

A STUDY ON THE PRICING OF DIGITAL CALL OPTIONS

Bruce Haydon, Citigroup Treasury Finance



ABSTRACT

This study attempts to examine the valuation of a binary call option through three different methods – closed form (analytical solution) using Black-Scholes, Explicit Finite-Difference, and Monte Carlo simulation using both the Forward Euler-Maruyama and Milstein methods. It was concluded that all three numerical methods were reliable estimators for the value of digital call options.

INTRODUCTION

A standard option is a contract that gives the holder the right to buy or sell an underlying asset at a specified price on a specified date, with the payoff depending on the underlying asset price. The call option gives the holder the right to buy an underlying asset at a strike price; the strike price is termed a specified price or exercise price. Therefore the higher the underlying asset price, the more valuable the call option (digital or vanilla).

If the underlying asset price falls below the strike price, the holder would not exercise the option, and payoff would be zero. The digital call option is an exotic option with discontinuous payoffs, meaning they are not linearly correlated with the price of the underlying. The contract pays off a fixed, predetermined amount if the underlying asset price is beyond the strike price on its expiration date.

Digital options (also known as *binary* options) have two general types: *cash-or-nothing* or *asset-or-nothing* options. In the first type, a fixed amount of cash is paid at expiry if option is in-the-money, whilst the second pays out the value of the underlying asset. Again, these contracts have a payoff at expiry that is discontinuous in the underlying asset price. In this experiment, we will be using the cash-or-nothing variant, with a payoff value of 1. This contract pays \$1 at expiry; time T , if the asset price is then greater than the exercise price E .

The payoff is defined in terms of the Heaviside 'unit step' function:

$$\mathcal{H}(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Which for a digital call option becomes

$$\text{Payoff}(S) = \mathcal{H}(S) = \begin{cases} 1 & S(T) > E \\ 0 & \text{otherwise} \end{cases}$$

The diagram below shows the value (red line) of a digital call at some point in time before expiration, and also illustrates the payoff function (blue line).

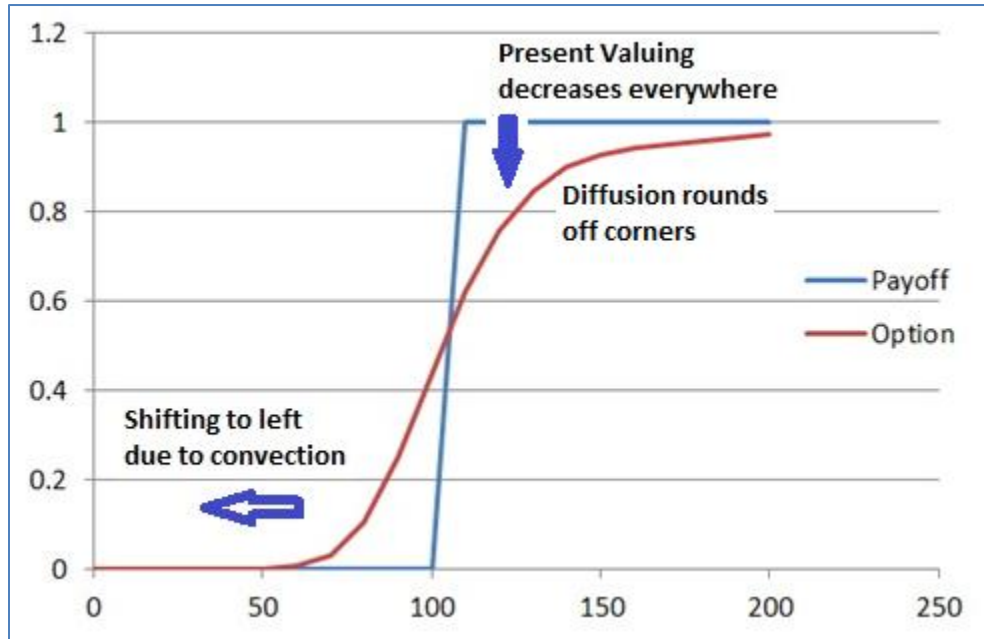


Figure 1 - Digital Call Option Payoff vs. Value of Underlying

Each element of the Black-Scholes Equation impacts the shape of the option value.

$-rV$: Discounting from expiration using TVM exerts downward pressure.

$\frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}$: Diffusion term rounds off the corners of the option value graph

$+rS \frac{\partial V}{\partial S}$: Convection shifts the profile to the left

For a binary option, the Black-Scholes formula is given by:

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

The payoff function for the binary call option:

$$V(S, T) = \mathcal{H}(S_T - E) = \begin{cases} 1 & S_T > E \\ 0 & S_T \leq E \end{cases}$$

S is the spot price of the underlying financial asset, t is the time, $E > 0$ is the strike price, T the expiry date, $r \geq 0$ the interest rate and σ is the volatility of S :

This has a closed-form solution :

$$V(S, t) = e^{-r(T-t)} N(d_2)$$

Where:

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp(-\phi^2/2) d\phi \quad d_2 = \frac{\log(S/E) + (r - D - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

The expected value of the discounted payoff under the risk-neutral density \mathbb{Q} is represented by

$$V(S, t) = \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T r_r d\tau} \text{Payoff}(S_T) \right]$$

An attempt will be made to solve the equation using three methods: (1) Analytical Solution, (2) Explicit Finite Difference Method using backwards marching scheme, and (3) Monte Carlo Simulation using both a Forward Euler-Maruyama and Milstein scheme.

Using the risk-neutral valuation approach, it can be seen

$$\begin{aligned} e^{-r(t-T)} \int_{-\infty}^{+\infty} V(s, t) f\{s(t)\} ds_t &= e^{-r(t-T)} \int_x^{+\infty} f\{s(t)\} ds_t \\ &= e^{-r(t-T)} P(s_t > x) \end{aligned}$$

Note the limits on the integral change to x to positive infinity due to the properties of the Heaviside function. If the asset price was below x, the value would be zero and $V(s, t)$ would be zero as well.

Recalling that the asset price follows a lognormal distribution

$$P(s_t > x) = P(\ln s_t > \ln x)$$

Common Parameters

In performing the calculations, we will use standard inputs into models.

Option Type:	Digital Call
Volatility:	20%
Expiration	1 (100 timesteps)
Risk-free rate:	5%
Strike Price:	\$100
S_0 (initial stock price):	\$100

(1) Numerical Solution

The solution using Black-Scholes closed form solution is as follows:

Price of a digital call option under Black Scholes: $V(S, t) = e^{-r(T-t)} N(d_2)$

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp(-\phi^2/2) d\phi \qquad d_2 = \frac{\log(S/E) + (r - D - \frac{1}{2}\sigma^2)(T - t)}{\sigma\sqrt{T - t}}$$

Given previously: $r = 0.05, t = 0, T = 1, E = \$100, \sigma = 0.2, S_0 = \100

$$\therefore d_2 = \frac{\ln(\frac{100}{100}) + (0.05 - 0.5(0.2^2))(1-0)}{0.2\sqrt{1-0}} = \frac{\log(1) + (0.05 - 0.02)(1)}{0.2\sqrt{1}} = \frac{0 + (0.03)}{0.2} = \mathbf{0.15}$$

$\therefore N(d_2) = N(0.1500) = \mathbf{0.559618}$ (using cumulative distribution function for the normal distribution).

$$\therefore V(S, t) = e^{-0.05(1-0)} * (0.559618) = e^{-0.05} * (0.559618) = (0.95) * (0.559618) = \mathbf{0.532325}$$

(1) EXPLICIT FINITE-DIFFERENCE METHOD

Finite-difference methods are designed for determining numerical solutions of differential equations using a grid/mesh, similar to the way the Binomial model uses a tree representation. In working with a mesh, the contract value at all points in the stock price-time continuum will be calculated. The Explicit Finite Difference method facilitates the transformation from a differential equation (Black-Scholes) to a difference equation, since its grid/mesh is 'square' and symmetric.

The asset is assumed to follow a lognormal random walk; the interest rate is taken to be as the continuous risk-free rate (following the no-arbitrage principal). The Explicit Finite-Difference valuation method utilizes a risk-neutral valuation approach, common to the Binomial Tree method.

The method is ideally suited to this type of valuation given we are working with a low number (three) of dimensions.

Boundary/Final Conditions

In this numerical scheme, the call defines the final condition at expiration. With this method we start at expiration and work backwards incrementally towards the present. T

The parameters of Finite Difference Method were varied to determine effect on accuracy, increasing the number of asset steps to iteratively refine the answer. The results (included in the spreadsheet submitted with this paper) are shown in Table 1 below .

Algorithm

The code associated with the Explicit Finite-Difference algorithm is included as Appendix “A”.

Results

Table 1 and Figure 2 below indicate the option value of the binary call as calculated by the Explicit Finite-Difference method with 10 asset steps:

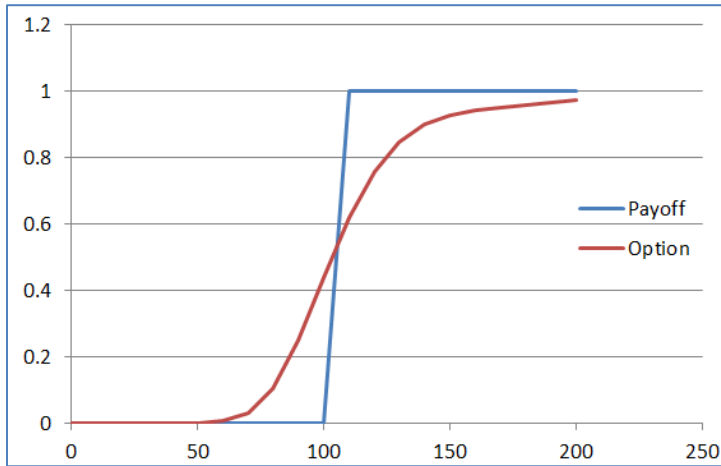


Figure 2 - Actual Payoff Function of EFD Method

The numerical experiment was performed using the algorithm to determine how the Explicit Finite-Difference method pricing would contrast against the pricing derived analytically from the Black-Scholes equation. Asset step size was increased from 10 through to 160 using a time period of 1. Figure 3 shows the effect of increasing asset step size on the degree of error, when comparing the EFD numerically derived result against the BSE analytically derived price (\$0.53233). It’s clear there is initially an exponential correction in the error rate in increasing asset steps, which then transitions to an approximately linear decay as the number of asset steps continues to increase.

Table 1 - Numerical Results for Cash-or-Nothing digital call option (s=100)

Analytical Solution:		0.53233											
# Asset Steps	10	20	30	40	50	60	70	80	90	100	150	160	
FD Value	0.33524	0.43867	0.47	0.48559	0.49492	0.50115	0.50559	0.50893	0.51153	0.5136	0.51984	0.5221	
% Error vs. BS value	58.79%	21.35%	13.26%	9.62%	7.56%	6.22%	5.29%	4.60%	4.07%	3.65%	2.40%	1.96%	
Time factor	1	8	27	64	125	216	343	512	729	1000	3375	4096	

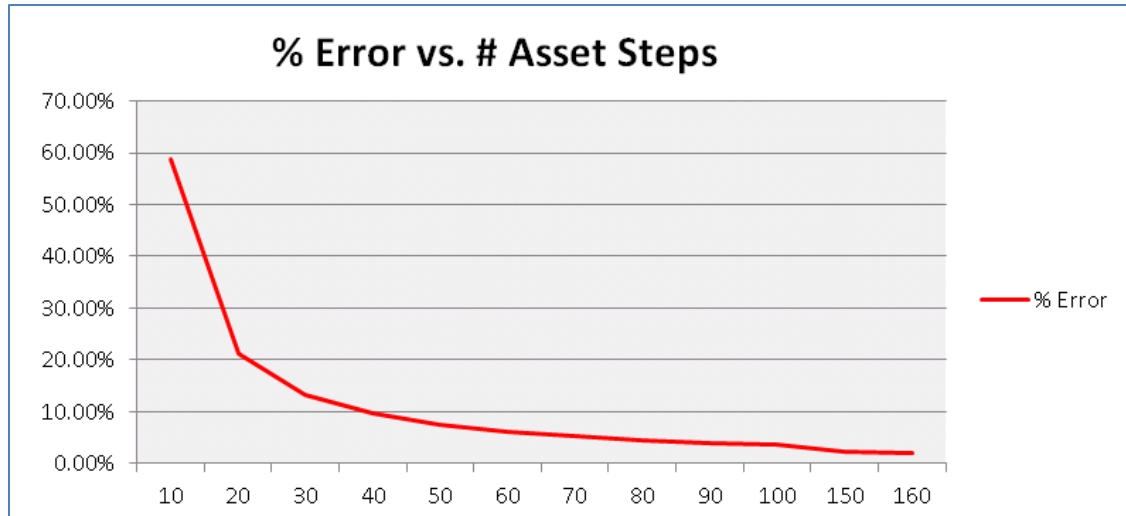


Figure 3 - % Error vs Asset Step Size

(2) MONTE CARLO SIMULATION (Forward Euler-Maruyama and Milstein Corrected)

Two types of Monte Carlo simulation were run as part of this investigation. The first type was the regular Forward Euler-Maruyama method, while the second was the Milstein method incorporating a correcting term.

The fair value of an option in the Black–Scholes world is the present value of the expected payoff at expiry under a *risk-neutral* lognormal random walk for the underlying.

Given the risk-neutral random walk for S is

$$dS = rS dt + \sigma S dX$$

We can therefore write: $option\ value = e^{-r(T-t)}E[payoff(S)]$

Provided the expectation is in relation to the risk-neutral random walk, and not the actual one.

Therefore the methodology used for MC simulation is:

1. Simulate the risk-neutral random walk as discussed below, starting at today's value of the asset S_0 , over the required time horizon. This time period starts today and continues until the expiry of the option. This gives one realization of the underlying price path.
2. For this realization calculate the option payoff.
3. Perform many more such realizations over the time horizon.
4. Calculate the average payoff over all realizations.
5. Take the present value of this average, which results in the derived the option value.

(a) Forward Euler-Maruyama Method

The Forward Euler-Maruyama method involves the use of the equation

$$S_{t+\delta t} \sim S_t (1 + r\delta t + \sigma\phi\sqrt{\delta t})$$

Which is based on Geometric Brownian Motion (GBM), specifically the relationship

$$\frac{\delta S_t}{S_t} = \frac{S_{t+\delta t} - S_t}{S_t} \sim r\delta t + \sigma\phi\sqrt{\delta t}$$

The Euler-Maruyama method represents a discrete way of simulating the time series for asset S , and has the advantage of being easy to apply to any stochastic differential equation, irrespective of complexity or order.

The calculation was performed using 16,378 discrete simulations, each representing a unique path through time and a constant $\delta t=0.01$. The number of simulations was increased from 100 to 16,000 while keeping all other variables constant to determine the effect on error.

Results

Table 2 below represents the numerically-derived results from the Monte Carlo simulation using the Forward Euler-Maruyama method. Figure 4 indicates the relationship between error and number of individual simulation paths.

Table 2 - Euler Method Numerical Results

Euler Simulation		
# Sims	Result	% Error
100	0.527414	0.93%
500	0.527828	0.85%
1000	0.541659	1.72%
2000	0.540504	1.51%
3000	0.542337	1.85%
4000	0.542065	1.80%
5000	0.537718	1.00%
6000	0.532124	0.04%
7000	0.53302	0.13%
8000	0.534049	0.32%
9000	0.534003	0.31%
10000	0.53235	0.00%
11000	0.531775	0.10%
12000	0.52994	0.45%
13000	0.531775	0.10%
14000	0.531971	0.07%
15000	0.531258	0.20%
16000	0.531823	0.09%

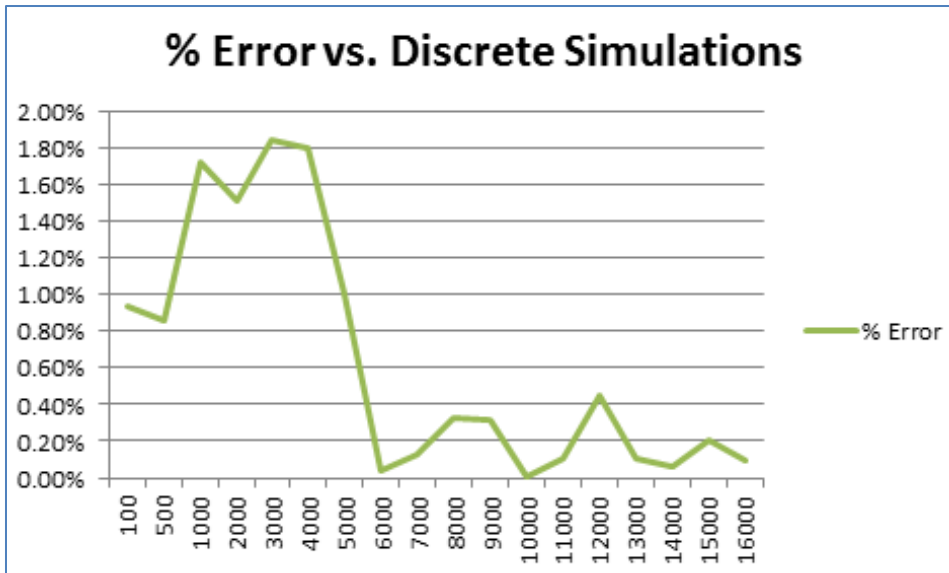


Figure 4 - Error vs. Discrete Simulations - Euler Method

(b) Milstein Method

The Milstein method of Monte Carlo simulation attempts to increase accuracy through the use of an additional second-order term.

By performing a Taylor series expansion of the exact solution

$$e^{(r-\frac{1}{2}\sigma^2)\delta t + \sigma\phi\sqrt{\delta t}} \sim 1 + (r - \frac{1}{2}\sigma^2)\delta t + \sigma\phi\sqrt{\delta t} + \frac{1}{2}\sigma^2\phi^2\delta t$$

The following result is obtained:

$$S_{t+\delta t} \sim S_t \left(1 + r\delta t + \sigma\phi\sqrt{\delta t} + \frac{1}{2}\sigma^2(\phi^2 - 1)\delta t + .. \right)$$

This equation differs from the Euler method at $O(\delta t)$ by the term $\frac{1}{2}\sigma^2(\phi^2-1)\delta t$. This additional term is referred to as the *Milstein correction*, and serves as a stochastic effect to reduce simulation error. While the Euler method has an error $O(\delta t)$, the Milstein method's error is reduced to $O(\delta t^2)$, theoretically making it a more accurate estimator.

Results

Table 2 represents the numerically-derived results from the Monte Carlo simulation using the Milstein method with the correcting stochastic term. Figure 5 indicates the relationship between error and number of individual Milstein simulation paths.

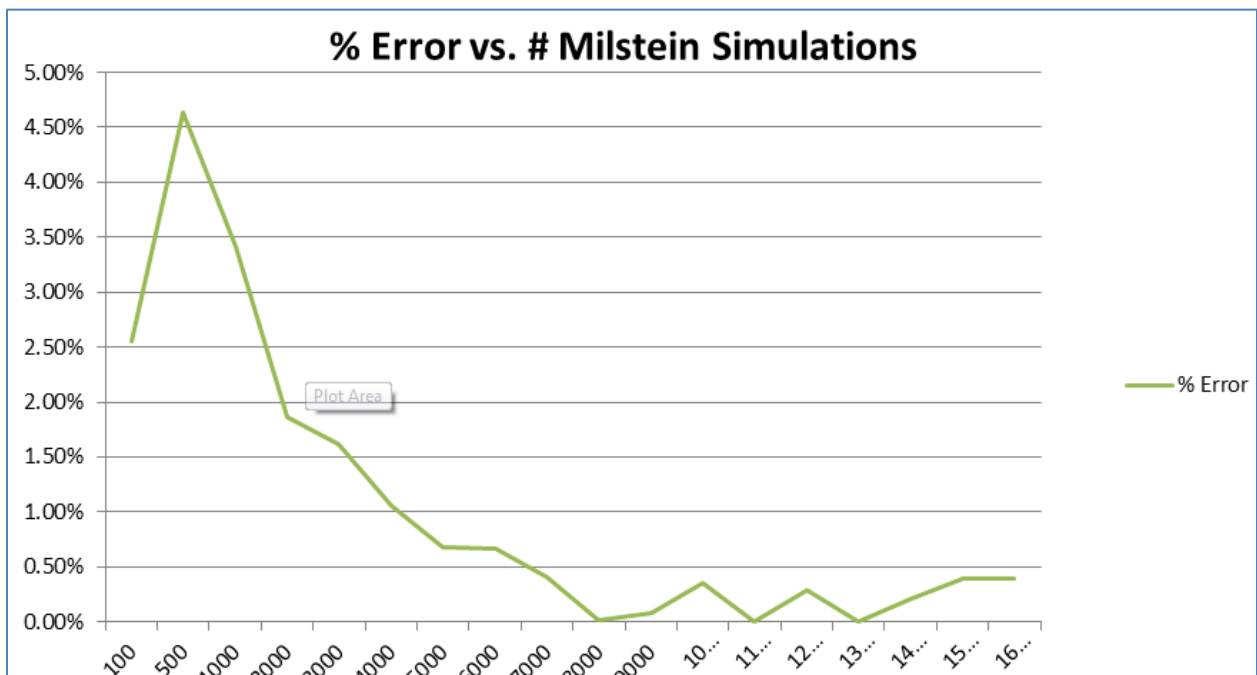


Figure 5 – Error vs. Discrete Simulations: Milstein Method

Table 3 - Milstein Method Numerical Results

Milstein Simulation		
# Sims	Result	% Error
100	0.546251	2.55%
500	0.558206	4.64%
1000	0.551162	3.42%
2000	0.542405	1.86%
3000	0.541069	1.62%
4000	0.538024	1.06%
5000	0.536006	0.69%
6000	0.535928	0.67%
7000	0.534515	0.41%
8000	0.532384	0.01%
9000	0.53189	0.08%
10000	0.534252	0.36%
11000	0.532381	0.01%
12000	0.530818	0.28%
13000	0.532381	0.01%
14000	0.531224	0.21%
15000	0.530243	0.39%
16000	0.530218	0.40%

(3) CONCLUSIONS

The Black-Scholes method is an exact calculation of the option value for a predetermined stream of underlying asset prices. The three analytical solutions investigated here served as the basis for determining their efficiency and accuracy at numerical valuation. In this study we derived the analytical solution for the binary option in the Black-Scholes method and numerical solutions for the Explicit Finite Difference method, the Forward Euler- Maruyama method, and the Milstein method. Examples were provided in all cases for the numerical methods to conclude that all three numerical methods provide valid estimates of the digital call option pricing as compared to the Black-Scholes method.

In the case of the Explicit-Finite Difference method, there was a fairly deterministic relationship between the increase in asset step size and accuracy, and we were able to obtain a result with a slightly less than 2% error factor. The Forward Euler- Maruyama method provided a less predictable result set, although one of the trials (10,000 simulations) produced an exact result to five decimal places – the best of all three numerical methods. Unfortunately, this error reduction did not appear to converge and continued to range between 0.00% and 0.4% as further simulation paths were added. The Milstein method of Monte Carlo simulation produced a similar pattern of error rate vs. number of simulation paths, with the minimum observed percentage error of 0.01% occurring at 8,000 and 13,000 simulation

paths. Again, the accuracy ranged between 0.01 and 0.4% beyond 8,000 iterations. This would tend to indicate that the stochastic correction provided by the Milstein method may not correlate with the number of simulation paths within the order of magnitude used in this study. It would be useful to provide a further study with orders of magnitude greater in simulation paths to determine if in fact the Milstein method outperforms its simpler Euler counterpart.

All three methods can be considered as accurate predictors for digital call-option pricing.