
Technical report, IDE1113 , September 27, 2011

Efficient Numerical Solution of PIDEs in Option Pricing

Master's Thesis in Financial Mathematics

Elena Bukina



School of Information Science, Computer and Electrical Engineering
Halmstad University

Efficient Numerical Solution of PIDEs in Option Pricing

Elena Bukina

Halmstad University
Project Report IDE1113

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 27, 2011

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

Preface

I would like to express my gratitude to my supervisor prof. Matthias Ehrhardt and to the director of the program Ljudmila A. Bordag for their help and advices in writing this thesis, this work would not exist without them.

Abstract

The estimation of the price of different kinds of options plays a very important role in the development of strategies on financial and stock markets. There many books and various papers which are devoted to the exist mathematical theory of option pricing. Merton and Scholes became winners of a Nobel Prize in economy who described the basic concepts of the mathematical theory development.

In this thesis a review of the basic numerical methods for option pricing including the treatment of jumps is given. These methods are compared and numerical results are presented.

Contents

1	Introduction	1
2	Introduction To Robust Numerical Methods.	3
2.1	The Basic Model	3
2.2	The Θ -Method Discretization	5
2.3	The Crank-Nicolson Discretization	9
2.4	The Fixed-Point Iteration Method	10
3	The Basic Numerical Methods For Option Pricing In The Models With Jump-Diffusion.	13
3.1	The Basic Concepts About Fast Fourier Transformation (FFT)	14
3.2	Specifications of FFT	16
4	A Numerical Example	19
5	Conclusions	27
	Notation	29
	Bibliography	31
	Appendix	35

Chapter 1

Introduction

Options are an important part of financial markets worldwide. The market of futures and option contracts gained a popularity since its inception in 1973, when the trade of options and futures began on the Chicago Board Options Exchange.

An option is a contract that gives to the buyer the right, but not the obligation, to buy or to sell an asset at a previous agreed strike price. An option contract with no special characteristics, which has a standard expiry date and strike price, contains no unusual provisions, is called a plain vanilla option [1]. With the development of the market some additional conditions were included in terms of option contracts. It was done in response to the requests from customers, who wished to hedge the risk.

Particularly successful inventions were offered on the market in droves. Thus the nonstandard options arose containing some provisions that make them different from a straightforward option contracts with a strike price, underlying asset, and expiration date. Such options are called exotic options. Existing methods of evaluation options can be divided into two main groups. Classical models, e.g. "Black-Scholes model", value the price of options using mathematically derived formulas. However, such models are limited in valuing exotic options. Therefore, it is necessary to use other methods. Numerical methods, e.g. binomial methods, finite difference methods or Monte-Carlo methods, allow to value most exotic options and derivatives.

Eraker [3] showed that the usual assumption of "*geometric Brownian motion*" (GBM) should be improved by including discontinuous jump processes. Such models were originally introduced 1976 in the option valuation context by Merton [4]. But most of them are confined to vanilla options. There was done a very little work on numerical methods for pricing exotic options because such techniques are required when jumps are combined with non-constant local volatilities.

In general, the valuation of options with jump diffusion processes requires solving a "partial integro-differential equation" (PIDE). Such equations are functional equations which involve both integrals and derivatives of a function. This method was suggested 1993 by Amin [5].

An advantage of that method is that it can easily modify existing option pricing software. Thus, a variety of exotic option contracts can be handled in a straightforward way.

The plan of this thesis is as follows. In this chapter the introduction of the notion of financial derivatives is given. In Chapter 2 the robust numerical methods are presented. The discretization of the Θ -method and Crank-Nicolson method are also considered. The fixed-point iteration method is described and the convergence of the method is mentioned. Chapter 3 is about the numerical method for option pricing in the jump-diffusion model. All results are presented in Chapter 4 and discussed and interpreted in Chapter 5. Finally, proofs of theoretical results and MATLAB program code may be found in the Appendix.

Chapter 2

Introduction To Robust Numerical Methods.

2.1 The Basic Model

First, let us consider time period dt and let S will change by the law law

$$dS = \mu(t)Sdt + \sigma(t)SdW + (q(t) - 1)Sdp, \quad (1)$$

where $\mu(t) = r(t) - d(t) - \lambda(t)k(t)$ is the drift rate, $\sigma(t)$ is the volatility, dW is the increment of a continuous-time stochastic process, called a Wiener process, dp is a Poisson process [2]. $k(t) = E(q(t) - 1)$, that depends from t are identically independent distributed random variables representing the expected relative jump size with $k(t)$. Any $k(t)$ that belong to an interval $(-1, \infty)$ for all t and $q(t) - 1$ is an impulse function producing a jump from S to $Sq(t)$. It is important that $dp = 0$ with probability $1 - \lambda dt$ and $dp = 1$ with probability λdt , where λ is the Poisson arrival intensity, which is the expected number of "events" or "arrivals" that occur per unit time.

Now let us consider a case, when $dp = 0$ in (1), then the given equation will be equivalent to the usual stochastic process of "geometric Brownian motion" (GBM) assumed in the Black-Scholes models. If the Poisson event occurs, then equation (1) can be written as

$$\frac{dS}{S} \simeq q(t) - 1, \quad (2)$$

In this case the function $q(t) - 1$ is an impulse function producing a jump from S to $Sq(t)$.

Now we consider $V(S, t)$ as the contingent claim depending of the asset price S and time t . Let in equation (1) $dt = 0$. Then the following backward-

in-time "partial integro-differential equation" (PIDE) may be solved to determine V :

$$\frac{\partial V}{\partial t} + \mu(t)S \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \lambda E[\Delta V] - rV = 0, \quad (3)$$

where $E[\Delta V]$ can be presented in the following form

$$E[\Delta V] = E[V] - V = \int_0^{\infty} V(Sq)g(q)dq - V,$$

and $E[\cdot]$ is the expectation operator for the given equation.

Next we can apply the reversal time $\tau = T - t$, $\mu(\tau) = r(\tau) - \lambda(\tau)k(\tau)$, where T is the expiry time of the contingent claim and r is the continuously compounded risk-free interest rate, and $g(q)$ is the probability density function of the jump amplitude q with the properties that for all q , $g(q) \geq 0$. In case when the jump size is log-normally distributed, the probability density function reads

$$g(q) = \frac{\exp\left\{-\frac{(\log(q)-\nu)^2}{2\varrho^2}\right\}}{\sqrt{2\pi\varrho q}},$$

where ν is the mean and ϱ^2 is the variance of the jump size probability distribution. It is possible to present the expectation operator in the form $E[q(t)] = \exp\left\{\nu + \frac{\varrho^2}{2}\right\}$, i.e.

$$k(\tau) = E[q(\tau) - 1] = \exp\left\{\nu + \frac{\varrho^2}{2}\right\} - 1.$$

This allows us to write the following partial integro-differential equation as

$$\frac{\partial V}{\partial \tau} = \mu(\tau)S \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \lambda \int_0^{\infty} V(Sq)g(q)dq - (\lambda - r)V, \quad (4)$$

which is supplied with the following

Boundary conditions. [4]

As the asset price $S \rightarrow 0$, the equation (4) converges to

$$\frac{\partial V}{\partial \tau} = -rV. \quad (5)$$

If $S \rightarrow \infty$, we can assume that

$$\frac{\partial^2 V}{\partial S^2} \simeq 0, \quad (6)$$

This equation means that

$$V \simeq A(\tau)S + B(\tau). \quad (7)$$

This condition is called a Dirichlet condition which can be determined by the option payoff. If (7) holds, then (4) reduces to

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV. \quad (8)$$

Thus, when $S = 0$ and $S \rightarrow \infty$, then PIDE (4) reduces to the Black–Scholes PDE, and the usual boundary conditions can be imposed.

2.2 The Θ –Method Discretization

In this section we consider θ –method sampling for PIDE without jump integral terms. Thus, we reformulate the integral (4) as a correlation integral for efficiency.

To begin with let us consider

$$I(S) = \int_0^\infty V(Sq)g(q)dq. \quad (9)$$

Assuming that $x = \log(S)$ and $y = \log(q)$ we get the following

$$I(S) = \int_{-\infty}^\infty V(e^{x+y})g(e^y)de^y = \int_{-\infty}^\infty \bar{V}(x+y)\bar{f}(y)dy, \quad (10)$$

where $V(e^{x+y}) = \bar{V}(x+y)$, $g(e^y)e^y = \bar{f}(y)$ and $\bar{f}(y)$ is the probability compactness function of a jump of size $y = \log(q)$. Equation (10) also can be like $\bar{V} \otimes \bar{f}$, as it corresponds to the interdependence product of $\bar{V}(y)$ and $\bar{f}(y)$. Let us recall that the discrete form of integral (10) is

$$I_i = \sum_{j=-N/2+1}^{j=N/2} \bar{V}_{i+j}\bar{f}_j\Delta y + O((\Delta y)^2), \quad (11)$$

where $I_i = I(i\Delta x)$, $\bar{V}_j = \bar{V}(j\Delta x)$,

$$\bar{f}_j = \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \bar{f}(x) dx. \quad (12)$$

It is assumed that $x_j = j\Delta x$, $\Delta x = \Delta y$ and N is large enough such that the solution in areas of interest can be approximated by using an asymptotic boundary condition for large values of S . Note, that $\bar{f}(y)$ extensionally defined by (12) is a probability density function [6], thus, it possesses an essential property

$$\sum_{j=-N/2+1}^{j=N/2} \bar{f}_j \Delta y \leq 1, \quad \bar{f}_j \geq 0, \quad \forall j. \quad (13)$$

Let us try to use the unequal distribution of S coordinates in the grid for the PDE discretization. Assume

$$V_i^n = V(S_i, \tau_n), \quad (14)$$

where \bar{V}_j will not certainly correspond necessarily with any of the discrete values V_i in (14). Applying Lagrange basis functions defined on the S grid we can conclude that, if

$$S_{\gamma(j)} \leq e^{j\Delta x} \leq S_{\gamma(j)+1}, \quad (15)$$

then it is possible to write down as

$$\bar{V}_j = \psi_{\gamma(j)} V_{\gamma(j)} + (1 - \psi_{\gamma(j)}) V_{\gamma(j)+1} + O((\Delta S_{\gamma(j)+1/2})^2), \quad (16)$$

where $\gamma(j)$, $\psi_{\gamma(j)}$ are interpolation weights [7], and $\Delta S_{i+1/2} = S_{i+1} - S_i$.

If $e^{x\Pi(k)} \leq S_k \leq e^{x\Pi(k)+1}$, then we obtain

$$I(S_k) = \phi_{\Pi(k)} I_{\Pi(k)} + (1 - \phi_{\Pi(k)}) I_{\Pi(k)+1} + O((e^{x\Pi(k)} - e^{x\Pi(k)+1})^2), \quad (17)$$

where $\phi_{\Pi(k)}$ is an interpolation weight and $0 \leq \phi_i \leq 1$, $0 \leq \psi_i \leq 1$. Combining (11), (16), (17) we get

$$I(S_k) = \sum_{j=-N/2+1}^{j=N/2} \xi(V, k, j) \bar{f}_j \Delta y, \quad (18)$$

where $V = [V_0, V_1, \dots, V_p]'$ and

$$\xi(V, k, j) = \phi_{\Pi(k)} [\psi_{\gamma(k)+j} V_{\gamma(\Pi(k)+j)} + (1 - \psi_{\gamma(\Pi(k)+j)}) V_{\gamma(\Pi(k)+j)+1}]$$

$$+(1 - \phi_{\Pi(k)})[\psi_{\gamma(\Pi(k)+1+j)}V_{\gamma(\Pi(k)+1+j)} + (1 - \psi_{\gamma(\Pi(k)+1+j)+1})]. \quad (19)$$

Also notice that $\xi(V, k, j)$ is linear in V , then it follows $\xi(1, k, j) = 1$ for all k, j .

Now we can use an implicit method for PDE, and then we use a weighted time discretization, obtaining a method for computing the jump integral. We will rewrite the discrete equation in a following form

$$\begin{aligned} & V_i^{n+1}[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau] - \Delta\tau\beta_i V_{i+1}^{n+1} - \Delta\tau\alpha_i V_{i-1}^{n+1} \\ &= V_i^n + (1 - \theta)\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \xi(V^{n+1}, i, j)\bar{f}_j\Delta y \\ & \quad + \theta\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \xi(V^n, i, j)\bar{f}_j\Delta y, \end{aligned} \quad (20)$$

where $\theta = 0$ and $\theta = 1$ are weighting parameters coinciding to an implicit handling of the jump integral and explicit treatment of this term respectively.

In common, discretizing the first derivative term of (4) with central differences leads to

$$\begin{aligned} \alpha_i^c &= \frac{\sigma^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \lambda k)S_i}{S_{i+1} - S_{i-1}}, \\ \beta_i^c &= \frac{\sigma^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r - \lambda k)S_i}{S_{i+1} - S_{i-1}}, \end{aligned} \quad (21)$$

and if $\alpha_i^c \leq 0$ or $\beta_i^c \leq 0$, oscillations may appear in the numeric solution. These can be avoided by applying forward or backward differences at the problem nodes, leading to forward difference [9]

$$\begin{aligned} \alpha_i^f &= \frac{\sigma^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})}, \\ \beta_i^f &= \frac{\sigma^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r - \lambda k)S_i}{S_{i+1} - S_i}, \end{aligned} \quad (22)$$

or backward difference

$$\begin{aligned} \alpha_i^b &= \frac{\sigma^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \lambda k)S_i}{S_i - S_{i-1}}, \\ \beta_i^b &= \frac{\sigma^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})}. \end{aligned} \quad (23)$$

We can choose between the forward and the backward differences using this algorithm [12].

1. *if* $\alpha_i^c \geq 0$ and $\beta_i^c \geq 0$ *then*

$$\alpha_i = \alpha_i^c$$

$$\beta_i = \beta_i^c$$

2. *elseIf* $\beta_i^f \geq 0$ *then*

$$\alpha_i = \alpha_i^f$$

$$\beta_i = \beta_i^f$$

3. *else*

$$\alpha_i = \alpha_i^b$$

$$\beta_i = \beta_i^b$$

end if

Let us notice that the test condition in the algorithm assures that α_i and β_i are non-negative. For normal parameter values and grid spacing, forward or backward differencing is rarely ordered for single factor options. As usual, since this happens at only a small quantity of nodes far apart from the region of interest, the limited use of a low order scheme does not arise in inferior convergence as the mesh is refined. As we shall see, requiring that all α_i and β_i are non-negative has significant theoretical offshoots. As $S \rightarrow 0$, equation (3) reduces to $V_\tau = -rV$, which is obtained from the discrete equations (20) by setting $\alpha_i = \beta_i = 0$ at $S_i = 0$. As a rule we confine the S grid at some large value $S_p = S_{max}$, where we pose Dirichlet conditions at $S = S_{max} = S_p$. These Dirichlet conditions can be resolved by substituting equation (7) into equation (8), and applying the option payoff as initial condition. It is done by replacing equation (20) at $S = S_{max} = S_p$ with the specification that V_p^{n+1} is identical to the relevant Dirichlet condition.

Now we are still considering the stability of the sampling (20). This theorem firstly was presented in paper of d'Halluin Y.D. [17]

Theorem 1.5 (*The Basic concepts about stability of the scheme*) *The discretization method (20) is unconditionally stable for any choice of θ , $0 \leq \theta \leq 1$, provided that*

(i) $\alpha_i, \beta_i \geq 0$;

(ii) the discrete probability density \bar{f}_j has the properties (13);

(iii) the interpolation weights satisfy $0 \leq \phi_i \leq 1$, $0 \leq \psi_i \leq 1$;

(iv) $r, \lambda \geq 0$.

2.3 The Crank–Nicolson Discretization

If it is necessary to reduce the time discretization error then we can use the Crank-Nicolson method [11].

We will consider the following approach of the equation:

$$\begin{aligned}
& V_i^{n+1} \left[1 + \frac{\Delta\tau}{2} (\alpha_i + \beta_i + r + \lambda) \right] - \frac{\Delta\tau}{2} \beta_i V_{i+1}^{n+1} - \frac{\Delta\tau}{2} \alpha_i V_{i-1}^{n+1} \\
& = V_i^n \left[1 - \frac{\Delta\tau}{2} (\alpha_i + \beta_i + r + \lambda) \right] + \frac{\Delta\tau}{2} \alpha_i V_{i-1}^n \\
& + \frac{\Delta\tau}{2} \beta_i V_{i+1}^n + (1 - \theta) \lambda \Delta\tau \sum_{j=-N/2+1}^{j=N/2} \xi(V^{n+1}, i, j) \bar{f}_j \Delta y \\
& + \theta \lambda \Delta\tau \sum_{j=-N/2+1}^{j=N/2} \xi(V^n, i, j) \bar{f}_j \Delta y.
\end{aligned} \tag{24}$$

The Crank–Nicolson method is based on the central difference in space, and the trapezoidal rule in time, giving second–order accuracy in time. Recall that this method is obtained by setting $\theta = 1/2$ in equation (24). If we specify the matrix D such that

$$\begin{aligned}
-[DV^n]_i &= V_i^n \frac{\Delta\tau}{2} (\alpha_i + \beta_i + r + \lambda) - \frac{\Delta\tau}{2} \alpha_i V_{i-1}^n \\
& - \frac{\Delta\tau}{2} \beta_i V_{i+1}^n - \frac{\Delta\tau}{2} \lambda \sum_{j=-N/2+1}^{j=N/2} \xi(V^n, i, j) \bar{f}_j \Delta y,
\end{aligned} \tag{25}$$

we can rewrite (24) as

$$[I - D]V^{n+1} = [I + D]V^n. \tag{26}$$

Matrix B can be defined such that $B = [I - D]^{-1}[I + D]$, so that (25) can be like

$$V^n = B^n V^0, \tag{27}$$

Here n is defined as a degree of the matrix B . And if we define p as number of grid points and n is a number of time steps at some norm $\|\cdot\|$, then we can notice that a matrix will strictly converge, and the following holds

$$\|(B)^n\| \leq 1 \quad \forall n, p \in N. \tag{28}$$

Following Giles [8], strong stability is defined as

$$\|(B)^n\| \leq C \quad \forall n, p \in N, \tag{29}$$

and algebraic stability is defined as

$$\|(B)^n\| \leq C n^s p^l \quad \forall n, p \in N, \tag{30}$$

where C , s and $l \geq 0$ are constants independent of n and p .

Algebraic stability is a weaker condition than either strict or strong stability. The Lax Equivalence Theorem states that strong stability is an essential and sufficient condition for convergence for all initial data.

If we assume that μ_i is an eigenvalue of D then a necessary condition for strong stability is given by $|\mu_i| \leq 1$ and $|\mu_i| = 1$ that should have the frequency rate is equal to 1. From equation (25) and properties (13), we have the following

- All diagonals of D are non-negative;
- All diagonals of D (except the last row) should be strictly negative;
- Assuming that $r > 0$ and $\sum_{j=0}^{j=p} D_{ij} < 0$ for $i = 0, \dots, p - 1$;
- The last row of D should be identically equal to zero because of Dirichlet conditions.

According to the assumptions made above we can conclude that the matrix D is strictly contained in the negative part of a complex plane with one eigenvector which is equal to zero. Thus, all eigenvectors of B are equal or less than 1, except one which modulus is equal to 1. And the matrix B satisfies to all necessary conditions for strict stability. On the other hand, B is not a symmetric matrix, thereby it is not enough conditions for its bound. In that case the algebraic stability can be guaranteed by checking the eigenvalues μ_i of matrix D .

2.4 The Fixed-Point Iteration Method

If you use an implicit discretization, it is computationally prohibited to resolve the full linear system because the correlation product term makes the system dense. Therefore, we suppose the use of a fixed-point iteration to solve the linear system.

Now we consider a matrix \hat{D} such that

$$-[\hat{D}V^n]_i = V_i^n \Delta\tau(\alpha_i + \beta_i + r + \lambda) - V_{i-1}^n \Delta\tau\alpha_i - V_{i+1}^n \Delta\tau\beta_i. \quad (31)$$

It is also necessary to define a vector $\Phi(V^n)$ which is a linear function of V^n that looks like

$$[\Phi(V^n)]_i = \sum_{j=-N/2+1}^{j=N/2} \xi(V^n, i, j) \bar{f}_j \Delta y. \quad (32)$$

From above we can present an implicit discretization of Crank-Nicolson method as follows

$$[I - (1 - \theta)\widehat{D}]V^{n+1} = [I + \theta\widehat{D}]V^n + (1 - \theta)\lambda\Delta\tau\Phi(V^{n+1}) + \theta\lambda\Delta\tau\Phi(V^n). \quad (33)$$

And the fixed-point method can be written as the following:

1. Let $(V^{n+1})^0 = V^n$
2. Let $\widetilde{V}^n = (V^{n+1})^k$
3. For $k = 0, 1, 2, 3, \dots$ until convergence (34)
4. Solve $[I - (1 - \theta)\widehat{D}]\widetilde{V}^{k+1} =$

$$[I + \theta\widehat{D}]V^n + \lambda\Delta\tau\Phi(\widetilde{V}^k)(1 - \theta) + \theta\lambda\Delta\tau\Phi(V^n)$$
5. If $\max_i \frac{|\widetilde{V}_i^{k+1} - \widetilde{V}_i^k|}{\max(1, |\widetilde{V}_i^{k+1}|)} < \text{tolerance}$ then quit
 End For

If we assume that $e^k = V^{n+1} - \widetilde{V}^k$ then we can say that the fixed-point iteration method will converge by the theorem which was formulated by d'Halluin. [18]

Theorem 2.5 (Convergence of the fixed-point iteration) *Provided that*

- (i) $\alpha_i, \beta_i \geq 0$,
- (ii) the discrete probability density \bar{f}_j will have properties (13),
- (iii) the interpolation weights satisfy $0 \leq \phi_i \leq 1, \quad 0 \leq \psi_i \leq 1$,
- (iv) $r, \lambda \geq 0$,

then we note that fixed-point iterations in (34) converge globally, and the maximum error at each iteration will satisfy the following equation

$$\|e^{k+1}\|_\infty \leq \|e^k\|_\infty \frac{(1 - \theta)\lambda\Delta\tau}{1 + (1 - \theta)(r + \lambda)\Delta\tau}.$$

The proof of this theorem can be found in the Appendix.

Chapter 3

The Basic Numerical Methods For Option Pricing In The Models With Jump–Diffusion.

Now we should develop the numerical methods for correct and rational estimation of the option price in order to use the model described above in practice. These methods should satisfy to integro-differential equation (3). The main goal is also to find numerical solution of (4) which can usually arise in iterative methods of calibration.

Thereupon there is not enough papers connected with numerical methods for a finding of the decision integro-differential the equation in private derivatives in models with diffusive jump. Such models were used and described Amin [5] Thereupon, there is a sufficiently small number of papers related to numerical methods that find solutions of partial integro-differential equations in models with a diffusion jump. Such models were used and described by Amin [5] and Andreasen and Gruenewald [13].

Such methods, known as explicit, have unstable time convergence. Thus, implicit methods are much more accurate and stable in comparison with explicit ones. And therefore they are preferable for option pricing.

There are different papers that describe the use of implicit methods, that can determine the solution of the partial integro–differential equation. But such methods are used for the finite-difference Crank-Nicolson scheme, and this approach assumes that finding the unknown elements of the matrix occurs at each time step. So, this method can be used for option pricing, but finding the elements in this way is inefficient, and that’s why it is necessary to use another approach.

There are works in which implicit methods and which are capable to solve the partial integro–differential equation are used, but such methods

were used for Crank–Nicolson difference scheme, and this method assumes that there is a reference to not resolved matrixes on each time step. Basically such method is possible in use, but the assumption of invariable parameters, such reference will be not already possible. But in a case if there is a time dependence the given parameters will demand other approach. Later in this thesis we present an efficient method that can be used to find the unknown elements in the matrix and is relatively fast.

3.1 The Basic Concepts About Fast Fourier Transformation (FFT)

It is necessary first to make some transformations to use this method and then rewrite (4) as

$$\frac{\partial V}{\partial \tau} + \left[a \frac{\partial}{\partial x} + \frac{1}{2} b^2 \frac{\partial^2}{\partial x^2} - r \right] V + \lambda \int_{-\infty}^{\infty} \zeta(\tau, x - y) V(\tau, y) dy - \lambda V = 0, \quad (35)$$

where $\zeta(\tau, \cdot)$ is a density function, and $r = r(\tau)$, $a = a(\tau, x)$, $b = b(\tau, x)$, $\lambda = \lambda(\tau)$. Next we define the operator

$$D \equiv a \frac{\partial}{\partial x} + \frac{1}{2} b^2 \frac{\partial^2}{\partial x^2}$$

and if we use convolutions we can rewrite (35) in a compact form

$$\frac{\partial V}{\partial \tau} + DV - \lambda V + \lambda \zeta^* V = 0. \quad (36)$$

We discretize $V(\tau) = V(\tau, \cdot)$ with respect to time such that

$$0 = \frac{V(\tau + \Delta\tau) - V(\tau)}{\Delta\tau} + D[\theta_k V(\tau) + (1 - \theta_k) V(\tau + \Delta\tau)] \\ + \lambda(\zeta^* - 1)[\theta_l V(\tau) + (1 - \theta_l) V(\tau + \Delta\tau)],$$

where $\theta_k, \theta_l \in [0, 1]$ are constants. And according by we get

$$\left[\frac{1}{\Delta\tau} - \theta_k D - \theta_l (\zeta^* - 1) \right] V(\tau) \\ = \left[\frac{1}{\Delta\tau} + (1 - \theta_k) D + (1 - \theta_l) \lambda (\zeta^* - 1) \right] V(\tau + \Delta\tau). \quad (36)$$

We have mentioned earlier, that there are different numerical methods to determine the numerical solution of (36). We may use here the Crank–Nicolson finite difference scheme with parameters $\theta_k = \theta_l = \frac{1}{2}$, but such use is not entirely correct in this case. We need to treat to a matrix of size $N \times N$ after sampling the space of coordinates x by the number of points of N . And it is very inconvenient, since it has N^3 computational costs. But we have seen that such approach of the matrix treatment must be done at each time step, because its elements are not constant and we are not allowed to use the FFT. Then we should try the schemes, where the parameters $\theta_k = \frac{1}{2}, \theta_l = 0$ [15].

These schemes are not only stable but also effective. However, the stability decreases because of the asymptotic representation of discrete and continuous parts. Then we will use an implicit method of alternating directions. This is a method where each time step in the grid will be divided into two half-steps. For the first half-step we have $\theta_k = 1, \theta_l = 0$, and then we get.

$$\left[\frac{1}{\Delta\tau/2} - D \right] V(t + \Delta) = \left[\frac{1}{\Delta\tau/2} - \lambda + \lambda\zeta^* \right] V(\tau + \Delta\tau). \quad (37)$$

In a discrete grid we can solve this by first computing the convolution $\zeta^*V(\tau + \Delta\tau)$ in discrete Fourier space, where

$$\langle \zeta^*V(\tau + \Delta\tau) \rangle = \langle \zeta \rangle \langle V(\tau + \Delta\tau) \rangle.$$

When we notice that ζ only needs to be computed once, the computational costs associated with the convolution part of (37) is one FFT and one inverse FFT, i.e. $O(N \log_2 N)$. Then we note that the discrete version of the differential operator D is a tridiagonal matrix. Therefore, once the RHS of (37) is acquired by FFT methods, then we can resolve the system (37) at a cost of $O(N)$. Consequently, the total costs of solving (37) is $O(N \log_2 N)$.

We get the second half step if $\theta_k = 0, \theta_l = 1$, then we have

$$\left[\frac{1}{\Delta\tau/2} + \lambda - \lambda\zeta^* \right] V(\tau) = \left[\frac{1}{\Delta\tau/2} + D \right] V\left(\tau + \frac{\Delta\tau}{2}\right). \quad (38)$$

If we assume that $y = [\frac{2}{\Delta\tau} + D]V(t + \frac{\Delta\tau}{2})$ then we can apply FFT to (38).

$$\begin{aligned} \left(\frac{2}{\Delta\tau} + \lambda \right) \langle V(\tau) \rangle - \lambda \langle \zeta \rangle \langle V(\tau) \rangle &= \langle y \rangle \\ \langle V(\tau) \rangle &= \frac{\langle y \rangle}{\left(\frac{2}{\Delta\tau} \right) + \lambda - \lambda \langle \zeta \rangle} \end{aligned} \quad (39)$$

We need to define the following operators for the discrete scheme (37) and (38)

$$\delta_x f(x) = \frac{1}{2\Delta x} [f(x + \Delta x) - f(x - \Delta x)],$$

$$\begin{aligned}\delta_x x f(x) &= \frac{1}{(\Delta x)^2} [f(x + \Delta x) - 2f(x) + f(x - \Delta x)], \\ \bar{D}f(x) &= \left[a\delta_x + \frac{1}{2}b^2\delta_x x \right] f(x); \\ f(x)\bar{\zeta}^* &= \sum_j q_j(x) f(j\Delta x),\end{aligned}$$

where

$$q_j(x) = \int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} \zeta(x-y) dy.$$

Also we can present (37) and (38) in discrete form.

$$\left[\frac{2}{\Delta\tau} - \bar{D} \right] V \left(\tau + \frac{\Delta\tau}{2} \right) = \left[\frac{2}{\Delta\tau} - \lambda + \lambda\bar{\zeta}^* \right] V(\tau + \Delta\tau), \quad (40)$$

$$\left[\frac{2}{\Delta\tau} + \lambda - \lambda\bar{\zeta}^* \right] V(\tau) = \left[\frac{2}{\Delta\tau} + \bar{D} \right] V \left(\tau + \frac{\Delta\tau}{2} \right). \quad (41)$$

The next assumption describes the properties of schemes (40), (41).

Proposition 1.5 *The given properties are necessary for schemes (40), (41):*

- (i) *The given schemes are unconditionally stable in the von Neumann sense.*
- (ii) *If the parameters are known, then the numerical solution will be locally stable and has $O(\Delta\tau^2 + \Delta x^2)$ order of accuracy.*
- (iii) *If M is a number of time steps, N is the number of space steps, then the order of accuracy will be $O(MN\log_2 N)$*

3.2 Specifications of FFT

The scheme (40) and (41) is convenient for finding the numerical solution: it is unconditionally stable and has $O(N\log_2 N)$ order of accuracy. Also note that it is important to make correct representation of the convolution integral. And if FFT uses steps with identical length then the accuracy of the solution will be low in areas of interest. To avoid such a problem, we

have made the assumption in the algorithm that the linear part of an option will be equal to a number of standard deviations from the basic process out of a grid. Then the linear part should be solved in the closed form, and the solution can be found on an inner grid by FFT algorithm [15]

It is necessary to divide function into two parts. So, it turns out

$$V = V1_{x \in (\underline{x}, \bar{x})} + V1_{x \notin (\underline{x}, \bar{x})} \equiv G + H,$$

On $x \in (\underline{x}, \bar{x})$, G , then we have

$$\frac{\partial G}{\partial \tau} + DG + \lambda(\zeta^* - 1)V = \frac{\partial G}{\partial \tau} + DG - \lambda G + \lambda\zeta^*[G + H].$$

If we assume that H is linear in e^x we can write

$$H(\tau, x) \cong [g_l(\tau)e^x + h_l(\tau)]1_{x < \underline{x}} + [g_u(\tau)e^x + h_u(\tau)]1_{x > \bar{x}},$$

where g_l, g_u, h_l, h_u are deterministic functions. It means that

$$\begin{aligned} \zeta^* H(\tau, x) &\cong g_l(\tau)e^x(1 + m(t))Pr'(x + \ln J(\tau) < \underline{x}) \\ &\quad + h_l(\tau)Pr(x + \ln J(\tau) < \underline{x}) \\ &\quad + g_u(\tau)e^x(1 + m(t))Pr'(x + \ln J(\tau) > \bar{x}) \\ &\quad + h_u(\tau)Pr(x + \ln J(\tau) > \bar{x}) \end{aligned} \quad (42)$$

where $Pr(\cdot)$ is a probability in sense of distribution which defines ζ and $Pr'(\cdot)$ is a Radon-Nikodim distribution [14] represented as

$$\zeta'(\tau, x) = \frac{\zeta(\tau, x)e^x}{(1 + m(\tau))}.$$

As for lognormaly distributed jump (Merton [4]) we can evaluate the given probabilities in the closed form like function of Gaussian distribution. But if jump distributions are not set parametrical then we consider probability by elementary numerical integration on a plane ζ, ζ' . And then we obtain

$$\begin{aligned} \left[\frac{2}{\Delta\tau} - D \right] G\left(\tau + \frac{\Delta\tau}{2}\right) &= \left(\frac{2}{\Delta\tau} - \lambda \right) G(\tau + \Delta\tau) + \lambda\zeta^* G(\tau + \Delta\tau) \\ &\quad + \lambda\zeta^* H(\tau + \Delta\tau), \left[\frac{2}{\Delta\tau} + \lambda - \lambda\zeta^* \right] G(\tau) \\ &= \left[\frac{2}{\Delta\tau} + D \right] G\left(\tau + \frac{\Delta\tau}{2}\right) + \lambda\zeta^* H(\tau), \end{aligned} \quad (43)$$

where ζ^*G can be numerically evaluated by FFT, and ζ^*G can be found from (42). We assumed that H is a linear function and then functions g_l, g_u, h_l, h_u can be written as

$$rf = f_\tau + (r(\tau) - g(\tau))f_x. \quad (44)$$

For discrete time distribution (44) we have

$$f(\tau, x) = e^{-r\Delta\tau} f(\tau + \Delta\tau, x + (r(\tau) - q(\tau))\Delta\tau). \quad (45)$$

together with the boundary conditions this define g_l, g_u, h_l, h_u .

Now we use the above assumptions for various methods and programs, even for options with barriers and the American options.

Chapter 4

A Numerical Example

In the two previous chapters we described the numerical methods used for evaluating the partial integro-differential equation which arise, as the equations for option pricing with jump diffusion processes. We mentioned two various approaches to evaluate the correlation integral, FFT and the alternative approach with a vector $\Phi(V^N)$ which is a linear function of V^N . In this chapter results which were obtained by applying the numerical method with different approaches are shown. For numerical experiments we took the following parameters:

Parameters			
r = 0.2	interest rate	K = 100	strike price
vol = 0.3	volatility σ	T = 0.25	maturity time
w = 0.5	weighting coefficient θ	lam = 1	Poisson arrival intensity λ
nu = 0	mean ν	g = 1	variance γ
phi = 0.5	weighting coefficient ϕ	psi = 0.5	weighting coefficient ψ

Numerical Results for European Vanilla Call Option.

On the Figure 4.1 the exact solution of the European vanilla Call option is presented. The Figure 4.2 demonstrates the solution of the European Call obtained by the numerical method with using FFT approach. On the Figure 4.3 we can see the solution of the same option, but obtained by using the numerical method with the alternative approach. Figures 4.4 and 4.5 illustrate the error of evaluating the option with using different approaches.

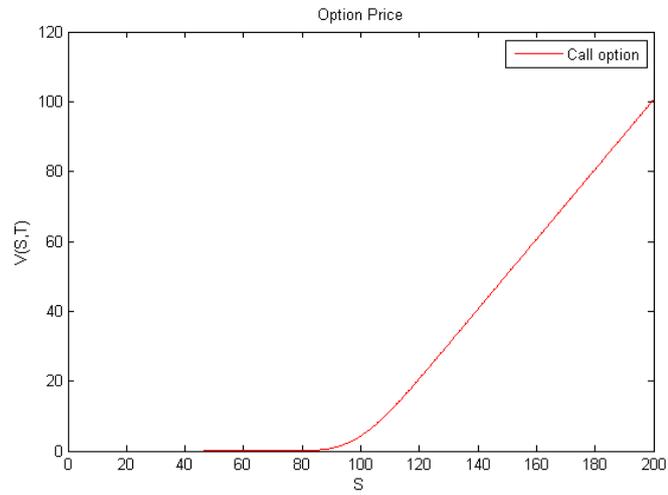


Figure 4.1: Exact solution of the European vanilla Call option.

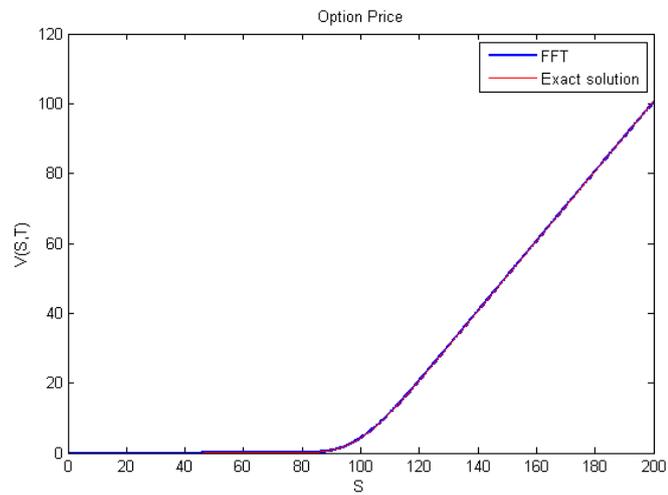


Figure 4.2: European Call obtained by the FFT approach.

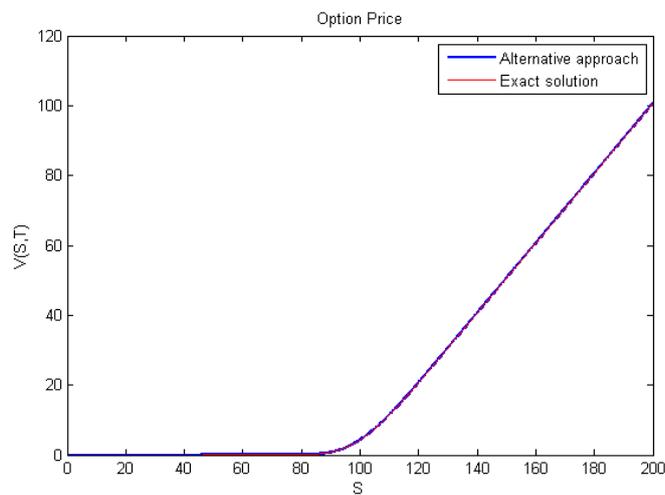


Figure 4.3: European Call obtained by the numerical method with using alternative approach.

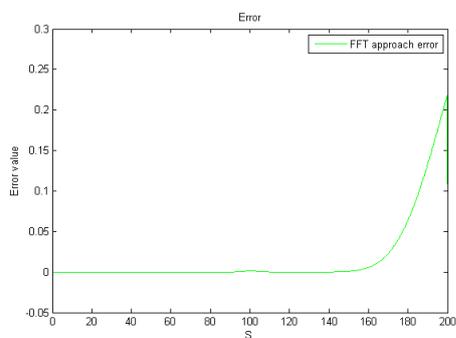


Figure 4.4: Error of evaluating the European Call option with using FFT approach.

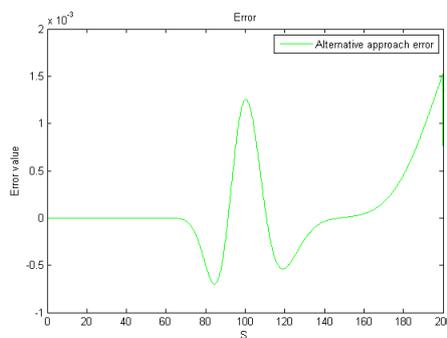


Figure 4.5: Error of evaluating the European Call option with using alternative approaches.

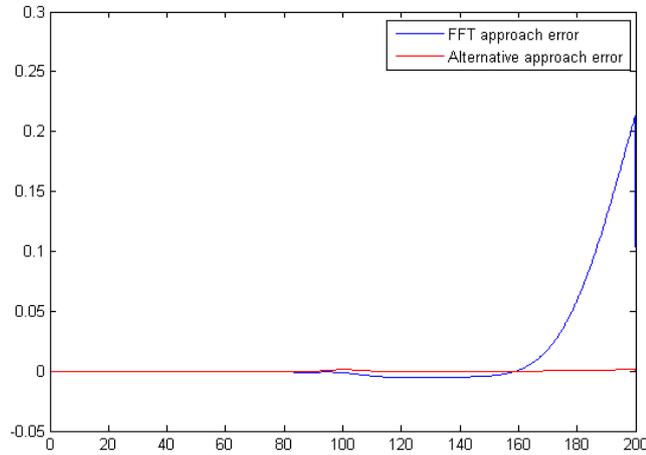


Figure 4.6: Two errors of evaluating the European Call option.

Figure 4.6 shows a comparison of two errors of numerical solutions for the call option obtained by two different approaches, by alternative and by FFT.

Numerical results for European vanilla Put option.

The Figure 4.7 demonstrates the exact solution of the European vanilla Put option. On the Figure 4.8 the solution of the European Put obtained by the numerical method with using FFT approach is presented. On the Figure 4.9 we can see the solution of the same option, but obtained by using the numerical method with the alternative approach. Figures 4.10 and 4.11 show the error of evaluating the option with using different approaches.

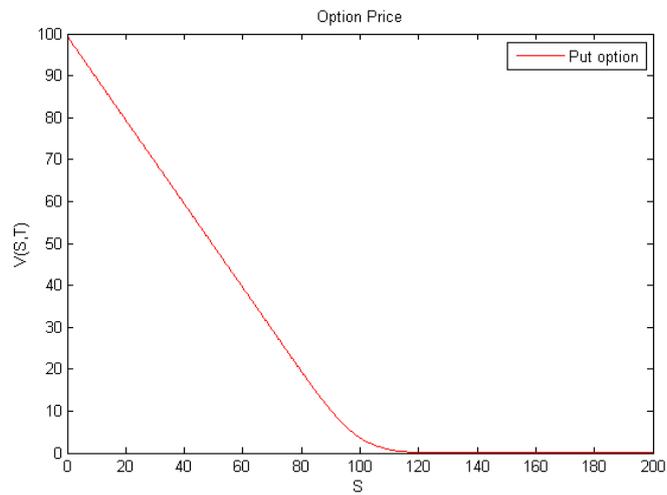


Figure 4.7: Exact solution of the European vanilla Put option.

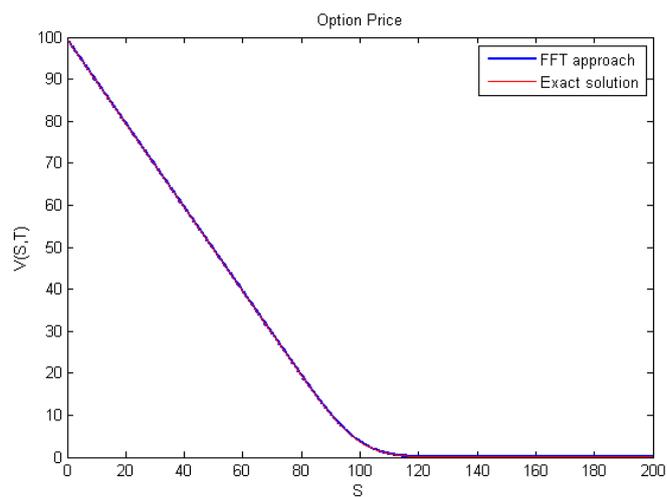


Figure 4.8: European Put obtained by the FFT approach.

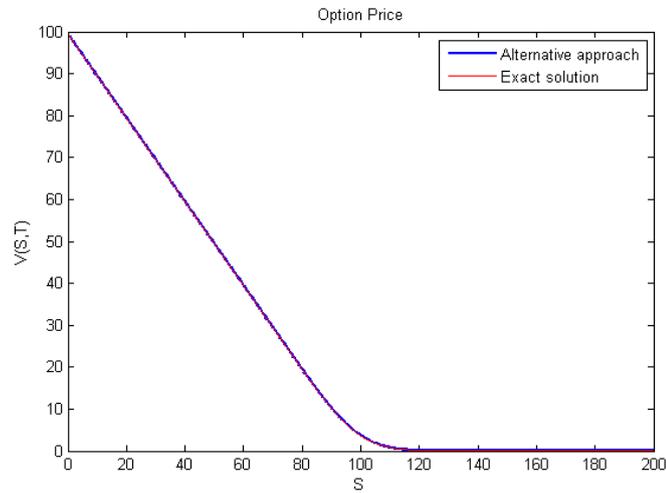


Figure 4.9: European Put obtained by the numerical method with using alternative approach.

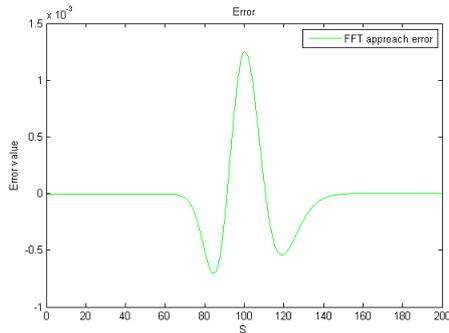


Figure 4.10: Error of evaluating the European Put option with using FFT approach.

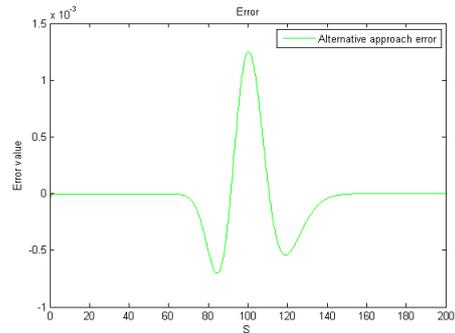


Figure 4.11: Error of evaluating the European Put option with using alternative approaches.

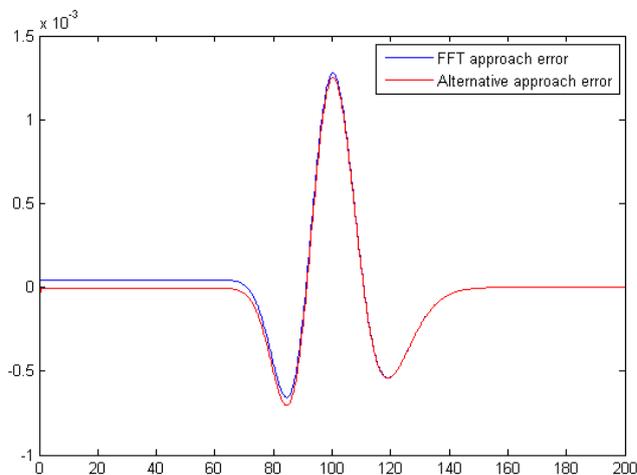


Figure 4.12: Two errors of evaluating the European Put option.

Figure 4.12 shows a comparison of two errors of numerical solutions for the put option obtained by two different approaches, by alternative and by FFT. The figure shows that these two methods accurately approximate the numerical solution to the exact one. However, an alternative approach approximates a bit better than FFT.

The accuracy of the approximated solution obtained by an alternative method, explicitly depends on the choice of algorithms and their implementation. And since the considered model is with jump diffusion processes, then methods are based on solving the integral term. It needs much less time to obtain the numerical solution using FFT, and the obtained solution differs from the exact one. If we use an alternative method then we need much more time to get the result. Despite that an alternative method approximates much better than FFT, it requires more time to get the numerical solution.

Chapter 5

Conclusions

In this thesis we consider basic methods for finding the solution of the partial integro–differential equations. Here both implicit and explicit methods were described. It is done as the explicit method for the valuation of the correlation integral with the implicit PDE discretization becomes stable. But as this method has the first order of accuracy then it is better to use an implicit method and in this case there is decided to use the Crank–Nicolson method. There is an algebraic stability of this method was proved, and the stability by the norm l_2 is proved also when grid S is equally distributed.

We suppose that the using of implicit time steps will lead to the fact that the direct estimation of the correlation integral is included in PIDE and then the solution of the dense matrix is required. In order to avoid these calculations we use the fixed–point iteration method. It converges very quickly, and each fixed point requires the correlation integral estimation.

Next we use the FFT approach to evaluate the correlation integral present an alternative approach by the linear function.

As the given method is developed for financial tools with jump diffusion processes then it suites for a finding the price for American and exotic options also.

The described methods were applied to find the price of European vanilla options and programmed in MATLAB. Finally, all numerical results were presented and concisely.

Notation

$V(S, t)$	Price of option
S	Price of an asset
S_0	Initial price of an asset
t	Forward time
T	Maturity date
τ	Reversal time
$r(t)$	Interest rate
$\sigma(t)$	Volatility
K	Strike price
N	Number of the time steps
$\mu(t)$	Drift rate
dW	Wiener process
dp	Poisson process
$\lambda(t)$	Poisson arrival intensity
$k(t)$	Expected relative jump size
$q(t)$	Impulse function
$g(q(t))$	Probability Density Function of $q(t)$
ν	Mean
ϱ^2	Variance
$E[\cdot]$	Expectation operator
θ	Weight parameter
$\gamma(j), \psi_{\gamma(j)}, \phi_{\Pi(k)}$	Interpolation weights

Bibliography

- [1] [http://en.wikipedia.org/wiki/Option-\(finance\)](http://en.wikipedia.org/wiki/Option-(finance)) (2011)
- [2] <http://en.wikipedia.org/wiki/Stochastic-process> (2011)
- [3] B. Eraker, M. Johannes and N. Polson (2003)
The impact of jumps in volatility and returns. J. Finance, **58**, 1269 – 1300.
- [4] R. C. Merton (1976)
Option pricing when underlying stock returns are discontinuous. J. Financial Econ., **3**, 125 – 144.
- [5] K. I. Amin (1993)
Jump diffusion option valuation in discrete time. J. Finance, **48**, 1833 – 1863.
- [6] <http://http://en.wikipedia.org/wiki/Probability-density-function> (2011)
- [7] <http://www.ems-i.com/gmshelp/Interpolation/Interpolation-Schemes/Inverse-Distance-Weighted/Computation-of-Interpolation-Weights.htm> (2011)
- [8] M. B. Giles (1997)
On the stability and convergence of discretizations of initial value PDEs. IMA J. Numer. Anal., **17**, 563 – 576.
- [9] <http://en.wikipedia.org/wiki/Backward-differentiation-formula> (2011)
- [10] X. L. Zhang (1997)
Numerical analysis of American option pricing in a jump-diffusion model. Math. Oper. Res., **22**, 668 – 690.
- [11] <http://en.wikipedia.org/wiki/Crank-Nicolson-method> (2011)

- [12] <http://en.wikipedia.org/wiki/Finite-difference-method> (2011)
- [13] J. Andreasen and B. Gruenewald (1996)
American Option Pricing in the Jump-Diffusion Model. Working Paper, (<http://www.cmap.polytechnique.fr/~rama/dea/andreasen.pdf>), Aarhus University and University of Mainz.
- [14] <http://en.wikipedia.org/wiki/Radon-Nikodym-theorem> (2011)
- [15] J. Andreasen (2000)
American Option Pricing in the Jump-Diffusion Model. Review of Derivatives Research, **4**, 231–262, .
- [16] A. Mitchell, and D. F. Griffiths (1980)
The Finite Difference Method in Partial Differential Equations. Wiley and Sons.
- [17] Y. d’Halluin, P.A. Forsyth and K.R. Vetzal (2005)
Robust Numerical Methods for Contingent Claims under Jump Diffusion Processes. IMA J. on Num. Anal., **25**, 87–112.
- [18] Y. d’Halluin, P.A. Forsyth and K.R. Vetzal (2004)
Numerical Methods for Real Options in Telecommunications. Working Paper, (<http://uwspace.uwaterloo.ca/bitstream/10012/1206/1/ydhallui2004.pdf>), Waterloo, Ontario, Canada.
- [19] A. Mayo (2008)
Methods for the rapid solution of the pricing PIDEs in exponential and Merton models. J. Comput. Appl. Math., **222**, 128–143.
- [20] C. Johnson (1987)
Numerical Solutions of Partial Differential Equations by the Finite Element Method. Cambridge University Press.
- [21] C. Anderson and M.D. Dahleh (1996)
Rapid computation of the discrete Fourier transform. SIAM J.Sc.Comp., **17**, 913-919.
- [22] A. Almendral and C.W. Oosterlee (2005)
Numerical valuation of options with jumps in the underlying. Appl. Numer. Math., **57**, 1–18.

- [23] C.S. Clift and P.A. Forsyth (2008)
Numerical Solution of Two Asset Jump Diffusion Models for Option Valuation. Appl. Numer. Math., **58**, 743–782.
- [24] C. Chiarella, B. Kang, G.H. Meyer and A. Ziogas (2009)
The Evaluation of American Option Prices Under Stochastic Volatility and Jump-Diffusion Dynamics Using the Method of Lines. Int. J. of Theor. and Appl. Finance, **12**, 393–425.
- [25] W.H. Hundsdorfer and J.G. Verwer (2003)
Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer.
- [26] S. Ikonen and J. Toivanen (2007)
Componentwise splitting methods for pricing American options under stochastic volatility. Int. J. Theor. Appl. Fin., **10**, 331–361
- [27] R. Lord, F. Fang, F. Bervoets and C.W. Oosterlee (2008)
A Fast and Accurate FFT-Based Method for Pricing Early-Exercise Options under Levy Processes. SIAM J. Sci. Comp., **30**, 1678–1705.

Appendix

In this part of thesis the theorems and the code listing are given.

Theorem 1. [17] (*The Basic concepts about stability of the scheme*) *The discretization method in the equation (20) is unconditionally stable for any choice of θ , $0 \leq \theta \leq 1$, provided that*

(i) $\alpha_i, \beta_i \geq 0$;

(ii) the discrete probability density \bar{f}_j has the properties (13);

(iii) the interpolation weights satisfy $0 \leq \phi_i \leq 1$, $0 \leq \psi_i \leq 1$;

(iv) $r, \lambda \geq 0$.

Proof. The proof of the theorem was presented in d'Halluin [17]. Let us provide the text of the proof.

Let $V^n = [V_0^n, V_1^n, \dots, V_p^n]'$ be the discrete solution vector to (20). Suppose the initial solution vector is perturbed, i.e

$$\widehat{V}^0 = V^0 + E^0,$$

where $E^n = [E_0^n, \dots, E_p^n]'$ is the perturbation vector. Note that $E_p^0 = 0$ since Dirichlet boundary conditions are imposed at this node. Then we obtain the following equation for the propagation of the perturbation (noting that ξ is a linear operator):

$$\begin{aligned} & E_i^{n+1}[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau] - \Delta\tau\beta_i E_{i+1}^{n+1} - \Delta\tau\alpha_i E_{i-1}^{n+1} \\ = & E_i^n + (1-\theta)\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \xi(E^{n+1}, i, j)\bar{f}_j\Delta y + \theta\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \xi(E^n, i, j)\bar{f}_j\Delta y. \end{aligned}$$

Defining

$$\|E\|_\infty^n = \max_i |E_i|^n.$$

We also know $\alpha_i, \beta_i \geq 0$ that

$$|E_i^{n+1}|[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau] \leq (\Delta\tau\beta_i + \Delta\tau\alpha_i)\|E\|_\infty^{n+1}$$

$$+\|E\|_\infty^n + (1 - \theta)\Delta\tau\lambda\|E\|_\infty^{n+1} + \theta\Delta\tau\lambda\|E\|_\infty^n.$$

Now, valid for all $i < p$. In particular, it is true for node i^* , where

$$\max_i |E_i^{n+1}| = |E_{i^*}^{n+1}|.$$

Now $i = i^*$ and we can write down an expression as

$$\|E\|_\infty^{n+1}[1 + (r + \theta\lambda)\Delta\tau] = \|E\|_\infty^n(1 + \theta\Delta\tau\lambda),$$

and thus

$$\begin{aligned} \|E\|_\infty^{n+1} &\leq \|E\|_\infty^n \frac{(1 + \theta\Delta\tau\lambda)}{(1 + (r + \theta\lambda)\Delta\tau)} \\ &\leq \|E\|_\infty^n. \end{aligned}$$

□

Theorem 2. [18] (*Algebraic stability of Crank-Nicolson timestepping*) *The Crank-Nicolson discretization (24) is algebraically stable in the sense that*

$$\|(B)^n\|_\infty \leq Cn^{1/2}, \quad \forall n, p,$$

where C is independent of n, p .

Theorem 3. [17] (*Convergence of the fixed-point iteration*) *Provided that*

(i) $\alpha_i, \beta_i \geq 0$,

(ii) *the discrete probability density \bar{f}_j has the properties (13),*

(iii) *the interpolation weights satisfy $0 \leq \phi_i \leq 1, \quad 0 \leq \psi_i \leq 1$,*

(iv) $r, \lambda \geq 0$, *if properties are executed, iterations with the fixed point of expression (34) will be global converges, and the maximum error on each iteration should satisfy necessarily to a condition*

$$\|e^{k+1}\|_\infty \leq \|e^k\|_\infty \frac{(1 - \theta)\lambda\Delta\tau}{1 + (1 - \theta)(r + \lambda)\Delta\tau}.$$

Proof. The proof of the theorem was presented in paper [17] From (34) where e^k , it is satisfied the following

$$[I - (1 - \theta)\bar{M}]e^{k+1} = (1 - \theta)\lambda\Delta\tau\Phi(e^k).$$

Using the proof of the Theorem 1 5 we get

$$\|e^{k+1}\|_\infty \leq \|e^k\|_\infty \frac{(1 - \theta)\lambda\Delta\tau}{1 + (1 - \theta)(r + \lambda)\Delta\tau}$$

$$< \|e^k\|_\infty.$$

It is important that $\lambda\Delta\tau \ll 1$, such that

$$\|e^{k+1}\|_\infty \simeq \|e^k\|_\infty(1 - \theta)\lambda\Delta\tau,$$

It leads to fast convergence of the iteration. It is also necessary to notice that number of iterations that are needed for convergence does not depend on number of points in grid S . \square

Proposition 1. [15] *The following properties hold for the scheme (40),(41):*

(i) *The scheme is unconditionally stable in the von Neumann sense.*

(ii) *For the case of deterministic parameters, the numerical solution of the scheme is locally accurate to order $O(\Delta\tau^2 + \Delta x^2)$*

(iii) *If M is the number of time steps and N is the number of steps in the spatial direction, the computational burden is $O(MN\log_2 N)$*

Proof. The proof of the proportion was presented in J. Andreasen [15]. Let us provide the text of the proof. We first consider the vonNeumann stability. Inserting $u_1(\tau, x) = v_1^{-\tau} e^{ikx}$ into (40) and $u_2(\tau, x) = v_2^{-\tau} e^{ikx}$ and (41) where v_1, v_2 are complex numbers, yields

$$v \equiv v_1^{\frac{\Delta\tau}{2}} v_2^{\frac{\Delta\tau}{2}} = \frac{[\frac{-2}{\Delta\tau} + (a\delta_x + \frac{1}{2}b^2\delta_{xx} - r)]e^{ikx}}{[\frac{-2}{\Delta\tau} - (a\delta_x + \frac{1}{2}b^2\delta_{xx} - r)]e^{ikx}} \cdot \frac{\frac{2e^{ikx}}{\Delta\tau} - \lambda e^{ikx} + \lambda \sum_j q_j(x) e^{ikx\Delta x}}{\frac{2e^{ikx}}{\Delta\tau} + \lambda e^{ikx} + \lambda \sum_j q_j(x) e^{ikx\Delta x}}.$$

For a von Neumann criteria [16] it is necessary, that $|v| \leq 1$ was satisfied for all k . We have $v = A(k) \cdot B(k)$, where

$$A(k) = \frac{\frac{2}{\Delta\tau} - r - \frac{b^2}{\Delta x^2}(1 - \cos(k\Delta x)) + i\frac{a}{\Delta x}\sin(k\Delta x)}{\frac{2}{\Delta\tau} + r + \frac{b^2}{\Delta x^2}(1 - \cos(k\Delta x)) - i\frac{a}{\Delta x}\sin(k\Delta x)}$$

$$B(k) = \frac{\frac{2}{\Delta\tau} - \lambda(1 - \sum_j q_j(x)\cos(kj\Delta x)) + i\lambda\sum_j q_j(x)\sin(kj\Delta x)}{\frac{2}{\Delta\tau} + \lambda(1 - \sum_j q_j(x)\cos(kj\Delta x)) - i\lambda\sum_j q_j(x)\sin(kj\Delta x)}.$$

A geometric argument shows that $|A(k)| \leq 1$ when $r \geq 0$ for all k . Noting that

$$\sum_j q_j(x) \cos(kj\Delta x) \leq \sum_j q_j(x) = 1,$$

Also we have $|B(k)| \leq 1$ for all k . And we can conclude now that the scheme is unconditionally stable.

Now it is necessary to consider an accuracy of the given scheme

$$V(s) = \left[\sum_{n=0}^{\infty} \frac{(s-n)^n}{n!} \left(\frac{\partial}{\partial \tau} \right)^n \right] V(\tau) = e^{(s-\tau) \frac{\partial}{\partial \tau}} V(\tau).$$

Now we get

$$V[(\zeta^* - 1)\lambda + D + \frac{\partial}{\partial \tau}] = 0.$$

We can rewrite it as

$$e^{-\frac{\Delta t}{2} D} V\left(\tau + \frac{\Delta \tau}{2}\right) = e^{\frac{\Delta \tau}{2} \lambda (\zeta^* - 1)} V(\tau) = e^{\frac{\Delta \tau}{2} D} V\left(\tau + \frac{\Delta \tau}{2}\right).$$

The exponent expansion gives us

$$\begin{aligned} & \left[1 - \frac{1}{2} \Delta \tau D + \frac{1}{2} \left(\frac{\Delta \tau}{2} \right)^2 D^2 \right] V\left(\tau + \frac{\Delta \tau}{2}\right) = \\ & \left[1 + \frac{1}{2} \Delta \tau \lambda (\zeta^* - 1) + \frac{1}{2} \left(\frac{\Delta \tau}{2} \right)^2 \lambda^2 (\zeta^* - 1)^2 \right] V(\tau + \Delta \tau) + O(\Delta \tau^3), \\ & \left[1 - \frac{1}{2} \Delta \tau \lambda (\zeta^* - 1) + \frac{1}{2} \left(\frac{\Delta \tau}{2} \right)^2 D^2 \right] V\left(\tau + \frac{\Delta \tau}{2}\right) + O(\Delta \tau^3). \end{aligned}$$

We can notice that for any analytical function F , we have following expression,

$$\bar{D}f(x) = Df(x) + O(\Delta x^2), \bar{\zeta}^* f(x) = \zeta^* f(x) + O(\Delta x^2).$$

And now such representation will give us two new equations

$$\begin{aligned} & V\left(\tau + \frac{\Delta \tau}{2}\right) \left[1 - \frac{1}{2} \Delta \tau \bar{D} \right] = \left[1 + \frac{1}{2} \Delta \tau \lambda (\zeta^* - 1) \right] V(\tau + \Delta \tau) \\ & + \frac{1}{2} \left(\frac{1}{2} \Delta \tau \right)^2 \left(-\bar{D}^2 V\left(\tau + \frac{\Delta \tau}{2}\right) + \lambda^2 (\bar{\zeta}^* - 1)^2 V(\tau + \Delta \tau) \right) + O(\Delta \tau \Delta x^2 + \Delta \tau^3) \\ & V(\tau) \left[1 - \frac{1}{2} \Delta \tau \lambda (\bar{\zeta}^* - 1) \right] = \left[1 + \frac{1}{2} \Delta \tau \bar{D} \right] V\left(\tau + \frac{\Delta \tau}{2}\right) \\ & + \frac{1}{2} \left(\frac{1}{2} \Delta \tau \right)^2 \left(\bar{D}^2 V\left(\tau + \frac{\Delta \tau}{2}\right) - \lambda^2 (\bar{\zeta}^* - 1) V(\tau) \right) + O(\Delta \tau \Delta x^2 + \Delta \tau^3). \end{aligned}$$

Substituting the first equation in the second one it can be written as

$$\begin{aligned}
[1 - \frac{1}{2}\Delta\tau\lambda(\bar{\zeta}^* - 1)]V(\tau) &= [1 + \frac{1}{2}\Delta\tau\bar{D}][1 - \frac{1}{2}\Delta\tau\bar{D}]^{-1}[1 + \frac{1}{2}\Delta\tau\lambda(\bar{\zeta}^* - 1)]V(\tau + \Delta\tau) \\
&+ [1 + \frac{1}{2}\Delta\tau\bar{D}][1 + \frac{1}{2}\Delta\tau\bar{D}]^{-1}\frac{1}{2}(\frac{1}{2}\Delta\tau)^2(-\bar{D})^2V(\tau\frac{\Delta\tau}{2}) + \lambda^2(\bar{\zeta}^2 - 1)^2V(\tau + \Delta\tau) \\
&+ \frac{1}{2}(\frac{1}{2}\Delta\tau)^2(\bar{D}^2V(\tau + \frac{\Delta\tau}{2}) - \lambda^2(\bar{\zeta}^* - 1)^2V(\tau)) + O(\Delta\tau\Delta x^2\Delta\tau^3),
\end{aligned}$$

where the term $[1 - \frac{1}{2}\Delta\tau\bar{D}]^{-1}$ should be interpreted in the sense of matrix inversion. We now use the two observations

$$[1 - \frac{1}{2}\Delta\tau\bar{D}]^{-1} = 1 + O(\Delta\tau),$$

$$V(\tau + \Delta\tau) = V(\tau) + O(\Delta\tau).$$

From this we have

$$[1 - \frac{1}{2}\Delta\tau\lambda(\bar{\zeta}^* - 1)]V(\tau) = [1 + \frac{1}{2}\Delta\tau\bar{D}]^{-1}[1 + \frac{1}{2}\Delta\tau\lambda(\bar{\zeta}^* - 1)]V(\tau + \Delta\tau) + O(\Delta\tau\Delta x^2 + \Delta\tau^3).$$

Now we can rewrite (40) and (41). The local approximation error of schemes (40) and (41) is $O(\Delta\tau\Delta x^2 + \Delta\tau^3)$ and thus the scheme has $\Delta\tau^2 + \Delta x^2$ order of accuracy. \square

The programme in MATLAB.

Function f, equation (12).

```
function f=funcf(param,deltax,mean)

f=0.8862269255*(erf(1.225030476*deltax*param+.6125152382*deltax-...
    1.225030476*mean)-erf(1.225030476*deltax*param-.6125152382*deltax-...
    1.225030476*mean))/sqrt(pi);
```

Main method (34).

```
close all
clear all

M=801; % space points
N=400; % time points

r=0.2; % interest rate
vol=0.3; % volatility
T=0.25; % maturity time
K=100; % strike price
w=0.5; % weighting coefficient
lam=1; % Poisson arrival intensity
nu=0; % mean
g=1; % variance
phi=0.5;
psi=0.5;

% grid
Smax=200;
Xmax=log(Smax);
dx=Xmax/M;
x=0:dx:Xmax;
Si=1:M+1;
S=exp(x);
dt=T/N;
t=0:dt:T;
ti=1:N+1;
```

```

dy=dx;

% matrices and coefficients
E=speye(M-1);
tol=0.01; %tolerance
q=exp(nu+0.5*g^2)-1; %equation on page 4

% option
V=zeros(M+1,N+1);
%PUT
V(Si,1)=max(K-S(Si),0); %payoff t=T or tau=0
V(1,ti)=K*exp(-r*t(ti)); %boundary condition S=0
V(M+1,ti)=0; %boundary condition S=Smax

% alpha and beta, eqs.(21),(22),(23) and algorithm on page 7
for i=2:M
    ac=(((vol^2)*S(i)^2)/((S(i)-S(i-1))*(S(i+1)-S(i-1))))-...
        (((r-lam*q)*S(i))/(S(i+1)-S(i-1)));
    bc=(((vol^2)*S(i)^2)/((S(i+1)-S(i))*(S(i+1)-S(i-1))))+...
        (((r-lam*q)*S(i))/(S(i+1)-S(i-1)));
    bf=(((vol^2)*S(i)^2)/((S(i+1)-S(i))*(S(i+1)-S(i-1))))+...
        (((r-lam*q)*S(i))/(S(i+1)-S(i)));
    if ((ac>=0) && (bc>=0))
        a(i-1)=ac;
        b(i-1)=bc;
    elseif (bf>=0)
        a(i-1)=((vol^2)*S(i)^2)/((S(i)-S(i-1))*(S(i+1)-S(i-1))) ;
        b(i-1)=bf;
    else
        a(i-1)=(((vol^2)*S(i)^2)/((S(i)-S(i-1))*(S(i+1)-S(i-1))))-...
            (((r-lam*q)*S(i))/(S(i)-S(i-1))) ;
        b(i-1)=((vol^2)*S(i)^2)/((S(i+1)-S(i))*(S(i+1)-S(i-1))) ;
    end
end

end

% main method on page 10
D=zeros(M-1,M-1); % matrix D
for i=2:M-1
    D(i-1,i-1) = -dt*(a(i-1)+b(i-1)+r+lam);
    D(i-1,i) = dt*b(i-1);

```

```

        D(i,i-1)    = dt*a(i);
    end
    D(M-1,M-1) = -dt*(a(M-1)+b(M-1)+r+lam);

% main time loop
for j=2:N+1
    z=1;

    % FFT approach
    %F(1:M-1)=funcf(1:M-1,dx,nu);
    %OmV(1:M-1)=ifft(fft(V(2:M,j-1)).*conj(fft(F')));

    % linear function approach
    for h=1:M-1
        temp=0;
        for i=2-h:M-h-2
            temp=temp+funcf(i,dx,nu)*(phi*(psi*V(h+i,j-1) + ...
                (1-psi)*V(h+i+1,j-1)) + (1-phi)*...
                (psi*V(h+i+1,j-1) + (1-psi)*V(h+i+2,j-1)));
        end
        OmV(h)=temp;
    end

    % inner loop until convergence
    U(1:M-1,z)=V(2:M,j-1);

    while (z>0)
        %FFT approach
        %OmU(1:M-1)=ifft(fft(U(1:M-1,z)).*conj(fft(F')));

        % linear function approach
        for h=1:M-1
            temp=0;
            for i=1-h:M-h-3
                temp=temp+funcf(i,dx,nu)*(phi*(psi*U(h+i,z) + ...
                    (1-psi)*U(h+i+1,z)) + (1-phi)*...
                    (psi*U(h+i+1,z) + (1-psi)*U(h+i+2,z)));
            end
            OmU(h)=temp;
        end
    end
end

```

```

        % system matrix
        A = E-(1-w)*D;
        B = (E+w*D)*V(2:M,j-1) + (1-w)*lam*dt*OmU' + w*lam*dt*OmV';
        U(1:M-1,z+1)=A\B;

        %condition for convergence
        num=abs(U(1:M-1,z+1)-U(1:M-1,z));
        denum=max(1,U(1:M-1,z+1));
        cond=max(num./denum);
        if (cond<tol)
            break
        else
            z=z+1;
        end
    end
    V(2:M,j)=U(1:M-1,z+1);

end % for j

% Exact solution
Vex=zeros(M+1,1);
coeff = r+0.5*vol^2;
for i=1:M+1
    d1=(log(S(i)/K)+coeff*T)/(vol*sqrt(T));
    d2=d1-vol*sqrt(T);
    D1=normcdf(d1,0,1);
    D2=normcdf(d2,0,1);
    %Put
    Vex(i,1)=K*exp(-r*T)*(1-D2)-S(i)*(1-D1);
end

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

% plot error

```

```
%plot(S,abs(Vex(:,1)-V(:,N+1)),'-g')  
%xlabel ('S')  
%ylabel ('Error value')  
%title ('Error')
```