Spreadsheets in Education (eJSiE)

Volume 7 | Issue 2 Article 5

3-18-2014

Multivariate Monte-Carlo Simulation and Economic Valuation of Complex Financial Contracts: An Excel Based Implementation.

Timothy J. Kyng
Macquarie University, timothy.kyng@mq.edu.au

Otto Konstandatos
University of Technology, Sydney, otto.konstandatos@uts.edu.au

Multivariate Monte-Carlo Simulation and Economic Valuation of Complex Financial Contracts: An Excel Based Implementation.

Abstract
The economic valuation of complex financial contracts is often done using Monte-Carlo simulation. We show how to implement this approach using Excel. We discuss Monte-Carlo evaluation for standard single asset European options and then demonstrate how the basic ideas may be extended to evaluate options with exotic multi-asset multi-period features. Single asset option evaluation becomes a special case. We use a typical Executive Stock Option to motivate the discussion, which we analyse using novel theory developed in our previous works. We demonstrate the simulation of the multivariate normal distribution and the multivariate Log-Normal distribution using the Cholesky Square Root of a covariance matrix for replicating the correlation structure in the multi-asset, multi period simulation required for estimating the economic value of the contract. We do this in the standard Black Scholes framework with constant parameters. Excel implementation provides many pedagogical merits due to its relative transparency and simplicity for students. This approach also has relevance to industry due to the widespread use of Excel by practitioners and for graduates who may desire to work in the finance industry. This allows students to be able to price complex financial contracts for which an analytic approach is intractable.

Keywords
Finance, Options, Simulation, Executive Compensation, Multivariate Normal Distribution, Cholesky Decomposition.

Multivariate Monte-Carlo Simulation and Economic Valuation of Complex Financial Contracts: An Excel Based Implementation

Otto Konstandatos
Finance Discipline Group
Faculty of Business
University of Technology, Sydney
Otto.Konstandatos@uts.edu.au

Timothy Kyng
Faculty of Business and Economics
Macquarie University
timothy.kyng@mq.edu.au

Abstract

The economic valuation of complex financial contracts is often done using Monte-Carlo simulation. We show how to implement this approach using Excel. We discuss Monte-Carlo evaluation for standard single asset European options and then demonstrate how the basic ideas may be extended to evaluate options with exotic multi-asset multi-period features. Single asset option evaluation becomes a special case. We use a typical Executive Stock Option to motivate the discussion, which we analyse using novel theory developed in our previous works. We demonstrate the simulation of the multivariate normal distribution and the multivariate Log-Normal distribution using the Cholesky Square Root of a covariance matrix for replicating the correlation structure in the multi-asset, multi period simulation required for estimating the economic value of the contract. We do this in the standard Black Scholes framework with constant parameters. Excel implementation provides many pedagogical merits due to its relative transparency and simplicity for students. This approach also has relevance to industry due to the widespread use of Excel by practitioners and for graduates who may desire to work in the finance industry. This allows students to be able to price complex financial contracts for which an analytic approach is intractable.

Keywords: Finance, Options, Simulation, Executive Compensation, Multivariate Normal Distribution, Cholesky Decomposition.

1. Introduction

Monte Carlo (MC) simulation is often used for evaluating both simple and complex financial instruments. The purpose of the paper is to explain the theory behind MC simulation valuation of financial instruments with multi-asset multi-period features, and to demonstrate the numerical implementation in Excel.

Most discussions of advanced financial engineering concepts are hampered by the difficulty and opaqueness of the mathematical underpinnings of the method. This is especially so for the student. However being able to implement some realistic examples of such financial calculations greatly aids the understanding of the method and of the financial considerations involved. We believe it may also aid the development of more sophisticated computer software for industrial applications of the methods. Our discussion should be accessible to readers and students with some knowledge of Excel, option pricing, matrix algebra and statistical theory. Our aim is to explain MC simulation and risk neutral valuation from a statistical perspective.

Use of spreadsheet software is widespread throughout the financial services industry. Kyng Tickle and Wood (2013) found that spreadsheets are used by finance graduates for both simple and complex calculations including valuation of complex financial contracts, statistical modelling and also simulation. Contact with recent graduates indicates that the valuation of complex employee share options of the type we consider here is often done by actuarial consulting firms. A recent survey undertaken in Kyng and Taylor (2008) consisting of 93 postgraduate students and 70 graduates working in the financial services industry found that students prefer learning financial mathematics using spreadsheets to the traditional pen and paper / calculator method normally used in teaching this material. Graduates find spreadsheets easy to learn and easy to use when applying financial and actuarial theory in the workplace due to their transparent nature. In this regard we hope that the implementations we present here will be of some value to industry practitioners as well as to students who may be studying at a traditional university. We have been teaching senior actuarial and finance students the theory of Monte Carlo Simulation and its application to option pricing for several years and we have found that students respond well to learning through the Excel based examples. This approach enhances their learning by allowing the transparent implementation of the method. The learning hurdles for Excel are much lower than with other software packages, and students can quickly develop spreadsheet programs for computing the value of many complex financial contracts.

Monte-Carlo Simulation is in practice a computer based numerical method. It is not feasible to implement using the calculator and pen-and-paper approach used in traditional expositions of financial and economic subject matter. The traditional pen-and-paper approach does not allow the treatment of realistic examples nor does it give much scope for scenario analysis to readily vary the input parameters. Given the complexity of the calculations and the need for the large number of trials required for accuracy, a computer implementation of the method is therefore essential. Traditional computer programming languages such as C++ and advanced computer packages such as Matlab and Mathematica are suited to these tasks, however they require a considerable overhead in learning programming syntax and methods. To master, they usually require specific study as separate subjects in their own right. Most finance and actuarial students have little or no exposure to these more advanced tools before undertaking our courses. Spreadsheets in contrast require little overhead in programming knowledge and are accessible by the beginner. The spreadsheet paradigm, where the details of calculations are developed on the screen in a manner mimicking the steps one would take on a sheet of paper, is easy to understand. This makes it a very good way to implement and illustrate the methods for teaching purposes without being distracted by the intricacies of writing code. The input assumptions, the intermediate calculations and the final results are all visible to the student, making it easier to understand how the method we're trying to teach actually works. In our experience students from industry usually come with some exposure to Excel in their daily working lives, as spreadsheet software is widely used in business. Given the student backgrounds and skills sets, spreadsheet software seems a natural choice.

This paper is organised as follows. Section 2 gives an overview of option pricing theory and statistical modelling of asset prices. We also detail the fundamental results we use in our novel approach, in particular the use of building block instruments such as power options and binary options. Section 3 covers the MC simulation pricing of plain vanilla (European) options, and explores the issue of required sample size to achieve accuracy to a desired level of confidence. Section 4 covers the generalisation of the univariate simulation methods of Section 3 to the case of multiple Log-Normally distributed assets. We utilise an algorithm derived from the constructive proof of the Cholesky square-root of a given covariance matrix (provided in Appendix B) and present diagnostics for valid covariance matrices. An example Excel implementation of the algorithm is presented. Section 5 presents a case study of the pricing of several multi-asset multi-period exotic options, including the Executive Stock Option which is the focus of this paper. Section 6 gives a brief conclusion.

2. Statistical modelling of asset prices and financial contract valuation

Monte-Carlo Simulation was first suggested as a numerical method for the economic valuation of options and other financial contracts in Boyle (1977). Since then it has become a standard tool of financial modelling. Typically this numerical approach is used in circumstances where an analytic formula for the option value is not available or difficult to obtain and where other numerical methods are not feasible. Examples include path dependent options such as Asian options which have a payoff based on the average stock price over the term of the option; spread options which have a payoff based on the difference between two asset prices, and rainbow options which have payoffs based on the value of a portfolio of assets.

Under the Black-Scholes option pricing assumptions, the maturity value of the assets which define the payoff on an option will be Log-normally distributed. This fact allows a closed form formula for the value of standard call and put European options, as first demonstrated in the celebrated paper of Black and Scholes (1973).

A linear combination of Normally distributed random variables will also be Normally distributed. In other words the Normal distribution is stable. However a linear combination of Log-normally distributed random variables will *not* be Log-normally distributed. Since the Log-normal distribution is not stable, this makes it very difficult in general to obtain analytic valuation formulae for multi asset, multi period options where each of the assets defining the payoff has its own Log-normal distribution. It may not in fact be possible, depending on the exact scenario. Accordingly other methods need to be employed and MC simulation is a common choice.

MC simulation is a very flexible method and can be used to compute estimates of the expectation of a random variable; the expectation of some function of the variable;

the variance of a random variable or function thereof; the 100×α percentile of the distribution of the variable or function as well as other things of interest.

Although the technique is quite general, our interest in Monte-Carlo simulation is for the numerical calculation of option prices, and in particular for those with non-standard or *exotic* features. These arise naturally in many areas of finance, including the valuation of Real Options in mining (for examples see Konstandatos and Kyng (2012)) as well as in the Executive compensation area we explore here.

The modern approach to financial valuation rests on the intuitively appealing notion that there is no such thing as a free lunch. An *arbitrage* is a situation where it is possible, with certainty, to make a positive profit for zero initial investment, namely to have a free lunch. To avoid free lunches therefore we require the *absence* of arbitrage opportunities in a well-functioning market. The absence of arbitrage, though taken as axiomatic, should be understood as applying in the statistical sense, rather than the usual sense that axioms of mathematics are understood. Namely, arbitrage opportunities may arise for short periods but that they will be eliminated by active traders as they attempt to exploit them in the operation of a free market.

The *Fundamental Theorem of Asset Pricing* (Harrison and Pliska (1981)) states that an arbitrage-free market *(S(t),B(t))* consisting of risky asset *(S)* and a risk-free asset *B(t)* (either cash or bond) is *complete* if and only if there exists a unique risk-neutral probability measure Q that is equivalent to the probability measure P for the dynamics of the risky assets *S(t)* and has numéraire *B(t)*. The risk-neutral probability measure expresses the risk-preferences for a risk-neutral investor. Under the risk-neutral measure, the discounted risky asset is a mathematical martingale, namely:

$$S(t)/B(t) = E_Q\{S(T)/B(T)|S_t\} \qquad (1)$$

Completeness of the market is also important because in an incomplete market the price of assets or any *contingent claim* dependent on the risky assets will not be unique. The technical details are beyond the scope of this paper, however an expanded discussion may be found in Konstandatos (2008) or Buchen (2012). An intuitive and pedagogical introduction to risk-neutral pricing may be found in Tham (2001). The simulation approach to option pricing we employ is based on the result that *any* contingent claim depending on a risky asset or even a collection of risky assets may be uniquely priced as a discounted expectation with respect to the risk-neutral measure Q.

Under the risk neutral discounted expectations approach to option valuation therefore, we compute the price of any contingent claim or option as the product

$$[expected\ payoff] \times [discount\ factor] \qquad (2)$$

In this formulation, the expected payoff is calculated under the *risk-neutral* probability distribution of the price of the underlying asset. The discount factor is calculated using the risk free interest rate for the term of the option contract, which is assumed to be a constant which may be determined from the T-bill (Treasury Bill), a risk-free government Bond rate or the return on some other essentially risk-free asset over the term of the contract. By simulating the dynamics of the assets defining the payoff using the risk-neutral distribution, Monte-Carlo simulation allows us to compute an estimate of the expected option payoff. It is a simple matter to multiply this by the discount factor to obtain the estimated economic value or price of the contract. The estimate of the price can be made more accurate by increasing the simulated sample size. In the economic valuation of options and other complex financial contracts, we typically use MC simulation to estimate both the expected payoff on the contract and the standard deviation of the payoff. The standard deviation provides information useful for deciding on the sample size required to achieve a desired level of confidence in the estimate. With a large enough sample in principle we can estimate the option price to any desired level of accuracy.

The financial derivative contracts we are interested in are options with payoffs that depend on the prices of one or more other assets. These other assets could be shares, currencies, commodities such as gold or various other tradeable items. These are usually referred to as the *underlying assets*.

The Black-Scholes model is the standard framework of financial valuation. The framework consists of a set of now-standard assumptions about financial markets. Since its publication in Black and Scholes (1973), the model and pricing methods deriving from it have been extended to cover many other more complex financial contracts. The model assumes the underlying asset price process follows the *stochastic differential equation* given by geometric Brownian motion, and makes several idealized assumptions about market frictions:

$$dS(t) = (\mu - y)\, S(t)dt + \sigma\, S(t)dW(t) \qquad (3)$$

where $W(t)$ is a standard Wiener process. The parameters $(\mu, \sigma\ )$ are known as the drift and volatility of the stock process, and are specific to each stock, which one estimates from market data. The parameter y is the continuous dividend yield of the stock. These parameters are taken as known or determinable constants in the Black Scholes model. Pricing of contingent claims is done using the so-called risk-neutral measure, under which the discounted stock process is a mathematical martingale. This can be shown to be mathematically equivalent to setting $\mu = r$, the risk-free rate of interest, for the stock drift. This means that the expected rate of return on the underlying asset is the risk free rate of interest. See Konstandatos (2008), Buchen (2012) for details.

Financial contingent claims typically have a future (time T) payoff which is some function $P(S_T)$ of the time T price of the underlying asset. It can be shown that the economic value at times $t < T$ of this contract is the discounted value of the expected payoff on the contract, where the underlying asset has the risk neutral distribution.

$$V(S,t) = e^{-r(T-t)}E\{P(S_T)|S_t\} \tag{4}$$

The most basic contingent claims are European call and put options. The *European call option* is a financial contract granting the holder the right to sell the underlying asset at the future *maturity* time T at *strike* price K, whereas the *European put* grants the holder the right to sell the underlying asset at strike K at maturity. The call option's payoff function may be written $P(S_T) = (S_T - K)I\{S_T \geq K\}$ and the European put option's payoff is given by $P(S_T) = (K - S_T)I\{S_T < K\}$. We have defined the indicator function based on some condition $\varkappa$:

$$I\{X\} = \begin{Bmatrix} 1 & \big| \; \varkappa \; true \\ 0 & \big| \; \varkappa \; false \end{Bmatrix} \tag{5}$$

Typically, we encounter conditions of the form $I\{S_T \geq X\} = \begin{Bmatrix} 1 | S_T \geq X \\ 0 | S_T < X \end{Bmatrix}$ for some value X. Indicator functions are closely related to the *Heaviside Step Function* of mathematics. They are useful in expressing the complex hurdle conditions which will arise in our later analysis.

The discounted expectations approach allows us to obtain analytic formulae for the value of the contract in some cases by evaluating the expectation, as per Harrison and Pliska (1983). In the case of the standard European calls and puts we readily obtain the following results by evaluating the Gaussian expectations for the given payoff functions:

$$V_c(S,t) = e^{-y(T-t)}S\,N(d_1) - K\,e^{-r(T-t)}S\,N(d_2) \tag{6}$$

$$V_p(S,t) = K\,e^{-r(T-t)}S\,N(-d_2) - e^{-y(T-t)}S\,N(-d_1) \tag{7}$$

where

$$[d_1, d_2] = \left[ln(S_T/S_t) + \left(r - y \pm \tfrac{1}{2}\sigma^2\right)(T-t)\right]/\sigma\sqrt{(T-t)} \tag{8}$$

This solution is mathematically equivalent to solving the Black-Scholes partial differential equation (9) on the domain $D = \{(S,t)|S > 0, 0 < t < T\}$

$$\frac{\partial V}{\partial t} + (r-y)S\,\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0 \tag{9}$$

subject to the terminal boundary condition $V(S,t) = P(S)$. This result follows from the theorem of Feynman & Kac, (see Konstandatos (2008), Buchen (2012) for details; Kac (1949) for the original formulation for parabolic PDEs).

2.1. Statistical modelling of individual risky assets

Let us consider a future maturity date at time T and let time t be the *current* date, at which we want to value some derivative contract. The stock price S_t is observable, but the future stock price at maturity, S_T is a random variable. The capital gain component of return on some asset S over the period from times t to T is measured via the logarithm of the *price relative* for the underlying asset, which is the ratio S_T/S_t. Under the dynamics assumed by the Black-Scholes model, the logarithm of the price relative is Normally distributed, with parameters corresponding to those which make the discounted stock price a mathematical martingale. This is easily seen by an elementary application of Ito's Lemma

$$ln(S(T)/S(t)) \sim N\left(\left(r - y - \tfrac{1}{2}\sigma^2\right)(T-t), \sigma^2(T-t)\right) \qquad (10)$$

The mean m and variance $s^2$of the log-price relative are, therefore under the *risk neutral* distribution:

$$m = E\{ln(S_T/S_0)\} = \left(r - y - \tfrac{1}{2}\sigma^2\right)(T-t) \qquad (11)$$

$$s^2 = var\{ln(S_T/S_t)\} = \sigma^2(T-t) \qquad (12)$$

The parameters r, y, σ are the risk free rate of interest, the dividend yield and the volatility of the asset respectively which are readily determined, see Hull (2006). The expectation of the price relative at time t is $E_t(S_T/S_t) = exp\big((r-y)(T-t)\big)$ and the expected rate of the asset's capital gain return per year is $\frac{1}{T} ln\big\{exp\big((r-y)T\big)\big\} = r - y$. Adding the dividend yield gives the total expected return on the asset per year, namely $(r-y) + y = r$ which is the risk free rate of interest. This is an essential feature of the risk neutral distribution.

Under the assumption of Geometric Brownian Motion, two log price-relatives defined over different and non overlapping intervals (u, U) and (t, T) will be independent of each other with zero correlation.

$$if\ (u, U) \cap (t, T) = \emptyset\ then\ corr\{ln(S_T/S_t), ln(S_U/S_u)\} = 0 \qquad (13)$$

Two log price relatives defined over overlapping intervals will have a non zero correlation. The correlation between the log price relatives over the intervals $(0, U)$ and $(0, T)$ is

$$corr\{ln(S_T/S_0), ln(S_U/S_0)\} = \sqrt{\frac{min(U,T)}{max(U,T)}} \quad (14)$$

See Konstandatos (2003,2008) and Buchen (2004, 2012) for further details.

We can statistically model the log price relative from current time t to some future time T as a Gaussian random variable

$$ln(S_T/S_t) = \left(r - y - \tfrac{1}{2}\sigma^2\right)(T - t) + Z \times \sigma\sqrt{(T - t)} \quad (15)$$

where $Z \sim N(0,1)$. Simple re-arrangement gives the value of the stock at time T in terms of its value at an earlier time t and an *N(0,1)* distributed random variable *Z*.

$$S_T = S_t \; exp\left[\left(r - y - \tfrac{1}{2}\sigma^2\right)(T - t) + Z \times \sigma\sqrt{(T - t)}\right]. \quad (16)$$

2.2. Statistical model of multiple risky assets

In the multi-asset, multi-period Black-Scholes framework, assets have correlated log price relative returns which are jointly multivariate Normally distributed. Suppose we have several assets $S(1), S(2), \dots, S(n)$. Consider any pair of assets $\left(S(i), S(j)\right)$ over time intervals (u, U) and (t, T). The following results define the correlation structure of the joint multivariate normal distribution. We refer the reader to Konstandatos (2003, 2008) and Buchen (2012) for further details.

For any asset $S(i)$ over a time interval (t, T) we have the log-returns obeying individual log-normal dynamics. This requires that the returns for any asset $S(i)$ have mean and variance:

$$E\{log(S(i)_T/S(i)_t)\} = \left(r - y_i - \frac{1}{2}{\sigma_i}^2\right)\tau \quad (17)$$

$$var\{log(S(i)_T/S(i)_t)\} = {\sigma_i}^2\tau \quad (18)$$

Here $\tau = T - t$ is the length of the time interval. The parameters (r, y_i, σ_i) are respectively the risk free rate of interest, the dividend yield on asset i, and the volatility of asset $S(i)$ for $i = 1, \dots, n$. Along with the correlation coefficients ρ_{ij} (defined below), they are taken as inputs to our model. As indicated previously, the risk free rate may be determined from observing the T-bill or some other risk-free government Bond rate, whereas the other parameters need to be estimated using historical data. The estimation of these parameters is outside the scope of this paper, however Hull (2006) provides details for their practical estimation. With this set of parameters each asset has an expected return equal to the risk free rate of interest. Together, these parameters will define the multivariate risk neutral distribution.

In the discussion that follows we assume the constant correlations of the standard multivariate Black Scholes model we focus on in this paper. Additional complications arise in the statistical estimation of non-constant correlations. Such complications increase the complexity of the simulation procedure. Estimating non-constant correlations requires additional statistical techniques and considerations which we do not pursue here. See Engle (2002) and Fonseca et al. (2007) for additional details.

The returns on an asset $S(i)$ over time intervals $(u,U)\ and\ (t,T)$ are uncorrelated if the time intervals don't overlap:

$$(u,U) \cap (t,T) = \emptyset \quad \Rightarrow \quad corr\{log(S(i)_U/S(i)_u)\,, log(S(i)_T/S(i)_t)\} = 0 \qquad (19)$$

Define the overlapping intervals $(t,T_K)\ and(t,T_L)$ with intersection $(t,T_K) \cap (t,T_L) = \big(t, min(T_K,T_L)\big)$ where $\tau_K = T_K - t, \tau_L = T_L - t$. Then returns on an asset $S(i)$ are correlated, with

$$corr\big\{log\big(S(i)_{T_K}/S(i)_t\big)\,, log\big(S(i)_{T_L}/S(i)_t\big)\big\} = R_{KL} \qquad (20)$$

$$R_{KL} = \sqrt{min(\tau_K,\tau_L)/max(\tau_K,\tau_L)} \qquad (21)$$

The correlation structure between the returns (log price relatives) on different pairs of assets also depends on whether we are examining the returns over overlapping time intervals or over disjoint time intervals, although the structure is slightly more complicated.

The returns on the assets $S(i)$ and $S(j)$ over non-overlapping intervals $(u,U)\ and\ (t,T)$ are also uncorrelated:

$$(u,U) \cap (t,T) = \emptyset \Rightarrow corr\{log(S(i)_U/S(i)_u)\,, log(S(j)_T/S(j)_t)\} = 0 \qquad (22)$$

The returns on the assets $S(i)$ and $S(j)$ over the same time interval will have correlation:

$$corr\{log(S(i)_T/S(i)_t)\,, log(S(j)_T/S(j)_t)\} = \rho_{ij} \qquad (23)$$

The correlation between returns on assets $(S(i),\ S(j))$ over overlapping intervals $(t,T_K)\ and(t,T_L)$ is given by:

$$corr\big\{log\big(S(i)_{T_K}/S(i)_t\big)\,, log\big(S(j)_{T_L}/S(j)_t\big)\big\} = R_{KL} \times \rho_{ij}. \quad (24)$$

This is due to the two effects of the overlapping time intervals and the different assets.

Finally, we have the result that

$$if\ \ (u,U)\ \cap\ (t,T) \neq \emptyset\ then\ min(U,T) - max(u,t) > 0 \qquad (25)$$

In this case we have

$$cov\{log(S(i)_U/S(i)_u)\,, log(S(j)_T/S(j)_t)\} = \rho_{ij}\ \sigma_i\ \sigma_j\big(min(U,T) - max(u,t)\big). \quad (26)$$

Given the above assumptions about the statistical model, we can formulate a vector of log price relatives, each one over possibly a different time interval. This vector will have the multivariate normal distribution. For most applications of interest to us all of the different log price relatives will be defined over the same time interval.

2.3. The mean vector and covariance matrix of the *n* dimensional vector of log price relatives

We may apply the results of the previous section to specify the correlation structure of the log price relatives of *n* different assets. The idea is that we specify the structure pairwise of each pair of assets under consideration. It is convenient to express the grouping as a vector. In particular, suppose we have *n* different assets $[S(1) \quad S(2) \quad \ldots \quad S(n)]$. For each asset *i* we consider the return over the overlapping intervals (t, T_i) of length $\tau_i = (T_i - t)$. This is the log price relative $Y(i) = ln(S(i)_T/S(i)_t)$.

We obtain a vector $[Y(1) \quad Y(2) \quad \ldots \quad Y(n)]$ of such log price relatives. This vector has a multivariate normal distribution.

The mean of this vector is $\underline{m'} = [m(1) \quad m(2) \quad \ldots \quad m(n)]$ where $m(i) = \left(r - y_i - \frac{1}{2}{\sigma_i}^2\right)\tau_i$. The correlation between $Y(i)$ and $Y(j)$ is $corr\big(Y(i), Y(j)\big) = \rho_{ij}R_{ij}$ where $R_{ij} = \sqrt{min(\tau_i,\tau_j)/max(\tau_i,\tau_j)}$ and the covariance between $Y(i)$ and $Y(j)$ is $cov\big(Y(i), Y(j)\big) = \rho_{ij}\sigma_i\sigma_j min(\tau_i,\tau_j)$.

2.3.1. Power and Binary Power Option contracts

In this paper we will replicate standard call and put option prices in terms of simpler instruments. The basic instrument we use is the *power option with power n,* where *n* is some constant which may or may not be an integer. This is a financial contract that pays the holder some power of the price of the underlying asset at the maturity date *T,* with payoff $P = (S_T)^n$. For example when *n = 1* the power contract provides the holder with one unit of the stock and if *n=0* the contract provides the holder with one unit of cash at maturity.

The *binary power options* also pay the holder a power of the price of the underlying asset at the maturity, conditional however on the stock price being either above or below some threshold level, namely $P = (S_T)^n I\{S_T \gtrless X\}$. The *up binary power option* is exercised if the underlying asset price S_T is above the threshold value X (the exercise price) and the *down binary power option* is exercised if S_T is below X.

The value of the power option payoff at time t depends on the stock price S_t and on other economic variables in the Black-Scholes formula. We will use the risk neutral discounted expectation approach to value this contract. Computing the discounted expectation of the payoff we may determine the value of these contracts.

The value of the power option is $V(S,t) = e^{-r(T-t)}E\{P\} = (S_t)^n exp(\gamma(n)\tau)$ where $\gamma(n) = \frac{1}{2}n^2\sigma^2 + n\left(r - y - \frac{1}{2}\sigma^2\right) - r$ and $\tau = T - t$. The calculation of the expected payoff was computed using the risk-neutral distribution. We include a derivation of this result in Appendix B.

Consider the exponent $\gamma(n)\tau$ in this formula. For *n = 0* this is $\gamma(n)\tau = -r\tau$ and the value of the contract is $V(S,t) = e^{-r\tau}$. We recognize this as the present value at time t of a $1.00 paid at time T. For *n = 1* this is $\gamma(n)\tau = -y\tau$. The value of the contract is $V(S,t) = Se^{-y\tau}$. This is the current stock price discounted at rate y for term τ.

In a similar manner *binary power options* may also be defined and valued. The binary power option which also pays some arbitrary power of the risky asset at maturity, only this time provided that an exercise condition on the stock price is met. There are in fact two types, the up-power binary and the down-power binary. The payoffs at expiry from both may be written as follows:

$$V^s(S,T) = (S_T)^n I\{sS_T > sX\} \tag{27}$$

We have introduced the notation $= \pm 1$, namely a *state indicator* determining if we're in the *up*-state for an up-power binary or in the *down* state for a down power binary respectively. We can see this since

$$if\ s = -1\ then\ sS_T > sX\ is\ equivalent\ to\ S_T < X \tag{28}$$

$$if\ s = +1\ then\ sS_T > sX\ is\ equivalent\ to\ S_T > X \tag{29}$$

The expectation may be computed similarly for the power binary option. Details are provided in Appendix B. See Konstandatos (2008) or Buchen (2012) for further details. The result is expressed in terms of the univariate Normal distribution:

$$V^s(S,t) = exp(\gamma(n)\tau)(S_t)^n N\left(s\left(d_2 + n\sigma\sqrt{\tau}\right)\right). \tag{30}$$

2.4. Binary power options as building blocks

The idea of decomposing option prices into more basic building blocks has been a recurring theme of our previous work. Konstandatos (2003, 2008) and Buchen (2004, 2012) first illustrated this approach for all European options with payoffs which are affine in the stock price. In particular, Konstandatos (2008) demonstrates a complete framework for pricing both European and also path dependent options such as multi-dimensional barrier options and lookback options in terms of recurring building blocks. It transpires that these techniques are applicable to pricing Executive Stock Options in terms of combinations of the power options and binary power options defined above.

A simple demonstration of this is as follows. The choices $n = 1$ and $s = +1$ give $\gamma(n)\tau = -y\tau$ and $d_2 + n\sigma\sqrt{\tau} = d_1$ The value of the contract is $V(S,t) = Se^{-y\tau}N(d_1)$. This is the first term in the BS formula for a call option. For $n = 0$ and $s = +1$ we get $\gamma(n)\tau = -r\tau$ and $d_2 + n\sigma\sqrt{\tau} = d_2$ The value of the contract is $V(S,t) = e^{-r\tau}N(d_2)$. This is the second term in the BS formula for a call option. It follows that a combination of long *1* unit of the first contract and short X units of the second contract will reproduce the Black Scholes formula for the European call.

Similarly, the choices $n = 1$ and $s = -1$ give $\gamma(n)\tau = -y\tau$ and $s\left(d_2 + n\sigma\sqrt{\tau}\right) = -d_1$ to give $V(S,t) = Se^{-y\tau}N(-d_1)$. This is the second term in the BS formula for a put option. The choices $n = 0$ and $s = 1$ we get $\gamma(n)\tau = -r\tau$ and $s\left(d_2 + n\sigma\sqrt{\tau}\right) = -d_2$, with value $V(S,t) = -e^{-r\tau}N(d_2)$. This is part of the first term in the BS formula for a put option. A combination of long X units of the 2nd contract and short 1 units of the 1st contract is therefore equivalent to the Black Scholes European put option formula.

It should be apparent that many types of option contracts can be expressed as a linear combination of power options and binary options. This includes as mentioned standard call options and put options, and also cash or nothing binary options, asset or nothing binary options, and *gap* call and put options which are calls and puts with exercise prices which are different to their strike prices.

3. Random number generation in Excel and Monte Carlo evaluation of standard European options.

The pricing of plain vanilla options dependent on one source of uncertainty is a classic example of the application of the MC method. We refer the reader to Brewster et al. (2012) as a readable exposition of basic aspects of the application of the technique for plain vanilla call options on a single underlying asset. In this section we recap the MC pricing of a plain vanilla call, and extend the analysis to show how the methods may be extended to provide an estimate of the accuracy of the computed price.

Although not immediately apparent, it is possible to place some confidence bounds on the numerically obtained prices of options obtained using Monte Carlo simulation. We will use the standard European call option to illustrate. We begin by noting that the payoff at maturity for the standard call is

$$P = max(S_T - X, 0) = (S_T - X)I\{S_T \geq X\} \qquad (31)$$

This payoff is a random variable with a mean, variance and a standard deviation which we may determine analytically in terms of our power binary options. For the call (and put, which we omit) option payoffs it is possible to do this analytically but others may require numerical evaluation. We will use these results to illustrate the MC method for the European call option and compare the numerical results with the analytic results. We start by noting that the expectation of the payoff is the call option price multiplied by the factor $exp(r\tau)$

$$E(P) = \left[e^{(r-y)\tau} S_t N(d_1) - XN(d_2)\right] \qquad (32)$$

To compute the variance, we note that

$$\begin{aligned} P^2 &= \left({S_T}^2 - 2S_T X + X^2\right)(I\{S_T \geq X\})^2 \\ &= {S_T}^2 I\{S_T \geq X\} - 2XS_T I\{S_T \geq X\} + X^2 I\{S_T \geq X\}. \end{aligned} \qquad (33)$$

So that

$$E(P^2) = E\left({S_T}^2 I\{S_T \geq X\}\right) - 2XE\left\{S_T I\{S_T \geq X\}\right\} + X^2 E(I\{S_T \geq X\}). \qquad (34)$$

We simply note that the right hand side involves a combination of binary power option payoffs with *n = 2*, *n = 1* and *n = 0*. The relevant power binary multiplied by the accumulation factor $exp(r\tau)$ gives each expectation. Using our power binary formula we obtain the discounted expectations:

$$exp(-r\tau)E\left({S_T}^2 I\{S_T \geq X\}\right) = {S_t}^2 e^{(\sigma^2 + r - 2y)\tau} N\left(d_2 + 2\sigma\sqrt{\tau}\right) \qquad (35)$$

$$exp(-r\tau)E(2XS_T I\{S_T \geq X\}) = 2XS_t e^{-y\tau} N\left(d_2 + 1\sigma\sqrt{\tau}\right) \qquad (36)$$

$$exp(-r\tau)\ E(X^2 I\{S_T \geq X\}) = X^2 e^{-r\tau} N\left(d_2 + 0\sigma\sqrt{\tau}\right) \qquad (37)$$

The undiscounted expectations are thus:

$$E\left({S_T}^2 I\{S_T \geq X\}\right) = {S_t}^2 e^{(\sigma^2 + 2r - 2y)\tau} N\left(d_2 + 2\sigma\sqrt{\tau}\right), \qquad (38)$$

$$E(2XS_T I\{S_T \geq X\}) = 2XS_t e^{(r-y)\tau} N(d_1) \qquad (39)$$

$$E(X^2 I\{S_T \geq X\}) = X^2 N(d_2) \qquad (40)$$

After some algebra we obtain

$$var(P) = S_t{}^2 e^{(\sigma^2+2r-2y)\tau} N\left(d_2 + 2\sigma\sqrt{\tau}\right) - 2XS_t e^{(r-y)\tau} N(d_1) + X^2 N(d_2) - \left[e^{(r-y)\tau} S_t N(d_1) - XN(d_2)\right]^2 \quad (41)$$

We now consider a call option example over a stock currently valued at *S = $10.00,* with a term of *3* months (*T=0.25* years), with an exercise price of *X = $10.00.* Assume the stock pays no dividend during the term of the option and that the volatility is *40%* per annum and the risk free rate is *10%* per annum.

For our example we can compute the expected payoff and the variance of the payoff. We can also compute the expectation and variance of the sample payoff for a simulated sample of size *n*. This is what we do next. For our data we have

$$t = 0, T = 0.25, S_t = 10, X = 10, r = 10.00\%, \sigma = 40.00\%, y = 0.00\%, \tau = T - t = 0.25$$

The mean of the log price relative is

$$m = \left(r - y - \frac{1}{2}\sigma^2\right)\tau = \left(0.10 - 0.00 - \frac{1}{2}0.40^2\right)0.25 = 0.005.$$

The standard deviation of the log price relative is

$$s = \sigma\sqrt{\tau} = 0.40\sqrt{0.25} = 0.20.$$

Based on this we obtain:

$$d_2 = \frac{ln(S_t/X) + m}{s} = \frac{ln(10.0/10.0) + 0.005}{0.20} = 0.025.$$

We need to compute the prices of the up-type binary power options with powers *n = 0,1,2* to determine the value of the option, the expected option payoff and the variance of the option payoff. Details of the calculations are as follows:

$$d_2 + 0\sigma\sqrt{\tau} = 0.025 + 0.00 = 0.025 \rightarrow N(d_2) = 0.509973,$$

$$d_2 + 1\sigma\sqrt{\tau} = 0.025 + 0.20 = 0.225 \rightarrow N\left(d_2 + \sigma\sqrt{\tau}\right) = 0.589010,$$

$$d_2 + 2\sigma\sqrt{\tau} = 0.025 + 2 \times 0.20 = 0.425 \rightarrow N\left(d_2 + 2\sigma\sqrt{\tau}\right) = 0.664582,$$

$n = 0 \rightarrow \gamma(n)\tau = -r\tau = -2.50\%,$

$n = 1 \rightarrow \gamma(n)\tau = -y\tau = -0.00\%,$

$n = 2 \rightarrow \gamma(n)\tau = (\sigma^2 + r - 2y)\tau = (0.16 + 0.10 - 2 \times 0.00)0.25 = 6.50\%.$

It follows that the values of the three binary power options are:

$n = 0 \rightarrow V = S^n e^{\gamma(n)\tau} N(d_2 + n\sigma\sqrt{\tau}) = 1 \times e^{-0.025} \times 0.509973 = \$0.497381,$

$n = 1 \rightarrow V = S^n e^{\gamma(n)\tau} N(d_2 + n\sigma\sqrt{\tau}) = 10 \times e^{-0.000} \times 0.589010 = \$5.890104,$

$n = 2 \rightarrow V = S^n e^{\gamma(n)\tau} N(d_2 + n\sigma\sqrt{\tau}) = 100 \times e^{+0.065} \times 0.664582 = \$70.921432.$

The call option is a combination of long *1* unit of the power binary option with $n = 1$ and short *10* units of the power binary option with $n = 0$ so the value of the call is $\$5.890104 - 10 \times \$0.497381 = \$0.916291$. The expected payoff on the call is $\$0.916291 \times e^{0.025} = \$0.939487$

Next we want to compute the expectation of the square of the payoff and from this the variance of the payoff. The expected squared payoff is the expected payoff on a portfolio comprised of:

$+X^2 = 100$ units of the power binary option with $n = 0$,
$-2X = -20$ units of the power binary option with $n = 1$,
$+1 = 1$ units of the power binary option with $n = 2$.

The value of this combination of binary power options is 49.74-117.80+70.92=2.8574845.

The expectation of the squared payoff is $\$2.8574845 \times e^{0.025} = \$2.929822$,

the variance of the payoff is $2.929822 - 0.939487^2 = 2.047186$,

the standard deviation of the call option payoff is $\sqrt{2.047186} = 1.430799$.

3.1. Computing the value of the call option via Monte-Carlo Simulation

Using Excel we can generate a random number U from the uniform distribution on the range *0* to *1* using the $rand(\ \ )$ function. This uniform random number can be used as the argument to the inverse of the cumulative density function of the standard normal distribution to obtain a random number Z from standard normal *N(0,1)* distribution. In Excel we use $normsinv(rand(\ \ ))$ to compute such a random number Z.

From Z we compute $Y = sZ + m$ and this has a normal distribution with mean m and standard deviation s. This Y has the same distribution as the logarithm of the price relative. From Y we compute $W = exp(Y) = exp(sZ + m)$. This variable W has the same Log-Normal distribution as the price relative. From W we compute $S = S_t W$ which is the product of the price relative and the initial stock price at time t. This S has the same Log-Normal distribution as the stock price at maturity. Next we compute the option payoff, which for a call option is $P = max(S - X, 0)$. P will be a random number from the distribution of the payoff.

The formulae to apply are:

$U = rand(\quad), Z = N^{-1}(U), \qquad Y = sZ + m, W = exp(Y), S = S_0W, P = max(S - X, 0).$

where U is a uniform random number.

If we do this set of calculations once, we generate a sample of size *1* from the distribution of the payoff from the option. If we repeat this calculation n times, we generate a sample of size n from the distribution of the payoff of the option. Suppose we do this and generate a sample $P_1, P_2, \dots, P_n$ of size n from the distribution of the payoff from the option. This is an *i.i.d.* sample. We can compute the sample mean of the sample of payoffs $\bar{P} = \frac{1}{n}(P_1 + P_2 + \cdots + P_n) = \frac{1}{n}(\sum_{i=1}^{n} P_i)$. This sample mean is an unbiased estimate of the expectation of the payoff.

Suppose that $E(P)$ is the expectation of the payoff P and $var(P)$ is the variance of the payoff P, then statistical theory tells us that $E(\bar{P}) = E(P)$ and $var(\bar{P}) = \frac{1}{n}var(P)$ and $sd(\bar{P}) = \frac{1}{\sqrt{n}}sd(P)$. This means that the expectation of the sample average (the "estimate") is the same as the expectation of the payoff and the standard deviation of the sample average is $\frac{1}{\sqrt{n}}$ times the standard deviation of the payoff. This means that if we increase the sample size n, we will not change the expectation of the sample average but the variance and the standard deviation of the sample average will get smaller.

Every time you repeat the statistical experiment of generating a sample of size n from the distribution of P, you will obtain a different sample average $\bar{P}$. The central limit theorem of statistics tells us that as the sample size n increases, the probability distribution of $\bar{P}$ converges to a normal distribution with $E(\bar{P}) = E(P)$ and $var(\bar{P}) = \frac{1}{n}var(P)$.

So by increasing the sample size we reduce the variance of the sample mean and hence increase the accuracy of the sample mean as an estimator of the expected payoff. For a sample of size *1* the Monte-Carlo simulation will produce an estimated payoff $\bar{P}$ with $E(\bar{P}) =$ \$0.939487 and $var(\bar{P}) = 2.047186$ and $sd(\bar{P}) = 1.430799$.

For a sample of size n the Monte-Carlo simulation will produce an estimated payoff $\bar{P}$ with $E(\bar{P}) =$ \$0.939487, $var(\bar{P}) = 2.047186/n$ and $sd(\bar{P}) = 1.430799/\sqrt{n}$. The coefficient of variation is $CV(\bar{P}) = sd(\bar{P})/E(\bar{P})$. We now look at the effect of an increase in the sample size on the mean, standard deviation and coefficient of variation of the estimated expected payoff.

n	$E(\bar{P})$	$sd(\bar{P})$	$CV(\bar{P})$
1	0.939487	1.430799	152.30%
10	0.939487	0.452458	48.16%
100	0.939487	0.143080	15.23%

1,000	0.939487	0.045246	4.82%
10,000	0.939487	0.014308	1.52%
100,000	0.939487	0.004525	0.48%
1,000,000	0.939487	0.001431	0.15%

Table 1: Convergence of MC simulated European Option prices

The standard deviation of the estimated payoff $sd\left(\overline{P}\right)$ decreases as the simulated sample size n increases. The ratio of the standard deviation to the mean is known as the *coefficient of variation* and it is a measure of relative variability. It provides us with a way of measuring the accuracy of our estimate. More precisely, it allows us to determine confidence intervals for the estimated price. To see this, the sample mean has a distribution that converges to the normal distribution as the sample size increases. Since the sample mean is a random variable, each time we use MC simulation we will get a different result. There is a *95%* probability that the true expected payoff is within two standard deviations of the estimated payoff. This means the accuracy of the calculation increases as the sample size increases. So with a sample of size *1000000* we can be *95%* confident that the estimated expected payoff given by the MC method is within *0.30%* of the true value predicted by the model.

The required sample size is often an issue of importance with the MC method. A large sample size makes for more accurate option prices but this comes at the expense of a longer computation time. For options that are actively traded, dealers may want to be able to obtain the prices quickly. They may need to have the option prices pre-calculated for a range of parameter values so they can get a quick estimate of the price for trading purposes. Often the more complicated types of option where MC simulation is required aren't actively traded. For instance many executive share options are highly complex and are valued via MC simulation on a computer using an overnight run of the program.

On a final note, our confidence in the accuracy of the estimates produced by the MC method assumes a correct specification of the model parameters. Assuming the input parameters are correctly specified, we can be confident of being within the stated tolerance of the theoretically correct answer predicted by the model for a given simulation sample size. Uncertainty about the parameter values assumed in this, or any, model will create an additional source of uncertainty in the calculation results. Parameter value uncertainty will also be present when using any other numerical method for computing the value of an option contract, even when an analytical valuation formula is available. Dealing with this type of uncertainty gives rise to its own set of issues, and is beyond the scope of this paper. A common approach employed by practitioners to deal with this additional uncertainty is to calculate a *range* of option prices for a plausible range of input parameters, to produce a range of plausible option values. Our main concern is to ensure we've selected a sample size large enough for our estimates to be tolerably close to the theoretically expected value for any specific set of input parameters.

3.2. Example Excel implementation

We perform a simulation to create a sample of size 25 from the distribution of the payoff.

	A	B	C	D	E	F	G
1	0.0050	m		$0.9395	expected payoff		
2	0.2000	s		$1.4308	sd of payoff		
3	$10.00	S(t)		0.28616	sd of sample ave		
4	$10.00	X		0.30459	c.v. of sample ave		
5				$0.9357	sample av from simulation		
6				$1.4626	sample sd from simulation		
7				$0.9126	estimated option price		
8				$0.9163	analytic option price		

Figure 1: simulation spreadsheet rows 1:8

We start up Excel, open a blank sheet and enter data into the cell range A1:A4 and text / labels into the cell range B1:B4 as indicated in Figure 1. In the cell range A9:G9 we enter text for our column headings for the various calculations in the simulation. We then fill the cell range A10:A34 with the numbers from 1 to 25. We fill the cell range B10:G10 with the formulae as indicated in the Figure 2. These formulae are Excel implementations of the calculations for the first trial of the simulation.

cell range	excel code	explanation
B10	=RAND()	This generates a uniform U(0,1) random number
C10	=NORMSINV(B10)	This generates a normal N(0,1) random number
D10	=C10*$A$2+$A$1	This generates a normal N(m,s) random number
E10	=EXP(D10)	This computes W=ex(Y) which is a lognormal LN(m,s) random number
F10	=E10*$A$3	This is our simulated stock price at maturity
G10	=MAX(F10-$A$4,0)	This is our simulated option payoff

Figure 2

After we have done this, we can copy the cell range B10:G10 to the cell range B11:G34. This will get Excel to compute the results for trials 2:25 of the simulation. The results for the first 10 trials are shown in Figure 3 below, for one run of the simulation with 25 trials.

Next we enter data or Excel code in cells D1:D8 to compute the following items:

The theoretical expected payoff on the call option
The theoretical standard deviation of the payoff
The theoretical standard deviation of the sample average
The coefficient of variation of the sample average
The sample average from the simulation
The sample standard deviation from simulation

The estimated option price obtained from the simulation

In the cells E1:E8 we enter relevant text / labels to indicate what's in the adjacent cells in column D. This is shown in Figure 1. For one run of the simulation with a sample size of 25 we obtained the results shown in Figures 1 and 3 below. The expected payoff was \$0.9163. The theoretical standard deviation of the payoff is 1.4308. The sample size is 25 so the theoretical standard deviation of the sample average is $1.4308/5 = 0.28616$ and the coefficient of variation of this sample average is $0.28616/0.9163 = 30.459\%$. The sample average from the simulation was \$0.9357, the standard deviation of the payoff from the simulation was \$1.4626 and the coefficient of variation of the sample average was $(1.4626/5)/0.9357 = 31.261\%$. The analytic formula gives a value of \$0.9163 for the call option and the MC simulation gives a value of \$0.9126, which is the sample average, discounted at 10% per annum for 0.25 years.

	A	B	C	D	E	F	G
9	trial	U	Z	Y	W	S	P
10	1	0.8094	0.8759	0.1802	1.1974	11.9742	\$1.97
11	2	0.7615	0.7113	0.1473	1.1587	11.5865	\$1.59
12	3	0.5793	0.2000	0.0450	1.0460	10.4603	\$0.46
13	4	0.9086	1.3319	0.2714	1.3118	13.1178	\$3.12
14	5	0.9621	1.7758	0.3602	1.4336	14.3355	\$4.34
15	6	0.6497	0.3846	0.0819	1.0854	10.8536	\$0.85
16	7	0.6683	0.4351	0.0920	1.0964	10.9639	\$0.96
17	8	0.4112	-0.2245	-0.0399	0.9609	9.6089	\$0.00
18	9	0.8571	1.0676	0.2185	1.2442	12.4423	\$2.44
19	10	0.4883	-0.0293	-0.0009	0.9991	9.9914	\$0.00

Figure 3: simulation spreadsheet rows 9:19

3.2.1.Sample size and accuracy

Running the simulation allows us to estimate both the sample average and the coefficient of variation of the sample average, an estimate of the accuracy of the sample average. It allows us to estimate how large a sample we need for a specified target level of accuracy in the result.

The coefficient of variation gives us an indication of the accuracy of the result produced by the simulation. The sample average from the simulation is an estimator of the true expected payoff. A 95%confidence interval for the expected payoff is the sample average plus or minus 2 times the coefficient of variation multiplied by the sample average. The simulation sample average is accurate to within twice the coefficient of variation with a probability of 95%.

For the simulation results shown above, the sample average and sample standard deviation from the simulation are close to the theoretically correct results. This was a chance result. We ran the simulation with 25 trials again another 3 times and obtained sample averages of 1.2914, 0.6349 and 1.3847 and sample standard deviations of 1.4379, 1.0438 and 1.5807 respectively. These compare to the theoretical results of 0.9163 for the average and 1.4308 for the standard deviation. The simulation will produce a different result every time you run it. To improve the accuracy of the result we need a larger sample size. An accuracy of plus or minus 62.52% with probability of 95% isn't that good. To improve it by a factor of 100 to plus or minus 0.6252% we would need to increase the sample size by a factor of 10000. This will make the simulation run more slowly.

4. Multivariate Statistics: Some relevant theory

Often we want to generate a set of n correlated random variables. These may be normally distributed. In the option pricing context we may be interested in computing the economic value of a multi-asset, multi-period option or financial contract. Using simulation we may want to generate a sample of log-price relatives of these multiple assets, and these log price relatives are normally distributed. There may be n different log price relatives and these are assumed to have a multivariate normal distribution with a particular vector of means and matrix of covariances. The parameters of the distribution are those of the so called *risk neutral distribution*. The prices of the various assets are obtained by exponentiating the log price relatives and multiplying the result by the initial asset price.

Usually we will know the parameters we want to use in the simulation, i.e. the mean vector and the covariance matrix. Using Excel it is straightforward to generate random variables from the $N(0,1)$ normal distribution. Accordingly it is easy to create a vector of dimension n which is i.i.d. and which has this standard normal distribution.

If we can generate a sample of n independent identically distributed random variables, with mean *zero* and variance *one* then we can put these into a vector of dimension n. This will give us a vector random variable with mean vector equal to the zero vector and covariance matrix equal to the n by n identity matrix.

Let $\boldsymbol{x}' = [x_1 \quad x_2 \quad \cdots \quad x_n]$ be a vector of n random variables and let $\mathbf{S_x} \in R^n \times R^n$ be the covariance matrix of this vector, so that $s_{ij} = cov(x_i, x_j)$.

Let $\boldsymbol{\mu}' = [\mu_1 \quad \mu_2 \quad \cdots \quad \mu_n]$ be the vector of means so that $\mu_i = E(x_i)$ and let $\boldsymbol{C} \in R^m \times R^n$ be some m by n matrix of constants

Then the covariance matrix of the m dimensional random vector $\boldsymbol{z} = \boldsymbol{C} \times \boldsymbol{x}$ is given by the matrix calculation $\boldsymbol{S_z} = \boldsymbol{C} \times \boldsymbol{S_x} \times \boldsymbol{C}'$,the mean is $E(\boldsymbol{z}) = \boldsymbol{C} \times \boldsymbol{\mu}$.

Let $[\ x_1 \quad x_2 \quad \dots \quad x_n\ ]$ be a random vector with *i.i.d.* components where $E(x_i) = \mu, Var(x_i) = 1$. If m = n then $\boldsymbol{S_x}$ is the identity matrix and $\boldsymbol{S_z} = \boldsymbol{CIC}' = \boldsymbol{CC}'$ which is the product of the matrix $\boldsymbol{C}$ and its transpose.

4.1. Simulating multi-variate normally distributed random samples with given correlation structure.

Consider an *n* dimensional *i.i.d.* standard normal vector $\boldsymbol{x}' = [x_1 \quad x_2 \quad \cdots \quad x_n]$ with mean vector $\boldsymbol{\mu}' = [0 \quad 0 \quad \ldots \quad 0]$, and covariance matrix $\boldsymbol{S_x} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}$.

Suppose we want to simulate samples of a normally distributed vector random variable $\boldsymbol{z}' = [z_1 \quad z_2 \quad \cdots \quad z_n]$ and we know in advance the covariance matrix structure for $\boldsymbol{z}$, given by the matrix $\boldsymbol{S_z}$ which take as an input to our calculation, along with the mean vector $E(\boldsymbol{z}') = \boldsymbol{\mu_z}'$.

If we can find an *n* by *n* matrix $\boldsymbol{C}$ such that $\boldsymbol{S_z} = \boldsymbol{C} \times \boldsymbol{C}'$ then the matrix calculation $\boldsymbol{z}' = \boldsymbol{x}' \times \boldsymbol{C}' + \boldsymbol{\mu_z}'$ gives us a simulated vector of dimension *n* from the multivariate normal distribution with mean $\boldsymbol{\mu_z}'$ and covariance matrix $\boldsymbol{S_z}$.

Theorem: Given a positive definite covariance matrix $\boldsymbol{S}$, there is a unique n by n matrix $\boldsymbol{C}$ such that $\boldsymbol{C}$ is lower triangular: $c_{ij} = 0 \; for \; j > i$ and that $\boldsymbol{S} = \boldsymbol{C} \times \boldsymbol{C}'$. The matrix $\boldsymbol{C}$ is called the Cholesky Square Root of the covariance matrix.

We present a proof of this theorem in Appendix C. The proof is constructive, so the implementation of the proof provides an *algorithm* or *procedure* for calculating the Cholesky Square Root matrix in Excel, or in any other software package that is capable of the required matrix calculations. The algorithm also provides a diagnostic check on whether the proposed covariance matrix is positive definite. As the proof is by induction, the procedure is iterative. It also provides a diagnostic check on whether or not the proposed covariance matrix is positive definite, and therefore a valid covariance matrix.

4.2. Excel implementation of the Cholesky Square Root calculation

The theorem and proof in Appendix B provides us with a matrix based recursive algorithm for computing the Cholesky square root. Let $\mathbf{S_n}$ be a n by n positive definite covariance matrix. We partition this matrix is $\mathbf{S_n} = \begin{bmatrix} \mathbf{S_{n-1}} & \mathbf{Y_{n-1}} \\ \mathbf{Y_{n-1}}' & s_{nn} \end{bmatrix}$ where $\mathrm{S_{n-1}}$ is a *n-1* by *n-1* matrix obtained from $\mathbf{S_n}$ by deleting row n and column n, $\mathbf{Y_{n-1}}$ is a *n-1* by *1* matrix obtained from $\mathbf{S_n}$ by taking the first *n-1* entries from column n of $\mathbf{S_n}$ and s_{nn} is the entry in row *n*, column n of $\mathbf{S_n}$.

Let $\mathbf{C_n}$ be the Cholesky square root of $\mathbf{S_n}$. We partition $\mathbf{C_n}$ as $\mathbf{C_n} = \begin{bmatrix} \mathbf{C_{n-1}} & \mathbf{0_{n-1}} \\ \mathbf{X_{n-1}}' & c_{nn} \end{bmatrix}$ where $\mathbf{C_{n-1}}$ is the Cholesky square root of $\mathbf{S_{n-1}}$, $\mathbf{0_{n-1}}$ is an *n-1* by *1* matrix of zeros, $\mathbf{X_{n-1}}'$ is a *1* by *n-1* matrix containing the first *n-1* entries in row n of the matrix $\mathbf{C_n}$ and c_{nn} is the entry in bottom right hand corner of $\mathbf{C_n}$.

Then we compute $\mathbf{C_n} = \begin{bmatrix} \mathbf{C_{n-1}} & \mathbf{0_{n-1}} \\ \mathbf{X_{n-1}}' & c_{nn} \end{bmatrix}$ via $\mathbf{X_{n-1}}' = \mathbf{Y_{n-1}}' \times (\mathbf{C_{n-1}}')^{-1}$, and $c_{nn} = \sqrt{[\mathbf{s}_{nn} - (\mathbf{X_{n-1}}' \times \mathbf{X_{n-1}})]}$ respectively, in that order. We first compute $\mathbf{C_2}$ as shown below, then from that we compute $\mathbf{C_3}$, then $\mathbf{C_4}$ etc.

These calculations can be done using Excel's matrix functions. We have made a slight change in notation from that used in the theorem in the appendix. We make the following substitutions: $\mathbf{S_{11}} = \mathbf{S_{n-1}}; \mathbf{S_{12}} = \mathbf{Y_{n-1}}; \mathbf{C_{11}} = \mathbf{C_{n-1}}; \mathbf{C_{12}} = \mathbf{0_{n-1}}; \mathbf{C_{21}} = \mathbf{X_{n-1}}'$ in the theorem to obtain the recursive relationship.

Example: we shall use the matrix $\mathbf{S_4} = \begin{bmatrix} 16 & 8 & 12 & -4 \\ 8 & 5 & 11 & -4 \\ 12 & 11 & 70 & -31 \\ -4 & -4 & -31 & 63 \end{bmatrix}$ to illustrate the algorithm and its excel implementation by applying it to the problem of computing the Cholesky square root of a 4 by 4 covariance matrix. The matrix $\mathbf{S_3}$ is $\mathbf{S_3} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} = \begin{bmatrix} 16 & 8 & 12 \\ 8 & 5 & 11 \\ 12 & 11 & 70 \end{bmatrix}$ and the matrix $\mathbf{S_2}$ is $\mathbf{S_2} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} = \begin{bmatrix} 16 & 8 \\ 8 & 5 \end{bmatrix}$. The matrix $\mathbf{S_1} = [16]$. We compute $\mathbf{C_2}$ from $\mathbf{S_2}$ first, then we compute $\mathbf{C_3}$ from $\mathbf{S_3}$

Suppose the matrix $\mathbf{S_4}$ is in the cell range A2:D5 and we use the cell range F2:I5 to compute the Cholesky square root $\mathbf{C_4}$, as shown in figure 4.

	A	B	C	D	E	F	G	H	I
1	**S** = covariance matrix					**C** = cholesky square root of S			
2	16	8	12	-4		4	0	0	0
3	8	5	11	-4		2	1	0	0
4	12	11	70	-31		3	5	6	0
5	-4	-4	-31	63		-1	-2	-3	7

Figure 4: Excel Implementation of Cholesky Square Root Calculation

The calculations for the Cholesky square root of a 1 by 1 covariance matrix are trivial, for our example we have $\mathbf{C_1} = [\sqrt{16}] = [4]$.

Consider the 2 by 2 covariance matrix $\mathbf{S_2} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$. The Cholesky square root of $\mathbf{S_2}$ is $\mathbf{C}_2 = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$ where $c_{11} = \sqrt{s_{11}}$; $c_{21} = s_{12}/\sqrt{s_{11}}$; $c_{22} = \sqrt{s_{22} - c_{21}{}^2}$. The formula for $\mathbf{C}_2$ in terms of $\mathbf{S_2}$ is $\mathbf{C_2} = \begin{bmatrix} \sqrt{s_{11}} & 0 \\ s_{12}/\sqrt{s_{11}} & \sqrt{s_{22} - s_{12}{}^2/s_{11}} \end{bmatrix}$. Direct calculation confirms that $\mathbf{C}_2\mathbf{C}_2{}' = \mathbf{S}_2$. These results can also be obtained by applying the equations set out in Appendix C of Kwan (2011) to the two dimensional case, which outlines the implementation of a different procedure for doing this. Note that if $S_{22} - C_{21}{}^2 \leq 0$ then this means the proposed covariance matrix used as the input for the calculation was not positive definite. We obtain $\mathbf{C_2} = \begin{bmatrix} 4 & 0 \\ 2 & 1 \end{bmatrix}$ and we note that $\begin{bmatrix} 4 & 0 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 16 & 8 \\ 8 & 5 \end{bmatrix}$ for our example. The matrix $\mathbf{C_2}$ is in the cell range F2:G3.

We put the relevant excel calculations into that cell range as follows: F2**=A2^0.5**; G2**=0**; F3**=A3/F2**; G3**=(B3-F3^2)^0.5**

The matrix $\boldsymbol{S_3}$ obtained from $\boldsymbol{S_4}$ by deleting the 4th row and 4th column **can be written** as a *partitioned matrix* in the form $\mathbf{S}_3 = \begin{bmatrix} \mathbf{S_2} & \mathbf{Y_2} \\ \mathbf{Y_2}' & s_{33} \end{bmatrix}$. This matrix resides in the cell range A2:C4. This is comprised of the 2 by 2 submatrix $\boldsymbol{S_2}$ in the top left corner, the 1 by 2 matrix $\boldsymbol{Y_2}$ in the top right hand corner, the 2 by 1 matrix $\boldsymbol{Y_2}' = [12 \quad 11]$ in the bottom left corner (cell range A4:B4) and the 1 by 1 matrix containing the single number $s_{33} = 70$ (cell C4) in the bottom right hand corner.

Now consider the 3 by 3 matrix $\boldsymbol{C_3} = \begin{bmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C_2} & \mathbf{0} \\ \boldsymbol{X_2} & c_{33} \end{bmatrix}$ which is a lower triangular matrix, partitioned as for $\mathbf{S_3}$. This $\boldsymbol{C_3}$ resides in the cell range F2:H4 and $\boldsymbol{C_2}$ is as above (and in cell range F2:G3) , $\mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is a 2 by 1 matrix of zeros (in cell range H2:H3), $\boldsymbol{X_2}' = [c_{31} \quad c_{32}]$ is a 1 by 2 matrix containing the first 2 entries of row 3 of $\boldsymbol{C_3}$ (in cell range F4:G4)**,** and a 1 by 1 matrix containing the entry c_{33} in the bottom right hand corner (cell H4).

The non-diagonal entries in the last row of $\boldsymbol{C_3}$ are in $\mathbf{X_2}' = \mathbf{Y_2}' \times (\mathbf{C_2}')^{-\mathbf{1}}$. For our data this is $\mathbf{X_2}' = [\mathbf{12} \quad \mathbf{11}] \times \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix}^{-\mathbf{1}} = [\mathbf{12} \quad \mathbf{11}] \times \begin{bmatrix} +0.25 & -0.5 \\ 0 & 1 \end{bmatrix}^{-\mathbf{1}} = [\mathbf{3} \quad \mathbf{5}]$. In excel we compute this by selecting the cell range F4:G4 and entering the formula **=MMULT(A4:B4,MINVERSE(TRANSPOSE(F2:G3)))** and then pressing enter while holding down the control and shift keys. This is how you enter matrix formulae into excel. The diagonal entry in row 3 of $\boldsymbol{C_3}$ is $c_{nn} = \sqrt{[s_{nn} - (\mathbf{X_{n-1}}'\mathbf{X_{n-1}})]}$ **.**Using our data we get $\sqrt{\left[70 - \left([3 \quad 5] \times \begin{bmatrix} 3 \\ 5 \end{bmatrix}\right)\right]} = \sqrt{70 - 34} = 6$. In excel we compute this by entering the formula **=(C4-MMULT(F4:G4,TRANSPOSE(F4:G4)))^0.5** into cell H4. Hence we obtain $\boldsymbol{C_3} = \begin{bmatrix} 4 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 5 & 6 \end{bmatrix}$ as the square root of $\mathbf{S_3} = \begin{bmatrix} 16 & 8 & 12 \\ 8 & 5 & 11 \\ 12 & 11 & 70 \end{bmatrix}$

Next we want to compute $\boldsymbol{C_4}$ using $\boldsymbol{C_3}$ and $\mathbf{S_4}$**.** To implement $\mathbf{X_3}' = \mathbf{Y_3}' \times (\mathbf{C_3}')^{-\mathbf{1}}$ in excel we choose the cell range F5:H5 and enter the matrix formula

=MMULT(A5:C5,MINVERSE(TRANSPOSE(F2:H4)))

$$\mathbf{Y_3}' = [-4 \quad -4 \quad -31] \text{ and } (\boldsymbol{C_3}')^{-\mathbf{1}} = \begin{bmatrix} 4 & 2 & 3 \\ 0 & 1 & 5 \\ 0 & 0 & 6 \end{bmatrix}^{-1} = \begin{bmatrix} 0.25 & -0.5 & +0.291667 \\ 0 & 1 & -0.833333 \\ 0 & 0 & +0.166667 \end{bmatrix}$$

$$\therefore \mathbf{X_3}' = [-4 \quad -4 \quad -31] \times \begin{bmatrix} 0.25 & -0.5 & +0.291667 \\ 0 & 1 & -0.833333 \\ 0 & 0 & +0.166667 \end{bmatrix} = [-1 \quad -2 \quad -3]$$

The diagonal entry in row 4 of $\boldsymbol{C_4}$ is $c_{44} = \sqrt{[s_{44} - (\mathbf{X_3}'\mathbf{X_3})]}$ **.** In excel we enter the matrix formula **=(D5-MMULT(F5:H5,TRANSPOSE(F5:H5)))^0.5** into cell I5 to implement this. The result is $c_{44} = \sqrt{\left[63 - \left([-1 \quad -2 \quad -3] \times \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix}\right)\right]} = \sqrt{63 - 14} = 7$

It follows that $\boldsymbol{C_4} = \begin{bmatrix} +4 & +0 & +0 & +0 \\ +2 & +1 & +0 & +0 \\ +3 & +5 & +6 & +0 \\ -1 & -2 & -3 & +7 \end{bmatrix}$ is the Cholesky square root of the covariance matrix $\boldsymbol{S_4}$.

4.2.1. Diagnostic test for positive definite input covariance matrix $\boldsymbol{S_n}$

Finally, the implementation of the recursive calculation provides us with a diagnostic check-step in determining whether we have a valid (i.e. positive definite) covariance matrix. The matrix equation determining the diagonal entry in the last row of the Cholesky square root matrix is $c_{nn} = \sqrt{s_{nn} - (\boldsymbol{X_{n-1}}'\boldsymbol{X_{n-1}})}$. The input covariance matrix is positive definite if and only if $s_{nn} - (\boldsymbol{X_{n-1}}'\boldsymbol{X_{n-1}}) > 0$. If the proposed covariance matrix isn't positive definite then it won't have a Cholesky square root. The following example illustrates this.

The matrix $\mathbf{S_4} = \begin{bmatrix} 1 & 0 & 0.7 & 0.4 \\ 0 & 1 & 0.4 & 0.7 \\ 0.7 & 0.4 & 1 & 0 \\ 0.4 & 0.7 & 0 & 1 \end{bmatrix}$ is not positive definite.

We can write $\mathbf{S_4}$ as a partitioned matrix $\mathbf{S_4} = \begin{bmatrix} \boldsymbol{S_3} & \boldsymbol{Y_3} \\ \boldsymbol{Y_3}' & s_{44} \end{bmatrix}$ where $\boldsymbol{S_3} = \begin{bmatrix} 1 & 0 & 0.7 \\ 0 & 1 & 0.4 \\ 0.7 & 0.4 & 1 \end{bmatrix}$, $\boldsymbol{Y_3}' = [0.4 \quad 0.7 \quad 0]$, $s_{44} = 1$.

The matrix $\mathbf{S_3}$ is positive definite. Its Cholesky square root is $\boldsymbol{C_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.7 & 0.4 & 0.591608 \end{bmatrix}$

We try to compute $\boldsymbol{C_4} = \begin{bmatrix} \boldsymbol{C_3} & \boldsymbol{0_3} \\ \mathbf{X_3}' & c_{44} \end{bmatrix}$ using $\boldsymbol{X_3}' = \boldsymbol{Y_3}'(\boldsymbol{C_3}')^{-1}$ and $c_{44} = \sqrt{s_{44} - (\boldsymbol{X_3}'\boldsymbol{X_3})}$

We obtain $\mathbf{X_3}' = \mathbf{Y_3}'(\mathbf{C_3}')^{-1} = [0.4 \quad 0.7 \quad -0.94657]$ and $s_{44} - (\boldsymbol{X_3}'\boldsymbol{X_3}) = -0.546$. This last result is negative, so we can't compute the Cholesky square root. This indicates that the matrix $\mathbf{S_4}$ was not positive definite. The diagnostic check is to compute $s_{nn} - (\boldsymbol{X_{n-1}}'\boldsymbol{X_{n-1}})$ and check it the result is negative. If so then the input matrix $\mathbf{S_n}$ was not positive definite.

For this matrix we have a negative eigenvalue of -0.10 and an associated eigenvector of $\boldsymbol{y}' = [+1 \quad +1 \quad -1 \quad -1]$. The calculation $\boldsymbol{y}'\mathbf{S_4}\boldsymbol{y}$ gives the result $\boldsymbol{y}'\mathbf{S_4}\boldsymbol{y} = -0.40$. The implications of the proposed covariance matrix not being positive definite are discussed in Kwan (2010) in the context of portfolio optimisation. In the context of this paper, the implication is that the methodology is invalid and the option prices computed by the method will be incorrect.

4.3. Statistically modelling the asset prices in terms of the standard normal distribution

The following equations set up our statistical model for the MC simulation. The sequencing of formulae to apply are:

Compute $\boldsymbol{u}' = [u_1 \quad u_2 \quad \cdots \quad u_n]$ a vector of n uniform U(0,1) random variables using $u_i = rand()$
Compute $\boldsymbol{z}' = [z_1 \quad z_2 \quad \cdots \quad z_n]$ a vector of n N(0,1) distributed random variables using $Z(i) = N^{-1}(U(i))$
Compute $\boldsymbol{y}' = [y_1 \quad y_2 \quad \cdots \quad y_n] = \boldsymbol{m}' + \boldsymbol{z}'\boldsymbol{C}'$ where $\boldsymbol{C} = \sqrt{\boldsymbol{S}}$ is the Cholesky square root of $\boldsymbol{S}$. This is the vector of log price relatives.
Compute $\boldsymbol{w}' = [w_1 \quad w_2 \quad \cdots \quad w_n] = [exp(y_1) \quad exp(y_2) \quad \ldots \quad exp(y_n)]$. This is the vector of log price relatives calculated by component-wise exponentiation.
Compute the vector $\boldsymbol{s}' = [s_1 \quad s_2 \quad \cdots \quad s_n]$. This is the vector of asset prices.

We have $s_i = S(i)_t \times w_i$ is the i^{th} entry in that vector and it represents the value of the i^{th} asset at time T_i. The term $S(i)_t$ here means the value of the i^{th} asset at time t, and we assume this is given / observable. The term $S(i)_T$ means the value of that asset at some future time T and this is to be modeled.

Compute the payoff function which is a function of the vector $\boldsymbol{s}'$of such future asset values.

These calculations create the simulated payoff for the first trial of our simulation. We can write the equations for the vector of future asset prices in terms of the vector of initial asset prices in vector form as $\boldsymbol{S_T'} = \boldsymbol{s_t'}exp(\boldsymbol{m}' + \boldsymbol{z}'\boldsymbol{C}')$. We would want to repeat the above calculations M times (for large M) so we can obtain an estimate of the expected payoff with low relative variability.

5. Case Study: Excel Implementation of a Multivariate Simulation

With multi asset, multi period financial contracts, the payoff at maturity will be a complicated function of the values of several different assets as observed at possibly several different times. Such features naturally arise in various option pricing scenarios. For example an *exchange* option allows the holder to exchange one risky asset for another. Clearly, the option payoff in this case depends on the value of the two different assets at the maturity date. Spread options provide a payoff equal to the greater of D – X or zero where D is the difference between two asset prices and X is some fixed value. Rainbow Options are call or put options over a portfolio of assets.

Below we present several examples of multi asset financial contracts. Multi-asset features also arise naturally in the context of executive compensation. The main example motivating our discussion is that of an *executive share option* (ESO) which contains various performance conditions to be satisfied for the vesting and exercising of the option. These conditions are typically encountered in ESO structures. Consider the following structure: The executive must be in the continuous service of the employer ABC for two years and meet several performance hurdles on the return on the company's stock price at the two year mark. Namely, the stock return over the two year period must have:

- increased by 20%,
- exceeded that of the market return over the same period by 10%,
- exceeded the stock price return on rival company XYZ stock by 15%.

If all three performance conditions are met then the executive is granted an option on the ABC stock that matures three years later with an exercise price of *150%* of the ABC share price value at the time the executive commenced employment. These conditions collectively are known as the *hurdle*. If these conditions aren't met then the option payoff at time $T = 5$ years is cancelled. If the hurdle conditions are met then the ESO grants a call option payoff at time $T = 5$ years on ABC with an exercise price equal to 150% of the ABC share price at time $t = 0$. This is a typical structure for an executive share option.

The ESO structure therefore defines an option contract with a payoff which depends on three different risky assets (ABC corporation shares, XYZ corporation shares and the market index) measured at three times, two of which are future times: $t = 0$, $U = 2$ and $T = 5$ years. The final payoff is made at time $T = 5$ and will correspond to that of a call option on ABC shares, with an exercise price equal to 150% of the ABC share price at time $t = 0$. To price the option we begin with noting that we have three sources of uncertainty, the two stock prices and the market index which we denote:

$$S(1) = ABC\ stock, \qquad S(2) = stock\ market\ index, \qquad \mathrm{S}(3) = XYZ\ stock$$

These asset prices are observed at time $t = 0$, $U = 2$ and $T = 5$ years. Translating the performance hurdles using this notation, the payoff of the ESO at its maturity date may be written as

$$P = I_1 I_2 I_3 \times max(S(1)_5 - 1.5S(1)_0, 0)$$

where

$$I_1 = I\left\{\frac{S(1)_2}{S(1)_0} > 1.2\right\}, I_2 = I\left\{\frac{S(1)_2/S(1)_0}{S(2)_2/S(2)_0} > 1.1\right\}, I_3 = I\left\{\frac{S(1)_2/S(1)_0}{S(3)_2/S(3)_0} > 1.15\right\}.$$

This is a product of three indicator functions and a call option payoff. The payoff is made at time $T = 5$ but it also depends on market conditions at time $U = 2$ and we want to compute the value of the contract as at time $t = 0$. The three conditions in the indicator functions collectively define the *hurdle* at time $U = 2$. Each of the assets defining the payoff has its own Log-Normal distribution. The final payoff is a function of four different Log-Normal random variables and obtaining an analytic formula for it is challenging. However it is possible to obtain an analytic solution involving the four dimensional cumulative normal distribution, but we omit the details as they are beyond the scope of this paper.

Another example of a multi-asset multi-period option we will examine is the three asset rainbow call option with maturity at time *3* and with payoff $P = max(S(1)_3 + S(2)_3 + S(3)_3 - X, 0)$. It is a call option with exercise price X, however the underlying asset is a portfolio of three assets consisting of their sum. The underlying asset will not have a Log-Normal distribution so it is not possible to derive a Black-Scholes type valuation formula for this payoff.

The underlying asset will not have a Log-Normal distribution so it is not possible to derive a Black-Scholes type valuation formula for this payoff. The spread option with maturity at time *3* and with payoff $P = max(S(1)_3 - S(2)_3 - X, 0)$ is a third example of a multi-asset multi-period option we consider. This is a call option with exercise price X however this time the underlying asset is a portfolio consisting of a long position in the first asset and a short position in the second asset.

The valuation of all three contracts involves calculating the expected payoff at maturity under the risk neutral distribution, which we estimate using the MC simulation approach. The following parameters are assumed, and these define the means, variances and covariances of returns on these assets. We are interested in four different returns but these are defined relative to three different assets. For one of these assets we have returns over two different times. The risk free rate of interest is $r = 5.00\%$.

Stock	$S(1)$	$S(2)$	$S(3)$
	ABC	XYZ	Index
Initial price	\$1.00	\$1.00	\$500.00
Dividend yield	$y_1 = 2.00\%$	$y_2 = 3.00\%$	$y_3 = 4.00\%$
Volatility	$\sigma_1 = 40.00\%$	$\sigma_2 = 30.00\%$	$\sigma_3 = 20.00\%$

Correlations:

stock	1	2	3
1	$\rho_{11} = 1.00$	$\rho_{12} = 0.70$	$\rho_{13} = 0.50$
2	$\rho_{21} = 0.70$	$\rho_{22} = 1.00$	$\rho_{23} = 0.40$
3	$\rho_{31} = 0.50$	$\rho_{32} = 0.40$	$\rho_{33} = 1.00$

We have two different intervals: $t = 0, U = 2, T = 5$: $(t, U) = (0,2)$ and $(t, T) = (0,5)$. The intersection of these is (2,5). We have four different log-price relatives which define the ESO contract payoff. These are

$y_1 = ln(S(1)_U / S(1)_t) = ln(S(1)_2 / S(1)_0)$,

this is the return on ABC over the first 2 years

$y_2 = ln(S(2)_U/S(2)_t) = ln(S(2)_2/S(2)_0)$,

this is the return on XYZ over the first 2 years

$y_3 = ln(S(3)_U/S(3)_t) = ln(S(3)_2/S(3)_0)$,

this is the return on the index over the first 2 years

$y_4 = ln(S(1)_T/S(1)_t) = ln(S(1)_5/S(1)_0)$,

this is the return on ABC over the first 5 years.

The payoff on the ESO contract is a function of these variables. The payoff is

$$P = I_1 I_2 I_3 S(1)_0 max(exp(y_4) - 1.50,0)$$

where

$$I_1 = I\{exp(Y_1) > 1.2\}; I_2 = I\{exp(Y_1 - Y_2) > 1.1\};\ I_3 = I\{exp(Y_1 - Y_3) > 1.15\}$$

The product $I\{exp(y_1) > 1.2\}I\{exp(y_1 - y_2) > 1.1\}I\{exp(y_1 - y_3) > 1.15\}$ is the hurdle and the term $S(1)_0 max(exp(y_1) - 1.50,0)$ is the call option payoff.

We need to obtain the covariance matrix of the vector of four log price relatives as part of the input to the MC simulation calculation. This is the covariance matrix

$$\underline{S}_4 = \begin{bmatrix} 0.320 & 0.168 & 0.080 & 0.320 \\ 0.168 & 0.180 & 0.048 & 0.168 \\ 0.080 & 0.048 & 0.080 & 0.080 \\ 0.320 & 0.168 & 0.080 & 0.800 \end{bmatrix}$$

These covariances are calculated as follows:

$$cov(y_1, y_1) = 1.00 \times 0.40 \times 0.40 \times 2 = 0.320,$$
$$cov(y_1, y_2) = 0.70 \times 0.40 \times 0.30 \times 2 = 0.168,$$
$$cov(y_1, y_3) = 0.50 \times 0.40 \times 0.20 \times 2 = 0.080,$$
$$cov(y_1, y_4) = 1.00 \times 0.40 \times 0.40 \times 2 = 0.320,$$
$$cov(y_2, y_2) = 1.00 \times 0.30 \times 0.30 \times 2 = 0.180,$$
$$cov(y_2, y_3) = 0.40 \times 0.30 \times 0.20 \times 2 = 0.048,$$
$$cov(y_2, y_4) = 0.70 \times 0.30 \times 0.40 \times 2 = 0.168,$$
$$cov(y_3, y_3) = 1.00 \times 0.20 \times 0.20 \times 2 = 0.080,$$
$$cov(y_3, y_4) = 0.50 \times 0.20 \times 0.40 \times 2 = 0.080,$$
$$cov(y_4, y_4) = 1.00 \times 0.40 \times 0.40 \times 5 = 0.800.$$

The Cholesky square root of this matrix was computed using the Excel code developed above. The result is the matrix

$$\underline{C}_4 = \begin{bmatrix} 0.565685 & 0.000000 & 0.000000 & 0.000000 \\ 0.296985 & 0.302985 & 0.000000 & 0.000000 \\ 0.141421 & 0.019803 & 0.244147 & 0.000000 \\ 0.565685 & 0.000000 & 0.000000 & 0.692820 \end{bmatrix}.$$

The mean of the vector of log price relatives is

$$\boldsymbol{\mu}' = E(\boldsymbol{y}') = [-0.10 \quad -0.05 \quad -0.02 \quad -0.25].$$

The components are calculated as follows:

$$\mu_1 = \left(r - y_1 - \frac{1}{2}\sigma_1{}^2\right) \times 2 = \left(0.05 - 0.02 - \frac{1}{2}0.4^2\right) \times 2 = -0.10,$$

$$\mu_2 = \left(r - y_2 - \frac{1}{2}\sigma_2{}^2\right) \times 2 = \left(0.05 - 0.03 - \frac{1}{2}0.3^2\right) \times 2 = -0.05,$$

$$\mu_3 = \left(r - y_3 - \frac{1}{2}\sigma_3{}^2\right) \times 2 = \left(0.05 - 0.04 - \frac{1}{2}0.2^2\right) \times 2 = -0.02,$$

$$\mu_4 = \left(r - y_1 - \frac{1}{2}\sigma_1{}^2\right) \times 5 = \left(0.05 - 0.02 - \frac{1}{2}0.4^2\right) \times 5 = -0.25.$$

We are now ready to demonstrate the calculations for the first iteration of the simulation. We generate a sample of four *i.i.d N(0,1)* random variables and put them into a row vector. Suppose we do this and obtain the following random 4-d normal vector: $\boldsymbol{z}' = [0.588339 \quad -0.630870 \quad 0.042154 \quad 2.539468]$.

We perform the matrix calculation

$$\boldsymbol{y} = \boldsymbol{C} \times \boldsymbol{z} + \boldsymbol{\mu}$$

$$= \begin{bmatrix} 0.565685 & 0.000000 & 0.000000 & 0.000000 \\ 0.296985 & 0.302985 & 0.000000 & 0.000000 \\ 0.141421 & 0.019803 & 0.244147 & 0.000000 \\ 0.565685 & 0.000000 & 0.000000 & 0.692820 \end{bmatrix} \times \begin{bmatrix} 0.588339 \\ -0.630870 \\ 0.042154 \\ 2.539468 \end{bmatrix} + \begin{bmatrix} -0.10 \\ -0.05 \\ -0.02 \\ -0.25 \end{bmatrix}.$$

The result is $\boldsymbol{y} = \begin{bmatrix} +0.232815 \\ -0.066417 \\ +0.061002 \\ +1.842210 \end{bmatrix}$ which is the vector of log price relatives.

We exponentiate this component wise to obtain the vector of price relatives

$$\boldsymbol{w} = exp(\boldsymbol{y}) = \begin{bmatrix} \exp(+0.232815) \\ \exp(-0.066417) \\ \exp\,(+0.061002) \\ \exp(+1.842210) \end{bmatrix} = \begin{bmatrix} 1.262148 \\ 0.935741 \\ 1.062901 \\ 6.310469 \end{bmatrix}.$$

From this we can compute the payoff at time T = 5. The three conditions defining the hurdle are all satisfied so the payoff P is calculated as follows:

$$I_1 = I\{1.262148 > 1.2\} = 1;\ I_2 = I\left\{\frac{1.262148}{0.935741} > 1.1\right\} = 1; I_3 = I\left\{\frac{1.262148}{1.062901} > 1.15\right\} = 1$$

$$P = I_1 \times I_2 \times I_3 \times \max(6.310469 - 1.50,0) \times 1.00 \ = \ 4.810469.$$

This completes the calculations for one iteration of the MC simulation calculation of the ESO contract payoff. We need to repeat these calculations a large number of times, and obtain the sample average payoff. This is an estimate of the expected payoff on the ESO.

We can also compute the payoffs on the other contracts: The payoff on a plain vanilla 5 year call option on Stock 1 with the same exercise price as the ESO would also be 4.810469. The ESO and the call option will have the same payoff provided the 3 conditions in the hurdle at time 2 years are all met, otherwise the ESO payoff is zero. The payoff on a spread option on the difference between the return on asset 1 and the return on asset 2 with an exercise price of 0.10 would be max(1.262148 - 0.935741-0.10,0) = 0.226407. The payoff on a 3 asset rainbow option over an equally weighted portfolio of assets 1, 2 and 3 with an exercise price of 1.1 would be max(1.08693-1.10,0) = 0. Each of these other payoffs is some function of the four price relatives.

5.1. The Excel implementation

The relevant inputs to the calculations are set out as shown in Figure 5 below. These apply to each of the examples considered.

	A	B	C	D	E	F	G
1		financial assumptions					
2							
3	5%	risk free rate					
4							
5		S(A)	S(B)	S(C)	asset		
6		$1.00	$1.00	$500.00	initial value		
7		2%	3%	4%	dividend yield		
8		40%	30%	20%	volatility		
9							
10		correlation matrix: correlation of returns over overlapping intervals					
11		A	B	C			
12	A	1	70%	50%			
13	B	70%	1	40%			
14	C	50%	40%	1			
15							
16		standard deviation of log price relatives					mean vector
17		price relatives		TERM	VOL	σ√τ	
18	1	S(A)(2)/S(A)(0)		2	40%	0.565685	-0.10
19	2	S(B)(2)/S(B)(0)		2	30%	0.424264	-0.05
20	3	S(C)(2)/S(C)(0)		2	20%	0.282843	-0.02
21	4	S(A)(5)/S(A)(0)		5	40%	0.894427	-0.25

Figure 5: the financial assumptions for the case study

	A	B	C	D	E	F
22						
23		mean vector of log price relatives				
24		-0.10	-0.05	-0.02	-0.25	
25						
26		covariance matrix of log price relatives				
27		0.320	0.168	0.080	0.320	
28		0.168	0.180	0.048	0.168	
29		0.080	0.048	0.080	0.080	
30		0.320	0.168	0.080	0.800	
31						
32		cholesky square root of covariance matrix				
33		0.565685	0.000000	0.000000	0.000000	
34		0.296985	0.302985	0.000000	0.000000	
35		0.141421	0.019803	0.244147	0.000000	
36		0.565685	0.000000	0.000000	0.692820	
37						

Figure 6: mean vector, covariance and Cholesky matrix

The cell range B24:E24 is the input data for the mean vector. The cell range B27:E30 is the covariance matrix. The cell range B33:E36 is the Cholesky square root of the covariance matrix. These are shown in figure 6. We set up the calculations for the first trial of the simulation. This is shown in Figures 7 and 8.

	A	B	C	D	E	F	G	H	I
39		Z VECTOR				Y VECTOR			
40	TRIAL	Z(1)	Z(2)	Z(3)	Z(4)	Y(1)	Y(2)	Y(3)	Y(4)
41	1	1.0542	-0.0230	-2.1998	-0.2904	0.4963	0.2561	-0.4085	0.1452
42	2	0.8246	-0.5089	-0.2099	0.3694	0.3665	0.0407	0.0353	0.4724
43	3	-1.0879	-0.5846	-0.0951	-1.5781	-0.7154	-0.5502	-0.2086	-1.9587
44	4	-1.6260	-0.0595	-0.4399	1.5327	-1.0198	-0.5509	-0.3585	-0.1080
45	5	0.7783	-0.5592	-0.0148	2.1281	0.3403	0.0117	0.0754	1.6647
46	6	0.7863	-0.2913	0.1506	0.8822	0.3448	0.0953	0.1222	0.8060
47	7	0.1487	-0.4428	1.1199	0.7883	-0.0159	-0.1400	0.2657	0.3803
48	8	0.6032	-0.7070	0.3481	-1.5082	0.2412	-0.0851	0.1363	-0.9537
49	9	1.5060	-1.7208	1.0321	0.4849	0.7519	-0.1241	0.4109	0.9378
50	10	0.0174	-0.8188	-0.8094	-0.2473	-0.0902	-0.2929	-0.2314	-0.4115

Figure 7: results for the Z and Y vectors for the first 10 trials

A	J	K	L	M	N	O	P	Q
39	W VECTOR				ESO	Call	Spread	Rainbow
40 TRIAL	W(1)	W(2)	W(3)	W(4)	Payoff	Payoff	Payoff	Payoff
41 1	1.6427	1.2919	0.6647	1.1562	0.0000	0.0000	0.2508	0.0998
42 2	1.4426	1.0415	1.0359	1.6038	0.1038	0.1038	0.3011	0.0734
43 3	0.4890	0.5768	0.8117	0.1410	0.0000	0.0000	0.0000	0.0000
44 4	0.3607	0.5764	0.6987	0.8977	0.0000	0.0000	0.0000	0.0000
45 5	1.4053	1.0118	1.0783	5.2840	3.7840	3.7840	0.2936	0.0651
46 6	1.4117	1.1000	1.1300	2.2389	0.7389	0.7389	0.2117	0.1139
47 7	0.9842	0.8694	1.3043	1.4627	0.0000	0.0000	0.0149	0.0000
48 8	1.2728	0.9184	1.1460	0.3853	0.0000	0.0000	0.2543	0.0124
49 9	2.1211	0.8833	1.5081	2.5545	1.0545	1.0545	1.1378	0.4042
50 10	0.9138	0.7461	0.7935	0.6627	0.0000	0.0000	0.0677	0.0000

Figure 8: W vector and payoffs for ESO, Call, Spread and Rainbow Options, first 10 trials

In the cell range B41:E41 we calculate the 4 entries in the $\boldsymbol{z}$ vector of standard normal random variables. The Excel code is **=NORMSINV(RAND())** in each of these cells. In the cell range F41:I41 we compute the 4 entries in the $\boldsymbol{y}$ vector of log price relatives. This is computed using the Excel matrix formula to compute $\boldsymbol{y}' = \boldsymbol{z}'\boldsymbol{C}' + \boldsymbol{\mu}'$ which is **=MMULT(B41:E41,TRANSPOSE($B$33:$E$36))+$B$24:$E$24**. Note that in the excel implementation the $\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{w}$ vectors are set up as row vectors, as is the mean vector, hence the need to compute $\boldsymbol{y}' = \boldsymbol{z}'\boldsymbol{C}' + \boldsymbol{\mu}'$ instead of $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{z} + \boldsymbol{\mu}$. In the cell range J41:M41 we compute the components (price relatives) for the $\boldsymbol{w}$ vector by exponentiating the components of the $\boldsymbol{y}$ vector. Having set up the code for one iteration of the simulation we can copy the range B41:M41 to the rows below to compute the results for other iterations of the simulation.

In Figure 8 we show the payoff calculations for the ESO and the other example multi asset options we considered. These are all functions of the four components of the W vector.

To test the simulation we copied these rows so we had a simulation sample of size *1,000,000*, making the spreadsheet file very large. In practice the iterations of the calculation would probably be done using the Excel add-on Visual Basic (VBA) so as to avoid the need to keep all of the results for each of the iterations and store only the relevant final results. The results were as follows:

Call Option	Spread Option	Rainbow Option	Item calculated
$0.3073	$0.1398	$0.1271	estimated expected payoff
$1.0060	$0.3267	$0.2688	estimated standard deviation of payoff
1000000	1000000	1000000	sample size
0.33%	0.23%	0.21%	estimated coefficient of variation
$0.2394	$0.1089	$0.0989	estimated value of option

Figure 9: Option price results

From Figure 9 we see that the sample average, which is our estimated expected payoff, was *0.1704*. The sample standard deviation was *0.8545.* The estimated coefficient of variation of the sample average was *0.50%* based on the sample average, sample standard deviation and sample size, which was 1,000,000. The estimated option value is $0.1327. Based on the coefficient of variation we claim that the ESO value is 95% certain to be within 1.00% of the estimated value. For the Call option the estimated value is $0.2394 and we are 95% confident that the true value of the call is within 0.66% of this estimated value. Pricing the call analytically the result is $0.2397. The ratio of the MC price of the call to the analytic value of the call is 99.87%. Based on the results we've achieved a reasonable level of accuracy for the contract prices.

There are methods to make the MC simulation approach faster and more accurate. These include variance reduction techniques such as antithetic variates and more sophisticated programming approaches but these are beyond the scope of this paper and would use more sophisticated tools such as Matlab, R, Mathematica, C, C++. Using Excel for this calculation however is beneficial for understanding how the method works. Using Excel we started with the input data, performed the intermediate calculations which are quite involved, and produced the final results all in the one file. This file gets big and unwieldy with a large sample size. In practice, industry practitioners would want to automate these calculations in the Excel Visual Basic programming language or possibly perform them using some other software package.

6. Conclusion

In this paper we have discussed the Monte Carlo simulation method for the pricing of options on multiple underlying sources of uncertainty. We covered various aspects of the theory of the MC method and demonstrated its application to the problem of option pricing for options over a single asset, and also for an executive compensation option pricing example involving multiple periods with multiple sources of uncertainty. We have covered the detailed Excel implementation of these calculations. Our Excel demonstration for options on a single underlying asset with a single source of uncertainty (for the plain vanilla call and put options) for which the price and variance was computed analytically, illustrate the sample size required to achieve a given level of confidence in the result. This highlights the statistical nature of the MC method for the student.

The methods we have presented provide the student with a readily extendable procedure to price options with an arbitrary number of sources of uncertainty, which may in practice consist of multiple underlying assets evaluated at multiple times. The major advantage of an Excel illustration is that the input assumptions, intermediate calculations and final results are all visible to the student in the one file, thus helping the student to gain a better understanding of the method and its implementation.

In our discussion on the extension of the MC method to the pricing of multivariate contingent financial contracts, we provided a complete coverage of all the background theory required to understand the multi asset Black-Scholes framework.

This included the relevant multivariate statistical theory and the theory of covariance matrices and their Cholesky square root. In this regard, we have endeavoured to construct a self-contained exposition for students and practitioners interested in learning and applying these methods. We have presented a constructive proof of the existence and uniqueness of the Cholesky square root matrix which forms the basis of the multi-dimensional simulation algorithm, and shown how to perform the calculations in Excel. The implementation procedure of the algorithm naturally contains a diagnostic check of whether the proposed covariance matrix used in the calculations is positive definite. This is an important intermediate step to ensure that the final calculated prices are correct, as otherwise the methodology we present is not applicable and the final results will not be reliable.

The method for simulating correlated normally distributed random variables and the simulation of correlated Log-Normally distributed random variables has applications to statistical modelling outside of option pricing which we do not cover here. The detailed implementation of the MC method for the economic valuation of multi asset contingent contracts using Excel is an advanced topic in financial mathematics, normally covered in senior undergraduate units or postgraduate units. The demonstration of the method in a commonly available software package such as Excel is a useful way to facilitate a wider understanding of these methods.

References

[1] Black, F. and Scholes, M. (1973). The pricing of corporate liabilities., *Journal of Financial Economics*: **4(3)**: 323-338.

[2] Boyle, P. (1977). Options: a monte carlo approach., *Journal of Political Economy*: **81**:637–654.

[3] Brewster, K.D., Feng, Y. and Kwan, C.Y., (2012). Geometric Brownian Motion, Option Pricing, and Simulation: Some Spreadsheet-Based Exercises in Financial Modeling., *Spreadsheets in Education*: **5(3)**: Article 4.

[4] Buchen, P. (2004). The pricing of dual expiry exotics *Quantitative Finance*: **4**(1):1–8.

[5] Buchen, P. (2012). An introduction to exotic option pricing, Boca Raton, FL: Chapman & Hall/CRC.

[6] Engle, R.F. (2002). Dynamic conditional correlation: A simple class of multivariate GARCH models. *Journal of Business and Economic Statistics*: **20**(3):339–350.

[7] Fonseca, J., Grasselli, M. and Tebaldi, C. (2007). Option pricing when correlations are stochastic: An analytical framework. *Review of Derivatives Research*: **10**:151–180.

[8] Graybill, F.A. (1976), Theory and Application of the Linear Model. ISBN 0-87872-108-8.

[9] Harrison, J.M. and Kreps, D. (1979). Martingales and arbitrage in multi-period securities markets, *Journal of Economic Theory*, **20(3):**381-408.

[10] Harrison, J, and Pliska, S: (1981) Martingales and stochastic integrals in the theory of continuous trading, *Stochastic Processes and their Applications* **11(3):**15-260.

[11] Harrison, J.M. and Pliska, S.R. (1983). Stochastic calculus model of continuous trading: complete markets, *Stochastic Processes and their Applications*: **15:**313–316.

[12] Hull, J.C. (2006) Options , Futures and Other Derivatives, 6th ed ISBN 0-13-197705-9

[13] Kac, M. (1949). On distributions of certain Wiener functionals *Transactions of the American Mathematical Society*: **65**(1) 1–13.

[14] Konstandatos, O. (2003). A new framework for pricing barrier and lookback options, PhD Thesis, School of Mathematics and Ststistics, The University of Sydney.

[15] Konstandatos, O. (2008). Pricing path dependent options: A comprehensive mathematical framework, Saarbrücken, FL: VDM Verlag Dr Mueller & Co Kg Aktiengesellschaft. ISBN:9783639055917.

[16] Konstandatos, O. and Kyng, T. (2012). Real options analysis for commodity based mining enterprises with compound and barrier features, *Accounting and Finance Research*: **1**(2) 216–234.

[17] Kwan, C. C. Y. (2010) "The Requirement of a Positive Definite Covariance Matrix of Security Returns for Mean-Variance Portfolio Analysis: A Pedagogic Illustration," *Spreadsheets in Education (eJSiE)*: Vol. 4: Iss. 1, Article 4.

[18] Kwan, C. C. Y. (2011) "An Introduction to Shrinkage Estimation of the Covariance Matrix: A Pedagogic Illustration," *Spreadsheets in Education (eJSiE)*: Vol. 4: Iss. 3, Article 6.

[19] Kyng, T. and Taylor, P. (2008). Graduate's use of spreadsheet tools in learning and applying financial mathematics, *Asian Social Science*: **4**(3).

[20] Kyng, T. Tickle, L, and Wood, L. (2013). Perceptions of the software skills of graduates by employers in the financial services industry, *International Journal of Mathematical Education in Science and Technology*: **44**(8). [19] Tham, J. (2001). Risk-neutral valuation, a gentle introduction, *Available at SSRN: http://ssrn.com/abstract=290044 or* http://dx.doi.org/10.2139/ssrn.290044 Nov 2001.

[21] Tham, J. (2001). Risk-neutral valuation, a gentle introduction, *Available at SSRN: http://ssrn.com/abstract=290044 or* http://dx.doi.org/10.2139/ssrn.290044 Nov 2001.

7. Appendix A: Notational Conventions

Before proceeding, we give a brief account of notational conventions we employ in this paper. A bold face upper case letter e.g. $\boldsymbol{A}$ denotes a matrix. The transpose of this matrix is denoted $\boldsymbol{A}'$. A bold face lower case letter e.g. $\boldsymbol{b}$ denotes a column vector. The transpose of this vector is denoted $\boldsymbol{b}'$. We use $\mathbf{0}$ to denote a column vector of zeros. A plain text lower case letter with two subscripts e.g. a_{ij} denotes a matrix element in row i and column j of the matrix $\boldsymbol{A}$. A plain text lower case letter with one subscript e.g. b_i denotes a vector element from the vector $\boldsymbol{b}$. A plain text lower case letter without subscripts denotes a scalar.

8. Appendix B: Derivation of the Power Option and Power Binary Option Analytic Formulae.

The *power option with power n*, where n is some constant, is a financial contract that pays the holder some power of the price of the underlying asset at the maturity date T, with payoff $P = (S_T)^n$. The binary power option also pays some arbitrary power of the risky asset at maturity, provided that an exercise condition on the stock price is met. There are in fact two types, the up and the down power binary options. The payoffs at expiry from both may be written as: $V^s(S,T) = (S_T)^n I\{sS_T > sX\}$

The price relative S_T/S_t has a lognormal distribution and it can be written in terms of the standard normal variable Z as $S_T/S_t = exp\left(\left(r - y - \frac{1}{2}\sigma^2\right)\tau + \sigma\sqrt{\tau}Z\right)$ where $\tau = T - t$. The nth power of this price relative is also lognormally distributed: $(S_T/S_t)^n = exp\left(n\left(r - y - \frac{1}{2}\sigma^2\right)\tau + n\sigma\sqrt{\tau}Z\right)$ Using standard properties of the lognormal distribution the expectation of the nth power of the price relative is $E\{(S_T/S_t)^n\} = exp\left(n\left(r - y - \frac{1}{2}\sigma^2\right)\tau + \frac{1}{2}n^2\sigma^2\tau\right)$. The discounted expectation is $e^{-r\tau}E\{(S_T/S_t)^n\} = exp\left(\frac{1}{2}n^2\sigma^2\tau + n\left(r - y - \frac{1}{2}\sigma^2\right)\tau - r\tau\right)$. We can write this as $e^{-r\tau}E\{(S_T/S_t)^n\} = exp(\gamma(n)\tau)$ where $\gamma(n) = \frac{1}{2}n^2\sigma^2 + n\left(r - y - \frac{1}{2}\sigma^2\right) - r$. It follows that the value at time t of the power option with payoff $P = (S_T)^n$ is $e^{-r\tau}E\{(S_T/S_t)^n\}{S_t}^n = {S_t}^n exp(\gamma(n)\tau)$

The up binary power option with payoff $V^{+1}(S,T) = (S_T)^n I\{S_T > X\}$ has a value at time t given by the discounted expectation $e^{-r\tau}E[(S_T)^n I\{S_T > X\}|S_t]$. The indicator function expressed in terms of S_T has an equivalent indicator function expressed in terms of the variable Z, and the equivalence is $I\{S_T > X\} \equiv I\{Z > -d_2\}$ where d_2 is as in the Black Scholes formula for a European call option with an exercise price of X. This allows us to obtain a formula for the discounted expectation of the payoff on the binary power option contract expressed as an integral in terms of z as follows:

$$e^{-r\tau}E[(S_T)^n I\{S_T > X\}|S_t] =$$
$$(S_t)^n \int_{-d_2}^{\infty} exp\left(n\left(r - y - \frac{1}{2}\sigma^2\right)\tau + n\sigma\sqrt{\tau}z - r\tau\right)p(z)dz \quad (A1)$$

where $p(z) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{1}{2}z^2\right)$ is the pdf of $\boldsymbol{Z}$.

The integral is of the form $I = \int_{\delta}^{\infty} exp(\alpha + \beta z)p(z)dz$. Straight-forward integration gives $I = exp\left(\alpha + \frac{1}{2}\beta^2\right)\int_{\delta-\beta}^{\infty} p(u)du$ on changing variables to $u = z - \beta$. In turn this yields $I = exp(\gamma(n)\tau)N(d_2 + n\sigma\sqrt{\tau})$ on substituting $\alpha = n\left(r - y - \frac{1}{2}\sigma^2\right)\tau - r\tau, \beta = n\sigma\sqrt{\tau}$ and $\delta = -d_2$. It follows that the value of the binary power option is as stated in section 2.3.1.

These binary power options can be combined to create other options such as European call and put options, gap call and put options and various other types of options. In particular they can be used to obtain analytic formulae for the mean and the variance of the call and put option payoffs.

9. Appendix C: Existence of Cholesky Square Root Matrix

The following result is used in the constructive proof of the Cholesky Square Root matrix which follows.

Theorem [Determinants of Partitioned Matrices] Let $\boldsymbol{B}$ be an $n \times n$ partitioned matrix, partitioned as follows: $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B_{11}} & \boldsymbol{B_{12}} \\ \boldsymbol{B_{21}} & \boldsymbol{B_{22}} \end{bmatrix}$ where $\boldsymbol{B_{ij}}$ has size $n_i \times n_j$ for $i, j = 1{,}2$, where $n_1 + n_2 = n$ and $0 < n_1 < n$. Let $\boldsymbol{A} = \boldsymbol{B^{-1}}$ be the $n \times n$ inverse of $\boldsymbol{B}$, and partitioned as $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} \end{bmatrix}$ where $\boldsymbol{A_{ij}}$ has size $\boldsymbol{n_i} \times \boldsymbol{n_j}$.

If $det(\boldsymbol{B_{22}}) \neq \boldsymbol{0}$ then $det(\boldsymbol{B}) = det(\boldsymbol{B_{22}}) det\left(\boldsymbol{B_{11}} - \boldsymbol{B_{12}} \boldsymbol{B_{22}}^{-1} \boldsymbol{B_{21}}\right)$

If $det(\boldsymbol{B_{11}}) \neq \boldsymbol{0}$ then $det(\boldsymbol{B}) = det(\boldsymbol{B_{11}}) det\left(\boldsymbol{B_{22}} - \boldsymbol{B_{21}} \boldsymbol{B_{11}}^{-1} \boldsymbol{B_{12}}\right)$

Proof: See Graybill (1976) pp19-21

Definition. [Positive Definite Matrix] An n by n matrix $\underline{S}$ is positive definite if and only if all of its eigenvalues are strictly positive. If so then $\underline{y}' \times \underline{S} \times \underline{y} > 0$ for all vectors $\underline{y} \in R^n$ such that $\underline{y} \neq \underline{0}$. If all of the eigenvalues are non negative (but 0 is an eigenvalue) then the matrix is non negative definite instead of positive definite.

We next state and prove an important theorem about positive definite covariance matrices. This theorem and its proof follow Graybill (1976) p260 but with slight modifications for the purposes of this paper.

Theorem: [Existence of the Cholesky Square Root]. Let $\boldsymbol{S}$ be a $p \times p$ positive definite symmetric real matrix. Then there exists an upper triangular $p \times p$ real matrix $\boldsymbol{T}$ of rank p with $\boldsymbol{t_{ii}} > 0$ for $i = 1{,}2, \dots, p$ such that $\boldsymbol{TT'} = \boldsymbol{S}$ and $\boldsymbol{T}$ is unique.

Proof: We use mathematical induction. First, let $p = 1$. Since $\boldsymbol{S}$ is positive definite and real we can write $\boldsymbol{S} = \boldsymbol{s_{11}}^{2}$. So $t_{11} = |s_{11}|$ and $\boldsymbol{T}$ is upper triangular with $\boldsymbol{t_{11}} > 0$ and it is unique.

Now suppose the theorem is true for $p = k$, so that for any $k \times k$ positive definite symmetric real matrix $\boldsymbol{S_{11}}$ there exists a unique upper triangular real matrix $\boldsymbol{T_{11}}$ with $\boldsymbol{t_{ii}} > 0$ for $i = 1{,}2, \dots, k$ such that $\boldsymbol{T_{11}} \boldsymbol{T_{11}}' = \boldsymbol{S_{11}}$. Let $\boldsymbol{S}$ be a $k + 1 \times k + 1$ positive definite symmetric real matrix. Since it is positive definite we can write $\boldsymbol{S}$ as a partitioned matrix $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S_{11}} & \boldsymbol{S_{12}} \\ \boldsymbol{S_{21}} & \boldsymbol{S_{22}} \end{bmatrix}$ where $\boldsymbol{S_{11}}$ is of size $k \times k$ $\boldsymbol{S_{12}}$ is of size $k \times 1$, $\boldsymbol{S_{21}} = \boldsymbol{S_{12}}'$ is of size $1 \times k$ and $\boldsymbol{S_{22}}$ is of size 1×1 (a scalar). By our induction hypothesis we can write $\boldsymbol{S_{11}} = \boldsymbol{T_{11}}\boldsymbol{T_{11}}'$ where $\boldsymbol{T_{11}}$ is a unique upper triangular $k \times k$ real matrix with positive diagonal elements.

Let $\boldsymbol{C_{11}} = \boldsymbol{T_{11}}'$ and define $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T_{11}} & \boldsymbol{T_{12}} \\ \boldsymbol{T_{21}} & \boldsymbol{T_{22}} \end{bmatrix}$ as a $k+1 \times k+1$ partitioned matrix where where $\boldsymbol{T_{11}}$ is of size $k \times k$ $\boldsymbol{T_{12}}$ is of size $k \times 1$, $\boldsymbol{T_{12}} = \boldsymbol{C_{11}}^{-1}\boldsymbol{S_{12}}$, $\boldsymbol{T_{21}}$ is of size $1 \times k$ and all its entries are zero, and $\boldsymbol{T_{22}}$ is of size 1×1 (a scalar), with $\boldsymbol{T_{22}} = b$ and $b = \sqrt{\left(s_{22} - \boldsymbol{S_{21}}\boldsymbol{S_{11}}^{-1}\boldsymbol{S_{12}}\right)}$

Then $\boldsymbol{T}'\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T_{11}}' & \boldsymbol{0} \\ \boldsymbol{S_{12}}'\boldsymbol{T_{11}}^{-1} & b \end{bmatrix} \begin{bmatrix} \boldsymbol{T_{11}} & \boldsymbol{C_{11}}^{-1}\boldsymbol{S_{12}} \\ \boldsymbol{0} & b \end{bmatrix} = \begin{bmatrix} \boldsymbol{T_{11}}'\boldsymbol{T_{11}} & \boldsymbol{S_{12}} \\ \boldsymbol{S_{21}} & \boldsymbol{S_{12}}'\boldsymbol{S_{11}}^{-1}\boldsymbol{S_{12}} + \boldsymbol{b^2} \end{bmatrix} = \boldsymbol{S}$

Because: $\boldsymbol{T_{11}}'\boldsymbol{T_{11}} = \boldsymbol{S_{11}}$, $\boldsymbol{S_{12}}'\boldsymbol{T_{11}}^{-1}\boldsymbol{T_{11}} = \boldsymbol{S_{12}}' = \boldsymbol{S_{21}}$, $\boldsymbol{S_{12}}'\boldsymbol{T_{11}}^{-1}\boldsymbol{C_{11}}^{-1}\boldsymbol{S_{12}} = \boldsymbol{S_{12}}'(\boldsymbol{C_{11}}\boldsymbol{T_{11}})^{-1}\boldsymbol{S_{12}} = \boldsymbol{S_{12}}'(\boldsymbol{S_{11}})^{-1}\boldsymbol{S_{12}}$

To complete the proof we need to show that $s_{22} - \boldsymbol{S_{21}}\boldsymbol{S_{11}}^{-1}\boldsymbol{S_{12}} > 0$ so that b is real. To show this we write $det(\boldsymbol{S}) = det(\boldsymbol{S_{11}})\boldsymbol{det}\left(s_{22} - \boldsymbol{S_{21}}\boldsymbol{S_{11}}^{-1}\boldsymbol{S_{12}}\right)$. Since both $\boldsymbol{S}$ and $\boldsymbol{S_{11}}$ are positive definite we have $det(\boldsymbol{S}) > 0$ and $det(\boldsymbol{S_{11}}) > 0$ from which it follows that $\boldsymbol{det}\left(s_{22} - \boldsymbol{S_{21}}\boldsymbol{S_{11}}^{-1}\boldsymbol{S_{12}}\right) > 0$ and since $s_{22} - \boldsymbol{S_{21}}\boldsymbol{S_{11}}^{-1}\boldsymbol{S_{12}}$ is a scalar it equals its determinant and hence $s_{22} - \boldsymbol{S_{21}}\boldsymbol{S_{11}}^{-1}\boldsymbol{S_{12}} > 0$. Hence the matrix $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T_{11}} & \boldsymbol{C_{11}}^{-1}\boldsymbol{S_{12}} \\ \boldsymbol{0} & b \end{bmatrix}$ is unique, upper triangular and all its diagonal elements are positive. The transpose of $\boldsymbol{T}$ is **C**=$\begin{bmatrix} \boldsymbol{C_{11}} & \boldsymbol{0} \\ \boldsymbol{S_{12}}'\boldsymbol{T_{11}}^{-1} & b \end{bmatrix}$ which is the Cholesky Square Root of **S.**

This result allows us to compute the Cholesky Square Root of the $k+1 \times k+1$ matrix **S** from the Cholesky square root of the $k \times k$ matrix $\boldsymbol{S_{11}}$ and the other entries in **S**. This allows us to compute the Cholesky square root matrix recursively. We can implement the calculations in Excel using the built in matrix functions such as transpose, minverse, and mmult, as demonstrated in Section 4.