

QUANTITATIVE FINANCE IN C++

THE BINOMIAL METHOD AND ONE-FACTOR OPTION MODEL

Oromion

January 7, 2020

SoftButterfly

CONTENTS

Introduction

- Two-point boundary value problem

- Linear boundary value problems

- Black-Scholes-Merton**: exact, log-normal distribution

- Trees**: numerical method

The Finite Difference Method for PDEs

- Mathematical Background

PDE Preprocessing

- Log Transformation

Software Framework for One-Factor Option Models

- Finite Difference Method**: numerical method

- Monte Carlo Simulation**: numerical method

INTRODUCTION

INTRODUCTION

We introduce a class of *second-order ordinary differential equations* as they contain derivatives up to order 2 in one independent variable. Furthermore, the (unknown) function appearing in the differential equation is a function of a single variable. A simple example is the *linear* equation

$$Lu \equiv a(x)u'' + b(x)u' + c(x)u = f(x) \quad (1)$$

In general we seek a solution u of (1) in conjunction with some auxiliary conditions. The coefficients a , b , c and f are known functions of the variable x . Equation (1) is called linear because all coefficients are independent of the unknown variable u . Furthermore, we have used the following shorthand for the first and second order derivatives with respect to x :

$$u' = \frac{du}{dx} \quad \text{and} \quad u'' = \frac{d^2u}{dx^2}. \quad (2)$$

INTRODUCTION

We examine (1) in some detail because it is part of the Black-Scholes equation

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC = 0 \quad (3)$$

where the asset price S plays the role of the independent variable x and t plays the role of time.

We replace the unknown function u by C (the option price). Furthermore, in this case, the coefficients in (1) have the special form

$$\begin{aligned} a(S) &= \frac{1}{2}\sigma^2 S^2 \\ b(S) &= rS \\ c(S) &= -r \\ f(S) &= 0 \end{aligned} \quad (4)$$

In the following chapters our intention is to solve problems of the form (1) and we then apply our results to the specialized equations in quantitative finance.

INTRODUCTION

TWO-POINT BOUNDARY VALUE PROBLEM

TWO-POINT BOUNDARY VALUE PROBLEM

Let us examine a general second-order differential equation given in the form

$$u'' = f(x; u, u') \quad (5)$$

where the function f depends on three variables. The reader may like to check that (1) is a special case of (5). In general, there will be many solutions of (5) but our interest is in defining extra conditions to ensure that it will have a unique solution. Intuitively, we might correctly expect that two conditions are sufficient, considering the fact that you could integrate (5) twice and this will deliver two constants of integration. To this end, we determine these extra conditions by examining (5) on a *bounded* interval (a, b) . In general, we discuss linear combinations of the unknown solution u and its first derivative at these end-points:

$$\begin{aligned} a_0 u(a) - a_1 u'(a) &= \alpha, & |a_0| + |a_1| &\neq 0 \\ b_0 u(b) - b_1 u'(b) &= \beta, & |b_0| + |b_1| &\neq 0 \end{aligned} \quad (6)$$

We wish to know the conditions under which problem (5), (6) has a unique solution.

TWO-POINT BOUNDARY VALUE PROBLEM

The function $f(x, u, v)$ is called *uniformly Lipschitz continuous* iff

$$|f(x; u, v) - f(x; w, z)| \leq K \max(|u - w|, |v - z|) \quad (7)$$

where K is some constant, and x, u, w and z are real numbers.

Consider the function $f(x; u, v)$ in (5) and suppose that it is uniformly Lipschitz continuous in the region R , defined by:

$$R: a \leq x \leq b, u^2 + v^2 < \infty. \quad (8)$$

Suppose, furthermore, that f has continuous derivatives in R satisfying, for some constant M ,

$$\frac{\partial f}{\partial u} > 0, \quad \left| \frac{\partial f}{\partial v} \right| \leq M \quad (9)$$

and, that

$$a_0 a_1 \geq 0, \quad b_0 b_1 \geq 0, \quad |a_0| + |b_0| \neq 0. \quad (10)$$

Then the boundary-value problem (5), (6) has a unique solution.

INTRODUCTION

LINEAR BOUNDARY VALUE PROBLEMS

We now consider a special case of (5), namely (1). This is called a **linear equation** and is important in many kinds of applications.

Main equation

All formulas are derived by (11) equation.

$$dS = \mu S dt + \sigma S dz, \quad (11)$$

where μ is the expected instantaneous rate of return on the underlying asset, σ is the instantaneous volatility of the rate of return and dz is a Wiener process.

Listing 1: updateLattice method of BinomialLatticeStrategy .

```
BinomialLatticeStrategy :: BinomialLatticeStrategy(double vol, double interest, double delta)
{
    s = vol;
    r = interest;
    k = delta;
    bType = Multiplicative;
}
```

Listing 2: CRRStrategy method of BinomialLatticeStrategy .

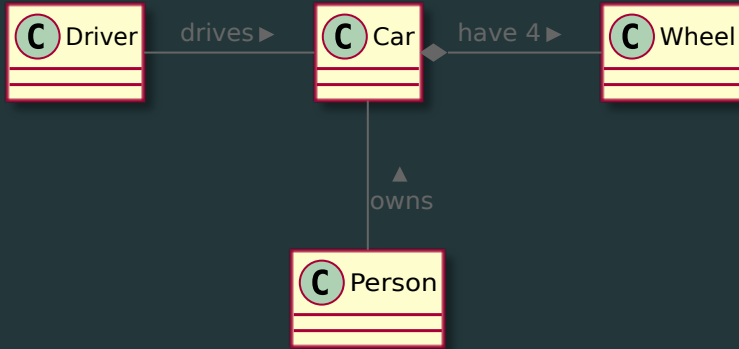
```
CRRStrategy :: CRRStrategy(double vol, double interest, double delta)
    : BinomialLatticeStrategy(vol, interest, delta)
{
    /* double e = :: exp((r)*k);
    double sr = :: sqrt(exp(vol*vol*k) - 1.0);
    u = e * (1.0 + sr);
    d = e * (1.0 - sr);*/

    double R1 = (r - 0.5 * s * s) * k;
    double R2 = s * std :: sqrt(k);

    u = std :: exp(R1 + R2);
    d = std :: exp(R1 - R2);

    std :: cout << "UD[" << u << ", " << d;

    double discounting = std :: exp(- r*k);
```



INTRODUCTION

BLACK-SCHOLES-MERTON: EXACT, LOG-NORMAL DISTRIBUTION

Table 1: First table.

Formula	Description
Samuelson	Precursor of B-S.

$$c = S e^{(\rho-w)T} N(d_1) - X e^{-wT} N(d_2) \quad (\text{Samuelson})$$

where

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(\rho + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}},$$
$$d_2 = d_1 - \sigma\sqrt{T}.$$

where ρ is the average rate of growth of the share price and w is the average rate of growth of the share price and w is the average rate of growth in the value of the call.

BLACK-SCHOLES-MERTON: EXACT, LOG-NORMAL DISTRIBUTION

Table 1: First table.

Formula	Description
Black-Scholes-Merton	Progress in option pricing.

$$\begin{aligned}c &= SN(d_1) - Xe^{-rT} N(d_2) \\p &= Xe^{-rT} N(-d_2) - SN(-d_1)\end{aligned}\tag{B-S}$$

where

$$\begin{aligned}d_1 &= \frac{\ln\left(\frac{S}{X}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}. \\d_2 &= d_1 - \sigma\sqrt{T}.\end{aligned}$$

where S is the stock price, X is the strike price of option, r is the risk-free interest rate, T is the time to expiration in years, σ is the volatility of the relative price change of the underlying stock price and $N(x)$ is the cumulative normal distribution.

INTRODUCTION

TREES: NUMERICAL METHOD

Formula	Description
Binomial trees	

The number of paths leading to a node (j, i) is equal to $\binom{j}{i}$ and the corresponding probability of reaching node (j, i) is $\binom{j}{i} p^i (1-p)^{j-i}$.

$$\begin{aligned} c &= e^{-rT} \sum_{i=0}^n \binom{n}{i} p^i (1-p)^{n-i} \max[Su^i d^{n-i} - X, 0]. \\ p &= e^{-rT} \sum_{i=0}^n \binom{n}{i} p^i (1-p)^{n-i} \max[X - Su^i d^{n-i}, 0]. \end{aligned} \tag{12}$$

Table 2: European Put (exact: 4.0733): $T = 1$, $S = 5$, $K = 10$, $r = 0.12$, $\sigma = 0.5$.

	$N = 256$	$N = 512$	$N = 1024$
CRR	4.07278	4.07309	4.07325
JR	4.07248	4.07324	4.07316
TRG	4.07248	4.07324	4.07316
EQP	4.07266	4.07302	4.0732
Modified CRR	4.07174	4.0725	4.07288
Cayley JR Transform	4.07248	4.07324	4.07316
Cayley CRR	4.07275	4.07308	4.07325

Table 3: European Call (exact: 2.1334): $T = 0.25$, $S = 60$, $K = 65$, $r = 0.08$, $\sigma = 0.30$.

	$N = 256$	$N = 512$	$N = 1024$
CRR	2.13395	2.13464	2.13404
JR	2.13468	2.1319	2.13405
TRG	2.13471	2.13191	2.13406
EQP	2.12669	2.12962	2.13046
Modified CRR	2.12951	2.13144	2.1324
Cayley JR Transform	2.13468	2.1319	2.13405
Cayley CRR	2.13398	2.13466	2.13405

Three different block environments are pre-defined and may be styled with an optional background color.

Default	Default
Block content.	Block content.
Alert	Alert
Block content.	Block content.
Example	Example
Block content.	Block content.

THE FINITE DIFFERENCE METHOD FOR PDEs

THE FINITE DIFFERENCE METHOD FOR PDEs

MATHEMATICAL BACKGROUND

MATHEMATICAL BACKGROUND

In general, the PDEs of relevance are of the *convection-diffusion-reaction* type in n space variables and one time variable. The space variables correspond to underlying financial quantities such as an asset, volatility or interest rate while the non-negative time variable t is bounded above by the expiration T . The space variables are usually defined in their respective positive half-planes.

We model derivatives by *initial boundary value problems* of parabolic type. To this end, consider the general parabolic equation:

$$Lu \equiv \sum_{i,j=1}^n a_{ij}(x,t) \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{j=1}^n b_j \frac{\partial u}{\partial x_j} + c(x,t)u - \frac{\partial u}{\partial t} = f(x,t) \quad (13)$$

where the functions a_{ij} , b_j , c and f are real valued, $a_{ij} = a_{ji}$ and:

$$\sum_{i,j=1}^n a_{ij}(x,t) a_i a_j > 0 \text{ if } \sum_{j=1}^n a_j^2 > 0. \quad (14)$$

MATHEMATICAL BACKGROUND

In equation (13) the variable x is a point in n -dimensional space and t is considered to be a positive time variable. Equation (13) is the general equation that describes the behaviour of many derivative types. For example, in the one-dimensional case ($n = 1$) it reduces to the Black-Scholes equation (here $t^* = T - t$):

$$\frac{\partial V}{\partial t^*} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D) S \frac{\partial V}{\partial S} - rV = 0 \quad (15)$$

where V is the derivative type (for example, a call or put option), S is the underlying asset (or stock), σ is the constant volatility, r is the constant interest rate and D is a constant dividend.

Equation (15) is a one-factor case and it can be generalized, for example to the multivariate case:

$$\frac{\partial V}{\partial t^*} + \sum_{j=1}^n (r - D_j) S_j \frac{\partial V}{\partial S_j} + \frac{1}{2} \sum_{i,j=1}^n \rho_{ij} \sigma_i \sigma_j S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} - rV = 0. \quad (16)$$

This equation models a multi-asset environment. In this case σ_i is the volatility of the i th asset and ρ_{ij} is the correlation ($-1 \leq \rho_{ij} \leq 1$) between assets i and j . The term D_j is the constant dividend for asset j .

MATHEMATICAL BACKGROUND

In this case we see that the elliptic part of equation (16) ($t^* = T - t$, where T is the expiration) is written as the sum of three terms:

- Interest earned on cash position

$$r \left(V - \sum_{j=1}^n S_j \frac{\partial V}{\partial S_j} \right).$$

- Gain from dividend yield:

$$\sum_{j=1}^n D_j S_j \frac{\partial V}{\partial S_j}.$$

- Hedging costs or slippage:

$$-\frac{1}{2} \sum_{i,j=1}^n \rho_{ij} \sigma_i \sigma_j S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j}.$$

Our interest is in discovering robust numerical schemes that produce reliable and accurate results irrespective of the size of the parameter values in equation (16).

Equation (13) has an infinite number of solutions in general. In order to define a unique solution, we need to define some constraints. To this end, we define *initial condition* and *boundary conditions* for (13). We achieve this by defining the space in which equation (13) is assumed to be valid. In general, we denote that there are three types of boundary conditions associated with equation (13). These are:

- First boundary value problem (*Dirichlet problem*).
- Second boundary value problem (*Neumann, Robin problems*).
- Cauchy problem.

The first boundary value problem is concerned with the solution of equation (13) in a bounded domain $D = \Omega \times (0, T)$, where Ω is a bounded subset of \mathbb{R}^n and T is a positive number. In this case we seek a solution of equation (13) satisfying the conditions:

$$\begin{aligned} u|_{t=0} &= \phi(x) && \text{(initial condition)} \\ u|_{\Gamma} &= \Psi(x, t) && \text{(boundary condition)} \end{aligned} \tag{17}$$

where Γ is the boundary of Ω .

The boundary conditions in (17) are called *Dirichlet boundary conditions* because the solution is given on the boundary. These conditions arise when we model single and double barrier options in the one-factor case. They also occur when we model plain options on a transformed or truncated domain.

The second boundary value problem is similar to (17) except instead of giving the value of u on the boundary Γ the *directional derivatives* are included, as seen in the following example:

$$\left(\frac{\partial u}{\partial \eta} + a(x, t) u \right) = \Psi(x, t), \quad x \in \Gamma. \quad (18)$$

In the $a(x, t)$ and $\Psi(x, t)$ are known functions of x and t , and $\frac{\partial}{\partial \eta}$ denotes the derivative of u with respect to the outward normal η at Γ . A special case of (18) is when $a(x, t) \equiv 0$; then (18) represents *Neumann boundary conditions*. Finally, the solution of the Cauchy problem for (13) in the infinite region $\mathbb{R}^n \times (0, T)$ is given by the initial condition:

$$u|_{t=0} = \varphi(x) \quad (19)$$

where $\varphi(x)$ is a given continuous function and $u(x, t)$ satisfies equation (13) in $\mathbb{R}^n \times (0, T)$ and the initial condition (19).

MATHEMATICAL BACKGROUND

This problem allows negative values of the components of the independent variable (x_1, \dots, x_n) . A special case of the Cauchy problem can be seen when modeling one-factor European and American options where x plays the role of underlying asset S . Boundary conditions are given by values at $S = 0$ and at $S = \infty$. For European call options these conditions are:

$$\begin{aligned} C(0, t) &= 0 \\ C(S, t) &\rightarrow S \text{ as } S \rightarrow \infty. \end{aligned} \tag{20}$$

Here C (the role played by u in equation (13)) is the price of the call option. For European put options the boundary conditions are:

$$\begin{aligned} P(0, t) &= Ke^{-rt} \\ P(S, t) &\rightarrow 0 \text{ as } S \rightarrow \infty. \end{aligned} \tag{21}$$

Here P (the role played by u in equation (13)) is the variable representing the price of the put option, K is the strike price, r is the risk-free interest rate, T is the expiration and t is the current time.

We sometimes assume the following ‘*canonical form*’ for the operator L in equation (13) in the one-factor case:

$$Lu \equiv -\frac{\partial u}{\partial t} + \sigma(x, t) \frac{\partial^2 u}{\partial x^2} + \mu(x, t) \frac{\partial u}{\partial x} + b(x, t) u = f(x, t) \quad (22)$$

where σ , μ , b and f are known functions of x and t .

For completeness, we formulate the one-factor initial boundary value problem whose solution we wish to approximate using the finite difference method. To this end, we define the interval $\Omega = (A, B)$ where A and B are two real numbers with $A < B$. Further, let $T > 0$ and $D = \Omega \times (0, T)$. The statement is:

Find a function $u: D \rightarrow \mathbb{R}$ such that:

$$Lu = \frac{\partial u}{\partial t} + \sigma(x, t) \frac{\partial^2 u}{\partial x^2} + \mu(x, t) \frac{\partial u}{\partial x} + b(x, t) u = f(x, t) \text{ in } D \quad (23)$$

$$u(x, 0) = \varphi(x), \quad x \in \Omega \quad (24)$$

$$u(A, t) = g_0(t), \quad u(B, t) = g_1(t), \quad t \in (0, T). \quad (25)$$

The initial boundary value problem (23)–(25) subsumes many specific cases (in particular it is a generalization of the Black-Scholes equation).

In general, the coefficients $\sigma(x, t)$ and $u(x, t)$ represent *volatility* (diffusivity) and *drift* (convection), respectively. Equation (23) is called a *convection-diffusion-reaction* equation. It serves as a model for many kinds of problems.

Much research has been carried out in this area, both on the continuous problem and its discrete formulations (for example, using finite difference and finite element methods).

The essence of the finite difference method is to discretise equations (23)–(25) by defining *discrete mesh points* and approximating the derivatives of the unknown solution of this system at these mesh points.

The goal is to find accurate schemes that will be implemented in a programming language as C++ and C#. Some typical attention points are:

- The PDE being approximated may need to be preprocessed in some way, for example transforming it from one on a semi-infinite domain to one on a bounded domain.
- Determining which specific finite difference schemes(s) to use based on requirements such as accuracy, efficiency and maintainability.
- Essential difficulties to resolve: *convection dominance*, avoiding oscillations and how to handle discontinuous initial conditions, for example.
- Developing the algorithms and assembling the discrete system of equations prior to implementation.

PDE PREPROCESSING

PDE PREPROCESSING

A PDE is defined in a region of two-dimensional space defined by the variables x (the underlying space) and time t . Regarding the x variable, the range of values can be:

Infinite domain $-\infty < x < \infty$.

Semi-infinite domain $0 < x < \infty$.

Bounded domain $A < x < B$, where $-\infty < A < B < \infty$.

When approximating PDEs by FDM the underlying domain must be transformed to a bounded domain. Thus, we can decide to modify the PDE in some way before approximating it by the FDM:

- Change the coefficients of the PDE.
- Truncate an infinite or semi-infinite domain to a bounded domain.
- Transform an infinite or semi-infinite domain to a bounded domain by a change of independent variables.
- Change the structure of a PDE (for example, we can use an *integrating factor* to transform a non-conservative PDE to a conservative PDE).

It is important to determine what the consequences are when modifying a PDE. We now discuss each of the above topics.

PDE PREPROCESSING

LOG TRANSFORMATION

LOG TRANSFORMATION

This is a popular method to transform the Black-Scholes PDE on a semi-infinite interval to a PDE with constant coefficients on a infinite interval. For example, we define $y = \log S$ and we use this new variable in equation (15) to produce a PDE with constant coefficients:

$$\frac{\partial V}{\partial t^*} + \frac{1}{2}\sigma^2 \frac{\partial v}{\partial y} + v \frac{\partial V}{\partial y} - rV = 0 \text{ where } v = r - D - \frac{1}{2}\sigma^2 \quad (26)$$

which may be more computationally attractive in certain cases. Furthermore, we can remove the explicit dependence on the convection term by writing equation (26) in *conservative form*:

$$\frac{\partial V}{\partial t^*} + A \frac{\partial}{\partial y} \left(B \frac{\partial V}{\partial y} \right) - rV = 0 \text{ where } A = \frac{1}{2}\sigma^2 e^{cy}, B = e^{-cy}, c = \frac{2v}{\sigma^2}. \quad (27)$$

SOFTWARE FRAMEWORK FOR ONE-FACTOR OPTION MODELS

SOFTWARE FRAMEWORK FOR ONE-FACTOR OPTION MODELS

- To create a seamless mapping from FD algorithms to C++, thus helping us to create code that is built to be maintained and extended. We achieve this end by using decomposition methods and design patterns.
- To experiment with a range of FD schemes (some of which are new in finance) and determine how stable and accurate they are.
- Having determined that a scheme is to our liking, to optimize the code by modifying the design or the code in some way in order to improve the performance.

We examine *linear convection-diffusion-reaction PDEs* of non-conservative type and their approximation by finite difference schemes.

We implement a number of *one-step schemes* in this chapter; they all share the property that they compute a solution at time level $n + 1$ in terms of the solution at time level n . Furthermore, some schemes are implicit (for example, S2, S4 and S5 below) which demands our having to solve a *tridiagonal system* at each time level while others are explicit (S1, S3) and then the solution at time level $n + 1$ is computed directly in terms of the solution at time level n without the need to solve a matrix system. The schemes are as follows.

- S1 Explicit Euler (FTCS) scheme.
- S2 Fully implicit (BTCS) scheme.
- S3 ADE (several variants based on how we approximate the diffusion, convection and reaction terms in the PDE).
- S4 Crank-Nicolson.
- S5 Richardson extrapolation applied to scheme S2.

The use of finite difference methods in finance was first described by Brennan and Schwartz (1978). Finite difference methods, also called grid models, are simply a numerical technique to solve partial differential equations (PDE). Different finite difference methods can be used to price European and American options, as well as many types of exotic options. Finite difference models are as we soon will see very similar to tree models. The finite difference methods is basically a numerical approximation of the PDE. here we will give an overview of the three most common finite difference techniques in option pricing:

- Explicit finite difference
- Implicit finite difference
- Crank-Nicolson finite difference

SOFTWARE FRAMEWORK FOR ONE-FACTOR OPTION MODELS

All the finite difference models described here build on the same main principle. First we build a grid with time along one dimension/axis and price along the other dimension/axis. Just as in a tree model, the time and price movements are discretized. Time increases in increments of Δt , while the asset changes in amounts of ΔS . These increments are then used to construct a grid of possible combinations of time and asset price levels. The finite difference technique is then used to approximately solve the relevant PDE on this grid. Just as in a tree model one starts at the “end” of the grid, at time T , and rolls back through the grid. The calculations done on the grid are a bit different, however.

The finite difference models can be used to solve a large class of PDEs, and thereby a large class of options. If we assume that the underlying asset follows a geometric Brownian motion, we get the following Black-Scholes-Merton PDE for any single asset derivatives

$$\frac{\partial f}{\partial x} + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 + b \frac{\partial f}{\partial S} S = rf, \quad (28)$$

where f is the value of a derivative security for example a European call c or an American call C or some type of exotic option. We want to solve this PDE along the grid for the particular derivative instrument under consideration. How this is done depends on the chosen finite difference technique, as well as the derivative's contractual details. We start with the explicit finite difference method.

SOFTWARE FRAMEWORK FOR ONE-FACTOR OPTION MODELS

FINITE DIFFERENCE METHOD: NUMERICAL METHOD

Formula	Description
Explicit Euler	

$$\frac{\partial f}{\partial t} \approx \frac{f_{j+1,i} - f_{j,i}}{\Delta t},$$

where $f_{j,i}$ is the value of the derivative instrument at time step j and price level i . The delta, $\frac{\partial f}{\partial S}$, and the gamma, $\frac{\partial^2 f}{\partial S^2}$, are approximated by central differences:

$$\frac{\partial f}{\partial S} \approx \frac{f_{j+1,i+1} - f_{j+1,i-1}}{2\Delta S} \quad \frac{\partial^2 f}{\partial S^2} \approx \frac{f_{j+1,i+1} - 2f_{j+1,i} + f_{j+1,i-1}}{\Delta S^2}.$$

Replacing the partial derivatives in (28) with these approximations, we get

$$\frac{f_{j+1,i} - f_{j,i}}{\Delta t} + \frac{1}{2} \frac{f_{j+1,i+1} - 2f_{j+1,i} + f_{j+1,i-1}}{\Delta S^2} \sigma^2 S^2 + b \frac{f_{j+1,i+1} - f_{j+1,i-1}}{2\Delta S} = r f_{j,i},$$

which can be rewritten as

$$f_{j,i} = \frac{1}{1 + r\Delta t} (p_u f_{j+1,i+1} + p_m f_{j+1,i} + p_d f_{j+1,i-1}),$$

where

Formula	Description
Implicit Euler	

$$\frac{f_{j+1,i} - f_{j,i}}{\Delta t} + \frac{1}{2} \frac{f_{j,i+1} - 2f_{j,i} f_{j,i-1}}{\Delta S^2} \sigma^2 S + b \frac{f_{j,i+1} - f_{j,i-1}}{2\Delta S} S = r f_{j,i},$$

which can be rewritten as

$$f_{j+1,i} = p_u f_{j,i+1} + p_m f_{j,i} + p_d f_{j,i-1}$$

$$p_u = \frac{1}{2} i (b + -v^2 i) \Delta t$$

$$p_m = 1 + (r + v^2 i^2) \Delta t$$

$$p_d = \frac{1}{2} i (-b - v^2 i) \Delta t$$

If we use M as the number of price steps in the grid, we now need to solve for $M - 1$ unknown derivatives values, $f_{j,i}$ on the grid simultaneously

FINITE DIFFERENCE METHOD: NUMERICAL METHOD

Formula	Description
Crank-Nicholson	

In this method the approximation of the PDE equation is done by central differences at time step $j + \frac{1}{2}$ instead of $j + 1$ as in the explicit finite difference method, or at point j as in the implicit finite difference method.

$$-\frac{f_{j+1,i} - f_{j,i}}{\Delta t} = \frac{1}{2}\sigma^2 \frac{(f_{j,i+1} - 2f_{j,i} f_{j,i-1}) + (f_{j+1,i+1} - 2f_{j+1,i} f_{j+1,i-1})}{2\Delta x^2} + \left(b - \frac{\sigma^2}{2}\right) \frac{(f_{j+1,i+1} - f_{j+1,i-1}) + (f_{j,i+1} - f_{j,i-1})}{4\Delta x} - r \left(\frac{f_{j+1,i} + f_{j,i}}{2}\right), \quad (29)$$

As we can see, the Crank-Nicolson method is combination of the explicit and implicit methods. It is more efficient than the others. In combination with the same boundary conditions as in the implicit finite difference method, the Crank-Nicolson method will make up a tridiagonal system of equations.

SOFTWARE FRAMEWORK FOR ONE-FACTOR OPTION MODELS

Monte Carlo Simulation: NUMERICAL METHOD

Formula	Description
Monte Carlo	

$$S + dS = S \exp \left[\left(\mu - \frac{1}{2} \sigma^2 \right) dt + \sigma dz \right],$$

where dz is a Wiener process with standard deviation 1 and mean 0. To simulate the process, we consider its values at given discrete time intervals, Δt apart:

$$S + \Delta S = S \exp \left[\left(\mu - \frac{1}{2} \sigma^2 \right) \Delta t + \sigma \epsilon_t \sqrt{\Delta t} \right],$$

where ΔS is the change in S in the chosen time interval Δt , and ϵ_t is a random drawing from a standard normal distribution. The main drawback of Monte Carlo simulation is that it is computer-intensive. A minimum of 10000 simulations are typically necessary to price an option with satisfactory accuracy. The standard error in the estimated value from the standard Monte Carlo simulation is normally related to the square root of the number of simulations. More precisely, if s is the standard deviation of the payoffs from n simulations, then the standard error is given by $\frac{s}{\sqrt{n}}$. This means that to double of accuracy, we will need to quadruple the number of simulations.

- Books



Carlo Mariconda and Alberto Tonolo. *Discrete calculus. Methods for counting*. Vol. 103. UNITEXT - La Matematica per il 3+2. Springer International Publishing, 2016. ISBN: 978-3-319-03037-1. DOI: [10.1007/978-3-319-03038-8](https://doi.org/10.1007/978-3-319-03038-8).



Tom Jenkyns and Ben Stephenson. *Fundamentals of Discrete Math for Computer Science: A Problem-Solving Primer*. Second. Undergraduate Topics in Computer Science. Springer International Publishing AG, part of Springer Nature, 2018. ISBN: 978-3-319-70150-9.

REFERENCES

- Articles



Stefan Hilger. “Analysis on Measure Chains — A Unified Approach to Continuous and Discrete Calculus”. In: *Results in Mathematics* 18.1 (Aug. 1990), pp. 18–56. ISSN: 1420-9012.

- Website



Purificación Nadal Morales. *Matemáticas II: Grado en economía. Lección 5: Ecuaciones en diferencias*. 2014. URL: <http://personal.us.es/pnadal/Informacion/leccion5ecdiferencias.pdf> (visited on 06/10/2019).



Tom Cuchta and Matthias Baur. *Timescalecalculus: Python library*. 2019. URL: http://timescalewiki.org/index.php/Timescalecalculus_python_library_documentation (visited on 05/30/2019).

$$x=\begin{cases}x&1,\\x&2,\\x&2\\x&\text{otherwise}\end{cases}\qquad \lim_{n\rightarrow +\infty}\sqrt{\frac{\int_{-\epsilon}^{\epsilon}\cos^n x}{\int_{-\pi/2}^{\pi/2}\cos^n x}}=1$$

$$\begin{pmatrix}a&b\\c&d\\e&f\\g&h\end{pmatrix}$$

$$\left\{ \begin{array}{l} 1, \quad 1 \\ 2, \quad 2 \\ 3, \quad 3 \\ 4, \quad 4 \end{array} \right\}$$

$$\left\{ \begin{array}{l} 1, \quad 1 \\ 2, \quad 2 \\ 3, \quad 3 \\ 4, \quad 4 \end{array} \right\}$$

$$\left\{ \begin{array}{l} 1 \quad 1 \quad 1 \\ 2 \quad 2 \quad 2 \\ 3 \quad 3 \quad 3 \\ 4 \quad 4 \quad 4 \\ 5 \quad 5 \quad 5 \end{array} \right\}$$

$$\left\{ \begin{array}{l} 1 \quad 1 \quad 1 \\ 2 \quad 2 \quad 2 \\ 3 \quad 3 \quad 3 \\ 4 \quad 4 \quad 4 \end{array} \right\}$$