CY-TECH

Model calibration and simulation

ING3 IFI

Simulation and Calibration of the Heston Model

Author:
Farès Djerourou

February 2023

Grande Ecole de sciences,
d'ingénierie, d'économie et de
gestion de **CY Cergy Paris Université**

Contents

1 Introduction

The original Black-Scholes model assumes that the volatility parameter associated with the dynamic is constant, empirical evidence suggests it's not the case, the Heston model is a two-factor model in which the volatility component of the dynamic follows its own dynamic. The Heston model like the local volatility models seems to more aligned with the reality of the markets, producing the volatility smile characteristic of options with different strikes and maturities. In this short paper we will go through the simulation of the Heston model using Monte-Carlo simulation, we will compute the call price under the Heston model using the same method and we will then proceed to explore some ways to optimize our Monte-Carlo simulation using antithetic variates for variance reduction. Finally we will calibrate the Heston model to some market data using the Levenberg-Marquardt algorithm.

2 The Heston Dynamic

2.1 Stochastic Volatility

The dynamics of the asset price and of the asset's volatility under the Heston model are given by the following system.

$$\begin{cases} \frac{dS_t}{S_t} = rdt + \sqrt{v_t}W_t \\ dv_t = k(\theta - v_t)dt + \eta\sqrt{v_t}B_t \\ < dW_t, dB_t > = \rho dt \end{cases} \tag{1}$$

Here ρ is the correlation coefficient between B_t and W_t, k is the rate of mean reversion, η is the volatility of volatility, and θ is the long-term mean of the volatility.

We will now partition t and use Ito's formula to get the asset price.

$$S_{t+dt} = S_t e^{(r-\frac{v_t}{2})dt+\sqrt{v_t}dW_t}$$

Thus we only have to simulate the volatility paths and plug the values in our asset dynamic. We will use Milstein's scheme for the stochastic volatility discretization.

2.2 Numerical Application

Now we only need to get our Monte-Carlo simulations going, here are the asset and volatility paths in the Heston model.

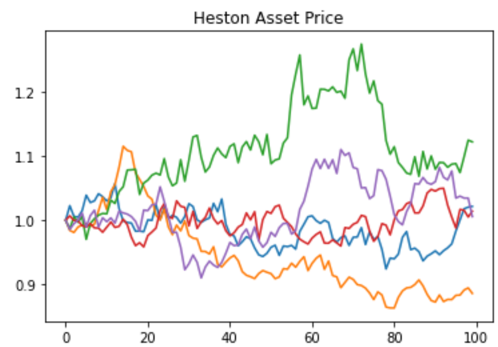

Figure 1: Asset Paths in the Heston Model

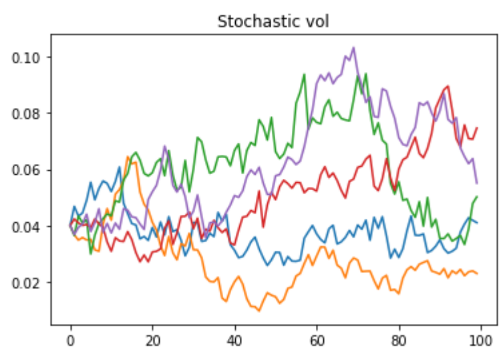

Figure 2: Stochastic Vol Paths in the Heston Model

We will know simulate the Log-return distribution in this model, we notice that it's skewed to the left or to the right depending on the sign of the ρ parameter, thus a tail is heavier than the other..

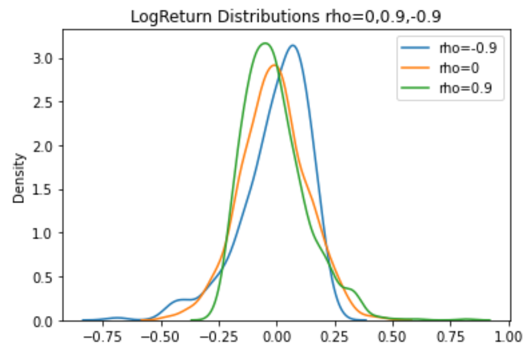

Figure 3: Log-return distributions for different correlation coefficients

Next we compute a call price using Monte-Carlo simulations, we proceed to plot the call for different initial values and use antithetic variates to reduce the variance of the computed call. We compare the call's price accuracy for the same number of Monte-Carlo simulations.

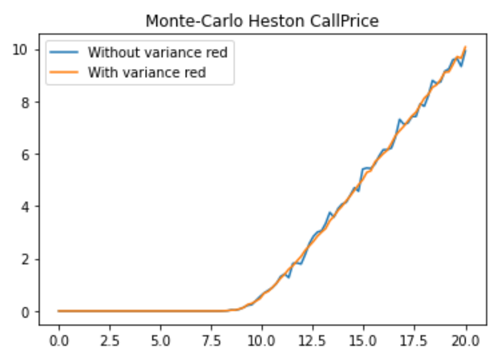

Figure 4: Call price in the Heston model with and without variance reduction

Now it's time to look at the "Greeks", we will need them later for calibration. We used finite differences to compute them since we do not have an analytic formula and everything was done using Monte-Carlo simulations.

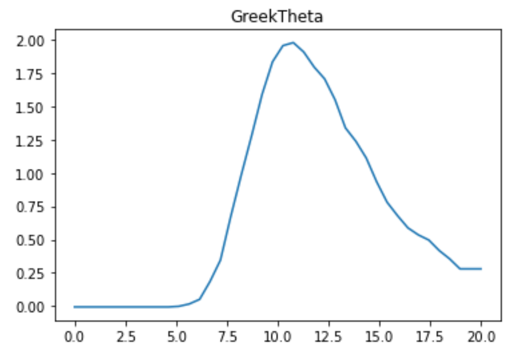

Figure 5: Derivative of the call with respect to Theta

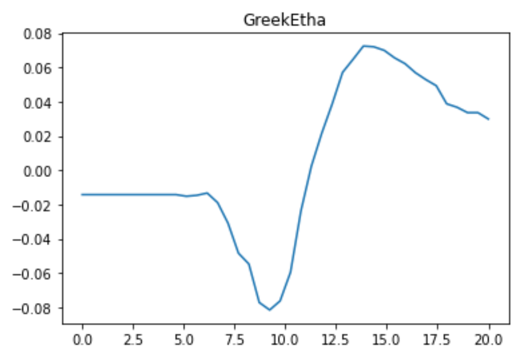

Figure 6: Derivative of the call with respect to Eta

3 Calibration of the Heston Model

3.1 Levenberg-Marquardt

In this section we will calibrate the stochastic volatility in the Heston model using the Levenberg-Marquardt algorithm and the observed volatilities of the asset for some K values. To calibrate the model is to find the parameters minimizing our objective function, there are a lot of optimisation algorithms someone could use to achieve that.

Gradient Descent

$$x_{n+1} = x_n - \lambda J_r^T r(x_n)$$

Gauss-Newton

$$x_{n+1} = x_n - (J_r^T J_r)^{-1} J_r^T r(x_n)$$

The advantage of the Gauss-Newton algorithm over Gradient Descent lies in the fact that $J_r^T J_r$ approximates the Hessian of the function $H \approx J_r^T J_r$ instead of relying on the λ constant at each step, and thus gives faster convergence.
There's a problem with Gauss-Newton's algorithm nevertheless, the Hessian matrix can become ill-conditioned, that is the case, for example, when the initial guess if far from the target solution of the problem, here enters Levenberg-Marquardt.

Levenberg Marquardt

$$x_{n+1} = x_n - (J_r^T J_r + \lambda D)^{-1} J_r^T r(x_n)$$

Unlike the previous algorithms, with the adjustment λD to the Hessian approximation, Levenberg-Marquardt can (often) still find the solution even if the initial guess is far from it, it is more robust, although it might be slower than Gauss-Newton for well-behaved functions.

3.2 Calibration in the Heston Model

The objective function that we are looking to minimize is:

$$\sum_{p=1}^{10} |V^{market}(T_p, K_p) - V^{Heston}(T_p, K_p, \beta_p)|^2$$

We recall that the stochastic volatility's dynamic is given by

$$dv_t = k(\theta - v_t)dt + \eta\sqrt{v_t}B_t$$

We will be using Levenberg-Marquardt as our our optimisation algorithm. The starting values for the parameters will be $\theta = 0.2$ and $\eta = 0.5$. The other parameters will remain constant with the following values:

$$k = 3$$
$$v_0 = 0.04$$
$$\rho = 0.5$$

After a Jacobian calculation and a quick Python implementation we find the following values after 7 iterations!

$$\theta = 0.0817554$$
$$\eta = 0.383547$$

To calibrate the model properly we fixed the random normal matrices beforehand so the Heston price would be less erratic.

4 Volatility Smile

Our aim here was to use the Heston prices and Newton's algorithm to find the corresponding BS implied volatility.

$$\begin{cases} \sigma_{n+1} = \sigma_n - \dfrac{F(i)}{F'(i)} \\ F(i) = Call_{BS}(K_i) - Call_{Heston}(K_i) \end{cases} \tag{2}$$

We find the following volatility smile.

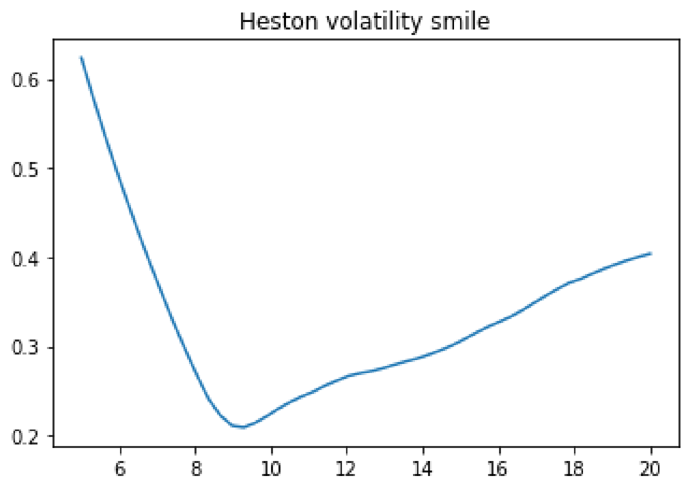

Figure 7: Volatility Smile from Heston Prices