Option Pricing Models

Rossano Giandomenico

Independent

September 2016

Online at https://mpra.ub.uni-muenchen.de/80540/
MPRA Paper No. 80540, posted 2 August 2017 09:15 UTC
Option Pricing Models

Rossano Giandomenico

Abstract: The study faces the problem of the skew for American and European options by using stochastic volatility and optimal stopping problem by simulating till Asian options in binomial model. The problem of local volatility is also faced with geometrical applications for option pricing with implications for the smile phenomenon.

Contents

Stochastic Differential Equations; 2
Interest Rate Models; 5
Caplet, Floorlet and Swap; 9
Monte Carlo Simulations; 11
European Options; 13
Stochastic Volatility; 23
Greeks; 33
Heston & Lewis Model: A Revision; 34
Levy Process; 59
American Options; 65
Binomial Model; 70
Asian Options; 76
Local Volatility; 80
Valuing Barrier Options; 83
Intensity Model; 84
References; 85
Stochastic Differential Equations

In finance Wiener process and geometric Brown process are largely used, the name came from George Brown in the 1827 that noted that the volatility of a small particle suspended in a liquid increases with the time, Wiener gave a mathematical formal assumption on the phenomena from this the term Wiener process. The main properties of the Wiener process is that it is a forward process such that we may integrate it although it is a function of infinite variation, the main idea is that the process converges to the discrete process because is the limit toward tends the discrete process when it is shared in sub intervals. From this we may approximate the Wiener process in the instant as \( \sqrt{dt} \), this permits to give the proof of Ito’s lemma by using Taylor series, indeed, the problem is more complicated because the diffusion process is a standardized normal distribution, this permits to give the solution to the Cauchy problem subject to boundary although the problem is parabolic. The geometric Brown process is used in finance to indicate a formal assumption for the dynamic of the prices that does not permit to assume negative value, formally we have:

\[
\frac{dS(t)}{S(t)} = \mu(t)dt + \sigma dWs
\]

Where \( \mu(t) \) denotes the drift of the distribution and it is the average in the \( dt \), \( \sigma \) denotes the volatility of the distribution and \( dWs \) denotes a Wiener process such that it may be decomposed by the following:

\[
dWs = N[0,1]\sqrt{dt}
\]

We may assume the following for the Wiener process:

\[
E \left[ \int_t^T dWs^2 \right] = T - t \Rightarrow dWs^2 \sim dt
\]

This means that a Wiener process is a forward process, the uncertainty is to the end of the process in \( T + dt \). From this we may obtain explanation of Ito’s lemma by using Taylor series, if we take a function of \( S \) as \( F(S) \) we may write Ito’s lemma in the following way:

\[
dF(S) = \frac{\partial F}{\partial t} dt + \frac{\partial F}{\partial S} dS + \frac{1}{2} \left( \frac{\partial^2 F}{\partial S^2} dS^2 + \frac{\partial^2 F}{\partial^2 t} dt^2 + 2 \frac{\partial^2 F}{\partial S \partial t} dt dS \right) + Q
\]

We may note that:

\[
dS^2 = \mu^2 S^2 dt^2 + 2 \mu \sigma S^2 dt^{3/2} + \sigma^2 S^2 dt
\]

\[
dS dt = \mu S dt^2 + \sigma S dt^{3/2}
\]

From this we obtain as \( dt \) tends to zero:
\[
dF(S) = \frac{\partial F}{\partial t} dt + \frac{\partial F}{\partial S} dS + \frac{1}{2} \left( \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 dt \right)
\]

By substituting \(dS\) we obtain Ito’s lemma:

\[
dF(S) = \left( \frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial S} S + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 \right) dt + \sigma \frac{\partial F}{\partial S} S dZ
\]

We may see now as to obtain the expected value of a normal distribution as such we have the following:

\[
\int f(z) = \int z f(z)
\]

As such we have the following:

\[
\int \frac{1}{\sigma \sqrt{2\pi}} z e^{-\frac{1}{2} z^2}
\]

This may be rewritten by:

\[
\int \frac{1}{\sigma} z \left( \frac{1}{2} \sigma \right) z^2 - \frac{1}{2} \sigma^2
\]

From this we may obtain explanation for Ito’s lemma, if we take a function of \(S\) as \(F(S)\) we may write Ito’s lemma in the following way:

\[
dF(S) = \left( \frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial S} S + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 \right) dt + \sigma \frac{\partial F}{\partial S} S dZ
\]

Where:

\[
\sigma dZ = N \left[ \frac{z - \mu}{\sigma \sqrt{dt}} ; \sigma^2 dt \right]
\]

As result:

\[
E [\sigma dZ] = 0
\]

Because:

\[
\int f(z) = \int \frac{z - \mu}{\sigma \sqrt{dt}} f(z) = 0
\]

Now we may analyze the following parabolic problem:

\[
\frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial S} S + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 = 0
\]

Subject to the following constraint:
\[ F(S(T)) = F(S) \]

The solution it is easy to solve, because if we take Ito’s lemma and we take the expectation we obtain that the solution to the parabolic problem is given by:

\[ F(S(T)) = E[F(S)] \]

As result we may rewrite Ito’s lemma in the following form:

\[ dF(S) = \left( \frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial S} S - \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 \right) dt - \sigma \frac{\partial F}{\partial S} S dZ \]

We may see now the solution of geometric Brown process:

\[ F(S) = \ln S(t) \]

\[ dF(S) = \frac{1}{S} dS + \frac{1}{2} \left( -\frac{1}{S^2} \right) dS^2 \]

Because:

\[ \frac{\partial F}{\partial t} = 0 \]

As such we have:

\[ dF(S) = \left( \mu(t) - \frac{1}{2} \sigma^2 \right) dt + \sigma dWs \]

\[ F(S(T)) = F(S(t)) + dF(S) \]

\[ \ln S(T) = \ln S(t) + \left( \mu(t) - \frac{1}{2} \sigma^2 \right) dt + \sigma dWs \]

\[ S(T) = S(t) e^{\left( \mu(t) - \frac{1}{2} \sigma^2 \right) dt + \sigma dWs} \]

As result we have:

\[ F(S(T)) - F(S(t)) = \int_t^T \left( \frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial S} S + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 \right) dt + \int_t^T \sigma \frac{\partial F}{\partial S} S dZ \]

\[ \mu F(S) = F(S(T)) - F(S(t)) = \mu \frac{\partial F}{\partial S} \Rightarrow \frac{\partial F}{\partial t} \sim \left( \frac{\partial^2 F}{\partial S^2} \right) \Rightarrow F(S(T)) = \int_t^T \frac{\partial^2 F}{\partial S^2} = \int \frac{\partial F}{\partial S} \]

The problem for option pricing became:

\[ \pm C(S,K,T) = \int_{-\infty}^{+\infty} \text{Max} \left\{ S(t) - Ke^{-\left( \left( \mu(t) - \frac{1}{2} \sigma^2 \right) T + \sigma N_\sqrt{T} \right)}; 0 \right\} f(z) dz \]
Interest Rate Models

The price of a zero coupon bond $P(T)$ is given by the following:

$$P(T) = E \left[ e^{-\int_t^T r(s)\,ds} \right]$$

Where $r(t)$ denotes the short rate that is given by the following stochastic differential equation:

$$dr(t) = \mu(t)dt + \sigma_r \, dW_r$$
$$r(t) = R(t)$$

As result by applying Ito’s lemma we have the following for the price of a zero coupon bond:

$$P(T) = e^{-(r(t)T + \mu(t)T^2)}$$

From this we may see that the internal compounded $R(T)$ interest rate is given by the following:

$$R(T) = r(t) + \mu(t)T$$

We may investigate the drift by using the forward process as result we have the following:

$$f(t, t) = -\frac{\partial \log P(T)}{\partial T}$$

By applying Ito’s lemma we have the following process:

$$d \log P(T) = \left( r(t) - \frac{1}{2} \sigma_r^2 T \right) dt - \sigma_r T dW_r$$

As such we have:

$$f(t, T) = r(t) + \sigma_r^2 T + \sigma_r N[0,1] \sqrt{T}$$

$$f(t, T) = r(t) + \sigma_r^2 T + \sigma_r^2 T$$
$$P(T) = e^{-\int_t^T f(t, T) \, dt}$$

As result we have:

$$R(T) = r(t) + \frac{1}{2} \sigma_r^2 T + \frac{1}{2} \sigma_r^2 T = r(t) + \sigma_r^2 T$$

This is the future value of the short rate in fact if take the average of $R(T)$ for each maturities we have that the risk free rate is given by:

$$P(T) = e^{-\int_t^T R(t) \, dt}$$
\[ r(T) = r(t) + \frac{1}{2} \sigma_r^2 T \]

That is the drift condition. In absence of arbitrage we have the following:

\[ P(T) = e^{-[R(T)T + VAR]} \]

\[ VAR = 2 \sigma_r^2 \int \text{Cov}(dW_r \, dW_r) = \sigma_r^2 T^2 \]

Because:

\[ -\log P(T) = R(T)T + \frac{\partial R(T)}{\partial T} = R(T)T + \partial \frac{2 \sigma_r^2 \int \text{Cov}(dW_r \, dW_r)}{\partial T} \]

\[ = R(T) + \partial \frac{2 \sigma_r^2 \int u \, du \, dt}{\partial T} = R(T)T + \sigma_r^2 T^2 \]

As result we have:

\[ r(T) = r(t) + \sigma_r^2 T + \sigma_r^2 T \]

Where \( \frac{1}{2} \sigma_r^2 T \) denotes the liquidity risk and \( \sigma_r^2 T \) denotes the risk premium. Now if we build a portfolio of default free bonds by shorting the bonds overvalued and acquiring the bonds undervalued we obtain a relation rule that the yields curve must respect given by the following:

\[ \gamma = \frac{R(T) - r(t)}{\sigma_r T} \]

From this we may derive that in absence of arbitrage opportunities we have by assumption the following:

\[ \gamma = \sigma_r \]

This is the risk premium requested by the markets, and it is a function of the risk associated with the volatility of the short rate. In absence of arbitrage opportunities the stochastic differential equation that a default free bond must satisfy is given by the following for \( P(T) = F(r,t) \):

\[ \frac{\partial F}{\partial t} + (\mu - \gamma \sigma_r) \frac{\partial F}{\partial \sigma_r} + \frac{1}{2} \frac{\partial^2 F}{\partial \sigma_r^2} \sigma^2 - r F(r,t) = 0 \]

The solution to this parabolic problem is given by the integrant factor, as such we have the following:

\[ F(r, t) = E(e^{-\int_0^T r(t) \, dt} \times 1) \]

Where:

\[ dr(T) = (\mu - \gamma \sigma_r) dt + \sigma_r \, dW_r \]

\[ r(t) = R(t) \]
Because $\mu = \sigma^2 r T + \sigma^2 T$ and $\gamma \sigma_r = \sigma^2 r$ as result we have that in absence of arbitrage opportunities the integral is equal to:

$$P(T) = e^{-\left[r(t)T + \frac{1}{2}\sigma^2 T^2\right]}$$

Now we may assume the following affine form that $F(r, t)$ must satisfy:

$$F(r, t) = A(t, T)e^{-B(t, T)r(t)}$$

Where:

$$dr(t) = (b - ar(t))dt + \sigma_r dWr$$

$$r(t) = R(t)$$

We may note that:

$$\mu(t) = b - ar(t)$$

At this point we may solve the stochastic differential equation:

$$A_t - rAB_t - AbB_t + ABar + \frac{1}{2} AB^2 \sigma^2 - rA = 0$$

Where:

$$A(T, T) = 1$$

$$B(T, T) = 0$$

$$A_t - ABb + \frac{1}{2} AB^2 \sigma^2 = 0$$

$$B_t + Ba - 1 = 0$$

From this we obtain:

$$B(t, T) = \frac{1 - e^{-a(T-t)}}{a}$$

$$A(t, T) = \exp \int_t^T Bb - \frac{1}{2} B^2 \sigma^2 = \exp \left[ \frac{(B(T, t) - T + t)(ab - \frac{1}{2} \sigma^2)}{a^2} - \sigma^2 \frac{B(t, T)^2}{4a} \right]$$

We may connect the model with the forward approach, as such we have the following:

$$R(T) = r(t) + \text{VAR} \left[ \int_t^T r(t)dt \right]$$
As such we have:

\[
VAR \left[ \int_t^T r(t) dt \right] = 2 \sigma_r^2 \int_t^T Cov \left( r(t) r(t) \right)
\]

\[
Cov \left( r(t) r(t) \right) = \sigma_r^2 \int_t^T e^{-2a t} dt = \frac{\sigma_r^2}{2a} \left( 1 - e^{-2a(T-t)} \right)
\]

Integrating twice we obtain:

\[
VAR \left[ \int_t^T r(t) dt \right] = \frac{\sigma_r^2}{a^2} \left[ (T-t) + 2 \left( \frac{e^{-a(T-t)} - 1}{a} \right) + \frac{\sigma_r^2}{2a} \left( 1 - e^{-2a(T-t)} \right) \right]
\]

We may now assume the following distribution for the short rate:

\[
dr(t) = (b - ar(t)) dt + \sqrt{r(t)} \sigma_r dW r
\]

\[r(t) = R(t)\]

At this point we may solve the stochastic differential equation:

\[A_t - rAB_t - AB_t r + AB_r + \frac{1}{2} AB^2 \sigma^2 - rA = 0\]

As such we have the following system:

\[A_t - ABb = 0\]

\[B_t - Ba - \frac{1}{2} B^2 \sigma^2 + 1 = 0\]

From this we may obtain the following solution:

\[B(t,T) = \frac{2 \left( e^{(T-t)} - 1 \right)}{(y + a) \left( e^{(T-t)} - 1 \right) + 2y}\]

\[A(t,T) = \left( \frac{2y e^{(y+a)(T-t)}}{(y + a) \left( e^{y(T-t)} - 1 \right) + 2y} \right)^{2ab} \sigma_r^2\]

\[y = a + 2\sigma\]

We may note that:

\[R(T) = - \frac{\ln A(t,T) - B(t,T) r(t)}{T - t}\]
Caplet, Floorlet and Swap

The price of Caplet and Floorlet may be derived directly from the arbitrage condition between Cap, Floor and Swap that is given by the following relation:

\[ \text{Cap} - \text{Floor} + \text{Swap} = \text{Fixed Income} \]

The three positions in Cap, Floor and Swap generates a fixed income that has to be equal to zero, as such to avoid arbitrage opportunities we must have the following prices for Caplet and Floorlet:

\[ \text{Caplet} = P(T)\left[ F(T)N[h1] - K N[h2] \right] \]
\[ \text{Floorlet} = P(T)\left[ K N[-h2] - F(T)N[-h1] \right] \]

Where:

\[ h1 = \frac{\ln \left( \frac{F(T)}{K} \right) + \frac{1}{2} \sigma_f^2 (T - t)}{\sigma_f \sqrt{T - t}} \]
\[ h2 = \frac{\ln \left( \frac{F(T)}{K} \right) - \frac{1}{2} \sigma_f^2 (T - t)}{\sigma_f \sqrt{T - t}} \]
\[ \sigma_f = \frac{\sigma_r}{F(T)} \]

\( F(T) \) denotes the Forward of the Swap rate given by:

\[ \ln \left( \frac{\exp \left( \text{Swap}(T)xt \right)}{\exp \left( \text{Swap}(t)xt \right)} \right) \]
\[ \frac{T - t}{T - t} \]

This may be considered the fair value as well but if we go in the OTC market we will not get the market prices because there isn’t arbitrage and the price of Cap and Floor are equals, this suggests that they are priced with a geometric martingale such that we have the following pricing formula:

\[ \text{Caplet} = P(T)\left[ r(t)N[h1] - K N[h2] \right] \]
\[ \text{Floorlet} = P(T)\left[ K N[-h2] - r(t)N[-h1] \right] \]

Where:

\[ h1 = \frac{\ln \left( \frac{r(t)}{K} \right) + \frac{1}{2} \sigma_r^2 T}{\sigma_r \sqrt{T}} \]
\[ h2 = \frac{\ln \left( \frac{r(t)}{K} \right) - \frac{1}{2} \sigma_r^2 T}{\sigma_r \sqrt{T}} \]
As result we may assume the following:

\[
\frac{dP(T)}{P(T)} = r(t) + \frac{1}{2} \sigma^2 dt - \sigma dW r
\]

The expectation of \(-\sigma dW r\) is equal to \(-\frac{1}{2} \sigma^2 dt\) as result we have the short rate \(r(t)\) as average, but if we simulate \(\frac{dP(T)}{P(T)}\) cannot assume negative value because is a geometric Brown process, as result we have that \(-\sigma dW r\) simulated is equal to \(\leq +\frac{1}{2} \sigma^2 dt\) that represents the liquidity risk. As result in absence of arbitrage opportunities we have that the yield curve is given by the following:

\[
r(T) \leq r(t) + \sigma_r^2 T
\]

Because if we apply regular Ito’s lemma, we have by simulating as well:

\[
\frac{dP(T)}{P(T)} = r(t) + \frac{1}{2} \sigma^2 dt - \frac{1}{2} \sigma^2 dt - \frac{2}{2} \sigma^2 dt
\]

Otherwise, if we consider the time decay we may simulate the following process:

\[
\log P(T) = r(t)T + \frac{1}{2} \sigma_r^2 T^2 - \sigma_r \sqrt{T} N(0,1) \sqrt{T}
\]

As result we have:

\[
r(T) \leq r(t) + \sigma_r^2 T
\]

From this we may obtain the price of a Put option maturing in \(T < S\), on the zero coupon bond \(P(S)\), as such by considering the time decay we have the following pricing formula:

\[
P(P(S), T, K) = KP(T)N[d1] - P(S)N[d2]
\]

Where:

\[
d1 = \ln \left( \frac{KP(T)}{P(S)} \right) + \frac{1}{2} \int \sigma_N^2 \sqrt{\int \sigma_N^2}
\]

\[
d2 = \ln \left( \frac{KP(T)}{P(S)} \right) - \frac{1}{2} \int \sigma_N^2 \sqrt{\int \sigma_N^2}
\]

\[
\int \sigma_N^2 = \int \sigma_r^2 + \sigma_T^2 - 2 \rho \sigma_S \sigma_r = \frac{1}{2} \sigma_r^2 S^2 + \frac{1}{2} \sigma_T^2 T^2 - 2 \rho \sigma_r^2 \frac{2}{3} S^2 \frac{2}{3} T^2
\]

\[
= 2 \rho \sigma_r^2 \frac{2}{3} S^2 \frac{3}{3} T^2 - \frac{1}{2} \sigma_r^2 S^2 - \frac{1}{2} \sigma_r^2 T^2
\]
Monte Carlo Simulations

We may obtain a normal distribution by doing the standardized normal inverse of a random (RND) that produce in a stochastic way a series of number from zero to one, in notation: $N^{-1}[RND]$. With this approach we obtain a normal distribution with zero average and one volatility with zero kurtosis. The lognormal distribution is obtained by doing the exponential of the normal distribution, in notation: $e^{N^{-1}[RND]}$, this produce a lognormal distribution with one volatility by applying Ito’s lemma and a very high kurtosis. This is a problem because it will infer problems in every kind of simulations by giving wrong result for the tails. Indeed, it is possible to eliminate the excess kurtosis, we may construct a lognormal distribution by doing the following: $round(1/N[RND]);round(1/N[RND+1]);round(1/N[RND+2]);- round(1/N[RND]); - round(1/N[RND +1]); - round(1/N[RND+2])$, from this approach we obtain a kind of normal distribution with zero average and 1.2 volatility and low kurtosis. Now if we take the exponential of this distribution we obtain a lognormal distribution that keeps the same average with one volatility and zero kurtosis. To run the simulation we may use the following VBA code:

```vba
Function sim(r, sigma, T, ITER)
    ITER = Application.Round(5 / T * iT, 0)
    iT = 180 for sigma = 0.04
    Dim Path() As Double, e() As Double
    simPath = 0
    ReDim Path(ITER * 6) As Double, e(ITER * 6) As Double
    For itcount = 1 To ITER
        e(itcount) = Application.Round(1 / Application.NormSDist(Rnd), 4)
        e(itcount) = Application.Round(1 / Application.NormSDist(Rnd + 1), 4)
        e(itcount) = Application.Round(1 / Application.NormSDist(Rnd + 2), 4)
        e(itcount) = Application.Round(-1 / Application.NormSDist(Rnd), 4)
        e(itcount) = Application.Round(-1 / Application.NormSDist(Rnd + 1), 4)
        e(itcount) = Application.Round(-1 / Application.NormSDist(Rnd + 2), 4)
        Path(itcount) = Exp( - sigma * Sqr(T) * e(itcount))
    Next itcount
    simPath = simPath + r + Path(ITER) / ITER
End Function
```
sim = simPath
End Function

Function Swap(r, sigma, T, ITER)
ITER = Application.Round(5 / T * iT, 0)

iT = 180 for sigma = 0.04

Dim Swapt() As Double

simPath = 0

ReDim Swapt(ITER) As Double

For itcount = 1 To ITER
    e = Application.NormSInv(Rnd)
    Swapt(itcount) = Exp(-(sigma * Sqr(T) * e))
Next itcount

simPath = simPath + Application.Average(Swapt(ITER))

Swap = r + simPath / ITER

End Function
European Options

The option pricing model is based on the arbitrage setting, the main idea is that the pay off and the price of the option may be replicated so its value is directly determinate to avoid arbitrage opportunity called hedging relation. Further application was about the dividend because when the stock pays the dividend its prices will decrease for the same amount. Now we assume the following distribution for the stock prices:

\[ \frac{dS(t)}{S(t)} = (r(t) - q) dt + \sigma S dW \]

$q$ denotes the dividend yield and $r(t)$ the risk free interest rate, This let us to introduce arbitrage theory, in practice if we built the following portfolio we have:

\[ V_t = \pm S \frac{\partial F(S)}{\partial S} - F(S) \]

The portfolio is risk free, as such by using Ito’s lemma we obtain the following stochastic differential equation:

\[ \frac{\partial F}{\partial t} + (r - q) \frac{\partial F}{\partial S} S + \frac{1}{2} \frac{\partial^2 F}{\partial^2 S} S^2 \sigma^2 - rF(S) = 0 \]

We may solve the stochastic differential equation by using the integrant factor:

\[ F(S) = Z(S)e^{-rT} \]

By solving the stochastic differential equation for $Z(S)$ we obtain that the solution is given by:

\[ Z(S) = E(S(T)) \]

By solving we obtain:

\[ F(S) = E(S(T)e^{-rT}) \]

This means that if we replicate the prices of options by using delta hedging the value of options are given by the expected value of the pay off discounted where the drift of the process is given by the short rate less the dividend yield, this is what it is called risk neutral world. The final pay off of a Call and Put option is given by the following:

\[ Call = Max[S(T) - K ; 0] \]
\[ Put = Max[K - S(T) ; 0] \]

The prices of the options are given by the expectation of the final pay off discounted:

\[ C(S,T,K) = P(T)E[Max(S(T) - K ; 0)] \]
\[ P(S,T,K) = P(T)E[Max(K - S(T) ; 0)] \]

Instead, we assume the following process for the default free zero coupon bond:

\[
\frac{dP(T)}{P(T)} = r(t) dt + \sigma_P dW_p
\]

\[ \sigma_P = \sigma_r T \]

Now to compute the value of the option is a problem because we have stochastic interest rate so the solution is to take the default free zero coupon bond as forward measure, so by using it as numeraire we have the following process:

\[
\frac{dN(t)}{N(t)} = -q \ dt + \sigma_N dW_n
\]

Where:

\[ N(t) = \frac{S(t)}{P(T)} \]

\[ \sigma_N^2 = \int_t^T \frac{\sigma_S^2 + \sigma_P^2 - 2\rho \sigma_S \sigma_P}{T-t} \ dt \]

Now we derive the price of a Call option, as such we have the following:

\[ \frac{C(S,T,K)}{P(T)} = \int_{-\infty}^{+\infty} \ Max \ [N(t)e^{-qT} \frac{1}{2} \sigma_N^2 T + \sigma_N \sqrt{T} - K ; 0] f(z)dz \]

The integral vanishes when \( N(T) < K \), thus by solving for \( z \) we have:

\[ z^* = \ln \left( \frac{K P(T)}{S} \right) + q T + \frac{1}{2} \sigma_N^2 T \]

As result we may rewrite the integral in the following form:

\[ \int_{z^*}^{+\infty} [N(t)e^{-qT} \frac{1}{2} \sigma_N^2 T + \sigma_N \sqrt{T}] f(z)dz \int_{z^*}^{+\infty} K f(z)dz \]

By using the symmetry property of a normal distribution we obtain the following pricing formula:

\[ \frac{C(S,T,K)}{P(T)} = N(t)e^{-qT} N[-z^* + \sigma_N \sqrt{T}] - KN[-z^*] \]

As result we obtain the following pricing formula for the options by using the respective numeraire:
\[ C(S, T, K) = S(t)e^{-qT} N[d1] - P(T)KN[d2] \]

Where:

\[ d1 = \ln\left(\frac{S}{KP(T)}\right) - qT + \frac{1}{2}\sigma^2 T \]
\[ d2 = \ln\left(\frac{S}{KP(T)}\right) - qT - \frac{1}{2}\sigma^2 T \]

\[ P(S, T, K) = P(T)KN[-d2] - S(t)e^{-qT} N[-d1] \]

Where:

\[ -d2 = \ln\left(\frac{KP(T)}{S}\right) + qT + \frac{1}{2}\sigma^2 T \]
\[ -d1 = \ln\left(\frac{KP(T)}{S}\right) + qT - \frac{1}{2}\sigma^2 T \]

We may note that between the two formulations there is a parity relation such that we have:

\[ P(S, T, K) - C(S, T, K) + S(t)e^{-qT} = P(T)K \]

Indeed, we have got the same formulation of Black, Scholes (1973) with the changes of measure and by considering that the dividend is income so to have the same final pay off in the hedge portfolio. Indeed, we may estimate the prices of the options by using a numerical procedure, as such we may rewrite the stochastic differential equation that an option must satisfy by using the following notation:

\[ S_i = ih \quad T_j = jdt \]
\[ \frac{\partial F}{\partial t} = \frac{C(i, j) - C(i, j - 1)}{dt} \]
\[ \frac{\partial F}{\partial S} = \frac{C(i + 1, j) - C(i, j)}{h} \]
\[ \frac{\partial^2 F}{\partial S^2} = \frac{C(i + 1, j) - 2C(i, j) - C(i - 1, j)}{h^2} \]

By substituting these values in the stochastic differential equation we obtain the following:

\[ aC(i - 1, j) + bC(i, j) + cC(i + 1, j) = (1 + r dt) C(i, j - 1) \]
Where:

\[ a = -\frac{1}{2} \sigma^2 i^2 dt \]
\[ b = 1 - \sigma^2 i^2 dt - dt r i \]
\[ c = dt r i + \frac{1}{2} \sigma^2 i^2 dt \]

As such for the Put options we have the following:

\[ S_i = \Delta h \quad T_j = jdt \]
\[ \frac{\partial F}{\partial t} = \frac{P(i, j) - P(i, j - 1)}{dt} \]
\[ \frac{\partial F}{\partial S} = \frac{P(i - 1, j) - P(i, j)}{h} \]
\[ \frac{\partial^2 F}{\partial S^2} = \frac{P(i - 1, j) - 2P(i, j) - P(i + 1, j)}{h^2} \]

By substituting these values in the stochastic differential equation we obtain the following:

\[ cP(i - 1, j) + bP(i, j) + aP(i + 1, j) = (1 + r dt) P(i, j - 1) \]

Where:

\[ c = \frac{1}{2} \sigma^2 i^2 dt + dt r i \]
\[ b = 1 - \sigma^2 i^2 dt - dt r i \]
\[ a = -\frac{1}{2} \sigma^2 i^2 dt \]

We assume the following:

\[ u = e^{\sigma \sqrt{\Delta t}} \quad d = e^{-\sigma \sqrt{\Delta t}} \]

As result the pays off are given by:

\[ C_{N,i} = \text{Max} \left[ Su^i d^{N-i} - K, 0 \right] \]
\[ P_{N,i} = \text{Max} \left[ K - Su^i d^{N-i}, 0 \right] \]

As result the prices for European options are given by:

\[ C_{j,i} = \left( \frac{1}{(1 + r dt)} \right) a C_{j+1,i-1} + b C_{j+1,i} + c C_{j+1,i+1} \]
\[ P_{j,i} = \left( \frac{1}{(1 + r \, dt)} \right) a \, P_{j+1,i+1} + b \, P_{j+1,i} + c \, P_{j+1,i-1} \]

We may compare now the model with the European Call options without dividend:

<table>
<thead>
<tr>
<th>( \sigma_S )</th>
<th>( K )</th>
<th>( r )</th>
<th>( S_r )</th>
<th>( T )</th>
<th>Numerical 3 Grids</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.25</td>
<td>0.52365</td>
<td>0.50747</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.25</td>
<td>0.27204</td>
<td>0.25747</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.25</td>
<td>0.02040</td>
<td>0.01796</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.5</td>
<td>0.52965</td>
<td>0.51489</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.5</td>
<td>0.27935</td>
<td>0.26489</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.5</td>
<td>0.02896</td>
<td>0.02793</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.75</td>
<td>0.53282</td>
<td>0.52225</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.75</td>
<td>0.28428</td>
<td>0.27225</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.75</td>
<td>0.03557</td>
<td>0.03666</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

We may compare now the model with the European Put options without dividend:

<table>
<thead>
<tr>
<th>( \sigma_S )</th>
<th>( K )</th>
<th>( r )</th>
<th>( S_r )</th>
<th>( T )</th>
<th>Numerical 3 Grids</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.25</td>
<td>0.00023</td>
<td>0.01049</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.25</td>
<td>0.23332</td>
<td>0.24253</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.25</td>
<td>0.48641</td>
<td>0.49253</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.5</td>
<td>0.00066</td>
<td>0.01305</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.5</td>
<td>0.22594</td>
<td>0.23511</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.5</td>
<td>0.47902</td>
<td>0.48511</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.75</td>
<td>0.00120</td>
<td>0.01442</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.75</td>
<td>0.22027</td>
<td>0.22775</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.75</td>
<td>0.47279</td>
<td>0.47775</td>
</tr>
</tbody>
</table>

The numerical results gives the proof that the solution to the partial differential equation is given by the solution of Black, Scholes (1973).
To run the simulation we used the following VBA code:

Function NumeriCallEuropean(Spot, k, T, r, sigma, n)
    Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(sigma * (dt ^ 0.5))
    d = 1 / u
    Dim S() As Double
    ReDim S(n + 1, n + 1) As Double
    For i = 1 To n + 1
        For j = i To n + 1
            S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
        Next j
    Next i
    Dim Opt() As Double
    ReDim Opt(n + 1, n + 1) As Double
    For i = 1 To n + 1
        Opt(i, n + 1) = Application.Max(S(i, n + 1) - k, 0)
    Next i
    Dim a() As Double, b() As Double, c() As Double
    ReDim a(n + 1) As Double, b(n + 1) As Double, c(n + 1) As Double
    For i = 1 To n + 1
        a(i) = (-0.5 * sigma ^ 2 * i ^ 2 * dt)
        b(i) = (1 - (sigma ^ 2 * i ^ 2 * dt) - (dt * r * i))
        c(i) = (dt * r * i + 0.5 * sigma ^ 2 * i ^ 2 * dt)
    Next i
For j = n To 1 Step -1
    For i = 2 To j
        Opt(i, j) = (1 / (1 + r * dt)) * (a(i) * Opt(i + 1, j + 1) + b(i) * Opt(i, j + 1) + c(i) * Opt(i - 1, j + 1))
    Next i
    NumeriCallEuropean = Opt(i, j)
Next j
End Function

Function NumeriPutEuropean(Spot, k, T, r, sigma, n)
Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(sigma * (dt ^ 0.5))
    d = 1 / u
Dim S() As Double
ReDim S(n + 1, n + 1) As Double
For i = 1 To n + 1
    For j = i To n + 1
        S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
    Next j
Next i
Dim Opt() As Double
ReDim Opt(n + 1, n + 1) As Double
For i = 1 To n + 1
    Opt(i, n + 1) = Application.Max(k - S(i, n + 1), 0)
Next i
Dim a() As Double, b() As Double, c() As Double
ReDim a(n + 1) As Double, b(n + 1) As Double, c(n + 1) As Double

For i = 1 To n + 1
    a(i) = (-0.5 * sigma ^ 2 * i ^ 2 * dt)
    b(i) = (1 - (sigma ^ 2 * i ^ 2 * dt) - (dt * r * i))
    c(i) = (dt * r * i + 0.5 * sigma ^ 2 * i ^ 2 * dt)
Next i

For j = n To 1 Step -1
    For i = 2 To j
        Opt(i, j) = (1 / (1 + r * dt)) * (a(i) * Opt(i - 1, j + 1) + b(i) * Opt(i, j + 1) + c(i) * Opt(i + 1, j + 1))
        NumeriPutEuropean = Opt(i, j)
    Next i
    Next j
End Function

Function ImplicitNumeriCallEuropean(Spot, k, T, r, sigma, n)
Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(sigma * (dt ^ 0.5))
    d = 1 / u

Dim S() As Double
ReDim S(n + 1, n + 1) As Double

For i = 1 To n + 1
    For j = i To n + 1
        S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
    Next j
Next i
End Function
Dim Opt() As Double
ReDim Opt(n + 1, n + 1) As Double
For i = 1 To n + 1
    Opt(i, n + 1) = Application.Max(S(i, n + 1) - k, 0)
Next i

Dim a() As Double, b() As Double, c() As Double
ReDim a(n + 1) As Double, b(n + 1) As Double, c(n + 1) As Double
For i = 1 To n + 1
    a(i) = (-0.5 * sigma ^ 2 * i ^ 2 * dt)
    b(i) = (1 - (sigma ^ 2 * i ^ 2 * dt) - (dt * r * i))
    c(i) = (dt * r * i + 0.5 * sigma ^ 2 * i ^ 2 * dt)
Next i

For j = n To 1 Step -1
    For i = 2 To j
        Opt(i, j) = (1 / (1 + r * dt)) * Opt(i, n + 1) / (a(i) + b(i) + c(i))
        ImplicitNumeriCallEuropean = Opt(i, j)
    Next i
Next j
End Function

Function ImplicitNumeriPutEuropean(Spot, k, T, r, sigma, n)
Dim dt As Double, u As Double, d As Double, p As Double

dt = T / n
u = Exp(sigma * (dt ^ 0.5))
d = 1 / u
Dim S() As Double

21
ReDim S(n + 1, n + 1) As Double

For i = 1 To n + 1
    For j = i To n + 1
        S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
    Next j
Next i

Dim Opt() As Double
ReDim Opt(n + 1, n + 1) As Double
For i = 1 To n + 1
    Opt(i, n + 1) = Application.Max(k - S(i, n + 1), 0)
Next i

Dim a() As Double, b() As Double, c() As Double
ReDim a(n + 1) As Double, b(n + 1) As Double, c(n + 1) As Double
For i = 1 To n + 1
    a(i) = (-0.5 * sigma ^ 2 * i ^ 2 * dt)
    b(i) = (1 - (sigma ^ 2 * i ^ 2 * dt) - (dt * r * i))
    c(i) = (dt * r * i + 0.5 * sigma ^ 2 * i ^ 2 * dt)
Next i

For j = n To 1 Step -1
    For i = 2 To j
        Opt(i, j) = (1 / (1 + r * dt)) * Opt(i, n + 1) / (a(i) + b(i) + c(i))
    Next i
    ImplicitNumeriPutEuropean = Opt(i, j)
Next j
Next j
End Function
Stochastic Volatility

The case of stochastic volatility may be viewed as Cauchy problem where the diffusion process of Ito’s lemma is a Bivariate standardized normal distribution, thus we may solve easily the problem of option pricing with stochastic volatility in risk neutral world by using the integrant factor. It is interesting to introduce the concept of stochastic volatility, as such we may write Ito’s lemma in the following form:

\[
dF(S) = \left( \frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial S} S + a \frac{\partial F}{\partial \sigma} + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 + \rho \sigma \delta S \frac{\partial F}{\partial S \delta \sigma} + \frac{1}{2} \frac{\partial^2 F}{\partial \sigma^2} \delta^2 \right) dt + \sigma_{xy} \frac{\partial F}{\partial S} SdZ
\]

Where \(dZ\) denotes a standardized Bivariate normal distribution with the following form:

\[
f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(z_x^2 + z_y^2 - 2\rho z_x z_y)}{2(1-\rho^2)}}
\]

Where:

\[
z_x = \frac{x - \mu_x}{\sigma_x} \quad z_y = \frac{y - \mu_y}{\sigma_y}
\]

\[
f(x, y = y_0) = N[\mu_{x,y=y_0}, \sigma_{x,y=y_0}]
\]

\[
\mu_{x,y=y_0} = \mu_x + \rho \sigma_x \frac{(y - \mu_y)}{\sigma_y}
\]

\[
\sigma_{x,y=y_0} = \sigma_x \sqrt{(1 - \rho^2)}
\]

The PDE that an option must satisfy by assuming stochastic volatility is given by the following:

\[
\frac{\partial F}{\partial t} + r \frac{\partial F}{\partial S} S + k(\alpha - \gamma \sigma(t)^2) \frac{\partial F}{\partial \sigma} + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 + \rho \sigma \delta S \frac{\partial F}{\partial S \delta \sigma} + \frac{1}{2} \frac{\partial^2 F}{\partial \sigma^2} \delta^2 \sigma^2 - r F(S) = 0
\]

Where:

\[
\frac{dS(t)}{S(t)} = r \ dt + \sqrt{\sigma(t)^2} dW_S
\]

\[
d\sigma(t)^2 = k(\alpha - \gamma \sigma(t)^2) dt + \delta \sqrt{\sigma(t)^2} dW_\sigma
\]

\[
dW_S dW_\sigma = \rho \ dt
\]

The solution it is easy to solve, because if we take Ito’s lemma and we take the expectation we obtain that the solution to the parabolic problem is given by the following by using the integrant factor \(e^{-rT}\):

\[
F(S(T)) = e^{-rT} E [F(S, \sigma)]
\]
The final pay off of a Call and Put option is given, respectively, by the following:

\[
Call = \text{Max}[S(T) - K ; 0] \\
Put = \text{Max}[K - S(T) ; 0]
\]

The prices of the options are given by the expectation of the final pay off discounted for the Call options and Put options, respectively:

\[
C(S, T, K) = P(T)E[\text{Max}(S(T) - K ; 0)] \\
P(S, T, K) = P(T)E[\text{Max}(K - S(T) ; 0)]
\]

Now to compute the value of the option is a problem because we have stochastic interest rate so the solution is to take the default free zero coupon bond as forward measure. We assume the following process for the default free zero coupon bond:

\[
\frac{dP(T)}{P(T)} = r(t)dt + \sigma_p dWp
\]

\[
\sigma_p = \sigma_r T
\]

As result we obtain the following pricing formula for the options by using the respective numeraire:

\[
C(S, T, K) = S(t)e^{\rho \frac{k(y\sigma(t)^2 - a)}{\delta} T}[d1] - P(T)KN[d2]
\]

Where:

\[
\mu_{x,y=y_0} = r(t) + \rho \sqrt{\sigma(t)^2} \frac{k(y\sigma(t)^2 - a)}{\delta \sqrt{\sigma(t)^2}}
\]

\[
\sigma_{x,y=y_0} = \sqrt{\sigma_N^2(1 - \rho^2)}
\]

\[
\sigma_N^2 = \int_t^T \sigma(t)^2 + \sigma_p^2 - 2 \rho \sigma(t)^2 \sigma_p dt \\
T - t
\]

\[
d1 = \ln \left( \frac{S}{KP(T)} \right) + \rho \frac{k(y\sigma(t)^2 - a)}{\delta} T + \frac{1}{2} \left(1 - \rho^2\right)\sigma_N^2 T
\]

\[
\sqrt{1 - \rho^2}\sigma_N \sqrt{T}
\]
\[ d2 = \frac{\ln \left( \frac{S}{KP(T)} \right) + \rho \frac{k(y(t)^2 - a)}{\delta} T - \frac{1}{2} (1 - \rho^2) \sigma_N^2 T}{\sqrt{(1 - \rho^2) \sigma_N \sqrt{T}}} \]

\[ P(S, T, K) = KP(T)N[d1] - S(t)e^{\rho \frac{k(y(t)^2 - a)}{\delta} T} N[d2] \]

Where:

\[ d1 = \frac{\ln \left( \frac{KP(T)}{S} \right) - \rho \frac{k(y(t)^2 - a)}{\delta} T + \frac{1}{2} (1 - \rho^2) \sigma_N^2 T}{\sqrt{(1 - \rho^2) \sigma_N \sqrt{T}}} \]

\[ d2 = \frac{\ln \left( \frac{KP(T)}{S} \right) - \rho \frac{k(y(t)^2 - a)}{\delta} T - \frac{1}{2} (1 - \rho^2) \sigma_N^2 T}{\sqrt{(1 - \rho^2) \sigma_N \sqrt{T}}} \]

**Numerical Results**

We may compare the model with Black, Scholes (1973), as results we have the following figures for Call options:

1.2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>1</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (T - t)</td>
<td>2</td>
</tr>
<tr>
<td>Rho ((\rho))</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa ((\kappa))</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta ((\theta))</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda ((\lambda))</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance ((\sigma))</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance ((\nu))</td>
<td>0.01</td>
</tr>
<tr>
<td>Current variance ((\nu))</td>
<td>0.01</td>
</tr>
<tr>
<td>Bivariate Call Price</td>
<td>0.0970</td>
</tr>
<tr>
<td>Black Scholes Call Price</td>
<td>0.0887</td>
</tr>
</tbody>
</table>

For rational value of parameters the Bivariate converges to Black, Scholes (1973)
1.2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>1.5</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (T – t)</td>
<td>2</td>
</tr>
<tr>
<td>Rho (ρ)</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa (κ)</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta (θ)</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda (λ)</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance (σ)</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Bivariate Call Price</td>
<td>0.5885</td>
</tr>
<tr>
<td>Black Scholes Call Price</td>
<td>0.5583</td>
</tr>
</tbody>
</table>

The Bivariate approach permits to capture the skew for options deep in the money.

1.3

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>2</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (T – t)</td>
<td>2</td>
</tr>
<tr>
<td>Rho (ρ)</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa (κ)</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta (θ)</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda (λ)</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance (σ)</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Bivariate Call Price</td>
<td>1.0986</td>
</tr>
<tr>
<td>Black Scholes Call Price</td>
<td>1.0582</td>
</tr>
</tbody>
</table>

The Bivariate approach permits to capture the skew for options deep in the money. We may compare the result for European Put options by setting a positive correlation coefficient:

2.1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>1</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (T – t)</td>
<td>2</td>
</tr>
<tr>
<td>Rho (ρ)</td>
<td>0.5</td>
</tr>
<tr>
<td>Kappa (κ)</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta (θ)</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda (λ)</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance (σ)</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Bivariate Put Price</strong></td>
<td>0.0302</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>Black Scholes Put Price</strong></td>
<td>0.0305</td>
</tr>
</tbody>
</table>

For rational value of parameters the Bivariate converges to Black, Scholes (1973)

2.2

<table>
<thead>
<tr>
<th>Paramter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price ((S))</td>
<td>0.75</td>
</tr>
<tr>
<td>Strike Price ((K))</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate ((r))</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity ((T - t))</td>
<td>2</td>
</tr>
<tr>
<td>Rho ((\rho))</td>
<td>0.5</td>
</tr>
<tr>
<td>Kappa ((\kappa))</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta ((\theta))</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda ((\Lambda))</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance ((\sigma))</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance ((v))</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Bivariate Put Price</strong></td>
<td>0.2074</td>
</tr>
<tr>
<td><strong>Black Scholes Put Price</strong></td>
<td>0.1945</td>
</tr>
</tbody>
</table>

The Bivariate approach permits to capture the skew for options deep in the money.

2.3

<table>
<thead>
<tr>
<th>Paramter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price ((S))</td>
<td>0.5</td>
</tr>
<tr>
<td>Strike Price ((K))</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate ((r))</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity ((T - t))</td>
<td>2</td>
</tr>
<tr>
<td>Rho ((\rho))</td>
<td>0.5</td>
</tr>
<tr>
<td>Kappa ((\kappa))</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta ((\theta))</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda ((\Lambda))</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance ((\sigma))</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance ((v))</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Bivariate Put Price</strong></td>
<td>0.4517</td>
</tr>
<tr>
<td><strong>Black Scholes Put Price</strong></td>
<td>0.4418</td>
</tr>
</tbody>
</table>

The Bivariate approach permits to capture the skew for options deep in the money. Alternatively, we may simulate the following process:

\[
\frac{dS(t)}{S(t)} = \left[ r(t) + \rho \frac{k(y\sigma(t)^2 - a)}{\delta} \right] dt - \frac{1}{2} \sigma^2 (1 - \rho^2) dt + \sqrt{(1 - \rho^2)} \sigma dW_s
\]
Where:

\[ d\sigma(t)^2 = k(\alpha - \gamma \sigma(t)^2) dt + \delta \sqrt{\sigma(t)^2} dW_\sigma \]

\[ dW_S dW_\sigma = \rho \ dt \]

We may compare the result for European Call options:

1.1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>1</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Rho ((\rho))</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa ((\kappa))</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta ((\Theta))</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda ((\lambda))</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance ((\sigma))</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance ((v))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
<tr>
<td><strong>Bivariate Call Price</strong></td>
<td>0.0978</td>
</tr>
<tr>
<td><strong>Black Scholes Call Price</strong></td>
<td>0.0887</td>
</tr>
</tbody>
</table>

Monte Carlo simulations converge to the same result of closed form solution.

1.2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>1.5</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Rho ((\rho))</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa ((\kappa))</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta ((\Theta))</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda ((\lambda))</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance ((\sigma))</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance ((v))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
<tr>
<td><strong>Bivariate Call Price</strong></td>
<td>0.5841</td>
</tr>
<tr>
<td><strong>Black Scholes Call Price</strong></td>
<td>0.5583</td>
</tr>
</tbody>
</table>

Monte Carlo simulations converge to the same result of closed form solution.
1.3

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spot Price (S)</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Strike Price (K)</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Risk Free Rate (r)</strong></td>
<td>0.03</td>
</tr>
<tr>
<td><strong>Time to Maturity (Days)</strong></td>
<td>730</td>
</tr>
<tr>
<td><strong>Rho (ρ)</strong></td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Kappa (κ)</strong></td>
<td>0.2</td>
</tr>
<tr>
<td><strong>Theta (θ)</strong></td>
<td>0.03</td>
</tr>
<tr>
<td><strong>Lambda (λ)</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Volatility of Variance (σ)</strong></td>
<td>0.1</td>
</tr>
<tr>
<td><strong>Current variance (v)</strong></td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Number of Simulations</strong></td>
<td>1.000</td>
</tr>
<tr>
<td><strong>Bivariate Call Price</strong></td>
<td>1.0946</td>
</tr>
<tr>
<td><strong>Black Scholes Call Price</strong></td>
<td>1.0582</td>
</tr>
</tbody>
</table>

Monte Carlo simulations converge to the same result of closed form solution. We may compare the result for European Put options by setting a positive correlation coefficient:

2.1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spot Price (S)</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Strike Price (K)</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Risk Free Rate (r)</strong></td>
<td>0.03</td>
</tr>
<tr>
<td><strong>Time to Maturity (Days)</strong></td>
<td>730</td>
</tr>
<tr>
<td><strong>Rho (ρ)</strong></td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Kappa (κ)</strong></td>
<td>0.2</td>
</tr>
<tr>
<td><strong>Theta (θ)</strong></td>
<td>0.03</td>
</tr>
<tr>
<td><strong>Lambda (λ)</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Volatility of Variance (σ)</strong></td>
<td>0.1</td>
</tr>
<tr>
<td><strong>Current variance (v)</strong></td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Number of Simulations</strong></td>
<td>1.000</td>
</tr>
<tr>
<td><strong>Bivariate Put Price</strong></td>
<td>0.0305</td>
</tr>
<tr>
<td><strong>Black Scholes Put Price</strong></td>
<td>0.0305</td>
</tr>
</tbody>
</table>

Monte Carlo simulations converge to the same result of closed form solution.

2.2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spot Price (S)</strong></td>
<td>0.75</td>
</tr>
<tr>
<td><strong>Strike Price (K)</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Risk Free Rate (r)</strong></td>
<td>0.03</td>
</tr>
<tr>
<td><strong>Time to Maturity (Days)</strong></td>
<td>730</td>
</tr>
<tr>
<td><strong>Rho (ρ)</strong></td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Kappa (κ)</strong></td>
<td>0.2</td>
</tr>
<tr>
<td><strong>Theta (θ)</strong></td>
<td>0.03</td>
</tr>
<tr>
<td><strong>Lambda (λ)</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Volatility of Variance (σ)</strong></td>
<td>0.1</td>
</tr>
</tbody>
</table>
Monte Carlo simulations converge to the same result of closed form solution.

2.3

Monte Carlo simulations converge to the same result of closed form solution.

**Appendix**

To run the simulation we used the following VBA code

```vba
Option Base 1

' Bivariate Call & Put Price by Monte Carlo Simulation

Function BivariateMCCall(kappa, theta, lambda, rho, sigmav, daynum, startS, r, startv, K, ITER)
    Dim allS() As Double
    simPath = 0
    ReDim allS(daynum) As Double
    deltat = (1 / 365)
    For itcount = 1 To ITER
```
lnSt = Log(startS)
lnvt = Log(startv)
curv = startv
curS = startS

For daycnt = 1 To daynum
    e = Application.NormSInv(Rnd)
eS = Application.NormSInv(Rnd)
ev = rho * eS + Sqr(1 - rho ^ 2) * e

    'update the stock price
    lnSt = lnSt + (r + (rho * kappa * (lambda * curv - theta) / sigmav) - 0.5 * startv * (1 - rho ^ 2)) * deltat + Sqr(startv) * Sqr(1 - rho ^ 2) * Sqr(deltat) * eS
    curS = Exp(lnSt)

    lnvt = lnvt + (kappa * theta - (kappa + lambda) * curv) * deltat + Sqr(curv) * sigmav * Sqr(deltat) * ev
    curv = Exp(lnvt)

    allS(daycnt) = curS

Next daycnt

simPath = simPath + Exp((-daynum / 365) * r) * Application.Max(allS(daynum) - K, 0)

Next itcount

BivariateMCCall = simPath / ITER

End Function
For itcount = 1 To ITER

lnSt = Log(startS)
lnvt = Log(startv)
curv = startv
curS = startS

For daycnt = 1 To daynum
    e = Application.NormSInv(Rnd)
eS = Application.NormSInv(Rnd)
ev = rho * eS + Sqr(1 - rho ^ 2) * e

    'update the stock price
    lnSt = lnSt + (r + (rho * kappa * (lambda * curv - theta) / sigmav) - 0.5 * startv * (1 - rho ^ 2)) * deltat + Sqr(startv) * Sqr(1 - rho ^ 2) * Sqr(deltat) * eS
    curS = Exp(lnSt)
    lnvt = lnvt + (kappa * theta - (kappa + lambda) * curv) * deltat + Sqr(curv) * sigmav * Sqr(deltat) * ev
    curv = Exp(lnvt)
    allS(daycnt) = curS

Next daycnt

simPath = simPath + Exp((-daynum / 365) * r) * Application.Max(K - allS(daynum), 0)

Next itcount

BivariateMCPut = simPath / ITER

End Function
Now it is interesting to give the sensitivity analysis of European options with respect their parameters, as such we have the following:

\[
\frac{\partial C}{\partial S} = e^{-qT} N[d1], \quad \frac{\partial P}{\partial S} = -e^{-qT} N[-d1]
\]

\[
\frac{\partial^2 C}{\partial^2 S} = \frac{\partial^2 P}{\partial^2 S} = \frac{e^{-qT} N'[d1]}{S \sigma_N \sqrt{T}} = \frac{P(T) K N'[d2]}{e^{-qT} S^2 \sigma_N \sqrt{T}}
\]

\[
\frac{\partial C}{\partial \sigma} = \frac{\partial P}{\partial \sigma} = e^{-qT} S N'[d1] \sqrt{T} = P(T) K N'[d2] \sqrt{T}
\]

\[
\frac{\partial^2 C}{\partial^2 \sigma} = \frac{\partial^2 P}{\partial^2 \sigma} = e^{-qT} S N'[d1] \sqrt{T} \frac{d1d2}{\sigma_N}
\]

\[
\frac{\partial C}{\partial T} = P(T) K T N(d2), \quad \frac{\partial P}{\partial T} = -P(T) K T N(-d2)
\]

\[
\frac{\partial C}{\partial t} = q e^{-qT} S N[d1] - r P(T) K N[d2] - \frac{e^{-qT} S \sigma_N N'[d1]}{2 \sqrt{T}}
\]

\[
\frac{\partial P}{\partial t} = -q e^{-qT} S N[-d1] + r P(T) K N[-d2] - \frac{e^{-qT} S \sigma_N N'[d1]}{2 \sqrt{T}}
\]

\[
\frac{\partial C}{\partial T} = -q e^{-qT} S N[d1] + r P(T) K N[d2] + \frac{e^{-qT} S \sigma_N N'[d1]}{2 \sqrt{T}}
\]

\[
\frac{\partial P}{\partial T} = q e^{-qT} S N[-d1] - r P(T) K N[-d2] + \frac{e^{-qT} S \sigma_N N'[d1]}{2 \sqrt{T}}
\]

\[
\frac{\partial C}{\partial \sigma} = e^{-qT} N'[d1] \frac{\sigma_N \sqrt{T} - d1}{\sigma_N}, \quad \frac{\partial P}{\partial \sigma} = e^{-qT} N'[-d1] \frac{d1 - \sigma_N \sqrt{T}}{\sigma_N}
\]

\[
\frac{\partial C}{\partial t} = q e^{-qT} N'[d1] d1 \frac{\sigma_N}{4 \sqrt{T}}, \quad \frac{\partial P}{\partial t} = -q e^{-qT} N'[-d1] d1 \frac{\sigma_N}{4 \sqrt{T}}
\]

\[
\frac{\partial C}{\partial K} = -P(T) N[d2], \quad \frac{\partial P}{\partial K} = P(T) N[-d2]
\]

\[
\frac{\partial^2 C}{\partial^2 K} = \frac{P(T) N'[d2]}{K \sigma_N \sqrt{T}}, \quad \frac{\partial^2 P}{\partial^2 K} = \frac{P(T) N'[-d2]}{K \sigma_N \sqrt{T}}
\]
Heston & Lewis Model: A Revision

The PDE that an option must satisfy by assuming stochastic volatility is given by the following:

\[
\frac{\partial F}{\partial t} + r \frac{\partial F}{\partial S} S + k(\alpha - \gamma \sigma(t)^2) \frac{\partial F}{\partial \sigma} + \frac{1}{2} \sigma^2 \frac{\partial^2 F}{\partial S^2} S^2 + \rho \sigma^2 \Delta \frac{\partial F}{\partial \sigma} S + \frac{1}{2} \sigma^2 \Delta^2 \sigma^2 - r F(S) = 0
\]

Where:

\[
\frac{dS(t)}{S(t)} = r dt + \sigma(t)^2 dW_S
\]

\[
d\sigma(t)^2 = k(\alpha - \gamma \sigma(t)^2) dt + \delta \sigma(t)^2 dW_{\sigma}
\]

\[
dW_S dW_{\sigma} = \rho dt
\]

By analogy to Black, Scholes (1973) we have:

\[
C(S, T, K) = S(t)P_1 - P(T)KP_2
\]

Heston expressed the PDE in terms of \( x = \ln S \), that \( P_1 \) and \( P_2 \) must satisfy:

\[
\frac{\partial P_j}{\partial t} + (r + u_j \sigma^2) \frac{\partial P_j}{\partial x} + (a_j - b_j \sigma^2) \frac{\partial P_j}{\partial \sigma} + \frac{1}{2} \sigma^2 \frac{\partial^2 P_j}{\partial x^2} \sigma^2 + \rho \sigma^2 \delta \frac{\partial P_j}{\partial x} \sigma = \frac{1}{2} \sigma^2 \frac{\partial^2 P_j}{\partial \sigma^2} \delta^2 \sigma^2
\]

\[
u_1 = \frac{1}{2} \quad \nu_2 = -\frac{1}{2} \quad a_j = k\alpha \quad b_1 = k + \gamma - \rho \delta \quad b_2 = k + \gamma
\]

The inverse transform is given by:

\[
P(x, v, t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-ikx} P(k, v, t) \, dk
\]

By subsisting we obtain:

\[
-\frac{\partial P_j}{\partial t} - k^2 \sigma^2 P_j - \left( \frac{1}{2} - j \right) i k \sigma^2 P_j + \frac{1}{2} \sigma^2 \frac{\partial^2 P_j}{\partial \sigma^2} + \delta \sigma \frac{\partial P_j}{\partial \sigma} + (a_j - b_j \sigma^2) \frac{\partial P_j}{\partial \sigma}
\]

Now define:

\[
\alpha = -\frac{k^2}{2} - \frac{ik}{2} + ijk
\]

\[
\beta = \lambda - \delta pf - \delta pik
\]

\[
\gamma = \frac{\delta^2}{2}
\]
Then the partial differential equation becomes:

\[ \sigma^2 \left[ \alpha P_j - \beta \frac{\partial P_j}{\partial \sigma} + \gamma \frac{\partial^2 P_j}{\partial^2 \sigma} \right] + \alpha_j \frac{\partial P_j}{\partial \sigma} - \frac{\partial P_j}{\partial t} \]

Now if we substitute:

\[ P_j(k, v, t) = \exp[C(k, t) + D(k, t)v] P_j(k, v, 0) \]

\[ = \frac{1}{ik} \exp[C(k, t) + D(k, t)v] \]

It follows that:

\[ \frac{\partial P_j}{\partial t} = \left[ \frac{\partial C}{\partial t} + \frac{\partial D}{\partial t} v \right] P_j \]

\[ \frac{\partial P_j}{\partial \sigma} = D P_j \]

\[ \frac{\partial^2 P_j}{\partial^2 \sigma} = D^2 P_j \]

The PDE is satisfied when:

\[ \frac{\partial C}{\partial t} = \lambda D \]

\[ \frac{\partial D}{\partial t} = \alpha - \beta D + \gamma D^2 = \gamma (D - r_+)(D - r_-) \]

\[ r_\pm = \frac{\beta \pm \sqrt{-\beta^2 - 4 \alpha \gamma}}{2 \gamma} \sim \frac{\beta \pm \sqrt{\beta^2 - 4 \alpha \gamma}}{2 \gamma} \sim \frac{\beta \pm d}{\delta^2} \]

\[ D = -\frac{r_\pm e^{r_\pm T} + Kr_\pm e^{r_\pm T}}{e^{r_\pm T} + Ke^{r_\pm T}} \]

Integrating with the final condition \( C(k, 0) = 0, \; D(k, 0) = 0 \), we obtain the following:

\[ D(k, t) = r_- \left( \frac{1 - \exp(-td)}{1 - h \exp(-td)} \right) \]

\[ C(k, t) = \lambda \left[ r_- t - \ln\left( \frac{1 - h \exp(-td)}{1 - h} \right) \right] \]

\[ h \sim \frac{r_-}{r_+} \]
The Heston integral may be valued by using fundamental transform, the value of options become:

\[ C(S, T, K) = S(t)e^{-qT} - KP(T) \frac{e^{-ikX}}{2\pi} \int_{l}^{i} \frac{e^{-ikX}}{k^2 - ik} dk \]

Where:

\[ X = \ln \left( \frac{S}{K} \right) + (r - q) T \]

\[ H(k, V, T) = \exp(f_1(t) + f_2(t)V_t) \]

\[ f_1(t) = \omega \left[ tg - \ln \left( \frac{1 - h \exp(-td)}{1 - h} \right) \right] \]

\[ f_2(t) = g \left( \frac{1 - \exp(-td)}{1 - h \exp(-td)} \right) \]

\[ d = \sqrt{\theta + 4c} , \quad g = \frac{1}{2} (\theta - d) , \quad h = \frac{\theta - d}{\theta + d} \]

\[ \theta = \frac{2}{\delta^2} \left[ (1 - \gamma - ik)\rho\delta + \sqrt{k^2 - \gamma (1 - \gamma)\delta^2} \right] \text{ for } \gamma < 1 \]

Where:

\[ t = \frac{\delta^2 T}{2} , \omega = \frac{2k\theta}{\delta^2} , c = \frac{2c(k)}{\delta^2} , c(k) = \frac{k^2 - ik}{2} \]

We have assumed the following as Lewis (2000):

\[ \frac{\partial F}{\partial T} = \frac{1}{2} \delta^2 T \Rightarrow 2 \frac{\partial F}{\delta^2 \partial t} \]

We may see some examples:

1.1

| Spot Price (S) | 1 |
| Strike Price (K) | 1 |
| Risk Free Rate (r) | 0.03 |
| Dividend Yield (δ) | 0 |
| Time to Maturity (τ = T - t) | 2 |
| Rho (ρ) | -0.5 |
| Kappa (κ) | 0.2 |
| Theta (θ) | 0.03 |
| Volatility of Variance (σ) | 0.1 |
| Current variance (v) | 0.01 |
| Imaginary Part of k (ki) | 0.5 |
| Gamma (γ) | -5 |
| Heston Call Price | 0.0976 |
| Black Scholes Call Price | 0.0887 |
| Heston Put Price | 0.0394 |
| Black Scholes Put Price | 0.0305 |

1.2

| Spot Price (S) | 1 |
| Strike Price (K) | 1 |
| Risk Free Rate (r) | 0.03 |
| Dividend Yield (δ) | 0 |
| Time to Maturity (τ = T – t) | 2 |
| Rho (ρ) | -0.8 |
| Kappa (κ) | 0.2 |
| Theta (θ) | 0.03 |
| Volatility of Variance (σ) | 0.1 |
| Current variance (v) | 0.01 |
| Imaginary Part of k (ki) | 0.5 |
| Gamma (γ) | -5 |
| Heston Call Price | 0.1001 |
| Black Scholes Call Price | 0.0887 |
| Heston Put Price | 0.0419 |
| Black Scholes Put Price | 0.0305 |

1.3

| Spot Price (S) | 1 |
| Strike Price (K) | 1 |
| Risk Free Rate (r) | 0.03 |
| Dividend Yield (δ) | 0 |
| Time to Maturity (τ = T – t) | 5 |
| Rho (ρ) | -0.8 |
| Kappa (κ) | 0.2 |
| Theta (θ) | 0.03 |
| Volatility of Variance (σ) | 0.1 |
| Current variance (v) | 0.01 |
| Imaginary Part of k (ki) | 0.5 |
| Gamma (γ) | -5 |
| Heston Call Price | 0.2022 |
| Black Scholes Call Price | 0.1703 |
| Heston Put Price | 0.0629 |
| Black Scholes Put Price | 0.0310 |
The imaginary solution is given by:

$$D = - \frac{Kn e^{rT} + r e^{rT}}{K e^{rT} + e^{rT}}$$

Integrating with the final condition $C(k, 0) = 0, D(k, 0) = 0$, we obtain the following:

$$D(k, t) = r_t \left( \frac{1 - \exp(-td)}{1 - h \exp(-td)} \right)$$

$$C(k, t) = \lambda \left[ r_t t - \ln \left( \frac{1 - h \exp(-td)}{1 - h} \right) \right]$$

$$h \sim \frac{r_+}{r_-}$$

The imaginary solution gives negative values for option prices. The real solution is positive in real Heston solution as well. Heston assumes the following characteristic equation:

$$f(x, \phi, \sigma^2) = e^{C(T,\phi) + D(T,\phi) + i\phi x}$$

By solving we obtain:

$$\frac{\partial D_j}{\partial T} = \rho \delta \phi i D_j - \frac{1}{2} \phi^2 + \frac{1}{2} \delta^2 D_j^2 + u_j i \phi - b_j D_j$$

$$\frac{\partial C_j}{\partial T} = r i \phi + a D_j$$
The Heston Ricatti equation is:

\[
\frac{\partial D_j}{\partial T} = P_j - Q_jD_j + R D_j^2 = R (D - r_+) (D - r_-)
\]

\[
r_\pm = \frac{Q_j \pm \sqrt{-Q_j^2 - 4 P_j R}}{2 R} \approx \frac{Q_j \pm \delta}{\delta^2}
\]

\[
D = -\frac{r_\pm e^{r_\pm^T} + K r_\pm e^{r_\pm^T}}{e^{r_\pm^T} + Ke^{r_\pm^T}}
\]

The real solution is given by the following by using the condition \(D(0, \varnothing) = 0\):

\[
D_j = \frac{b_j - \rho \delta \varnothing i - d_j}{\delta^2} \frac{1 - e^{-d_j T}}{1 - g_j e^{-d_j T}}
\]

\[
d_j = \sqrt{(\rho \delta \varnothing i - b_j)^2 - \delta^2 (2 u_j \varnothing - \varnothing^2)}
\]

\[
g_j = \frac{b_j - \rho \delta \varnothing i - d_j}{b_j - \rho \delta \varnothing i + d_j}
\]

Using the condition \(C(0, \varnothing) = 0\) we obtain the solution for \(C_j\):

\[
C_j = \int_0^T ri\varnothing \, dy + a \frac{Q_j - d_j}{\delta^2} \int_0^T \frac{1 - e^{-d_j y}}{1 - g_j e^{-d_j y}} \, dy + K_1
\]

By solving we obtain the following result for \(C_j\):

\[
C_j = ri\varnothing T + a \frac{b_j - \rho \delta \varnothing i - d_j}{\delta^2} [ (b_j - \rho \delta \varnothing i - d_j) T - \ln(\frac{1 - g_j e^{-d_j T}}{1 - g_j})]
\]

Heston inverted the characteristic function to obtain the desired probability:

\[
P_j(x, \varnothing, \sigma^2, \ln K) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \text{Re} \left[ \frac{e^{-i\varnothing \ln K f(x, \varnothing, \sigma^2, T)}}{i\varnothing} \right] d\varnothing
\]

We may compare the result for European Call options:

1.1

| Spot Price (S) | 1 |
| Strike Price (K) | 1 |
| Risk Free Rate (r) | 0.03 |
| Time to Maturity (T - t) | 2 |
| Rho (\(\rho\)) | -0.5 |
| Kappa (\(\kappa\)) | 0.2 |
| Theta (\(\theta\)) | 0.06 |
| Lambda (\(\lambda\)) | 1 |
We may compare the result for European Put options by setting a positive correlation coefficient:

2.1
We may note that for rational value of parameters Heston (1993) converges to Black, Scholes (1973). Indeed, it is possible to obtain the Heston prices as well by simulating, in combination, the dynamic of the stock prices and the dynamic of the variance. The distributions are given by the following:

\[
\frac{dS(t)}{S(t)} = r \, dt + \sqrt{\sigma(t)^2} \, dW_s
\]

\[
d\sigma(t)^2 = (k\alpha - (k + \gamma)\sigma(t)^2) \, dt + \delta \sqrt{\sigma(t)^2} \, dW_\sigma
\]
We may compare the result for European Call options:

1.1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>1</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Rho (ρ)</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa (κ)</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta (θ)</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda (λ)</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance (σ)</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
<tr>
<td><strong>Heston Call Price</strong></td>
<td>0.0944</td>
</tr>
<tr>
<td><strong>Black Scholes Call Price</strong></td>
<td>0.0887</td>
</tr>
</tbody>
</table>

1.2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>1.5</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Rho (ρ)</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa (κ)</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta (θ)</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda (λ)</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance (σ)</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
<tr>
<td><strong>Heston Call Price</strong></td>
<td>0.5599</td>
</tr>
<tr>
<td><strong>Black Scholes Call Price</strong></td>
<td>0.5583</td>
</tr>
</tbody>
</table>

1.3

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Price (S)</td>
<td>2</td>
</tr>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Rho (ρ)</td>
<td>-0.5</td>
</tr>
<tr>
<td>Kappa (κ)</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta (θ)</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda (λ)</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance (σ)</td>
<td>0.1</td>
</tr>
</tbody>
</table>

\[ dW_S dW_\sigma = \rho \, dt \]
Current variance \( v \) | 0.01
---|---
Number of Simulations | 1,000

| Heston Call Price | 1.0589
| Black Scholes Call Price | 1.0582

We may compare the result for European Put options by setting a positive correlation coefficient:

2.1

| Spot Price \( S \) | 1 |
| Strike Price \( K \) | 1 |
| Risk Free Rate \( r \) | 0.03 |
| Time to Maturity (Days) | 730 |
| Rho \( \rho \) | 0.5 |
| Kappa \( \kappa \) | 0.2 |
| Theta \( \theta \) | 0.03 |
| Lambda \( \lambda \) | 2 |
| Volatility of Variance \( \sigma \) | 0.1 |
| Current variance \( v \) | 0.01 |
| Number of Simulations | 1,000 |

| Heston Put Price | 0.0303 |
| Black Scholes Put Price | 0.0305 |

2.2

| Spot Price \( S \) | 0.75 |
| Strike Price \( K \) | 1 |
| Risk Free Rate \( r \) | 0.03 |
| Time to Maturity (Days) | 730 |
| Rho \( \rho \) | 0.5 |
| Kappa \( \kappa \) | 0.2 |
| Theta \( \theta \) | 0.03 |
| Lambda \( \lambda \) | 2 |
| Volatility of Variance \( \sigma \) | 0.1 |
| Current variance \( v \) | 0.01 |
| Number of Simulations | 1,000 |

| Heston Put Price | 0.1945 |
| Black Scholes Put Price | 0.1945 |

2.3

<p>| Spot Price ( S ) | 0.5 |
| Strike Price ( K ) | 1 |
| Risk Free Rate ( r ) | 0.03 |
| Time to Maturity (Days) | 730 |
| Rho ( \rho ) | 0.5 |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kappa (κ)</td>
<td>0.2</td>
</tr>
<tr>
<td>Theta (θ)</td>
<td>0.03</td>
</tr>
<tr>
<td>Lambda (λ)</td>
<td>2</td>
</tr>
<tr>
<td>Volatility of Variance (σ)</td>
<td>0.1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Price Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heston Put Price</td>
<td>0.4418</td>
</tr>
<tr>
<td>Black Scholes Put Price</td>
<td>0.4418</td>
</tr>
</tbody>
</table>

We may note that for rational value of parameters Heston (1993) converges to Black, Scholes (1973).

**Appendix**

To run the simulation we used the following VBA code

```vba
Option Base 1
Private Const fSteps = 400
Private Const fLength = 80
' ---------------------------------- COMPLEX LIBRARY ----------------------------------
Public Type cNum
    rP As Double
    iP As Double
End Type
Function thePI()
    thePI = Application.Pi()
End Function
Function set_cNum(rPart, iPart) As cNum
    set_cNum.rP = rPart
    set_cNum.iP = iPart
End Function
Function cNumProd(cNum1 As cNum, cNum2 As cNum) As cNum
    cNumProd.rP = (cNum1.rP * cNum2.rP) - (cNum1.iP * cNum2.iP)
    cNumProd.iP = (cNum1.rP * cNum2.iP) + (cNum1.iP * cNum2.rP)
End Function
```
cNumProd.iP = (cNum1.rP * cNum2.iP) + (cNum1.iP * cNum2.rP)
End Function

Function cNumConj(cNum1 As cNum) As cNum
    cNumConj.rP = cNum1.rP
    cNumConj.iP = -cNum1.iP
End Function

Function cNumDiv(cNum1 As cNum, cNum2 As cNum) As cNum
    Dim conj_cNum2 As cNum
    conj_cNum2 = cNumConj(cNum2)
    cNumDiv.rP = (cNum1.rP * conj_cNum2.rP - cNum1.iP * conj_cNum2.iP) / (cNum2.rP ^ 2 + cNum2.iP ^ 2)
    cNumDiv.iP = (cNum1.rP * conj_cNum2.iP + cNum1.iP * conj_cNum2.rP) / (cNum2.rP ^ 2 + cNum2.iP ^ 2)
End Function

Function cNumAdd(cNum1 As cNum, cNum2 As cNum) As cNum
    cNumAdd.rP = cNum1.rP + cNum2.rP
    cNumAdd.iP = cNum1.iP + cNum2.iP
End Function

Function cNumSub(cNum1 As cNum, cNum2 As cNum) As cNum
    cNumSub.rP = cNum1.rP - cNum2.rP
    cNumSub.iP = cNum1.iP - cNum2.iP
End Function

Function cNumSqrt(cNum1 As cNum) As cNum
    r = Sqr(cNum1.rP ^ 2 + cNum1.iP ^ 2)
    y = Atn(cNum1.iP / cNum1.rP)
    cNumSqrt.rP = Sqr(r) * Cos(y / 2)
    cNumSqrt.iP = Sqr(r) * Sin(y / 2)
Function cNumPower(cNum1 As cNum, n As Double) As cNum
    r = Sqr(cNum1.rP ^ 2 + cNum1.iP ^ 2)
    y = Atn(cNum1.iP / cNum1.rP)
    cNumPower.rP = r ^ n * Cos(y * n)
    cNumPower.iP = r ^ n * Sin(y * n)
End Function

Function cNumExp(cNum1 As cNum) As cNum
    cNumExp.rP = Exp(cNum1.rP) * Cos(cNum1.iP)
    cNumExp.iP = Exp(cNum1.rP) * Sin(cNum1.iP)
End Function

Function cNumSq(cNum1 As cNum) As cNum
    cNumSq = cNumProd(cNum1, cNum1)
End Function

Function cNumReal(cNum1 As cNum) As Double
    cNumReal = cNum1.rP
End Function

Function cNumImag(cNum1 As cNum) As Double
    cNumImag = cNum1.iP
End Function

Function cNumLn(cNum1 As cNum) As cNum
    r = (cNum1.rP ^ 2 + cNum1.iP ^ 2) ^ 0.5
    theta = Atn(cNum1.iP / cNum1.rP)
    cNumLn.rP = Application.Ln(r)
    cNumLn.iP = theta
End Function
Function cNumPowercNum(cNum1 As cNum, cNum2 As cNum) As cNum
    r = Sqr(cNum1.rP ^ 2 + cNum1.iP ^ 2)
    y = Atn(cNum1.iP / cNum1.rP)
    cNumPowercNum.rP = r ^ cNum2.rP * Exp(-cNum2.iP * y) * Cos(cNum2.rP * y + cNum2.iP * Log(r))
    cNumPowercNum.iP = r ^ cNum2.rP * Exp(-cNum2.iP * y) * Sin(cNum2.rP * y + cNum2.iP * Log(r))
End Function

' Lewis (2000) Integrand
Function intH(K As cNum, X, V0, tau, thet, kappa, SigmaV, rho, gam) As Double
    Dim b As cNum, im As cNum, thetaadj As cNum, c As cNum
    Dim d1 As cNum, d As cNum, f As cNum, h As cNum, AA As cNum, BB As cNum
    Dim Hval As cNum, t As cNum, a As cNum, re As cNum

    omega = kappa * thet
    ksi = SigmaV
    theta = kappa
    t = set_cNum(ksi ^ 2 * tau / 2, 0)
    a = set_cNum(2 * omega / ksi ^ 2, 0)
    If (gam = 1) Then
        thetaadj = set_cNum(theta, 0)
    Else
        thetaadj = set_cNum((1 - gam) * rho * ksi + Sqr(theta ^ 2 - gam * (1 - gam) * ksi ^ 2), 0)
    End If
    im = set_cNum(0, 1)
    re = set_cNum(1, 0)
b = cNumProd(set_cNum(2, 0), cNumDiv(cNumAdd(thetaadj, cNumProd(im, cNumProd(K, set_cNum(rho * ksi, 0)))), set_cNum(ksi ^ 2, 0))))
c = cNumDiv(cNumSub(cNumSq(K), cNumProd(im, K)), set_cNum(ksi ^ 2, 0))
d1 = cNumSqrt(cNumAdd(cNumSq(b), cNumProd(set_cNum(4, 0), c)))
d = set_cNum(-cNumReal(d1), -cNumImag(d1))
f = cNumDiv(cNumAdd(b, d), set_cNum(2, 0))
h = cNumDiv(cNumAdd(b, d), cNumSub(b, d))
AA = cNumSub(cNumProd(cNumProd(f, a), t), cNumProd(a, cNumLn(cNumDiv(cNumSub(re, cNumProd(h, cNumExp(cNumProd(d, t)))), cNumSub(re, h))))))
BB = cNumDiv(cNumProd(f, cNumSub(re, cNumExp(cNumProd(d, t)))), cNumSub(re, cNumProd(h, cNumExp(cNumProd(d, t))))))
Hval = cNumExp(cNumAdd(AA, cNumProd(BB, set_cNum(V0, 0))))
intH = cNumReal(cNumProd(cNumDiv(cNumExp(cNumProd(cNumProd(set_cNum(-X, 0), im), K)), cNumSub(cNumSq(K), cNumProd(im, K))), Hval))
End Function
' Heston Price by Fundamental Transform
Function HCTrans(S, K, r, delta, V0, tau, ki, thet, kappa, SigmaV, rho, gam, PutCall As String)
    Dim int_x() As Double, int_y() As Double
    Dim pass_phi As cNum
    ' Lewis Parameters
    omega = kappa * thet
    ksi = SigmaV
    theta = kappa
    kmax = Round(Application.Max(1000, 10 / Sqr(V0 * tau)), 0)
    ReDim int_x(kmax * 5) As Double, int_y(kmax * 5) As Double
    X = Application.Ln(S / K) + (r - delta) * tau
    cnt = 0
    For phi = 0.000001 To kmax Step 0.2
\begin{verbatim}
cnt = cnt + 1
    int_x(cnt) = phi
    pass_phi = set_cNum(phi, ki)
    int_y(cnt) = intH(pass_phi, X, V0, tau, thet, kappa, SigmaV, rho, gam)

Next phi

CallPrice = (S * Exp(-delta * tau) - (1 / thePI) * K * Exp(-r * tau) * TRAPnumint(int_x, int_y))

If PutCall = "Call" Then
    HCTrans = CallPrice
ElseIf PutCall = "Put" Then
    HCTrans = CallPrice + K * Exp(-r * tau) - S * Exp(-delta * tau)
End If

End Function

' Trapezoidal Rule

Function TRAPnumint(X, y) As Double

Dim n As Integer, t As Integer

    n = Application.Count(X)
    TRAPnumint = 0

For t = 2 To n
    TRAPnumint = TRAPnumint + 0.5 * (X(t) - X(t - 1)) * (y(t - 1) + y(t))

Next

End Function
\end{verbatim}
Public Type cNum
  rP As Double
  iP As Double
End Type

Function set_cNum(rPart, iPart) As cNum
  set_cNum.rP = rPart
  set_cNum.iP = iPart
End Function

Function cNumProd(cNum1 As cNum, cNum2 As cNum) As cNum
  cNumProd.rP = (cNum1.rP * cNum2.rP) - (cNum1.iP * cNum2.iP)
  cNumProd.iP = (cNum1.rP * cNum2.iP) + (cNum1.iP * cNum2.rP)
End Function

Function cNumConj(cNum1 As cNum) As cNum
  cNumConj.rP = cNum1.rP
  cNumConj.iP = -cNum1.iP
End Function

Function cNumDiv(cNum1 As cNum, cNum2 As cNum) As cNum
  Dim conj_cNum2 As cNum
  conj_cNum2 = cNumConj(cNum2)
  cNumDiv.rP = (cNum1.rP * conj_cNum2.rP - cNum1.iP * conj_cNum2.iP) / (cNum2.rP ^ 2 + cNum2.iP ^ 2)
  cNumDiv.iP = (cNum1.rP * conj_cNum2.iP + cNum1.iP * conj_cNum2.rP) / (cNum2.rP ^ 2 + cNum2.iP ^ 2)
End Function
Function cNumAdd(cNum1 As cNum, cNum2 As cNum) As cNum
    cNumAdd.rP = cNum1.rP + cNum2.rP
    cNumAdd.iP = cNum1.iP + cNum2.iP
End Function

Function cNumSub(cNum1 As cNum, cNum2 As cNum) As cNum
    cNumSub.rP = cNum1.rP - cNum2.rP
    cNumSub.iP = cNum1.iP - cNum2.iP
End Function

Function cNumSqrt(cNum1 As cNum) As cNum
    r = Sqr(cNum1.rP ^ 2 + cNum1.iP ^ 2)
    y = Atn(cNum1.iP / cNum1.rP)
    cNumSqrt.rP = Sqr(r) * Cos(y / 2)
    cNumSqrt.iP = Sqr(r) * Sin(y / 2)
End Function

Function cNumPower(cNum1 As cNum, n As Double) As cNum
    r = Sqr(cNum1.rP ^ 2 + cNum1.iP ^ 2)
    y = Atn(cNum1.iP / cNum1.rP)
    cNumPower.rP = r ^ n * Cos(y * n)
    cNumPower.iP = r ^ n * Sin(y * n)
End Function

Function cNumExp(cNum1 As cNum) As cNum
    cNumExp.rP = Exp(cNum1.rP) * Cos(cNum1.iP)
    cNumExp.iP = Exp(cNum1.rP) * Sin(cNum1.iP)
End Function

Function cNumSq(cNum1 As cNum) As cNum
cNumSq = cNumProd(cNum1, cNum1)
End Function

Function cNumReal(cNum1 As cNum) As Double
    cNumReal = cNum1.rP
End Function

Function cNumImag(cNum1 As cNum) As Double
    cNumImag = cNum1.iP
End Function

Function cNumLn(cNum1 As cNum) As cNum
    r = (cNum1.rP ^ 2 + cNum1.iP ^ 2) ^ 0.5
    theta = Atn(cNum1.iP / cNum1.rP)
    cNumLn.rP = Application.Ln(r)
    cNumLn.iP = theta
End Function

Function cNumPowercNum(cNum1 As cNum, cNum2 As cNum) As cNum
    r = Sqr(cNum1.rP ^ 2 + cNum1.iP ^ 2)
    y = Atn(cNum1.iP / cNum1.rP)
    cNumPowercNum.rP = r ^ cNum2.rP * Exp(-cNum2.iP * y) * Cos(cNum2.rP * y + cNum2.iP * Log(r))
    cNumPowercNum.iP = r ^ cNum2.rP * Exp(-cNum2.iP * y) * Sin(cNum2.rP * y + cNum2.iP * Log(r))
End Function

Function thePI()
    thePI = Application.Pi
End Function
' Heston Price in the Real part

' Trapezoidal Rule

Function TRAPnumint(X, y) As Double
    n = Application.Count(X)
    TRAPnumint = 0
    For T = 2 To n
        TRAPnumint = TRAPnumint + 0.5 * (X(T) - X(T - 1)) * (y(T - 1) + y(T))
    Next
End Function

' Heston Option Price

Function Heston(PutCall As String, kappa, theta, lambda, rho, sigma, tau, k, S, r, v)
    Dim P1_int(1001) As Double, P2_int(1001) As Double, phi_int(1001) As Double
    Dim p1 As Double, p2 As Double, phi As Double
    cnt = 1
    For phi = 0.0001 To 100.0001 Step 0.1
        phi_int(cnt) = phi
        P1_int(cnt) = HestonP1(phi, kappa, theta, lambda, rho, sigma, tau, k, S, r, v)
        P2_int(cnt) = HestonP2(phi, kappa, theta, lambda, rho, sigma, tau, k, S, r, v)
        cnt = cnt + 1
    Next phi
    p1 = 0.5 + (1 / thePl) * TRAPnumint(phi_int, P1_int)
    p2 = 0.5 + (1 / thePl) * TRAPnumint(phi_int, P2_int)
    If p1 < 0 Then p1 = 0
    If p1 > 1 Then p1 = 1
End Function
If p2 < 0 Then p2 = 0
If p2 > 1 Then p2 = 1

HestonC = S * p1 - k * Exp(-r * tau) * p2

If PutCall = "Call" Then
    Heston = HestonC
ElseIf PutCall = "Put" Then
    Heston = HestonC + k * Exp(-r * tau) - S
End If
End Function

'Risk-Neutral Probability P1

Function HestonP1(phi, kappa, theta, lambda, rho, sigma, tau, k, S, r, v)
Dim tmp_f1 As cNum, f1 As cNum, d As cNum, d1 As cNum, g1 As cNum, DD1 As cNum, cc1 As cNum, b1 As cNum
Dim DD1_1 As cNum, DD1_2 As cNum, DD1_3 As cNum, CC1_1 As cNum, CC1_2 As cNum, CC1_3 As cNum, CC1_4 As cNum
Dim t1 As cNum, t2 As cNum, t3 As cNum
Dim tt1 As Variant, mu1 As Double

mu1 = 0.5
b1 = set_cNum(kappa + lambda - rho * sigma, 0)

d = cNumSqrt(cNumSub(cNumSq(cNumSub(set_cNum(0, rho * sigma * phi), b1)), cNumSub(set_cNum(0, sigma ^ 2 * 2 * mu1 * phi), set_cNum(sigma ^ 2 * phi ^ 2, 0))))
d1 = set_cNum(-cNumReal(d), -cNumImag(d))
g1 = cNumDiv(cNumAdd(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1), cNumSub(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1))

DD1_1 = cNumDiv(cNumAdd(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1), cNumSub(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1))

DD1_2 = cNumSub(set_cNum(1, 0), cNumExp(cNumProd(d1, set_cNum(tau, 0))))

54
DD1_3 = cNumSub(set_cNum(1, 0), cNumProd(g1, cNumExp(cNumProd(d1, set_cNum(tau, 0)))))

DD1 = cNumProd(DD1_1, cNumDiv(DD1_2, DD1_3))

CC1_1 = set_cNum(0, r * phi * tau)

CC1_2 = set_cNum((kappa * theta) / (sigma ^ 2), 0)

CC1_3 = cNumProd(cNumAdd(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1), set_cNum(tau, 0))

CC1_4 = cNumProd(set_cNum(1, 0), cNumLn(cNumDiv(cNumSub(set_cNum(1, 0), cNumProd(g1, cNumProd(d1, set_cNum(tau, 0))))), cNumSub(set_cNum(1, 0), g1))))

cc1 = cNumAdd(CC1_1, cNumProd(CC1_2, cNumSub(CC1_3, CC1_4)))

f1 = cNumExp(cNumAdd(cc1, cNumProd(DD1, set_cNum(v, 0))), set_cNum(0, phi * Application.Ln(S)))

HestonP1 = cNumReal(cNumDiv(cNumProd(cNumExp(set_cNum(0, -phi * Application.Ln(k))), f1), set_cNum(0, phi)))

'HestonP1 = cNumImag(cNumDiv(cNumProd(cNumExp(set_cNum(0, -phi * Application.Ln(k))), f1), set_cNum(0, phi))) 'for Imaginary solution

End Function

' Risk-Neutral Probability P2

Function HestonP2(phi, kappa, theta, lambda, rho, sigma, tau, k, S, r, v)

Dim tmp_f1 As cNum, f1 As cNum, d As cNum, d1 As cNum, g1 As cNum, DD1 As cNum, cc1 As cNum, b1 As cNum

Dim DD1_1 As cNum, DD1_2 As cNum, DD1_3 As cNum, CC1_1 As cNum, CC1_2 As cNum, CC1_3 As cNum, CC1_4 As cNum

Dim mu1 As Double

mu1 = -0.5

b1 = set_cNum(kappa + lambda, 0)

d = cNumSqrt(cNumSub(cNumSq(cNumSub(set_cNum(0, rho * sigma * phi), b1)), cNumSub(set_cNum(0, sigma ^ 2 * 2 + mu1 * phi), set_cNum(sigma ^ 2 * phi ^ 2, 0))))

d1 = set_cNum(-cNumReal(d), -cNumImag(d))
g1 = cNumDiv(cNumAdd(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1),
cNumSub(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1))

DD1_1 = cNumDiv(cNumAdd(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1), set_cNum(sigma ^ 2, 0))

DD1_2 = cNumSub(set_cNum(1, 0), cNumExp(cNumProd(d1, set_cNum(tau, 0))))

DD1_3 = cNumSub(set_cNum(1, 0), cNumProd(g1, cNumExp(cNumProd(d1, set_cNum(tau, 0)))))

DD1 = cNumProd(DD1_1, cNumDiv(DD1_2, DD1_3))

CC1_1 = set_cNum(0, r * phi * tau)

CC1_2 = set_cNum((kappa * theta) / (sigma ^ 2), 0)

CC1_3 = cNumProd(cNumAdd(cNumSub(b1, set_cNum(0, rho * sigma * phi)), d1), set_cNum(tau, 0))

CC1_4 = cNumProd(set_cNum(1, 0), cNumLn(cNumDiv(cNumSub(set_cNum(1, 0), cNumProd(g1, cNumExp(cNumProd(d1, set_cNum(tau, 0)))))), cNumSub(set_cNum(1, 0), g1)))

c1 = cNumAdd(CC1_1, cNumProd(CC1_2, cNumSub(CC1_3, CC1_4)))

f1 = cNumExp(cNumAdd(cNumAdd(cc1, cNumProd(DD1, set_cNum(v, 0))), set_cNum(0, phi * Application.Ln(S))))

HestonP2 = cNumReal(cNumDiv(cNumProd(cNumExp(set_cNum(0, -phi * Application.Ln(k))), f1), set_cNum(0, phi)))

'HestonP2 = cNumImag(cNumDiv(cNumProd(cNumExp(set_cNum(0, -phi * Application.Ln(k))), f1), set_cNum(0, phi))) ' for Imaginary solution

'---------------------------------------------------------------------------------------------

Heston Put & Call price by Monte Carlo Simulation

Function HestonMCCall(kappa, theta, lambda, rho, sigmav, daynum, startS, r, startv, K, ITER)
Dim allS() As Double

simPath = 0
ReDim allS(daynum) As Double
deltat = (1 / 365)

56
For itcount = 1 To ITER

lnSt = Log(startS)
lnvt = Log(startv)
curv = startv
curS = startS

For daycnt = 1 To daynum
    e = Application.NormSInv(Rnd)
eS = Application.NormSInv(Rnd)
ev = rho * eS + Sqr(1 - rho ^ 2) * e
lnSt = lnSt + (r - 0.5 * curv) * deltat + Sqr(curv) * Sqr(deltat) * eS
curS = Exp(lnSt)

lnvt = lnvt + (kappa * theta - (kappa + lambda) * curv) * deltat + Sqr(curv) * sigmav * Sqr(deltat) * ev
curv = Exp(lnvt)
allS(daycnt) = curS
Next daycnt

simPath = simPath + Exp((-daynum / 365) * r) * Application.Max(allS(daynum) - K, 0)
Next itcount

HestonMCCall = simPath / ITER
End Function

Function HestonMCPut(kappa, theta, lambda, rho, sigmav, daynum, startS, r, startv, K, ITER)
Dim allS() As Double

simPath = 0
ReDim allS(daynum) As Double
deltat = (1 / 365)
For itcount = 1 To ITER
57
\[ \text{lnSt} = \log(\text{startS}) \]
\[ \text{lnvt} = \log(\text{startv}) \]
\[ \text{curv} = \text{startv} \]
\[ \text{curS} = \text{startS} \]

For daycnt = 1 To daynum

\[ e = \text{Application.NormSInv(Rnd)} \]
\[ eS = \text{Application.NormSInv(Rnd)} \]
\[ \text{ev} = \rho * eS + \sqrt{1 - \rho^2} * e \]
\[ \text{lnSt} = \text{lnSt} + (r - 0.5 * \text{curv}) * \text{deltat} + \sqrt{\text{curv}} * \sqrt{\text{deltat}} * eS \]
\[ \text{curS} = \exp(\text{lnSt}) \]

\[ \text{lnvt} = \text{lnvt} + (\kappa \theta - (\kappa + \lambda) \text{curv}) * \text{deltat} + \sqrt{\text{curv}} * \text{sigmav} * \sqrt{\text{deltat}} * \text{ev} \]
\[ \text{curv} = \exp(\text{lnvt}) \]
\[ \text{allS(daycnt)} = \text{curS} \]

Next daycnt

\[ \text{simPath} = \text{simPath} + \exp((-\text{daynum} / 365) * r) * \text{Application.Max(K - allS(daynum), 0)} \]

Next itcount

\[ \text{HestonMCPut} = \text{simPath} / \text{ITER} \]

End Function
Levy Process

The jump diffusion process was first introduced by Merton (1976) by using the Poisson process, formally:

\[
\frac{dS(t)}{S(t)} = \mu(t)dt - \gamma \tau + \sigma dWs + dJ
\]

Where \(\mu(t)\) denotes the drift of the distribution and it is the average in the \(dt\), \(\sigma\) denotes the volatility of the distribution and \(dWs\) denotes a Wiener process, instead, \(dJ\) denotes a Poisson process with average \(\gamma \tau\), this permits to have a martingale, the Poisson distribution is given by the following:

\[
\frac{e^{\gamma \tau} (\gamma \tau)^n}{n!} = \text{Exp} \left( \ln \frac{e^{\gamma \tau} (\gamma \tau)^n}{n!} \right) = \text{Exp} \left( \gamma \tau n \ln \tau - \sum_{i=1}^{n} \ln i \right)
\]

For \(n = 1\) we have the following:

\(\gamma \tau e^{\gamma \tau}\)

As result we may rewrite the jump diffusion process in risk neutral world in the following way:

\[
\frac{dS(t)}{S(t)} = r(t)dt - \frac{1}{2} \sigma^2 dt + \sigma dWs + \gamma \tau (e^{\gamma \tau} - 1)
\]

We may note that:

\[
\ln \frac{S(T)}{S(t)} = r(t)dt - \frac{1}{2} \sigma^2 dt + \sigma dWs + \gamma \tau \int_{0}^{\infty} (e^{\gamma \tau} - 1)
\]

If we assume that the Poisson process follows a normal distribution we have the following:

\[
\ln \frac{S(T)}{S(t)} = r(t)dt - \frac{1}{2} \sigma^2 dt + \sigma dWs + \gamma \tau \int_{0}^{\infty} (e^{\mu t + \frac{1}{2} \sigma^2 \tau} - 1)
\]

From this we may obtain explanation for Ito’s lemma, if we take a function of \(S\) as \(F(S)\) we may write Ito’s lemma in the following way:

\[
dF(S) = \left( \frac{\partial F}{\partial t} + (\mu(t) - \gamma) \frac{\partial F}{\partial S} S + \frac{\gamma}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 + \gamma \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 \right) dt
\]

\[
+ \sqrt{1 + \gamma \sigma} \frac{\partial F}{\partial S} SdZ
\]
Where:

\[ \sqrt{1 + \gamma} \sigma dZ = N \left( \frac{z - \mu}{\sqrt{1 + \gamma} \sigma \sqrt{dt}} ; (1 + \gamma) \sigma^2 dt \right) \]

As result:

\[ E \left[ \sqrt{1 + \gamma} \sigma dZ \right] = 0 \]

Because:

\[ \int f(z) = \int \frac{z - \mu}{\sqrt{1 + \gamma} \sigma \sqrt{dt}} f(z) = 0 \]

From this we may rewrite the Levy process in the following way:

\[ \ln \frac{S(T)}{S(t)} = r(t) dt - \frac{1}{2} \sigma^2 (1 + \gamma) dt + \sigma dW_s + \gamma t \int_0^\infty (e^{\mu + \frac{1}{2} \sigma^2} - 1) \]

The stochastic differential equation that an option must satisfy in risk neutral world is given by the following:

\[ \frac{\partial F}{\partial t} + r(t) \frac{\partial F}{\partial S} S + (1 + \gamma) \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 - r(t) F(S, t) = 0 \]

The solution it is easy to solve, because if we take Ito’s lemma and we take the expectation we obtain that the solution to the parabolic problem is given by the following by using the integrant factor \( e^{-rt} \):

\[ F(S(T)) = e^{-rT} E \left[ F(S) \right] \]

The final pay off of a Call and Put option is given, respectively, by the following:

\[ Call = Max[S(T) - K ; 0] \]
\[ Put = Max[K - S(T) ; 0] \]

We may obtain the price of the options by simulating and discounting the final pays off, we prefer to simulate with the first Levy process because is a martingale without applying Levy Ito’s lemma, anyway, by changing the parameter of the Poisson distribution we achieve the same results. It is interesting to compare the approach with Black, Scholes (1973) to note the differences, as such we have the following prospects for Call options:

1.1

<table>
<thead>
<tr>
<th>Spot Price (S)</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Eta (λ)</td>
<td>1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Levy Call Price</th>
<th>0.0881</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black, Scholes Call Price</td>
<td>0.0887</td>
</tr>
</tbody>
</table>

For rational value of parameters simulations converge to Black, Scholes (1973)

1.2

<table>
<thead>
<tr>
<th>Spot Price (S)</th>
<th>1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Eta (λ)</td>
<td>1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Levy Call Price</th>
<th>0.5630</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black, Scholes Call Price</td>
<td>0.5583</td>
</tr>
</tbody>
</table>

Simulations permits to capture the skew for options deep in the money

1.3

<table>
<thead>
<tr>
<th>Spot Price (S)</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Eta (λ)</td>
<td>1</td>
</tr>
<tr>
<td>Current variance (v)</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Levy Call Price</th>
<th>1.0750</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black, Scholes Call Price</td>
<td>1.0582</td>
</tr>
</tbody>
</table>

Simulations permits to capture the skew for options deep in the money. As results we have the following prospects for Put options:

2.1

<table>
<thead>
<tr>
<th>Spot Price (S)</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Eta (λ)</td>
<td>1</td>
</tr>
</tbody>
</table>
Current variance ($\nu$) | 0.01
---|---
Number of Simulations | 1,000

| Levy Put Price | 0.0302 |
| Black, Scholes Put Price | 0.0305 |

For rational value of parameters the simulations converge to Black, Scholes (1973)

2.2

| Spot Price ($S$) | 0.75 |
| Strike Price ($K$) | 1 |
| Risk Free Rate ($r$) | 0.03 |
| Time to Maturity (Days) | 730 |
| Eta ($\lambda$) | 1 |
| Current variance ($\nu$) | 0.01 |
| Number of Simulations | 1,000 |

| Levy Put Price | 0.1951 |
| Black, Scholes Put Price | 0.1945 |

For rational value of parameters the simulations converge to Black, Scholes (1973)

2.3

| Spot Price ($S$) | 0.5 |
| Strike Price ($K$) | 1 |
| Risk Free Rate ($r$) | 0.03 |
| Time to Maturity (Days) | 730 |
| Eta ($\lambda$) | 1 |
| Current variance ($\nu$) | 0.01 |
| Number of Simulations | 1,000 |

| Levy Put Price | 0.4408 |
| Black, Scholes Put Price | 0.4418 |

For rational value of parameters the simulations converge to Black, Scholes (1973)

Appendix

To run the simulation we used the following VBA code

Function LevyMCCallNorm(eta, daynum, startS, r, startv, K, ITER)

Dim allS() As Double
Function LevyMCCallNorm(eta, daynum, startS, r, startv, K, ITER)
    Dim allS() As Double
    simPath = 0
    ReDim allS(daynum) As Double
    deltat = (1 / 365)
    For itcount = 1 To ITER
        lnSt = Log(startS)
        curS = startS
        For daycnt = 1 To daynum
            eS = Application.NormSInv(Rnd)
            lnSt = lnSt + (r - 0.5 * startv) * deltat + Sqr(startv) * Sqr(deltat) * eS + eta * deltat * Application.Sum((Exp((r + 0.5 * startv) * deltat) - 1))
            curS = Exp(lnSt)
            allS(daycnt) = curS
        Next daycnt
        simPath = simPath + Exp((-daynum / 365) * r) * Application.Max(allS(daynum) - K, 0)
    Next itcount
    LevyMCCallNorm = simPath / ITER
End Function

Function LevyMCPutNorm(eta, daynum, startS, r, startv, K, ITER)
    Dim allS() As Double
    simPath = 0
    ReDim allS(daynum) As Double
    deltat = (1 / 365)
    For itcount = 1 To ITER
        lnSt = Log(startS)
        curS = startS
        For daycnt = 1 To daynum
            eS = Application.NormSInv(Rnd)
            lnSt = lnSt + (r - 0.5 * startv) * deltat + Sqr(startv) * Sqr(deltat) * eS + eta * deltat * Application.Sum((Exp((r + 0.5 * startv) * deltat) - 1))
            curS = Exp(lnSt)
            allS(daycnt) = curS
        Next daycnt
        simPath = simPath + Exp((-daynum / 365) * r) * Application.Max(allS(daynum) - K, 0)
    Next itcount
    LevyMCPutNorm = simPath / ITER
End Function
eS = Application.NormSInv(Rnd)

InSt = lnSt + (r - 0.5 * startv) * deltat + Sqr(startv) * Sqr(deltat) * eS + eta * deltat * Application.Sum((Exp((r + 0.5 * startv) * deltat) - 1))

curS = Exp(lnSt)

allS(daycnt) = curS

Next daycnt

simPath = simPath + Exp((-daynum / 365) * r) * Application.Max(K - allS(daynum), 0)

Next itcount

LevyMCPutNorm = simPath / ITER

End Function
American Options

We introduce arbitrage theory, in practice if we built the following portfolio we have:

\[ V_t = \pm S \frac{\partial F(S)}{\partial S} - F(S) \]

The portfolio is risk free, as such by using Ito’s lemma we obtain the following stochastic differential equation:

\[
\frac{\partial F}{\partial t} + (r - q) \frac{\partial F}{\partial S} S + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} S^2 \sigma^2 - rF(S) = 0
\]

We may solve the stochastic differential equation by using the integrant factor:

\[ F(S) = Z(S)e^{-rT} \]

By solving the stochastic differential equation for \( Z(S) \) we obtain that the solution is given by:

\[ Z(S) = E(S(T)) \]

By solving we obtain:

\[ F(S) = E(S(T)e^{-rT}) \]

The problem for American options is termed as follows:

\[
Call = \sup [S(T) - K ; 0] \\
Put = \sup [K - S(T) ; 0]
\]

The prices of the options are given by the expectation of the final pay off discounted:

\[
C(S, T, K) = E[\max(S(t) - K ; P(T)(S(T) - K))] \\
P(S, T, K) = E[\max(K - S(t); P(T)(K - S(T)))]
\]

As result we may rewrite the American options as follows:

\[
C(S, T, K) = P(T)E[\max(S(t) ; P(T)S(T))] \\
P(S, T, K) = -E[\max(S(t); P(T)S(T))] = E[\min(S(t); P(T)S(T))]
\]

The value of the stock simulated is given by the following:

\[ S(T) = S(t)e^{(r-q)T + \frac{1}{2} \sigma^2 T} \]

As such we have the following:
\[
C(S, T, K) = \mathbb{E} \{ \max \left[ \max \left( S(t)e^{(r-q-\frac{1}{2}\sigma^2)dt+\sigma N(0;1)\sqrt{\Delta t}}; S(t)e^{-qT+\frac{1}{2}\sigma^2T} \right) - K; 0 \right] \} \\
P(S, T, K) = \mathbb{E} \left\{ \min \left[ K - \min \left( S(t)e^{(r-q-\frac{1}{2}\sigma^2)dt+\sigma N(0;1)\sqrt{\Delta t}}; S(t)e^{-qT+\frac{1}{2}\sigma^2T} \right); 0 \right] \right\}
\]

**Numerical Results**

We may compare the model with Black, Scholes (1973) to note the premium of American options without considering the dividend. As such we have the following prospects:

1.1

<table>
<thead>
<tr>
<th>Spot Price (S)</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Current variance ((\nu))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1,000</td>
</tr>
</tbody>
</table>

**American Call Price** 0.0988
**Black, Scholes Call Price** 0.0887

Simulations converge to Black, Scholes (1973)

1.2

<table>
<thead>
<tr>
<th>Spot Price (S)</th>
<th>1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Current variance ((\nu))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1,000</td>
</tr>
</tbody>
</table>

**American Call Price** 0.6445
**Black, Scholes Call Price** 0.5583

Simulations permits to capture the skew for options deep in the money

1.3

<table>
<thead>
<tr>
<th>Spot Price (S)</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price (K)</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate (r)</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity (Days)</td>
<td>730</td>
</tr>
<tr>
<td>Current variance ((\nu))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>American Call Price</strong></td>
<td>1.1918</td>
</tr>
<tr>
<td><strong>Black, Scholes Call Price</strong></td>
<td>1.0582</td>
</tr>
</tbody>
</table>

Simulations permits to capture the skew for options deep in the money

2.1

<table>
<thead>
<tr>
<th>Spot Price ((S))</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price ((K))</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate ((r))</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity ((Days))</td>
<td>730</td>
</tr>
<tr>
<td>Current variance ((v))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
<tr>
<td><strong>American Put Price</strong></td>
<td>0.0309</td>
</tr>
<tr>
<td><strong>Black, Scholes Put Price</strong></td>
<td>0.0305</td>
</tr>
</tbody>
</table>

Simulations converge to Black, Scholes (1973)

2.2

<table>
<thead>
<tr>
<th>Spot Price ((S))</th>
<th>0.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price ((K))</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate ((r))</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity ((Days))</td>
<td>730</td>
</tr>
<tr>
<td>Current variance ((v))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
<tr>
<td><strong>American Put Price</strong></td>
<td>0.2711</td>
</tr>
<tr>
<td><strong>Black, Scholes Put Price</strong></td>
<td>0.1945</td>
</tr>
</tbody>
</table>

Simulations permits to capture the skew for options deep in the money

2.3

<table>
<thead>
<tr>
<th>Spot Price ((S))</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike Price ((K))</td>
<td>1</td>
</tr>
<tr>
<td>Risk Free Rate ((r))</td>
<td>0.03</td>
</tr>
<tr>
<td>Time to Maturity ((Days))</td>
<td>730</td>
</tr>
<tr>
<td>Current variance ((v))</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Simulations</td>
<td>1.000</td>
</tr>
</tbody>
</table>
American Put Price 0.5131
Black, Scholes Put Price 0.4418

Simulations permits to capture the skew for options deep in the money

**Appendix**

To run the simulation we used the following VBA code:

Function AmericanMCCall(daynum, startS, r, startv, K, ITER)

Dim allS() As Double

simPath = 0

ReDim allS(daynum) As Double

deltat = (1 / 365)

For itcount = 1 To ITER

    lnSt = Log(startS)

    curS = startS

    For daycnt = 1 To daynum

        eS = Application.NormSInv(Rnd)

        lnSt = lnSt + (r - 0.5 * startv) * deltat + Sqr(startv) * Sqr(deltat) * eS

        curS = Exp(lnSt)

        allS(daycnt) = Application.Max(curS, startS * Exp(0.5 * startv * (daycnt / 365)))

    Next daycnt

    simPath = simPath + Exp((-daynum / 365) * r) * Application.Max(allS(daynum) - K, 0)

Next itcount

    AmericanMCCall = simPath / ITER

End Function
Function AmericanMCPut(daynum, startS, r, startv, K, ITER)

Dim allS() As Double

simPath = 0

ReDim allS(daynum) As Double
deltat = (1 / 365)

For itcount = 1 To ITER
    lnSt = Log(startS)
curS = startS
    For daycnt = 1 To daynum
        eS = Application.NormSInv(Rnd)
        lnSt = lnSt + (r - 0.5 * startv) * deltat + Sqr(startv) * Sqr(deltat) * eS
        curS = Exp(lnSt)
        allS(daycnt) = Application.Min(curS, startS * Exp(0.5 * startv * (daycnt / 365)))
    Next daycnt
    simPath = simPath + Application.Max(K - allS(daynum), 0)
Next itcount

AmericanMCPut = simPath / ITER

End Function
Binomial Model

Now it is interesting to introduce the lattice methods in binomial model as such we assume the following:

\[
\begin{align*}
    u &= e^{-q\Delta t + \sigma\sqrt{\Delta t}} \\
    d &= e^{-q\Delta t - \sigma\sqrt{\Delta t}} \\
    a &= e^{(r-q)\Delta t}
\end{align*}
\]

The risk neutral probability is given by the following for up and down respectively:

\[
p = \frac{a - d}{u - d} (1 - p)
\]

The pays off are given by:

\[
C_{N,i} = \text{Max} \left[ S_u d^{N-i} - K, 0 \right]
\]

\[
P_{N,i} = \text{Max} \left[ K - S_u d^{N-i}, 0 \right]
\]

The prices are given for European options by:

\[
C_{j,i} = e^{-r\Delta t} \left[ p C_{j+1,i+1} + (1 - p)C_{j+1,i} \right]
\]

\[
P_{j,i} = e^{-r\Delta t} \left[ p P_{j+1,i+1} + (1 - p)P_{j+1,i} \right]
\]

Instead, for American options by:

\[
C_{j,i} = \text{Max} \left[ S_u d^{j-i} - K, e^{-r\Delta t} (p C_{j+1,i+1} + (1 - p)C_{j+1,i}) \right]
\]

\[
P_{j,i} = \text{Max} \left[ K - S_u d^{j-i}, e^{-r\Delta t} (p P_{j+1,i+1} + (1 - p)P_{j+1,i}) \right]
\]

We may compare now the model with the European Call options without dividend:

<table>
<thead>
<tr>
<th>$\sigma_S$</th>
<th>$K$</th>
<th>$r$</th>
<th>$S_0$</th>
<th>$T$</th>
<th>Lattice 150 Nodes</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.25</td>
<td>0.50747</td>
<td>0.50747</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.25</td>
<td>0.25747</td>
<td>0.25747</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.25</td>
<td>0.01794</td>
<td>0.01796</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.5</td>
<td>0.51489</td>
<td>0.51489</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.5</td>
<td>0.26489</td>
<td>0.26489</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.5</td>
<td>0.02790</td>
<td>0.02793</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.75</td>
<td>0.52225</td>
<td>0.52225</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.75</td>
<td>0.27225</td>
<td>0.27225</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.75</td>
<td>0.03662</td>
<td>0.03666</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>
We may compare now the model with the European Put options without dividend:

<table>
<thead>
<tr>
<th>$\sigma_S$</th>
<th>$K$</th>
<th>$r$</th>
<th>$S_0$</th>
<th>$T$</th>
<th>Lattice 150 Nodes</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.25</td>
<td>0.01047</td>
<td>0.01049</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.25</td>
<td>0.24253</td>
<td>0.24253</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.25</td>
<td>0.49253</td>
<td>0.49253</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.5</td>
<td>0.01301</td>
<td>0.01305</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.5</td>
<td>0.23511</td>
<td>0.23511</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.5</td>
<td>0.48511</td>
<td>0.48511</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.75</td>
<td>0.01437</td>
<td>0.01442</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.75</td>
<td>0.22775</td>
<td>0.22775</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.75</td>
<td>0.47775</td>
<td>0.47775</td>
</tr>
</tbody>
</table>

As we may see the two formulations converge, this is due to the fact that binomial distribution converges in the limit to the normal distribution. For American Call options without dividend we obtain the same result of European Call options. It is interesting to compare now the model for American Put options:

<table>
<thead>
<tr>
<th>$\sigma_S$</th>
<th>$K$</th>
<th>$r$</th>
<th>$S_0$</th>
<th>$T$</th>
<th>Lattice 150 Nodes</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.25</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1</td>
<td>0.25</td>
<td>0.01117</td>
<td>0.01049</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.25</td>
<td>0.25000</td>
<td>0.24253</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.25</td>
<td>0.50000</td>
<td>0.49253</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.5</td>
<td>0.01449</td>
<td>0.01305</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.5</td>
<td>0.25000</td>
<td>0.23511</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.5</td>
<td>0.50000</td>
<td>0.48511</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>0.75</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>0.75</td>
<td>0.01664</td>
<td>0.01442</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>0.75</td>
<td>0.25000</td>
<td>0.22775</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>0.75</td>
<td>0.50000</td>
<td>0.47775</td>
</tr>
</tbody>
</table>

We may note that for the American Put options there is parity relations Put Call at the money.
Appendix

Function BinomialCallAmerican(Spot, q, k, T, r, sigma, n)
Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(- q * dt + sigma * (dt ^ 0.5))
    d = Exp(- q * dt - sigma * (dt ^ 0.5))
    p = (Exp((r – q) * dt) - d) / (u - d)
Dim S() As Double
ReDim S(n + 1, n + 1) As Double
    For i = 1 To n + 1
        For j = i To n + 1
            S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
        Next j
    Next i
Dim Opt() As Double
ReDim Opt(n + 1, n + 1) As Double
    For i = 1 To n + 1
        Opt(i, n + 1) = Application.Max(S(i, n + 1) - k, 0)
    Next i
    For j = n To 1 Step -1
        For i = 1 To j
            Opt(i, j) = Application.Max(S(i, j) - k, Exp(- r * dt) * (p * Opt(i, j + 1) + (1 - p) * Opt(i + 1, j + 1)))
        Next i
    Next j
BinomialCallAmerican = Opt(i, j)
Next j
Next i
End Function
Function BinomialCallEuropean(Spot, q, k, T, r, sigma, n)

    Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(- q * dt + sigma * (dt ^ 0.5))
    d = Exp(- q * dt - sigma * (dt ^ 0.5))
    p = (Exp((r – q) * dt) - d) / (u - d)

    Dim S() As Double
    ReDim S(n + 1, n + 1) As Double
    For i = 1 To n + 1
        For j = i To n + 1
            S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
        Next j
    Next i

    Dim Opt() As Double
    ReDim Opt(n + 1, n + 1) As Double
    For i = 1 To n + 1
        Opt(i, n + 1) = Application.Max(S(i, n + 1) - k, 0)
    Next i
    For j = n To 1 Step -1
        For i = 1 To j
            Opt(i, j) = Exp(-r * dt) * (p * Opt(i, j + 1) + (1 - p) * Opt(i + 1, j + 1))
        Next i
    Next j
    BinomialCallEuropean = Opt(i, j)

End Function
Function BinomialPutAmerican(Spot, q, k, T, r, sigma, n)
Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(- q * dt + sigma * (dt ^ 0.5))
    d = Exp(- q * dt - sigma * (dt ^ 0.5))
    p = (Exp((r – q) * dt) - d) / (u - d)
Dim S() As Double
ReDim S(n + 1, n + 1) As Double
    For i = 1 To n + 1
        For j = i To n + 1
            S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
        Next j
    Next i
Dim Opt() As Double
ReDim Opt(n + 1, n + 1) As Double
    For i = 1 To n + 1
        Opt(i, n + 1) = Application.Max(k - S(i, n + 1), 0)
    Next i
    For j = n To 1 Step -1
        For i = 1 To j
            Opt(i, j) = Application.Max(k - S(i, j), Exp(-r * dt) * (p * Opt(i, j + 1) + (1 - p) * Opt(i + 1, j + 1)))
        Next i
    Next j
    BinomialPutAmerican = Opt(i, j)
Next i
Next j
End Function
Function BinomialPutEuropean(Spot, q, k, T, r, sigma, n)

    Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(- q * dt + sigma * (dt ^ 0.5))
    d = Exp(- q * dt - sigma * (dt ^ 0.5))
    p = (Exp((r – q) * dt) - d) / (u - d)

    Dim S() As Double
    ReDim S(n + 1, n + 1) As Double
        For i = 1 To n + 1
            For j = i To n + 1
                S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
            Next j
        Next i

    Dim Opt() As Double
    ReDim Opt(n + 1, n + 1) As Double
        For i = 1 To n + 1
            Opt(i, n + 1) = Application.Max(k - S(i, n + 1), 0)
        Next i
        For j = n To 1 Step -1
            For i = 1 To j
                Opt(i, j) = Exp(-r * dt) * (p * Opt(i, j + 1) + (1 - p) * Opt(i + 1, j + 1))
            Next i
        Next j
    BinomialPutEuropean = Opt(i, j)
    Next j
    Next

End Function
Asian Options

We have to note that the lattice methods may be generalized to price Asian options as such we have the following:

\[ C_{N,i} = \text{Max} \left[ \text{Average} \left( S_u d^{N-i} \right) - K, 0 \right] \]

\[ P_{N,i} = \text{Max} \left[ K - \text{Average} \left( S_u d^{N-i} \right), 0 \right] \]

As such we assume the following:

\[ u = e^{\sigma \sqrt{\Delta t}} \quad d = e^{-\sigma \sqrt{\Delta t}} \quad a = e^{r \Delta t} \]

The risk neutral probability is given by the following for up and down respectively:

\[ p = \frac{a - d}{u - d} \quad 1 - p \]

The prices are given by:

\[ C_{j,i} = e^{-r \Delta t} \left[ p \ C_{j+1,i+1} + (1 - p) C_{j+1,i} \right] \]

\[ P_{j,i} = e^{-r \Delta t} \left[ p \ P_{j+1,i+1} + (1 - p) P_{j+1,i} \right] \]

The pricing formula depends from the number of steps that is a function of time of maturity so we do run the simulated result by assuming nodes equal to time for two that is equal to assume that the average will be computed two times for years, as such we have the following result for Asian Call options with respect the European options:

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( K )</th>
<th>( r )</th>
<th>( S_0 )</th>
<th>( T )</th>
<th>Binomial Asian</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1,5</td>
<td>1</td>
<td>0,42186</td>
<td>0,52955</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1,25</td>
<td>1</td>
<td>0,21075</td>
<td>0,27956</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1</td>
<td>1</td>
<td>0,02070</td>
<td>0,04477</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>0,75</td>
<td>1</td>
<td>0,00000</td>
<td>0,00000</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>0,5</td>
<td>1</td>
<td>0,00000</td>
<td>0,00000</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1,5</td>
<td>2</td>
<td>0,48482</td>
<td>0,55824</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1,25</td>
<td>2</td>
<td>0,24969</td>
<td>0,30830</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1</td>
<td>2</td>
<td>0,04406</td>
<td>0,07427</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>0,75</td>
<td>2</td>
<td>0,00000</td>
<td>0,00030</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>0,5</td>
<td>2</td>
<td>0,00000</td>
<td>0,00000</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1,5</td>
<td>5</td>
<td>0,56863</td>
<td>0,63930</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1,25</td>
<td>5</td>
<td>0,33150</td>
<td>0,38975</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>1</td>
<td>5</td>
<td>0,11394</td>
<td>0,15236</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>0,75</td>
<td>5</td>
<td>0,00518</td>
<td>0,01308</td>
</tr>
<tr>
<td>0,07</td>
<td>1</td>
<td>0,03</td>
<td>0,5</td>
<td>5</td>
<td>0,00000</td>
<td>0,00001</td>
</tr>
</tbody>
</table>
And the following result for Asian Put options:

<table>
<thead>
<tr>
<th>$\sigma_S$</th>
<th>$K$</th>
<th>$r$</th>
<th>$S_t$</th>
<th>$T$</th>
<th>Binomial Asian</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>1</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>1</td>
<td>0.02107</td>
<td>0.01522</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>1</td>
<td>0.21149</td>
<td>0.22045</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>1</td>
<td>0.42260</td>
<td>0.47045</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>2</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>2</td>
<td>0.00000</td>
<td>0.00007</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>2</td>
<td>0.22058</td>
<td>0.19207</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>2</td>
<td>0.45572</td>
<td>0.44176</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.5</td>
<td>5</td>
<td>0.00002</td>
<td>0.00001</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>1.25</td>
<td>5</td>
<td>0.00111</td>
<td>0.00046</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.75</td>
<td>5</td>
<td>0.15121</td>
<td>0.12379</td>
</tr>
<tr>
<td>0.07</td>
<td>1</td>
<td>0.03</td>
<td>0.5</td>
<td>5</td>
<td>0.38425</td>
<td>0.36071</td>
</tr>
</tbody>
</table>

Appendix

To run the simulation we used the following VBA code:

```vba
Function BinomialAsianCall(Spot, k, T, r, sigma, n)
    Dim dt As Double, u As Double, d As Double, p As Double
    dt = T / n
    u = Exp(sigma * (dt ^ 0.5))
    d = 1 / u
    p = (Exp(r * dt) - d) / (u - d)
    Dim S() As Double
    ReDim S(n + 1, n + 1) As Double
    For i = 1 To n + 1
        For j = i To n + 1
            S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
        Next j
    Next i
    Dim Opt() As Double
End Function
```
ReDim Opt(n + 1, n + 1) As Double

For i = 1 To n
    Opt(i, n + 1) = Application.Max(Application.Average(S(i, n + 1), S(i + 1, n + 1)) - k, 0)
Next i

For j = n To 1 Step -1
    For i = 1 To j
        Opt(i, j) = Exp(-r * dt) * (p * Opt(i, j + 1) + (1 - p) * Opt(i + 1, j + 1))
        BinomialAsianCall = Opt(i, j)
    Next i
    Next j

End Function

Function BinomialAsianPut(Spot, k, T, r, sigma, n)
Dim dt As Double, u As Double, d As Double, p As Double

dt = T / n
u = Exp(sigma * (dt ^ 0.5))
d = 1 / u
p = (Exp(r * dt) - d) / (u - d)

Dim S() As Double
ReDim S(n + 1, n + 1) As Double

    For i = 1 To n + 1
    For j = i To n + 1
        S(i, j) = Spot * u ^ (j - i) * d ^ (i - 1)
    Next j
Next i

Dim Opt() As Double
ReDim Opt(n + 1, n + 1) As Double

For i = 1 To n
    Opt(i, n + 1) = Application.Max(k - Application.Average(S(i, n + 1), S(i + 1, n + 1)), 0)
Next i

For j = n To 1 Step -1
    For i = 1 To j
        Opt(i, j) = Exp(-r * dt) * (p * Opt(i, j + 1) + (1 - p) * Opt(i + 1, j + 1))
    Next i

    BinomialAsianPut = Opt(i, j)
Next i

Next j

End Function
Local Volatility

Now it is interesting to introduce the concept of implied volatility, in practice the implied volatility is the value of volatility that gives you the market prices of the options. The problem is geometric with respect the normal distribution or the cumulative of the normal distribution, as such we may get the implied volatility by using the following formulation:

$$\frac{\sigma T}{2} = \sqrt{2\pi} \frac{C(S, K, T)}{S + K}$$

The market prices are not continuous processes, so we have to model the dynamic of the jump to obtain the effective market prices of the options. We start with the presentation of a jump diffusion process:

$$\frac{dS(t)}{S(t)} = \mu(t)dt + \sigma(t)dWs + P(dt)N[a, \sigma]$$

$P(dt)$ denotes a Poisson distribution and counts the number of jumps that are measured by the Normal distribution that is perfectly correlated with the Wiener process, the jump has the same direction. The problem it is easy to solve because in real markets the jump happens in every instant because the markets prices are not continuous as result we may solve the equation in the following way:

$$\frac{dS(t)}{S(t)} = (\mu - a)dt + \sigma(j)dWj$$

Where:

$$\sigma(j) = \sigma(t) + \sigma$$

So the effective volatility may be decomposed in two parts, a continuous part and a jump part so by taking the instantaneous volatility we may obtain the effective market prices of the options that may be approximate by sharing for two the three month volatility. If we want to refuse this hypothesis we have to accept that the implied volatility is given as jump process, as such we have the following equality:

$$2\sigma T = \sqrt{2\pi} \frac{C(S, K, T)}{S + K} + \sqrt{2\pi} \left[\frac{C(S, K, T)}{S + K}\right]^2$$

The volatility increases as the option goes in the money and decreases as the option goes out money. Moreover, the implied volatility extracted is an increasing function of interest rate. The interesting fact is that the value extracted for options at the money usually are the half of three month volatility. These induce to search for a geometric relation between historical volatility, interest rate and the grade the options are in the money such that we may obtain the smile phenomenon with the following relation:
\[ \sigma = \sqrt{r(t)} 2 \sigma_h + \sqrt{2\pi} \frac{\text{Max}(S - K, K - S) \sqrt{1 - a}}{S + K} \]

Otherwise, we may use the following equality:

\[ \sigma = \sigma_h + \sqrt{2\pi} \frac{\text{Max}(S - K, K - S) \sqrt{1 - a}}{S + K} \]

The historical volatility must be the volatility at the money and by calibrating the parameters \(a\) we may obtain different grades of skew for Call and Put options. We may see an example for \(\sigma_h = 10\%\) and \(a = 2\):

We may see an example with \(a = 1.5\), as such we have the following prospect:
We are assuming symmetric parameter skew but we may have different grades of skew for Call and Put options such that we have the following formulation:

\[
\sigma = \sigma_h + \sigma_a \sqrt{\frac{2\pi \max(S - K, 0)}{S + K}} + \sigma_b \sqrt{\frac{2\pi \max(K - S, 0)}{S + K}}
\]

Where we have to calibrate the different parameters skew \(a\) and \(b\) for Call options and Put options respectively. We may extend the analysis to the case of local volatility that represents a kind of average on continuous volatilities in stochastic volatilities world. The undiscounted final pays off of the options prices with different strike yields the risk neutral density function \(\varphi\) through the following relation:

\[
C(S, T, K) = \int_K^{+\infty} dS_T \varphi(S_T, T, S_0)(S_T - K)
\]

Differentiating twice with respect to \(K\) we obtain the following:

\[
\varphi(S_T, T, S_0) = \frac{\partial^2 C}{\partial^2 K}
\]

Furthermore, by differentiating, we obtain the following:

\[
\frac{\partial C}{\partial T} = \frac{\sigma^2 K^2}{2} \frac{\partial^2 C}{\partial^2 K}
\]

By inverting we obtain:

\[
\sigma^2(K, T, S_0) = \frac{\frac{\partial C}{\partial T}}{\frac{1}{2} K^2 \frac{\partial^2 C}{\partial^2 K}}
\]

For a given sets of values of European and American options for each strike and maturities gives us a set of local volatilities.
Valuing Barrier Options

We may obtain the value of in barrier alive from the following equalities:

\[ C(S, K, T) = \text{Call out}(S, K, H, T) + \text{Call in alive}(S, K, H, T) \]

\[ P(S, K, T) = \text{Put out}(S, K, H, T) + \text{Put in alive}(S, K, H, T) \]

\[ S < H \]

\[ \text{Call out}(S, K, H, T) = N[d_1] C(S, K, T) \]

\[ d_1 = \frac{\ln \left( \frac{H P(T)}{S} \right) - \frac{1}{2} \sigma^2 T}{\sigma \sqrt{T}} \]

\[ \text{Call in alive}(S, K, H, T) = (1 - N[d_1]) C(S, K, T) \]

\[ \text{Put out}(S, K, H, T) = N[d_1] P(S, K, T) \]

\[ \text{Put in alive}(S, K, H, T) = (1 - N[d_1]) P(S, K, T) \]

\[ S > H \]

\[ \text{Call out}(S, K, H, T) = N[d_1] C(S, K, T) \]

\[ d_1 = \frac{\ln \left( \frac{S}{H P(T)} \right) - \frac{1}{2} \sigma^2 T}{\sigma \sqrt{T}} \]

\[ \text{Call in alive}(S, K, H, T) = (1 - N[d_1]) C(S, K, T) \]

\[ \text{Put out}(S, K, H, T) = N[d_1] P(S, K, T) \]

\[ \text{Put in alive}(S, K, H, T) = (1 - N[d_1]) P(S, K, T) \]
Intensity Model

Now we will present an intensity model approach that is based on the instantaneous probability of default. The price of a credit with face value 1 may be expressed by the following formulation:

\[ 1 - h(t)(1 - R)dt \]

Where \( h(t) \) denotes the probability of default and \( R \) the recovery rate. The formulation may be rewritten by the following:

\[ e^{-h(t)(1 - R)T} \]

This is the survival probability, so by weighting the face value of the credit with the survival probability we may obtain a credit value adjustment (CVA). On the other side we have the following for the probability of default:

\[ 1 - e^{-h(t)(1 - R)T} \]

We have along the interest rate models the following:

\[ Credit\ Risk\ Yield = r(T) + h(t)(1 - R) + \frac{1}{2} \sigma_h^2 T \quad \sigma_h = \sqrt{PD \ast (1 - PD) (1 - R)} \]

On the credit risk yield we may have liquidity risk as in the interest rate models. We have assumed no drift for the probability of default that is equal to assume that it is stable. From this we may have the CVA by considering the liquidity risk, so the main point is the variance such that we may consider the case of bilateral credit risk and the wrong way risk by taking the variance between the two references and computing the correlation coefficient and the PD spread between the two references. Indeed, we may have a greater credit risk yield due to the systemic risk, but we may obtain the information with the copula approach. The systemic risk may be estimated on the base of deco relation risk, the main idea is that an entity is very correlated with other entities the systemic risk is low because the system will cover each entities with each others, it is the case for example when assets and liabilities between different entities is mixed but for instance if an entity is deco related from the others that have the liabilities of the deco related entity in their assets, i.e. negative correlation, we have the systemic risk, i.e. the deco related entity may bring a systemic risk in the mixed entities or the group, as such we have the following measure of systemic risk:

\[ N \left[ N^{-1}(1 - e^{-h(t)(1 - R)T} - \rho) \right] \]

If the correlation with the group increases the systemic risk decreases, instead, if the correlation with the group decreases the systemic risk increases such that we may have negative correlation that is really systemic risk as the deco relation increases the systemic risk increases and after begins to decrease as the entities become totally different from the group, i.e. without relationing.
References


Gatheral, J. (2002): Lecture 1: Stochastic Volatility and Local Volatility, Courant Institute of Mathematical Sciences


