The Relationship Implied Volatility and Black Scholes using Markov Chain Monte Carlo Methods and Newton-Raphson Iteration

Travis Nestor

Abstract

The paper is a comprehensive analysis of the evaluating the Black-Scholes option formula via Markov Chain Monte Carlo process. Specifically, evaluating option prices and implied volatility. We will discuss its inlucding its methodology, functionality, and practical application. More specifically, how Monte Carlo valuation is best utilized in a financial aspect, comparing the theoretical components and translating using real world metrics, and explaining why it is the most efficient method.

1 Introduction

Markov Chain Monte Carlo (MCMC) methods are used to draw from a distribution that is an approximation to a target density. More specifically, they generate samples of an expectation of a specific function. A Markov chain decribes a time-series sequence that is either continuous or discrete that has an approximated distribution fitting a specified function.

The Black-Scholes model is designed to give the price of a European call option under certain assumptions. A European Call option allows one to trade a security at a predetermined time for a predetermined price, called the strike. It follows the assumptions that the underlying derrivative follows a geometric brownian motion process, there exists a risk-free interest rate, and the lack of transaction costs. To calculate implied volatility, the Newton-Raphson method will be used a parameter estimation method.

Using similar inputs as in the Black-Scholes equation, one can derrive the volatility parameter using Newton Raphson, a Maximum Likelihood Estimation. In doing so, one can properly parameterized a model, namely, Black-Scholes.

2 Mathematical Intuition

First we define the mathematical intution involved in MCMC methods, starting with geometric brownian motion used for finding the underlying asset price, $S(T)$. As μ is the drift term, we can set it equal to the risk-free interest rate r, and volatility as σ.

2.1 Asset Pricing

To estimate the price of the underlying asset, a stock, we model it as Geometric Bronian Motion given by the SDE:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW_t$$

by Ito's Lemma, this becomes:

$$S(T) = S(t)exp((\mu - 0.5\sigma^2)(T - t) + \sigma W_t)$$

where W_t is defined as Brownian motion.

2.2 Black-Scholes

The Black-Scholes valuation for call and put options where: T = option maturity, S = spot price of underlying asset, K = strike price, r = risk free interest rate, σ = asset volatility, and q = asset dividend. Also important is the concept of risk-neutral distribution. This describes a lognormal distribution with a volatility parameter σ, but μ is replaced with the risk-free rate. These factors modify the cumulative stanadard normal distribution to fit the model.

$$Call(S, t) = Se^{-q(T-t)}N[d_1] - Ke^{-r(t,T)}N[d_2]$$

$$Put(S, t) = Ke^{-r(t,T)}N[-d_2] - Se^{-q(T-t)}N[-d_1]$$

where

$$d_1 = \frac{log(S/K) + (r - q + 0.5\sigma^2)(T - t)}{\sigma\sqrt{T - t}};\ d_2 = d_1 - \sigma\sqrt{T - t}$$

and $N[.]$ describes the risk neutral distribution:

$$N(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-1/2x} dx$$

In the call option formula, we have $Se^{-q(T-t)}N[d_1]$ which denotes the present value of the underlying asset and $Ke^{-r(t,T)}N[d_2]$ denoting present value of exercising the option, both under the risk-neutral measure.

2.3 Theoretical MCMC

Our primary use of MCMC is to generate simulations from expectations of functions. These samples are used the gather data about a specific Markov Chain. To do so, we calculated the expected value of some function $f(x)$ where x is a random variable. After generating random values x_i and evaluating $f(x_i)$, we can take the average of the $f(x_i)$'s to get an accurate estimate provided i is sufficiently large to ensure accuracy.

As related to the Black-Scholes formula,$f(x_i)$ is our payoff of an option.

2.4 Implied Volatility via Newton-Raphson

If one looks at the valuation formulas for the call and put options (above) as functions of σ, they can calculate the applied volatility by solving the non-linear problem $f(\sigma) = 0$. Using this, we get:

$$0 = f(\sigma) = Se^{-q(T-t)}N[d_1] - Ke^{-r(t,T)}N[d_2] - Call(S, t)$$

and

$$0 = f(\sigma) = Ke^{-r(t,T)}N[-d_2] - Se^{-q(T-t)}N[-d_1] - Put(S,t)$$

where

$$d_1(\sigma) = \frac{log(S/K) + (r - q + 0.5\sigma^2)T}{x\sqrt(T)};\ d_2(\sigma) = \frac{log(S/K) + (r - q - 0.5\sigma^2)T}{x\sqrt(T)}$$

By differentiating $f(\sigma)$ with respect to σ, we get:

$$f'(\sigma) = \frac{1}{\sqrt{2\pi}}Se^{-qT}\sqrt(T)exp(-\frac{d_1(x)^2}{2})$$

Since $f'(\sigma) \geq 0$, we know $f(\sigma)$ is strictly increasing, and therefore has only one solution. Therefore we can use the Newton-Raphson Method to solve for

$$\sigma_{k+1} = \sigma_k - \frac{f(\sigma_k)}{f'(\sigma_k)}$$

Monte Carlo Simulation For simplicity, we will use only the Call option. While the algorithm can be used to for asset, we will use a derrivative with specific parameters

```
#s0= initial price, k = strike, r = risk free rate
#sigma = volatiltiy, T = maturity, q = dividend rate

#closed form
BSMC <- function(s0, k, r, sigma, T, q){

  d1 <- (log(s0/k) + (r - q + 0.5 * sigma^2)*(T)) / (sigma*sqrt(T))
  d2 <- d1 - (sigma*sqrt(T))

  call <- ((s0*pnorm(d1)) - (k*exp(-r*T)*pnorm(d2)))
  call
}
BSMC(25, 20, r=0.05, 0.36306, 1, 0)
```

```
## [1] 6.999979
```

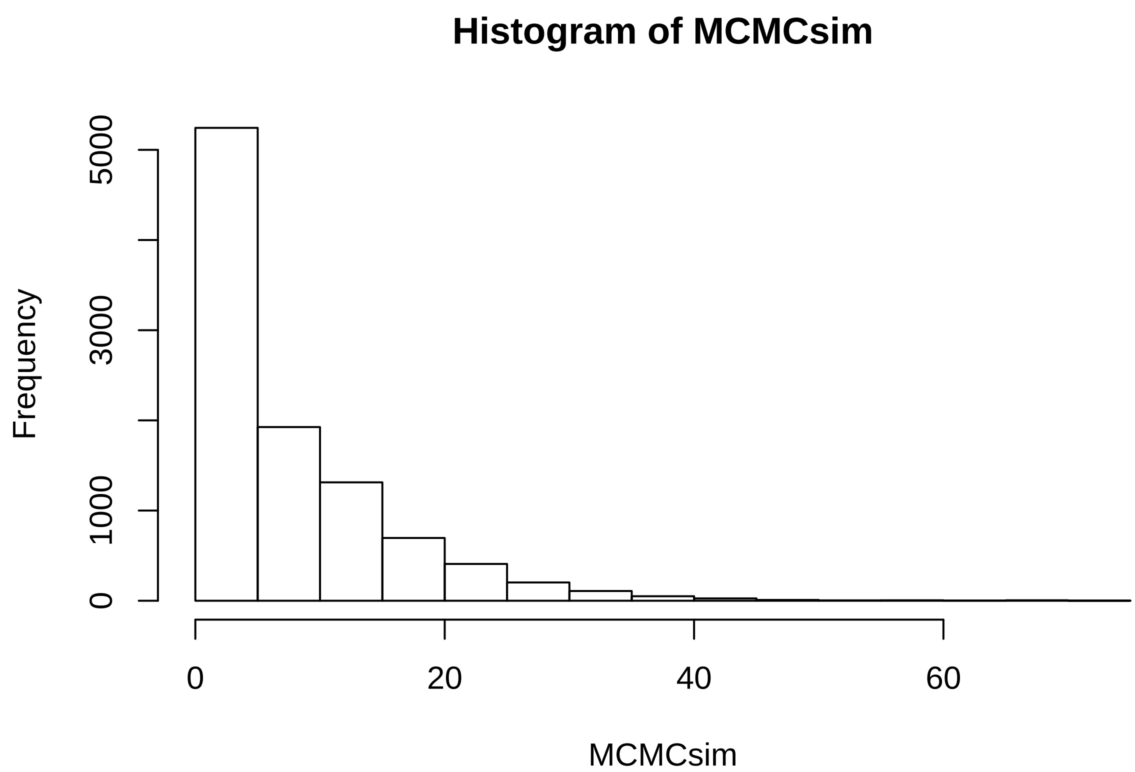
```
#Monte Carlo Method
mcbs <- function(s0, k, r, sigma, T, q){

  dt <- T/365  #daily steps
  t  <- seq(1/365, T, dt)   #At each time iteration for S(t)
  n  <- length(t)
  ST <- numeric(n)
  ST[1] <- s0
  delta <- 0
  for (i in 2:n){
    delta <- r*ST[i-1]*dt + sigma*ST[i-1]*rnorm(1, 0, sd=sqrt(dt))
    ST[i] <- ST[i-1]+delta
  }
  (max(ST[n] -k,0))*exp(-r*T)
```

```
}
MCMCsim <- replicate(10000,mcbs(25, 20, r=0.05, 0.36306, 1, 0))
callval <- mean(MCMCsim)
callval
```

```
## [1] 7.094794
```

```
hist(MCMCsim)
```



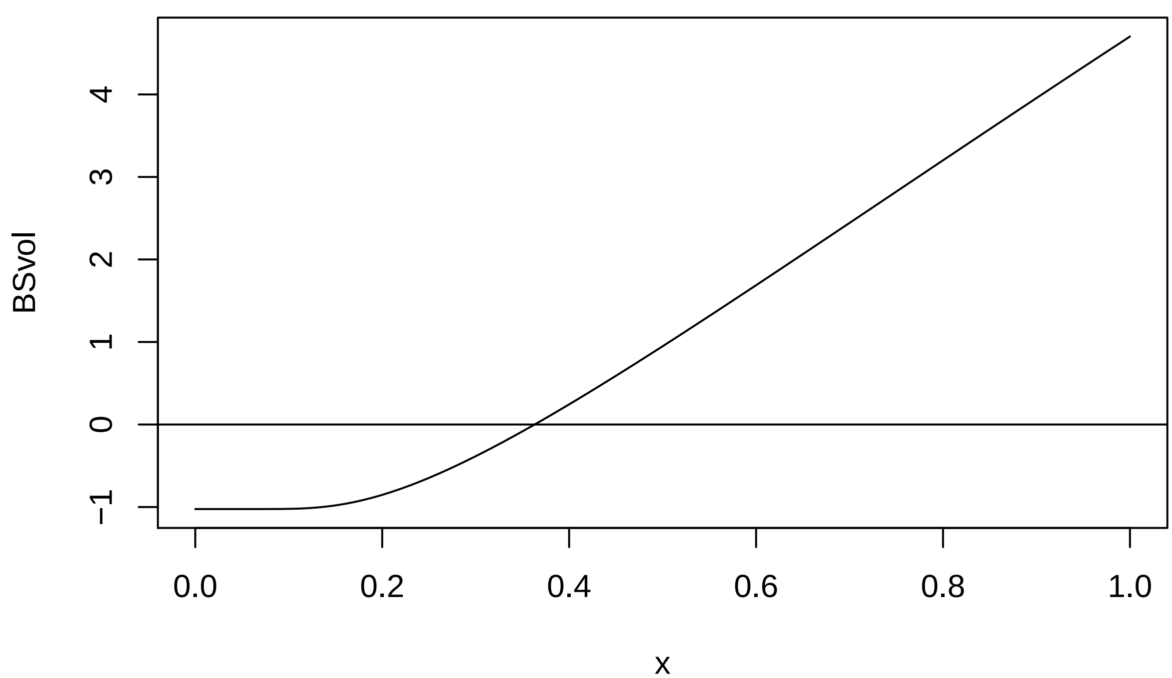
The blue line shows the average price calculated from the MC method, while the red was used to show the true value via the closed form formula.

2.5 Implied Volatility using Newton-Raphson

Using the same labels for the variables as in the Call Option pricing Algorithm:

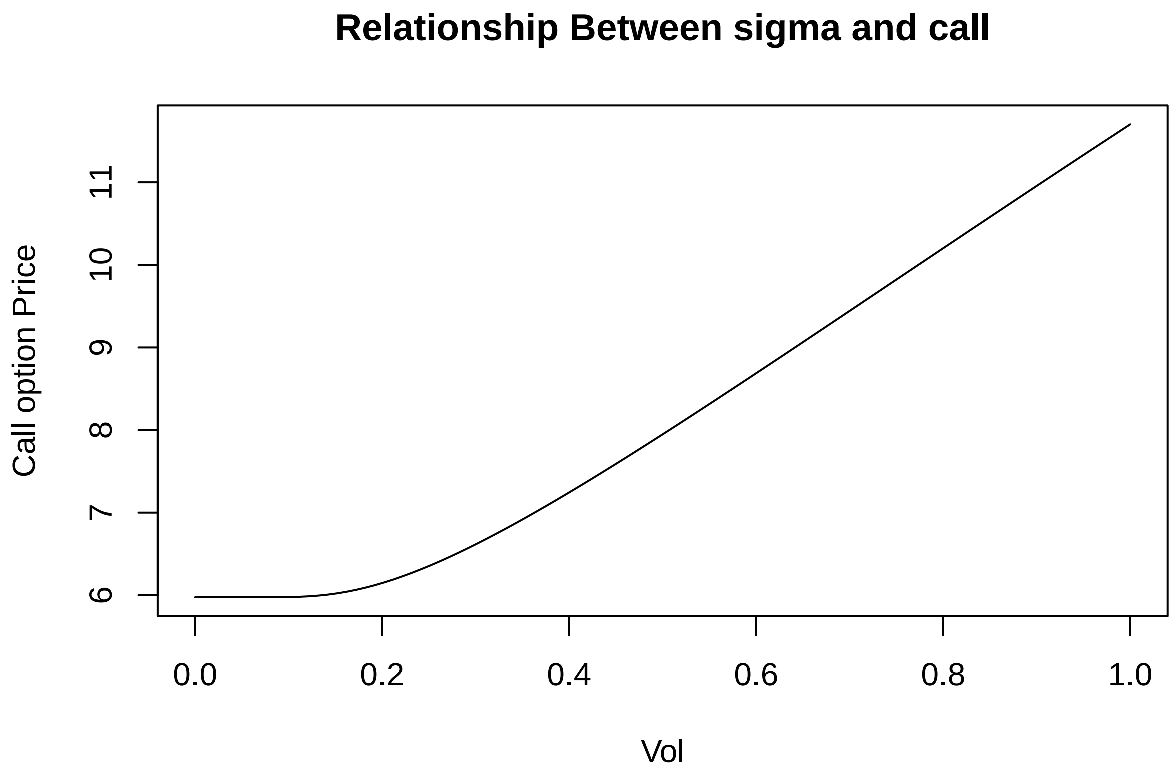
```
#Note: C = Price of a call option, S = spot rate of the underlying asset
newton <- function(C, S, K, T, q, r, n=100){
  tol <- 1e-7
  sig1 <- -.75   #Starting value for sigma
  sig  <- c(sig1)
  BSvol <- function(sigma) {
    nd1 <-  (log(S/K) + (r-q + 0.5*sigma^2)*T) / (sigma * sqrt(T))
    nd2 <-  (log(S/K) + (r-q - 0.5*sigma^2)*T) / (sigma * sqrt(T))
```

```
    S*exp(-q*T)*pnorm(nd1) - K*exp(-r*T)*pnorm(nd2) - C
  }
  BSvega <- function(sigma){
    nd1 <-  (log(S/K) + (r-q + 0.5*sigma^2)*T) / (sigma * sqrt(T))
    nd2 <-  (log(S/K) + (r-q - 0.5*sigma^2)*T) / (sigma * sqrt(T))
    (1 / sqrt(2*pi))*S*exp(-q*T)*sqrt(T)*exp(-(nd1^2/2))
  }
  for (i in 1:n){
    bb <- sig[i]
    sig[i+1] <- sig[i] - BSvol(sig[i]) / BSvega(sig[i])
    if (abs(sig[i+1] - sig[i]) <= tol){
      break
    }
    i <- i+1
  }
  plot(BSvol)
  abline(h=0)
sig
}
newton(7, 25, 20, 1, 0, 0.05, n=100)
```




```
## [1] -0.7500000  0.6792472  0.3775072  0.3632030  0.3630632  0.3630632
```

While the above plot shows volatility as a function with a specific solution, we can amend the algorithm to display how the price of the call impacts volatility.

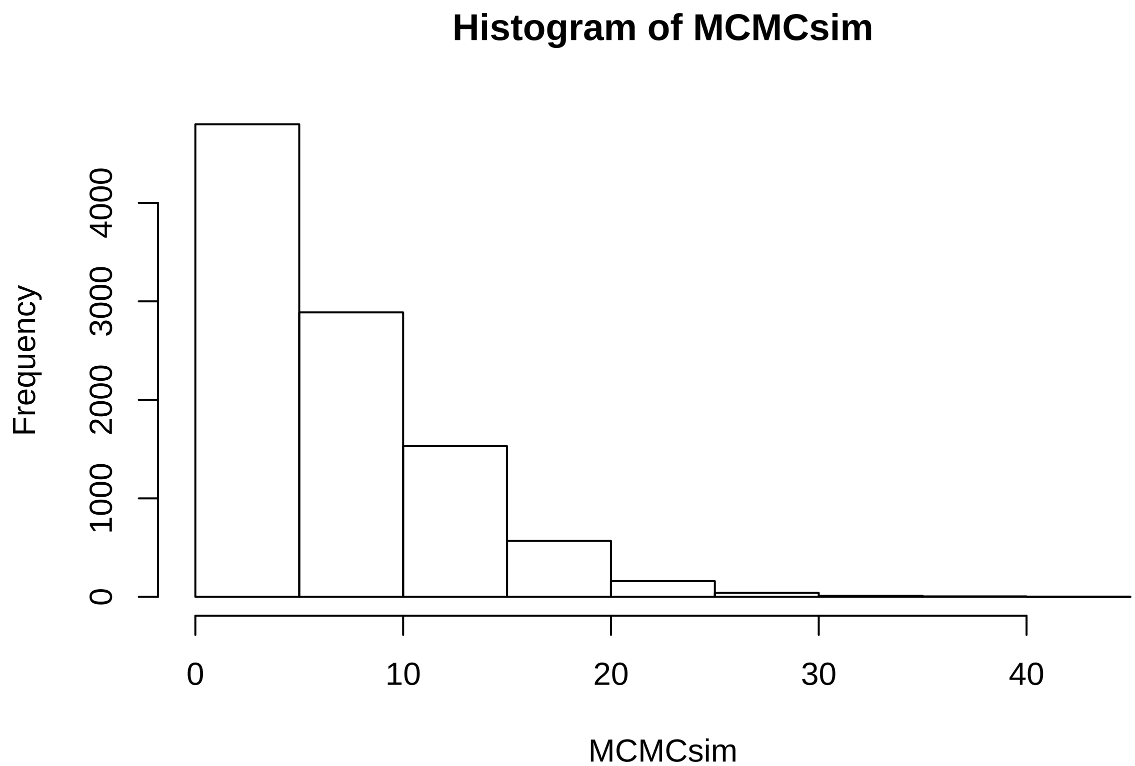


Conclusion

From the above data, there is an obvious positive correlation between implied volatility and the price of a call option. More specifically, the rate at which the price increases is slow until the maximum likely risk-neutral volatility measure is reached, then the correlaton becomes more positive.

Using either MCMC methods or simply using the closed formula, the option price can be calculated accurately for any given volatility. Recursively, given a option price, one can accurately measure the implied volatility via Newton-Raphson iterations. Where theese models separate is when one wishes to account for a non-constant volatility, It makes logical sense for the price of a call option to be less volatile as $t \to T$, as the random variables would seemingly have less of an impact on the underlying asset price.

```
## [1] 6.298986
```



In the above result, we see that having a time dependent σ lowers the price of the option. However, the accuracy comes called into question as it violates the rule of an independent random distribution. Furthermore, as we have shown above, the strike price has an impact on the estimation of implied volatility. Implied volatility generally is a convex function of strike price. This is important as while MCMC estimation can yield a new option price, the recursive form we had genereated before with Newton's method is no longer applicable.

3 References

Bunnin, et al. "Option Pricing under Model and Parameter Uncertainty Using Predictive Densities." Imperial College, 2000.

Givens, Geof, and Jennifer Hoeting. Computational Statistics. Wiley, 2013.

Montag, and Persson. MODEL RISK QUANTIFICATION IN OPTION PRICING. 2015, MODEL RISK QUANTIFICATION IN OPTION PRICING.

Pollard, Matthew. Markov Chain Monte Carlo Analysis of Option Pricing Models. 2007, Markov Chain Monte Carlo Analysis of Option Pricing Models.

Shreve, Steven. Stochastic Calculus for Finance. Springer, 2004.

Simkus, Darius. "Monte Carlo Pricing of American Style Options under Stochastic Volatility." Uppsala University.

Stefanica, Dan. A Primer for the Mathematics of Financial Engineering. FE Press, 2011.