Simulation scheme for the Heston Stochastic Volatility Model

Ricardo Rincón García

Directed by Ahmed KEBAIER

1 Inverse Gaussian approximation

Premia 22

The Heston model is given by the coupled two-dimensional stochastic differential equations,

$$dS_t = rS_t dt + \sqrt{V_t} S_t (\rho dW_t^V + \sqrt{1-\rho^2} dW_t^S) \quad (1.1)$$

$$dV_t = k(\theta - V_t)dt + \sigma \sqrt{V_t} dW_t^V \quad (1.2)$$

where W^V and W^S are two correlated brownian motions in the time variable t. k, θ, σ are positive constants, r is a non-negative constant, and the correlation ρ is a constant in $[-1, 1]$. S_t represents the price of an underlying asset and V_t represents the variance of its instantaneous returns. The initial conditions S_0 and V_0 are assumed to be strictly positive.

Assuming $t_2 > t_1$, the variance process distribution of V_{t_2} conditional on V_{t_1} is given by

$$V_{t_2} = C_0 \chi^2_\delta(\lambda) \quad (1.3)$$

$$C_0 = \frac{\sigma^2(1 - e^{-k(t_2-t_1)})}{4k}, \delta = \frac{4k\theta}{\sigma^2}, \lambda = \frac{4ke^{-k(t_2-t_1)}V_{t_1}}{\sigma^2(1 - e^{-k(t_2-t_1)})}$$

where $\chi^2_\delta(\lambda)$ denotes a non-central chi-squared distribution with δ degrees of freedom and non-centrality parameter λ.

The independence of W^V and W^S implies [3] that the distribution of S_{t_2} given V_{t_1}, V_{t_2}, S_{t_1}, and $I \equiv \int_{t_1}^{t_2} V_s \, ds$ comes from

$$S_{t_2} \sim S_{t_1} e^{\mathcal{N}(r(t_2-t_1) - 0.5I + \frac{\rho}{\sigma}(V_{t_2} - V_{t_1} - k\theta(t_2-t_1) + kI), \sqrt{(1-\rho^2)I})} \quad (1.4)$$

For more details one can see [1] and [2].

1.1 IPZ-IG scheme

The combined IPZ–IG scheme is the scheme proposed in [1] for simulating the Heston model. It is composed of 4 algorithms.

We define two equally spaced grids $\vec{u} = \{0, ..., 1\}$ and $\vec{v} = \{v_{min}, ..., v_{max}\}$, both with N_u number of nodes.

Algorithm 2: Before simulation, we approximate $\vec{Q} = (q_i)$,

- Compute a vector $\vec{p} = (Prob(V_{t_2} \leq v_i))$ whose components are cumulative probabilities on v_i.
- For each index i, if $u_i < p_0$, set $q_i = 0$.
- Otherwise, use a binary search to look for the index j such that p_j is closest to u_i, and set $q_i = v_j$.

Algorithm 1: To sample V_{t_2} conditional on V_{t_1},

- Sample $m_p = Poisson(\lambda/2)$. If $V_{t_1} = 0$, simply set $m_p = 0$ since $Poisson(0)$ is always zero.
- If $m_p = 0$, draw a uniform variate U, find the index i such that u_i is closest to U, and use q_i as the sample. Otherwise, sample from $Gamma(m_p + \delta/2)$.

It is denoted IPZ scheme since we are Interpolating for the case when the Poisson variate is equal to Zero.

Algorithm 4 - Precomputation: Before simulation,

- Precompute $E[I_c]_{V_{t_1}V_{t_2}}$ and $Var[I_c]_{V_{t_1}V_{t_2}}$ on the equally spaced grid $\sqrt{V_{t_1}V_{t_2}} = \vec{v}$.

We do this precomputation to avoid evaluating many times the modified Bessel functions (very expensive) implied if not in the calculations of Algorithm 4.

Algorithm 4: Compute $E[I_c]$ and $Var[I_c]$,

- Compute $E[S_1]$ and $\sigma^2_{S_1}$.
- If $V_{t_1} = 0$ or $V_{t_2} = 0$, $E[\eta] = E[\eta^2] = 0$, and hence $E[I_c] = E[S_1] + E[S_2]$ and $Var[I_c] = \sigma^2_{S_1} + \sigma^2_{S_2}$, i.e. only two additions are required in this step as $E[S_2]$ and $\sigma^2_{S_2}$ are constants.
- Otherwise, use nearest neighbour interpolation to approximate $E[I_c]_{V_{t_1}V_{t_2}}$ and $Var[I_c]_{V_{t_1}V_{t_2}}$ (we use again a binary search). Add them to $E[S_1]$ and $\sigma^2_{S_1}$ to obtain $E[I_c]$ and $Var[I_c]$.

Algorithm 3: Sampling I_c by the moment-matched Inverse Gaussian (IG) distribution, $IG(m, s)$ where $m = E[I_c]$ and $s = E[I_c]^3/Var[I_c]$

- Generate a standard normal variate N and a uniform variate U.
- Compute $x = 1 + N^2/(2s/m) - \sqrt{(2s/m + 2s/m)N^2 + N^4}/(2s/m)$.
- If $U(1 + x) > 1$, $I_c = m/x$. Otherwise, $I_c = mx$.

Finally, we can replace V_{t_1}, V_{t_2}, S_{t_1}, and $I \equiv I_c$ in equation (1.4) to sample S_{t_2}.

1.2 Implemented method

In addition, we implemented directly (1.3) [2] to sample V_{t_2} conditional on V_{t_1}. For this simulation purpose, we use the representations of the non-central chi-squared distribution (see Johnson et al. [4] and Glasserman (2003) [5]):

$$\chi^2_\delta(\lambda) \sim \begin{pmatrix} (Z+\sqrt{\lambda})^2 + \chi^2_{\delta-1} & for\, \delta > 1, \\ \chi^2_{\delta+2N} & for\, \delta > 0, \end{pmatrix} \tag{1.5}$$

with $Z \sim \mathcal{N}(0,1)$, χ^2_δ an ordinary chi-squared distribution with δ degrees of freedom and where N is Poisson distributed with mean $\mu = \lambda/2$.

We noticed that substituting algorithms 1 and 2 for (1.5), we improve the time invested, so we keep the second implementation.

On the other hand, in $Algorithm\, 4 - Precomputation$ we do full precomputation at every node, even though it is said in [1] that doing full precomputation only at one-fourth of the nodes and calculating the values at other nodes by linear interpolation give negligible errors in their numerical tests and save part of the expensive cost in computing the modified Bessel functions.

We have fixed $v_{min} = 0.0001$, $v_{max} = 8\sigma$, and $N_u = 2^{15+ceil(log_2(4))} + 1$.

Bibliography

[1] S. T. Tse & Justin W. L. Wan, (2013) *Low-bias simulation scheme for the Heston model by Inverse Gaussian approximation*. Quantitative Finance, 13:6, 919-937.

[2] A. van Haastrecht & A. A. J. Pelsser, *Efficient, Almost Exact Simulation of the Heston Stochastic Volatility Model*. September 12, 2008.

[3] Broadie, M. and Kaya, O., *Exact simulation of stochastic volatility and other affine jump diffusion processes*. Operations Research, 2006, 54, 217–231.

[4] N.L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions, volume 2*. Wiley, second edition.

[5] P. Glasserman, *Monte Carlo Methods in Financial Engineering*. Springer Verlag, 2003.