Note #4: Stochastic Processes and Option Pricing

Wenhua Wang

Mar.20,2006

1. Heston Vanilla Option Model

Model description

Heston's model is based on the following equations, which represent the dynamics of the asset price and the variance processes under the risk-neutral measure:

$$\begin{aligned}\frac{\mathrm{d}S_t}{S_t} &= \mu \mathrm{d}t + \sqrt{v_t}\mathrm{d}X_t^{(1)} \\ \mathrm{d}v_t &= \kappa(\varphi - v_t)\mathrm{d}t + \xi\sqrt{v_t}\mathrm{d}X_t^{(2)}\end{aligned}$$

$$dX_t^{(1)} dX_t^{(2)} = r dt$$

basic share data			
drift	μ	initial volatility	$\sigma_0 := \sqrt{v_0}$
initial value	S_0	time (years)	T
parameters of volatility			
longterm average	$\sqrt{\varphi}$	Speed of mean reversion	κ
vol of vol	ξ	correlation	ρ

The critical problem in Monte Carlo simulation is to simulate two correlated Brownian motion. This can be done by Cholesky decomposition,

$$X_t^1 = f_t^1$$

$$X_t^2 = r f_t^1 + \sqrt{1-r^2} f_t^2$$

The Euler discretizations for the Heston processes are as following:

$$S_{t_i} = S_{t_{i-1}}\left[1 + m\Delta t + \sqrt{v_{t_{i-1}}}\left(r\Delta X_{t_i}^{(1)} + \sqrt{1-r^2}\Delta X_{t_i}^{(2)}\right)\right]$$

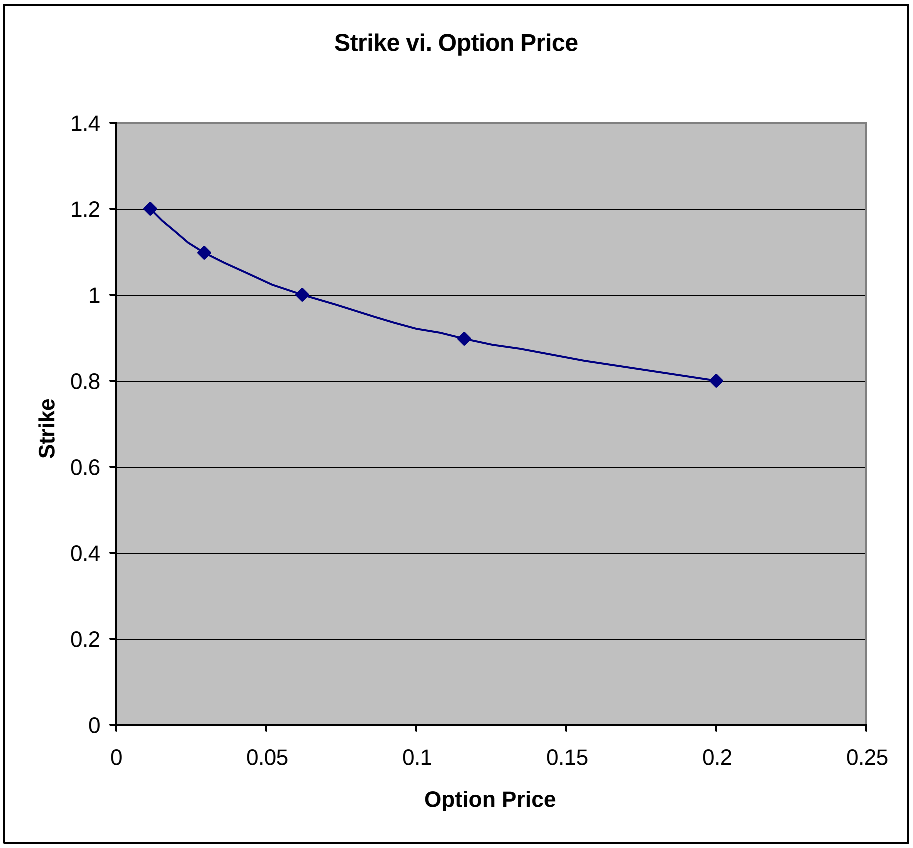
$$v_{t_i} = v_{t_{i-1}} + k\left(j - v_{t_{i-1}}\right)\Delta t + \sqrt{v_{t_{i-1}}} x\Delta X_{t_i}^{(1)}$$

European option simulation is relative simple; we just need to consider the payoff at expiry. But American option simulation is quite complicated. There are some research results on this subject, see [1].

Input							
Time to maturity		T	1				
Initial price		S0	1				
Initial voalitity		v0	0.04				
Longterm average		vbar	0.04				
mean-reverting speed		lambda	1.15				
Volatility of volatility		eta	0.39				
Correlation coefficient		rho	-0.64				
Strike Price		K	0.8	0.9	1	1.1	1.2
Number of time steps		M	150				
Number of simulation		N	30000				
Risk-free rate		r	0				
Dividend		div	0				

Simulation results:

Strike	0.8	0.9	1	1.1	1.2
Option Price	0.2	0.116	0.062	0.029	0.011
Implied Vol	0.10909	0.05250	0.04470	0.00172	0.00004



The results show that the higher the asset price the lower the volatility. The higher the strike the lower the option price given the asset price fixed.
The results agree with Schmalensee and Trippi's empirical study[9], that is a strong negative relationship between stock price changes and changes in implied volatility.

There are two critical technical problems encountered:

1) Underflow
   Some tricks can be used to resolve this problem. If is x is too small, we may not get expected value of $\sqrt{x}$, so we can try $\frac{\sqrt{10000x}}{100}$
2) The algorithm is critical in calculating implied volatilities. Usually people use simple binomial search algorithm. But this algorithm is not accurate, it often returns unexpected results. Here I use Newton-Raphson method[8].

Source code(please refer to "Notes on the source code organization" for detail):
HestonEuropeanMC.java
HestonMCTester.java

2. CEV (Constant Elasticity of Variance) Model

Model description

The constant elasticity of variance approach (CEV) assumes that in the risk-neutral world, the process of the stock price is:

$$dS = \mu S dt + \sigma S^{\alpha} dz$$

Compared to the geometric Brownian motion model risk-neutral process of:

$$dS = \mu S dt + \sigma S dz$$

Where μ is the drift rate [which in the risk-neutral world is given as risk free rate (r) - dividend yield (D)], σ is the volatility of the stock price, dz is a Wiener Process and α is a constant which takes on a value greater than 0.

Under the CEV approach, when α is given as 1, then the CEV collapses to the standard log-normal stock price process.

From the above given process, we can price European options in a similar fashion to that of the standard Black-Scholes, but instead of using a cumulative normal distribution, a non-central chi-squared distribution is used, with variables defined as below.

I order to find out the correct closed-form formula for European options under CEV, I did a lot of research. I found both from Hull[5] and Mark Schroder's original paper. None of them generated the expected results. For time limitation, I can only present the results I found here. Further research should be done in the future.

Formulae from Hull[5].

$$C_{CEV} = S_0 e^{-qT}[1 - c^2(a, b+2, c)] - Ke^{-rT} c^2(c, b, a)$$

$$P_{CEV} = Ke^{-rT}\left[1 - c^2(c, b, a)\right] - S_0 e^{-qT} c^2(a, b+2, c)$$

when $0 < a < 1$ and

$$C_{CEV} = S_0 e^{-qT}[1 - c^2(c, -b, a)] - Ke^{-rT} c^2(a, 2-b, c)$$

$$P_{CEV} = Ke^{-rT}\left[1 - c^2(a, 2-b, c)\right] - S_0 e^{-qT} c^2(c, -b, a)$$

when $a > 1$, where

$$a = \frac{K^{2(1-a)}}{(1-a)^2 s^2 T}, \quad b = \frac{1}{1-a}, \quad c = \frac{\left(S_0 e^{(r-q)T}\right)^{2(1-a)}}{(1-a)^2 s^2 T}$$

and $c^2(z,v,k)$ is the cumulative probability that a variable with a noncentral c^2 distribution with noncentrality parameter v and k degrees of freedom is less than z.

Mark Schroder's formula

In Mark Schroder's original paper "Computing the Constant Elasticity of Variance Option Pricing Formula". The formulas are as following.

$$C = S_t e^{-at} Q[2y;2+2/(2-b),2x] - Ee^{-rt}(1-Q[2x;2/(2-b),2y])$$

where a is the dividend yield, r is risk-free interest rate, t is $T-t$

$$k = \frac{2(r-a)}{d^2(2-b)[e^{(r-a)(2-b)t}-1]}$$

$$x = kS_t^{2-b} e^{(r-a)(2-b)t}$$

$$y = kE^{2-b}$$

$Q(z;,v,k)$ is Sankaren's approximating to the Nocentral Chi-Square Distribution, i.e.

$$Q(z;v,k) \sim f\left[\frac{1-hp[1-h+0.5(2-h)mp]-[\frac{z}{v+k}]^h}{h\sqrt{2p(1+mp)}}\right]$$

where

$$h = 1-(2/3)(v+k)(v+3k)(v+2k)^{-2}$$

$$p = \frac{v+2k}{(v+k)^2}$$

$$\mathrm{m} = (\mathrm{h}-1)(1-3\mathrm{h})$$

and f is the standard normal distribution function.

Theoretically, Hull's formula should return the same results as that of Schroder's given the same parameters (Schroder's beta=2*alpha of Hull).

However, non of them returned the expected results. I check the my Gamma functions results with R, it shows that it is correct. The problem might related to Chi-square approximation.

Monte Carlo simulation on CEV

First I'd point out that Monte Carlo for American options is not as easy as that for European options. Some scholars have done research on it. My program is based on Mark Broadie, Paul Glasserman's work[10]. The MC for American options is very slow because of the complexity.

The discretization is as following of the asset dynamic is as following:

$$S_i = S_{i-1}\left(1 + m\Delta t + s^{\,a}\,\Delta Z_i\right)$$

Test Results:

a) CEV MC on European Call, American Put and closed-form formulas

```
##################################################
#             CEV Model Test
##################################################
#                  Input
##################################################
K    =35.00000
T    =0.50000
S    =40.00000
Sig  =0.20000
r    =0.10000
div  =0.05000
Alpha=0.95000
##################################################
Closed Form European Call (Hull):-33.29303
--------------------------------------------------
Closed Form European Call (Mark Schroder):-10.96066
--------------------------------------------------
##################################################
European Call (MC)     :5.92190
--------------------------------------------------
Alpha=0.35000
Branches=5
Exercise Opportunities=4
American Put (MC)     : 0.32391
```

b) CEV MC on European Call vs. B-S

```
##################################################
#             CEV Model Test2
##################################################
#                  Input
```

```
##################################################
K    =50.00000
T    =1.00000
S    =50.00000
Sig  =0.40000
r    =0.06000
div  =0.00000
Alpha=1.00000
Nubmer of Steps =1000
Nubmer of Simulations=1000
#Monte Carlo for European Call
--------------------------------------------------
Option Price (MC)     :9.36112
--------------------------------------------------
Option Price (BS)     :9.23630
```

The results show that the MC on CEV is very close to that of B-S model.

Source Code(please refer to “Notes on the source code organization” for detail):

```
CEVAmericanMC.java
CEVEuropeanMC.java
CEVTester.java (test for part one of problem 2)
CEVTester2.java (test for part two of problem 2)
CEVCF.java (closed-form formula implementations)
```

3. Merton’s Jump Diffusion Model

In order to model fluctuations of the stock price process, Merton (1976) suggested a model in which the stock price following a geometric Brownian motion and a series of 'jumps' which assume are Poisson driven. You can picture each jump as a sudden movement in the stock price caused by any number of economic, industry or company factors. In addition to the standard pricing model, we define 2 additional variables in order to price under jumps:

1) The average number of jumps each year -- λ

2) The average jump size as a proportion of the stock price - or, alternatively the percentage of the stock volatility explained by the jumps -- γ

The stochastic process of the stock price is given as:

$$\frac{dS}{S} = (\mu - \lambda\gamma)dt + \sigma dz + dp$$

Where μ is the expected return of the stock, λ is the number of jumps per year and γ is the jump size as a proportion of the stock price. Furthermore, dp is the Poisson process which governs the jumps, dz is a Wiener process.

The respective call and put values respectively can be defined from the above equation to:

$$c_{jump} = \sum_{i=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^i}{i!} c_{BS}$$

and

$$p_{jump} = \sum_{i=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^i}{i!} p_{BS}$$

Where c_{BS} and p_{BS} are the respective Black-Scholes values for a European call and put option. We must make an adjustment to the volatility used to calculate these BS values as follows:

$$\sigma = \sqrt{(\sigma_o^2 - \lambda)\left(\frac{\gamma\sigma_o^2}{\lambda}\right) + \left(\frac{\gamma\sigma_o^2}{\lambda}\right)\frac{i}{T}}$$

Where σ_o is the observed volatility and lambda & gamma are the same as defined earlier.

The discretization is as following of the asset dynamic is as following:

$$S_i = S_{i-1}\left(1 + (m - l g)\Delta t + s\Delta Z_i + \Delta p\right)$$

The critical part in Monte Carlo simulation for the asset price is to simulate Δp. My understanding is that Δp is the contribution of the jumps to the asset price change. Obviously this should not simply be the number jumps during a small time interval Δt, instead it should be $g \times poisson(l \times \Delta t)$. Justin London approximates it by $\sqrt{\Delta t} \times poisson(l)$ (page 100 of [8]). I could not find the reason for it. So I implement Δp according to my own understanding.

The Poisson process was calculated by the following algorithm:

a) Set n=0, Tn=0
b) Generate the random deviate x from an exponential (l) distribution
c) Set n=n+1, $T_n = T_{n-1} \times x$
d) Repeat b) and c) until $T_n < e^{-l}$

Test Results:

```
##################################################
#           Merton Jump Test
```

```
##################################################
K     =80.00000
T     =0.50000
S     =100.00000
Sig   =0.25000
r     =0.08000
div   =0.00000
lambda=10.00000
gamma =0.25000
kapa  =0.02000
--------------------------------------------------
 Closed Form    | Monte Carlo
--------------------------------------------------
23.27726        | 23.71013
--------------------------------------------------
```

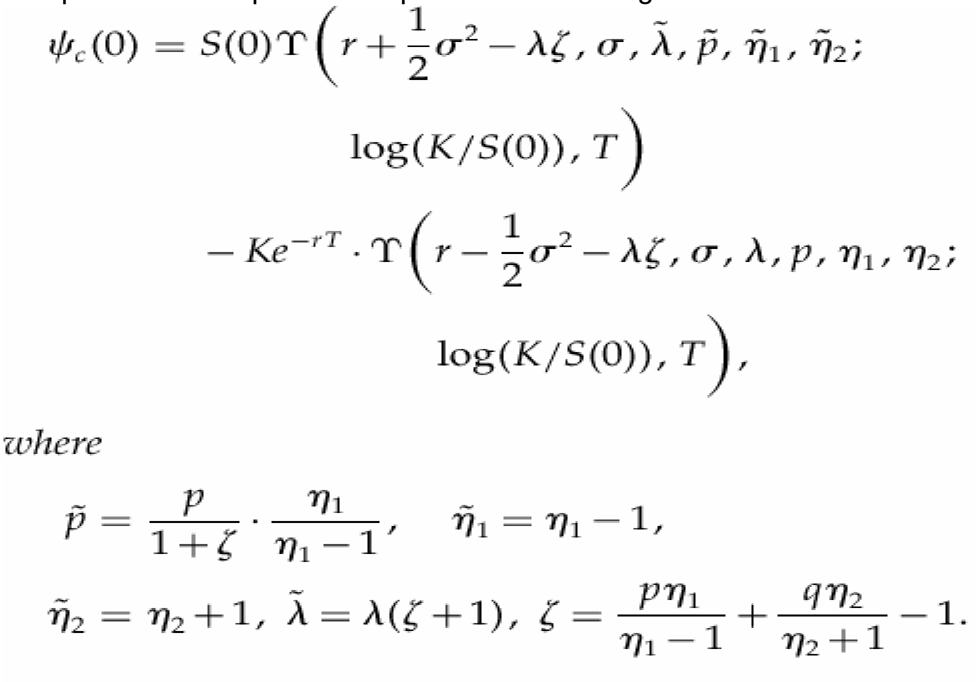
Source code(**please refer to “Notes on the source code organization” for detail**)**:**

```
JumpCFTester.java
MertonJumpCF.java (closed form)
MertonJumpEuropeanMC.java (Monte Carlo)
```

4. Kou’s Jump Diffusion Model

The Kou’s model consists of two parts, a continuous part modeled by a geometric Brownian motion, and a jump part, with the logarithm of the jump sizes having a double exponential distribution and the jump times corresponding to the event times of a Poisson process.

The price of a European Call option is as following:



$$\psi_c(0) = S(0)\Upsilon\Big(r + \frac{1}{2}\sigma^2 - \lambda\zeta, \sigma, \tilde{\lambda}, \tilde{p}, \tilde{\eta}_1, \tilde{\eta}_2; \log(K/S(0)), T\Big) - Ke^{-rT}\cdot\Upsilon\Big(r - \frac{1}{2}\sigma^2 - \lambda\zeta, \sigma, \lambda, p, \eta_1, \eta_2; \log(K/S(0)), T\Big),$$

where

$$\tilde{p} = \frac{p}{1+\zeta}\cdot\frac{\eta_1}{\eta_1 - 1}, \quad \tilde{\eta}_1 = \eta_1 - 1,$$

$$\tilde{\eta}_2 = \eta_2 + 1,\ \tilde{\lambda} = \lambda(\zeta + 1),\ \zeta = \frac{p\eta_1}{\eta_1 - 1} + \frac{q\eta_2}{\eta_2 + 1} - 1.$$

g function is defined as following:

$$\sum_{i=1}^{n} Y_i \overset{d}{=} \begin{cases} \sum_{i=1}^{k} \xi_i^{+}, & \text{with probability } P_{n,k}, k = 1, 2, \ldots, n \\ -\sum_{i=1}^{k} \xi_i^{-}, & \text{with probability } Q_{n,k}, k = 1, 2, \ldots, n \end{cases}$$

where $P_{n,k}$ and $Q_{n,k}$ are given by

$$P_{n,k} = \sum_{i=k}^{n-1} \binom{n-k-1}{i-k} \binom{n}{i} \cdot \left(\frac{\eta_1}{\eta_1+\eta_2}\right)^{i-k} \left(\frac{\eta_2}{\eta_1+\eta_2}\right)^{n-i} p^i q^{n-i},$$

$$1 \leq k \leq n-1,$$

$$Q_{n,k} = \sum_{i=k}^{n-1} \binom{n-k-1}{i-k} \binom{n}{i} \cdot \left(\frac{\eta_1}{\eta_1+\eta_2}\right)^{n-i} \left(\frac{\eta_2}{\eta_1+\eta_2}\right)^{i-k} p^{n-i} q^i,$$

$$1 \leq k \leq n-1, P_{n,n} = p^n, \quad Q_{n,n} = q^n,$$

For the time constraints, I only implemented the simplified Kou’s model. The critical part is the hh function.

Test Results:

```
##################################################
#             CEV Model Test2
##################################################
#                  Input
##################################################
S0    =100.00000
T     =0.50000
X     =90.00000
Sig   =0.16000
r     =0.05000
lambda=1.00000
kappa =0.40000
eta   =5.0
--------------------------------------------------
Price of a call: 24.25356246812068
Price of a put: 12.031454550670617
```

Source Code (please refer to “Notes on the source code organization” for detail):

KouCF.java
KouCFTester.java

Notes on the source code organization

There are four packages:

opt.util	Utilities, e.g. statistics functions
opt.test	Classes for testing purposes
opt.mc	Classes for Monte Carlo simulations
opt.formula	Classes for closed-form formulas

References:

[1] Broadie, M. and Glasserman, P. 1997: "Pricing American-Style Securities Using Simulation," Journal of Economics Dynamics and Control, 21, 1323-1352
[2] http://www.ijbe.org/table%20of%20content/pdf/vol4-2/vol4-2-05.pdf
[3] http://web.hku.hk/~jinzhang/ust/Schroder_JF.pdf
[4] Approximations to the non-central chi-square distribution
[5] www.stat.nctu.edu.tw/subhtml/E_source/teachers_eng/jclee/course/Chapter8.doc
[6] http://minorthird.sourceforge.net/
[7] Justin London, Modeling Derivatives in C++, John Wiley & Sons, Inc. 2005
[8] Bernt Arne Ødegaard, Financial Numerical Recipes in C ++,http://finance-old.bi.no/~bernt/gcc_prog/recipes/recipes/
[9] Richard Schmalensee and Robert R. Trippi. “Common Stock Volatility Expectations Implied by Option Premia” Journal of Finance 33 (March 1978), 129-47
[10]Mark Broadie, Paul Glasserman, “Pricing American-Style Securities using Simulation”
http://www1.gsb.columbia.edu/mygsb/faculty/research/pubfiles/1844/pricing%20securities%2Epdf