Technical report, IDE1114 , September 29, 2011

Finite Volume Methods for Option Pricing

Master's Thesis in Financial Mathematics

Mikhail Demin

School of Information Science, Computer and Electrical Engineering
Halmstad University

Finite Volume Methods for Option Pricing

Mikhail Demin

Halmstad University
Project Report IDE1114

Master's thesis in Financial Mathematics, 15 ECTS credits

Supervisor: Prof. Dr. Matthias Ehrhardt
Examiner: Prof. Dr. Ljudmila A. Bordag
External referees: Prof. Mikhail Babich

September 29, 2011

Department of Mathematics, Physics and Electrical engineering
School of Information Science, Computer and Electrical Engineering
Halmstad University

Preface

Firstly, I would like to thank Prof. Dr. Ljudmila A. Bordag who gave me an opportunity to study on the program of Financial Mathematics in Halmstad. I also would like to thank my supervisor Prof. Dr. Matthias Ehrhardt for his help and useful remarks and comments in writing the thesis. And of course I would like to thank my family for their love and support during the whole period of study at the university.

Abstract

There are three important reasons why stock options are better than just trading stocks or futures: they are leverage, protection and flexibility. As stock options cost only a small fraction of the price of the underlying stock while representing the same amount of shares, it allows anyone to control the profits on the same amount of shares with a much smaller amount of money. Moreover, buying stock options is like buying an insurance. They allow traders to hedge away the directional risk anytime they want. Also, options give traders the flexibility to change from one market opinion to another without significantly changing current holdings. That is why, it is so important to estimate the current value of the option or to predict the future value.

In this thesis we present an implicit method for solving nonlinear partial differential equations. We construct a fitted finite volume scheme. The use of the implicit method leads to the convergence in fewer time steps compared to explicit schemes. But every step is much more costly. We also consider and present a classical finite difference scheme and a stable box scheme of Keller. The results obtained by using box scheme are compared with results of applying fitted finite volume scheme and Crank–Nicolson scheme.

Contents

Chapter 1

Introduction

Since the early 70s, *options* have become one of the most important derivative instruments besides futures. They are simple contracts that allow the owners to buy or to sell an underlying stock at a specific price before the *expiration time* or *maturity time* (the time after which options become worthless). That specific price at which stocks are traded is called the *strike* or *exercise price*. Thus, the trading of underlyings at the exercise price is an *exercising* of option. This simplicity has made them one of the most speculative and hedging instrument ever created, elevating options trading to its current level of importance.

There exist two types of options: the *call option*, that gives the holder the right to buy the underlying stock for the strike price by the maturity date, and the *put option*, that gives the holder the right to sell the underlying stock for the strike price by the maturity date.

By the dates on which options may be exercised they are divided into two styles: they are, in most cases, either European or American. These options are referred to "*vanilla options*" (commonly traded options with similarly calculated value of the option at a final condition called payoff). The main difference between American and European options is in different exercising time: an European option may be exercised only at the maturity date, however an American option may be exercised at any time before the maturity date.

Options where the payoff is calculated differently are referred to "*exotic options*". Although these options are more unusual than vanilla they can also be of two styles: European or American. These are few of them: a *lookback option* (a path–dependent option where the holder of an option has the right to buy or to sell the underlying stock at its lowest or highest price), an *Asian* or *average option* (an option where the payoff is determined by the average underlying price over some period of time), a barrier option (an option that becomes valuable or worthless if the price of underlying reaches a predetermined level called barrier) and others.

Let us consider some examples of options. Everywhere below in the examples we denote S as the price of the underlying asset, K as the strike or exercise price, T as the maturity date.

- **European vanilla options**
  Payoff for a call: $C(S,T) = max(S-K,0)$,
  payoff for a put: $P(S,T) = max(K-S,0)$.

- **European exotic options**
  Payoff for a lookback call (average strike): $C(S,R,T) = max(S-R,0)$,
  payoff for a lookback put (average strike): $P(S,R,T) = max(R-S,0)$,
  where $R = \int\limits_0^t S(\tau)d\tau$.

Options traders who hold stock option contracts usually try to estimate the future price of options. It can be estimated by using a variety of quantitative techniques based on the concept of risk neutral pricing and using stochastic calculus. It was shown by Fischer Black and Myron Scholes that the value of an option can be modeled by a second order backward–in–time parabolic partial differential equation (PDE) and the price of its underlying stock.

There exist several methods in the literature for the valuation of options (finite difference method or finite element method, for instance). The finite volume method is a discretization method which is well suited for the numerical simulation of various types of conservation laws. It has been extensively used in several engineering fields, such as fluid mechanics, heat and mass transfer. This method was also used for option pricing in finance [1]. It is also very suitable to solve the standard Black–Scholes equation, since it can overcome the difficulty caused by the drift–dominated phenomena.

The finite volume method was firstly used to price the standard European options. Zhang and Wang [1] showed that the finite volume scheme in space with the implicit scheme in time is consistent and stable, hence it converges to the financial relevant solution. Moreover, an efficient iterative method was proposed and the convergence property of this iterative method was proved.

The main goal of this thesis is to present the finite volume method and to compare the numerical results obtained by applying that method on Black–Scholes equation with the results obtained by other difference schemes.

This thesis is organized as follows. In this chapter the background is given. In Chapter 2 the discretization in time and space of the finite volume method is presented and the theorems about the consistency and stability of the method are presented. Chapter 3 is about a box scheme of Keller and its discretization in time and space. The description of a classical finite difference scheme can be found in Chapter 4. In this Chapter a finding of the solution of three–points–compact finite difference scheme with optimal choice of coefficients is also considered. All results of numerical experiments are presented in Chapter 5 and discussed and interpreted in Chapter 6. Finally, proofs of theoretical results and the MATLAB program code can be found in Appendix.

Chapter 2

The Finite Volume Discretization

The Black–Scholes model of the market for a particular equity makes the assumption that the stock price follows a *"geometric Brownian Motion"* (GBM) with constant drift and volatility. It means that the volatility for a particular stock would be the same for all fixed prices called strikes. But in practice, the volatility curve (the two-dimensional graph of volatility implied by the market price of the option based on an option pricing model against strike) is not straight and depends on the underlying instrument.

The Black–Scholes equation, derived by Black and Scholes and published in 1973, is a linear parabolic partial differential equation (PDE). It is also valid if the volatility depends on time

$$\sigma = \sigma(S, \tau).$$

In this case the PDE used to evaluate the price of the option contract is determined by

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sigma^2(S,\tau)S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV, \tag{2.1}$$

where $V = V(S,\tau)$ is the price of an option, and we applied the time reversal $\tau = T - t \in [0,T]$. S is the price of the underlying asset, t is the current time, T is the expiring date (or maturity time) and r is the risk-free interest rate. The boundary conditions that show us the behavior of the solution for all time at certain values of the asset are the following. Equation (2.1) reduces to

$$\frac{\partial V}{\partial \tau} = -rV,$$

as $S = 0$. For $S \to \infty$ we have a Dirichlet condition

$$V(S,\tau) = f(S,\tau), \tag{2.2}$$

where $f(S,\tau)$ can be determined by financial reasoning. The appropriate Dirichlet condition depends on the payoff of the option and on the assumptions about the stochastic process followed by the underlying asset. Let us consider as an example an European Call option, for which the PDE is supplied with the following boundary conditions

$$\begin{array}{l} V(S,\tau) = 0,\ S = 0,\ \forall \tau \in [0,T], \\ V(S,\tau) \to S - Ke^{-r\tau},\ S \to \infty. \end{array}$$

Or for the European Average Strike option we have

$$\begin{array}{l} V(S,\tau) = 0,\ S = 0,\ \forall \tau \in [0,T], \\ V(S,\tau) \to R,\ S \to \infty, \end{array}$$

where $R = \int\limits_0^\tau S(z)dz$.

The asset region $(0,\infty)$ we change to $I = (0, S_{max})$ for computational convenience, where $S_{max} \to \infty$, by truncating the domain and imposing the condition (2.2) at $S = S_{max}$.

Let us turn to the spatial discretization of equation (2.1).

2.1 The Spatial Discretization

The first step in the spatial discretization is to consider the Black-Scholes equation:

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV,\ \tau \in [0,T], \tag{2.3}$$

and apply a finite volume discretization strategy.

As the asset region $(0,\infty)$ is changed to $I = (0, S_{max})$ for computational convenience, it can be solved by numerical techniques. Now we can apply the finite volume discretization strategy to equation (2.3), written in the conservative form

$$\frac{\partial V}{\partial \tau} = \frac{\partial}{\partial S}\left(\frac{1}{2}\sigma^2 S^2 \frac{\partial V}{\partial S} + rSV - \sigma^2 SV\right) - 2rV + \sigma^2 V =$$

$$= \frac{\partial}{\partial S}\left(aS^2 \frac{\partial V}{\partial S} + bSV\right) - cV, \tag{2.4}$$

where $a = \frac{\sigma^2}{2}$, $b = r - \sigma^2$, $c = r + b$.

Next step is to divide I into two parts. The first part consists of N sub-intervals $I_i = (S_i, S_{i+1})$, $i = 0, \ldots, N-1$ with $0 = S_0 < S_1 < \ldots < S_N = S_{max}$. The second part is $J_i = (S_{i-1/2}, S_{i+1/2})$, where $S_{i-1/2} = \frac{S_{i-1}+S_i}{2}$, $S_{i+1/2} = \frac{S_i+S_{i+1}}{2}$ with $S_{-1/2} = S_0$ and $S_{N+1/2} = S_{N+1}$.

Integrating (2.4) over J_i

$$\int_{J_i} \frac{\partial V}{\partial \tau} dS = S\left(aS\frac{\partial V}{\partial S} + bV\right)\Bigg|_{S_{i-1/2}}^{S_{i+1/2}} - \int_{J_i} cV dS =$$

$$= \int_{J_i} \frac{\partial V}{\partial \tau} dS = S\rho(V)|_{S_{i-1/2}}^{S_{i+1/2}} - \int_{J_i} cV dS, \tag{2.5}$$

and applying the one–point quadrature rule to all terms in that equation except the first one in the right–hand side, we get

$$\frac{\partial V(S_i,\tau)}{\partial \tau}\ell_i = S_{i+1/2}\rho(V)|_{S_{i+1/2}} - S_{i-1/2}\rho(V)|_{S_{i-1/2}} - c\ell_i V(S_i,\tau), \tag{2.6}$$

for each $i = 1, \ldots, N-1$, where $\ell_i = S_{i+1/2} - S_{i-1/2}$ is the length of J_i.

Here the flux $\rho(V)$ associated with V is defined as $\rho(V) = aSV' + bV$.

Let us derive now an approximation of the flux $\rho(V)$ at the points $S_{i+1/2}$ and $S_{i-1/2}$ of the interval I_i for all $i = 2, \ldots, N-1$. Considering the two–point boundary value problem

$$\begin{cases} (a_{i+1/2}SV' + b_{i+1/2}V)' = 0, & S \in I_i, \\ V(S_i) = V_i, & V(S_{i+1}) = V_{i+1} \end{cases} \tag{2.7}$$

at the point $S_{i+1/2}$, and

$$\begin{cases} (a_{i-1/2}SV' + b_{i-1/2}V)' = 0, & S \in I_i, \\ V(S_i) = V_i, & V(S_{i-1}) = V_{i-1} \end{cases} \tag{2.8}$$

at the point $S_{i-1/2}$ and solving that equations analytically, we get

$$\rho_i(V) = b_{i+1/2}\frac{V_i S_i^{k_i} - V_{i+1}S_{i+1}^{k_i}}{S_i^{k_i} - S_{i+1}^{k_i}} \tag{2.9}$$

at $S_{i+1/2}$, and

$$\rho_i(V) = b_{i-1/2}\frac{V_i S_i^{k_i} - V_{i-1}S_{i-1}^{k_i}}{S_i^{k_i} - S_{i-1}^{k_i}} \tag{2.10}$$

at $S_{i-1/2}$, where $k_i = \frac{b_{i+1/2}}{a_{i+1/2}}$ for all $i = 2, \ldots, N-1$.

The equation (2.5) is degenerated for $S \to 0$ when the flux $\rho(v)$ is defined on the interval $I_0 = (0, S_1)$. In that case, we can consider this equation in the following form

$$\begin{cases} (a_{1/2}SV' + b_{1/2}V)' = C, \quad S \in I_0, \\ V(0) = V_0, \qquad V(S_1) = V_1. \end{cases} \tag{2.11}$$

Solving these equations, we get

$$\begin{aligned} &\rho_0(V) = \tfrac{1}{2}((a_{1/2} + b_{1/2})V_1 - (a_{1/2} - b_{1/2})V_0), \\ &V = V_0 + (V_1 - V_0)\tfrac{S}{S_1}. \end{aligned} \tag{2.12}$$

Substituting (2.9), (2.10) and (2.12) into (2.6), we obtain

$$\frac{\partial V(S_i, \tau)}{\partial \tau} = \xi_i V(S_{i-1}, \tau) + \psi_i V(S_i, \tau) + \varphi_i V(S_{i+1}, \tau), \tag{2.13}$$

for $i = 1, \ldots, N-1$, where

$$\begin{aligned} \xi_1 &= \tfrac{S_1(a_{1/2} - b_{1/2})}{4\ell_1}, \\ \varphi_1 &= \tfrac{b_{3/2}S_{3/2}S_2^{k_1}}{\ell_1(S_2^{k_1} - S_1^{k_1})}, \\ \psi_1 &= -\tfrac{S_1(a_{1/2} + b_{1/2})}{4\ell_1} - \tfrac{b_{3/2}S_{3/2}S_1^{k_1}}{\ell_1(S_2^{k_1} - S_1^{k_1})} - c, \end{aligned} \tag{2.14}$$

and

$$\begin{aligned} \xi_i &= \tfrac{b_{i-1/2}S_{i-1/2}S_{i-1}^{k_i}}{\ell_i(S_i^{k_i} - S_{i-1}^{k_i})}, \\ \varphi_i &= \tfrac{b_{i+1/2}S_{i+1/2}S_{i+1}^{k_i}}{\ell_i(S_{i+1}^{k_i} - S_i^{k_i})}, \\ \psi_i &= -\tfrac{b_{i-1/2}S_{i-1/2}S_i^{k_i}}{\ell_i(S_i^{k_i} - S_{i-1}^{k_i})} - \tfrac{b_{i+1/2}S_{i+1/2}S_i^{k_i}}{\ell_i(S_{i+1}^{k_i} - S_i^{k_i})} - c, \end{aligned} \tag{2.15}$$

for $i = 2, \ldots, N-1$. As $\xi_i = \xi_i(\sigma), \varphi_i = \varphi_i(\sigma)$ and $\psi_i(\sigma)$, then the semi-discretization form of (2.3) is the following

$$\frac{\partial V(S_i, \tau)}{\partial \tau} = \xi_i(\sigma) V(S_{i-1}, \tau) + \psi_i(\sigma) V(S_i, \tau) + \varphi_i(\sigma) V(S_{i+1}, \tau), \tag{2.16}$$

or, equivalently

$$\frac{\partial V_i}{\partial \tau} = \xi_i(\sigma)V_{i-1} + \psi_i(\sigma)V_i + \varphi_i(\sigma)V_{i+1}, \tag{2.17}$$

for $i = 1, \ldots, N-1$, where

$$\Gamma_i = \frac{V_{i-1}(S_i - S_{i+1}) + V_i(S_{i+1} - S_{i-1}) + V_{i+1}(S_{i-1} - S_i)}{(S_i - S_{i-1})(S_{i+1/2} - S_{i-1/2})(S_{i+1} - S_i)}. \tag{2.18}$$

2.2 The Time Discretization

Now let us consider the time discretization of (2.17). Let τ_i denote points from $[0,T]$ such that $0=\tau_0<\tau_1<\ldots<\tau_M=T$ and $\Delta\tau_n=\tau_n-\tau_{n-1}\geq 0$, where $M>1$ is a positive integer. Applying the fully implicit time discretization to (2.17) for simplicity, we get

$$\frac{V_i^{n+1}-V_i^n}{\Delta\tau_{n+1}}=\xi_i^{n+1}(\sigma^{n+1})V_{i-1}^{n+1}+\psi_i^{n+1}(\sigma^{n+1})V_i^{n+1}+\varphi_i^{n+1}(\sigma^{n+1})V_{i+1}^{n+1}, \quad (2.19)$$

or

$$\begin{aligned} V_i^n=(1-\Delta\tau_{n+1}\psi_i^{n+1}(\sigma^{n+1}))V_i^{n+1}-\Delta\tau_{n+1}\xi_i^{n+1}(\sigma^{n+1})V_{i-1}^{n+1}-\\ -\Delta\tau_{n+1}\varphi_i^{n+1}(\sigma^{n+1})V_{i+1}^{n+1}, \end{aligned} \quad (2.20)$$

where V_i^n is the solution at node S_i and time τ_n.

The matrix form of (2.20) is

$$V^n=(I-\Delta\tau_{n+1}M^{n+1})V^{n+1}-\Delta\tau_{n+1}R^n, \quad (2.21)$$

where

$$\begin{aligned} V^n&=\left[V_1^n,\ldots,V_{N-1}^n\right]^T,\\ R^n&=\left[\xi_1^{n+1}V_0^{n+1},0,\ldots,0,\varphi_{N-1}^{n+1}V_N^{n+1}\right]_{N-1}^T, \end{aligned} \quad (2.22)$$

and

$$M^{n+1}=M(\sigma^{n+1})=\begin{bmatrix} \psi_1^{n+1} & \varphi_1^{n+1} & & \\ \xi_2^{n+1} & \psi_2^{n+1} & \varphi_2^{n+1} & \\ & \ddots & \ddots & \ddots \\ & & \xi_{N-1}^{n+1} & \psi_{N-1}^{n+1} \end{bmatrix}_{(N-1)\times(N-1)}. \quad (2.23)$$

Equation (2.21) is the full discretization of (2.3) in the matrix form.

2.3 Consistency and Stability

Firstly, let us introduce a measure of the accuracy of the discrete approximation that is called a truncation error.

Definition 1 *The truncation error is the error resulting from the approximation of a derivative or differential by a finite difference.*

Now let us consider the consistency and the stability of the discretization (2.19).

Theorem 1 *(6) The discretization (2.19) is consistent, i.e. the truncation error (TE) tends to zero uniformly with the mesh size $h = S_{i+1} - S_i$.*

Theorem 2 *(6) The discretization (2.19) is stable, if*

$$|V_j^n| = \left| \frac{1}{1 - \Delta\tau_{n+1}(\xi_j e^{-ikh} + \varphi_j e^{ikh} + \psi_j)} \right| \leq 1,$$

where $h = \Delta S$.

All the proves of the theorems can be found in Appendix.

Chapter 3

The Box Scheme of Keller

Let us consider the general case of the Black–Scholes equation

$$\frac{\partial V}{\partial \tau} = \sigma \frac{\partial^2 V}{\partial S^2} + \mu \frac{\partial V}{\partial S} + bV - f, \ \tau \in [0, T], \tag{3.1}$$

with the initial condition

$$V(S, 0) = g(S), \tag{3.2}$$

and supplied with Robin–type boundary conditions

$$\begin{aligned} &\alpha_0 V(0,\tau) + \alpha_1 \sigma \tfrac{\partial V}{\partial S}(0,\tau) = g_0(\tau), \\ &\beta_0 V(S_{max},\tau) + \beta_1 \sigma \tfrac{\partial V}{\partial S}(S_{max},\tau) = g_1(\tau), \end{aligned} \tag{3.3}$$

where $(S, \tau) \in (0, S_{max}) \times (0, T)$.

Let us rewrite (3.1) as a system of first–order equations in the following form

$$\sigma \frac{\partial V}{\partial \tau} = \sigma \frac{\partial \delta}{\partial S} + \mu\delta + \sigma b V - \sigma f, \tag{3.4}$$

setting $\sigma \frac{\partial V}{\partial S} = \delta$. Then the initial condition and the Robin–type boundary conditions read

$$\begin{aligned} &V(S,0) = g(S), \\ &\alpha_0 V(0,\tau) + \alpha_1 \delta(0,\tau) = g_0(\tau), \\ &\beta_0 V(S_{max},\tau) + \beta_1 \delta(S_{max},\tau) = g_1(\tau). \end{aligned} \tag{3.5}$$

To reduce sensitivities of an option to the movements of the underlying asset we need to approximate the derivatives of the solution of the Black–Scholes equation, which are called ”greeks”, e.g. Delta:

$$\Delta = \frac{\partial V}{\partial S}$$

and Gamma

$$\Gamma = \frac{\partial^2 V}{\partial S^2}.$$

The difference quotient approximations to Δ and Γ may give rise to problems of oscillating when the stock price is close to the exercise price (the situation is called '*at the money*'). It is well known that the fitted schemes give no oscillations in solutions for the Δ and Γ. These methods were obtained by the following

$$\begin{aligned} \Delta_j^n &\approx \tfrac{V_{j+1}^n - V_{j-1}^n}{2h}, \\ \Gamma_j^n &\approx \tfrac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{h^2}, \end{aligned} \tag{3.6}$$

where V_j^n is the value of the option at the grid point S_j and at the time level τ_n, $h = \Delta S = S_{i+1} - S_i$.

The main goal is to approximate V and δ in (3.4), and then we will have values for both the option price and its delta. A scheme for approximating (3.4) was proposed by Keller in 1970 [3].

Assume that $\sigma = Const$, $\mu = Const$ and $f \equiv b \equiv 0$ in (3.4) and $\alpha_0 = \beta_0 = 1$, $\alpha_1 = \beta_1 = 0$ in (3.5). Defining the quantities

$$\begin{aligned} S_{j\pm1/2}^n &= \tfrac{1}{2}\left(S_j^n + S_{j\pm1}^n\right), & \tau_{n\pm1/2} &= \tfrac{1}{2}\left(\tau_n + \tau_{n\pm1}\right), \\ V_{j\pm1/2}^n &= \tfrac{1}{2}\left(V_j^n + V_{j\pm1}^n\right), & V_j^{n\pm1/2} &= \tfrac{1}{2}\left(V_j^n + V_j^{n\pm1}\right), \\ D_S^+ V_j^n &= \tfrac{V_{j+1}^n - V_j^n}{h}, & D_\tau^+ V_j^n &= \tfrac{V_j^{n+1} - V_j^n}{\Delta\tau_{n+1}}, \end{aligned}$$

we introduce the so-called *"Keller Box Scheme"*

$$\begin{aligned} &-\sigma D_\tau^+ V_{j+1/2}^n + \sigma D_S^+ \delta_j^{n+1/2} + \mu \delta_{j+1/2}^{n+1/2} = 0, \\ &\sigma D_S^+ V_j^n = \delta_{j+1/2}^n, \end{aligned} \tag{3.7}$$

$$\begin{aligned} u_0(\tau) &= g_0(\tau), \\ u_J(\tau) &= g_1(\tau), \end{aligned} \tag{3.8}$$

where $j = 1, \ldots, J-1$ that can be obtained by discretising (3.4) and (3.5) firstly in the space by centered difference and then in time.

This scheme is unconditionally stable and has the second order of accuracy in space and time. It is also A-stable. A method is called A-stable, if all numerical solutions tend to zero, as n tends to infinity, when the method applied with the fixed positive $h = S_{j+1} - S_j$ to any differential equation $\frac{dx}{dt} = qx$, where q is a complex constant with negative real part.

The semi-discretized scheme in space for (3.4) and (3.5) is given by

$$-\sigma \frac{dV_{j+1/2}}{d\tau} + \sigma D_S^+ \delta_j + \mu \delta_{j+1/2} = 0, \; j = 0, \ldots, J-1, \tag{3.9}$$

$$\begin{array}{l} \sigma D_S^+ V_j = \delta_{j+1/2}, \\ V_j(0) = g(S_j), \end{array} \quad j = 0, \dots, J, \tag{3.10}$$

$$\begin{array}{l} V_0(\tau) = g_0(\tau), \\ V_J(\tau) = g_1(\tau), \end{array} \quad j = 1, \dots, J-1. \tag{3.11}$$

Combining the terms in (3.9)

$$-\sigma \frac{d}{d\tau}(V_{j+1/2} + V_{j-1/2}) + \sigma(D_S^+ \delta_j + D_S^+ \delta_{j-1}) + \mu(\delta_{j+1/2} + \delta_{j-1/2}) = 0, \tag{3.12}$$

we get a system of Ordinary Differential Equations (ODEs) which is A-stable. According to the following lemma

Lemma 1

(a) $D_S^0 \delta_j = \sigma D_S^+ D_S^- V_j,$
(b) $\delta_{j+1/2} + \delta_{j-1/2} = 2\sigma D_S^0 V_j,$
(c) $D_S^+ \delta_j + D_S^+ \delta_{j-1} = 2 D_S^0 \delta_j,$

where $j = 1, \dots, J-1,$

the equation (3.12) can be written in the form

$$\frac{1}{4}\frac{d}{d\tau}(V_{j-1} + 2V_j + V_{j+1}) + \sigma D_S^+ D_S^- V_j + \mu D_S^0 V_j = 0. \tag{3.13}$$

Finally, we can obtain a system of ODEs from this equation

$$C\frac{dU}{d\tau} + AU = 0, \ U(0) = U_0, \tag{3.14}$$

where

$$C = \begin{pmatrix} 2 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 2 \end{pmatrix}$$

and the matrix A is known. If we will use fitting in (3.14), we can see that this system is unconditionally stable.

Now, the full–discretized scheme for (3.4) can be obtained if we apply a Crank–Nicolson time discretization on equation (3.14). Then we get

$$C\frac{U^{n+1} - U^n}{\Delta\tau} + AU^{n+1/2} = 0,$$

$$C\frac{U^{n+1} - U^n}{\Delta\tau} + A\frac{U^{n+1} + U^n}{2} = 0,$$

or

$$2C(U^{n+1} - U^n) + \Delta\tau A(U^{n+1} + U^n) = 0.$$

And finally,

$$(2C + \Delta\tau A)U^{n+1} = (2C - \Delta\tau A)U^n. \tag{3.15}$$

Equation (3.15) is the full discretization of (3.4) in the matrix form.

Chapter 4

The Classical Finite Difference Scheme

As it was mentioned before Black and Scholes derived a model equation

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{4.1}$$

It is also well known that the European vanilla call option at the maturity time T has the value

$$V(S,t) = max(S-K,0),\ t = T, \tag{4.2}$$

where S is the asset price and K is the exercise or strike price.

Let us consider an extended Black–Scholes model [6] with right–hand side source, gradient and curvature term

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = -q - S\frac{\partial f}{\partial S} - S\frac{\partial}{\partial S}\left(S\frac{\partial g}{\partial S}\right). \tag{4.3}$$

A sign change and a logarithmic change of dependent variable

$$\begin{aligned} x &= \log\left(\tfrac{S}{S_0}\right) - \int_{t^0}^{t} R(t')dt', \\ S &= S_0 \exp\left(x + \int_{t^0}^{t} R(t')dt'\right), \end{aligned} \tag{4.4}$$

convert equation (4.3) to a time–reversed advection–diffusion equation with forcing

$$-\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2\frac{\partial^2 V}{\partial x^2} + \left(\frac{1}{2}\sigma^2 - r + R\right)\frac{\partial V}{\partial x} + rV = q + \frac{\partial f}{\partial x} + \frac{\partial^2 g}{\partial x^2}. \tag{4.5}$$

If the grid inflation $R(t)$ is chosen to vanish the advective term $\frac{\partial V}{\partial x}$ from equation (4.5), then we can assimilate the mean drift of the $V(S,t)$ in original S–variable in the stretching of the x–coordinate. The optimal choice for $R(t)$, even for the case of different $r(t)$ and $\sigma^2(t)$, given in [6], is

$$\widetilde{R(t)} = r(t) - \frac{1}{2}\sigma^2(t). \tag{4.6}$$

From equation (4.4) we can see that if computational grid is chosen uniformly in the financial variables S_j, then it is non–uniform in x_j. Time is indexed forwards from the present time t^0 by superscripts n and the numerical approximations of the option price at the grid points are denoted as V_j^n.

Let us consider the compact finite difference scheme. Such schemes use the minimum number of successive mesh points (x_j) and of time steps (t^n), i.e. 3 by 2. The matching of that approximation to the equation (4.3) within minimum number of points (3 by 2) involves the combination of the six point-grid. The accuracy of the compact finite difference scheme has the fourth order in space and second order in time.

To find compact finite difference approximated values of $V(S,t)$, $\frac{\partial V}{\partial x}$ and $\frac{\partial^2 V}{\partial x^2}$ we use the x–centroid point [5]. Let introduce the difference operators D, D_x and D_{xx}

$$\begin{aligned}
D(V^n) &= \tfrac{V_{j-1}^n + V_j^n + V_{j+1}^n}{3} - \tfrac{2b_j}{3dx_j}\left(\tfrac{V_{j+1}^n - V_j^n}{x_{j+1}-x_j} - \tfrac{V_j^n - V_{j-1}^n}{x_j - x_{j-1}}\right),\\
D_x(V^n) &= \tfrac{V_{j+1}^n - V_{j-1}^n}{dx_j} - \tfrac{x_{j-1}-2x_j+x_{j+1}}{3dx_j}\left(\tfrac{V_{j+1}^n - V_j^n}{x_{j+1}-x_j} - \tfrac{V_j^n - V_{j-1}^n}{x_j - x_{j-1}}\right),\\
D_{xx}(V^n) &= \tfrac{2}{dx_j}\left(\tfrac{V_{j+1}^n - V_j^n}{x_{j+1}-x_j} - \tfrac{V_j^n - V_{j-1}^n}{x_j - x_{j-1}}\right),
\end{aligned} \tag{4.7}$$

where $dx_j = x_{j+1} - x_{j-1}$ and b_j denotes a local effective mean–square spacing for x_j

$$b_j = \frac{1}{3}\left((x_j - x_{j-1})^2 + (x_j - x_{j-1})(x_{j+1} - x_j) + (x_{j+1} - x_j)^2\right). \tag{4.8}$$

In order to find out the time dependence of $R(t)$, $r(t)$ and $\sigma^2(t)$ within $\Delta t = t^{n+1} - t^n$ we will use the following hat notations, provided in [6]

$$\begin{aligned}
\widehat{R} &= \tfrac{1}{\Delta t}\int_0^{\Delta t} R(t^n + \tau)d\tau,\\
\widehat{r} &= \tfrac{1}{\Delta t}\int_0^{\Delta t} r(t^n + \tau)d\tau,\\
\widehat{\sigma}^2 &= \tfrac{1}{\Delta t}\int_0^{\Delta t} \sigma^2(t^n + \tau)d\tau.
\end{aligned} \tag{4.9}$$

The choice of the difference operator D and hat notations allows us to rewrite the optimal or near optimal compact finite difference scheme with parameters Θ, Σ and Ω

$$
\begin{aligned}
&(1+\frac{1}{2}\widehat{r}\Delta t+\Theta(\widehat{r}\Delta t)^2)\Bigg[\frac{D(V^n)}{\Delta t}+(\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})(\frac{1}{2}-\Sigma)D_x(V^n)-\\
&-(\frac{1}{4}\widehat{\sigma}^2+\frac{1}{2}\Sigma\Delta t(\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})^2-\Omega)D_{xx}(V^n)\Bigg]+\\
&+(1-\frac{1}{2}\widehat{r}\Delta t+\Theta(\widehat{r}\Delta t)^2)\Bigg[-\frac{D(V^{n+1})}{\Delta t}+(\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})(\frac{1}{2}+\Sigma)D_x(V^{n+1})-\\
&-(\frac{1}{4}\widehat{\sigma}^2+\frac{1}{2}\Sigma\Delta t(\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})^2+\Omega)D_{xx}(V^{n+1})\Bigg]=\\
&=(1+\frac{1}{2}\widehat{r}\Delta t+\Theta(\widehat{r}\Delta t)^2)\Bigg[\frac{D(q^n)+D_x(f^n)+D_{xx}(g^n)}{2}-\\
&-\frac{\Sigma\Delta t(\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})(D_x(q^n)+D_{xx}(f^n))}{2}+\frac{\Omega\Delta t D_{xx}(q^n)}{2}\Bigg]+\\
&+(1-\frac{1}{2}\widehat{r}\Delta t+\Theta(\widehat{r}\Delta t)^2)\Bigg[\frac{D(q^{n+1})+D_x(f^{n+1})+D_{xx}(g^{n+1})}{2}-\\
&-\frac{\Sigma\Delta t(\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})(D_x(q^{n+1})+D_{xx}(f^{n+1}))}{2}+\frac{\Omega\Delta t D_{xx}(q^{n+1})}{2}\Bigg].
\end{aligned}
\tag{4.10}
$$

In the unforced (European option) case, varying Θ has only an order $(\widehat{r}\Delta t)^3$ effect. Then, according to the paper of Smith, Part 2 [5], the optimal Θ should be

$$
\widetilde{\Theta}=\left(\frac{1}{1-\exp(-\widehat{r}\Delta t)}-\frac{1}{\widehat{r}\Delta t}-\frac{1}{2}\right)\frac{1}{r\Delta t}\approx\frac{1}{12}-\frac{(\widehat{r}\Delta t)^2}{720}+\frac{(\widehat{r}\Delta t)^4}{30240}. \tag{4.11}
$$

In the unforced (European option) case, varying Ω has only a third-order effect in terms of the length of x step. In that case, as derived in the paper of Smith, Part 2 [5], the optimal Ω is

$$
\Omega=\frac{1}{12}((\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})^2\Delta t)+\frac{b_j}{6\Delta t}-\frac{\Sigma\widehat{\sigma}^2}{2}. \tag{4.12}
$$

In Part 1 of Smith's paper [4] was shown that the choice $\Sigma=0$ leads to the Courant condition

$$
(\frac{1}{2}\widehat{\sigma}^2+\widehat{R}-\widehat{r})^2\Delta t^2<b_j, \tag{4.13}
$$

i.e. the time-step is short enough that the price drift of the solution is less than a space grid in one time-step. It was obtained in the paper of Smith, Part 2 [5] that the unforced error at the grid points is reduced to the fifth order as Σ is

$$\Sigma = \frac{2\widehat{\sigma}^2(b_j - (\widehat{\sigma}^2 + 2\widehat{R} - 2\widehat{r})^2\Delta t^2)}{\Delta t(12\widehat{\sigma}^4 - (\widehat{\sigma}^2 + 2\widehat{R} - 2\widehat{r})^2 b_j + \frac{1}{4}(\widehat{\sigma}^2 + 2\widehat{R} - 2\widehat{r})^4\Delta t^2)}. \tag{4.14}$$

Avoiding Σ be singular, the local effective mean-square spacing for the x_j grid is

$$b_j < \frac{12\widehat{\sigma}^4}{(\widehat{\sigma}^2 + 2\widehat{R} - 2\widehat{r})^2} + \frac{(\widehat{\sigma}^2 + 2\widehat{R} - 2\widehat{r})^2\Delta t^2}{4}. \tag{4.15}$$

If we substitute $\widetilde{R(t)}$ from equation (4.6) into $\widehat{R}$ in equations (4.12) and (4.14), we get optimal values for Σ

$$\widetilde{\Sigma} = \frac{b_j}{6\widehat{\sigma}^2\Delta t}, \tag{4.16}$$

and Ω

$$\widetilde{\Omega} = \frac{b_j}{12\Delta t}, \tag{4.17}$$

where the mean-square grid spacing b_j can be equal to $\sqrt{5}\widehat{\sigma}^2\Delta t$, if $\widehat{\sigma}^2$ does not change between time steps.

Now let us proceed to finding the solution of three-points-compact finite-difference schemes. First, we transfer the V_j^{n+1} terms in equation (4.10) to the right–hand side. It will have the form then

$$-\alpha_j V_{j-1}^n + \beta_j V_j^n - \gamma_j V_{j+1}^n = \delta_j^n, \tag{4.18}$$

where coefficients α_j, β_j and γ_j are known. δ_j has known value and α_j, β_j and γ_j will depend on time n, if parameters r, R and σ are time-dependent.

The solution V_j^n, with given end value V_J^n in the form

$$V_J^n = S_J^n - K \exp\left\{ -\int_{t^0}^{t^n} r(z)dz \right\}, \qquad (4.19),$$

at the highest computational extremity S_J^n of the share price and the strike or exercise price K, is followed by a right–to–left sweep to smaller share prices

$$V_j^n = \epsilon_j V_{j+1}^n + \varsigma_j^n, \; for \; j < J, \tag{4.20}$$

for some appropriate coefficients ϵ_j and ς_j^n. If we use equation (4.20) to express V_{j-1}^n from equation (4.18), the we get

$$V_j^n = \frac{\gamma_j V_{j+1}^n + \delta_j^n + \alpha_j \varsigma_{j-1}^n}{\beta_j - \alpha_j \epsilon_{j-1}}. \tag{4.21}$$

From equations (4.20) and (4.21) we can conclude that ϵ_j and ς_j^n will have following forms

$$\left.\begin{array}{l} \epsilon_j = \frac{\gamma_j}{\beta_j - \alpha_j \epsilon_{j-1}}, \\ \varsigma_j^n = \frac{\delta_j^n + \alpha_j \varsigma_{j-1}^n}{\beta_j - \alpha_j \epsilon_{j-1}}, \end{array}\right. \quad for\ j > 1. \tag{4.22}$$

We know that the value V_1^n of the European Call option is zero at the share price S_1^n, and thus, we have

$$\begin{array}{l} \epsilon_1 = 0, \\ \varsigma_1^n = 0. \end{array} \tag{4.23}$$

Finally, the derived ideas of finding the solution we can write down as backward in time from $n+1$ to n algorithm. It states as following:

1. Initialize ϵ_1 and ς_1^n with equation (4.23).

2. Calculate ϵ_j and ς_j^n for $j > 1$ using a left–to–right j–increasing sweep (4.22).

3. Evaluate V_J^n from equation (4.19).

4. Get final values by right–to–left j–decreasing sweep (4.20).

In Smith's paper [6] it was numerically shown that the optimal or near optimal classical compact finite–difference scheme is an accurate one. In the least accurate case, when $S = 0$ and the grid inflation $R = 0$, the scheme has an error that is less then 1 percent of the largest found numerical value [6]. In most accurate case the scheme error is less than 0.07 of percent of the estimated value. According to the presented algorithm to estimate the price of the vanilla call option we can say that the classical finite–difference scheme is simple and feasible to use as it includes only six grid points, three points in space and two time points.

4.1 The Crank–Nicolson Method

The Crank–Nicolson method is a classical Finite Difference Method for option pricing which includes the explicit and the implicit methods as a combination. It was developed by John Crank and Phyllis Nicolson and has a second order of convergence in time.

If we apply this method to the Black–Scholes equation, we obtain

$$\frac{V_j^{n+1}-V_j^n}{\Delta\tau} = \frac{rhj}{2}\left(\frac{V_{j+1}^{n+1}-V_{j-1}^{n+1}}{2h}+\frac{V_{j+1}^{n}-V_{j-1}^{n}}{2h}\right)-\frac{r}{2}\left(V_j^{n+1}+V_j^n\right)+$$
$$+\frac{\sigma^2h^2j^2}{4h}\left(\frac{V_{j-1}^{n+1}-2V_j^{n+1}+V_{j+1}^{n}}{h^2}+\frac{V_{j-1}^{n}-2V_j^{n}+V_{j+1}^{n}}{h^2}\right), \tag{4.24}$$

where $\Delta\tau = \tau_{n+1}-\tau_n$ and $h = \Delta S = S_{i+1}-S_i$.

We can rewrite this expression as

$$a_j^d V_{j-1}^{n+1} + a_j^c V_j^{n+1} + a_j^u V_{j+1}^{n+1} = b_j^d V_{j-1}^n + b_j^c V_j^n + b_j^u V_{j+1}^n, \tag{4.25}$$

where

$$\begin{aligned} a_j^d &= -\tfrac{\Delta\tau}{4}\left(\sigma^2 j^2 - rj\right),\\ a_j^c &= 1+\tfrac{\Delta\tau}{2}\left(\sigma^2 j^2 + r\right),\\ a_j^u &= -\tfrac{\Delta\tau}{4}\left(\sigma^2 j^2 + rj\right), \end{aligned} \tag{4.26}$$

and

$$\begin{aligned} b_j^d &= \tfrac{\Delta\tau}{4}\left(\sigma^2 j^2 - rj\right),\\ b_j^c &= 1-\tfrac{\Delta\tau}{2}\left(\sigma^2 j^2 + r\right),\\ b_j^u &= \tfrac{\Delta\tau}{4}\left(\sigma^2 j^2 + rj\right). \end{aligned} \tag{4.27}$$

Let matrices A and B be

$$A = \begin{bmatrix} a_1^c & a_1^u & & & \\ a_2^d & a_2^c & a_2^u & & \\ & \ddots & \ddots & \ddots & \\ & & a_{M-2}^d & a_{M-2}^c & a_{M-2}^u \\ & & & a_{M-1}^d & a_{M-1}^c \end{bmatrix}, \tag{4.28}$$

$$B = \begin{bmatrix} b_1^c & b_1^u & & & \\ b_2^d & b_2^c & b_2^u & & \\ & \ddots & \ddots & \ddots & \\ & & b_{M-2}^d & b_{M-2}^c & b_{M-2}^u \\ & & & b_{M-1}^d & b_{M-1}^c \end{bmatrix}, \tag{4.29}$$

where M is a number of space points. After rearranging we get a system of equations with tridiagonal matrices

$$AV^{n+1} = BV^n. \tag{4.30}$$

Chapter 5

Numerical Examples

In this Chapter we present numerical solutions for European vanilla options. We use results obtained by Finite Volume Method and by Box Scheme of Keller to compare with the exact solution of the Black–Scholes equation. Each simulation was obtained by using the same parameters

$$\begin{aligned} r &= 0.015 & &\textit{interest rate} \\ \sigma &= 0.2 & &\textit{volatility} \\ K &= 100 & &\textit{strike price} \\ S_{max} &= 200 & &\textit{maximum price} \\ T &= 0.25 & &\textit{maturity date} \\ \mu &= 0.1 & &\textit{drift rate} \\ ns &= 1601 & &\textit{number of space points} \\ nt &= 800 & &\textit{number of time points} \end{aligned}$$

I. The Finite Volume Method

Before proceeding to the numerical results obtained by the Finite Volume Method let us describe an algorithm which the program is based on. The algorithm states as the following:

1. Let n=1.
2. Solve $[I - \Delta\tau M^{n+1}]V^{n+1} = V^n + \Delta\tau R^n$.
3. $n = n + 1$ and go to step 2 till $n = nt$.

An European Call option

The first simulation was used to find the solution for the European vanilla Call option. On the Figure 5.1 the numerical solution is shown. The exact

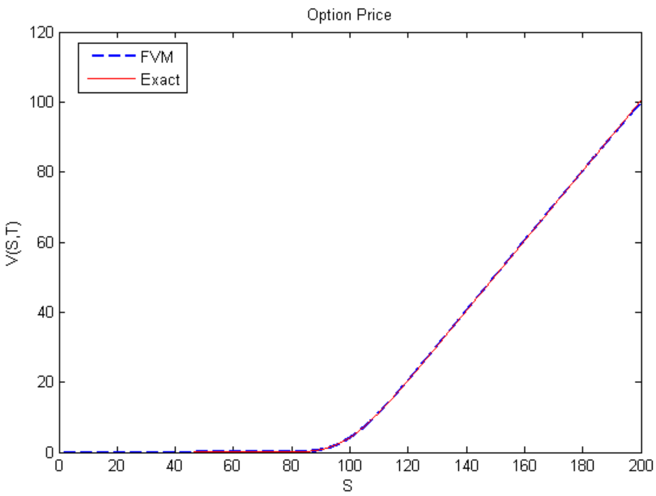

Figure 5.1: The numerical solution of the European vanilla Call option obtained by the Finite Volume Method

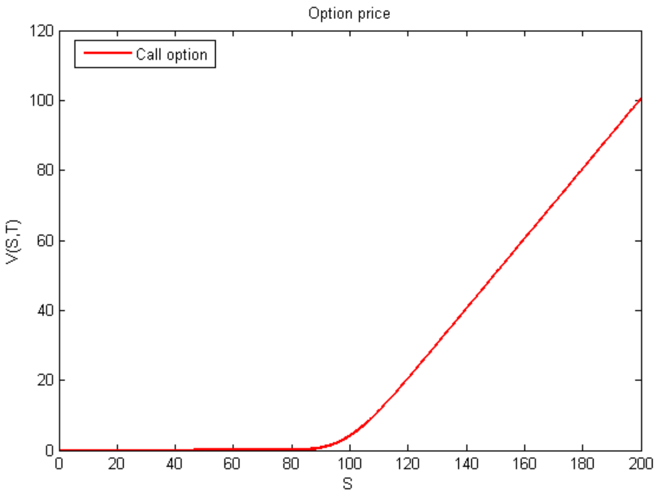

Figure 5.2: The exact solution of the European vanilla Call option

solution obtained analytically is presented on the Figure 5.2. The Figure 5.3 demonstrates the error of the Finite Volume Scheme.

According to the Figure 5.3 we can conclude that the Finite Volume Method is accurate. The numerical solution diverges a little with the exact solution in the area when the asset price is near to S_{max}.

An European Put option

The second simulation was applied on the Black–Scholes equation in order to find the solution for the European vanilla Put option. The Figure 5.4 shows

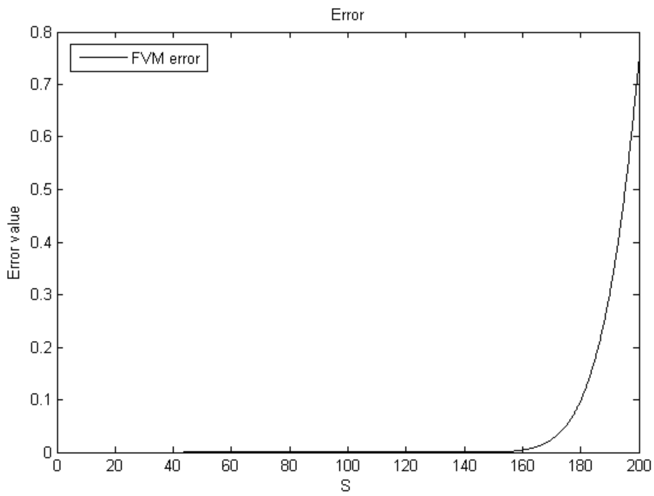

Figure 5.3: The error of the Finite Volume Method for valuation of the European vanilla Call option

the numerical solution. The exact solution obtained analytically is presented on the Figure 5.5. The Figure 5.6 illustrates the error of the Finite Volume Scheme.

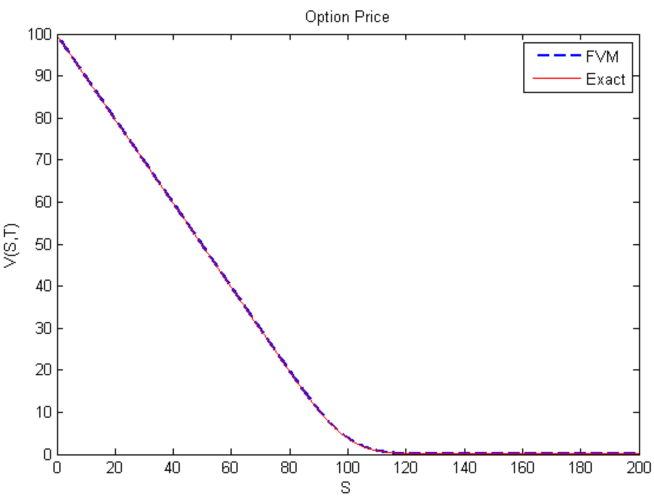

Figure 5.4: The numerical solution of the European vanilla Put option obtained by the Finite Volume Method

As we can see from the Figure 5.6, the numerical solution converges to the exact solution with small error value that appears near the strike price K. So, the approximated results obtained for the European Put option are better and much more accurate than for the European Call option.

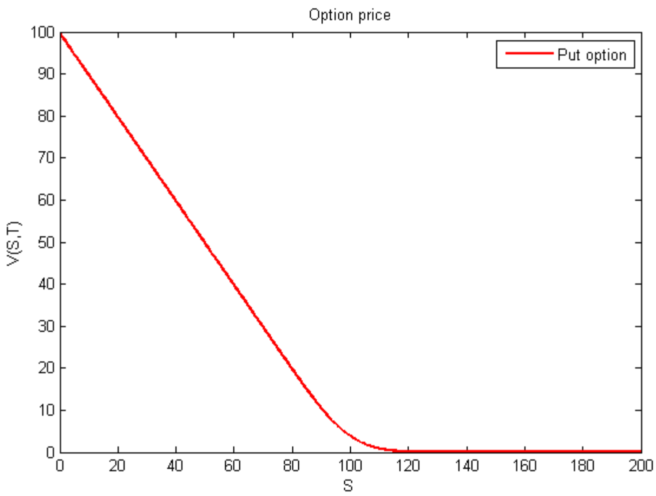

Figure 5.5: The exact solution of the European vanilla Put option

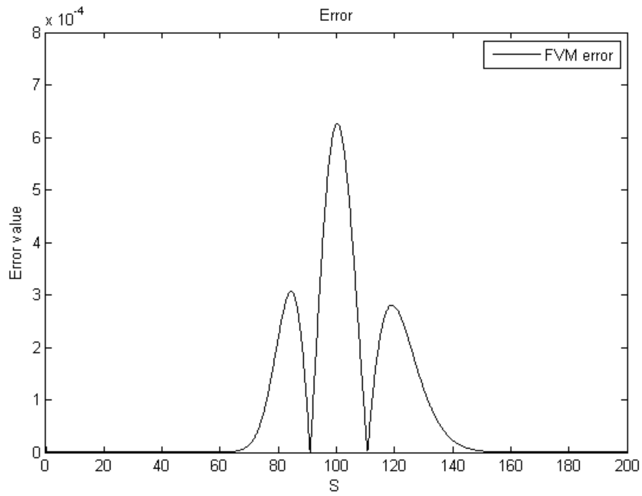

Figure 5.6: The error of the Finite Volume Method for valuation of the European vanilla Put option

II. The Box Scheme of Keller (6)

Before continuing with other numerical experiments let us describe an algorithm which the next simulations are based on. The algorithm is the following:

1. Let n=1.

2. Solve $(2C + \Delta\tau A)U^{n+1} = (2C - \Delta\tau A)U^n$.

3. $n = n + 1$ and go to step 2 till $n = nt$.

An European Call option

The next simulation was used to find the solution for the European vanilla Call option. The Figure 5.7 presents the numerical solution. The exact solution obtained analytically is illustrated on the Figure 5.8. The Figure 5.9 shows the error of the Box Scheme of Keller.

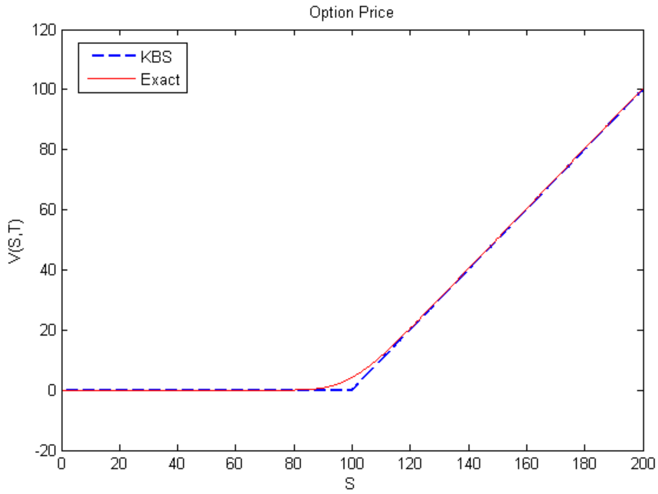

Figure 5.7: The numerical solution of the European vanilla Call option obtained by the Box Scheme of Keller

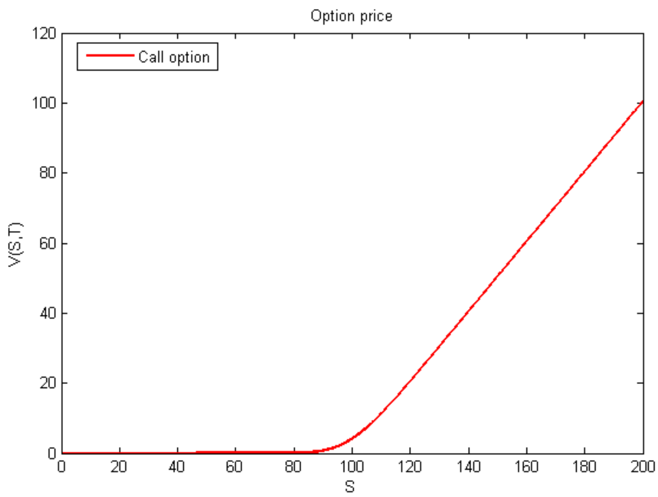

Figure 5.8: The exact solution of the European vanilla Call option

According to the Figure 5.9 we can conclude that the Box Scheme of Keller is not accurate, as the numerical solution diverges a lot with the exact solution when asset price is almost the same as the strike price.

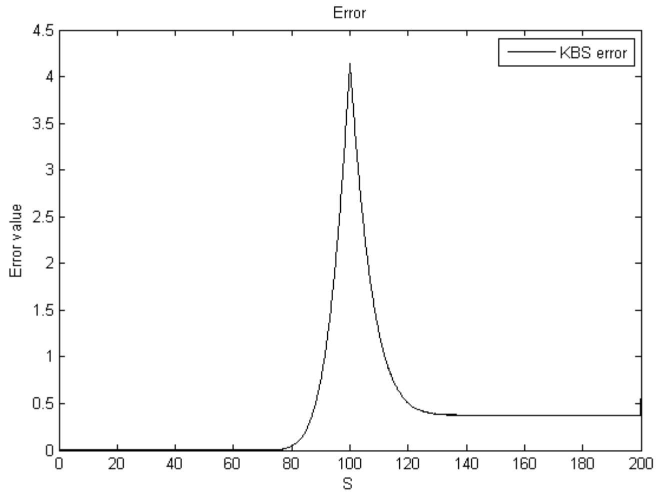

Figure 5.9: The error of the Box Scheme of Keller for valuation of the European vanilla Call option

An European Put option

The other simulation was made in order to find the solution for the European vanilla Put option. The Figure 5.10 demonstrates the numerical solution. The exact solution obtained analytically is presented on the Figure 5.11. The Figure 5.12 shows the error of the Box Scheme of Keller.

As we can see from the Figure 5.12, the numerical solution also diverges with the exact solution. So, the approximated results obtained for the European Call and Put options differ from the analytical solutions when the asset price is near to strike price.

Thus, the Box scheme of Keller is not an accurate scheme for evaluating the options numerically. Obviously, it is more convenient to use the Finite Volume Method instead for pricing options.

III. The Crank–Nicolson method (6)

We also would like to present numerical solutions for european call and put options which were obtained by standard Finite–difference method. The main idea was to compare obtained results with Crank–Nicolson method which is known to give good approximations except area around the strike price.

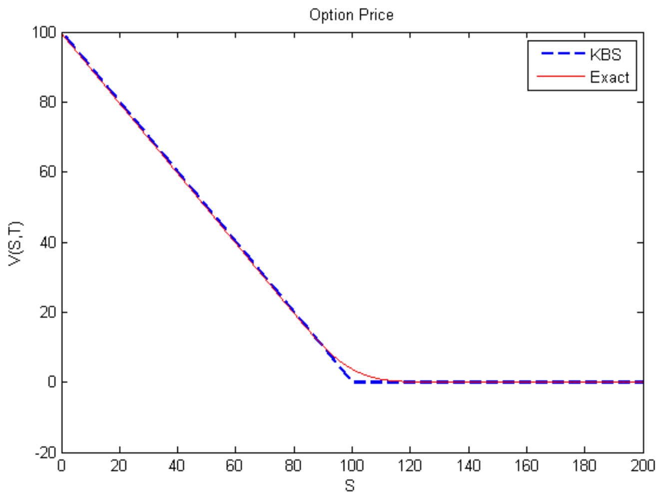

Figure 5.10: The numerical solution of the European vanilla Put option obtained by the Finite Volume Method

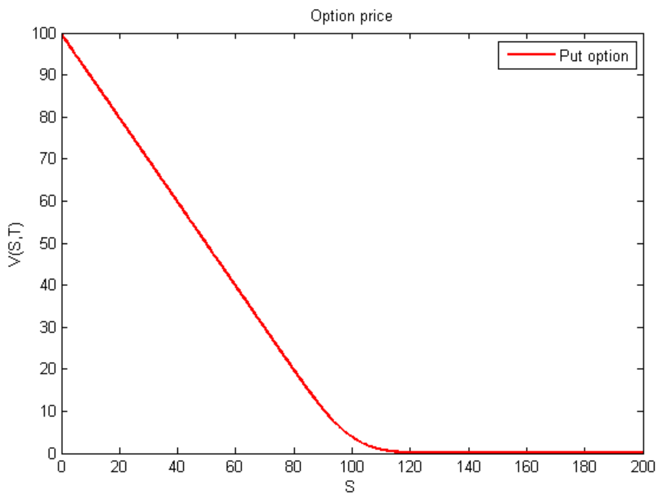

Figure 5.11: The exact solution of the European vanilla Put option

An European Call option

The next simulation was used to find the solution for the European vanilla Call option. The Figure 5.13 presents the numerical solution. The exact solution obtained analytically is illustrated on the Figure 5.14. The Figure 5.15 shows the error of the Box Scheme of Keller.

According to the Figure 5.15 we can say that the Crank–Nicolson method was predictable. For vanilla call option we can see the divergence with the exact solution around the strike price.

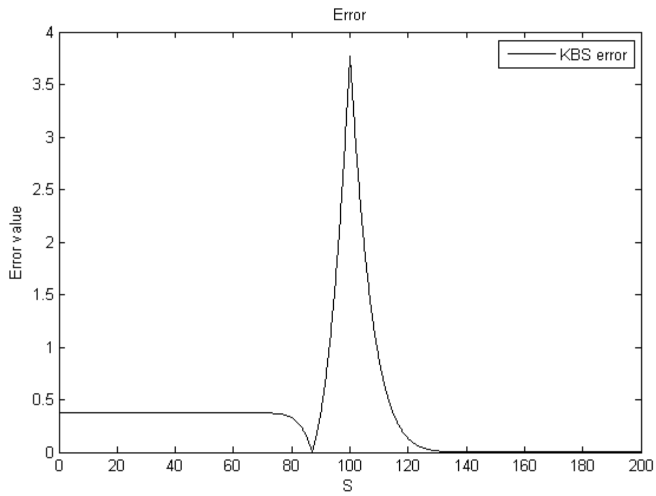

Figure 5.12: The error of the Finite Volume Method for valuation of the European vanilla Put option

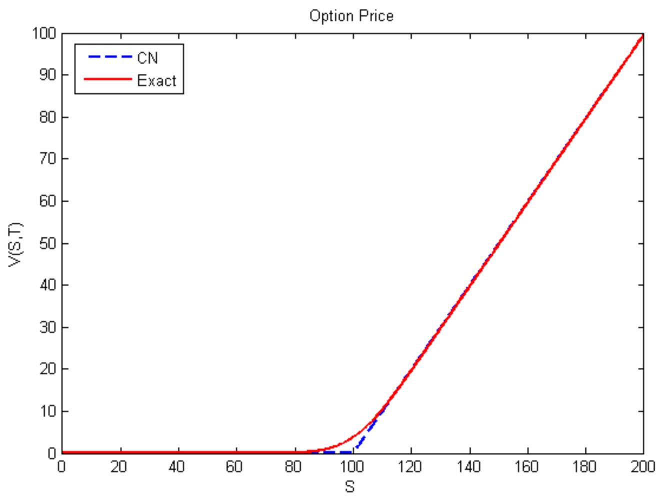

Figure 5.13: The numerical solution of the European vanilla Call option obtained by the Crank–Nicolson method

An European Put option

The last simulation was made in order to find the solution for the European vanilla Put option. The Figure 5.16 demonstrates the numerical solution. The exact solution obtained analytically is presented on the Figure 5.17. The Figure 5.18 shows the error of the Crank–Nicolson method.

As we can see from the Figure 5.18, the numerical solution also diverges with the exact solution as the asset price is near to strike price.

Thus, the Crank–Nicolson method is an accurate scheme for evaluating

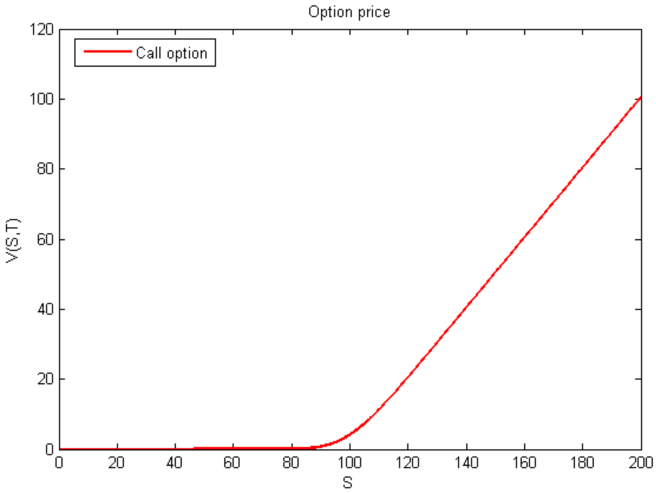

Figure 5.14: The exact solution of the European vanilla Call option

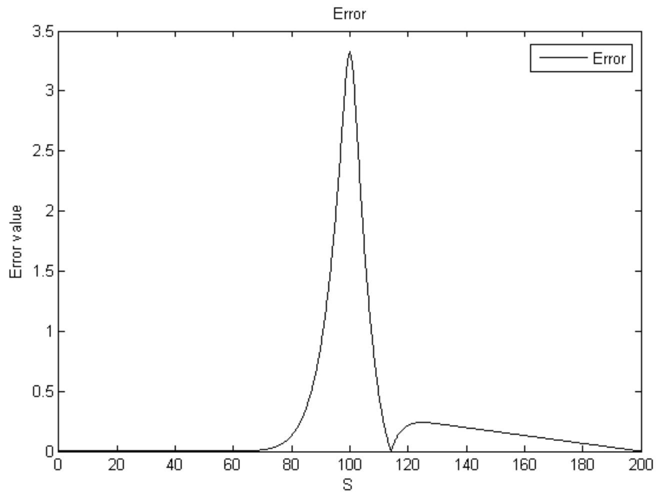

Figure 5.15: The error of the Crank–Nicolson method for valuation of the European vanilla Call option

the options numerically but gives great error in the area where the asset price is about the strike price.

According to obtained numerical results we can say that the Finite Volume Method has the least error comparing with other mentioned methods, the box scheme of Keller and the Crank–Nicolson method, that diverge from the exact solution when the value of an asset is close to the strike price. And the Finite Volume Method allows to avoid such errors in the this case called 'At-the-money'.

Thus, we can conclude that the Finite Volume Method is better to use

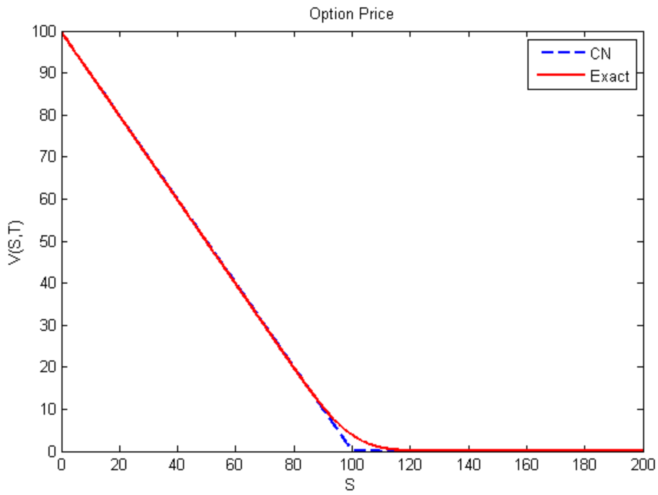

Figure 5.16: The numerical solution of the European vanilla Put option obtained by the Crank–Nicolson method

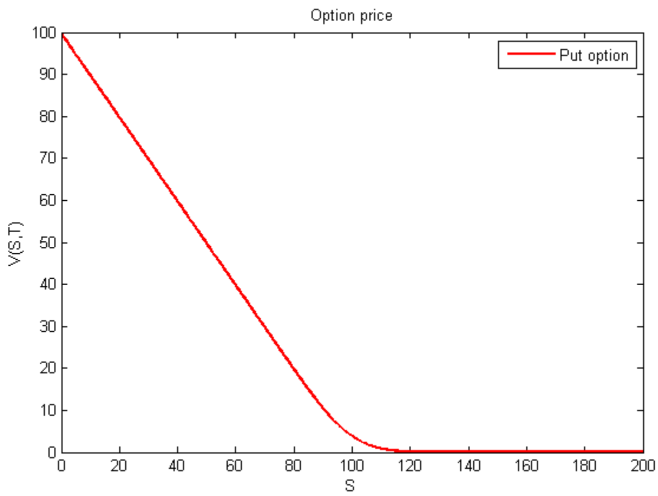

Figure 5.17: The exact solution of the European vanilla Put option

than the other methods to evaluate the price of options.

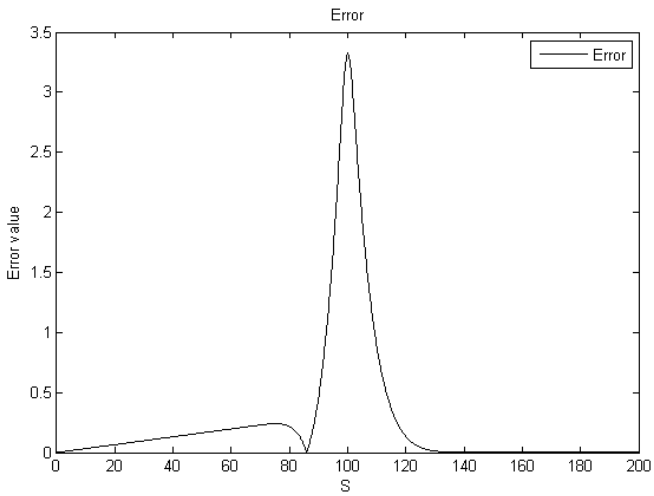

Figure 5.18: The error of the Finite Volume Method for valuation of the European vanilla Put option

Chapter 6

Conclusions

In this thesis were described various types and styles of options. We showed different numerical schemes for solving Black–Scholes partial differential equation (PDE). The main goal was to concentrate on the Finite Volume method and on comparison this method with other numerical schemes. We used the Finite Volume Method (Chapter 2), the Box Scheme of Keller (Chapter 3) and the Crank–Nicolson scheme (Chapter 4.1) to estimate the price of different options. It was shown in theorems that the Finite Volume Method is consistent and stable and proofs of these theorems were provided in Appendix 6. We also introduced the Box Scheme of Keller and mentioned that this scheme is unconditionally stable and has the second order of accuracy in space and time according to the Keller's paper [3]. Then we also described the Classical Compact Finite Difference scheme (Chapter 4) and the Crank–Nicolson scheme (Chapter 4.1).

All algorithms based on mentioned schemes were presented in Chapter 5. We wrote programs in MATLAB which we apply on the Black–Scholes equation to obtain numerical solutions for European vanilla Call and Put options. All program codes may be found in Appendix 6.

It was very important to demonstrate the useful discretization method which is very suitable for solving Black–Scholes equation, as it can overcome many difficulties that may cause, and for the various numerical simulations. In Chapter 5 numerical examples obtained by the Finite Volume Method and compared with other discretization schemes were illustrated.

According to obtained numerical results we can say that the Finite Volume Method has the least error comparing with other methods. So the usage of the Finite Volume Method is more preferable to use for options evaluating than the other methods.

Notation

$V(S,\tau)$	The price of an option.
S	The price of an underlying asset.
$\tau = T - t \in [0,T]$	The reversal time.
T	The expiring date or maturity time.
r	The riskfree interest rate.
$\sigma(S,\tau)$	The volatility.
μ	The drift rate.
K	The strike price.
$\rho(V)$	The flux function.
Δ	The Delta function, "greek".
Γ	The Gamma function, "greek".
q	The source term.
f	The gradient term.
g	The curvature term.
D, D_x, D_{xx}	The difference operators.
b_j	The local effective mean-square spacing.
$R(t)$	The grid inflation.
Θ, Σ, Ω	The parameters for a compact finite difference scheme.
$\widehat{R(t)}, \widehat{\Theta}, \widehat{\Sigma}, \widehat{\Omega}$	Optimal values.

Bibliography

[1] K. Zhang, S. Wang (2008)
Pricing Options under Jump Diffusion Processes with a Fitted Finite Volume Method. Appl. Math. Comput. **201**, 398 – 413.

[2] S. Wang (2004)
A novel fitted finite volume method for the Black-Scholes equation governing option pricing. IMA J. Numer. Anal. **24**, 699 - 720.

[3] H. Keller (1970)
A new Difference Scheme for Parabolic Problems. In Numerical Solution of Partial Differential Equations, II (SYNSPADE), Academic Press, New York, 327 – 350.

[4] R. Smith (1999)
Optimal or near optimal advection–diffusion finite–difference schemes. I. Constant coefficient in one dimension. Proc. R. Soc. Lond. A **455**, 2371 – 2387.

[5] R. Smith (2000)
Optimal and near-optimal advection–diffusion finite–difference schemes. II. Unsteadiness and non-uniform grid. Proc. R. Soc. Lond. A **456**, 489 – 502.

[6] R. Smith (2000)
Optimal and near-optimal advection–diffusion finite–difference schemes. III. Black–Scholes equation. Proc. R. Soc. Lond. A **456**, 1019 – 1028.

[7] D. C. L. LAM (1976)
Centered Differencing and the Box Scheme for Diffusion Convection Problems. Journal of computational physics **22**, 486 – 500.

[8] J.-P. Croisille, Metz (2002)
Kellers Box-Scheme for the One–Dimensional Stationary Convection–Diffusion Equation. Computing **68**, 37 - 63.

[9] A. Anderson, J. Andresen (2000)
Jump diffusion process: Volatility smile fitting and numerical methods for option pricing. Rev. Derivatives Res. **4**, 231 – 262.

[10] L. Angermann, S. Wang (2007)
Convergence of a fitted finite volume method for European and American option valuation. Numer. Math. **106**, 1 – 40.

[11] D.J. Duffy (2006)
Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach. John Wiley and Sons, Chichester, England.

[12] K. Zhang, S. Wang (2009)
A computational scheme for uncertain volatility model in option pricing. Appl. Numer. Math. **59**, 1754 – 1767.

[13] A. A. Samarskii (2001)
The theory of difference schemes. Marcel Dekker, Inc., New York.

[14] J. Hull (1993)
Options, futures, and other derivative securities. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[15] P. Willmot (2006)
Paul Wilmott On Quantitative Finance. John Wiley and Sons Ltd, The Atrium, Southern Gate, Chichester, England.

Appendix

Theorem 1 *The discretization (2.19) is consistent, i.e. the truncation error* (TE) *tends to zero uniformly with the mesh size* $h = S_{i+1} - S_i$.

Proof: The consistency of scheme (2.19) is based on the consistency of the flux $\rho(V)$, i.e. the numerical flux of a regular function interpolation tends to the continuous flux as the mesh size vanishes. In [2] it was shown that $|[\rho(V) - \rho_h(V)]_{S_{i+1/2}}| = O(h)$. Consequently, the discretization (2.19) is consistent.

□

Theorem 2 *The discretization (2.19) is stable, if*

$$|V_j^n| = \left| \frac{1}{1 - \Delta\tau_{n+1}(\xi_j e^{-ikh} + \varphi_j e^{ikh} + \psi_j)} \right| \leq 1,$$

where $h = \Delta S$.

Proof: We will use the Fourier ansatz for proving the stability of (2.19)

$$V_j^n = \Lambda^n e^{ikjh},$$

where k denotes the wave number and $i = \sqrt{-1}$. Substituting such expression into (2.19) we obtain

$$\Lambda^{n+1} e^{ijkh} - \Lambda^n e^{ikjh} = \Delta\tau_{n+1} \left(\xi_j \Lambda^{n+1} e^{(i(j-1)kh} + \psi_j \Lambda^{n+1} e^{ijkh} + \varphi_j \Lambda^{n+1} e^{i(j+1)kh} \right),$$

where $\Delta\tau_{n+1} = Const$, ξ_j, φ_j, ψ_j are coefficients which do not depend on time. Then

$$\Lambda - 1 = \Delta\tau\Lambda \left(\xi_j e^{-ikh} + \psi_j + \varphi_j e^{ikh} \right),$$

or

$$\frac{1}{\Lambda} = 1 - \Delta\tau \left(\xi_j e^{-ikh} + \psi_j + \varphi_j e^{ikh} \right).$$

Finally, we have

$$\Lambda = \left(1 - \Delta\tau\left(\xi_j e^{-ikh} + \psi_j + \varphi_j e^{ikh}\right)\right)^{-1}.$$

The stability requires that the absolute value of Λ be less than or equal to 1

$$|\Lambda| = \left|\frac{1}{1 - \Delta\tau\left(\xi_j e^{-ikh} + \psi_j + \varphi_j e^{ikh}\right)}\right| \leq 1.$$

The stability of discretization (2.19) depends on the coefficients ξ_j, φ_j, ψ_j which can be found numerically.

□

Finite Volume Method

This program finds the approximated value of the Vanilla Put option by Finite Volume Method and compare with the exact solution.

```
% program code

close all
clear all

% parameters
r=0.015;   % interest rate
sigma=0.2; % volatility
K=100;     % strike price
Smax=200;  % maximal asset price
T=0.25;    % maturity time

% grid
ns=1601;      % space points
nt=800;       % time points
dS=Smax/ns;   % space step, l(i)
dtau=T/nt;    % time step
S=0:dS:Smax;  % asset prices grid
tau=0:dtau:T; % time grid

% put option
V=zeros(ns+1,nt+1); % array for option
%PUT
for i=1:ns+1
```

```
      V(i,1)=max(K-S(i),0); % payoff t=T or tau=0
end
for i=1:nt+1
      V(1,i)=K*exp(-r*tau(i)); % boundary condition S=0
      V(ns+1,i)=0; % boundary condition S=Smax
end

% constants
a = 0.5*sigma^2; b = r-sigma^2; c = r+b; % eq. (2.4)
k=b/a;

% arrays
xi  = zeros(ns,1);
phi = zeros(ns,1);
psi = zeros(ns,1);

M = zeros(ns-1,ns-1);
R = zeros(ns-1,1);

% coefficients, eq-s. (2.14), (2.15)
xi(2)  = (S(2)*(a-b)) / (4*dS);
phi(2) = (0.5*b*S(3)^k*(S(2)+S(3))) / (dS*(S(3)^k - S(2)^k));
psi(2) = -(S(2)*(a+b))/(4*dS) ...
         - (0.5*b*S(2)^k*(S(2)+S(3))) / (dS*(S(3)^k - S(2)^k)) -c;

for j=3:ns
      xi(j)  = 0.5*b*S(j-1)^k*(S(j-1)+S(j)) / (S(j)^k-S(j-1)^k) /dS;
      phi(j) = 0.5*b*S(j+1)^k*(S(j)+S(j+1)) / (S(j+1)^k-S(j)^k) /dS;
      psi(j) = -0.5*b*S(j)^k*(S(j-1)+S(j))  / (S(j)^k-S(j-1)^k) /dS-...
                0.5*b*S(j)^k*(S(j+1)+S(j))  / (S(j+1)^k-S(j)^k) /dS -c;
end

for j=2:nt+1 % main time loop

      % Inhomogeneity, eq. (2.22)
      R(1)    = xi(2)*V(1,j-1);
      R(ns-1) = phi(ns)*V(ns+1,j-1);

      % Matric, eq. (2.23)
      for i=1:ns-2
            M(i+1,i) = xi(i+2);
```

```
            M(i,i+1) = phi(i+1);
            M(i,i)   = psi(i+1);
      end
      M(ns-1,ns-1) = psi(ns);

      V(2:ns,j) = (eye(ns-1,ns-1) - dtau*M)\(V(2:ns,j-1) + dtau*R);
end

% EXACT SOLUTION
Vex=zeros(ns+1,1);
coeff = r+0.5*sigma^2;
for i=1:ns+1
      d1=(log(S(i)/K)+coeff*T)/(sigma*sqrt(T));
      d2=d1-sigma*sqrt(T);
      D1=normcdf(d1,0,1);
      D2=normcdf(d2,0,1);
      Vex(i)=K*exp(-r*T)*(1-D2)-S(i)*(1-D1);
end
% end EXACT SOLUTION

%graph
plot (S,V(:,nt+1),'--b',LineWidth,2);
xlabel ('S')
ylabel ('V(S,T)')
title ('Option Price')
hold on
plot(S,Vex,'-r',LineWidth,2)
hold off

% graph for error
plot (S, (Vex(:,1)-V(:,nt+1)),'-g',LineWidth,2)
xlabel ('S')
ylabel ('Error value')
title ('Error')
```

Box Scheme of Keller

This program finds the approximated value of the Vanilla Put option by Box Scheme of Keller and compare with the exact solution.

```
close all
clear all

% parameters
r=0.015;   % interest rate
sigma=0.2; % volatility
K=100;     % strike price
Smax=200;  % maximal asset price
T=0.25;    % maturity time
mu=0.1;    % drift rate

% grid
ns=1601;      % space points
nt=800;       % time points
dS=Smax/ns;   % space step, l(i)
dtau=T/nt;    % time step
S=0:dS:Smax;  % asset prices grid
tau=0:dtau:T; % time grid

%option
V=zeros(ns+1,nt+1); %array for option
%PUT
for i=1:ns+1
      V(i,1)=max(K-S(i),0); %payoff t=T or tau=0
end
for i=1:nt+1
      V(1,i)=K*exp(-r*tau(i)); %boundary condition S=0
      V(ns+1,i)=0; %boundary condition S=Smax
end

% matrix C
C=zeros(ns-1,ns-1);
C(ns-1,ns-1)=2;
for i=1:ns-2
      C(i,i)=2;
      C(i+1,i)=1;
      C(i,i+1)=1;
```

```
end

% coefficients
ac = 2 + (8*dtau*sigma) / (dS^2);
ad = 1 - (4*dtau*sigma) / (dS^2) + (2*mu*dtau) / (dS);
au = 1 - (4*dtau*sigma) / (dS^2) - (2*mu*dtau) / (dS);

% matrix A
A=zeros(ns-1,ns-1);
A(ns-1,ns-1)=ac;
for i=1:ns-2
      A(i,i)=ac;
      A(i+1,i)=au;
      A(i,i+1)=ad;
end

% Crank Nicolson time discretization
B=2*C+dtau*A; % system matrix
% method
for j=2:nt+1
      V(2:ns,j)=B\((2*C-dtau*A)*V(2:ns,j-1));
end

% EXACT SOLUTION
Vex=zeros(ns+1,1);
coeff = r+0.5*sigma^2;
for i=1:ns+1
      d1=(log(S(i)/K)+coeff*T)/(sigma*sqrt(T));
      d2=d1-sigma*sqrt(T);
      D1=normcdf(d1,0,1);
      D2=normcdf(d2,0,1);
      Vex(i)=K*exp(-r*T)*(1-D2)-S(i)*(1-D1);
end
% end EXACT SOLUTION

%graph
plot (S,V(:,nt+1),'--b',LineWidth,2);
xlabel ('S')
ylabel ('V(S,T)')
title ('Option Price')
hold on
```

```
plot(S,Vex,'-r',LineWidth,2)
hold off

% graph for error
plot (S, (Vex(:,1)-V(:,nt+1)),'-g',LineWidth,2)
xlabel ('S')
ylabel ('Error value')
title ('Error')
```

Crank–Nicolson scheme

This program finds the approximated value of the Vanilla Put option by Crank–Nicolson method and compare with the exact solution.

```
close all
clear all

% parameters
r=0.015;    % interest rate
sigma=0.2;  % volatility
K=100;      % strike price
Smax=200;   % maximal asset price
T=0.25;     % maturity time

% grid
ns=1601;      % space points
nt=800;       % time points
dS=Smax/ns;   % space step, l(i)
dtau=T/nt;    % time step
S=0:dS:Smax;  % asset prices grid
tau=0:dtau:T; % time grid

%option
V=zeros(ns+1,nt+1); %array for option
%PUT
for i=1:ns+1
```

```
      V(i,1)=max(K-S(i),0); %payoff t=T or tau=0
end
for i=1:nt+1
      V(1,i)=K*exp(-r*tau(i)); %boundary condition S=0
      V(ns+1,i)=0; %boundary condition S=Smax
end

% coefficients
for i=1:ns-1
      ad(i) = -0.25*dtau*(sigma*sigma*S(i)*S(i) - r*S(i));
      ac(i) = 1 + 0.5*dtau*(sigma*sigma*S(i)*S(i)+r);
      au(i) = -0.25*dtau*(sigma*sigma*S(i)*S(i) + r*S(i));
      bd(i) = 0.25*dtau*(sigma*sigma*S(i)*S(i) - r*S(i));
      bc(i) = 1 - 0.5*dtau*(sigma*sigma*S(i)*S(i)+r);
      bu(i) = 0.25*dtau*(sigma*sigma*S(i)*S(i) + r*S(i));
end

% matrixes A and B
A=zeros(ns-1,ns-1);
B=zeros(ns-1,ns-1);
A(ns-1,ns-1) = ac(ns-1);
B(ns-1,ns-1) = bc(ns-1);
for i=1:ns-2
      A(i,i) = ac(i);
      A(i+1,i) = ad(i+1);
      A(i,i+1) = au(i);
      B(i,i) = bc(i);
      B(i+1,i) = bd(i+1);
      B(i,i+1) = bu(i);
end

% method
for j=2:nt+1
      V(2:ns,j)=A\B*V(2:ns,j-1);
end

% EXACT SOLUTION
Vex=zeros(ns+1,1);
coeff = r+0.5*sigma^2;
for i=1:ns+1
      d1=(log(S(i)/K)+coeff*T)/(sigma*sqrt(T));
```

```
      d2=d1-sigma*sqrt(T);
      D1=normcdf(d1,0,1);
      D2=normcdf(d2,0,1);
      Vex(i)=K*exp(-r*T)*(1-D2)-S(i)*(1-D1);
end
% end EXACT SOLUTION

%graph
plot (S,V(:,nt+1),'--b',LineWidth,2);
xlabel ('S')
ylabel ('V(S,T)')
title ('Option Price')
hold on
plot(S,Vex,'-r',LineWidth,2)
hold off

% graph for error
plot (S, (Vex(:,1)-V(:,nt+1)),'-g',LineWidth,2)
xlabel ('S')
ylabel ('Error value')
title ('Error')
```