Computational Finance Fall 2020 - Assignment 2

Hien Lê - hien.le@epfl.ch

October 23, 2020

Exercise 2

OU process:

$$dX_t = \kappa(\theta - X_t)dt + \lambda dW_t$$

$$v(t,x) = E[\exp(i\nu X_T)|X_t = x];\ v(T,x) = \exp(i\nu x)$$

a. Deduce a system of ODEs for ϕ and ψ in $v(t,x) = \exp(\phi(T-t,\nu) + \psi(T-t,\nu)x)$.

Given that we have the PDE:

$$v_t + \kappa(\theta - x)v_x + \frac{\lambda^2}{2}v_{xx}$$

we proceed to calculate:

$$\begin{aligned} v_t &= (-\phi(T-t,\nu) - \psi(T-t,\nu)x)v(t,x) \\ v_x &= \psi(T-t,\nu)v(t,x) \\ v_{xx} &= \psi^2(T-t,\nu)v(t,x) \end{aligned}$$

Substituting to the PDE, we get:

$$[-\phi'(T-t,\nu) - \psi'(T-t,\nu)x]v(t,x) + \kappa(\theta - x)\psi(T-t,\nu)v(t,x) + \frac{\lambda^2}{2}\psi^2(T-t,\nu)v(t,x) = 0 \quad (1)$$

Grouping the terms that have x and those that don't, and discarding $v(t,x)$, we get:

$$x(-\psi'(T-t,\nu) - \kappa\psi(T-t,\nu)) - \phi'(T-t,\nu) + \kappa\theta\psi(T-t,\nu) + \frac{\lambda^2}{2}\psi^2(T-t,\nu) = 0 \quad (2)$$

Since the above must hold for all x, we can deduce the following system of ODEs:

$$\psi'(T-t,\nu) + \kappa\psi(T-t,\nu) = 0 \tag{3}$$

$$-\phi'(T-t,\nu) + \kappa\theta\psi(T-t,\nu) + \frac{\lambda^2}{2}\psi^2(T-t,\nu) = 0$$

$$\Leftrightarrow \phi'(T-t,\nu) - \kappa\theta\psi(T-t,\nu) - \frac{\lambda^2}{2}\psi^2(T-t,\nu) = 0 \tag{4}$$

b. Solve the newly found system explicitly, given $X_0 = x$.

Given that the boundary condition is $v(T,x) = e^{i\nu x}$:

$$v(T,x) = exp[\phi(0,\nu) + \psi(0,\nu)x]$$

and so $\phi(0,\nu) = 0;\ \psi(0,\nu) = i\nu$.

Now, coming back to the ODE, we can solve the first ODE (3) as follows:

$$\begin{aligned}
\psi'(T-t,\nu) &= -\kappa\psi(T-t,\nu) \\
\Rightarrow \frac{d\psi}{\psi} &= -\kappa dt \\
\int \frac{d\psi}{\psi} &= \int -\kappa dt \\
\ln(\psi(T-t,\nu)) - \ln(\psi(0,\nu)) &= -\kappa(T-t) \\
\Rightarrow \psi(T-t,\nu) &= i\nu e^{-\kappa(T-t)}
\end{aligned}$$

Knowing $\psi(T-t,\nu)$, solving for $\phi(T-t,\nu)$ from (4) is simple:

$$\begin{aligned}
\phi'(T-t,\nu) &= \kappa\theta\psi(T-t,\nu) + \frac{\lambda^2}{2}\psi^2(T-t,\nu) \\
\phi'(T-t,\nu) &= \kappa\theta i\nu e^{-\kappa(T-t)} - \frac{\lambda^2}{2}\nu^2 e^{-2\kappa(T-t)} \\
\Rightarrow \phi(T-t,\nu) &= i\kappa\nu\theta\int_t^T e^{-\kappa(T-s)}ds - \frac{\lambda^2}{2}\nu^2\int_t^T e^{-2\kappa(T-s)}ds \\
&= i\nu\theta(1-e^{-\kappa(T-t)}) + \frac{\lambda^2\nu^2}{4\kappa}(e^{-2\kappa(T-t)} - 1)
\end{aligned}$$

Substituting the above to the Ansatz of $v(t,x)$, we get:

$$v(t,x) = \exp\left(i\nu e^{-\kappa(T-t)}x + i\nu\theta(1-e^{-\kappa(T-t)}) - \frac{\lambda^2\nu^2}{4\kappa}(1-e^{-2\kappa(T-t)})\right)$$

The above is therefore the characteristic function of X_t. c. Deduce that $X_t \sim N$ and give its parameters.

The characteristic function of any RV X is defined as $v_X(\nu) = E[e^{i\nu X}]$ and the characteristic of a Normal RV is $v_X(\nu) = e^{i\mu\nu + \frac{\nu^2\sigma^2}{2}}$ where μ, σ are the parameters of the Normal distribution in question. In our case, we saw this function to be:

$$\exp\left(i\nu[e^{-\kappa(T-t)}x + \theta(1 - e^{-\kappa(T-t)})] - \frac{\lambda^2\nu^2}{4\kappa}(1 - e^{-2\kappa(T-t)})\right)$$

If we take $\mu = e^{-\kappa(T-t)}x + \theta(1 - e^{-\kappa(T-t)})$, and $\sigma^2 = -\frac{\lambda^2}{2\kappa}(1 - e^{-2\kappa(T-t)})$, we can deduce that the variable X_t is normally distributed with the above mean and variance.

d. Solve the OU process explicitly using Itô.

To solve the OU process, we multiply X_t by the integrating factor $e^{-t)}$ and solve for $d(e^{-\kappa t}X_t)$, using the Ito's formula:

$$\begin{aligned}
d(e^{\kappa t}X_t) &= \kappa e^{\kappa t}X_t dt + e^{\kappa t}dX_t \\
&= e^{\kappa t}X_t dt + e^{\kappa t}\kappa(\theta - X_t)dt + e^{\kappa t}\lambda dW_t \\
&= e^{\kappa t}\kappa\theta dt + e^{\kappa t}\lambda dW_t \\
\Rightarrow e^{\kappa t}X_t - X_0 &= \int_0^t e^{\kappa s}\kappa\theta ds + \lambda\int_0^t e^{\kappa s}dW_s \\
X_t &= e^{-\kappa t}x + \kappa\theta\int_0^t e^{-\kappa(t-s)}ds + \lambda\int_0^t e^{-\kappa(t-s)}dW_s \\
&= e^{-\kappa t}x + \theta(1 - e^{-\kappa t}) + \lambda\int_0^t e^{-\kappa(t-s)}dW_s
\end{aligned}$$

The above gives the solution of X on the interval $[\mathbf{0}, \mathbf{t}]$. Now, as found in the first take-home exam, we have that $X \sim N$ with the parameters:

$$\mu = E[X_t] = e^{-\kappa t}x + \theta(1 - e^{-\kappa t}) \tag{5}$$

$$\sigma^2 = Var[X_t] = \lambda^2\int_0^t e^{-2\kappa(t-s)}ds \tag{6}$$

$$= \frac{\lambda^2}{2\kappa}(1 - e^{-2\kappa t}) \tag{7}$$

This is consistent with the result found in part c where X was calculated on the interval $[\mathbf{t}, \mathbf{T}]$.

Exercise 4

See code attached.

Reminder of Heston model:

$$\begin{aligned}
dS_t &= rS_t dt + S_t\rho\sqrt{V_t}dW_{1t} + S_t\sqrt{1-\rho^2}\sqrt{V_t}dW_{2t} \\
dV_t &= \kappa(\theta - V_t)dt + \sigma\sqrt{V_t}dW_{1t}
\end{aligned} \tag{8}$$

For this exercise, we used the provided functions `Call_Heston` and `fminsearchcon` (these programs remained unchanged) to find the value of the set of parameters q that minimises the root mean squared error between the Heston-computed prices and the observed ones in the data (`Call_20050103.mat`), where q is $(\theta\ \kappa\ \sigma\ \rho\ V_0)^T$. The initial values of each of these parameters have been given by the exercise. Looking at the system in Equation (8), we notice that some of these parameters are constrained, specifically:

$$\begin{aligned} \kappa &\geq 0 \\ \sigma &\geq 0 \\ -1 \leq &\rho \leq 1 \\ V_t &\geq 0 \end{aligned}$$

these are to be defined as the lower and upper bounds in our optimisation algorithm.

The other two elements that are used to compute the Heston prices are $r = 0.015$ and $S = 1202.10$, which are constant.

Take N the number of entries in our data and $\hat{C}(T_i, K_i)$ each observed price, eventually what we want is q^* such that:

$$q^* = \min_q \sqrt{\frac{1}{N}\sum_i^N [heston(q, T_i, K_i) - \hat{C}(T_i, K_i)]^2}$$

Here we are dealing with a vector-valued optimisation function, which makes the process differ from a simple scalar optimisation. One idea to simplify the process is to solve for q^* that minimises the **sum of squared errors**, which is equivalent to the problem posed above. I first tried using the built-in *lsqnonlin* function in Matlab, but found that this function is rather unstable and the values given by its two options of algorithms differ rather significantly. The default algorithm (trust-region-reflective) gave the following results for $q*$ after 22 iterations:

$$\begin{aligned} \theta &= 0.0130 \\ \kappa &= 5.6221 \\ \sigma &= 0.2707 \\ \rho &= -0.7228 \\ V_0 &= 0.0119 \end{aligned}$$

Meanwhile, when using a manually written cost function that takes the **square root of the mean of** the squared errors (RMSE), I got the following set of parameters after 492 iterations, which are slightly different from those above but of which the optimisation appears

to be more stable:

$$\begin{aligned}\theta &= 0.0132\\ \kappa &= 4.6449\\ \sigma &= 0.2457\\ \rho &= -0.6728\\ V_0 &= 0.0115\end{aligned}$$