**IVAN FRANKO NATIONAL UNIVERSITY OF LVIV**

Faculty of Applied Mathematics and Informatics

information systems department

Course report

**«Computer Methods of Financial Mathematics»**

Made by:

Bohdan Romaniuk

student of AMI-43 group

Checked by:

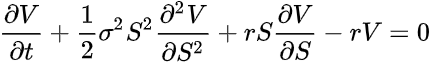
Yaroslav Kondratyuk

Lviv-2019

1. **Black-Sholes Partial Differential Equation (Black-Sholes PDE)**

An option is a derivative security, its value depends on the value, or price, of some other underlying security, called the underlying security. Let **S** denote the value, or price, of this underlying security. We need to keep track of what time this price is observed at, so let **St** denote that the price is observed at time **t**. A call (put) option gives the holder the right, but not the obligation, to buy (sell) some underlying asset at a given price **K**, called the exercise price, on or before some given date **T**. If the option is a so called European option, it can only be used (exercised) at the maturity date. If the option is of the so called American type, it can be used (exercised) at any date up to and including the maturity date **T**. If exercised at time **T**, a call option provides payoff  and a put option provides payoff 

**Black–Scholes equation** is a [partial differential equation](https://en.wikipedia.org/wiki/Partial_differential_equation), which describes the price of the option over time. The equation is:



The key financial insight behind the equation is that one can perfectly [hedge](https://en.wikipedia.org/wiki/Hedge_(finance)) the option by buying and selling the [underlying](https://en.wikipedia.org/wiki/Underlying) asset in just the right way and consequently "eliminate risk". This hedge, in turn, implies that there is only one right price for the option, as returned by the Black–Scholes formula.

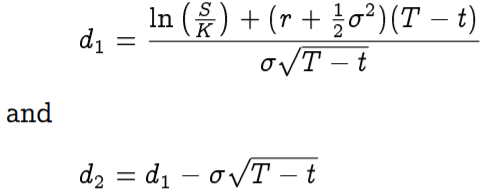
**The Black–Scholes formula** calculates the price of [European](https://en.wikipedia.org/wiki/European_option) [put](https://en.wikipedia.org/wiki/Put_option) and [call options](https://en.wikipedia.org/wiki/Call_option). This price is [consistent](https://en.wikipedia.org/wiki/Consistency) with the Black–Scholes equation [as above](https://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model#Black%E2%80%93Scholes_equation_and_its_derivation), this follows since the formula can be obtained [by solving](https://en.wikipedia.org/wiki/Equation_solving#Differential_equations) the equation for the corresponding terminal and boundary conditions.

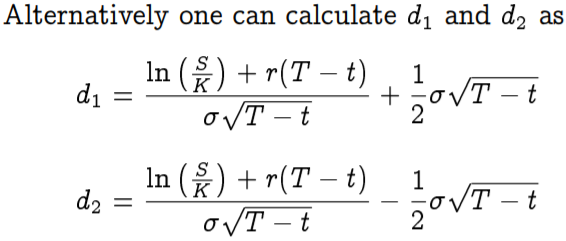
**The Black Scholes formula** provides analytical solutions for European put and call options, options which can only be exercised at the options maturity date. Black and Scholes showed that the addition information needed to price the option is the (continuously compounded) risk free interest rate **r**, the variability of the underlying asset, measured by the standard deviation **σ** of (log) price changes, and the time to maturity **(T - t)** of the option, measured in years. The original formula was derived under the assumption that there are no payouts, such as stock dividends, coming from the underlying security during the life of the option. Such payouts will affection option values, as will become apparent later.

The Black Scholes formula for a ***call option*** is:



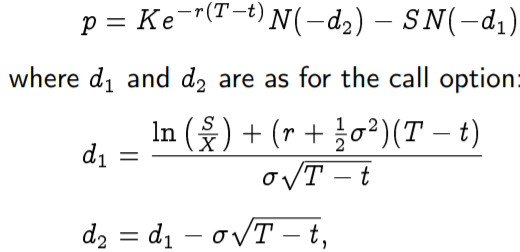
Where **d1** and **d2** can be calculated from:





Where **S** is the price of the underlying security, **K** the exercise price, r the (continuously compounded) risk free interest rate, **σ** the standard deviation of the underlying asset, t the current date, **T** the maturity date, **T - t** the time to maturity for the option and **N(·)** the cumulative normal distribution.

The Black Scholes price for a ***put option*** is:



Where **S** is the price of the underlying security, **K** the exercise price, **r** the (continuously compounded) risk free interest rate, **σ** the standard deviation of the underlying asset, **T - t** the time to maturity for the option and **N(·)** the cumulative normal distribution.

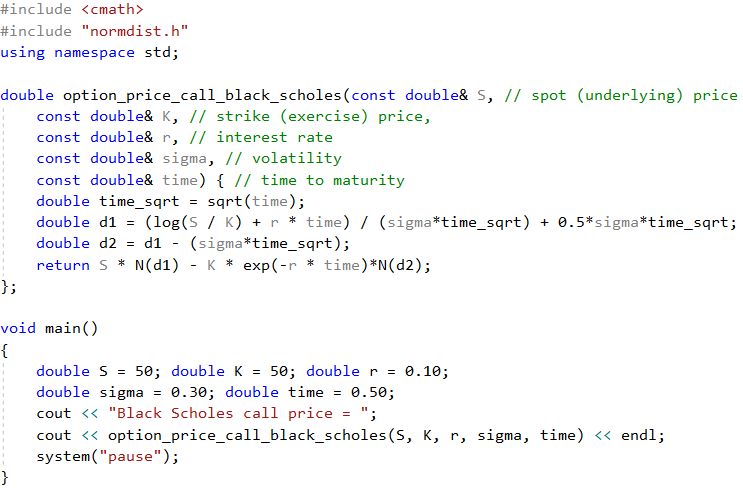
**Example 1.1**

Stock in company “Apple” is currently trading at 50. Consider a call option on “Apple” stock with an exercise price of K = 50 and time to maturity of 6 months. The volatility of “Apple” stock has been estimated to be σ = 30%. The current risk free interest rate (with continuous compounding) for six month borrowing is 10%.

The task is to find the call option price.

To calculate the price of this option we use the Black Scholes formula with inputs S = 50, K = 50, r = 0.10, σ = 0.3 and (T - t) = 0.5.

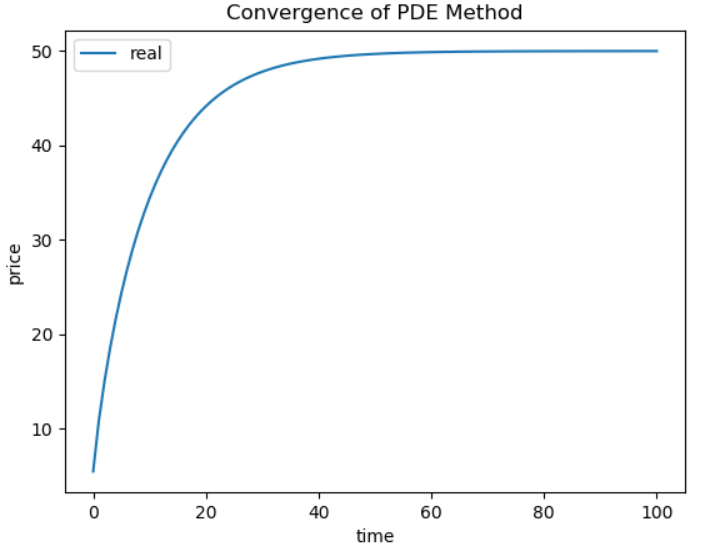
C++ programming language with ***normdist.h*** library is used to implement this task.



**Price of European call option using the Black Scholes formula**

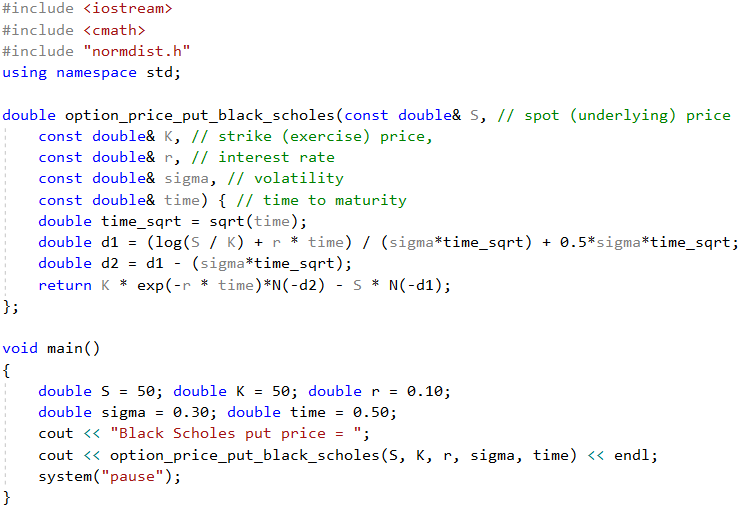
The result is: Black Scholes call price = 5.45325

Expected price is S=50, as we can use during period of time it converges to that point.



**Example 1.2**

We have the same company with the same data as above in **Example 1.1**, but now need to find put option price using another equations.



**Price of European put option using the Black Scholes formula**

The result is: Black Scholes put price = 3.01472

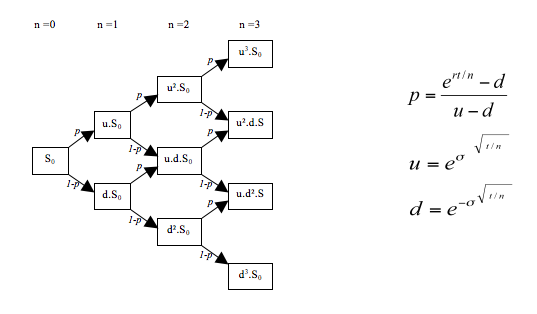
1. **Binomial Method**

The Binomial Method provides a generalizable [numerical method](https://en.wikipedia.org/wiki/Numerical_analysis) for the valuation of [options](https://en.wikipedia.org/wiki/Option_(finance)). Essentially, the model uses a "discrete-time" model of the varying price over time of the [underlying](https://en.wikipedia.org/wiki/Underlying) financial instrument.

Binomial Method is based on the description of an [underlying instrument](https://en.wikipedia.org/wiki/Underlying_instrument) over a period of time rather than a single point. Method is computationally slower than the [Black–Scholes formula](https://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model), it is more accurate, particularly for longer-dated options on securities with [dividend](https://en.wikipedia.org/wiki/Dividend) payments. For these reasons, various versions of the binomial model are widely used in the options markets.

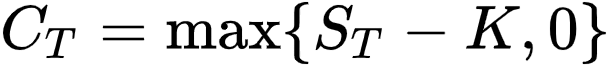
The binomial pricing model traces the evolution of the option's key underlying variables in discrete-time. This is done by means of a binomial tree, for a number of time steps between the valuation and expiration dates. Each node in the tree represents a possible price of the underlying at a current point in time.

Valuation is performed iteratively, starting at each of the final nodes, and then [working backwards](https://en.wikipedia.org/wiki/Backward_induction) through the tree towards the first node (valuation date). The value computed at each stage is the value of the option at that point in time.



**General schema of Binomial tree**

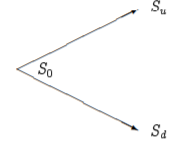
So there is a tree-step process:

1. Price tree generation
2. Calculation of option value at each final node 
3. Sequential calculation of the option value at each previous node

### Step 1: Create the binomial price tree

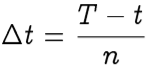
The tree of prices is produced by working straight forward from valuation date to expiration.

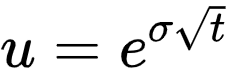
The price of the underlying is currently **S0**. The price in next period can only take on of two values, **Su** and **Sd**. For each step, it is assumed that the [underlying instrument](https://en.wikipedia.org/wiki/Underlying_instrument) will move up or down by a specific factor (**u**{\displaystyle u} or **d**{\displaystyle d}ddddd) per step of the tree (where, by definition, {\displaystyle u\geq 1} **u≥1** and **{\displaystyle 0<d\leq 1}0<d≤1**). So, if **S0** is the current price, then in the next period of time the price will either be **Su** = **S0·u** or **Sd** = **S0·d**

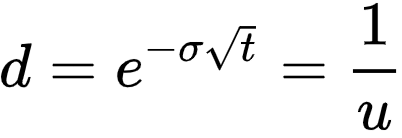


**Pricing in binomial tree model**

The up and down factors are calculated using the underlying [volatility](https://en.wikipedia.org/wiki/Volatility_(finance)), **{\displaystyle \sigma }σ**, and the time duration of a step, **t**, measured in years (using the [day count convention](https://en.wikipedia.org/wiki/Day_count_convention) of the underlying instrument). From the condition that the [variance](https://en.wikipedia.org/wiki/Variance) of the log of the price is {\displaystyle \sigma ^{2}t}σ2t, we have:

****





The binomial model is pretty simple, since there are only two possible states. If we find the probability **p** of one state, we also find the probability of the other as **(1 – p)**, as it shows on the picture the state will be **Su** if the probability is equal to **p** and will be **Sd** if probability is equal to **1-p** (or otherwise if not equal to **p**).

Besides that asset at each node can be calculated directly via formula, and does not require the tree to be built at first. The node-value will be:  Where **Nu**{\displaystyle N\_{u}} is the number of up ticks and **Nd**{\displaystyle N\_{d}} is the number of down ticks.

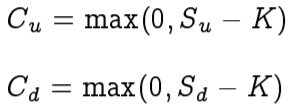
### Step 2: Find Option value at each final node

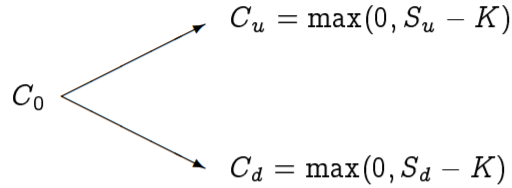
At each final node of the tree, at expiration of the option the option value is simply its time value, or exercise, value.

For call option: [**max**](https://en.wikipedia.org/wiki/Extreme_value)**((*Sn*− *K*), 0)** and for put option **max ((*K* − *Sn*), 0)**

Where **K**is the [strike price](https://en.wikipedia.org/wiki/Strike_price) and **{\displaystyle S\_{n}}Sn** is the spot price of the underlying asset at the ***n-th***period.

These last nodes have payoffs:





### Step 3: Find Option value at earlier nodes

Once the above steps are completed, the option value is then found for each node, starting at the penultimate time step, and working back to the first node of the tree (the valuation date) where the calculated result is the value of the option.

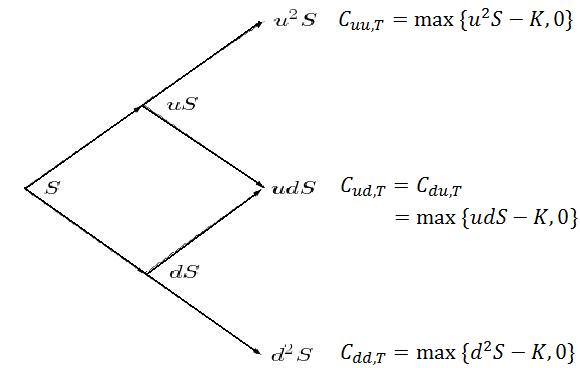
In overview: the "binomial value" is found at each node, using the [risk neutrality](https://en.wikipedia.org/wiki/Risk-neutral_measure) assumption. If exercise is permitted at the node, then the model takes the greater of binomial and exercise value at the node. Expected value is calculated using the option values from the latter two nodes (*Option up* and *Option down*) weighted by their probabilities **p** of an up move in the underlying, and probability **(1−p)** of a down move.

Moving back for each previous node option values is calculated using formula:



where   
{\displaystyle C\_{t,i}\,} C0 (Ct,i) is the option's value for the **{\displaystyle i^{th}\,}i-th** node at time **t** and artificial probability **q** calculates from formula:

This result is known as the "Binomial Value". It represents the fair price of the derivative at a particular point in time, given the evolution in the price of the underlying to that point. It is the value of the option if it were to be held - as opposed to exercise at that point.



**Binomial tree schema at each point of time and option value formulas at the last nodes**

Then the final step is to find the price at time 0:

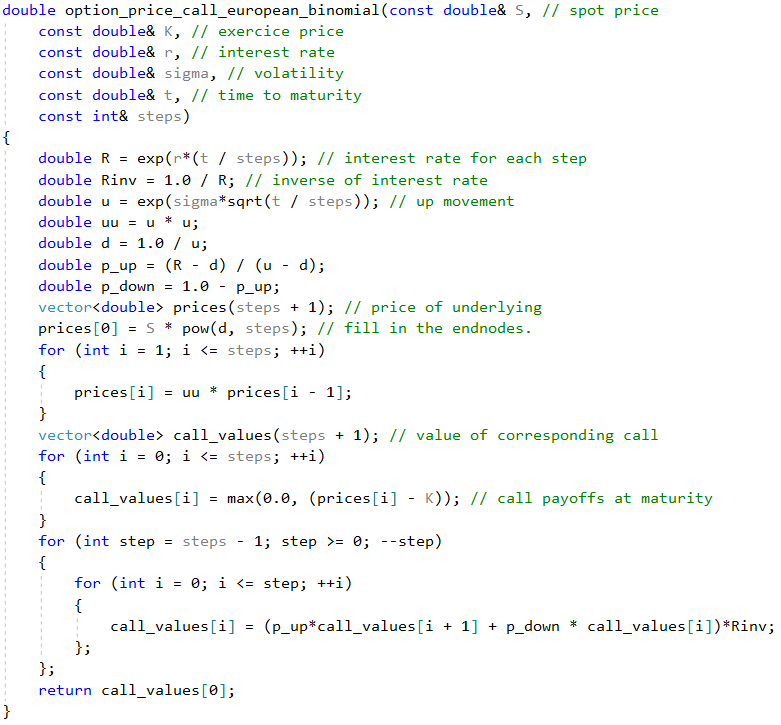


Depending on type of options (European or American) we have next rules:

* For a [European option](https://en.wikipedia.org/wiki/European_option), there is no option of early exercise, and the binomial value applies at all nodes
* For an [American option](https://en.wikipedia.org/wiki/American_option), since the option may either be held or exercised prior to expiry, the value at each node is: Max (Binomial Value, Exercise Value).

**European Options**

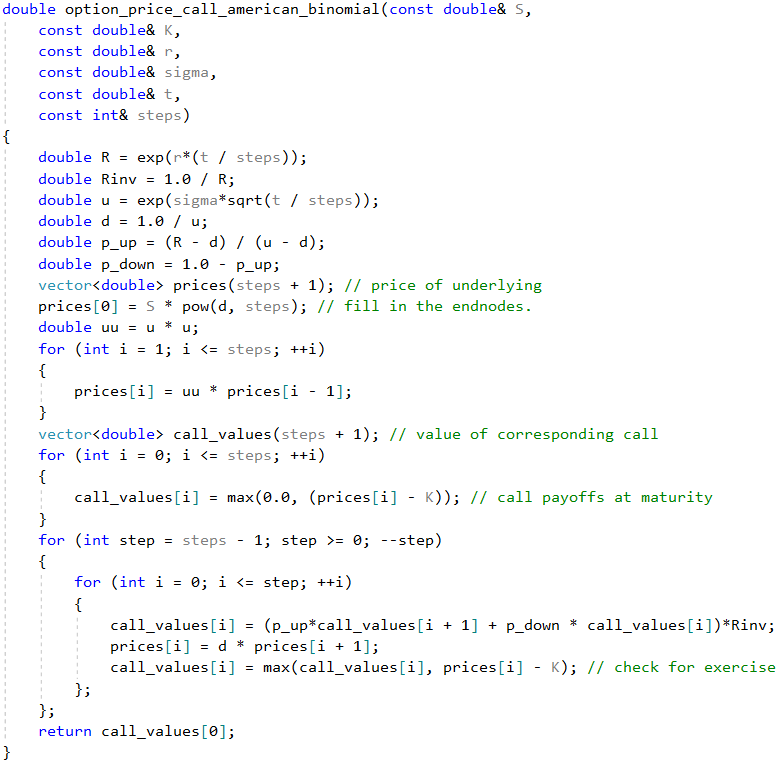
For European options, binomial trees are not that much used, since the Black Scholes model will give the correct answer, but it is useful to see the construction of the binomial tree without the checks for early exercise, which is the American case.



**Option price for binomial European option call**

**American Options**

An American option differs from an European option by the exercise possibility. An American option can be exercised at any time up to the maturity date, unlike the European option, which can only be exercised at maturity. In general, there is unfortunately no analytical solution to the American option problem, but in some cases it can be found. For example, for an American call option on non-dividend paying stock, the American price is the same as the European call. It is in the case of American options, allowing for the possibility of early exercise, that binomial approximations are useful. At each node we calculate the value of the option as a function of the next period’s prices, and then check for the value exercising of exercising the options.

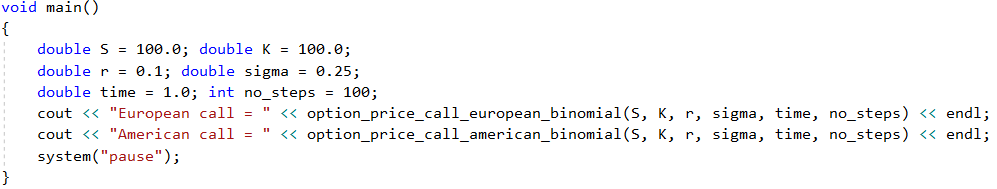


**Price of American call option using a binomial approximation**

**Example**

You are given the following information about an option: S = 100, K = 100, r = 0.1, σ = 0.25 and time to maturity is 1 year. PriceEuropean and American calls and puts using binomial approximations with 100 steps.

Implementation is using previously mentioned functions to calculate both European and American options in C++ programming language.



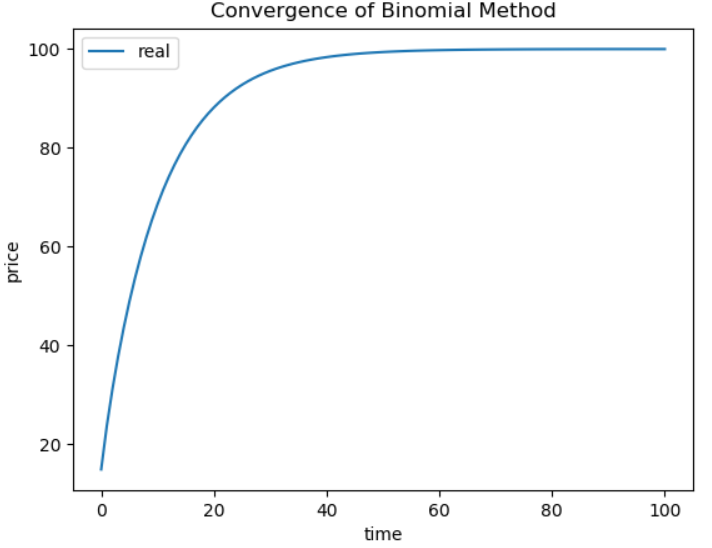
The result is:

European call = 14.9505

American call = 14.9505

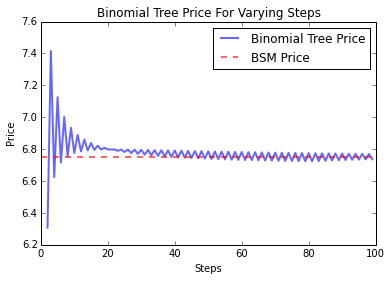
Actually, for this particular case, the American price will equal the European.

Expected price is 100, as we can use during period of time it converges to that point.



**Convergence of Binomial method**

To illustrate the behavior of the binomial approximation figure 12.1 plots a comparison with the binomial approximation as a function of n, the number of steps in the binomial approximation, and the true (Black Scholes) value of the option. Note the “sawtooth” pattern, the binomial approximation jumps back and forth around the true value for small values of n, but rapidly moves towards the Black Scholes value as the n increases.

****

**Illustrating convergence of binomial to Black Scholes**

1. **Monte Carlo Method**

**Monte Carlo simulation** is a technique used to understand the impact of risk and uncertainty in financial, project management, cost, and other forecasting models. A Monte Carlo simulator helps one visualize most or all of the potential outcomes to have a better idea regarding the risk of a decision.

We now consider using Monte Carlo methods to estimate the price of an European option, and let us first consider the case of the “usual” European Call, which can priced by the Black Scholes equation. Since there is already a closed form solution for this case, it is not really necessary to use simulations, but we use the case of the standard call for illustrative purposes.

At maturity, a call option is worth



At an earlier date t, the option value will be the expected present value of this



Now, an important simplifying feature of option pricing is the “risk neutral result,” which implies that we can treat the (suitably transformed) problem as the decision of a risk neutral decision maker, if we also modify the expected return of the underlying asset such that this earns the risk free range.

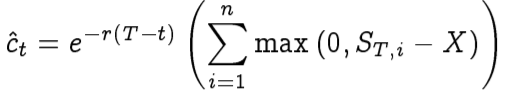


Where E [\*] is a transformation of the original expectation. One way to estimate the value of the call is to simulate a large number of sample values of ST according to the assumed price process, and find the estimated call price as the average of the simulated values. By appealing to a law of large numbers, this average will converge to the actual call value, where the rate of convergence will depend on how many simulations we perform.

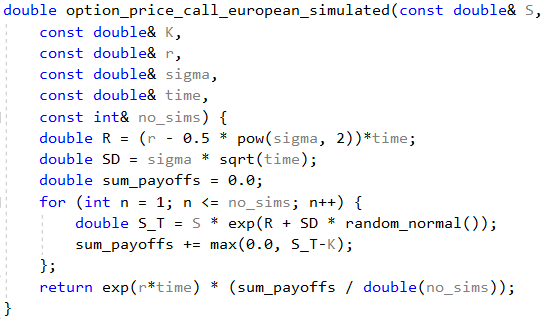
Pricing of European call option. For the purposes of doing the Monte Carlo estimation of the price of an European call.



Note that here one merely need to simulate the terminal price of the underlying, ST , the price of the underlying at any time between t and T is not relevant for pricing. We proceed by simulating log normally distributed random variables, which gives us a set of observations of the terminal price ST . If we let ST,1 ST,2 ST,3 …. ST,n denote the n simulated values, we will estimate E [max(0, ST-X)] as the average of option payoffs at maturity, discounted at the risk



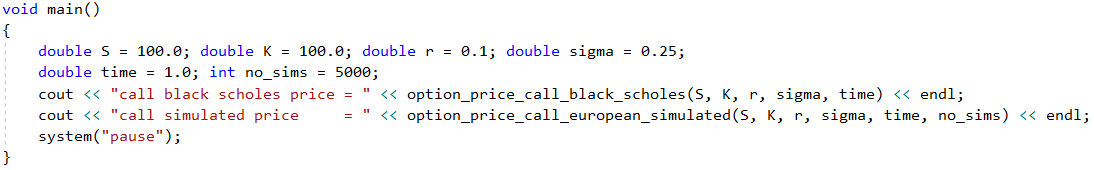
Below code shows the implementation of a Monte Carlo estimation of an European call.



**Example 3.1**

Given S = 100, K = 100, r = 0.1, σ = 0.25, time=1. Use 5000 simulations. Price put and call option using Black Scholes and simulations.

C++ program will looks like that:

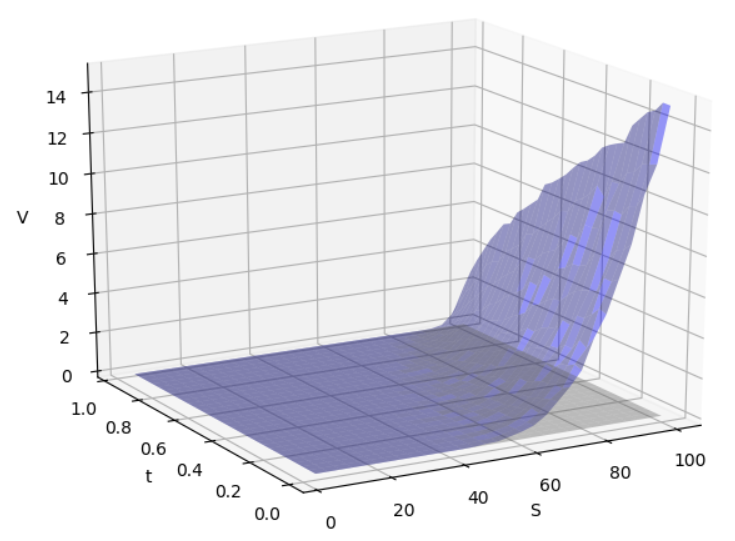


The result is:

call black scholes price = 14.9758

call simulated price = 18.7344

And the surface for this case using mathplot library.

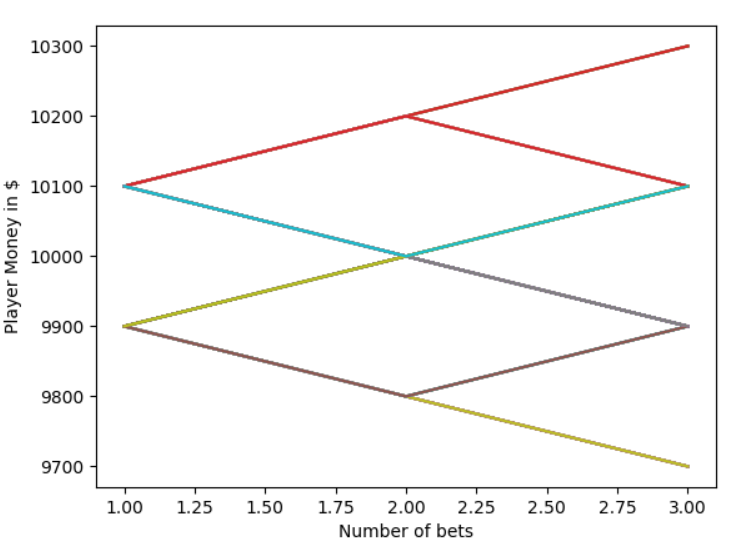


**Illustrating of Monte Carlo method example**

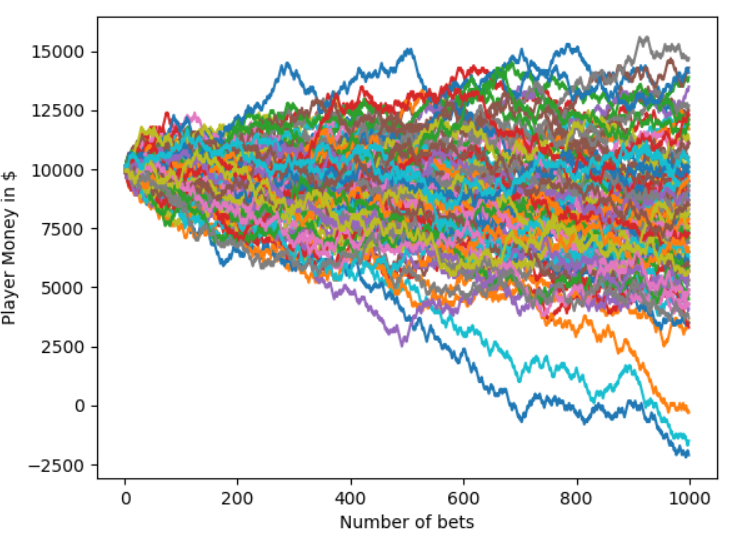
**Example 3.2**

We have an imaginary game in which our player, rolls an imaginary dice to get an outcome of 1 to 100. If player rolls anything from 1–51, the casino wins, but if the number rolled is from 52–100, player wins.

5 plays: The player starts the game with $10,000 and ends with $9974.0



1000 plays: The player starts the game with $10,000 and ends with $7940.0



It is easily understood from the simulation experiment, that player has a better chance of making a profit (or minimize loss), if he plays a few games than a big amount.