

# MTH 9875 The Volatility Surface: Fall 2017

## Lecture 3: The Heston Model

Jim Gatheral

Department of Mathematics



## Outline of lecture 3

- The Heston process
- Derivation of the Heston formula for European options
- The Heston characteristic function
- Implementation of the Heston model in R
- Numerical experiments
- Simulation of the Heston model

## The Heston process

The Heston stochastic volatility model may be written as

(1)

$$dS_t = \mu_t S_t dt + \sqrt{v_t} S_t dZ_t$$

and

(2)

$$dv_t = -\lambda (v_t - \bar{v}) dt + \eta \sqrt{v_t} dW_t$$

with  $\mathbb{E} [dZ_t dW_t] = \rho dt$  and where  $\lambda$  is the speed of reversion of  $v_t$  to its long term mean  $\bar{v}$ .

- The process followed by the instantaneous variance  $v_t$  may be recognized as a version of the square root process or CIR process of [Cox, Ingersoll, Ross]<sup>[3]</sup>.
- It is a (jump-free) special case of a so-called *affine jump diffusion* (AJD).
  - Roughly speaking a jump-diffusion process for which the drifts and covariances and jump intensities are linear in the state vector (which is  $\{x, v\}$ ) in this case with

$$x = \log(S).$$

- [Duffie, Pan and Singleton]<sup>[4]</sup> show that AJD processes are analytically tractable in general.
  - The solution technique involves computing an "extended transform" which in the Heston case is a conventional Fourier transform.

## Definition

The following explanation is due to Martin Keller-Ressel:

An *affine process* can be described as a Markov process whose log-characteristic function is an affine function of its initial state vector.

And here's a definition of the word *affine* from Wikipedia:

In geometry, an affine transformation or affine map or an affinity (from the Latin, *affinis*, "connected with") between two vector spaces (strictly speaking, two affine spaces) consists of a linear transformation followed by a translation:

$$x \mapsto Ax + b$$

## The Heston valuation equation

In the Heston model, the value  $V$  of a European-style claim is a function of  $S$  and  $v$  only. An application of Itô's Lemma gives

(3)

$$dV = \partial_t V dt + \partial_S V dS + \partial_v V dv + \frac{1}{2} \partial_{S,S} V dS^2 + \frac{1}{2} \partial_{v,v} V dv^2 + \partial_{S,v} V dv dS$$

$V(S, v, t) = \mathbb{E}[(S_T - K)^+ | S, v]$  is a martingale so taking expectations (and again with zero rates and dividends), we obtain the valuation equation:

(4)

$$\frac{\partial V}{\partial t} + \frac{1}{2} v S^2 \frac{\partial^2 V}{\partial S^2} + \rho \eta v S \frac{\partial^2 V}{\partial v \partial S} + \frac{1}{2} \eta^2 v \frac{\partial^2 V}{\partial v^2} - \lambda(v - \bar{v}) \frac{\partial V}{\partial v} = 0$$

## Risk-neutral and statistical (physical) measures

- In Heston's original paper, the price of risk is assumed to be linear in the instantaneous variance  $v$  in order to be able to retain the form of the equation under the transformation from the statistical (or real) measure to the risk-neutral measure.
- In contrast, we assume that the Heston process with parameters fitted to option prices generates the risk-neutral measure.

- We may set the market price of volatility risk to zero.
- Since we are only interested in pricing, and we assume that the pricing measure is recoverable from European option prices, we are indifferent to the statistical measure.

## A simpler form of the valuation equation

Before solving equation (4) with the appropriate boundary conditions, we can simplify it by making some suitable changes of variable.

Let  $K$  be the strike price of the option,  $S_T$  the stock price at expiration  $T$  and  $x := \log(S_T/K)$ . We denote the time to expiration by  $\tau := T - t$ .

Then equation (4) simplifies to

$$(5) \quad -\frac{\partial C}{\partial \tau} + \frac{1}{2} v C_{11} - \frac{1}{2} v C_1 + \frac{1}{2} \eta^2 v C_{22} + \rho \eta v C_{12} - \lambda(v - \bar{v}) C_2 = 0$$

where the subscripts 1 and 2 refer to differentiation with respect to  $x$  and  $v$  respectively.

## Form of the solution

According to [Duffie, Pan and Singleton]<sup>[5]</sup>, the solution of equation (5) has the form

$$(6) \quad C(x, v, \tau) = K \{ e^x P_1(x, v, \tau) - P_0(x, v, \tau) \}$$

where, exactly as in the Black-Scholes formula,

- the first term in the brackets represents the pseudo-expectation of the final index level given that the option is in-the-money
- the second term represents the pseudo-probability of exercise.

## Equations for pseudo-probabilities

Substituting the proposed solution (6) into equation (5) implies that  $P_0$  and  $P_1$  must satisfy the equation

$$(7) \quad \left. \begin{aligned} -\frac{\partial P_j}{\partial \tau} + \frac{1}{2} v \frac{\partial^2 P_j}{\partial x^2} - \left( \frac{1}{2} - j \right) v \frac{\partial P_j}{\partial x} + \frac{1}{2} \eta^2 v \frac{\partial^2 P_j}{\partial v^2} \\ + \rho \eta v \frac{\partial^2 P_j}{\partial x \partial v} + (a - b_j v) \frac{\partial P_j}{\partial v} = 0 \end{aligned} \right|$$

for  $j = 0, 1$  where  $a = \lambda \bar{v}$ ,  $b_j = \lambda - j \rho \eta$

The terminal condition is

(8)

$$\lim_{\tau \rightarrow 0} P_j(x, v, \tau) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \\ := \theta(x)$$

## Solve using Fourier transforms

We solve equation (7) subject to the condition (8) using a Fourier transform technique. To this end define the Fourier transform of  $P_j$  through

$$\tilde{P}(u, v, \tau) = \int_{-\infty}^{\infty} dx e^{-i u x} P(x, v, \tau)$$

Then

$$\tilde{P}(u, v, 0) = \int_{-\infty}^{\infty} dx e^{-i u x} \theta(x) = \frac{1}{i u}$$

The inverse transform is given by

(9)

$$P(x, v, \tau) = \int_{-\infty}^{\infty} \frac{du}{2\pi} e^{i u x} \tilde{P}(u, v, \tau)$$

## Equation in Fourier space

Substituting this into equation (7) gives

(10)

$$\begin{aligned} -\frac{\partial \tilde{P}_j}{\partial \tau} - \frac{1}{2} u^2 v \tilde{P}_j - \left( \frac{1}{2} - j \right) i u v \tilde{P}_j \\ + \frac{1}{2} \eta^2 v \frac{\partial^2 \tilde{P}_j}{\partial v^2} + \rho \eta i u v \frac{\partial \tilde{P}_j}{\partial v} + (a - b_j v) \frac{\partial \tilde{P}_j}{\partial v} = 0 \end{aligned}$$

Now define

$$\begin{aligned} \alpha &= -\frac{u^2}{2} - \frac{i u}{2} + i j u \\ \beta &= \lambda - \rho \eta j - \rho \eta i u \\ \gamma &= \frac{\eta^2}{2} \end{aligned}$$

Then equation (10) becomes

(11)

$$v \left\{ \alpha \tilde{P}_j - \beta \frac{\partial \tilde{P}_j}{\partial v} + \gamma \frac{\partial^2 \tilde{P}_j}{\partial v^2} \right\} + a \frac{\partial \tilde{P}_j}{\partial v} - \frac{\partial \tilde{P}_j}{\partial \tau} = 0$$

Now substitute (the *affine ansatz*):

$$\begin{aligned} \tilde{P}_j(u, v, \tau) &= \exp \{ C(u, \tau) \bar{v} + D(u, \tau) v \} \tilde{P}_j(u, v, 0) \\ &= \frac{1}{i u} \exp \{ C(u, \tau) \bar{v} + D(u, \tau) v \} \end{aligned}$$

It follows that

$$\begin{aligned} \frac{\partial \tilde{P}_j}{\partial \tau} &= \left\{ \bar{v} \frac{\partial C}{\partial \tau} + v \frac{\partial D}{\partial \tau} \right\} \tilde{P}_j \\ \frac{\partial \tilde{P}_j}{\partial v} &= D \tilde{P}_j \\ \frac{\partial^2 \tilde{P}_j}{\partial v^2} &= D^2 \tilde{P}_j \end{aligned}$$

Then equation (11) is satisfied if

(12)

$$\begin{aligned} \frac{\partial C}{\partial \tau} &= \lambda D \\ \frac{\partial D}{\partial \tau} &= \alpha - \beta D + \gamma D^2 \\ &= \gamma (D - r_+)(D - r_-) \end{aligned}$$

where we define

$$r_{\pm} = \frac{\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\gamma} =: \frac{\beta \pm d}{\eta^2}$$

Integrating (12) with the terminal conditions  $C(u, 0) = 0$  and  $D(u, 0) = 0$  gives

(13)

$$\begin{aligned} D(u, \tau) &= r_- \frac{1 - e^{-d\tau}}{1 - g e^{-d\tau}} \\ C(u, \tau) &= \lambda \left\{ r_- \tau - \frac{2}{\eta^2} \log \left( \frac{1 - g e^{-d\tau}}{1 - g} \right) \right\} \end{aligned}$$

where we define

$$g := \frac{r_-}{r_+}$$

## The Heston formula

We then arrive at our final result:

### The Heston formula

$$C(x, v, \tau) = K \{ e^x P_1(x, v, \tau) - P_0(x, v, \tau) \}$$

where the pseudo-probabilities  $P(\cdot)$  are obtained by inverting the  $\tilde{P}(\cdot)$  using (9):

$$(14) \quad P_j(x, v, \tau) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty du \operatorname{Re} \left\{ \frac{\exp\{C_j(u, \tau) \bar{v} + D_j(u, \tau) v + i u x\}}{i u} \right\}$$

This integration of a real-valued function may be performed using standard numerical methods.

### Heston greeks

It is worth noting that taking derivatives of the Heston formula with respect to  $x$  or  $v$  in order to compute delta, gamma, vega and so on is extremely straightforward because the functions  $C(u, \tau)$  and  $D(u, \tau)$  are independent of  $x$  and  $v$ .

### A digression: the complex logarithm in the integration (14)

In Heston's original paper and in most other papers on the subject,  $C(u, \tau)$  is written (almost) equivalently as

$$(15) \quad C(u, \tau) = \lambda \left\{ r_+ \tau - \frac{2}{\eta^2} \log \left( \frac{e^{+d\tau} - g}{1 - g} \right) \right\}$$

The reason for the qualification "almost" is that this definition coincides with our previous one only if the imaginary part of the complex logarithm is chosen so that  $C(u, \tau)$  is continuous with respect to  $u$ .

It turns out that taking the principal value of the logarithm in (15) causes  $C(u, \tau)$  to jump discontinuously each time the imaginary part of the argument of the logarithm crosses the negative real axis.

### BCC parameters

[Bakshi, Cao and Chen]<sup>[2]</sup> found (more or less) the following Heston (SV) parameters in a fit of the Heston model to historical volatility surfaces:

Table 1: BCC parameters

$\eta$	0.04
$\bar{\eta}$	0.04
$\lambda$	1.15
$\eta$	0.39
$\rho$	-0.64

## A numerical example with BCC parameters

Recall the two versions of  $C(u, \tau)$  from (13) and (15) respectively:

$$C^{JG}(u, \tau) = \lambda \left\{ r_- \tau - \frac{2}{\eta^2} \log \left( \frac{1 - g e^{-d \tau}}{1 - g} \right) \right\}$$

and

$$C^{Heston}(u, \tau) = \lambda \left\{ r_+ \tau - \frac{2}{\eta^2} \log \left( \frac{e^{+d \tau} - g}{1 - g} \right) \right\}$$

We code these below.

```
In [1]: bigC <- function(params){

  lambda <- params$lambda
  rho <- params$rho
  eta <- params$eta
  vbar <- params$vbar
  v <- params$v

  fj <- function(u, t, j){

    al <- -u*u/2 - li*u/2 + li*j*u
    bet <- lambda - rho*eta*li*u-rho*eta*j
    gam <- eta^2/2
    d <- sqrt(bet*bet - 4*al*gam)
    rp <- (bet + d)/(2*gam)
    rm <- (bet - d)/(2*gam)
    g <- rm / rp
    C.JG <- lambda * (rm * t - 2/eta^2 * log( (1 - g*exp(-(d*t)))/(1 - g)))
    C.Heston <- lambda * (rp * t - 2/eta^2 * log( (exp(d*t) - g)/(1 - g)))
    return(list(C.JG=C.JG,C.Heston=C.Heston))

  }

}
```

Check the two versions with  $\tau = 3$ , and with BCC parameters:

```
In [2]: paramsBCC <- list(lambda = 1.15, rho = -0.64, eta = 0.39, vbar = 0.04, v = .04)
```

```
In [3]: bigC(paramsBCC)(u=10,t=3,j=0)
```

**\$C.JG**

-47.5424985325402+26.838817444334i

**\$C.Heston**

-47.54249853254+121.850824061777i

The imaginary parts are very different. Which function works better?

## Comparison plot

```
In [4]: library(repr)
options(repr.plot.height=5)
```

```
In [5]: u.vec <- seq(0,10,.01)
x <- bigC(paramsBCC)(u=u.vec,t=3,j=0)
```

```
In [6]: plot(Re(x$C.Heston),Im(x$C.Heston),type="l",col="red",lty=2,lwd=2,xlab="Re(C)
lines(Re(x$C.JG),Im(x$C.JG),type="l",col="green4",lwd=1)
```

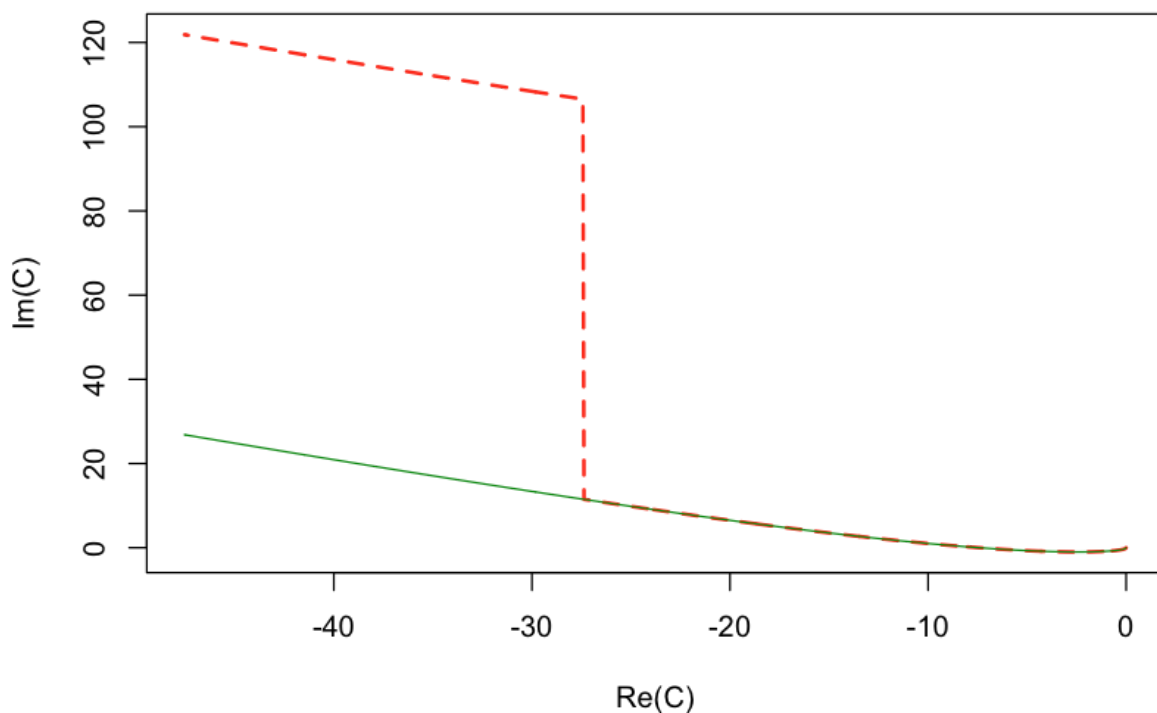


Figure 1:  $C^{\text{Heston}}(\cdot)$  in red,  $C^{\text{JG}}(\cdot)$  in green.

## Solutions in the literature



One conventional resolution is to keep careful track of the winding number in the integration (14) so as to remain on the same Riemann sheet.

This leads to practical implementation problems because standard numerical integration routines cannot be used. The paper of [Kahl and Jäckel]<sup>[10]</sup> concerns itself with this problem and provides an ingenious resolution.

## A better solution

- With *our* definition (13) of  $C(u, \tau)$  whenever the imaginary part of the argument of the logarithm is zero, the real part is positive.
  - Plotted in the complex plane, the argument of the logarithm never cuts the negative real axis.
  - This result was proved by [Albrecher, Mayer, Schoutens and Tistaert]<sup>[11]</sup>.
- It follows that with our definition of  $C(u, \tau)$  taking the principal value of the logarithm leads to a continuous integrand over the full range of integration.

## Characteristic functions and option prices

To a mathematician, it may appear perverse to first derive the option pricing formula then impute the characteristic function: the reverse might appear more natural.

However, in the context of understanding the volatility surface, option prices really are primary and it makes just as much sense for us to deduce the characteristic function from the option pricing formula as it does for us to deduce the risk-neutral density from option prices.

The following are equivalent in the sense that given one of these, all of the others may be derived:

- All  $T$ -expiration European option prices.
- The  $T$ -expiration risk-neutral density.
- The  $T$ -expiration implied volatilities.
- The  $T$ -expiration characteristic function.

## The Heston tail distribution

By definition, the characteristic function is given by

$$\phi_T(u) := \mathbb{E} \left[ e^{iux_T} \mid x_t = 0 \right]$$

The probability of the final log-stock price  $x_T$  being greater than the strike price is given by

$$\begin{aligned} & \Pr(x_T > x) \\ &= P_0(x, v, \tau) \\ &= \frac{1}{2} + \frac{1}{\pi} \int_0^\infty du \operatorname{Re} \left\{ \frac{\exp\{C(u, \tau) \bar{v} + D(u, \tau) v + i u x\}}{i u} \right\} \end{aligned} \quad \Bigg|$$

with  $x = \log(S_T/K)$  and  $\tau = T - t$

## The Heston marginal density

Let the log-strike  $k$  be defined by  $k = \log(K/S_t) = -x$ . Then, the probability density function  $p(k)$  must be given by

$$\begin{aligned} p(k) &= -\frac{\partial P_0}{\partial k} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} du' \exp\{C(u', \tau) \bar{v} + D(u', \tau) v - i u' k\} \end{aligned}$$

Then

(16)

$$\begin{aligned} \phi_T(u) &= \int_{-\infty}^{\infty} dk p(k) e^{i u k} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} du' \exp\{C(u', \tau) \bar{v} + D(u', \tau) v\} \int_{-\infty}^{\infty} du e^{i(u-u')k} \\ &= \int_{-\infty}^{\infty} du' \exp\{C(u', \tau) \bar{v} + D(u', \tau) v\} \delta(u - u') \\ &= \exp\{C(u, \tau) \bar{v} + D(u, \tau) v\} \end{aligned}$$

## The Heston characteristic function

For emphasis:

**The Heston characteristic function**

$$\phi_T(u) = \exp \{ C(u, \tau) \bar{v} + D(u, \tau) v \}$$

## The Heston density

We may then compute the Heston probability density function by Fourier inversion as:

(17)

$$\begin{aligned} p(y, \tau) &= \int_{-\infty}^{\infty} \frac{du}{2\pi} e^{-i u y} \phi_T(u) \\ &= \frac{1}{\pi} \int_0^{\infty} du \operatorname{Re} [\exp \{ C(u, \tau) \bar{v} + D(u, \tau) v - i u y \}] \end{aligned}$$

- Note that  $y = \log S_T/S_t$  is the forward variable.

## Autocorrelation of variance in the Heston model

The autocorrelation function of variance is estimated as

$$\rho(\tau) \sim \frac{\text{Cov}(v_t, v_{t+\tau})}{\text{Var}(v_t)}$$

To compute this, first compute

$$\mathbb{E}[v_{t+\tau} | v_t] = \bar{v} + (v_t - \bar{v}) e^{-\lambda \tau}$$

The unconditional expectation of  $v_t$  is  $\bar{v}$  so we get

$$\begin{aligned} \text{Cov}[v_{t+\tau}, v_t] &= \mathbb{E}[\mathbb{E}[v_{t+\tau} v_t | v_t]] - \bar{v}^2 \\ &= \mathbb{E}[v_t (\bar{v} + (v_t - \bar{v}) e^{-\lambda \tau})] - \bar{v}^2 \\ &= e^{-\lambda \tau} \text{Var}[v_t] \end{aligned}$$

and the autocorrelation function in the Heston model is just

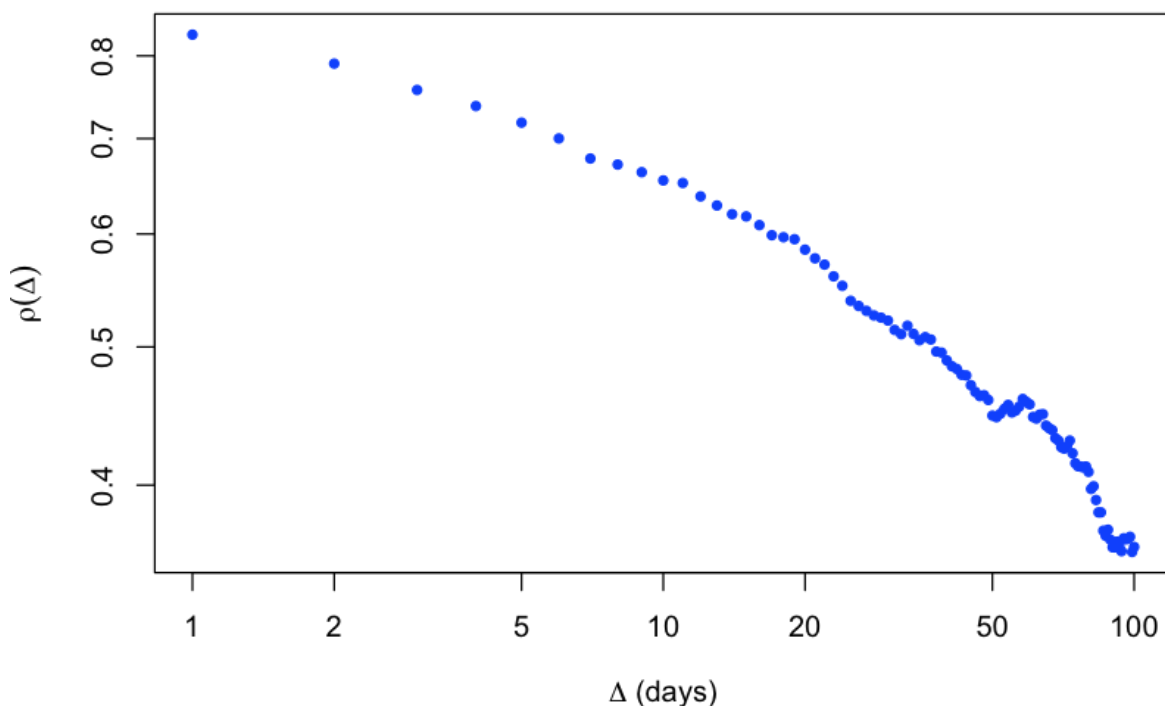
$$\rho(\tau) = e^{-\lambda \tau}$$

## Empirical autocorrelation of volatility

```
In [7]: load("oxfordRV.rData")
```

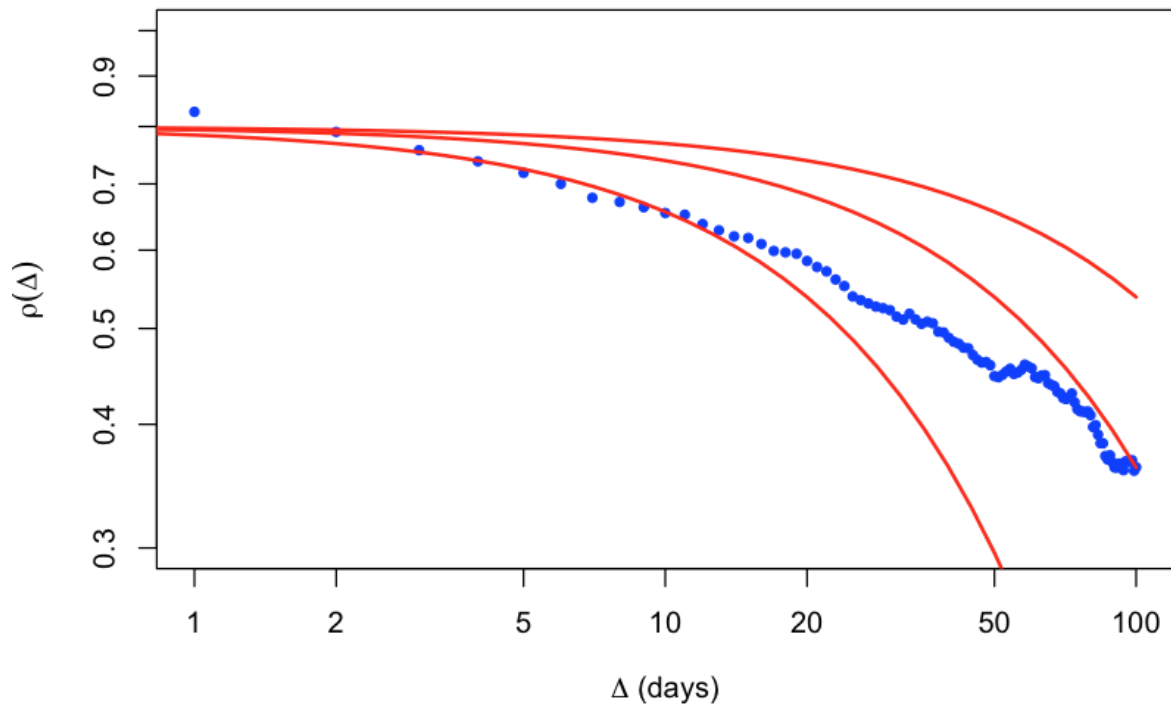
```
In [8]: v <- rv.list[[1]] # Pick spx.rk
sig <- sqrt(as.numeric(v))
aclog <- acf(log(sig), lag=100, plot=F)
```

```
In [9]: plot(aclog$lag[-1], aclog$acf[-1], pch=20, ylab=expression(rho(Delta)), xlab=exp
```



Now add the Heston autocorrelation function with  $\lambda = 1$ ,  $\lambda = 2$  and  $\lambda = 5$  respectively.

```
In [10]: plot(aclog$lag[-1],aclog$acf[-1],pch=20,ylab=expression(rho(Delta)),ylim=c(0.3,0.9),
             xlab=expression(paste(Delta," (days)")),log="xy",col="blue")
             curve(.8*exp(-x/252),from=0.001,to=100,col="red",add=T,lwd=2,log="xy")
             curve(.8*exp(-x*2/252),from=0.001,to=100,col="red",add=T,lwd=2,log="xy")
             curve(.8*exp(-x*5/252),from=0.001,to=100,col="red",add=T,lwd=2,log="xy")
```

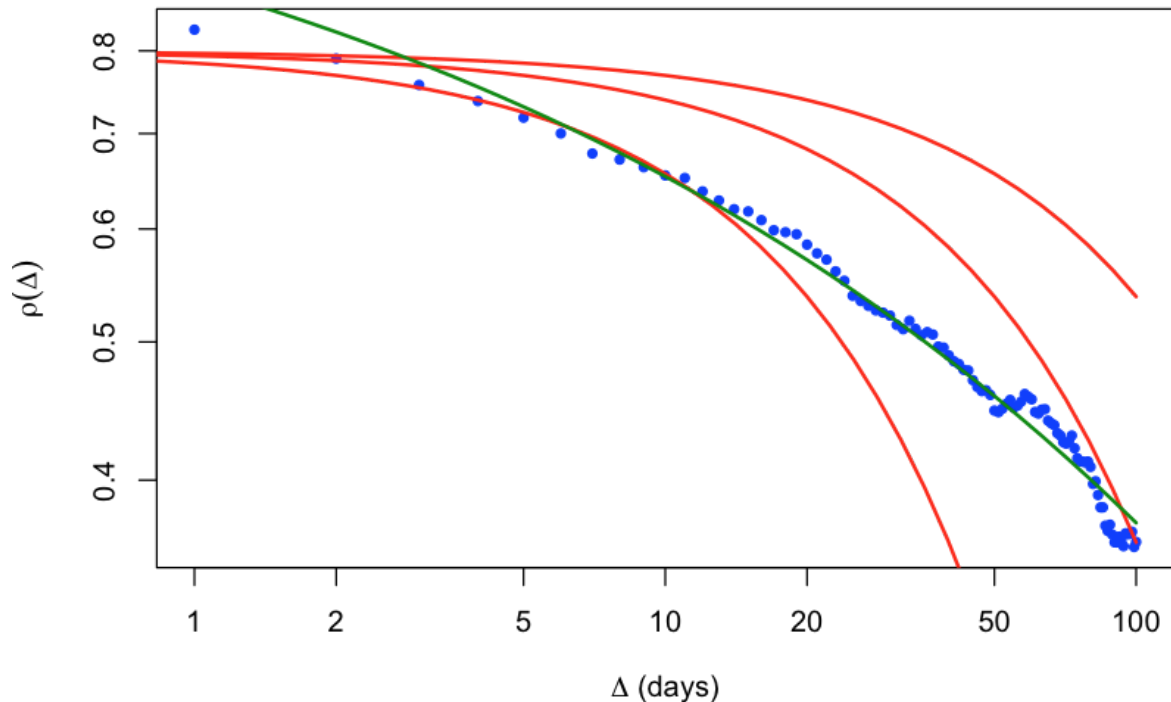


Now add the rough volatility approximate autocorrelation function  $\exp\{-\eta^2 \Delta^{2H}/2\}$

```
In [11]: v <- rv.list[[1]] # Pick spx.rk
          sig <- sqrt(as.numeric(v))

          h.spx.regression <- .13
          aclog <- acf(log(sig),lag=100,plot=F)
          y <- log(aclog$acf)
          x <- aclog$lag^(2*h.spx.regression)
          fit.lm <- lm(y[-1]~x[-1])
          a <- fit.lm$coef[1]
          b <- fit.lm$coef[2]
```

```
In [12]: plot(aclog$lag[-1],aclog$acf[-1],pch=20,ylab=expression(rho(Delta)),
             xlab=expression(paste(Delta," (days)")),log="xy",col="blue")
curve(.8*exp(-x/252),from=0.001,to=100,col="red",add=T,lwd=2,log="xy")
curve(.8*exp(-x*2/252),from=0.001,to=100,col="red",add=T,lwd=2,log="xy")
curve(.8*exp(-x*5/252),from=0.001,to=100,col="red",add=T,lwd=2,log="xy")
curve(exp(a+b*x^(2*h.spx.regression)),from=0.001,to=100,col="green4",add=T,lwd=2,log="xy")
```



## Model versus empirical observation

- In the Heston model, the variance autocorrelation function  $\sim \exp\{-\lambda \tau\}$ 
  - This is true for all conventional one-factor stochastic volatility models where the volatility process is of the form:
$$dv = -\lambda(v - \bar{v})dt + \text{noise}$$
  - Single-factor stochastic volatility models are not compatible with the volatility time series!
- We can match the autocorrelation function more realistically by adding another timescale as in the DMR model we presented in Lecture 2.
  - Or even better, as we can see from the plot, we can reproduce the observed autocorrelation function using the Rough Volatility model<sup>[6]</sup> which we will explore in a later lecture.

## Implementation of the Heston model in R

- In the next few slides, we will exhibit an implementation of the Heston model in R.

- We can then develop some intuition for the behavior of the model through experiment.

## Black-Scholes

First, we need an implementation of the Black-Scholes formula (this is in `BlackScholes.R`).

```
In [16]: BSFormula <- function(S0, K, T, r, sigma)
{
  x <- log(S0/K)+r*T
  sig <- sigma*sqrt(T)
  d1 <- x/sig+sig/2
  d2 <- d1 - sig
  pv <- exp(-r*T)
  return( S0*pnorm(d1) - pv*K*pnorm(d2))
}
```

## Implied volatility computation

Then, we need an implied volatility computation (this is also in `BlackScholes.R`).

```
In [17]: BSImpliedVolCall <- function(S0, K, T, r, C)
{
  nK <- length(K)
  sigmaL <- rep(1e-10,nK)
  CL <- BSFormula(S0, K, T, r, sigmaL)
  sigmaH <- rep(10,nK)
  CH <- BSFormula(S0, K, T, r, sigmaH)
  while (mean(sigmaH - sigmaL) > 1e-10)
  {
    sigma <- (sigmaL + sigmaH)/2
    CM <- BSFormula(S0, K, T, r, sigma)
    CL <- CL + (CM < C)*(CM-CL)
    sigmaL <- sigmaL + (CM < C)*(sigma-sigmaL)
    CH <- CH + (CM >= C)*(CM-CH)
    sigmaH <- sigmaH + (CM >= C)*(sigma-sigmaH)
  }
  return(sigma)
}
```

## A numerical example

First generate some option prices.

```
In [18]: vols <- c(0.23,0.20,0.18)
K <- c(0.9,1.0,1.1)
(optVals <- BSFormula(S0=1,K,T=1,r=0,sigma=vols))
```

```
0.145896960390544 0.079655674554058 0.0355767789605738
```

Then compute implied volatilities.

```
In [19]: (impVols <- BSImpliedVolCall(S0=1, K, T=1, r=0, C=optVals))
```

```
0.2300000000035418 0.2000000000065986 0.1799999999989352
```

## Heston formula

```
In [20]: callHeston <- function(params){

  lambda <- params$lambda
  rho <- params$rho
  eta <- params$eta
  vbar <- params$vbar
  v <- params$v

  res <- function(k,t){
    fj <- function(u, t, j){

      al <- -u*u/2 - 1i*u/2 + 1i*j*u
      bet <- lambda - rho*eta*1i*u-rho*eta*j
      gam <- eta^2/2
      d <- sqrt(bet*bet - 4*al*gam)
      rp <- (bet + d)/(2*gam)
      rm <- (bet - d)/(2*gam)
      g <- rm / rp
      D <- rm * (1 - exp(-d*t))/ (1 - g*exp(-d*t))
      C <- lambda * (rm * t - 2/eta^2 * log( (1 - g*exp(-(d*t)))/(1 - g*exp(-d*t))))
      return(exp(C*vbar + D*v))
    }
    x <- -k
    p1 <- 1/2+1/pi*integrate(function(u){Re(fj(u,t,1)*exp(1i*u*x)/(1i*u))},0,100)
    p0 <- 1/2+1/pi*integrate(function(u){Re(fj(u,t,0)*exp(1i*u*x)/(1i*u))},0,100)
    return(exp(k)*(exp(x)*p1-p0))
  }
  return(res) # Return price of Heston call expressed as percentage of spot
}
```

## Some notable features of R

- Complex arithmetic with `1i`
- Functional programming:
  - This is what allows us to code a function which is called as:

```
callHeston(params)(k,T)
```

- We can define a function that returns a function (and so on indefinitely).
- We could even conveniently define a new function:

```
callHestonBCC <- callHeston(paramsBCC)
```

- We can conceptually separate parameters and variables rather than having to carry all the parameters around with each function call.

## The Heston smile with BCC parameters

First we list the BCC parameters.

```
In [21]: paramsBCC <- list(lambda = 1.15, rho = -0.64, eta = 0.39, vbar = 0.04, v = .04)
```

Then we need a function to compute Heston implied volatilities.

```
In [22]: impvolHeston <- function(params){
  function(k, t){
    BSImpliedVolCall(1, exp(k), t, 0, callHeston(params)(k,t))
  }
}
```

Now finally, we can plot the Heston volatility smile with BCC paramets

```
In [23]: vol <- function(k){sapply(k,function(x){impvolHeston(paramsBCC)(x,1)}})
system.time(curve(vol(x),from=-0.4,to=0.4,col="red",ylab="Implied vol.",xlab=
```

user	system	elapsed
0.133	0.006	0.140

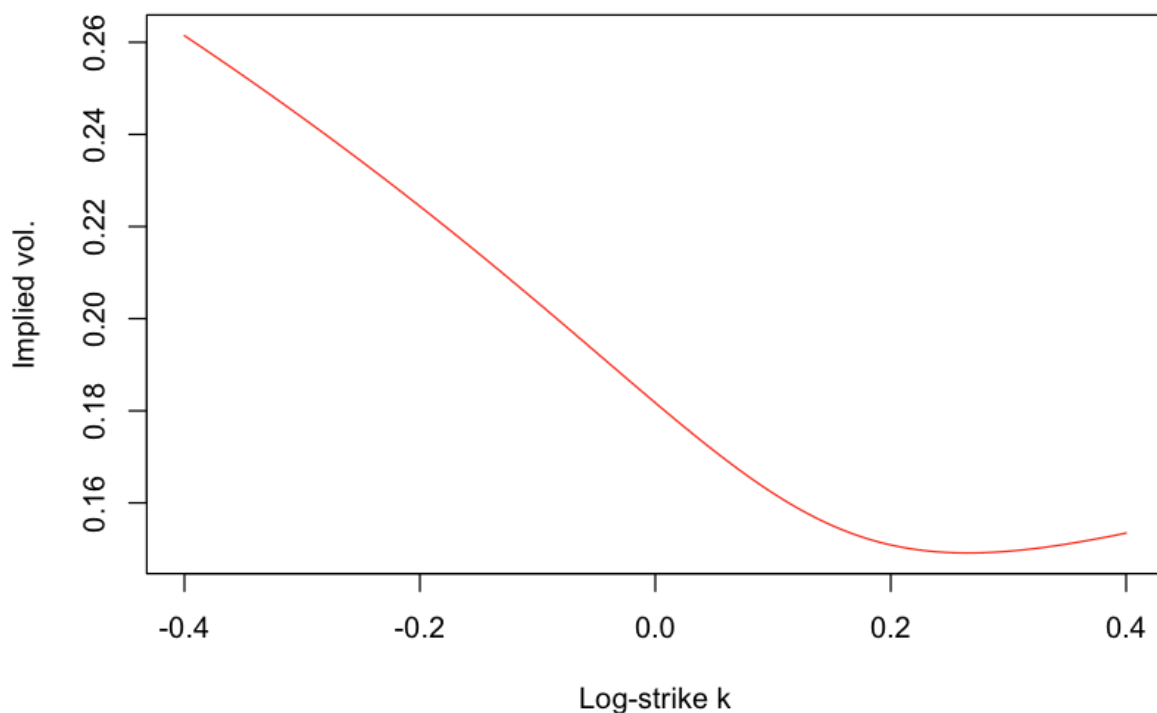


Figure 3: The  $\tau = 1$  Heston smile with BCC parameters.



## Sensitivity of the Heston smile to $\rho$

```
In [24]: subH <- function(incr){
  tmp <- paramsBCC
  tmp$rho <- tmp$rho+incr
  return(tmp)
}

yrange <- c(0.15,.35)
vol <- function(k){sapply(k,function(x){impvolHeston(paramsBCC)(x,1)}})
curve(vol(x),from=-.5,to=.5,col="red",ylim=yrange,lwd=2,ylab="Implied vol.",:
for (incr in seq(0.1,0.5,0.1))
{
  for (i in 1:100000){tmp=i}
  par(new=T)
  vol <- function(k){sapply(k,function(x){impvolHeston(subH(incr))(x,1)}})
}
curve(vol(x),from=-.5,to=.5,col="red",lty=2,ylim=yrange,ylab=NA,xlab=NA)
}
```

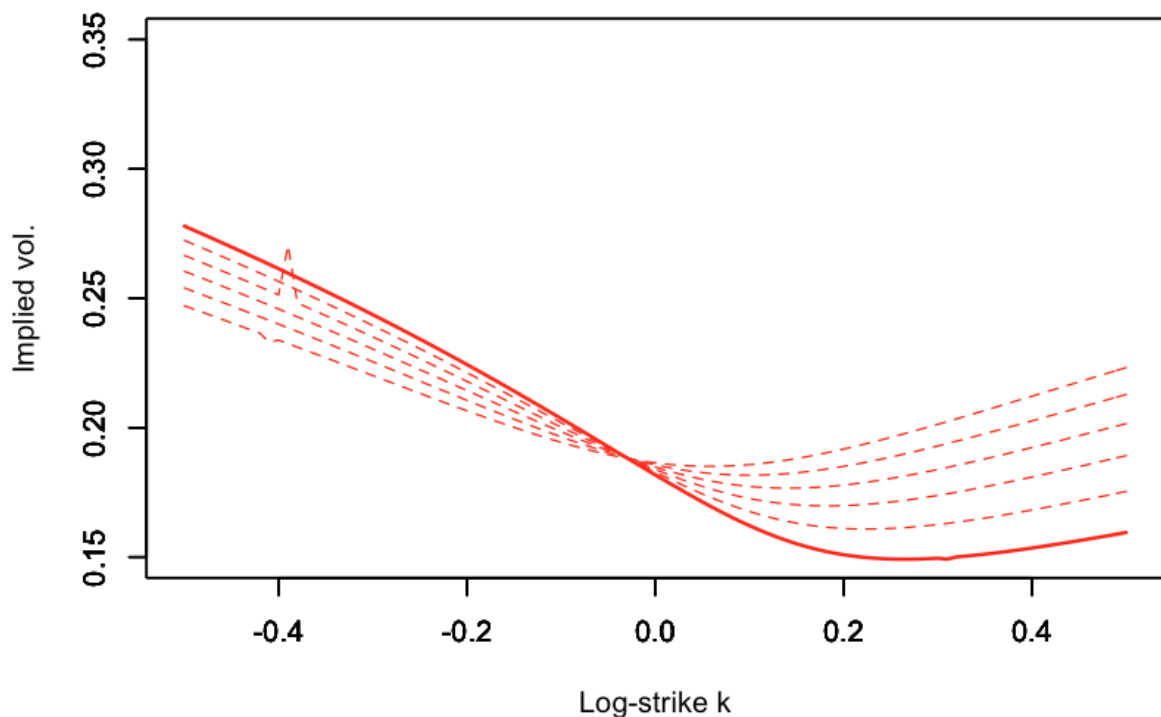


Figure 4: The dotted lines are smiles with  $\rho = \rho_{BCC} + \{0.1, 0.2, 0.3, 0.4, 0.5\}$

## Sensitivity of the Heston smile to $\eta$

```

In [25]: subH <- function(incr){
  tmp <- paramsBCC
  tmp$eta <- tmp$eta+incr
  return(tmp)
}

yrange <- c(0.1,.35)
vol <- function(k){sapply(k,function(x){impvolHeston(paramsBCC)(x,1)}})
curve(vol(x),from=-.5,to=.5,col="red",ylim=yrange,lwd=2,ylab="Implied vol.",:
for (incr in c(seq(0.1,0.5,0.1),0.64))
{
  for (i in 1:100000){tmp=i}
  par(new=T)
  vol <- function(k){sapply(k,function(x){impvolHeston(subH(incr))(x,1)}})
}
  curve(vol(x),from=-.5,to=.5,col="red",lty=2,ylim=yrange,ylab=NA,xlab=NA)
}

```

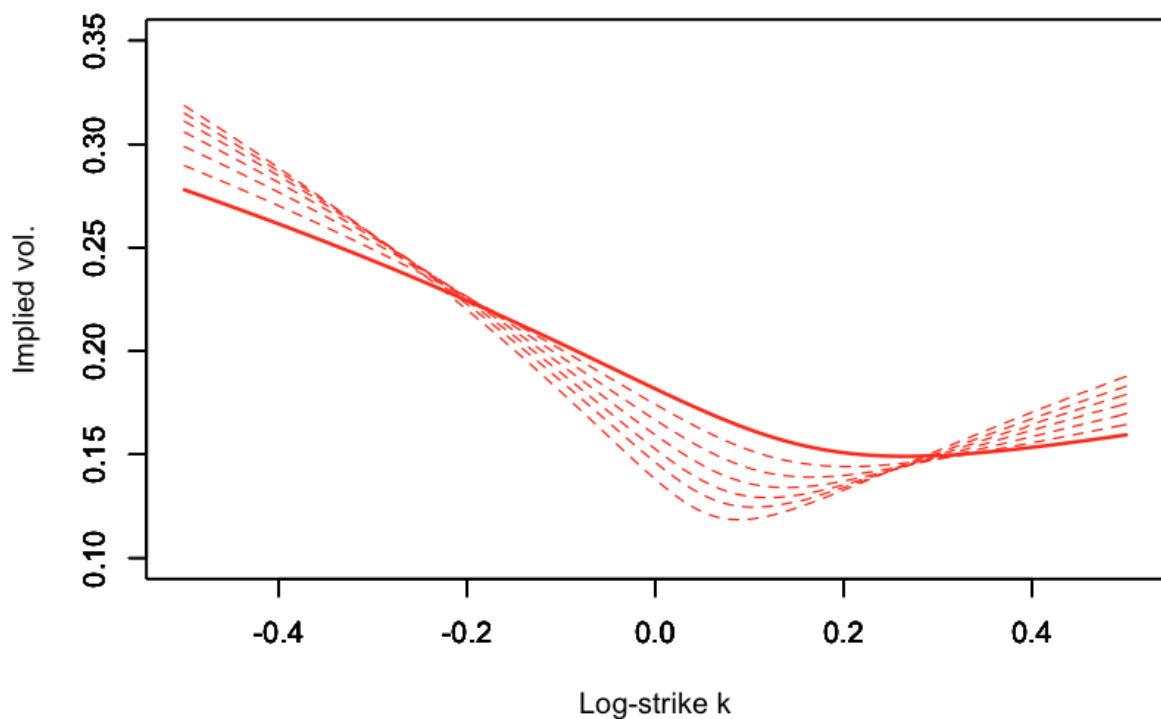


Figure 5: The dotted lines are smiles with  $\eta = \eta_{BCC} + \{0.1, 0.2, 0.3, 0.4, 0.5\}$

### Sensitivity of the Heston smile to $\lambda$

```

In [26]: subH <- function(incr){
  tmp <- paramsBCC
  tmp$lambda <- tmp$lambda+incr
  return(tmp)
}

yrange <- c(0.1,.35)
vol <- function(k){sapply(k,function(x){impvolHeston(paramsBCC)(x,1)}})
curve(vol(x),from=-.5,to=.5,col="red",ylim=yrange,lwd=2,ylab="Implied vol.",:
for (incr in seq(0.5,2.5,0.5))
{
  for (i in 1:100000){tmp=i}
  par(new=T)
  vol <- function(k){sapply(k,function(x){impvolHeston(subH(incr))(x,1
}
curve(vol(x),from=-.5,to=.5,col="red",lty=2,ylim=yrange,ylab=NA,xlab=NA)
}

```

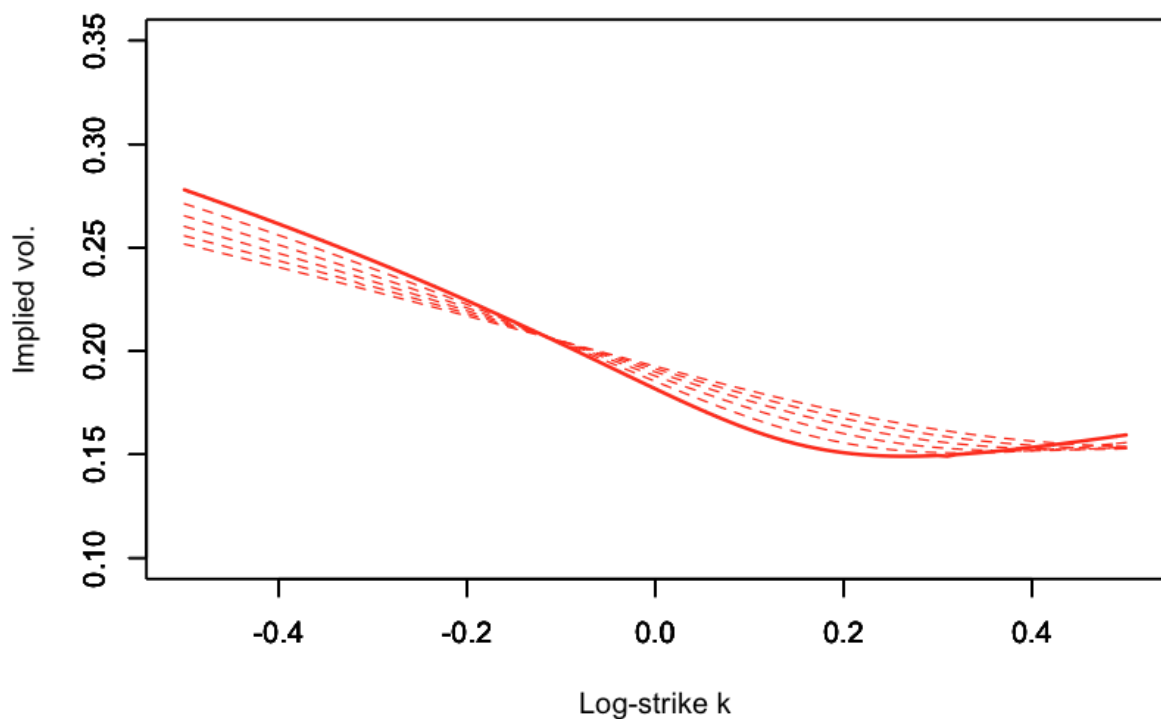


Figure 6: The dotted lines are smiles with  $\lambda = \lambda_{BCC} + \{0.5, 1.0, 1.5, 2.0, 2.5\}$

## Another sensitivity plot

Now increase both  $\lambda$  and  $\eta$  keeping the ratio  $\eta^2/\lambda$  constant.

```

In [27]: subH <- function(incr){
  tmp <- paramsBCC
  tmp$eta <- tmp$eta+incr
  tmp$lambda <- tmp$lambda*(tmp$eta/paramsBCC$eta)^2
  return(tmp)
}

yrange <- c(0.1,.35)
vol <- function(k){sapply(k,function(x){impvolHeston(paramsBCC)(x,1)}})
curve(vol(x),from=-.5,to=.5,col="red",ylim=yrange,lwd=2,ylab="Implied vol.",:
for (incr in seq(0.1,0.5,0.1))
{
  for (i in 1:100000){tmp=i}
  par(new=T)
  vol <- function(k){sapply(k,function(x){impvolHeston(subH(incr))(x,1)}})
}
curve(vol(x),from=-.5,to=.5,col="red",lty=2,ylim=yrange,ylab=NA,xlab=NA)
}

```

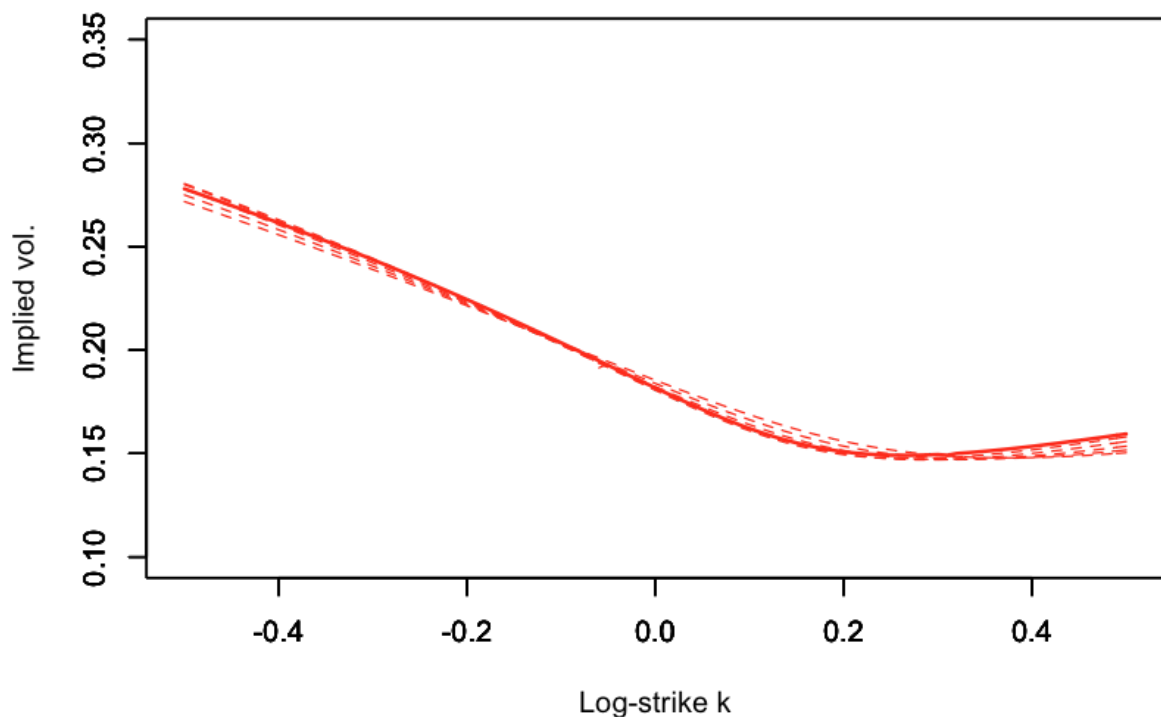


Figure 7: The dotted lines are smiles with  $\eta = \eta_{BCC} + \{0.1, 0.2, 0.3, 0.4, 0.5\}$  and  $\lambda$  incremented so as to keep the ratio  $\lambda/\eta^2$  constant.

## Expectation of instantaneous variance in the Heston model

Define the forward variance

$$\xi_t(u) = \mathbb{E}[v_u | \mathcal{F}_t]$$

Recall the Heston variance SDE

$$dv_t = -\lambda(v - \bar{v}) dt + \eta \sqrt{v_t} dW_t$$

Take expectations to get

$$d\xi_t(u) = -\lambda (\xi_t(u) - \bar{v}) du$$

This ODE has the solution

$$\xi_t(u) = (\xi_t(t) - \bar{v}) e^{-\lambda(u-t)} + \bar{v} = (v_t - \bar{v}) e^{-\lambda(u-t)} + \bar{v}$$

## Variance of instantaneous variance in the Heston model

Applying Itô's Lemma, we get

$$dv^2 = 2v dv + (dv)^2$$

With  $\hat{v} = \mathbb{E}[v]$ , this gives

$$\mathbb{E}[dv^2] = -2\lambda (\mathbb{E}[v^2] - \bar{v}\hat{v}) dt + \eta^2 \hat{v} dt$$

Also

$$d\hat{v}^2 = -2\lambda (\hat{v}^2 - \bar{v}\hat{v}) dt$$

So

$$d\text{Var}(v) = -2\lambda \text{Var}(v) dt + \eta^2 \hat{v} dt$$

Integrating, we get

$$\text{Var}(v_T) = \int_0^T e^{-2\lambda(T-t)} \eta^2 \hat{v}_t dt$$

Putting  $\hat{v} = \bar{v}$  for simplicity, we obtain

$$\begin{aligned} \text{Var}(v_T) &= \eta^2 \bar{v} \int_0^T e^{-2\lambda(T-t)} dt \\ &= \eta^2 \bar{v} \frac{1 - e^{-2\lambda T}}{2\lambda} \\ &\rightarrow \frac{\eta^2 \bar{v}}{2\lambda} \text{ as } T \rightarrow \infty \end{aligned}$$

- The smile depends only on the distribution of integrated variance and its correlation with the stock price, so for longer expirations, the smile should depend (roughly) only on  $\lambda$  and the ratio  $\eta^2 \bar{v}/\lambda$

## Simulation of the Heston model

## Numerical computation under stochastic volatility

- The popularity of the Heston model relates to the existence of a quasi-closed-form formula for European option prices.
  - To value other types of option, even American options, numerical computations are required.
- For other more realistic dynamics, numerical computation is always required.
  - For example, consider the much more realistic (lognormal) variance process:

$$dv = -\lambda (v - \bar{v}) dt + \xi v dZ_t$$

- The two numerical techniques that are typically employed are:
  - Numerical PDE,
  - Monte Carlo simulation.

## Feynman-Kac

The Feynman-Kac formula relates the solution of a diffusion PDE to a probabilistic expectation, which may be computed using Monte Carlo simulation.

Specifically, consider the SDE

$$dx_t = \mu_t dt + \sigma_t dZ_t$$

and consider the discounted expectation of some function  $V(x_T)$  of the terminal value  $x_T$ .

Then, applying Itô's Lemma to

$$f(x_t, t) = e^{-\int_0^t r_s ds} V(x_t, t)$$

we obtain

$$\begin{aligned} df &= -r_t f dt + e^{-\int_0^t r_s ds} \left( \partial_t V dt + \partial_x V dx + \frac{\sigma_t^2}{2} \partial_{xx} V dt \right) \\ &= e^{-\int_0^t r_s ds} \left\{ -r_t V + \partial_t V + \mu_t \partial_x V + \frac{\sigma_t^2}{2} \partial_{xx} V \right\} dt + \partial_x V \sigma_t dZ_t. \end{aligned}$$

Then, if  $V(x, t)$  satisfies the PDE

$$\partial_t V + \frac{\sigma_t^2}{2} \partial_{xx} V + \mu_t \partial_x V - r_t V = 0,$$

we have  $\mathbb{E}[df] = 0$  and

$$\mathbb{E}[f(x_T, T)] - f(x_t, t) = \int_t^T \mathbb{E}[df(x_s, s)] ds = 0$$

Rearranging gives

### The Feynman-Kac formula

$$V(x_t, t) = \mathbb{E} \left[ e^{-\int_t^T r_s ds} V(x_T, T) \right]$$

In our case, we assume zero interest rates and the terminal condition is of the form  $V(x, T) = g(x)$ . Then our problem is to compute

$$V(x_t, t) = \mathbb{E} [V(x_T, T) | \mathcal{F}_t] = \mathbb{E} [g(x_T) | \mathcal{F}_t].$$

### Implications for valuation of claims under stochastic volatility

- Risk-neutral (discounted) expectations of final payoffs may be valued using:
  - Numerical PDE,
  - Monte Carlo simulation.
- We may use whichever technique is more convenient.
- Numerical PDE techniques are well-explained in [Tavella and Randall]<sup>[9]</sup>.
- We will focus on Monte Carlo simulation.
- Simulation
  - is in general easier to understand,
  - is less reliant on specialized artisanal knowledge,
  - can deal with higher dimensional processes (no *curse of dimensionality*).

### Euler-Maruyama

- A very popular method to compute  $\mathbb{E} [g(x_T)]$  is to use the Euler-Maruyama (EM) scheme.
- Specifically, choose  $n$  independent random variables  $\{Y_1, \dots, Y_n\}$  such that if  $Z \sim N(0, 1)$ ,  $\mathbb{E} [p(Y_k)] = \mathbb{E} [p(Z)]$  for all polynomials of degree less than or equal to 3.
- Then, with  $\Delta = (T - t)/n$  and  $s = t + k \Delta$ , define recursively

$$x_{s+\Delta} = x_s + \mu \Delta + \sigma \sqrt{\Delta} Y_s.$$

- If for example, the  $Y_n$  were Bernoulli random variables,  $\mathbb{E} [g(x_T)]$  would be computed as a sum of  $2^n$  terms.
  - In practice of course, one would approximate this sum using Monte Carlo (MC) or quasi-Monte Carlo (QMC) methods.

### Weak and strong convergence

From page 327 of [Kloeden and Platen]<sup>[8]</sup>, a time discretization  $\hat{X}$  with time increment  $\Delta$  converges weakly with order  $\beta$  if there exists a constant  $C$  such that

$$\left| \mathbb{E} [g(\hat{X}_T)] - \mathbb{E} [g(X_T)] \right| \leq C \Delta^\beta$$

as  $\Delta \rightarrow 0$ .

From page 323 of [Kloeden and Platen]<sup>[1]</sup>,  $\hat{X}$  with time increment  $\Delta$  converges strongly with order  $\beta$  if there exists a constant  $C$  such that

$$\mathbb{E} [\|\hat{X}_T - X_T\|] \leq C \Delta^\beta$$

as  $\Delta \rightarrow 0$ .

- The exponent  $\beta$  is called the *order* of convergence.

## Monte Carlo simulation

Monte Carlo simulation proceeds in two steps:

1. Discretize the SDE with time step  $\Delta = T/m$ , denoting the discretized process by  $\hat{X}_t$ . This step induces a weak error

$$\epsilon_D = \mathbb{E}[g(X_T)] - \mathbb{E}[g(\hat{X}_T)] \sim \Delta^\beta$$

2. Estimate the expectation of the discretized process as

$$\mathbb{E}[g(\hat{X}_T)] = \frac{1}{N} \sum_{i=1}^N g(\hat{X}_T^{(i)})$$

This step induces a statistical error

$$\epsilon_S = \mathbb{E}[g(\hat{X}_T)] - \frac{1}{N} \sum_{i=1}^N g(\hat{X}_T^{(i)}) \sim \frac{1}{\sqrt{N}}$$

## Mean squared error (MSE)

If  $\hat{x}$  is an estimator of some quantity  $x$  (a Monte Carlo estimate for example), then

$$\begin{aligned} \text{MSE}(\hat{x}) &= \mathbb{E}[(\hat{x} - x)^2] \\ &= (\mathbb{E}[\hat{x}] - x)^2 + \mathbb{E}[(\hat{x} - \mathbb{E}[\hat{x}])^2] \\ &= \text{Bias}^2(\hat{x}) + \text{Variance}(\hat{x}). \end{aligned}$$

- Bias relates to the discretization.
  - The number of timesteps and how accurately the target process is approximated over each timestep.
- The variance relates to the number of samples and how variance is reduced.

## Computational complexity



- Following page 365 of [Glasserman]<sup>[7]</sup>, denoting MSE by  $\epsilon^2$ , we have

(18)

$$\epsilon^2 \sim \epsilon_D^2 + \epsilon_S^2 = c_1 \Delta^{2\beta} + \frac{c_2}{N}$$

for some constants  $c_1$  and  $c_2$

- The computational cost  $C$  is proportional to

$$m \times N = \frac{T}{\Delta} N$$

so

$$\epsilon^2 \sim c_1 \Delta^{2\beta} + \frac{c_3}{C \Delta}$$

for some  $c_3$

- Minimizing MSE for fixed cost  $C$  gives

$$\Delta \propto C^{-\frac{1}{2\beta+1}}$$

## Optimal allocation of computational budget

Substituting back into (18), we obtain

$$\epsilon^2 \sim c_1 C^{-\frac{2\beta}{2\beta+1}} + c_2 T C^{-\frac{2\beta}{2\beta+1}} \propto C^{-\frac{2\beta}{2\beta+1}}$$

The optimal allocation thus balances bias squared and variance.

- For the Euler scheme with Lipschitz payoff,  $\beta = 1$  so  $\epsilon \sim C^{-1/3}$
- For the Milstein scheme with Lipschitz payoff,  $\beta = 2$  so  $\epsilon \sim C^{-2/5}$
- If the discretization is unbiased,  $\epsilon \sim C^{-1/2}$

Thus, in the limit where very high accuracy is required, the better the discretization, the lower the computational cost.

- Although in practice, we have to take into account that a better discretization may involve more computation per time step.

## Why fast Monte Carlo?

- For pricing, a not-very-fast Monte Carlo routine might suffice.
- For model calibration, we need fast Monte Carlo.
  - Fitting a model to the data involves many calls to the pricing routine with different parameter sets.

- However fast Monte Carlo is only needed for European options.
  - We want to calibrate complex models to vanilla option prices.
  - Even if vanilla options are American-style, we can fairly accurately approximate Black-Scholes implied volatility.
- We are then only concerned with the order of weak convergence.
  - In contrast, the order of strong convergence may be relevant for barrier or lookback options.

## Heston simulation: General remarks

- Heston is popular because there is a quasi-closed form formula for European options.
- The Heston model is in fact very tricky to simulate.
  - This relates to the propensity of the Heston variance process to go to zero.
  - This same property makes Heston dynamics very unrealistic.
- If we can figure out how to simulate Heston efficiently, other stochastic volatility models will be easier.

## Variations on the BCC parameters

In numerical examples, we will consider the following parameter sets with  $\eta$  halved and doubled relative to BCC:

Table 1: BCC and BCC2 parameters

	BCC	BCC1	BCC2
$\mu$	0.04	0.04	0.04
$\bar{\nu}$	0.04	0.04	0.04
$\lambda$	1.15	1.15	1.15
$\rho$	-0.64	-0.64	-0.64
$\eta$	0.39	0.39 / 2	0.39 $\times$ 2
$\frac{2\lambda\bar{\nu}}{\eta^2}$	0.60	2.42	0.15

## Boundary behaviors of variance processes

Consider a fairly generic variance process:

(4)

$$dv = -\lambda(v - \bar{\nu})dt + \eta v^\alpha dZ$$

### Proposition 2.1 of Andersen and Piterbarg

For the process (4), the following holds:

1. 0 is always an attainable boundary for  $0 < \alpha < 1/2$ ;
2. 0 is an attainable boundary for  $\alpha = 1/2$  if  $2\lambda\bar{v} < \eta^2$ ;
3. 0 is an unattainable boundary for  $\alpha > 1/2$ ;
4.  $\infty$  is an unattainable boundary for all values of  $\alpha > 0$ .

## Implications for Heston

- Then, with BCC1 parameters, the Heston variance process does not hit zero.
- On the other hand, with both BCC and BCC2 parameters, the process can hit zero and does spend considerable time in the neighborhood of zero.
  - If the discretized variance process can go negative, this may lead to substantial bias.

## Lord, Koekkoek and Van Dyck

Write the Heston variance process as

$$v_{t+\Delta} = f_1(v_t) - \lambda (f_2(v_t) - \bar{v}) \Delta + \eta (f_3(v_t))^\alpha \sqrt{\Delta} Z$$

with  $Z \sim N(0, 1)$  [Lord, Koekkoek and Van Dyck]<sup>[9]</sup> classify discretization schemes as follows:

Table 2: Fixing functions  $f_i(\cdot)$  in various discretizations.

Scheme	$f_1(x)$	$f_2(x)$	$f_3(x)$
Absorption	$x^+$	$x^+$	$x^+$
Reflection	$ x $	$ x $	$ x $
Higham and Mao	$x$	$x$	$ x $
Partial truncation	$x$	$x$	$x^+$
Full truncation	$x$	$x^+$	$x^+$

Based on results of numerical experiments, Lord et al. recommend full truncation.

## Comments on discretization of the Heston variance process

- The choice of discretization makes a *big* difference.
  - The choices of discretization presented in *The Volatility Surface*<sup>[5]</sup> are particularly horrible; hundreds of thousands of time steps are required to get decent

convergence with typical Heston parameters.

- We will follow LKV's advice and implement full truncation. Specifically, we will discretize the Heston variance process as

$$v_{t+\Delta} = v_t - \lambda (v_t^+ - \bar{v}) \Delta + \eta \sqrt{v_t^+} \sqrt{\Delta} Z_t$$

## Richardson extrapolation

Suppose that the Euler scheme generates an estimate of the true value of a claim as follows:

$$\mathbb{E} [g(\hat{X}_T)] = \mathbb{E} [g(X_T)] + c \Delta + \mathcal{O}(\Delta^2)$$

where  $\hat{X}$  is an approximation to  $X$  based on this Euler discretization scheme. This is true under strong regularity assumptions on the drift and diffusion coefficients in the underlying SDE.

Then for positive integer  $M$  (typically  $M = 2$ ), we have

$$\frac{M}{M-1} \mathbb{E} [g(\hat{X}_T^\Delta)] - \frac{1}{M-1} \mathbb{E} [g(\hat{X}_T^{M\Delta})] = \mathbb{E} [g(X_T)] + \mathcal{O}(\Delta^2)$$

- Convergence looks linear for our Euler discretizations so Richardson extrapolation might work in this case, giving order 2 weak convergence.

## Variance of extrapolated estimate

Following [Glasserman]<sup>[6]</sup> again (page 361), note that

$$\begin{aligned} & \text{Var} \left[ \frac{M}{M-1} g(\hat{X}_T^\Delta) - \frac{1}{M-1} g(\hat{X}_T^{M\Delta}) \right] \\ &= \frac{M^2}{(M-1)^2} \text{Var} [g(\hat{X}_T^\Delta)] + \frac{1}{(M-1)^2} \text{Var} [g(\hat{X}_T^{M\Delta})] \\ & \quad - \frac{2M}{(M-1)^2} \text{Cov} [g(\hat{X}_T^\Delta), g(\hat{X}_T^{M\Delta})]. \end{aligned}$$

One might hope, and this is typically the case, that by using the same random numbers to simulate  $\hat{X}_T^\Delta$  and  $\hat{X}_T^{M\Delta}$ , the covariance term would be positive. However, in the case of the Heston model, especially with BCC2 parameters, this is not at all guaranteed.

- If the covariance term is zero or negative, the variance of the extrapolated estimate is greater than that of the original estimate (although the bias is smaller).

## Heston parameters

```
In [28]: paramsBCC <- list(lambda = 1.15, rho = -0.64, eta = 0.39, vbar = 0.04, v = 0.04)
        paramsBCC1 <- list(lambda = 1.15, rho = -0.64, eta = .39/2, vbar = 0.04, v = 0.04)
        paramsBCC2 <- list(lambda = 1.15, rho = -0.64, eta = 0.39*2, vbar = 0.04, v = 0.04)
```

We now check the Feller condition to see whether 0 is attainable. Recall from Proposition 2.1 of Andersen and Piterbarg above that 0 is attainable if  $2\lambda\bar{v}/\eta^2 < 1$

```
In [29]: feller <- function(params){2*params$lambda*params$vbar/(params$eta)^2}
        c(feller(paramsBCC), feller(paramsBCC1), feller(paramsBCC2))

        0.604865220249836  2.41946088099934  0.151216305062459
```

As we noted earlier, with both BCC and BCC2 parameters, zero is attainable.

## Summary of Lecture 3

- We derived a closed-form formula for European options in the Heston model and implemented it in R.
- We derived the Feynman-Kac formula.
- We explained why the choice of discretization of the continuous time stochastic process is critical for efficient simulation.
- We showed how further efficiency could potentially be achieved by using Richardson extrapolation.

In Lecture 4, we will code the Heston simulation and improve it further.

## References

1. <sup>△</sup> Albrecher, Hansjörg, Philipp Mayer, Wim Schoutens and Jurgen Tistaert, The Little Heston Trap, *Wilmott Magazine* 83–92 (January 2007).
2. <sup>△</sup> Bakshi, Gurdip, Charles Cao and Zhiwu Chen, Empirical performance of alternative pricing models. *Journal of Finance* **52**(5), 2003–2049 (1997).
3. <sup>△</sup> Cox, John, Jonathan Ingersoll and Stephen Ross, A theory of the term structure of interest rates. *Econometrica* **53**(2), 385–408 (1985).
4. <sup>△</sup> Duffie, Darrell, Jun Pan and Kenneth Singleton, Transform analysis and asset pricing for affine jump-diffusions, *Econometrica* **68**(6), 1343–1376 (2000).
5. <sup>△</sup> Jim Gatheral, *The Volatility Surface: A Practitioner's Guide*, John Wiley and Sons, Hoboken, NJ (2006).
6. <sup>△</sup> Jim Gatheral, Thibault Jaisson and Mathieu Rosenbaum, Volatility is rough, [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2509457](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2509457) ([http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2509457](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2509457)) (2014).
7. <sup>△</sup> Paul Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer-Verlag, Heidelberg (2004).
8. <sup>△</sup> Peter E. Kloeden and Eckhard Platen, *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, Heidelberg (1992).

9. ^ Roger Lord, Remmert Koekoek and Dick Van Dijk, A Comparison of biased simulation schemes for stochastic volatility models., *Quantitative Finance* **10**(2), 177–194 (2010).
10. ^Kahl, Christian and Peter Jäckel, Not-so-complex logarithms in the Heston model, *Wilmott Magazine*, 94–103 (September 2005).