HEC Montréal

Département des sciences de la décision

Estimation of the Heston (1993) model

September 9, 2021

Heston (1993) model

Heston (1993) proposed a stochastic volatility model where the asset price S follows

$$dS_t = \mu S_t dt + \sqrt{V_t} S_t dW_t^{\mathbb{P}}, \tag{1}$$

$$dV_t = \kappa(\theta - V_t)\, dt + \sigma\sqrt{V_t} dB_t^{\mathbb{P}}, \tag{2}$$

$W^{\mathbb{P}}$ and $B^{\mathbb{P}}$ are two correlated Brownian motions, and ρ denotes their correlation. Using Itô's lemma, the asset log-price satisfies

$$\begin{aligned} d\ln S_t &= (\mu - V_t/2)dt + \sqrt{V_t} dW_t^{\mathbb{P}}, \\ dV_t &= \kappa(\theta - V_t)\, dt + \sigma\sqrt{V_t} dB_t^{\mathbb{P}}. \end{aligned}$$

The model estimation requires a discrete time version. We use the convention that capital letters represent the continuous time model where the time t is expressed in years, and the lower case notation is for the discrete time framework in which the time t is expressed in number of periods. Let $y_t = \log S_{t\Delta} - \log S_{(t-1)\Delta}$ be the log-price variation also known as the log-return, and $v_t = V_{t\Delta}$.

$$\begin{aligned} y_{t+1} &= \mu\Delta - \frac{1}{2}\int_{t\Delta}^{(t+1)\Delta} V_s ds + \int_{t\Delta}^{(t+1)\Delta} \sqrt{V_s} dW_s^{\mathbb{P}}, \\ v_{t+1} &= v_t + \kappa\theta\Delta - \kappa\int_{t\Delta}^{(t+1)\Delta} V_s ds + \sigma\int_{t\Delta}^{(t+1)\Delta} \sqrt{V_s} dB_s^{\mathbb{P}}. \end{aligned}$$

For a daily discretization, the time step is $\Delta = 1/252$. The observed sample of returns up to time t is $y_{1:t} = (y_1, y_2, ..., y_t)$.

To use this model, we have to (i) estimate the latent (unobservable) process $\{v_t\}_{t\in\mathbb{N}}$ and (ii) estimate the model's parameters $\Theta = \{\mu, \kappa, \theta, \sigma, \rho\}$. We use the Sequential Monte-Carlo (SMC) algorithm also known as Particle Filters developed by Gordon et al. (1993).

The filtering of the latent variables requires the computation of $f(v_t|y_{1:t})$ since the filtered value is $\hat{v}_t = \int v_t f(v_t|y_{1:t}) dv_t$. The parameters are estimated by maximizing the returns likelihood function

$$f(y_{1:T};\Theta) = \prod_{t-1}^{T} f(y_{t+1}|y_{1:t};\Theta).$$

The likelihood of a new observation is

$$f(y_{t+1}|y_{1:t};\Theta) = \int f(y_{t+1}|v_t;\Theta)f(v_t|y_{1:t};\Theta)dv_t$$

where

$$\begin{aligned}
f(y_{t+1}|v_t;\Theta) &= \int f(y_{t+1}|v_{t+1},v_t;\Theta)f(v_{t+1}|v_t;\Theta)dv_{t+1}.\\
f(y_{t+1}|v_{t+1},v_t;\Theta) &\approx \frac{1}{\eta_{t+1}\sqrt{\pi}}\exp\left(-\frac{1}{2}\left(\frac{y_{t+1}-m_{t+1}}{\eta_{t+1}}\right)^2\right)\\
m_{t+1} &= \mu\Delta - \frac{IV_{t+1}}{2} + \frac{\rho}{\sigma}\left(v_{t+1}-v_t-\kappa\theta\Delta+\kappa IV_{t+1}\right),\\
\eta_{t+1} &= (1-\rho)^2 IV_{t+1}\\
IV_{t+1} &= \int_{t\Delta}^{(t+1)\Delta} V_s ds\\
f(v_t|y_{1:t};\Theta) &= f(y_t|v_t;\Theta)\int f(v_t|v_{t-1};\Theta)f(v_{t-1}|y_{1:t-1};\Theta)dv_{t-1}\frac{1}{f(y_t|y_{1:t-1})}.
\end{aligned}$$

The proof and derivations can be found in Broadie and Kaya (2006). Note that the last equation provides an iterative algorithm to compute $f(v_t|y_{1:t};\Theta)$. Indeed, the model implies that $f(v_t|v_{t-1})$ is the density function of a non-central chi-square distribution. In practice, we don't have to compute $f(y_t|y_{1:t-1})$ because it vanishes in the normalize weights step. However, the integrals involve in the first and the last lines cannot be computed analytically. Their evaluation is accomplished using Monte Carlo simulation. Finally, we need to address the computation of the integrated variance $IV_{t+1} = \int_{t\Delta}^{(t+1)\Delta} V_s ds$. To ease the computations an easy approximation for the integrated variance is

$$IV_{t+1} \approx \sum_{i=1}^{d} \frac{v_{t+i/d} + v_{t+(i-1)/d}}{2}\frac{\Delta}{d}.$$

where d represent the number of intraday steps. Usually, five intraday steps offer a good approximation. The naive SMC algorithm following Gordon et al. (1993) (Sequential Importance Resampling for uni-dimensional particles) is described as follow:

Algorithm 1 : Sequential Importance Resampling for uni-dimensional particles

1: For $k = 1 : N$, draw $\tilde{v}_{t+1}^k \sim f(v_{t+1}|v_t^k)$.

2: For $k = 1 : N$, draw $IV_{t,t+1}^k \sim f(IV_{t,t+1}|\tilde{v}_{t+1}^k, v_t^k)$.

3: For $k = 1 : N$, compute the weights $\omega_{t+1}^k = f(y_{t+1}|\tilde{v}_{t+1}^k, v_t^k, IV_{t,t+1}^k)$.

4: For $k = 1 : N$, normalize the weights

$$\pi_{t+1}^k = \frac{\omega_{t+1}^k}{\sum_{i=1}^N \omega_{t+1}^i}.$$

5: Compute the log-likelihood $L_{t+1} = \log\left(\frac{1}{N}\sum_{k=1}^N \omega_{t+1}^k\right)$

6: For $k = 1 : N$, resample N particles from a multinomial distribution with probabilities $\{\pi_{t+1}^{(k)}\}_{=i}^N$.

Because Algorithm 1 relies on Monte-Carlo simulations to approximate the integrals, a slight change in the parameter set results in large variations of the log-likelihood which prevents the use of common gradient-based optimization algorithms to estimate the parameter set. To offer a more stable log-likelihood estimation procedure, modifications of the Algorithm 1 are necessary.

This document aims at providing a comprehensive guide explaining how to (1) estimate the Heston (1993) model's parameters and (2) filter the latent variable at the same time on a conventional desktop computer in a reasonable amount of time.

Preparation: Model reparametrization

To use unconstrained numerical optimization algorithms, the optimization function optimizes a naturally transform set of parameters that can take values in $\mathbb{R}$. For the Heston (1993) model, many parameters requires to be reparametrized before the optimization process. Let $\Theta = \{\mu, \kappa, \theta, \sigma, \rho\}$ be the initial parameter set and $\tilde{\Theta} = \{\mu, \tilde{\kappa}, \tilde{\theta}, \tilde{\sigma}, \tilde{\rho}\}$ the reparametrized set. The parameters constraints and their respective choice of transformation are

- $\kappa > 0 \rightarrow \kappa = \exp(\tilde{\kappa})$
- $\theta > 0 \rightarrow \theta = \exp(\tilde{\theta})$
- $0 < \sigma \rightarrow \sigma = \exp(\tilde{\sigma})$

- $-1 \leq \rho \leq 1 \rightarrow \rho = \frac{1-\exp(-\tilde{\rho})}{1+\exp(-\tilde{\rho})}$.

Preparation: Pre-simulate the Monte-Carlo noises

Because the SMC algorithm uses Monte-Carlo simulations to approximate integrals, simulating new vectors of noises at each iteration of the optimization process adds noises to the log-likelihood estimation, which affects the performance of gradient-based optimization algorithms optimizing the log-likelihood. Indeed, gradient-based optimization algorithms measure the log-likelihood variation after slightly varying the parameter set to determine the numerical gradient. If the log-likelihood strongly varies due to the noise from Monte-Carlo simulations, the gradient-based optimization algorithm won't be able to compute adequately the numerical gradient. To circumvent this issue, simulating only one set of noises and using this same set for all the optimization iteration is absolutely necessary.

As discussed in the following sections, the noises distributions depend on the level of the model's parameters. To use the same noise vector for any parameter set, random variables following a uniform distribution are simulated and stored in memory. During each iteration step, the inverse transform sampling method is then applied to recover the desired distribution.

N.B.: Using the inverse transform sampling method is not necessary if the random variable follows a Gaussian distribution due to its scaling properties.

1 Propagating the variance

At each iteration of the SIR algorithm, the variance is propagated from v_t to v_{t+1}, for all particles. This section explains how to efficiently propagate the variance of the Heston (1993) model.

The diffusion process for the stochastic volatility process used in the Heston (1993) model is the square-root process

$$v_{t+1} = v_t + \kappa\theta\Delta - \kappa \int_{t\Delta}^{(t+1)\Delta} V_s ds + \sigma \int_{t\Delta}^{(t+1)\Delta} \sqrt{V_s} dB_s^{\mathbb{P}}$$

where κ, θ, and σ are positive constants. If the initial value $v_0 > 0$, then v_t remains strictly positive almost surely provided that $2\kappa\theta > \sigma^2$. It can be shown; see Cox et al. (1985), that the unique strong solution is positive and that the it can not be written explicitly. However, the conditional distribution of v_{t+1} given v_t is a Non-Central Chi-Square.

Discrete simulation of the square-root process

One way to simulate random paths following a square-root process is to use Euler-Maruyama scheme

$$v_{t+1} \approx v_t + \kappa(\theta - v_t)\Delta + \sigma\sqrt{v_t}\sqrt{\Delta}Z_t$$

where $Z \sim \mathcal{N}(0,1)$. This approximation will inevitably create errors. IN particular, it might produce negative variances. For small time steps Δ, these errors are negligible. However, intra-day simulation of the variance process to obtain small time steps makes larger noise vector that can rapidly overflow the computers memory if the number of particles is high or the number of observation is large.

Exact simulation

Another way of simulating paths from a square-root process is by directly generating Noncentral Chi-Square random variables.

Simulating Noncentral ChiSquare random variables

Some high-level programming languages such as MATLAB and PYTHON include built-in functions to simulate Noncentral Chi-Square random variables. However, these functions do not use the same noises when varying the parameters of the distribution. As previously explained, using the same noises is of paramount importance when trying to smooth the log-likelihood function with SMC algorithms. In this section, we will describe ways to simulate Noncentral Chi-Square random variables.

A random variable $X_{\alpha,\beta}$ follows a Noncentral Chi-Square distribution with α degrees of freedom and a Noncentrality parameter λ if it's cumulative distribution function is

$$F_{X_{\alpha,\beta}}(y) = e^{-\frac{\lambda}{2}} \sum_{j=0}^{\infty} \left(\frac{\frac{\lambda}{2}^j}{j! 2^{\frac{\alpha}{2}+j} \Gamma\left(\frac{\alpha}{2}+j\right)} \int_0^y z^{\frac{\alpha}{2}+j-1} e^{-\frac{z}{2}} dz \right) \mathbb{1}_{y>0}$$

where $\mathbb{1}$ is the indicator function. Within Heston's framework,

$$v_{t+1} = \frac{\sigma^2 \left(1 - e^{-\kappa\Delta}\right)}{4\kappa} \chi^2_{\alpha,\lambda_t}$$

where $\alpha = \frac{4\kappa\theta}{\sigma^2}$ and $\lambda_t = \frac{4\kappa e^{-\kappa\Delta}}{\sigma^2(1-e^{-\kappa\Delta})} v_t$. As explained in Glasserman (2003) (p. 123) it is possible to easily simulate $X_{\alpha,\lambda}$ for different scenarios.

1. If α is an integer ≥ 1 and $\lambda = 0$ then
   $X_{\alpha,0} = \sum_{i=1}^{\alpha} Z_i^2$ is a Chi-Square distribution with α degrees of freedom, where $Z_i \ \forall i \in \{1, 2, ...\alpha\}$ are i.i.d. standard Gaussian random variables.

2. If α is an integer ≥ 1 and $\lambda > 0$ then
   $X_{\alpha,\lambda} = \sum_{i=1}^{\alpha} (Z_i + a_i)^2$ is a Noncentral Chi-Square distribution with α degrees of freedom and a Noncentrality parameter of $\lambda = \sum_{i=1}^{\alpha} a_i^2$, where $Z_i \ \forall i \in \{1, 2, ...\alpha\}$ are i.i.d. standard Gaussian random variables and $a_i \ \forall i \in \{1, 2, ...\alpha\}$ are constants.

3. If $\alpha > 1$, then $X_{\alpha,\lambda} = X_{1,\lambda} + X_{\alpha-1,0}$ provided that $X_{1,\lambda}$ and $X_{\alpha-1,\lambda}$ are independent. Note that $X_{1,\lambda} = \left(Z + \sqrt{\lambda}\right)^2$ where $Z \sim \mathcal{N}(0, 1)$ and is independent of $X_{\alpha-1,0}$.

4. For any $\alpha > 0$, we may sample $X_{\alpha,\lambda}$ by generating a Poisson random variable N with mean $\lambda/2$ and then sample $X_{\alpha,\lambda}$ from a chi-square random variable with $\alpha + 2N$ degrees of freedom. For a detailed proof of the result refer to Glasserman (2003) (p. 123).

For the Heston (1993) model on a normal computer without a GPU the Euler-Maruyama sampling method is the most adequate and the fastest if the number of particles is not too large.

2 Smoothing the resampling distribution

Estimating the model's parameters using a gradient-based optimization function still necessitates to further adjusts Algorithm 1. In fact, a slight change of the parameter set can induce large variations of the log-likelihood even though the same vector of noises is used at each optimization iteration. This issue has been documented in Hürzeler and Künsch (2001), Malik and Pitt (2011) and Creal (2012).

Malik and Pitt (2011) addresses this issue by proposing to smooth the discrete resampling cumulative distribution. The smoothed and continuous cumulative filtering density function stabilizes the

log-likelihood estimation when the model's parameter set varies. Optimization algorithms are then able to compute numerical gradients to optimize the log-likelihood.

Figure 1 displays graphically the smoothing procedure. Each particle is sorted in ascending order. Using the particles associated normalized weights π_{t+1}^k where

$$\pi_{t+1}^k = \frac{\omega_{t+1}^k}{\sum_{i=1}^N \omega_{t+1}^i},\ \omega_{t+1}^k = f(\boldsymbol{y}_{t+1}|\tilde{v}_{t+1}^k, v_t^k, IV_{t+1}^k)$$

a discrete cumulative distribution is generated. This distribution is portrayed by the red line in the top panel of Figure 1. The discrete cumulative distribution is then smoothed using an interpolation scheme. The smoothed distribution is represented by the black line in the top panel of Figure 1. Resampling the particles with the inverse transform sampling method using the smoothed continuous cumulative distribution function stabilizes the particles resampling procedure by ensuring that a small variation in the particles weight results in a small variation in the set of resampled particles. Otherwise small variations of particles weights produce a larger corresponding variation of the resample particle set. These larger variations of the resampled particle set then impact the log-likelihood estimation stability.

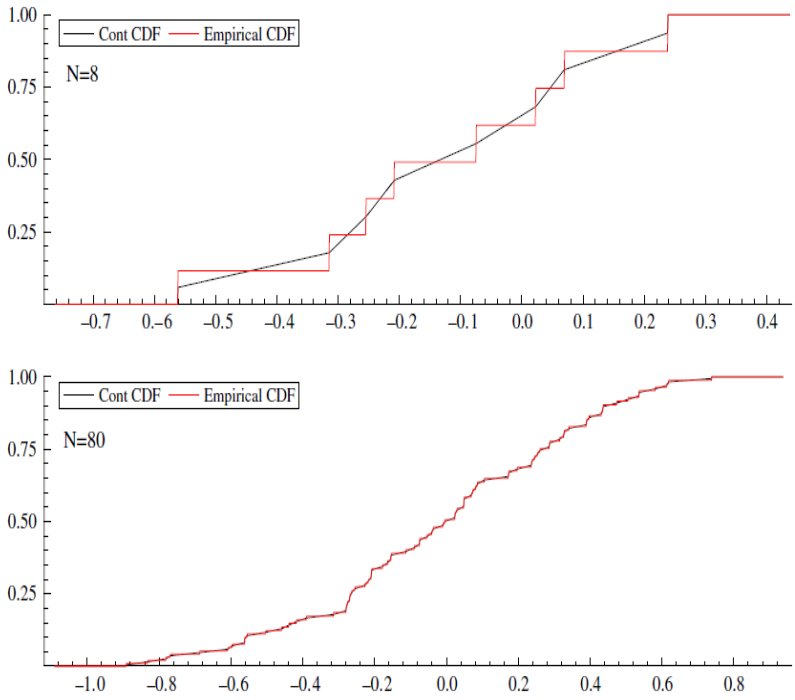

Fig. 1. The empirical cumulative distribution function and the approximating linear interpolation for Gaussian samples with a volatility measurement density assigned to the weights. Top: $N = 8$. Bottom: $N = 80$.

Figure 1: **Malik and Pitt continuous resampling distribution.**

Combining the blocks

Algorithm 2 summarizes the steps to adequately estimate the model's parameters using a gradient-based optimization function. Step 1 of Algorithm 2, propagating the variance, is explained in Section 1. Step 2 of Algorithm 2, is explained in Section **??** and relies on simulating Inverse Gaussian random variables to increase the computational speed. Step 3 to 7 of Algorithm 2, resampling the particles from a continuous cumulative distribution, is explained in Section 2, and stabilizes the log-likelihood estimation during the optimization procedure.

Algorithm 2 : Continuous Sequential Importance Resampling for uni-dimensional particles

1: For $k = 1 : N$ draw $\tilde{v}_{t+1}^k \sim f(v_t|v_t^k)$.

2: For $k = 1 : N$ draw $IV_{t+1}^k \sim f(IV|\tilde{v}_{t+1}^k, v_t^k)$.

3: $\tilde{v}_{t+1} = \text{sort}(\tilde{v}_{t+1})$.

4: For $k = 1 : N$, compute the weights $\omega_{t+1}^k = f(y_{t+1}|\tilde{v}_{t+1}^k, v_t^k, IV_{t,t+1}^k)$.

5: For $k = 1 : N$, normalize the weights

$$\pi_{t+1}^k = \frac{\omega_{t+1}^k}{\sum_{i=1}^N \omega_{t+1}^i}.$$

6: Compute the log-likelihood $L_{t+1} = \ln\left(\frac{1}{N}\sum_{k=1}^N \omega_{t+1}^k\right)$.

7: Compute the discrete filtering distribution $f(v_{t+1}|y_{1:t+1})$ using the normalized weights π_{t+1}^k.

8: Smooth the discrete filtering distribution $f(v_{t+1}|y_{1:t+1})$ using the Malik and Pitt (2011) smoothing procedure and generate a continuous cumulative resampling distribution $\tilde{f}(v_{t+1}|y_{1:t+1})$.

9: For $k = 1 : N$ draw $v_{t+1}^k \sim \tilde{f}(v_{t+1}|y_{1:t+1})$.

3 Integrating the Realized Variance

The previous sections discussed the steps and the methodology to (1) optimize the model's parameters and (2) filter the latent variable (variance) of the Heston (1993) model using a single source of information: the log-returns. Adding an observation that is related to the asset's volatility would significantly improve the estimation precision. A common practice (see: Barndorff-Nielsen and Shephard (2002), Barndorff-Nielsen and Shephard (2004), Zhang et al. (2005)) consist of estimating the volatility from intra-day squared log-returns. This new observation related to the volatility is known as the Realized Variance (RV) and is computed as follow:

$$RV = \sum_i (y_{t_{i+1}} - y_{t_i})^2 \approx \int_{t\Delta}^{(t+1)\Delta} V_s ds = IV_{t+1}.$$

where y_{t_i} are intraday log-returns, that is $t_i - t_{i-1} = \frac{\Delta}{n}$ where n is the number daily subintervals. To sample every 5 minutes of the 6.5 trading hours, $n = 6.5 \times 12 = 78$.

The next step consists of integrating this new source of information into the SMC algorithm. We

assume that the relative errors are iid Gaussian random variables, that is,

$$\frac{IV_t - RV_t}{RV_t} \sim \mathcal{N}\left(0, \eta^2\right).$$

Now that the distribution of RV has been determined, the particles weight can be computed using both the log-returns and the RV by assuming the independence between the two variable distributions

$$f(y_{t+1}, RV_{t+1}|\tilde{v}_{t+1}^k, v_t^k, IV_{t,t+1}^k) = f(y_{t+1}|\tilde{v}_{t+1}^k, v_t^k, IV_{t,t+1}^k) f(RV_{t+1}|\tilde{v}_{t+1}^k, v_t^k, IV_{t,t+1}^k).$$

Realized variance constitute one of many possible sources of information that can be integrated. The financial literature explored a plethora of sources of observable (see for example: Barndorff-Nielsen and Shephard (2004), Bardgett et al. (2019),Amaya et al. (2021)). Adding new sources of information allows (1) the estimation of more complex and adequate stochastic processes for the log-returns dynamic, (2) generates more precise parameter estimates and (3) a more precise estimation of the latent variables.

References

Amaya, D., Bégin, J.-F., and Gauthier, G. (2021). The informational content of high-frequency option prices. *Management Science*.

Bardgett, C., Gourier, E., and Leippold, M. (2019). Inferring volatility dynamics and risk premia from the s&p 500 and vix markets. *Journal of Financial Economics*, 131(3):593–618.

Barndorff-Nielsen, O. E. and Shephard, N. (2002). Estimating quadratic variation using realized variance. *Journal of Applied Econometrics*, 17(5):457–477.

Barndorff-Nielsen, O. E. and Shephard, N. (2004). Power and bipower variation with stochastic volatility and jumps. *Journal of Financial Econometrics*, 2(1):1–37.

Broadie, M. and Kaya, Ö. (2006). Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, 54(2):217–231.

Cox, J. C., Ingersoll Jr, J. E., and Ross, S. A. (1985). An intertemporal general equilibrium model of asset prices. *Econometrica: Journal of the Econometric Society*, pages 363–384.

Creal, D. (2012). A survey of sequential monte carlo methods for economics and finance. *Econometric Reviews*, 31(3):245–296.

Glasserman, P. (2003). *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F-radar and signal processing*, volume 140, pages 107–113. IET.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343.

Hürzeler, M. and Künsch, H. R. (2001). Approximating and maximising the likelihood for a general state-space model. In *Sequential Monte-Carlo Methods in Practice*, pages 159–175. Springer.

Malik, S. and Pitt, M. K. (2011). Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, 165(2):190–209.

Zhang, L., Mykland, P. A., and Aït-Sahalia, Y. (2005). A tale of two time scales: Determining integrated volatility with noisy high-frequency data. *Journal of the American Statistical Association*, 100(472):1394–1411.