2)$$

2.2 Quadratic-Exponential Scheme

The QE scheme is introduced in order to fix some of the Truncated-Gaussian flaws, starting by the fast decay around the upper tail of the distribution.

Let's set our main dynamics (Euler Scheme):

$$lnX_{t+\Delta t} = ln(X_t - \frac{1}{2}\hat{V}(t)^+\Delta + \sqrt{\hat{V}(t)^+}Z_X\sqrt{\Delta}$$

$$\hat{V}_{t+\Delta t} = \hat{V}_t + \kappa(\theta - \hat{V}_t^+)\Delta + \sqrt{\hat{V}(t)^+}Z_V\sqrt{\Delta}$$

2.2.1 Algorithm:

QE assumes that for sufficiently large values of $\hat{V}(t)$, we write $\hat{V}(t+\Delta) = a(b+Z_V)^2$ and we set the following parameters:

$$\begin{aligned} m &= \theta + \left(\hat{V}(t) - \theta\right) e^{-\kappa\Delta}; \\ s^2 &= \frac{\hat{V}(t)\varepsilon^2 e^{-\kappa\Delta}}{\kappa}\left(1-e^{-\kappa\Delta}\right) + \frac{\theta\varepsilon^2}{2\kappa}\left(1-e^{-\kappa\Delta}\right)^2, \end{aligned}$$

$$\psi = \frac{s^2}{m^2} = \frac{\frac{\hat{V}(t)\varepsilon^2 e^{-\kappa\Delta}}{\kappa}\left(1-e^{-\kappa\Delta}\right) + \frac{\theta\varepsilon^2}{2\kappa}\left(1-e^{-\kappa\Delta}\right)^2}{\left(\theta + \left(\hat{V}(t)-\theta\right)e^{-\kappa\Delta}\right)^2}.$$

if we set

$$b^2 = 2\psi^{-1} - 1 + \sqrt{2\psi^{-1}}\sqrt{2\psi^{-1} - 1} \geq 0$$

and

$$a = \frac{m}{1+b^2}$$

Then

$$\mathbf{E}\left(\hat{V}_{t+\Delta t}\right) = m, \quad Var\left(\hat{V}_{t+\Delta t}\right) = s^2$$

Having defined the necessary parameters and equation:

1. Given $\hat{V}_t$, compute m nad s^2.
2. Compute $= \frac{s^2}{m^2}$.
3. Draw U_v a random uniform number.
4. If $\psi \leq 1.5$:
   (a) Compute a and b.
   (b) Compute $Z_v = \phi^{-1}(U_v)$.
   (c) Set $\hat{V}_{t+\Delta t} = a(b+Z_V)^2$.
5. else:
   (a) compute β, p.
   (b) $\hat{V}_{t+\Delta t} = \psi^{-1}(U_v; p, \beta)$

3 Chapter 3: Semi-analytical Heston Model

The purpose of this section is to implement the semi-analytical method of the Heston model.

3.1 Class HestonModel

$$\begin{cases} dS_t = rS_t dt + \sqrt{V_t} S_t dW_t^S \\ dV_t = \kappa(\theta - V_t)dt + \sigma_v\sqrt{V_t} dW_t^V \\ d < W_t^S, W_t^V >= \rho dt \end{cases}$$

with :

- R : risk-free interest rate.
- ρ : the correlation between the two Brownian motions.
- S_0 : the initial value of the underlying asset at the valuation date $t_0= 0$.
- V_0 : the initial value of the variance process at the valuation date $t_0 = 0$.
- κ: the speed of mean reversion.
- θ : mean reversion.
- σ_v : the volatility of volatility.

These parameters will be the parameters defined in our HestonModel class.

3.2 Class HestonPricer

Using semi analytical method, , the price of a Call option is as follows:

$$C(S_0, V_0, K) = S_0 P_1 - K e^{-rT} P_2$$

3.3 Class AnalyticalHestonPricer

Now we need to calculate P1 and P2 to find the result. Therefore, we need to implement the following functions: with i = 1 or 2

$$\boxed{P_i = \frac{1}{2} + \frac{1}{2\pi} \int_{\omega=-\infty}^{+\infty} \Re\left[\frac{\phi_i(T, \ln(S_0), V_0, \omega) e^{-j\omega \ln(K)}}{j\omega}\right] d\omega}$$

$$\boxed{\phi_i(\tau, x, V, \omega) = e^{C^{(i)}(\tau,\omega) + D^{(i)}(\tau,\omega)V + j\omega x}}$$

$$\begin{cases} C^{(i)}(\tau, \omega) = j\omega r\tau + \frac{\kappa\theta}{\sigma_v^2}\left[(a-b)\tau - 2\ln\left(\frac{1-ge^{-b\tau}}{1-g}\right)\right] \\ D^{(i)}(\tau, \omega) = \frac{a-b}{\sigma_v^2} \frac{1-e^{-b\tau}}{1-ge^{-b\tau}} \end{cases}$$

$$\begin{cases} a = \kappa - \rho\sigma_v y_i \\ \Delta = a^2 + \sigma_v^2 \left(u_i j\omega + \omega^2\right) \\ b = \sqrt{\Delta} \\ g = \frac{a-b}{a+b} \end{cases} \qquad \begin{cases} u_1 = -1 \\ u_2 = 1 \\ y_1 = j\omega - 1 \\ y_2 = j\omega \end{cases}$$

3.4 Class Complex

To perform various operations such as division, multiplication, etc. on complex numbers, we need to define these methods within a Complex class

3.5 Class GaussLegendreQuadrature

To calculate the integral needed to determine P1 and P2 ,

$$\int_{\omega=-\infty}^{+\infty} \Re\left[\frac{\phi_i(T, \ln(S_0), V_0, \omega)e^{-j\omega \ln(K)}}{j\omega}\right] d\omega$$

and the approximations:

$$\boxed{\int_{-\infty}^{+\infty} f(\omega)d\omega = \int_{-1}^{1} g(x)dx \approx \sum_{i=1}^{n} w_i \; g(x_i)}$$

$$\boxed{g(x) = f\left(\frac{x}{1-x^2}\right)\frac{1+x^2}{(1-x^2)^2}}$$

4 Chapter 4: concluding

We can conclude(if we have done things correctly to a certain degree) that both methods are contenders for option pricing, the difference in prices might be to the limitations to both models, or due to the necessity of calibrating each model to market places to reach a convergence point. We suggest using a Maximum likelihood estimation to estimate the parameters.

We also used CHAT-GPT in order to solve some of our bugs, in order to calculate conditional expectance or the expectance in general, we used bins method which was tricky at first to implement, but using the AI, it oriented us by proposing using PAIR class to pair values in a vector and make calculations useful,and also introduced us to the Newton-Raphson optimizing method and templates in C++.