

# Heston Model

Graham Annett

13 December 2013

## Introduction

In finance, the Heston model, describes the evolution of volatility of an underlying asset. The Heston model is a stochastic volatility model. The model assumes that the volatility of the asset follows a random process, or random walk.

The Heston Model has five independent parameters, all of which can be determined by calibrating to the market-observed prices of European options of various strikes or maturity dates. Once a set of parameters has been determined, you can price other options (parameters are calibrated). Other European options can be priced or American options. The underlying asset price follows a lognormal process, the variance (V) follows a mean-reverting square root process:

$$dS_t = rS_t dt + \sqrt{V} S_t dZ_1$$
$$dV = -\kappa(V - V_\infty)dt + \omega\sqrt{V} dZ_2$$

where:

r is the risk-free interest rate

dZ<sub>1</sub> and dZ<sub>2</sub> are two correlated standard Brownian motions

5 Parameters are:

V<sub>0</sub>=initial variance

V<sub>∞</sub>= long-run variance

κ = speed of mean reversion

ω = volatility of volatility

ρ = correlation

In General, the price at time (t) of a European call options maturity date at time (T) is given by the discounted expected value:

$$C_T = e^{-r(T-t)} \int_0^\infty (e^x - K) + p(x) dx$$

Where Log[s] = x

And P(x) is the probability density function of the underlying logarithmic asset price.

## Code

For this, we use the mathematica function Ito Process to allow for the “randomness” that the Wiener process is. This part is the “cW” equation we use. The 2 differential equation's are “solved” in the hestonmodel variable and then the output is generated by running

```

hestonmodel = ItoProcess[{
  ds[t]==μ s[t] dt+Sqrt[r[t]] s[t] dw_s[t],
  dr[t]==κ (θ-r[t]) dt+ξ Sqrt[r[t]] dw_v[t]},
  {s[t],r[t]},{{s,r},{s_0,r_0}},t,{w_s,w_v}∈cW[ρ]];

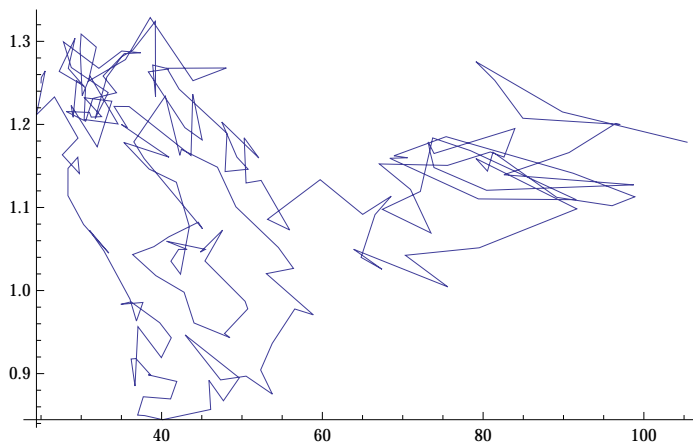
cW[ρ_]:=ItoProcess[{{0,0},IdentityMatrix[2]},{{w1,w2},{0,0}},t,{{1,ρ},{ρ,1}}];

output= RandomFunction[
  hestonmodel/.{μ→0,κ→2,θ→1,ξ→1/2,ρ→-1/3,s_0→25,Subscript[r,0]→1.25},
  {0,1,0.005},1,Method→"StochasticRungeKutta"];

```

From here, we will plot what the random process looks like:

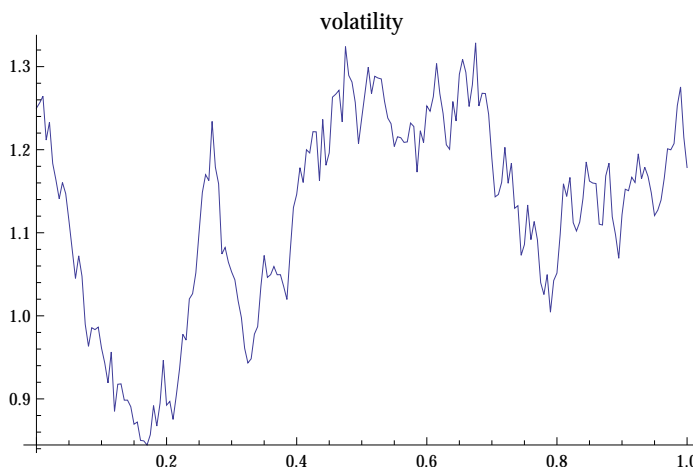
```
ListLinePlot[output["PathStates", 1]]
```



This plot is a Wiener Process that we then project onto the data for something such as an asset. We do this by showing how the volatility fluctuates due to the change in price (here we act as if the x axis were the price of the asset, it is not a time series) and the volatility is then taken from the y axis.

From here we can plot it as if it were a time series by not including the price since the Heston model is more interested in understanding the underlying volatility of some asset. You can follow the initial random process here; see how it follows the previous plot but this plot is “time series” in nature.

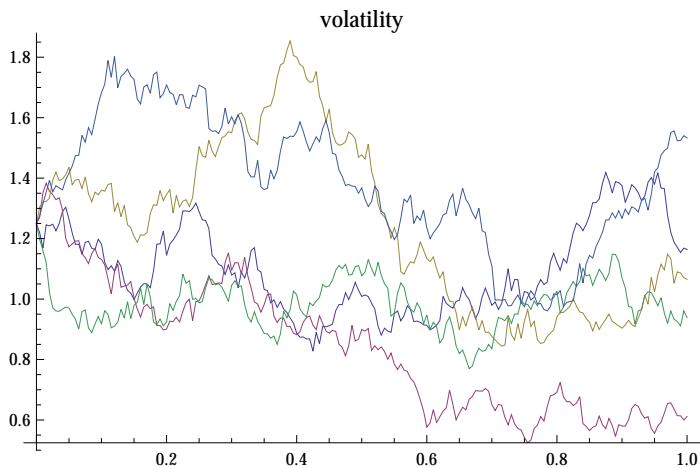
```
ListLinePlot[output["PathComponent", 2], PlotLabel -> "volatility"]
```



Since we are not limited to plotting to 1 asset, or even one possible outcome of an asset below we will plot 5 and show how they vary amongst themselves.

```
output = RandomFunction[
hestonmodel /.
  { $\mu \rightarrow 0$ ,  $\kappa \rightarrow 1$ ,  $\theta \rightarrow 1$ ,  $\xi \rightarrow 1/2$ ,  $\rho \rightarrow -1/3$ ,  $s_0 \rightarrow 25$ , Subscript[r, 0]  $\rightarrow 1.25$ },
  {0, 1, 0.005}, 5, Method  $\rightarrow$  "StochasticRungeKutta"];

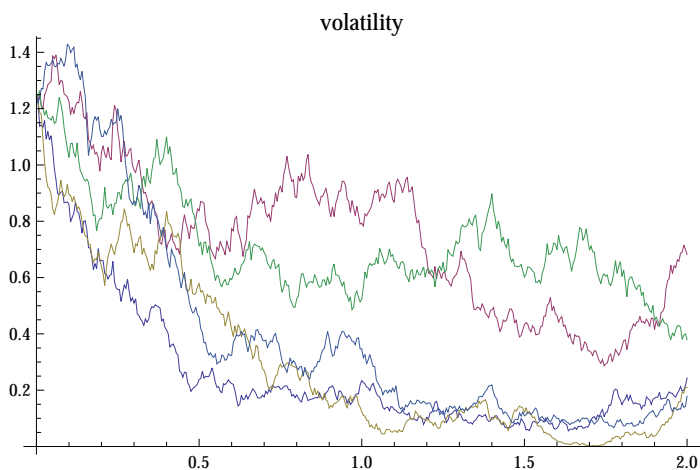
ListLinePlot[output["PathComponent", 2], PlotLabel  $\rightarrow$  "volatility"]
```



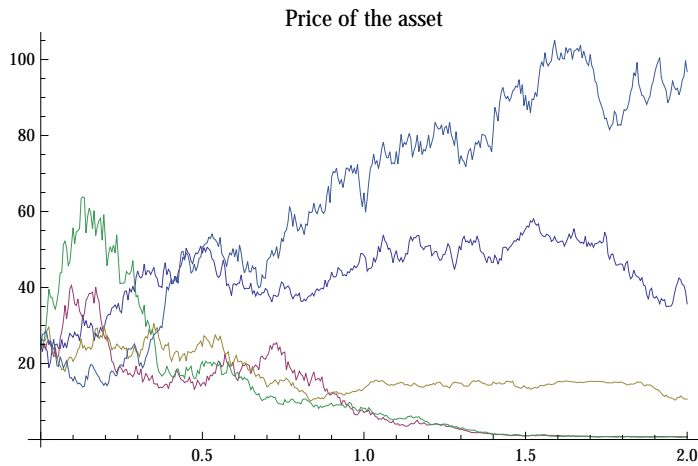
From here we can start to further understand the parameters, for instance, as  $\theta$  tends to 0, the volatility tends 0 as well. This intuitively makes sense as the long run variance is 0, then an assets price would become more stable as future expected price would be within a much smaller predicted range.

```
output = RandomFunction[
hestonmodel /.
  { $\mu \rightarrow 0$ ,  $\kappa \rightarrow 1$ ,  $\theta \rightarrow 0$ ,  $\xi \rightarrow 1/2$ ,  $\rho \rightarrow -1/3$ ,  $s_0 \rightarrow 25$ , Subscript[r, 0]  $\rightarrow 1.25$ },
  {0, 2, 0.005}, 5, Method  $\rightarrow$  "StochasticRungeKutta"];

ListLinePlot[output["PathComponent", 2], PlotLabel  $\rightarrow$  "volatility"]
```



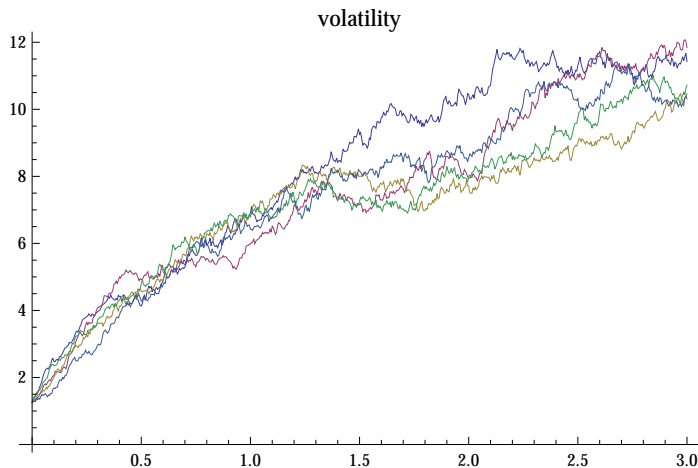
```
ListLinePlot[output["PathComponent", 1], PlotLabel → "Price of the asset"]
```



This follows for any number, as  $\theta$  tends to some  $\alpha$  in the long run the volatility will reach this level and the only volatility will be due to the random process, not long run expectations of an assets variance. Example, if the long run volatility was 10.

```
output = RandomFunction[
hestonmodel /.
  { $\mu \rightarrow 0$ ,  $\kappa \rightarrow 1$ ,  $\theta \rightarrow 10$ ,  $\xi \rightarrow 1/2$ ,  $\rho \rightarrow -1/3$ ,  $s_0 \rightarrow 25$ , Subscript[r, 0]  $\rightarrow 1.25$ },
  {0, 3, 0.005}, 5, Method → "StochasticRungeKutta"];
```

```
ListLinePlot[output["PathComponent", 2], PlotLabel → "volatility"]
```



Another interesting aspect to note is how  $\xi$ , the volatility of the volatility, controls the amount of "randomness" that the volatility will have. For instance when  $\xi$  tends to 0, the volatility will become "stable" at the level of long term volatility. This makes sense from the equations as the process evolves the WienerProcess will have less and less of an effect.

```

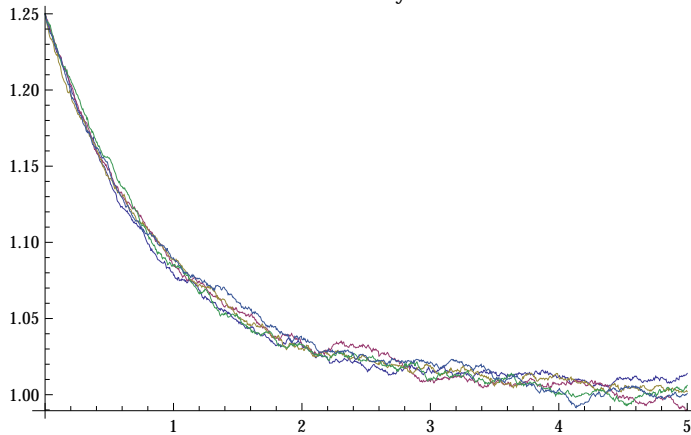
output = RandomFunction[
hestonmodel /.
  { $\mu \rightarrow 0$ ,  $\kappa \rightarrow 1$ ,  $\theta \rightarrow 1$ ,  $\xi \rightarrow .01$ ,  $\rho \rightarrow -1/3$ ,  $s_0 \rightarrow 25$ , Subscript[r, 0]  $\rightarrow 1.25$ },
  {0, 5, 0.005}, 5, Method  $\rightarrow$  "StochasticRungeKutta"];

```

```

ListLinePlot[output["PathComponent", 2], PlotLabel  $\rightarrow$  "volatility"]

```



This model has other extensions as well though, for instance the Chen Model is very similar and is used to model the evolution of interest rates. Using the same initial equation for  $dr[t]$ , we extend it to 3 different differential equations.

```

dr[t] == ( $\theta_1 - r[t]$ ) dt +  $\xi$  Sqrt[r[t]] dw_v[t]
da[t] == ( $\theta_2 - a[t]$ ) dt +  $\xi$  Sqrt[a[t]] dw_v[t]
dp[t] == ( $\theta_3 - p[t]$ ) dt +  $\xi$  Sqrt[p[t]] dw_v[t]

```

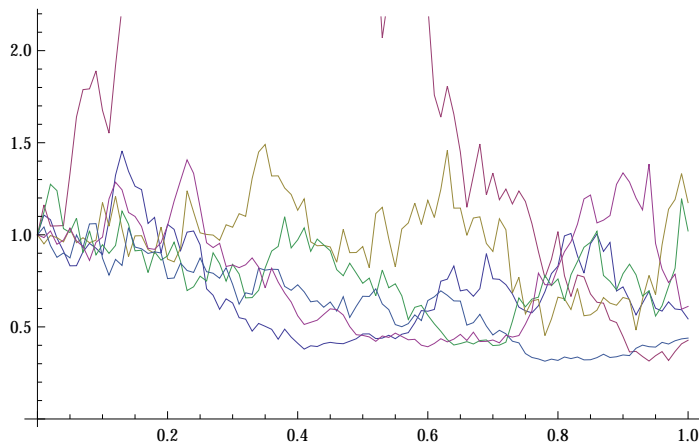
```

chenmodel[ $\theta_1$ _,  $\theta_2$ _,  $\theta_3$ _,  $\xi$ _, {r0_, a0_, p0_}] := ItoProcess[{
  dr[t] == ( $\theta_1 - r[t]$ ) dt + p[t] Sqrt[r[t]] dw[t],
  da[t] == ( $\theta_2 - a[t]$ ) dt + p[t] Sqrt[a[t]] dw[t],
  dp[t] == ( $\theta_3 - p[t]$ ) dt +  $\xi$  Sqrt[p[t]] dw[t]},
  r[t], {{r, a, p}, {r0, a0, p0}}, t, w  $\in$  WienerProcess[]]

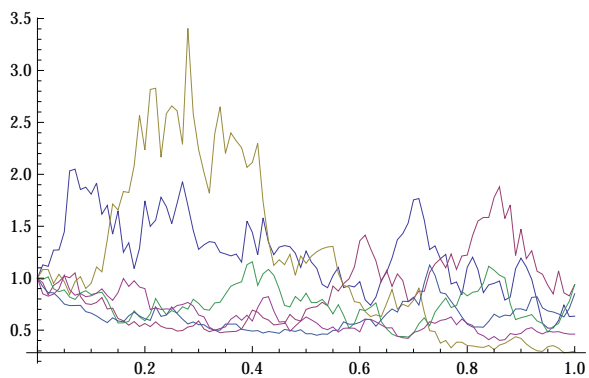
```

Which we can then use to show the volatility of the interest rates rather than a super generalized asset.

```
ListLinePlot[RandomFunction[chenmodel[1, 2, 1.2, 1, {1, 1, 1}],
{0, 1, 0.01}, 6, Method → {"StochasticRungeKutta"}]]
```



```
ListLinePlot[
RandomFunction[chenmodel[1, 2, 1.2, 1, {1, 1, 1}], {0, 1, 0.01}, 6, Method →
{"StochasticRungeKutta", "ProjectionFunction" → Function[{t, xvec}, Abs[xvec]]],
PlotRange → All, ImageSize → 300]
```



## References

Heston, S. (1993) 'A closed-form solution for options with stochastic volatility', Review of Financial Studies, 6, 327-43.

"The Heston Model of Stochastic Volatility: Fast Option Pricing and Accurate Calibration." FinCad. Web. 13 December 2013. <<http://www.fincad.com/derivatives-resources/articles/heston-model-stochastic-volatility.aspx#ref3>>.

Followed Examples on *Mathematica* Tutorials as well to understand how these models are implemented and what is happening.