Methods for Calculating Option Prices with Early-Exercise Features

by

Michael Cooney

B.A. (Mod.) (Hons.), MSc (High Performance Computing)

A Thesis submitted to
The University of Dublin
for the degree of

Doctor in Philosophy

Department of Mathematics
University of Dublin
Trinity College

October, 2005

Declaration

This thesis has not been submitted as an exercise for a degree at any other University. Except where otherwise stated, the work described herein has been carried out by the author alone. This thesis may be borrowed or copied upon request with the permission of the Librarian, University of Dublin, Trinity College.

Signature of Author ....Michael Cooney........

Michael Cooney

31 October, 2005

Abstract

In this dissertation we deal with two distinct methods for pricing financial options with early-exercise features. First we use finite difference methods to calculate the prices, examining in particular two new schemes designed to deal with problems where the problems becomes singly perturbed. The second method is MonteCarlo techniques which allow for the early exercise of the option using different techniques to determine whether or not it is optimal to exercise early. Two new algorithms in particular were developed, one which uses an interpolation method to calculate the expected payoffs, the other uses an iterative technique and Ito's Lemma to determine if the option should be exercised.

Summary

In this dissertation we discuss various methods used to price financial option contracts with early exercise features.

Three main methods are used to price these financial derivatives: finite difference solutions to the Black-Scholes PDE, MonteCarlo simulation, and tree methods. This dissertation discusses the first two approaches, but gives a passing reference to the tree methods used.

In Chapter 1 we discuss the Black-Scholes model, and introduce the field of stochastic calculus. We derive the Black-Scholes PDE and discuss the importance of the various partial derivatives of the option price (collectively known as the 'Greeks').

In Chapter 2 we introduce partial differential equations, and how this applies to the Black-Scholes PDE, paying particular attention to the boundary and initial/terminal conditions that arise as a result.

In Chapter 3 we discuss stochastic calculus, Brownian motion and function of stochastic variables. We derive Ito's Lemma, which plays a vital role in the derivation of the Black-Scholes equation, and discuss how the theory of stochastic mathematics and calculus applies to financial theory.

In Chapter 4 we derive an exact solution of the Black-Scholes equation for European options (contracts that do not have early-exercise features).

In Chapter 5 we introduce the finite difference approach to solving the problem of pricing the American option. We discuss five numerical schemes and discuss the results of the five for the solution of both European and American options. We look at the problem of singly-perturbed systems in particular, and discuss how the schemes perform under these parameters.

In Chapter 6 we introduce the MonteCarlo approach to pricing the options and discuss different methods used to capture the early-exercise feature of the American option pricing problem. Two algorithms in particular are produced, and the results of these are compared to existing methods for solving this problem.

In Chapter 7 we introduce the tree model approach, and discuss a few methods for solving options.

Chapter 8 is the conclusion, where we summarise all the work and compare results between the different methods.

Acknowledgements

I would like to acknowledge the following people for the help in this thesis.

Jim Sexton, my supervisor, for all his help and advice.

Karen O'Doherty, (or should that be Moroney?) for making all the irritating admin work easy.

Justin, **Alan**, **Derek**, **Dennis**, **Kev**, and all the others who had the dubious pleasure of sharing an office with me during the years.

John Butler, for invaluable advice on finite difference error analysis, and generally helping to calm me down. **Steve Watterson**, also for keeping me calm.

Niamh Black, aka the **Flaming Redhead**, for her tea and her laughs during a hard month.

Paul, **Steve**, **Des** and **John**, aka **The Lads**, for all their chronic lack of support and appalling abuse.

Ciaran, for his editing, and being one of **The Lads** of course.

Emma, my sister, for always being there and letting me be an uncle.

Finally, as always, **Michael** and **Patricia**, my parents, for all the phone calls, the help, the money and the food. I will never be in a position to repay a fraction of what I owe you.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This chapter defines the financial instruments for which we wish to develop pricing methodologies. These instruments include:

- European Options
- American Options
- Asian Options
- Barrier Options
- Path-dependant Options

After these definitions, a summary of the various pricing methodologies is provided. Forward references are included to later chapters in which these pricing methodologies are discussed.

1.1 Pricing Methodologies

There are a number of different models used to calculate the price of an option. Probably the most common and widely used model is the Black-Scholes model.

What follows is a brief description of the most common option-price models, with the more important models being treated more comprehensively later.

1.1.1 The Binomial Model

The basic concept for the binomial model is simple. Taking the current value of the underlying asset (often abbreviated to just the **underlying**), we discretise the lifetime of the option into a number of timesteps, allowing the value of the asset to either increase or decrease by a set ratio at each timestep. At expiry, we have a number of different possible values for the underlying, which are straightforward to calculate.

Having found all these possible values at expiry, we can then calculate the value of the option, since the value of an option at expiry is simply the payoff of the option. We can then use these set of values for the option to calculate the values at the previous time-step, and so on back to the initial option price.

1.1.2 The Black-Scholes Model

The Black-Scholes model [2] takes a similar approach, but produces a much more general result. The value of the underlying is modeled as a lognormal random walk, and the risk-free interest rate is assumed to be a known function of time. Using the no-arbitrage principle and some stochastic calculus theory, a partial differential equation is derived.

The Black-Scholes equation for a the price of a vanilla call or put, V, is given by:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{1.1}$$

The boundary conditions for this problem are dependent on the option being studied.

The vast majority of this dissertation will deal with the Black-Scholes Model.

1.2 The Black-Scholes Model

The price of the option depends on a number of variables and parameters, including the price of the underlying asset, S; the lifetime of the asset, T; the current time, t; the volatility of the returns σ; and the risk-free interest rate r.

We first list some basic assumptions made by the following model:

- The underlying follows a lognormal random walk.
- The risk-free interest rate is a known function of time (and is assumed constant for the moment).
- There are no dividends on the underlying (dividends are easily accounted for and are omitted for the sake of clarity).
- Delta-hedging can be performed continuously.
- There are no transaction costs on the underlying.
- There are no arbitrage opportunities.

Rather than use the unwieldy notation $V(S, t; \sigma, \mu; K, T; r)$, we shall just refer to V as a function of the underlying and time, $V(S, t)$.

Since a call option is a contract to buy the asset for a fixed price in the future, it should be immediately obvious that this contract increases in value as the value of the underlying increases.

We define Π to be the value of a portfolio consisting of a long option position and a short position of size Δ, **delta** of the asset:

$$\Pi = V(S, t) - \Delta S. \tag{1.2}$$

The first term on the RHS is the option, and the second term is the short asset position. It is negative because we have shorted the position, and the value of Δ is for the moment considered to be an arbitrary constant.

Assuming the underlying is following a log-normal random walk,

$$dS = \mu S \, dt + \sigma S \, dX, \tag{1.3}$$

we consider the change in value of the portfolio between time t and $t + \delta t$. The change in the portfolio value is due to changes in both the underlying and the option:

$$d\Pi = dV - \Delta\, dS. \tag{1.4}$$

Δ has not changed, but from Ito's Lemma [22], we have

$$dV = \frac{\partial V}{\partial t}\, dt + \frac{\partial V}{\partial S}\, dS + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\, dt, \tag{1.5}$$

and substituting Eq. (1.5) into Eq. (1.4), we get

$$d\Pi = \frac{\partial V}{\partial t}\, dt + \frac{\partial V}{\partial S}\, dS + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\, dt - \Delta\, dS. \tag{1.6}$$

We can see that Eq. (1.6) has both deterministic

$$\frac{\partial V}{\partial t}\, dt, \quad +\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\, dt, \tag{1.7}$$

and stochastic terms,

$$\frac{\partial V}{\partial S}\, dS, \quad -\Delta\, dS, \tag{1.8}$$

but we can never know the exact value of the stochastic terms. By its very nature, the value of dS is random.

Grouping the stochastic terms together, we get

$$\left(\frac{\partial V}{\partial S} - \Delta\right) dS. \tag{1.9}$$

It immediately becomes obvious that equating the two terms will eliminate the stochastic terms from Eq. (1.6) and allow us to deterministically calculate the value of $d\Pi$:

$$\Delta = \frac{\partial V}{\partial S}. \tag{1.10}$$

We now have

$$d\Pi = \left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right) dt. \tag{1.11}$$

Since Eq. (1.11) contains only deterministic terms, it is completely **riskless**. However, a risk-free change in a portfolio value over a time period must be equal to the change in an equivalent investment in a risk-free interest-bearing account, as otherwise there are arbitrage opportunities:

$$d\Pi = r\Pi \, dt. \tag{1.12}$$

This is because were the portfolio return to be greater than the interest return, we could borrow from the bank at rate r, invest in the portfolio and then pay off the loan to make a guaranteed profit. Were the return to be lower, we could short the option and invest the money in the bank to make a profit.

Either method guarantees a risk-free profit that exceeds the risk-free interest rate. Although this scenario is very possible in modern markets, such opportunities tend to be exploited very quickly by arbitrage traders. The natural consequence of this arbitrage trading is that prices move to close the arbitrage opportunity.

Returning to Eq. (1.12) and substituting the values for $d\Pi$ from Eq. (1.11), and Π from Eq. (1.2) and Eq. (1.10), we get

$$\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right) dt = r\left(V - S\frac{\partial V}{\partial S}\right) dt. \tag{1.13}$$

Some simple re-arranging gives us the **Black-Scholes Equation**:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{1.14}$$

It is a **linear, second order, parabolic partial differential equation** [42]. Almost all equations in finance are of a similar form, and are similar to heat and diffusion equations used in physics.

Note that the value of the drift, μ, has dropped out of Eq. (1.14), due to the fact that the hedging of asset has eliminated the dS term.

Initially, we assumed that Δ was constant. Now it is dependent on both the underlying and time. The result is that the value of Δ is constantly changing, and as a result, the portfolio must be constantly updated to preserve the equality of Eq.

(1.10).

This continuous-time process is referred to as **Dynamic Hedging**. Since our hedging strategy involves ensuring that the risk in the equation is removed by manipulation of the Δ, it is called **delta hedging**.

1.3 Boundary and Terminal Conditions

The Black-Scholes Equation is a general stochastic partial differential equation for options, and is valid for both call and put options, for any strike price.

These distinctions are made by the choice of boundary and terminal conditions used to solve the equation. The value of the option at expiry must always be equal to the payoff, and so we have for calls

$$C(S,T) = \max(S-K,0), \tag{1.15}$$

where K is the value of the strike price.

For puts,

$$P(S,T) = \max(K-S,0). \tag{1.16}$$

Also, when $S=0$, the value of the call option is worthless, since once the stock reaches zero it will never change. For puts, however, it is worth the discounted value of strike price. This is because, at expiry, we are guaranteed $P=K$:

$$C(0,T) = 0, \tag{1.17}$$

$$P(0,T) = Ke^{-r(T-t)}. \tag{1.18}$$

Conversely, for $S \to \infty$, the value of the put becomes worthless, whereas the value of the call approaches $S - Ke^{-r(T-t)}$:

$$C(\infty,t) \to S - Ke^{-r(T-t)}, \tag{1.19}$$

$$P(\infty,t) \to 0. \tag{1.20}$$

1.4 The 'Greeks'

1.4.1 Delta, Δ

The **Delta**, Δ, of an option is the sensitivity of its price to the value of the underlying:

$$\Delta = \frac{\partial V}{\partial S}. \tag{1.21}$$

For portfolios, the value of Delta, Δ, is just the sum of all the individual deltas for each instrument comprising the portfolio.

Delta-hedging is a practical technique for minimising the effect of changes in asset price on the value of the portfolio. It is a common practice of hedge traders.

In the real world, transaction costs limit the effectiveness of maintaining a **delta-neutral** position, since each change of portfolio incurs an overhead. In general, this problem is circumvented by trying to minimise the Delta, Δ, rather than maintaining a zero-value.

This provides a degree of certainty and immunity to changes in the market, while allowing the trader to avoid having to pay prohibitive transaction costs,

$$\Delta_C = N(d_1), \tag{1.22}$$

$$\Delta_P = N(d_1) - 1. \tag{1.23}$$

where

$$\begin{aligned} d_1 &= \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \\ d_2 &= \frac{\ln(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}. \end{aligned}$$

and $N(x)$ is the cumulative probability function for a normal distribution.

1.4.2 Gamma, Γ

The **Gamma**, Γ, of an option or portfolio is the second derivative of its value to the value of the underlying:

$$\Gamma = \frac{\partial^2 V}{\partial S^2}. \tag{1.24}$$

It is a measure of the sensitivity of the Delta to changes in the asset price and so gives an indication of how often and by how much a portfolio must be re-hedged to retain the delta-neutral position. Time does have an effect on this, but the dominant factor is the Brownian motion of the underlying.

In reality, re-hedging a portfolio incurs transaction costs, and so it is often desirable to have a portfolio which does not need a large amount of maintenance (this should also reduce the effect of model errors). In this case, it is common to construct **Gamma-neutral** portfolios, which seek to minimise the Gamma of the portfolio. This requires the incorporation of options into the portfolio.

$$\Gamma_C = \frac{N'(d_1)}{\sigma S\sqrt{T-t}}. \tag{1.25}$$

$$\Gamma_P = \frac{N'(d_1)}{\sigma S\sqrt{T-t}}. \tag{1.26}$$

1.4.3 Theta, Θ

The **Theta**, Θ, of an option or portfolio is the derivative of the option value to time:

$$\Theta = \frac{\partial V}{\partial t}. \tag{1.27}$$

The Theta is related to the Delta, Gamma and option price by the Black-Scholes equation.

In a Delta-hedged portfolio, the Theta contributes to ensuring that the portfolio earns the risk-free interest rate,

$$\Theta_C = -\frac{\sigma S N'(d_1)}{2\sqrt{T-t}} - rKe^{-r(T-t)}N(d_2). \tag{1.28}$$

$$\Theta_P = -\frac{\sigma S N'(-d_1)}{2\sqrt{T-t}} + rKe^{-r(T-t)}N(-d_2). \tag{1.29}$$

1.4.4 Vega

The **Vega** is the derivative of the option price to the underlying volatility, σ,

$$\text{Vega} = \frac{\partial V}{\partial \sigma}. \tag{1.30}$$

It is an unusual quantity in that, unlike the previous Greeks, it is a parameter derivative rather than a variable derivative (which usually makes it more difficult to find a numerical approximation of its value).

The volatility of the underlying asset is hard to quantify exactly in practice and can be a major source of model error. Thus, hedging a portfolio to make it **Vega-neutral** can go a long way to eliminating model-risk, as we are reducing the dependance of the model on a parameter whose value we cannot be truly confident in,

$$\text{Vega}_C = S\sqrt{T-t}\,N'(d_1), \tag{1.31}$$

$$\text{Vega}_P = S\sqrt{T-t}\,N'(d_1). \tag{1.32}$$

In this work, the volatility is assumed constant and so the Vega is zero.

1.4.5 Rho, ρ

The **Rho**, ρ, is the derivative of the option value to the interest-rate used in the Black-Scholes model,

$$\rho = \frac{\partial V}{\partial r}. \tag{1.33}$$

In practice, financial theory uses a term structure for interest rates which is time-dependent, $r(t)$. Rho is the sensitivity of the option price to the interest rate level:

$$\rho_C = K(T-t)e^{-r(T-t)}\,N(d_2), \tag{1.34}$$

$$\rho_P = -K(T-t)e^{-r(T-t)}\,N(-d_2). \tag{1.35}$$

In this work, the interest rate is assumed constant and so the Rho is zero.

Chapter 2

Partial Differential Equations

A **Partial Differential Equation (PDE)** is an equation that relates the partial derivatives of an unknown function of more than one variable. PDEs occur in all aspects of applied mathematics, and a large body of work has been amassed to try to solve the various types of PDE that occur.

A solution of the PDE is a functional form that satisfies the equation. A PDE usually has many (and possibly infinitely many) solutions that satisfy it. Thus, a particular problem often requires additional **boundary conditions** which further constrain the solution set.

Where **ordinary differential equations** (i.e equations that only contain ordinary derivatives) have solutions that are families that are characterised by some parameter value, solutions to PDEs are often parameterised by functions. Informally, this means that the solution set is much larger for PDEs.

The study of PDEs is a vast subject [42], and so the following discussion is limited to a few general types of PDE that occur in the financial problems that we are trying to solve.

A PDE is said to be **linear** if it can be rewritten in the form

$$A\mathbf{u} = f, \tag{2.1}$$

for some linear operator A and functions $\mathbf{u}$, and f.

The **order** of the PDE is the highest order of derivative that appears in the equation.

Some very commonly-occuring equations are:

$\nabla^2 \mathbf{u} = 0$	Laplace's Equation
$\nabla^2 \mathbf{u} = \mathbf{f}$	Poisson's Equation
$\dfrac{\partial^2 u}{\partial t^2} = c^2 \, \nabla^2 u, \ c$ constant	Wave Equation
$\dfrac{\partial u}{\partial t} = k \, \nabla^2 u, \ k$ constant	Diffusion (Heat) Equation

This discussion is limited to second-order linear PDEs, as most of option pricing theory results in this type of equation.

2.1 Classifying Second-Order PDEs

The general formula for a second-order, linear PDE is

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \, \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu + g = 0, \tag{2.2}$$

where a, b, c, d, e, f, and g are independant of x, y and u.

Second-order PDEs are classified by the relationship between the second-order derivative coordinates in the PDE. The equation is said to be:

Elliptic if $b^2 - 4ac < 0$,

Parabolic if $b^2 - 4ac = 0$,

Hyperbolic if $b^2 - 4ac > 0$.

Laplace's equation is a classic example of an elliptic equation. It occurs frequently in steady-state problem where there is no time-dependence. In two dimensions, we have

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0. \tag{2.3}$$

The diffusion equation is a parabolic equation. It is also known as the heat equation, and is often found when modelling the flow of heat into a region surrounding a heat source:

$$\frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}. \tag{2.4}$$

The above is the one dimension version of the equation.

The wave equation is a hyperbolic equation. It models the amplitude u of a wave moving through a medium with speed c:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}. \tag{2.5}$$

2.2 Well Posed Problems

A PDE problem is said to be **well-posed** if the solution exists, is unique and depends continuously on the problem data (i.e. the coefficients of the equation, the initial and boundary conditions and the problem space of the PDE).

The required conditions for a problem to be well-posed depend on the type of PDE involved:

- An elliptic problem is well-posed if the solution satisfies the boundary conditions at each point of the problem space.
- A parabolic problem is well-posed if the solution satisfies the boundary condition at each point of the boundary, and if the solution satisfies some initial condition.
- A hyperbolic problem is well-posed if the solution satisfies the boundary condition at each point on the boundary, and if the solution satisfies an initial condition for both the solution and the first derivative of the solution.

2.2.1 Boundary Conditions

Boundary conditions are extremely important when considering PDEs, as many PDE problems only make sense when accompanied by appropriate initial, final, and boundary conditions.

There are several different types of boundary conditions, and they are given different names depending on whether they are boundaries for space or time dimensions. Common boundary condition types for space coordinates are

Dirichlet Boundary Conditions The solution is specified at boundary.

Neumann Boundary Conditions The first derivative of the solution is specified at the boundary.

Periodic Boundary Conditions Similar to Dirichlet boundary conditions, but the value of the solution is the same on both sides of the boundary.

Robin Boundary Conditions The boundary condition is a linear combination of the solution and the first derivative of the solution.

2.3 Characteristics

Characteristics play an important role in PDEs, as they are used to classify the PDE, and a study of the characteristics associated with a problem can give insight into effective procedures for solving the PDE.

2.3.1 The Origin of Characteristics

Taking the general form of the second order, linear PDE, Eq. (2.2), we can re-write this equation as simply

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x\,\partial y} + C\frac{\partial^2 u}{\partial y^2} = f\left(u, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right). \tag{2.6}$$

Suppose, for simplicity, we have $f = 0$ everywhere. Now, our equation looks like

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x\,\partial y} + C\frac{\partial^2 u}{\partial y^2} = 0. \tag{2.7}$$

We shall now adopt the notation

$$u_x = \frac{\partial u}{\partial x}; \quad u_y = \frac{\partial u}{\partial y}; \quad u_{xx} = \frac{\partial^2 u}{\partial x^2}, \tag{2.8}$$

etc.

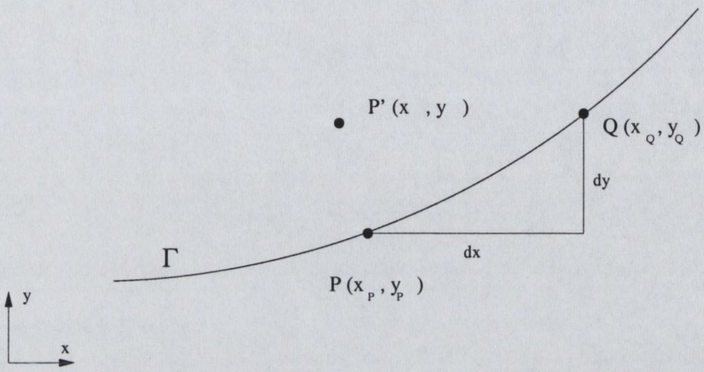

Figure 2-1: Derivation of Characteristics for a Second-Order PDE

Now suppose we have a curve in space, Γ, (see Fig 2-1) upon which we know u, u_x and u_y. Let P and Q be points on this curve Γ. Suppose we wish to know the value of u at another point P', which is not on Γ. This is given by the standard multi-dimensional Taylor expansion

$$\begin{aligned} u(P') &= u(P) - (x - x_P)u_x + (y - y_P)u_y \\ &\quad + \frac{1}{2!}\left[(x - x_P)^2 u_{xx} + 2(x - x_P)(y - y_P)u_{xy} + (y - y_P)^2 u_{yy}\right] + \dots \end{aligned} \tag{2.9}$$

Thus we need all the derivatives of $u(P)$. In particular, we need to find u_{xx}, u_{yy} and u_{xy}. Eq. (2.7) gives us one equation so two more are needed.

Let $p = u_x$, $q = u_y$, $r = u_{xx}$, $s = u_{xy}$ and $t = u_{yy}$. Thus

$$\begin{aligned} dp &= p(Q) - p(P) \\ &= \frac{\partial p}{\partial x} dx + \frac{\partial p}{\partial y} dy \\ &= r\,dx + s\,dy. \end{aligned} \tag{2.10}$$

Similarly

$$dq = s\,dx + t\,dy. \tag{2.11}$$

Thus we have a 3D linear system:

$$\begin{array}{rcrcrcl} Au_{xx} & + & Bu_{xy} & + & Cu_{yy} & = & 0, \\ u_{xx}\,dx & + & u_{xy}\,dy & + & 0 & = & dp, \\ 0 & + & u_{xy}\,dy & + & u_{yy}\,dx & = & dq, \end{array} \tag{2.12}$$

which can be rewritten as

$$\begin{pmatrix} A & B & C \\ dx & dy & 0 \\ 0 & dx & dy \end{pmatrix} \begin{pmatrix} u_{xx} \\ u_{xy} \\ u_{yy} \end{pmatrix} = \begin{pmatrix} 0 \\ dp \\ dq \end{pmatrix}. \tag{2.13}$$

The **characteristics** of a PDE are defined to be those curves in the problem space from which the second derivatives cannot be uniquely determined from this linear system. For this to occur the above matrix must be singular i.e. its determinant must be zero. So,

$$\begin{aligned} &A\,dy^2 - B\,dx\,dy + C\,dx^2 = 0, \\ \Rightarrow\; &A\left(\frac{dy}{dx}\right)^2 - B\frac{dy}{dx} + C = 0. \end{aligned} \tag{2.14}$$

Eq. (2.14) is said to be the **Characteristic Equation** of Eq. (2.7).

It is the characteristics of a PDE that define, in general, whether or not the PDE is elliptic, parabolic or hyperbolic. The general method of classifying a PDE is

Elliptic, if the PDE has no real characteristics.

Parabolic, if the PDE has less distinct characteristics than the order of the PDE.

Hyperbolic, if the PDE has the same number of distinct characteristics as the order of the PDE.

Characteristics can be thought of as curves through which information propagates throughout the problem space.

In elliptic problems, there are no real characteristics and so information does not propagate through the problem space. In fact, information travels instantaneously throughout the domain in all dimensions. Thus, elliptic problems tend to be steady-state problems since information cannot travel instantaneously through time or there would be no causality. A good example of where this occurs is in a steady-state potential problem, which is modeled by Laplace's Equation where a change at the source instantaneously affects the entire problem domain.

In parabolic problems, information tends to travel instantaneously through one or more dimensions, but travels with a finite time through the others. This occurs in the Heat Equation when a change in the source instantaneously affects the space coordinate, but there is causality as the information must propagate through the time dimension at finite speed (i.e. the heat requires time to spread out, but all directions are affected simultaneously).

In hyperbolic problems, all information must travel at finite speeds for each dimension in the problem. The Wave Equation is a fine example of this, all wave pulses travel along the characteristics in a finite time, in both forward and backward directions.

2.4 The Black-Scholes Partial Differential Equation

The Black-Scholes PDE,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{2.15}$$

is a second order, linear, parabolic PDE because comparing it to Eq. (2.2), we see that

$$A = \frac{1}{2}\sigma^2 S^2, \quad B = 0, \quad C = 0. \tag{2.16}$$

Thus, $B^2 - 4AC = 0 - 0 = 0$.

However, if $S = 0$, the problem reduces to that of an ordinary, first order hyperbolic equation with characteristic $S = 0$.

Thus our problem is parabolic for $S > 0$ but hyperbolic at $S = 0$. Financially, this is important: the line $S = 0$ is a barrier through which information cannot cross. Thus, we do not have to consider negative stock values in our calculations. This is just as well, as negative values of the underlying has no real financial interpretation for stock prices.

It is worth noting that forward contracts can have a negative value and so options on these assets require a different analysis. We do not concern ourselves with these derivatives however.

Thus, we need not worry about spurious results due to quirks in the mathematics of the models which do not have any real-world meaning.

As has been mentioned, the Black-Scholes equation is a type of diffusion equation, a mathematical system that has been known of since the beginning of the 19th century.

Versions of the diffusion equation have been used to successfully model the diffusion of one material within another (such as smoke particles in the air); the flow of heat from one region to another; the dispersion of populations with individuals moving both randomly and to avoid overcrowding; pursuit and evasion in predator–prey systems; and the dispersion of pollutants in a running stream.

In most of the above cases, the resulting equations are more complicated than the Black-Scholes equation.

The Black-Scholes equation can be accurately interpreted as a reaction-convection-diffusion equation. The diffusion part is a balance of a first-order t-derivative and a second-order S derivative:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}. \tag{2.17}$$

If those were the only terms in the Black-Scholes equation, it would still exhibit the smoothing out effect — any discontinuities in the payoff would be immediately smoothed out. The only difference between this and the basic diffusion equation is that the diffusion coefficient is a function of one of the variables, S. Thus we really have diffusion in a non-homogeneous medium.

The first-order S-derivative term

$$rS\,\frac{\partial V}{\partial S}, \tag{2.18}$$

can be thought of as a convection term. If this equation represented some physical system, such as the diffusion of smoke particles in the atmosphere, then the convective term would be due to a breeze, blowing the smoke in a preferred direction.

The final term

$$-rV, \tag{2.19}$$

is a reaction term. Balancing this term and the time derivative would give a model for decay of a radioactive body, with the half-life being related to r.

Put together, we get a reaction-convection-diffusion equation. An almost identical equation would arise when modelling the dispersion of a pollutant along a flowing river with absorption by the sand. In this, the dispersion is the diffusion, the flow is the convection and the absorption is the reaction.

2.4.1 Boundary and Initial/Terminal Conditions for Solving the Black-Scholes Equation

Since the Black-Scholes equation is a parabolic equation, to uniquely specify a solution, we need to specify an **initial** (or **terminal**) condition, and **boundary conditions**.

Our terminal condition is simply the payoff function for the option (which we shall assume to be a Put):

$$V(S_T, T) = \mathrm{Max}(K - S_T, 0). \tag{2.20}$$

The boundary conditions are those given in Chapter 1. If the stock S becomes zero, the payoff is simply the full strike price, discounted for the time:

$$V(0,t) = Ke^{-r(T-t)}. \tag{2.21}$$

As the stock gets larger and larger, it becomes less and less likely that the option will move into the money, and so the option becomes worthless:

$$V(\infty, t) = 0. \tag{2.22}$$

Thus, we have a terminal condition and two boundary conditions (all of which are Dirichlet conditions), as required.

The Black-Scholes equation is linear: add two solutions together and you will get a third. Linear diffusion equations have some useful properties, one being that any discontinuity in the boundary or initial/terminal conditions is immediately smoothed out in the solution.

This is useful because the payoff of the option is not differentiable at $S = K$ (i.e. the derivative of the function does not exist at $S = K$), but the linear nature of the equation means that we do not need to worry about this affecting the analytic solution, though this does have consequences for some numerical approaches (see Chapter 5).

Chapter 3

Stochastic Differential Equations

3.1 Introduction

One of the most fundamental assumptions made in modern theories of financial mathematics is the fact that the underlying assets do not behave deterministically (i.e. there is a certain amount of "randomness" intrinsic in the evolution of the asset through time).

For example, when studying the price of futures or options, it is assumed that the underlying asset price changes randomly. For interest-rate derivatives (such as bonds or swaps), it is the interest-rates which change randomly.

This assumption was made once it became apparent that techniques for predicting the future evolution of prices and interest rates were essentially worthless. Since it was effectively impossible to model these behaviours, the idea to assume random behaviour and work from there is logical.

Probably the most commonly known example of these models that assumes an inherent random element is **Brownian Motion**, the motion of a small pollen grain suspended in water first observed by Robert Brown in 1827.

In 1900, Louis Bachelier submitted his PhD thesis dissertation **Theorie de la Speculation** [1]. His thesis discussed the use of Brownian motion to evaluate stock options, and is historically the first paper to use advanced mathematics in the study of finance. Thus, Bachelier is considered a pioneer in the study of financial mathematics and stochastic processes, though his contribution is often overlooked (including by this author in the past).

His work predated Einstein's celebrated study of Brownian motion by about five years.

Einstein eventually explained the phenomenon of Brownian Motion by surmising that the pollen grain was in constant collision with water molecules, and the random nature of these collisions explained the erratic behaviour of the molecule. He did his work independently of Bachelier.

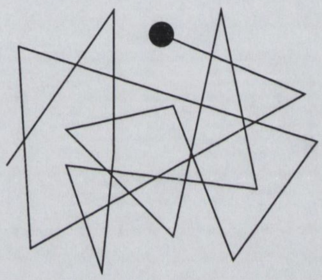

Figure 3-1: Example of a 2D Brownian Motion

In stochastic mathematics, Brownian Motion is considered to be an example of a **Random Walk**. The concept of a random walk is extremely important in financial applications of stochastic mathematics, as the underlying assets are assumed to undergo a random walk.

3.2 Stochastic Calculus

To explain the key ideas in stochastic calculus, I will use its application to option pricing theory. Stochastic calculus is much more general than this (many of the original theories for it were developed by scientists working on rocket propulsion, this is genuine "rocket-science"), but as this is the only application to which we shall apply it, there is no need to generalise further.

As mentioned previously, the aim of investing is, for a given level of risk, to maximise the **return**, that is, the percentage growth in the value of an asset over a period of time.

The percentage growth is much more important than the actual growth, since if two investments increase in value by \$10 over a period of time, but A involved an initial investment of \$100, whereas B involved \$1000, clearly A is the superior investment to B.

Table 3.1 shows a sample path of an asset starting from an initial value of \$40. Calculating the returns on the asset is relatively straightforward. Suppose the value of an asset on day i is given by S_i, then the return on the asset from day i to day $i+1$ is given by

$$R_{i+1} = \frac{S_{i+1} - S_i}{S_i}. \tag{3.1}$$

The above equation does not allow for dividends, and this study does not deal with their effect.

Figure 3-2 shows a plot of the daily returns of the above asset for a 100-day period. As can be easily seen, it appears very much like a random set of data, and it can be modelled as such.

From statistics, the sample mean of a distribution is given by

$$\bar{R} = \frac{1}{N} \sum_{i=1}^{N} R_i, \tag{3.2}$$

and the sample standard deviation is defined to be

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (R_i - \bar{R})^2}, \tag{3.3}$$

where N is the number of returns in the sample (which is one less than the number of asset prices).

The frequency distribution of the daily returns is easily done and is shown in Figure 3-3. The smooth curve is the normal curve for the mean and standard deviation of the sample data and closely follows the histogram.

Although the returns in this sample closely follow the normal distribution, this is not true in general. However, nearly all data sets are close enough to make the assumption that they are drawn from the normal distribution a reasonable one, and

Day	Asset	Return
00	\$40.0000	
01	\$39.2866	–0.0178
02	\$38.6155	–0.0171
03	\$39.0341	+0.0108
04	\$39.1175	+0.0021
05	\$38.9316	–0.0048
06	\$39.4114	+0.0123
07	\$38.6777	–0.0186
08	\$38.8083	+0.0034
09	\$38.9933	+0.0048
10	\$38.7756	–0.0056
11	\$38.7477	–0.0007
12	\$39.5847	+0.0216
13	\$40.5058	+0.0233
14	\$40.7901	+0.0070
15	\$40.5649	–0.0055
16	\$40.4339	–0.0032
17	\$39.7304	–0.0174
18	\$38.9643	–0.0193
19	\$38.7945	–0.0044
20	\$39.2986	+0.0130
21	\$38.8695	–0.0109
22	\$39.3801	+0.0131
23	\$39.6275	+0.0063
24	\$39.7340	+0.0027
25	\$39.6653	–0.0017

Table 3.1: Sample Path Data

we shall proceed as such.

Thus, the returns are treated as a random variable, drawn from a Normal distribution with non-zero, constant mean and non-zero constant standard deviation:

$$R_{i+1} = \frac{S_{i+1} - S_i}{S_i} = \text{mean} + \text{std dev} * \phi, \tag{3.4}$$

where ϕ is a random variable drawn from the standard, normal distribution.

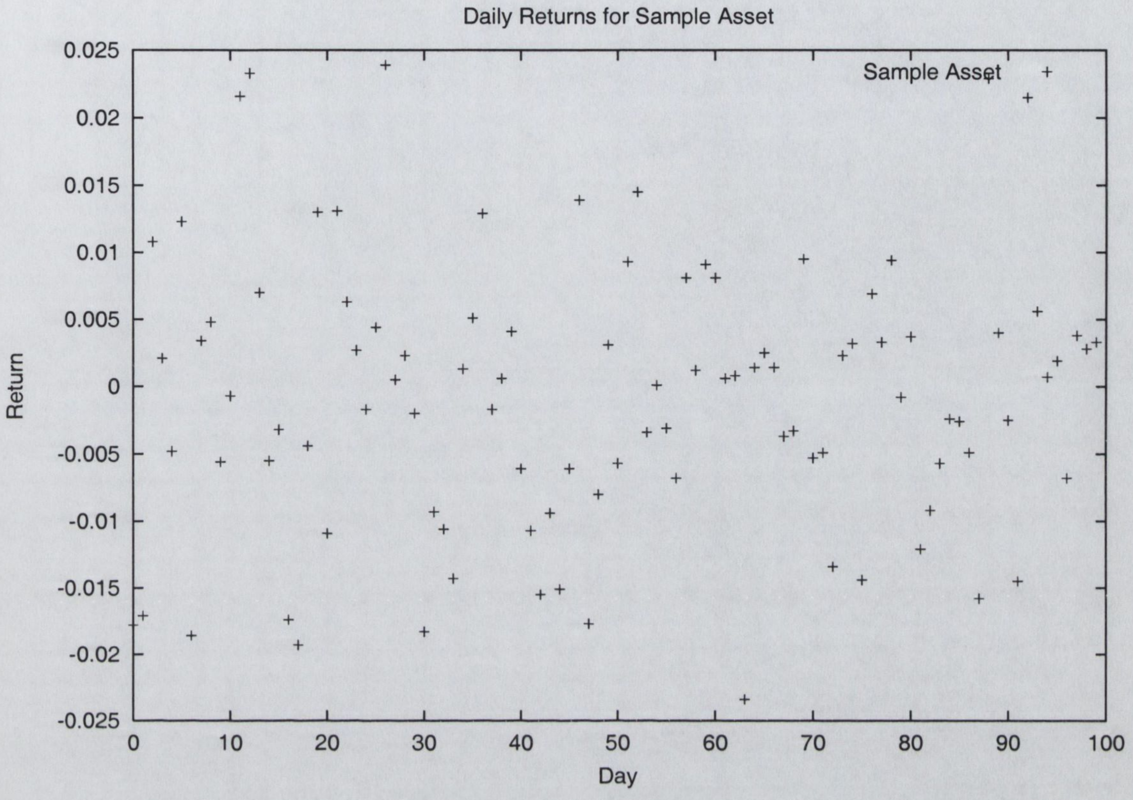

Figure 3-2: Daily Returns of a Sample Asset

3.3 Time Scales

Up to now, there has been little reference to the time-scales involved in these series of data. The timescale was stated to be daily, but all the above formulae work equally well using hours, minutes, months or years as the time intervals. How would this change of timescale affect the distribution?

Let the δt be the timestep between data measurements. We assume that the mean of the returns scales with the size of the timestep, i.e. the larger the sampling time, the larger the average. For simplicity, we shall assume this scaling is linear and constant.

$$\text{mean} = \mu \, \delta t, \tag{3.5}$$

for some constant μ.

Thus, by taking enough samples to eliminate the random element of the measure-

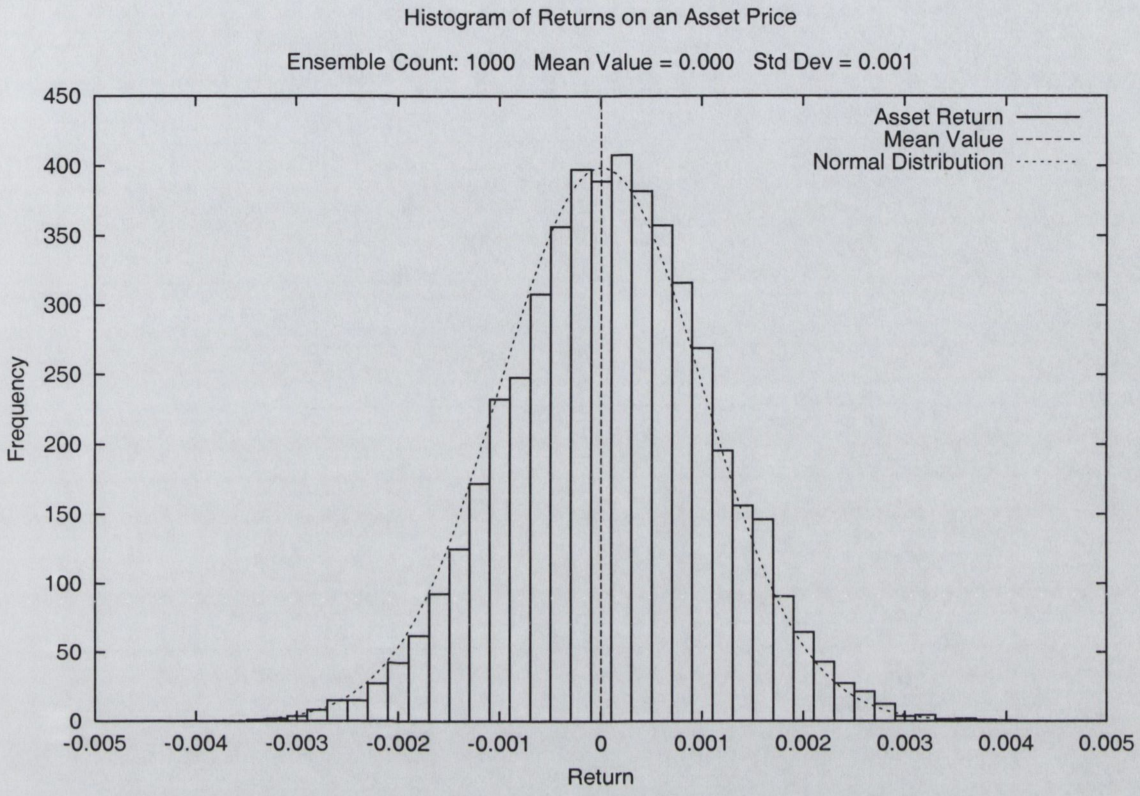

Figure 3-3: Histogram of the Daily Returns Frequency Distribution

ments, we have

$$E(R_{i+1}) = E\left(\frac{S_{i+1} - S_i}{S_i}\right) = \mu\,\delta t, \tag{3.6}$$

and rearranging,

$$S_{i+1} = S_i\,(1 + \mu\,\delta t). \tag{3.7}$$

Thus, an asset with initial value of S_0 at time $t = 0$ has a value at $t = \delta t$ of

$$S_1 = S_0\,(1 + \mu\,\delta t). \tag{3.8}$$

At $t = 2\,\delta t$, we have

$$S_2 = S_0\,(1 + \mu\,\delta t)^2. \tag{3.9}$$

and after M timesteps, $t = M\,\delta t$ and

$$S_M = S_0\,(1 + \mu\,\delta t). \tag{3.10}$$

With total time T fixed, then as the number of timesteps gets larger, δt get smaller, and Eq. (3.10) becomes

$$\begin{aligned} S_{t=T} &= \lim_{\delta t \to 0} S_0\,(1 + \mu\,\delta t), \\ &= S_0 \lim_{\delta t \to 0} (1 + \mu\,\delta t), \\ &= S_0\,e^{\mu T}. \end{aligned} \tag{3.11}$$

The above result is exact, and so the asset exhibits exponential growth in the absence of any randomness, just like cash in the bank.

Also, as the above result is finite in the limit as the timestep approaches zero, we can see that the model is meaningful. Any other power of δt would have given us either a trivial ($S_T = S_0$) or infinite answer.

With this in mind, we can now see that the obvious choice for scaling the change in standard deviation with time is $\sqrt{\delta t}$.

With $T/\delta t$ timesteps, Eq. (3.3) contains a sum of that many terms, and for the sum to remain finite, each must be of $O(\delta t)$. Since each is a square of the return, the standard deviation of the asset return over a timestep δt must be $O(\sqrt{\delta t})$.

Again, we assume we have some parameter in the scaling, σ, so that

$$\text{std dev} = \sigma\,\sqrt{\delta t}. \tag{3.12}$$

Note that σ is a measure of the randomness of the asset, since the larger this parameter is, the more uncertain the return. This value is often assumed to be constant and we make this assumption for the moment.

We can now substitute all this into our asset return model, Eq. (3.4)

$$R_i = \frac{S_{i+1} - S_i}{S_i} = \mu\,\delta t + \sigma\sqrt{\delta t}\,\phi. \tag{3.13}$$

Re-writing, we get

$$S_{i+1} - S_i = \mu S_i \, \delta t + \sigma S_i \sqrt{\delta t} \, \phi. \tag{3.14}$$

The LHS of Eq. (3.14) is the change in the asset price from timestep i to timestep $i+1$. The RHS is the 'model'. We can think of this equation as being a model for a random walk of the asset price. We know precisely what the current value of the asset is. The future values are unknown, but are distributed about the current value according to Eq. (3.14).

We now consider the two parameters μ and σ in more detail.

3.3.1 The Drift Rate

The parameter μ is called the **drift rate** or **expected return** or the **growth rate** of the asset. It is hard to measure statistically as it scales with the parameter δt, which is usually small, and is often estimated by

$$\mu = \frac{1}{M \, \delta t} \sum_{i=1}^{M} R_i. \tag{3.15}$$

The unit of time is usually one year, and so μ is often quoted as an **annualised** growth rate.

3.3.2 The Volatility

The parameter σ is called the **volatility** of the asset, and can be estimated by

$$\sigma = \sqrt{\frac{1}{(M-1) \, \delta t} \sum_{i=1}^{M} (R_i - \bar{R})^2}. \tag{3.16}$$

The volatility is usually quoted in annualised terms.

The volatility is an extremely important parameter in the theory of financial derivatives, as it estimates the 'randomness' of the asset price and dominates the short-term behaviour of the asset price. High volatility stocks can be very risky, with wildy fluctuating prices over short timescales (ready examples would be technology start-up companies), whereas low volatility stocks will have prices which will change very slowly (the so called 'blue-chip' companies, such as the old-school investment banks and industries).

However, when viewed over long timescales the randomness of the asset averages out, leaving just the effect of the drift.

3.4 Wiener Processes

Up to this point, everything has been modeled with **discrete time**. Financial models require the use of **continuous time**, i.e. where the time dimension is not discretised but continuous throughout the lifetime of the derivative.

In continuous time, the price path of the asset is a continuous function of time, unlike previously, where the price of the asset was only considered at the discrete timesteps. We now must consider the effect of continuous time on Eq. (3.13).

In the limit as $\delta t \to 0$, we use the notation d instead of δ to denote a change in a quantity. Thus, in continuous time, the change is the asset price is denoted dS and the change in time is denoted dt. However, it is not so easy to deal with the $\sqrt{\delta t}$. It will be shown that we can write this as

$$\phi\sqrt{\delta t} = dX. \tag{3.17}$$

We can treat dX as being a random variable drawn from a Normal distribution with zero mean and variance dt:

$$E[dX] = 0 \quad \text{and} \quad E[dX^2] = dt. \tag{3.18}$$

Using this continuous-time limit, and Wiener process notation, we can write our asset price model as

$$dS = \mu S\, dt + \sigma S\, dX. \tag{3.19}$$

3.4.1 Brownian Motion

We will now define **Brownian Motion**, $X(t)$ as being a random walk with the following properties:

Finiteness: The value of $X(t)$ should scale with the size of the timestep in a way that ensures that, in the limit of the zero timesteps, the value does not go to infinity or result in no change. For asset price movements, this means that the increment scales with the square root of the timestep.

Continuity: The paths are continuous and there are no discontinuities. Brownian motion is the continuous-time limit of the discrete time random walks.

Markov: The conditional distribution of $X(t)$ given information up until $\tau < t$ depends only on $X(\tau)$.

Martingale: Given information up until $\tau < t$, the conditional expectation of $X(t)$ is $X(\tau)$.

Quadratic Variation: If we divide up time 0 to t in a partition with $n+1$ partition points, with $t_i = \frac{it}{n}$ then

$$\sum_{j=1}^{n}(X(t_j) - X(t_{j-1}))^2 \to t. \tag{3.20}$$

Normality Over finite time increments t_i to t_i, $X(t_i) - X(t_{i-1})$ is Normally distributed with mean zero and variance $t_i - t_{i-1}$.

3.5 Stochastic Integration

A **stochastic integral** is defined as

$$W(t) = \int_0^t f(\tau)\, dX(\tau) = \lim_{n\to\infty} \sum_{n=0}^{\infty} f(t_{j-1})(X(t_j) - X(t_{j-1})), \tag{3.21}$$

with

$$t_j = \frac{jt}{n}.$$

In the above calculation, it is important that the function $f(\tau)$ being integrated is evaluated at the $t = t_{j-1}$ timestep. The integration is said to be **non-anticipatory** and this ensures there is no future information used in current calculations.

Since the use of such integrals can be extremely cumbersome in calculations, it is much more common to write Eq. (3.21) as

$$dW = f(t)\, dX. \tag{3.23}$$

Note that this notation comes from 'differentiating' Eq. (3.21), and has the benefit that it looks much more like an ordinary differential equation. We can think of dX as being an increment of X, a Normal random variable with zero mean and $\sqrt{dt}$ standard deviation.

We do not then divide the equation by dt, as this would introduce the difficulty of giving meaning to dX/dt.

Extending this idea, we can see that

$$dW = g(t)\, dt + f(t)\, dX, \tag{3.24}$$

is a simple shorthand for

$$W(t) = \int_0^t g(\tau)\, d\tau + \int_0^t f(\tau)\, dX(\tau). \tag{3.25}$$

These are examples of **stochastic differential equations**.

3.6 Functions of Stochastic Variables and Ito's Lemma

Now we consider functions of a stochastic variable, $F(X(t))$. Suppose we define $F = X^2$, and wish to find dF. The standard rules of differentiation do not hold in the stochastic environment and so we need to develop new rules of calculus.

To develop these rules, we introduce a very small timescale, denoted by h, such that

$$h = \frac{\delta t}{n}. \tag{3.26}$$

This timescale is so small that we can approximate $F(X(t+h))$ by a Taylor series:

$$F(X(t+h)) - F(X(t)) = (X(t+h) - X(t))\,\frac{dF}{dX}(X(t)) + \frac{1}{2}\,(X(t+h) - X(t))^2\,\frac{d^2F}{dX^2}(X(t)) + \ldots \tag{3.27}$$

We can then use this to show that [29]

$$F(X(t)) = F(X(0)) + \int_0^t \frac{dF}{dX}(X(\tau))dX(\tau) + \frac{1}{2}\int_0^t \frac{d^2F}{dX^2}(X(\tau))d\tau. \tag{3.28}$$

This is the integral version of **Ito's Lemma**, which is usually written in the form

$$dF = \frac{dF}{dX}\,dX + \frac{1}{2}\frac{d^2F}{dX^2}\,dt. \tag{3.29}$$

So, to return to our example of $F = X^2$ it is now obvious that since

$$\frac{dF}{dX} = 2X \text{ and } \frac{d^2F}{dX^2} = 2,$$

then we have $dF = 2X\,dX + dt$.

Generalising, suppose we have the stochastic differential equation

$$dS = a(S)\,dt + b(S)\,dX, \tag{3.31}$$

for some functions $a(S)$ and $b(S)$. A function of S, $V(S)$ would satisfy

$$dV = \frac{dV}{dS}\,dS + \frac{1}{2}b^2\,\frac{d^2V}{dS^2}\,dt. \tag{3.32}$$

This will be used when considering financial applications of stochastic differential equations.

Consider the application of Ito's Lemma to functions of one stochastic variable, S and one deterministic variable t, $V(S,t)$. We have

$$dS = a(S,t)\,dt + b(S,t)\,dX, \tag{3.33}$$

and it follows that the increment of the function, dV is

$$dV = \frac{\partial V}{\partial t}\,dt + \frac{\partial V}{\partial S}\,dS + \frac{1}{2}b^2\,\frac{\partial^2 V}{\partial S^2}dt. \tag{3.34}$$

3.7 Applications to Financial Derivatives

Consider a simple random walk with a drift

$$dS = \mu\,dt + \sigma\,dX. \tag{3.35}$$

Integrating it exactly, we get

$$S(t) = S(0) + \mu t + \sigma(X(t) - X(0)). \tag{3.36}$$

The above example is not particularly suited to option pricing as it allows the value of S to become negative (which is unrealistic for the the price of stock). A possibility is to scale the the drift and volatility with S:

$$dS = \mu S\,dt + \sigma S\,dX. \tag{3.37}$$

In this case, as the value of $S \to 0$, the size of dS also decreases, so that $S = 0$ can never actually be reached.

To show this, we examine the function $F(S) = \ln S$.

$$\begin{aligned} dF &= \frac{dF}{dS}\,dS + \frac{1}{2}\sigma^2 S^2\,\frac{d^2F}{dS^2}, \\ &= \frac{1}{S}(\mu S\,dt + \sigma S\,dX) - \frac{1}{2}\sigma^2\,dt, \\ &= (\mu - \frac{1}{2}\sigma^2)\,dt + \sigma\,dX. \end{aligned} \tag{3.38}$$

Thus, $-\infty \leq \ln S \leq \infty$.

Chapter 4

Exact Results

4.1 Exact Solution for Vanilla Calls and Puts

We now turn our attention towards deriving an analytical solution to the Black-Scholes equation.

The Black-Scholes equation is

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

The value of the option is the present value of the average risk-neutral payoff, and this suggests a possible simplification by converting to a future value price, U, given by

$$V(S,t) = e^{-r(T-t)}\,U(S,t). \tag{4.2}$$

Making this change of variable, we get

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + rS\,\frac{\partial U}{\partial S} = 0. \tag{4.3}$$

This is a **backwards** equation, because we have a terminal condition as opposed to an initial condition.

This is trivially changed by the simple change of variable

$$\tau = T - t, \tag{4.4}$$

which leaves us with

$$\frac{\partial U}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S}. \tag{4.5}$$

When first modeling equity prices, the stochastic differential equation model was built up around consideration for the **return** of the asset. As we assume the return is a lognormal walk, this would suggest another change of variable, namely,

$$\xi = \ln S. \tag{4.6}$$

Using this new variable, we find that

$$\frac{\partial}{\partial S} = e^{-\xi}\frac{\partial}{\partial \xi} \qquad \text{and} \qquad \frac{\partial^2}{\partial S^2} = e^{-2\xi}\frac{\partial^2}{\partial \xi^2} - e^{-2\xi}\frac{\partial}{\partial \xi}. \tag{4.7}$$

With this change of variable, our PDE becomes

$$\frac{\partial U}{\partial \tau} = \frac{1}{2}\sigma^2 \frac{\partial^2 U}{\partial \xi^2} + (r - \tfrac{1}{2}\sigma^2)\frac{\partial U}{\partial \xi}. \tag{4.8}$$

We now have a second-order partial differential equation with constant coefficients, which is relatively straightforward to solve.

Eq. (4.8) is a PDE in two dimensions, ξ and τ, a 'space' and 'time' dimension respectively.

We can again change the variables of the problem to simplify it down again, to

$$x = \xi + (r - \tfrac{1}{2}\sigma^2)\,\tau, \tag{4.9}$$

and

$$W(x, \tau) = U. \tag{4.10}$$

with this change of variables, we have reduced the problem down even further, to simply,

$$\frac{\partial W}{\partial \tau} = \frac{1}{2}\sigma^2 \frac{\partial^2 W}{\partial x^2}. \tag{4.11}$$

Linking the current dimensions back to our original Black-Scholes equation, we see that

$$\begin{aligned} V(S,t) &= e^{-r(T-t)}\,U(S,t), \\ &= e^{-r\tau}\,U(S,T-\tau), \\ &= e^{-r\tau}\,U(e^{\xi},T-\tau), \\ &= e^{-r\tau}\,U(e^{x-(r-\frac{1}{2}\sigma^2)\tau},T-\tau), \\ &= e^{-r\tau}\,W(x,\tau), \end{aligned} \tag{4.12}$$

and

$$\begin{aligned} x &= \xi + (r - \tfrac{1}{2}\sigma^2)\,\tau \\ &= \log S + (r - \tfrac{1}{2}\sigma^2)(T-t). \end{aligned}$$

To solve this equation, we first look for a special solution of Eq. (4.11) of the form

$$W_f(x,\tau) = \tau^{\alpha} f\left(\frac{(x-x')}{\tau^{\beta}}\right), \tag{4.13}$$

where x' is an arbitrary constant. The function f is unknown and it depends on a single variable, $(x-x')/\tau^{\beta}$, and so as well as determining f we need to fix the values of α and β. This has the benefit that Eq. (4.11) should reduce to an ordinary differential equation since f is a function of just one variable.

We now substitute Eq. (4.13) into Eq. (4.11),

$$\tau^{\alpha-1}\left(\alpha f - \beta\eta\frac{df}{d\eta}\right) = \frac{1}{2}\sigma^2\tau^{\alpha-2\beta}\,\frac{d^2 f}{d\eta^2}, \tag{4.14}$$

where

$$\eta = \frac{x-x'}{\tau^{\beta}}. \tag{4.15}$$

For equality to hold, the powers of η in Eq. (4.14) must be identical, and this gives us

$$\alpha - 1 = \alpha - 2\beta, \tag{4.16}$$

$$\Longrightarrow \beta = \frac{1}{2}. \tag{4.17}$$

The next step is to ensure that the solution has the property that the integral over all ξ is independant of τ (the reason for this will be explained later). Thus, we need to have

$$\int_{-\infty}^{\infty} \tau^{\alpha} f\left(\frac{x-x'}{\tau^{\beta}}\right) dx = C, \tag{4.18}$$

where C is a constant. Rewriting this as an integral of $d\eta$ this becomes

$$\int_{\infty}^{\infty} \tau^{\alpha+\beta} f(\eta) d\eta = C, \tag{4.19}$$

and so this requires that $\alpha = -\beta = \frac{1}{2}$.

The function f now satisfies the ordinary differential equation

$$-f - \eta\frac{df}{d\eta} = \sigma^2\frac{d^2 f}{d\eta^2}, \tag{4.20}$$

which can be rewritten in the form

$$\sigma^2\frac{d^2 f}{d\eta^2} + \frac{d(\eta f)}{d\eta} = 0, \tag{4.21}$$

This can be integrated once to get

$$\sigma^2\frac{df}{d\eta} + \eta f = a, \tag{4.22}$$

where a is an arbitrary constant. We now choose $a = 0$ and integrate again to get

$$f(\eta) = b \exp\frac{\eta^2}{2\sigma^2}. \tag{4.23}$$

Again, b is an arbitrary constant, but we will fix it so that the integral of f over

$[-\infty, \infty]$ is unity:

$$f(\eta) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\eta^2}{2\sigma^2}}. \tag{4.24}$$

So, substituting Eq. (4.24) into Eq. (4.13) we get

$$W(x,\tau) = \frac{1}{\sqrt{2\pi\tau\sigma}} e^{-\frac{(x-x')^2}{2\sigma^2\tau}}. \tag{4.25}$$

Fig. 4-1 shows the shape of $W(x,\tau)$ for various values of τ. As $\tau \to 0$, we see that the function gets taller and thinner around the point $x = x'$. Mathematically speaking, for decreasing values of τ the function grows without bound at this point and approaches zero everywhere else.

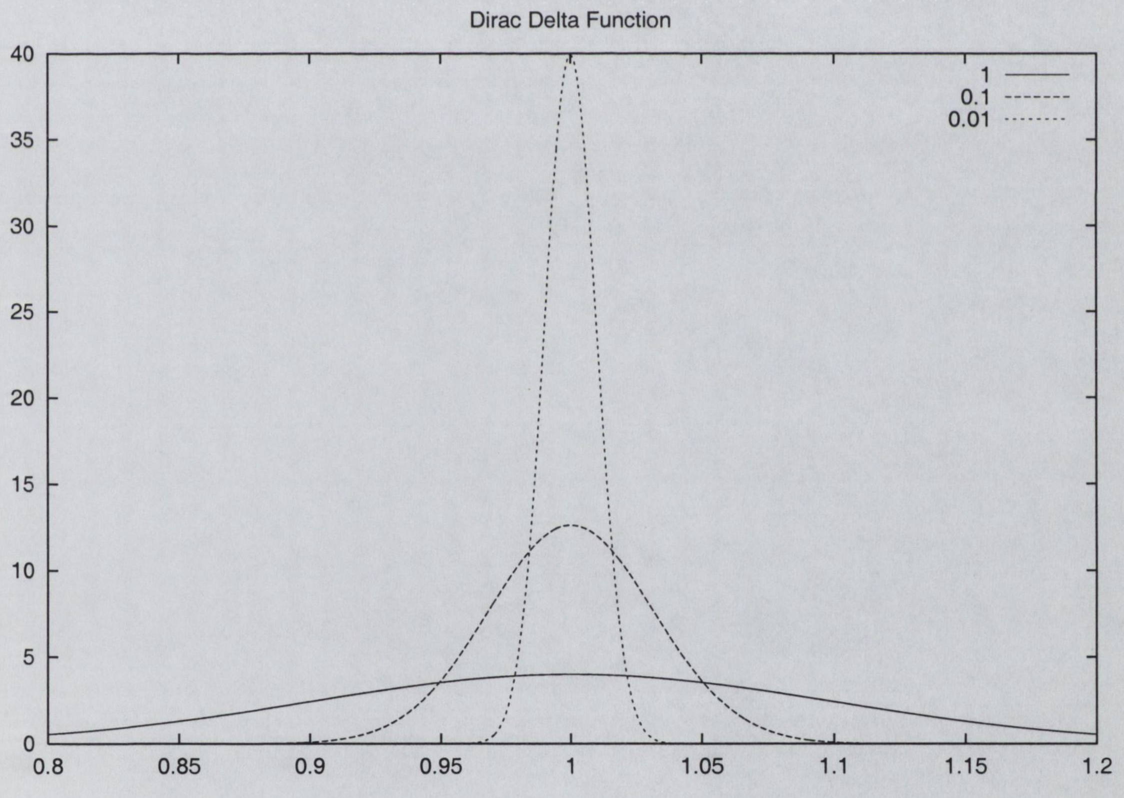

Figure 4-1: Plots of the Function $W(x,\tau)$ for different values of τ

However, we ensured earlier that the value of the integral was independent of the value of τ and so the area under this curve is constant for all values of τ. The result is that in the limit $\tau \to 0$, the function $W(x,0)$ becomes the **Dirac delta function**, which is a mathematical function with the important property that

$$\int \delta(x' - x)\, g(x')\, dx' = g(x). \tag{4.26}$$

Thus, integrating with the Dirac delta function essentially 'selects' the value of the function at the point where the Dirac function is singular. In the limit $\tau \to 0$, our function $W(x, \tau)$ is the dirac function so that

$$\lim_{\tau \to 0} \frac{1}{\sqrt{2\pi\tau}\sigma} \int_{-\infty}^{\infty} \exp\left(-\frac{(x' - x)^2}{2\sigma^2\tau}\right) g(x')\, dx' = g(x). \tag{4.27}$$

We now turn our attention to the payoff of the option, and denote it

$$H(S).$$

where H is the payoff function of the option.

This is the value of the option at expiry, where $t = T$. It is the final condition for $V(S, T)$, satifying the Black-Scholes equation:

$$V(S, T) = H(S). \tag{4.28}$$

In the other set of variables, this condition becomes

$$W(x, 0) = H(e^x). \tag{4.29}$$

It is easy to see that

$$W(x, \tau) = \int_{\infty}^{-\infty} W_f(x, \tau; x')\, H(e^{x'})\, dx'. \tag{4.30}$$

and this can be easily verified by substituting it into Eq. (4.11) and into Eq. (4.29) (from the properties of the function W_f).

Re-writing the function in terms of the new variables S and t, we have

$$V(S,t) = \frac{e^{-r(T-t)}}{\sigma\sqrt{2\pi(T-t)}} \int_0^\infty \exp\left(-\frac{(\ln(S/S') + (r-\frac{1}{2}\sigma^2)(T-t))^2}{2\sigma^2(T-t)}\right) \times H(S')\,\frac{dS'}{S'}. \quad (4.31)$$

4.1.1 Formula for Call Option

A call option has a payoff of

$$H(S) = \max(S-K, 0). \quad (4.32)$$

Substituting this into Eq. (4.31), we get

$$V(S,t) = \frac{e^{-r(T-t)}}{\sigma\sqrt{2\pi(T-t)}} \int_K^\infty \exp\left(-\frac{(\ln(S/S') + (r-\frac{1}{2}\sigma^2)(T-t))^2}{2\sigma^2(T-t)}\right) (S'-K)\,\frac{dS'}{S'}. \quad (4.33)$$

Returning to the variable $x = \ln S'$,

$$\begin{aligned}
&\frac{e^{-r(T-t)}}{\sigma\sqrt{2\pi(T-t)}} \int_{\ln K}^\infty \exp\left(-\frac{(-x' + \ln S' + (r-\frac{1}{2}\sigma^2)(T-t))^2}{2\sigma^2(T-t)}\right) (e^{x'} - K)\,dx' \\
&= \frac{e^{-r(T-t)}}{\sigma\sqrt{2\pi(T-t)}} \int_K^\infty \exp\left(-\frac{(-x' + \ln S' + (r-\frac{1}{2}\sigma^2)(T-t))^2}{2\sigma^2(T-t)}\right) e^{x'}\,dx' \\
&- E\,\frac{e^{-r(T-t)}}{\sigma\sqrt{2\pi(T-t)}} \int_K^\infty \exp\left(-\frac{(-x' + \ln S' + (r-\frac{1}{2}\sigma^2)(T-t))^2}{2\sigma^2(T-t)}\right) dx'. \quad (4.34)
\end{aligned}$$

Both integrals on the RHS can be written in the form

$$\int_d^\infty e^{-\frac{1}{2}x'^2}\,dx', \quad (4.35)$$

for some d, and so the option price can be written as two separate terms involving the cumulative distribution function for a Normal distribution:

$$C(S,t) = S\,N(d_1) - Ke^{-r(T-t)}\,N(d_2), \quad (4.36)$$

where

$$\begin{aligned} d_1 &= \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \\ d_2 &= \frac{\ln(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}. \end{aligned} \tag{4.37}$$

4.1.2 Formula for Put Option

The put option has a payoff

$$H(S) = \max(K - S, 0). \tag{4.38}$$

The value of the put option can be calculated in a similar manner to that of a call option, yielding the formula

$$P(S,t) = -SN(-d_1) + Ke^{-r(T-t)}N(-d_2), \tag{4.39}$$

with the same values for d_1 and d_2 as in Eq. (4.36).

Chapter 5

The Finite Difference Approach

This chapter defines the various finite difference schemes for partial differential equations.

Results of formal analysis for convergence and stability will be listed for each scheme.

5.1 The Finite Difference Method

5.1.1 Ordinary Derivatives

The finite difference method is a computational method for approximating continuous derivatives in a discrete way. Suppose we have a function u, of the variable x, defined on a discrete mesh with N mesh points. Suppose that the value of x is in $[0, X]$.

Thus, if the distance between mesh-points is constant and denoted dx, we have

$$dx = \frac{X}{N}. \tag{5.1}$$

We denote the values of u on each mesh-point as u_i, where $i = 0, 1, ..., N-1$ and u_i $= u(x = idx)$.

We now require some method of approximating derivatives on this mesh. Fortunately, the definition of the derivative immediately suggests an appropriate approximation:

$$\frac{du}{dx} = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}. \tag{5.2}$$

Since we are working on a discrete mesh, we do not take the limit as $h \to 0$ and instead fix it as the grid size Δx

$$\frac{du}{dx} = \frac{u_{i+1} - u_i}{\Delta x}. \tag{5.3}$$

Eq. (5.3) is called the **Forward difference approximation**.

There are many other ways of approximating the derivative (using different combinations of mesh-points). Two other very common approximations are

$$\frac{du}{dx} = \frac{u_i - u_{i-1}}{\Delta x}, \tag{5.4}$$

$$= \frac{u_{i+1} - u_{i-1}}{2\,\Delta x}, \tag{5.5}$$

which are known as the **Backward difference** and **Central difference** approximations respectively.

To approximate higher order derivatives, we use the fact that the n-th derivative is just a derivative of the $(n-1)$-th derivative, and apply a finite difference approximation. The following are finite difference approximations to the second- and third-order derivatives, using a central differencing approximation.

$$\frac{d^2u}{dx^2} = \frac{u^{n+1} - 2u^n + u^{n-1}}{(\Delta x)^2}, \tag{5.6}$$

$$\frac{d^3u}{dx^3} = \frac{u^{n+2} - 3u^{n+1} + 3u^{n-1} - u^{n-2}}{(\Delta x)^3}. \tag{5.7}$$

5.1.2 Partial Derivatives

We now turn our attention to functions of more than one variable and the partial derivatives of these functions. The discussion is limited to functions of two variables, such as $u(x, y)$, for the purposes of clarity and brevity, as the extension to higher dimensions is very straightforward.

So, suppose we have the function $u(x, t)$ which depends on two independant variables x and t. We now need a two-dimensional mesh to discretise x and t. Suppose the grid is of size $(N \text{ x } J)$, containing J points in the x-direction and N points in the t-direction.

We now need two indices to specify a value of u on this mesh, and for simplicity, we shall use u_j^n, where

$$u_j^n = u(x = j\,dx, t = n\,\Delta t), \tag{5.8}$$

where $n = 0, 1, ..., N-1$ is the t-index, and $j = 0, 1, ..., J-1$ is the x-index.

The partial derivatives are then approximated as follows:

$$\frac{\partial u}{\partial x} = \frac{u_{j+1}^n - u_j^n}{\Delta x}, \tag{5.9}$$

$$\frac{\partial u}{\partial t} = \frac{u_j^{n+1} - u_j^n}{\Delta t}. \tag{5.10}$$

Higher-order derivatives are approximated in a similar way:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}, \tag{5.11}$$

$$\frac{\partial^2 u}{\partial t^2} = \frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{(\Delta t)^2}. \tag{5.12}$$

Note that for time derivatives, using central differences results in schemes with mesh points along three different time-slices (u_j^{n-1}, u_j^n and u_j^{n+1}). This proves quite cumbersome to solve numerically, and so time derivatives are usually approximated with forward or backward differencing only.

5.2 Solving Partial Differential Equations

We now turn our attention to using these finite differences to solve partial differential equations (PDEs). The study of PDEs is a vast topic, and we shall concentrate on one particular type of PDE, second-order parabolic equations, which occur quite frequently in the pricing of financial instruments.

5.2.1 Consistency, Stability and Convergence

Before we can use any kind of finite difference scheme, it is very important to ensure that the scheme will give accurate results. Checking the accuracy of a scheme is very important in Numerical Analysis, and to do this we require a number of concepts.

Consistency: A finite difference scheme is said to be **consistent** if the discretised equation approaches the PDE as the number of mesh-points increases.

Stability: A finite difference scheme is said to be **stable** if the computed numerical solution approaches the actual solution of the discretised difference equation as the number of mesh-points increases.

Convergence: A finite difference scheme is said to be **convergent** if the computed numerical solution approaches the actual solution of the continuous PDE as the number of mesh-points increases.

The three concepts of Consistency, Stability and Convergence are linked by a principle called the Fundamental Equivalence Theorem of Lax:

> **For a well-posed Initial Value Problem, with a consistent discretisation scheme, stability is a necessary and sufficient condition for convergence.**

Thus, if we know our difference scheme is consistent with the PDE (and this is normally fairly straightforward to verify), then we just have to prove that the scheme is stable to show that the solution converges to the correct answer.

Stability is the main reason why explicit schemes are not used very often. They usually have highly restrictive stability conditions that impose strict bounds on the mesh-sizes. As a result, explicit schemes can be very expensive to calculate computationally due to sheer size, rather than complexity.

In some cases, explicit schemes are unconditionally unstable. When this happens, the scheme is completely useless for any kind of calculation.

Implicit schemes tend to have much less restrictive stability conditions (indeed many, such as the simple implicit scheme for the diffusion equation, are unconditionally stable) and so are the most common type of scheme used.

5.2.2 The Diffusion Equation

The diffusion equation is the most simple type of second-order parabolic equation:

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} \tag{5.13}$$

where κ is called the **diffusion constant**.

The diffusion equation commonly occurs in physics to describe the evolution of fluids as they move freely from regions of high concentration to regions of low concentration.

Also often referred to as the **heat equation** (since it was originally used to model how heat diffused through a medium), this equation is one of the most successful and widely used models in physics and applied maths.

One finite difference approximation of this PDE could be

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \kappa \left(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} \right). \tag{5.14}$$

This scheme is quite straightforward to solve. Assuming we have the initial values for u (along time-step $n = 0$), we can use these values to calculate the values for u along time-step $n = 1$. In turn, we can use these values to calculate u along each successive time-step, using the values calculated along the previous time-step. This type of difference scheme is referred to as an **explicit difference scheme**, since we can calculate the unknowns explicitly from the difference scheme and the known values in the equation.

In an **explicit** scheme, the difference equations contain single unknown mesh-point, allowing its value to be calculated directly from the known values. Thus, solutions arising from explicit schemes are very easy to find.

The above scheme, Eq. (5.14) is often referred to as the **FTCS** scheme (Forward-

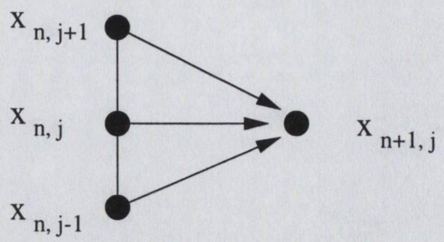

Figure 5-1: An Explicit Scheme

in-Time, Centred-in-Space), and is very simple to derive. Unfortunately, it is completely useless for computational purposes since it is easily shown to be unconditionally unstable for the simple diffusion equation [32], rendering any solutions it produces useless.

Now, instead of using a forward difference for the time-derivative, take a backward difference instead:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \kappa \left(\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2} \right). \tag{5.15}$$

A stability analysis of this scheme shows that this is unconditionally stable [32]. The disadvantage to this scheme is that of the four mesh-points used in the scheme, only one is known (u_j^n), whereas three are unknown (u_{j+1}^{n+1}, u_j^{n+1}, u_{j-1}^{n+1}), and so we are required to solve a system of these equations to find the values u^{n+1}. This type of difference scheme is referred to as an **implicit difference scheme**

In an **implicit** scheme, the difference equations contain multiple unknown mesh-points. These are still solvable but require linear methods. The problem is reduced to that of a standard linear system and solved using a linear solver.

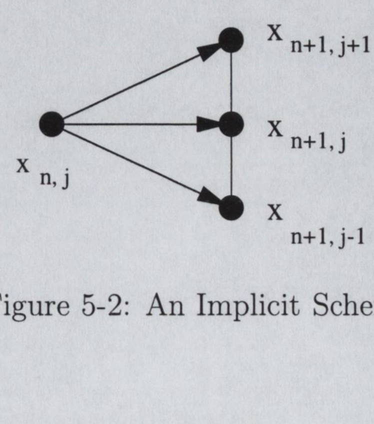

Figure 5-2: An Implicit Scheme

Despite being much easier to calculate, explicit schemes are not used very often, for reasons which will be explained in the next section.

Thus, we usually require matrix methods and linear solvers to obtain solutions to these numerical schemes.

5.3 Singly-Perturbed Problems

Problems can arise when the coefficient of the diffusion term in Eq. (5.17), ϵ, becomes small. In this case, the second-order equation begins to behave like a first-order equation (as the effect of the diffusion term is being reduced by the small coefficient).

This is a well-known problem, as it is a common occurence in fluid flow problems when the amount of diffusion is very small. This problem is exacerbated by the non-differentiability of the payoff function at $S = K$. Such problems are known as **singly-perturbed problems**.

The end result is oscillations in the numerical solution. While often small in the solution itself, these oscillations are greatly magnified should the derivatives of the solution be required, and this is indeed the case with the Fully Implicit and Crank-Nicolson schemes (see Figs 5-11 – 5-14). This problem can be eliminated by using a small enough timestep, but sometimes this is not practical.

These spurious oscillations are due to the fact that, when central differences are used for the derivative $\frac{\partial V}{\partial S}$, a scheme is not guaranteed to be a **positive coefficient discretisation**. Such a discretisation will prevent spurious oscillations, as the numerical scheme will obey a Local Extremum Diminishing (LED) property that guarantees local maxima are not increased and local minima are not decreased.

The Implicit scheme can be made to have all positive coefficients by using upstream differencing at nodes where the coefficients resulting from central differences are negative. In this way, spurious oscillations will not occur in the solution.

For a uniform mesh, central differencing can be used so long as

$$i \geq \frac{r}{\sigma^2} \qquad i = 1, 2, 3, ..., \frac{S_{\max}}{\Delta S} \tag{5.16}$$

where i is the index of the mesh node.

Thus, for "typical" financial parameters, on a few nodes near $S = 0$ need upstream weighting, so that fact that this discretization is $O(h)$ will not have a large impact.

However, in problems with low volatility and high interest rates, upstream weighting is required for all nodes to guarantee a positive coefficient discretisation.

Zvan, Forsyth and Vetzal also discuss this problem in detail in the article that first used the Van Leer flux-limiter in the financial area [43].

5.4 The Discretisation Schemes

Writing the Black-Scholes equation in a more general form (used quite often for this class of equations in Fluid Dynamics), getting

$$c\,\frac{\partial u}{\partial t} + \epsilon\,\frac{\partial^2 u}{\partial x^2} + a\,\frac{\partial u}{\partial x} - b\,u = f(x), \tag{5.17}$$

where

$$\begin{aligned} x &\rightarrow S, \\ c &= 1, \\ \epsilon &= \frac{1}{2}\sigma^2 S^2, \\ a &= rS, \\ b &= r, \\ f(x) &= 0. \end{aligned} \tag{5.18}$$

In this thesis, five different numerical discretisation schemes were used to solve the Black-Scholes equation:

1 – The Fully Implicit scheme

2 – The Fitted Duffy scheme

3 – The Crank-Nicolson scheme

4 – The Fitted Crank-Nicolson scheme

5 – The Van-Leer flux limiter scheme

5.4.1 The Fully Implicit Scheme

The Fully Implicit discretisation scheme is one of the simplest schemes in Numerical Analysis. It uses the backward difference approximation in the t-direction and the central difference approximation in the S-direction.

Thus,

$$\begin{aligned} \frac{\partial V}{\partial t} &= \frac{V_j^{n+1} - V_j^n}{\Delta t}, \\ \frac{\partial V}{\partial S} &= \frac{V_{j+1}^{n+1} - V_{j-1}^{n+1}}{2\,\Delta x}, \end{aligned} \tag{5.19}$$

and giving us

$$\begin{aligned} c_j^{n+1}\left(\frac{u_j^{n+1} - u_j^n}{\Delta t}\right) + \epsilon_j^{n+1}\left(\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2}\right) \\ + a_j^{n+1}\left(\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\,\Delta x}\right) - b_j^{n+1}u_j^{n+1} = 0 \end{aligned} \tag{5.20}$$

Putting all this together and grouping co-efficients, we get

$$\begin{aligned} \left(\frac{\epsilon_j^{n+1}}{(\Delta x)^2} - \frac{a_j^{n+1}}{2\,\Delta x}\right)u_{j-1}^{n+1} + \left(-b_j^{n+1} - 2\frac{2\epsilon_j^{n+1}}{(\Delta x)^2} - \frac{c_j^{n+1}}{\Delta t}\right)u_j^{n+1} \\ + \left(\frac{\epsilon_j^{n+1}}{(\Delta x)^2} + \frac{a_j^{n+1}}{2\,\Delta x}\right)u_{j+1}^{n+1} = -\frac{c_j^n}{\Delta x}u_j^n. \end{aligned} \tag{5.21}$$

The fully implicit scheme does have the disadvantage that it is just a first-order scheme, and so is not as accurate as some of the others.

5.4.2 The Crank-Nicolson Scheme

The explicit scheme is unconditionally unstable but is an $O(h^2)$ scheme nonetheless. The idea behind the Crank-Nicolson scheme was to combine both the implicit and an explicit schemes and try to gain the benefits of both.

The Crank-Nicolson scheme is a well-known numerical scheme. It is $O(h^2)$ and stable, and is found by taking an average of both the implicit and explicit schemes:

$$c_j^{n+1}\left(\frac{u_j^{n+1}-u_j^n}{\Delta t}\right)+\epsilon_j^{n+1}\left[\frac{1}{2}\left(\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{(\Delta h)^2}\right)+\frac{1}{2}\left(\frac{u_{j+1}^{n}-2u_j^{n}+u_{j-1}^{n}}{(\Delta h)^2}\right)\right]$$
$$+a_j^{n+1}\left[\frac{1}{2}\left(\frac{u_{j+1}^{n+1}-u_{j-1}^{n+1}}{2\,\Delta x}\right)+\frac{1}{2}\left(\frac{u_{j+1}^{n}-u_{j-1}^{n}}{2\,\Delta x}\right)\right]-b_j^{n+1}\left(\frac{1}{2}u_j^{n+1}+\frac{1}{2}u_j^n\right)=0 \tag{5.22}$$

Rearranging,

$$\left(\frac{\epsilon_j^n\,\Delta t}{2\,(\Delta x)^2}-\frac{a_j^k\,\Delta t}{4\,\Delta x}\right)u_{j-1}^{n+1}+\left(-c_j^n-\frac{\epsilon_j^n\,\Delta t}{(\Delta x)^2}-\frac{b\,\Delta t}{2}\right)u_j^{n+1}+\left(\frac{\epsilon_j^n\,\Delta t}{2\,(\Delta x)^2}+\frac{a_j^n\,\Delta t}{4\,\Delta x}\right)u_{j+1}^{n+1}$$
$$=\left(\frac{\epsilon_j^n\,\Delta t}{2\,(\Delta x)^2}-\frac{a_j^k\,\Delta t}{4\,\Delta x}\right)u_{j-1}^{n}+\left(c_j^n-\frac{\epsilon_j^n\,\Delta t}{(\Delta x)^2}-\frac{b\,\Delta t}{2}\right)u_j^{n}+\left(\frac{\epsilon_j^n\,\Delta t}{2\,(\Delta x)^2}+\frac{a_j^n\,\Delta t}{4\,\Delta x}\right)u_{j+1}^{n}. \tag{5.23}$$

For the Crank-Nicolson scheme, using upstream weighting does not guarantee a positive coefficient discretisation, as there is also a further restriction on the timestep that must be satisfied. However, **Rannacher smoothing** [33] — where two implicit steps are applied before Crank-Nicolson timestepping is used – usually damps out any initial spurious oscillations while maintaining $O(h^2)$ convergence.

5.4.3 The Fitted Duffy Scheme

Daniel J. Duffy in his doctoral thesis [11] developed a scheme for this type of problem that would be uniformly convergent. The scheme works by replacing the coefficient of diffusion with a **fitting factor**, which is defined as follows:

$$\sigma_j^n = \frac{a_j^n\,\Delta x}{2}\coth\left(\frac{a_j^n\,\Delta x}{2\epsilon_j^n}\right). \tag{5.24}$$

Note that this is a distinct quantity from the volatility of the underlying, and is an unfortunate clash of notation, but confusion can be avoided by noting that the fitting factor will also have a superscript and subscript, and the volatility will not.

Thus, the Fitted Duffy Scheme is as follows:

$$c_j^{n+1}\left(\frac{u_j^{n+1}-u_j^n}{\Delta t}\right)+\sigma_j^{n+1}\left(\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{(\Delta x)^2}\right) + a_j^{n+1}\left(\frac{u_{j+1}^{n+1}-u_{j-1}^{n+1}}{2\,\Delta x}\right)-b_j^{n+1}u_j^{n+1}=0 \tag{5.25}$$

and rearranging,

$$\left(\frac{\sigma_j^{n+1}}{(\Delta x)^2}-\frac{a_j^{n+1}}{2\Delta x}\right)u_{j-1}^{n+1}+\left(-b_j^{n+1}-2\frac{2\sigma_j^{n+1}}{(\Delta x)^2}-\frac{c_j^{n+1}}{\Delta t}\right)u_j^{n+1} + \left(\frac{\sigma_j^{n+1}}{(\Delta x)^2}+\frac{a_j^{n+1}}{2\Delta x}\right)u_{j+1}^{n+1}=-\frac{c_j^n}{\Delta t}u_j^n. \tag{5.26}$$

Note that as volatility gets smaller, $\sigma \to 0$, the discretisation of the $\frac{\partial V}{\partial S}$ in the Fitted Duffy scheme degenerates to upstream differencing.

5.4.4 The Fitted Crank-Nicolson Scheme

Since the Crank-Nicolson scheme has problems with these 'numerical' oscillations, we decided to try doing something equivalent for it. In the fitted Crank-Nicolson, the coefficient of diffusion, ϵ, is again replaced with this fitting factor, σ_j^n and the scheme is developed accordingly.

$$c_j^{n+1}\left(\frac{u_j^{n+1}-u_j^n}{\Delta t}\right)+\sigma_j^{n+1}\left[\frac{1}{2}\left(\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{(\Delta x)^2}\right)+\frac{1}{2}\left(\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{(\Delta x)^2}\right)\right]$$
$$+a_j^{n+1}\left[\frac{1}{2}\left(\frac{u_{j+1}^{n+1}-u_{j-1}^{n+1}}{2\,\Delta x}\right)+\frac{1}{2}\left(\frac{u_{j+1}^n-u_{j-1}^n}{2\,\Delta x}\right)\right]-b_j^{n+1}\left(\frac{1}{2}u_j^{n+1}+\frac{1}{2}u_j^n\right)=0, \tag{5.27}$$

and rearranging,

$$\left(\frac{\sigma_j^n\Delta t}{2\,(\Delta x)^2}-\frac{a_j^k\Delta t}{4\,\Delta x}\right)u_{j-1}^{n+1}+\left(-c_j^n-\frac{\sigma_j^n\,\Delta t}{(\Delta x)^2}-\frac{b\,\Delta t}{2}\right)u_j^{n+1}+\left(\frac{\sigma_j^n\,\Delta t}{2\,(\Delta x)^2}+\frac{a_j^n\,\Delta t}{4\,\Delta x}\right)u_{j+1}^{n+1}$$
$$=\left(\frac{\sigma_j^n\,\Delta t}{2\,(\Delta x)^2}-\frac{a_j^k\,\Delta t}{4\,\Delta x}\right)u_{j-1}^n+\left(c_j^n-\frac{\sigma_j^n\Delta t}{(\Delta x)^2}-\frac{b\Delta t}{2}\right)u_j^n+\left(\frac{\sigma_j^n\,\Delta t}{2(\Delta x)^2}+\frac{a_j^n\,\Delta t}{4\,\Delta x}\right)u_{j+1}^n. \tag{5.28}$$

5.4.5 The Van Leer Flux-Limiter

The Van Leer scheme is a slightly more complex scheme, first used to solve options by Zvan et al. [43]. It is non-linear, making it more difficult to implement.

A finite volume approach is taken, with the Black-Scholes equation Eq. (1.14) being rewritten in the form

$$\frac{V_i^{n+1}-V_i^n}{\Delta t}=\theta F_{i-\frac{1}{2}}^{n+1}\;-\;\theta F_{i+\frac{1}{2}}^{n+1}\;+\;\theta f_i^{n+1}\;+$$
$$(1-\theta)F_{i-\frac{1}{2}}^{n+1}\;-\;(1-\theta)F_{i+\frac{1}{2}}^{n+1}\;+\;(1-\theta)f_i^{n+1}, \tag{5.29}$$

where

$$\begin{aligned}\theta &= \text{temporal weighting } (0\le\theta\le 1),\\ F_{i-\frac{1}{2}} &= \text{flux entering cell } i \text{ at interface } i-\frac{1}{2},\\ F_{i+\frac{1}{2}} &= \text{flux leaving cell } i \text{ at interface } i+\frac{1}{2},\\ f_i &= \text{source/sink term.}\end{aligned} \tag{5.30}$$

In particular,

$$
\begin{aligned}
F_{i-\frac{1}{2}}^{n+1} &= \frac{1}{\Delta S_i}\left(-\frac{1}{2}\sigma^2 S^2 \frac{V_i^{n+1} - V_{i-1}^{n+1}}{\Delta S_{i-\frac{1}{2}}} + (-rS_i)\, V_{i-\frac{1}{2}}^{n+1}\right), \\
F_{i+\frac{1}{2}}^{n+1} &= \frac{1}{\Delta S_i}\left(-\frac{1}{2}\sigma^2 S^2 \frac{V_{i+1}^{n+1} - V_{i}^{n+1}}{\Delta S_{i+\frac{1}{2}}} + (-rS_i)\, V_{i+\frac{1}{2}}^{n+1}\right), \\
f_i^{n+1} &= (-r)\, V_i^{n+1}.
\end{aligned}
\tag{5.31}
$$

The above scheme is generalised for a non-uniform mesh.

Matters are simplified for a uniform mesh, since δS_i is a constant, and different schemes can be constructed by choosing different assumptions about how the value of the convective terms $V_{i+\frac{1}{2}}$ and $V_{i_{\frac{1}{2}}}$ are calculated. A central weighting scheme,

$$
V_{i+\frac{1}{2}}^{n+1} = \frac{1}{2}(V_{i+1}^{n+1} + V_i^{n+1}), \tag{5.32}
$$

gives us the familiar Crank-Nicolson for $\theta = \frac{1}{2}$, and the fully implicit scheme for $\theta = 1$.

The Van Leer approach is slightly more complicated, instead of taking the mid-point of the line-segment, it calculates a value depending on the gradient of the solution at that mesh-point. This will then make the scheme Total Variation Diminishing (TVD), which prevents oscillations. The formula used is:

$$
V_{i+\frac{1}{2}}^{n+1} = V_{\mathrm{up}}^{n+1} + \frac{1}{2}\,\phi\,(q_{i+\frac{1}{2}}^{n+1})\,(V_{\mathrm{down}}^{n+1} - V_{\mathrm{up}}^{n+1}). \tag{5.33}
$$

The direction of upstream is determined by the sign of the rS term, and since this must always be negative, $V_{i+\frac{1}{2}\mathrm{up}}$ is V_{i+1}. $V_{i+\frac{1}{2}\mathrm{down}}$ is V_i.

In Eq. (5.33), the value of q is determined by

$$
q_{i+\frac{1}{2}}^{n+1} = \frac{\left(\frac{V_{\mathrm{up}}^{n+1} - V_{2\mathrm{up}}^{n+1}}{S_{2\mathrm{up}} - S_{\mathrm{up}}}\right)}{\left(\frac{V_{\mathrm{down}}^{n+1} - V_{\mathrm{up}}^{n+1}}{S_{\mathrm{up}} - S_{\mathrm{down}}}\right)}, \tag{5.34}
$$

and

$$\phi = \frac{|q_{i+\frac{1}{2}}^{n+1}| + q_{i+\frac{1}{2}}^{n+1}}{1 + |q_{i+\frac{1}{2}}^{n+1}|}. \tag{5.35}$$

This scheme is non-linear, and so a linearised approximation is required for its implementation. This is done by calculating a starting ϕ using the values of V from the previous time-step.

This value of ϕ is then used to calculate a new V. This new V is then used to re-calculate a better approximation to ϕ, which in turn produces a better V, and so on until the value of the calculated ϕ stays sufficiently constant. The value of V taken from this final ϕ is then taken to be the value used.

It is immediately apparent that this means that the Van Leer scheme may take a significantly longer time to execute due to the required re-iterations at each time-step. In practice however, this is not necessary, as the first approximation (using the values obtained from the previous timestep) gives sufficiently accurate answers that the convergence iterations are not required, greatly enhancing the efficiency of the scheme.

5.5 Analysis of the Schemes for European Options

A full scaling analysis was performed for all the schemes, using uniform mesh-sizes of 100, 200, 400, 800 and 1600 meshpoints in the $S-$direction. 50 steps in the time direction were used initially.

Two factors were important in the analysis of these schemes, speed and accuracy.

The speed of a scheme was measured by timing its execution for each scheme using the internal timing mechanism of the computer architecture. This resulted in a time which was given in seconds, which is considerably less than the actual time require to execute the software, as it does not take into account such factors as file I/O and software multitasking. However, since all that is counted is the actual time the software was using system resources, this has the advantage that it is an excellent method of comparing relative values of the timings for the different schemes. Ambient factors such as the load and number of users on the system are removed by default.

Thus, although the values obtained may be of no use for absolute timings of the scheme, they are very good for calculating relative values of the speed of the schemes.

5.5.1 Error Analysis

Comparing the accuracy of the schemes was somewhat more involved.

The Black-Scholes equation is solvable analytically for European options under the conditions that the interest-rate and volatility are fixed and constant over the lifetime of the option. Under these (admittedly restrictive) conditions, the calculation of the option price is straight-forward and computationally inexpensive. Thus, the solutions generated by the four different schemes were compared to the exact solution, and a simple error-calculation routine was applied to allow quantitative analysis to be performed.

Calculating the Average Norm

The error routine used to calculate an error value for each scheme was:

$$\mathrm{Err} = \frac{\int_\Omega dS\,dt\,(u_{\mathrm{num}} - U_{\mathrm{ex}})^2 W(S,t)}{\int_\Omega dS\,dt\,U_{\mathrm{ex}}^2 W(S,t)}, \tag{5.36}$$

where

$W(S,t)$	is a weighting function over the solution domain,
u_{num}	is the solution generated by the numerical scheme, and
U_{ex}	is the exact solution.
Ω	is the domain over which the error analysis is performed.

The inclusion of the weighting function, $W(S,t)$ in Eq. (5.36) also gave the flexibility of performing a local analysis of the solution around specific areas of the solution domain, as will be seen later.

The above error analysis was performed on both the solution of the Black-Scholes equation, and the sensitivities of these solutions. Cross-sections of the full 2D surface were also taken to provide a visual aid to the accuracy of the schemes.

Calculating the Max Norm

Maximum norms are also very insightful for calculating errors for solution schemes to singly-perturbed problems. The maximum norm is simply

$$||u|| = \sup_{x\in\Omega} |u_j^n - U_j^n|, \tag{5.37}$$

where u_j^n is the numerical solution at the mesh-point (n, j) and U_j^n is the exact solution at the mesh-pooint (n, j)

For the purposes of this discussion, one set of parameters was used to calculate the solution. The Black-Scholes code developed required five parameters to specify the problem. These five parameters were T, the time to expiry; K, the strike price; $S_{\max}$, the maximum stock value used (typically set to be $10K$, ten times the strike price); σ, the volatility of the underlying asset; and r, the risk-free interest rate.

The parameter values where

$$\begin{aligned} T &= 1, \\ K &= 1, \\ S_{\max} &= 10, \\ \sigma &= 0.01, \\ r &= 15\% \end{aligned}$$

This discussion is limited to vanilla European call options. Put options are almost identical mathematically, and no exact solution exists for the American options, making it more difficult to perform an accuracy analysis.

5.5.2 Comparison of the Timings

It is immediately apparent from Table 5.1 that the quadrupling of the number of mesh points leads to an approximate ten-fold increase in the execution time of the

Scheme	500x500	1000x1000
Fully Implicit	1.750000	7.210938
Fitted Duffy	2.281250	9.539062
Crank-Nicolson	1.851562	7.632812
Fitted Crank-Nicolson	2.406250	10.015625
Van-Leer Flux Limiter	3.320312	13.250000

Table 5.1: Execution Times for the Numerical Schemes

Scheme	Ratio
Fully Implicit	1.00
Crank-Nicolson	1.06
Fitted Duffy	1.31
Fitted Crank-Nicolson	1.38
Van-Leer Flux Limiter	1.87

Table 5.2: Execution Time Ratios for the Numerical Schemes

code. On both meshes, the Fully Implicit method is the quickest, followed closely by the Crank-Nicolson method, with a big jump up to the fitted Duffy method. The Van Leer flux-limiting method is the slowest.

The execution times of the non-fitted schemes are roughly in the same proportion over the two meshes, as are the execution times for the two fitted schemes.

Table 5.2 gives an estimate of the execution times of the four discretisation schemes. The execution times for the two mesh-sizes was measured and averaged and compared to the corresponding time for the Fully Implicit scheme. This number is the ratio of those two timings. Thus, we have a measure of the relative performance of the schemes against the Fully Implicit.

5.5.3 Comparison of the Accuracy of the Schemes

The maximum norm is much more insightful for an error-analysis of singly-perturbed problems, the discussion will commence with this metric.

Max-Norm Analysis

Tables 5.3 - 5.6 show the error tables obtained for the 1600 x 1600 mesh-size, with an error calculation performed for the Call option price and Greeks.

It is readily seen from these tables that the Van-Leer flux limiter is the best performing scheme in terms of the Greeks, producing errors which are sometimes an order of magnitude lower than those produced by the other schemes.

Error-Scaling Analysis

A scaling analysis was also performed using all five mesh-sizes. The analysis was performed by taking the error measure of each scheme (measured against the data found from the exact solution), and plotting a graph of the log of the error against the step size of the mesh (i.e. the graph was ln Error vs $\ln \delta x$). The slope of this will give an approximation of the order of convergence for the errors.

The expected result is a straight line (showing that the scheme is $O(h^n)$ for some n, which will be the slope of the line in the graph).

The results obtained are shown in Figure 5-3. The lines shown are connecting the datapoints, and no fitting has been done on the data.

Calculating the scaling of errors for the Greeks is much more difficult since the Greeks are approximated by the finite difference approximations of the solutions. We would expect the Gamma to be particularly bad in this regard.

The results for the scaling of the Greeks is shown in Figs 5-4, 5-5, and 5-6.

As can be seen in Fig 5-3, a logscale plot of the errors against stepsize for each discretization scheme are roughly linear. The straightest one is the implicit scheme, and this also gives the best errors, as its scaling line is the lowest on the graph.

The Van Leer is also roughly linear (allowing for a small amount of noise) and the other schemes are slightly curved, showing that rate of increase in error slows down as the mesh gets coarser.

The error scaling of the Greeks was better than I had expected. While not displaying a linear relationship under a logscale, in most instances they exhibit the trend of decreasing the error as the stepsize gets smaller.

	Exact	Imp	Duff	CN	FittCN	VanLeer
Exact	—	0.002	0.009	0.002	0.009	0.001
Imp	0.002	—	0.008	0.000	0.008	0.002
Duff	0.009	0.008	—	0.008	0.000	0.009
CN	0.002	0.000	0.008	—	0.008	0.002
FittCN	0.009	0.008	0.000	0.008	—	0.008
VanLeer	0.001	0.002	0.009	0.002	0.008	—

Table 5.3: European call option price errors for mesh-size 1600×1600 with $K = 1$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$ using the Maximum Norm metric

	Exact	Imp	Duff	CN	FittCN	VanLeer
Exact	—	0.198	0.282	0.199	0.281	0.062
Imp	0.198	—	0.309	0.013	0.307	0.186
Duff	0.282	0.309	—	0.324	0.002	0.267
CN	0.199	0.013	0.324	—	0.320	0.188
FittCN	0.281	0.307	0.002	0.320	—	0.265
VanLeer	0.062	0.186	0.267	0.188	0.265	—

Table 5.4: European call option Delta errors for mesh-size 1600×1600 with $K = 1$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$ using the Maximum Norm metric

	Exact	Imp	Duff	CN	FittCN	VanLeer
Exact	—	77.974	61.739	78.230	61.326	40.398
Imp	77.974	—	59.111	1.466	59.086	47.411
Duff	61.739	59.111	—	59.697	0.542	38.329
CN	78.230	1.466	59.697	—	59.672	48.023
FittCN	61.326	59.086	0.542	59.672	—	38.071
VanLeer	40.398	47.411	38.329	48.023	38.071	—

Table 5.5: European call option Gamma errors for mesh-size 1600×1600 with $K = 1$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$ using the Maximum Norm metric

	Exact	Imp	Duff	CN	FittCN	VanLeer
Exact	—	0.072	0.131	0.072	0.131	0.069
Imp	0.072	—	0.059	0.003	0.059	0.026
Duff	0.131	0.059	—	0.059	0.003	0.062
CN	0.072	0.003	0.059	—	0.059	0.028
FittCN	0.131	0.059	0.003	0.059	—	0.062
VanLeer	0.069	0.026	0.062	0.028	0.062	—

Table 5.6: European call option Theta errors for mesh-size 1600×1600 with $K = 1$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$ using the Maximum Norm metric

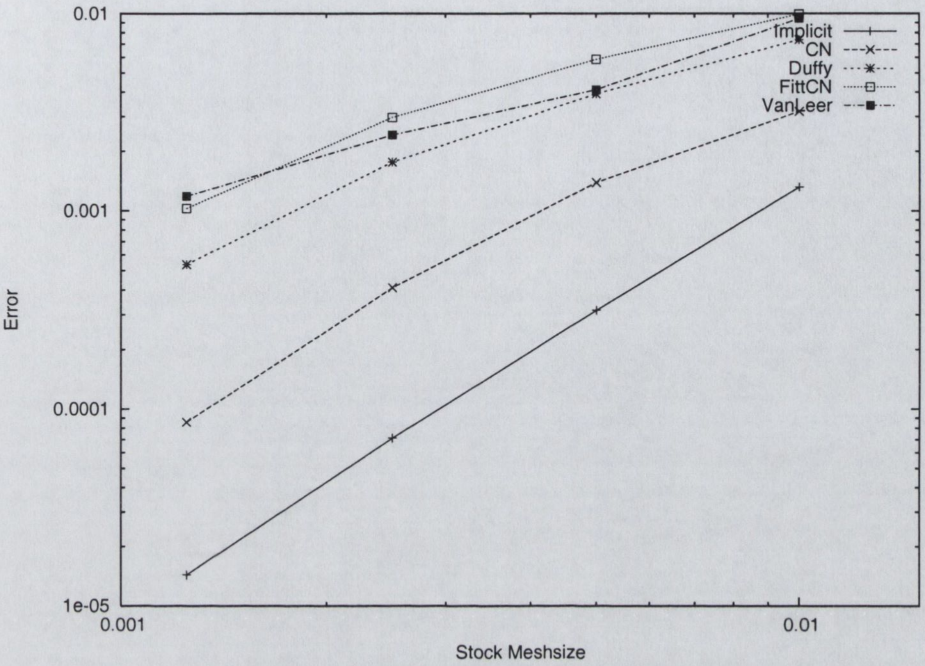

Figure 5-3: Scaling Analysis of the Price of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 50 timesteps ($\delta t = 0.02$) using the Maximum Norm metric

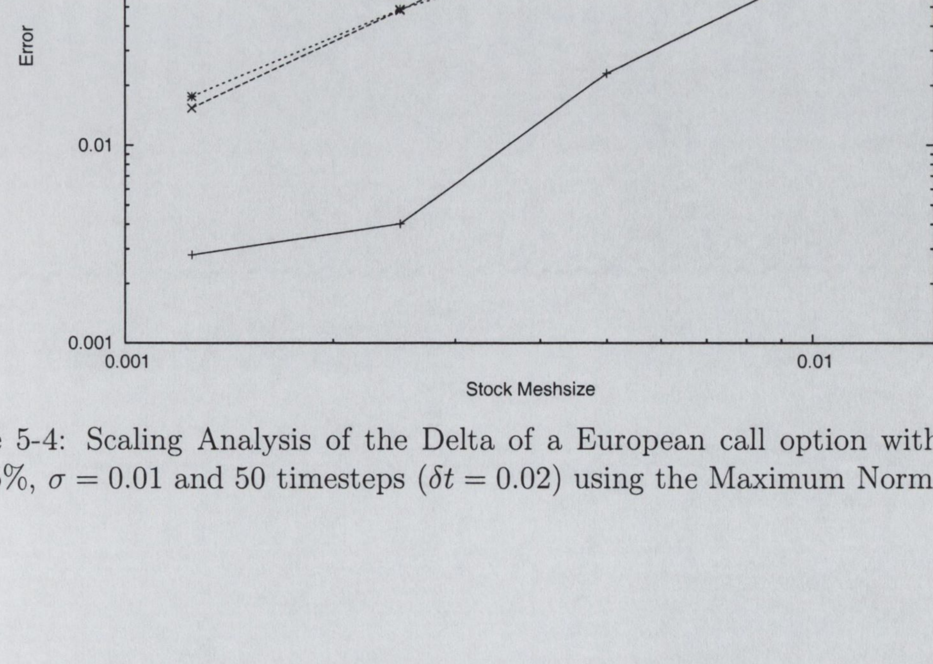

Figure 5-4: Scaling Analysis of the Delta of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 50 timesteps ($\delta t = 0.02$) using the Maximum Norm metric

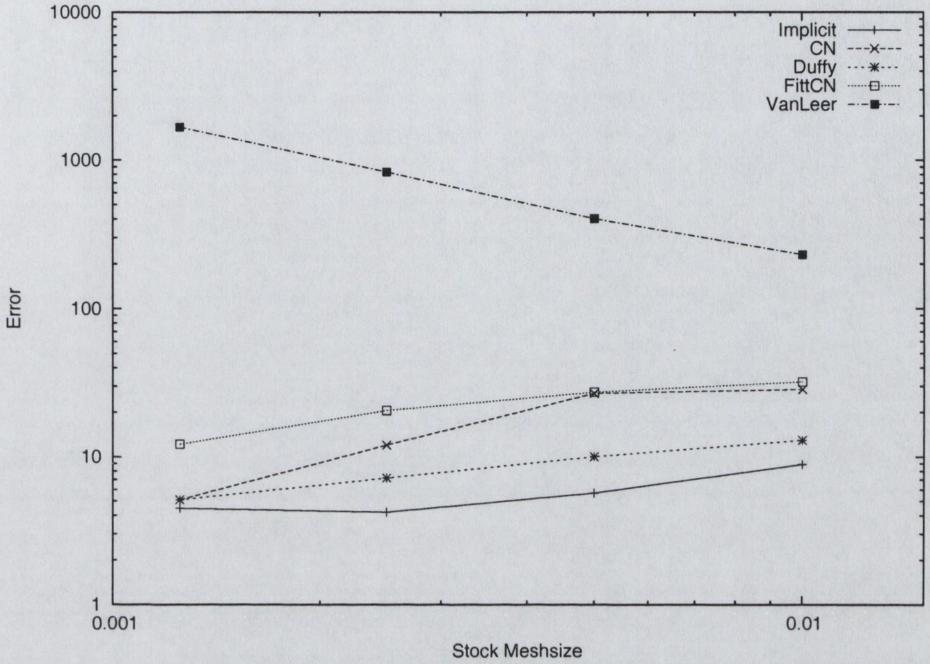

Figure 5-5: Scaling Analysis of the Gamma of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 50 timesteps ($\delta t = 0.02$) using the Maximum Norm metric

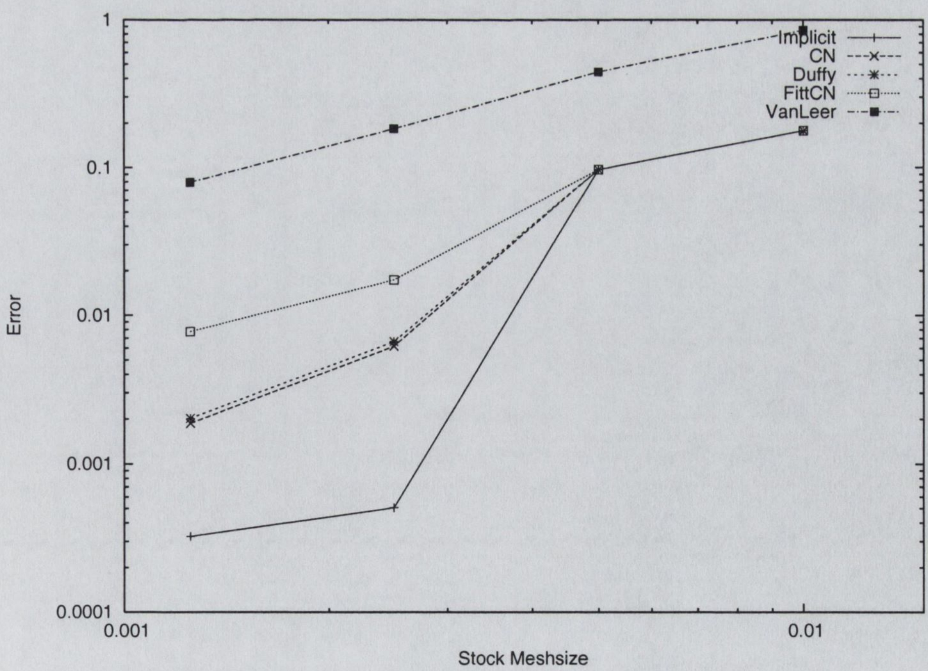

Figure 5-6: Scaling Analysis of the Theta of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 50 timesteps ($\delta t = 0.02$) using the Maximum Norm metric

It is possible that the high value for the timestep could distort these error calculations, and so to check this, the calculation was also performed for 1000 steps in the time direction. Another scaling analysis was performed, with every other parameter held unchanged.

The scaling of the errors are shown in Figs. 5-7, 5-8, 5-9, 5-10.

These graphs are very similar to those with 50 timesteps (Figs 5-3 - 5-6) showing that the errors due to the stockstep have a greater influence on the total error than those due to the timestep size.

Similar patterns emerge as for the 50 timestep case, although the error scaling for the Greeks is more unpredictable with sharp drops in errors despite an increase in stepsize.

Again, the errors for the option price scaling in a approximately linear fashion with the stepsize, and again the Implicit scheme has the lowest errors for a particular stepsize.

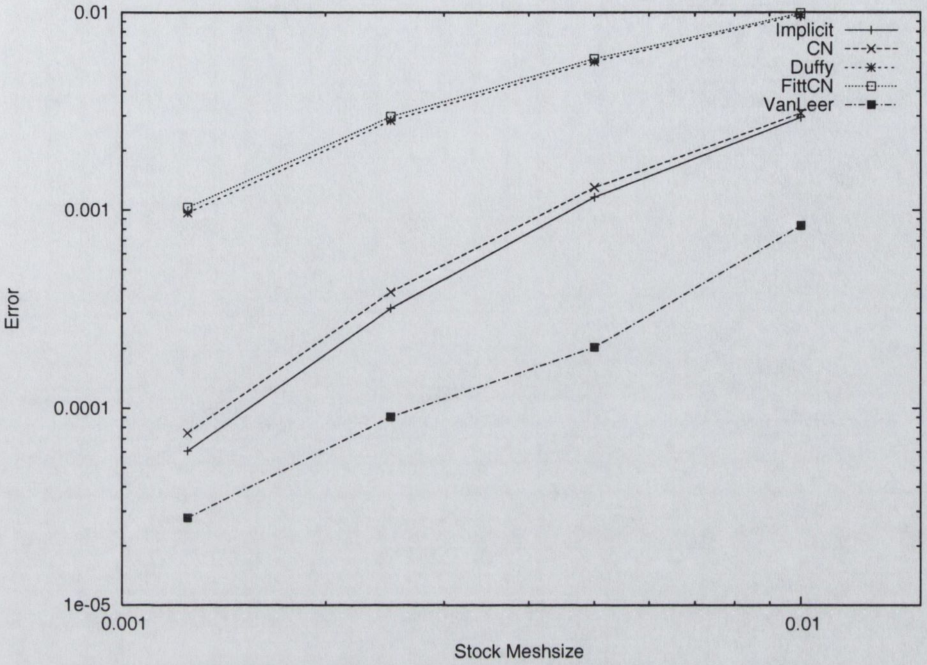

Figure 5-7: Scaling Analysis of the Price of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 1000 timesteps ($\delta t = 0.001$) using the Maximum Norm metric

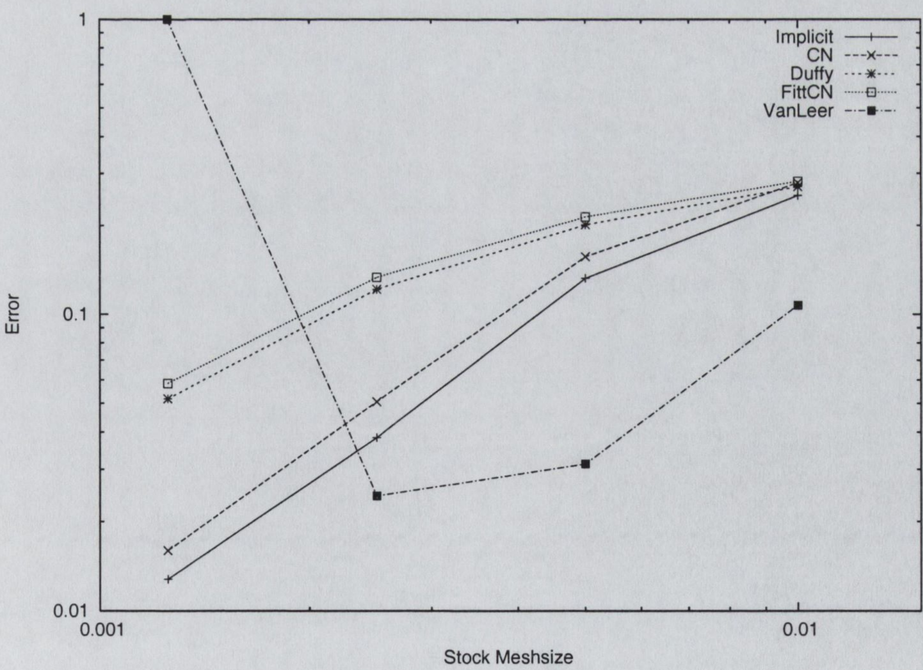

Figure 5-8: Scaling Analysis of the Delta of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 1000 timesteps ($\delta t = 0.001$) using the Maximum Norm metric

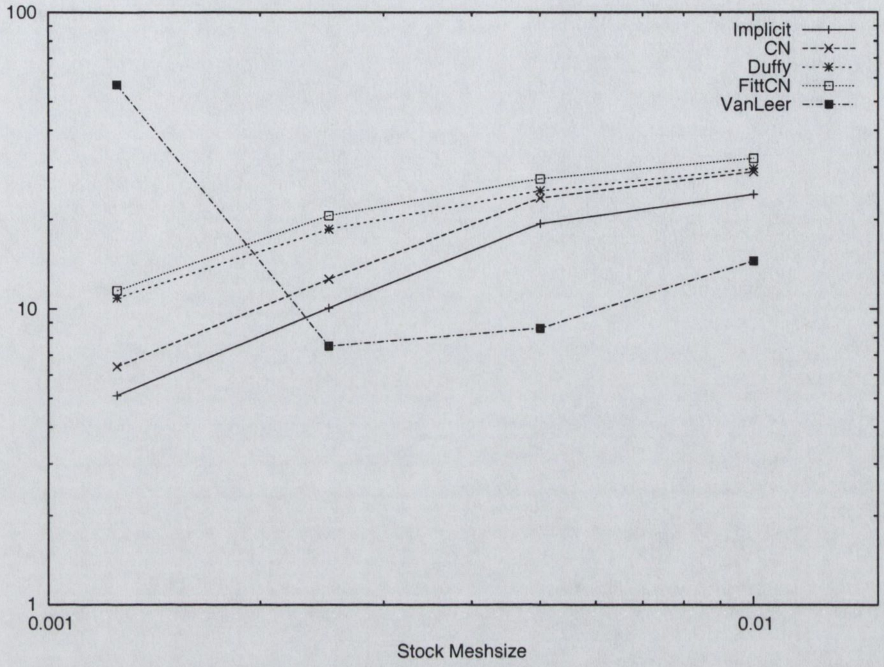

Figure 5-9: Scaling Analysis of the Gamma of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 1000 timesteps ($\delta t = 0.001$) using the Maximum Norm metric

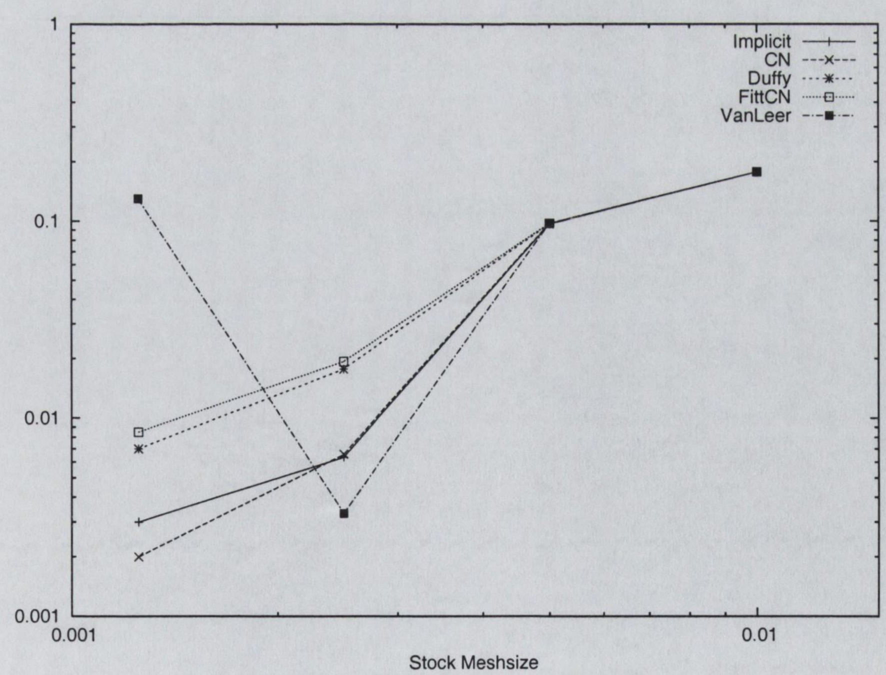

Figure 5-10: Scaling Analysis of the Theta of a European call option with $K = 1$, $r = 15\%$, $\sigma = 0.01$ and 1000 timesteps ($\delta t = 0.001$) using the Maximum Norm metric

5.5.4 Summary

Looking at Tables 5.3 - 5.6, the errors calculated for the fitted Crank-Nicolson scheme are the largest for the Price, the Delta and the Theta, but the difference is only marginal. They are also quite large for the Gamma.

Figs. 5-11 – 5-14 shows a plot of the cross-sections of the price, Delta, Gamma and Theta taken at $T - t = 0.1$.

The oscillations are shown to be especially bad for the Gamma, but are also very obvious with the Delta and Theta. The two fitted methods, however, do not contain any oscillations. This is as expected, since they were designed to prevent to spurious oscillations.

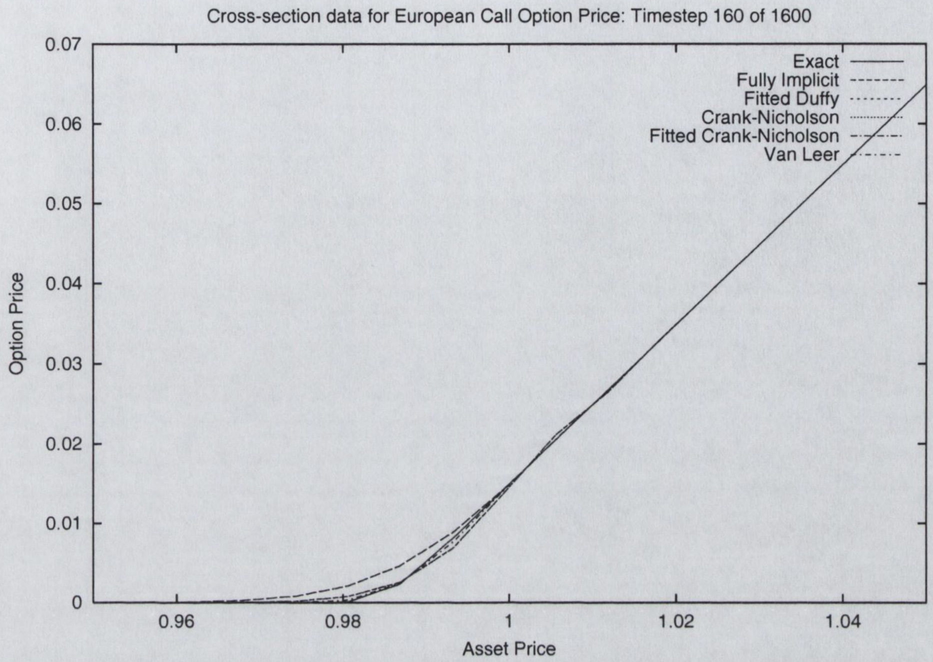

Figure 5-11: Plots of the European call option price, V around the money. The parameters are $K = 1$, $r = 15\%$, $\sigma = 0.01$, $T = 1$, $\delta S = 0.00625$, $\delta t = 0.000625$

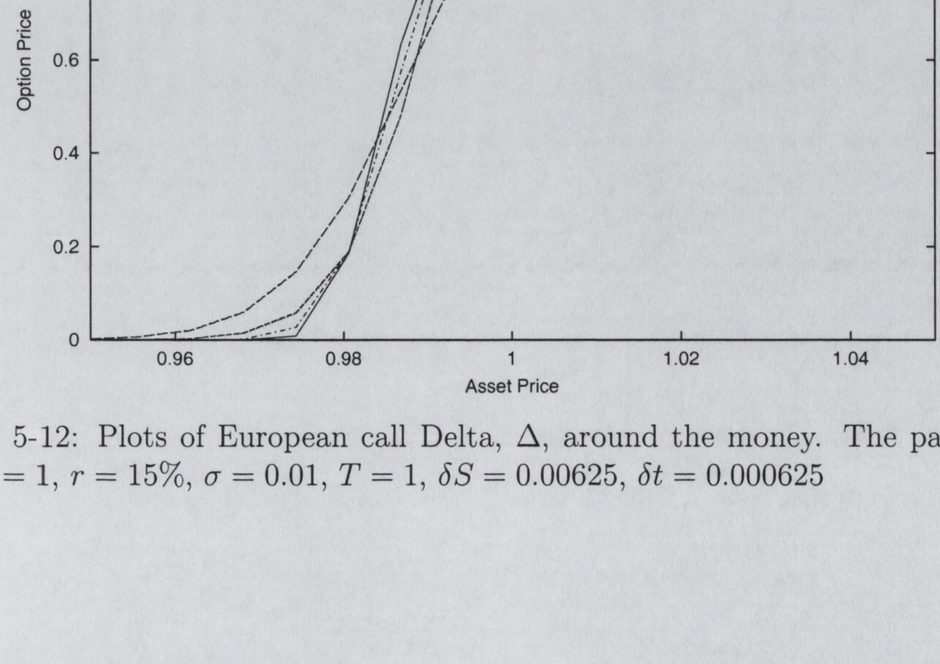

Figure 5-12: Plots of European call Delta, Δ, around the money. The parameters are $K = 1$, $r = 15\%$, $\sigma = 0.01$, $T = 1$, $\delta S = 0.00625$, $\delta t = 0.000625$

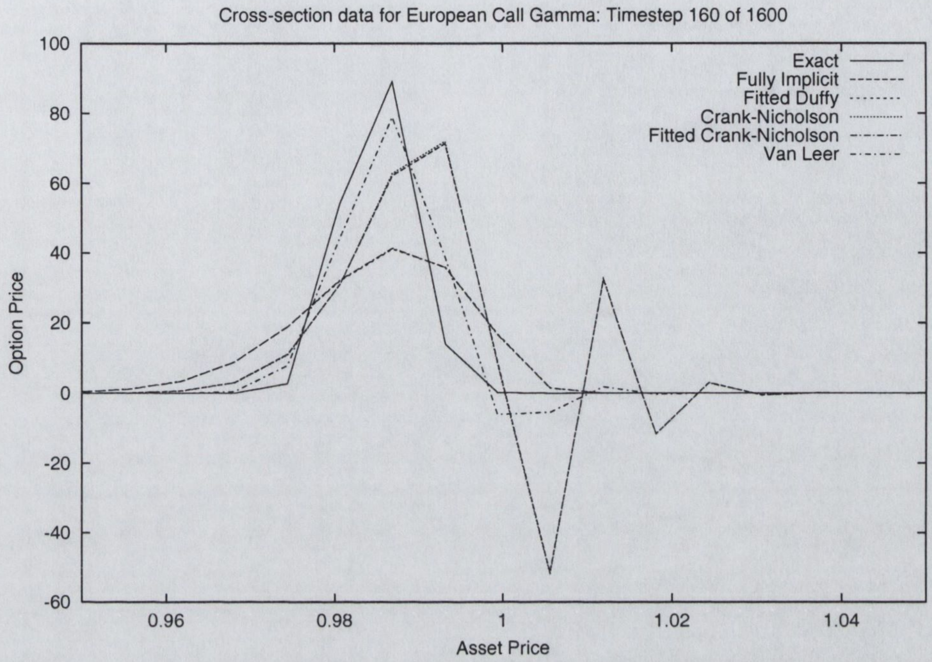

Figure 5-13: Plots of European call Gamma, Γ, around the money. The parameters are $K = 1$, $r = 15\%$, $\sigma = 0.01$, $T = 1$, $\delta S = 0.00625$, $\delta t = 0.000625$

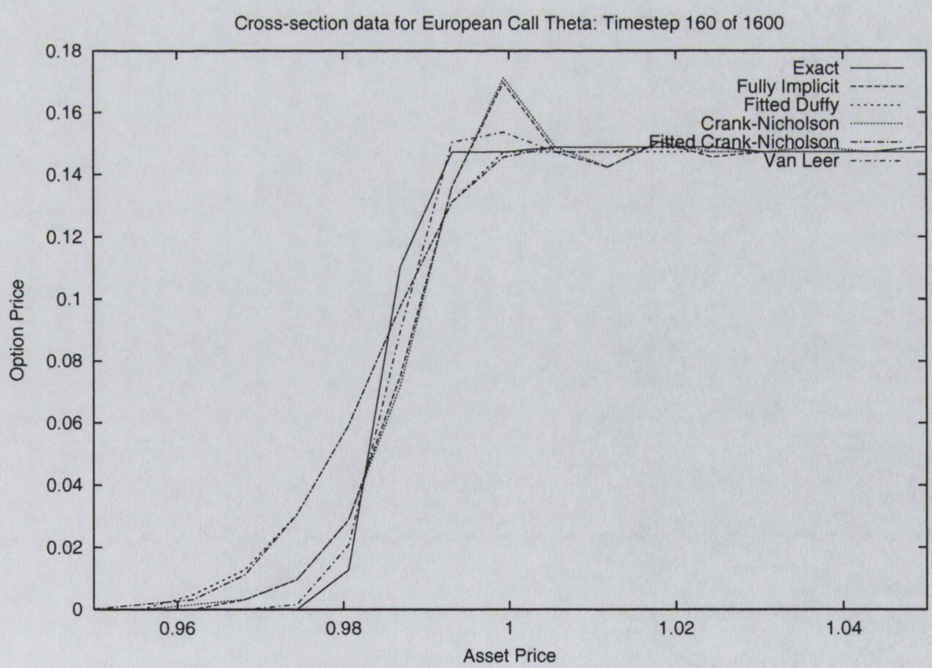

Figure 5-14: Plots of European call Theta, Θ, around the money. The parameters are $K = 1$, $r = 15\%$, $\sigma = 0.01$, $T = 1$, $\delta S = 0.00625$, $\delta t = 0.000625$

5.6 Analysis of the Schemes for American Options

The only difference between the American and European options is the fact that American options can be exercised at any time up to the expiry date. European options may only be exercised at the time of expiry.

This added complexity to the problem translates mathematically into a sub-region of the solution domain where early exercise is optimal. This region is not fixed, and is dependent on the parameters of the problem, as well as the value of the underlying and the time to expiry.

Thus, we have a free boundary which moves as time progresses. This free boundary has its own conditions (known as **free boundary conditions**), which the solution must also satisfy. It is the added complication of this free boundary which has prevented the American option pricing problem from being solved analytically, although approximate analytic solutions do exist.

When solving the American option pricing problem numerically, it is actually quite easy to account for the free boundary, creating only a small increase in execution time.

At each time step, the option price at each mesh-point is calculated, and is then compared to the payoff of the option at that time. If the option price is lower than the instantaneous payoff, then the option value is reset to be equal to this payoff. This must be done to avoid arbitrage opportunities, since if this were not done, it would be possible to buy the option and the underlying and immediately exercise the option to make an immediate risk-free profit. Thus, the value of the option in this case must be equal to the payoff.

Thus, the region in which this value reset occurs is the region of optimal exercise and the free boundary can easily be tracked by just taking note of the mesh-points where the change-over from taking the calculated value to the payoff value occurs.

In terms of coding, this simply requires the addition of a value-checking statement, which decides whether or not to reset the value of the put option. Tracking the free boundary is somewhat more involved but is also quite simple to implement.

Fig. 5-15 shows a comparison between the prices of a European and American Put for the same parameters as before, with the cross-section taken at the same

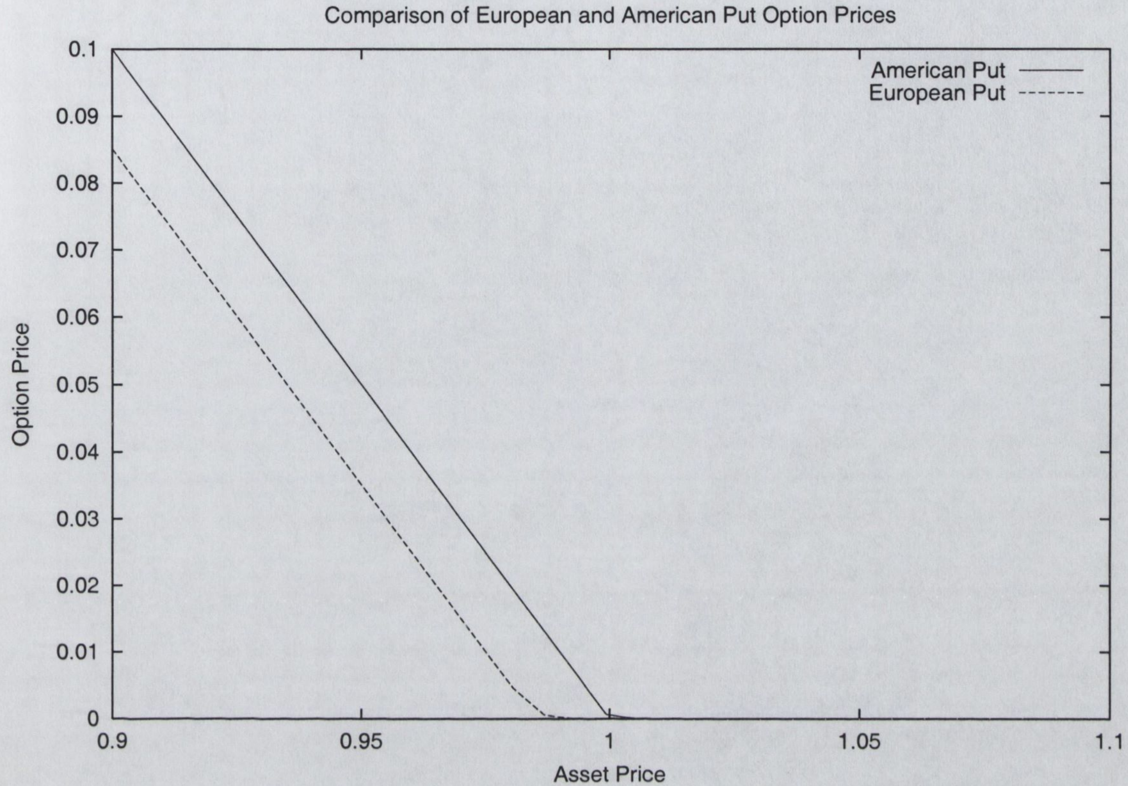

Figure 5-15: Comparison of European vs American Put Prices

time-step, $T - t = 0.1$. It is quite easy to see that the value of the American Put is higher than that of the European, and this value is that of the payoff of the option, suggesting that it is optimal to exercise the option at this time-step, if the option has not already been exercised.

The majority of this error analysis used a parameter configuration similar to those used in Chapter 6. We restrict the timesteps to 50 timesteps per year, and we use a constant interest rate of 6%.

5.6.1 Error analysis

Estimating the errors for the schemes for the American option solutions is not as simple as it is for European options, since there is no exact solution to use as a benchmark.

Scheme Comparison Analysis

One method was to use each scheme as the benchmark, and compare all the other schemes to it. Again, both the Average and Max Norm errors were used, and the results of this are tabulated in Tables 5.7 – 5.38

One thing is immediately apparent from these tables: there is very close agreement between the Implicit and Duffy schemes, and also between the Crank-Nicolson, Fitted Crank-Nicolson and VanLeer schemes.

The solution errors are very small between schemes of these groups and much larger between schemes in the other group. To cite an example, using the Average error calculation and the Implicit Scheme as a basis, the error for the Duffy scheme is $O(10^{-8})$, yet is $O(10^{-3})$ for other schemes.

Using the Max norm, these differences are even more pronounced: again using the Implicit scheme as the basis, the error in the Duffy scheme is $O(10^{-16})$, compared to $O(10^{-7})$ for the others.

Similarly, using the Crank-Nicolson scheme as the basis, we get similar results: the Implicit and Duffy schemes have errors of $O(10^{-3})$ and $O(10^{-7})$ for the Average and Max Norm errors, compared to errors in the Fitted Crank-Nicolson and VanLeer schemes of $O(10^{-8})$ and $O(h^{-16})$ respectively.

The reason for this is pretty straightforward: the Implicit and Duffy schemes are $O(h)$ schemes, with the other three being $O(h^2)$, so it is obvious there will be more agreement intra-group than inter-group.

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	5.19e-08	4.09e-03	4.09e-03	4.09e-03
Duff	5.19e-08	—	4.09e-03	4.09e-03	4.09e-03
CN	4.09e-03	4.09e-03	—	5.01e-08	6.49e-07
FittCN	4.09e-03	4.09e-03	5.01e-08	—	6.24e-07
VanLeer	4.09e-03	4.09e-03	6.49e-07	6.24e-07	—

Table 5.7: Error table for American Put Price, V, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.34e-07	1.15e-02	1.15e-02	1.15e-02
Duff	3.34e-07	—	1.15e-02	1.15e-02	1.15e-02
CN	1.15e-02	1.15e-02	—	3.23e-07	1.50e-05
FittCN	1.15e-02	1.15e-02	3.23e-07	—	1.47e-05
VanLeer	1.15e-02	1.15e-02	1.50e-05	1.47e-05	—

Table 5.8: Error table for American Put Delta, Δ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.17e-05	5.34e-01	5.34e-01	5.35e-01
Duff	3.17e-05	—	5.34e-01	5.34e-01	5.35e-01
CN	5.34e-01	5.34e-01	—	3.13e-05	6.88e-04
FittCN	5.34e-01	5.34e-01	3.13e-05	—	6.72e-04
VanLeer	5.35e-01	5.35e-01	6.88e-04	6.72e-04	—

Table 5.9: Error table for American Put Gamma, Γ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.21e-06	2.44e-01	2.44e-01	2.44e-01
Duff	1.21e-06	—	2.44e-01	2.44e-01	2.44e-01
CN	2.44e-01	2.44e-01	—	1.25e-06	3.25e-05
FittCN	2.44e-01	2.44e-01	1.25e-06	—	3.12e-05
VanLeer	2.44e-01	2.44e-01	3.25e-05	3.12e-05	—

Table 5.10: Error table for American Put Theta, Θ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	4.07e-16	1.24e-07	1.24e-07	1.24e-07
Duff	4.07e-16	—	1.24e-07	1.24e-07	1.24e-07
CN	1.24e-07	1.24e-07	—	4.14e-16	4.39e-14
FittCN	1.24e-07	1.24e-07	4.14e-16	—	3.97e-14
VanLeer	1.24e-07	1.24e-07	4.39e-14	3.97e-14	—

Table 5.11: Error table for American Put Price, V, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	2.02e-11	2.94e-04	2.94e-04	2.95e-04
Duff	2.02e-11	—	2.94e-04	2.94e-04	2.95e-04
CN	2.94e-04	2.94e-04	—	2.06e-11	2.43e-10
FittCN	2.94e-04	2.94e-04	2.06e-11	—	2.46e-10
VanLeer	2.94e-04	2.94e-04	2.43e-10	2.46e-10	—

Table 5.12: Error table for American Put Delta, Δ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	2.48e-10	5.00e-04	5.00e-04	5.01e-04
Duff	2.48e-10	—	5.00e-04	5.00e-04	5.01e-04
CN	5.00e-04	5.00e-04	—	2.51e-10	1.25e-09
FittCN	5.00e-04	5.00e-04	2.51e-10	—	1.27e-09
VanLeer	5.01e-04	5.01e-04	1.25e-09	1.27e-09	—

Table 5.13: Error table for American Put Gamma, Γ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	4.91e-10	4.96e-02	4.96e-02	4.96e-02
Duff	4.91e-10	—	4.96e-02	4.96e-02	4.96e-02
CN	5.63e-02	5.63e-02	—	4.39e-10	9.22e-09
FittCN	5.63e-02	5.63e-02	4.39e-10	—	9.27e-09
VanLeer	5.63e-02	5.63e-02	9.22e-09	9.27e-09	—

Table 5.14: Error table for American Put Theta, Θ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.22e-08	8.24e-03	8.24e-03	8.24e-03
Duff	3.22e-08	—	8.24e-03	8.24e-03	8.24e-03
CN	8.24e-03	8.24e-03	—	3.21e-08	1.75e-07
FittCN	8.24e-03	8.24e-03	3.21e-08	—	1.75e-07
VanLeer	8.24e-03	8.24e-03	1.75e-07	1.75e-07	—

Table 5.15: Error table for American Put Price, V, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.56e-07	1.21e-02	1.21e-02	1.21e-02
Duff	3.56e-07	—	1.21e-02	1.21e-02	1.21e-02
CN	1.21e-02	1.21e-02	—	3.31e-07	4.08e-06
FittCN	1.21e-02	1.21e-02	3.31e-07	—	4.08e-06
VanLeer	1.21e-02	1.21e-02	4.08e-06	4.08e-06	—

Table 5.16: Error table for American Put Delta, Δ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.01e-05	5.48e-01	5.48e-01	5.49e-01
Duff	3.01e-05	—	5.48e-01	5.48e-01	5.49e-01
CN	5.48e-01	5.48e-01	—	3.12e-05	2.03e-04
FittCN	5.48e-01	5.48e-01	3.12e-05	—	2.03e-04
VanLeer	5.49e-01	5.49e-01	2.03e-04	2.03e-04	—

Table 5.17: Error table for American Put Gamma, Γ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.20e-06	4.94e-01	4.94e-01	4.94e-01
Duff	1.20e-06	—	4.94e-01	4.94e-01	4.94e-01
CN	4.94e-01	4.94e-01	—	1.27e-06	8.71e-06
FittCN	4.94e-01	4.94e-01	1.27e-06	—	8.71e-06
VanLeer	4.94e-01	4.94e-01	8.71e-06	8.71e-06	—

Table 5.18: Error table for American Put Theta, Θ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.20e-16	8.37e-07	8.37e-07	8.37e-07
Duff	1.20e-16	—	8.37e-07	8.37e-07	8.37e-07
CN	8.37e-07	8.37e-07	—	1.22e-16	8.20e-15
FittCN	8.37e-07	8.37e-07	1.22e-16	—	7.95e-15
VanLeer	8.37e-07	8.37e-07	8.20e-15	7.95e-15	—

Table 5.19: Error table for American Put Price, V, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.63e-11	7.48e-04	7.48e-04	7.48e-04
Duff	1.63e-11	—	7.48e-04	7.48e-04	7.48e-04
CN	7.48e-04	7.48e-04	—	1.67e-11	9.10e-11
FittCN	7.48e-04	7.48e-04	1.67e-11	—	9.12e-11
VanLeer	7.48e-04	7.48e-04	9.10e-11	9.12e-11	—

Table 5.20: Error table for American Put Delta, Δ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.87e-10	5.84e-04	5.84e-04	5.84e-04
Duff	1.87e-10	—	5.84e-04	5.84e-04	5.84e-04
CN	5.84e-04	5.84e-04	—	1.92e-10	8.94e-10
FittCN	5.84e-04	5.84e-04	1.92e-10	—	8.98e-10
VanLeer	5.84e-04	5.84e-04	8.94e-10	8.98e-10	—

Table 5.21: Error table for American Put Gamma, Γ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.18e-11	3.43e-02	3.43e-02	3.43e-02
Duff	3.18e-11	—	3.43e-02	3.43e-02	3.43e-02
CN	3.75e-02	3.75e-02	—	2.92e-11	2.25e-10
FittCN	3.75e-02	3.75e-02	2.92e-11	—	2.25e-10
VanLeer	3.75e-02	3.75e-02	2.25e-10	2.25e-10	—

Table 5.22: Error table for American Put Theta, Θ, with $K = 40$, $T = 1$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	**Duff**	**CN**	**FittCN**	**VanLeer**
Imp	—	5.19e-08	4.09e-03	4.09e-03	4.09e-03
Duff	5.19e-08	—	4.09e-03	4.09e-03	4.09e-03
CN	4.09e-03	4.09e-03	—	5.01e-08	6.49e-07
FittCN	4.09e-03	4.09e-03	5.01e-08	—	6.24e-07
VanLeer	4.09e-03	4.09e-03	6.49e-07	6.24e-07	—

Table 5.23: Error table for American Put Price, V, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	**Duff**	**CN**	**FittCN**	**VanLeer**
Imp	—	3.34e-07	1.15e-02	1.15e-02	1.15e-02
Duff	3.34e-07	—	1.15e-02	1.15e-02	1.15e-02
CN	1.15e-02	1.15e-02	—	6.94e-07	1.50e-05
FittCN	1.15e-02	1.15e-02	6.94e-07	—	1.47e-05
VanLeer	1.15e-02	1.15e-02	1.50e-05	1.47e-05	—

Table 5.24: Error table for American Put Delta, Δ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	**Duff**	**CN**	**FittCN**	**VanLeer**
Imp	—	3.17e-05	5.34e-01	5.34e-01	5.35e-01
Duff	3.17e-05	—	5.34e-01	5.34e-01	5.35e-01
CN	5.34e-01	5.34e-01	—	5.23e-05	6.88e-04
FittCN	5.34e-01	5.34e-01	5.23e-05	—	6.72e-04
VanLeer	5.35e-01	5.35e-01	6.88e-04	6.72e-04	—

Table 5.25: Error table for American Put Gamma, Γ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	**Duff**	**CN**	**FittCN**	**VanLeer**
Imp	—	1.21e-06	2.44e-01	2.44e-01	2.44e-01
Duff	1.21e-06	—	2.44e-01	2.44e-01	2.44e-01
CN	2.44e-01	2.44e-01	—	2.85e-06	3.25e-05
FittCN	2.44e-01	2.44e-01	2.85e-06	—	3.12e-05
VanLeer	2.44e-01	2.44e-01	3.25e-05	3.12e-05	—

Table 5.26: Error table for American Put Theta, Θ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	8.01e-16	1.06e-07	1.06e-07	1.06e-07
Duff	8.01e-16	—	1.06e-07	1.06e-07	1.06e-07
CN	1.06e-07	1.06e-07	—	8.10e-16	6.20e-14
FittCN	1.06e-07	1.06e-07	8.10e-16	—	5.26e-14
VanLeer	1.06e-07	1.06e-07	6.20e-14	5.26e-14	—

Table 5.27: Error table for American Put Price, V, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.73e-11	1.12e-04	1.12e-04	1.12e-04
Duff	1.73e-11	—	1.12e-04	1.12e-04	1.12e-04
CN	1.12e-04	1.12e-04	—	1.75e-11	1.28e-10
FittCN	1.12e-04	1.12e-04	1.75e-11	—	1.30e-10
VanLeer	1.12e-04	1.12e-04	1.28e-10	1.30e-10	—

Table 5.28: Error table for American Put Delta, Δ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	9.29e-11	1.11e-04	1.11e-04	1.11e-04
Duff	9.29e-11	—	1.11e-04	1.11e-04	1.11e-04
CN	1.11e-04	1.11e-04	—	9.48e-11	3.52e-10
FittCN	1.11e-04	1.11e-04	9.48e-11	—	3.60e-10
VanLeer	1.11e-04	1.11e-04	3.52e-10	3.60e-10	—

Table 5.29: Error table for American Put Gamma, Γ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.19e-09	4.52e-02	4.52e-02	4.52e-02
Duff	1.19e-09	—	4.52e-02	4.52e-02	4.52e-02
CN	5.05e-02	5.05e-02	—	1.09e-09	1.19e-08
FittCN	5.05e-02	5.05e-02	1.09e-09	—	1.21e-08
VanLeer	5.05e-02	5.05e-02	1.19e-08	1.21e-08	—

Table 5.30: Error table for American Put Theta, Θ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.2$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.22e-08	8.24e-03	8.24e-03	8.24e-03
Duff	3.22e-08	—	8.24e-03	8.24e-03	8.24e-03
CN	8.24e-03	8.24e-03	—	3.21e-08	1.75e-07
FittCN	8.24e-03	8.24e-03	3.21e-08	—	1.75e-07
VanLeer	8.24e-03	8.24e-03	1.75e-07	1.75e-07	—

Table 5.31: Error table for American Put Price, V, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.56e-07	1.21e-02	1.21e-02	1.21e-02
Duff	3.56e-07	—	1.21e-02	1.21e-02	1.21e-02
CN	1.21e-02	1.21e-02	—	3.31e-07	4.08e-06
FittCN	1.21e-02	1.21e-02	3.31e-07	—	4.08e-06
VanLeer	1.21e-02	1.21e-02	4.08e-06	4.08e-06	—

Table 5.32: Error table for American Put Delta, Δ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	3.01e-05	5.48e-01	5.48e-01	5.49e-01
Duff	3.01e-05	—	5.48e-01	5.48e-01	5.49e-01
CN	5.48e-01	5.48e-01	—	3.12e-05	2.03e-04
FittCN	5.48e-01	5.48e-01	3.12e-05	—	2.03e-04
VanLeer	5.49e-01	5.49e-01	2.03e-04	2.03e-04	—

Table 5.33: Error table for American Put Gamma, Γ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.20e-06	4.94e-01	4.94e-01	4.94e-01
Duff	1.20e-06	—	4.94e-01	4.94e-01	4.94e-01
CN	4.94e-01	4.94e-01	—	1.27e-06	8.71e-06
FittCN	4.94e-01	4.94e-01	1.27e-06	—	8.71e-06
VanLeer	4.94e-01	4.94e-01	8.71e-06	8.71e-06	—

Table 5.34: Error table for American Put Theta, Θ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Avg method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	2.28e-16	7.25e-07	7.25e-07	7.25e-07
Duff	2.28e-16	—	7.25e-07	7.25e-07	7.25e-07
CN	7.25e-07	7.25e-07	—	2.29e-16	1.22e-14
FittCN	7.25e-07	7.25e-07	2.29e-16	—	1.15e-14
VanLeer	7.25e-07	7.25e-07	1.22e-14	1.15e-14	—

Table 5.35: Error table for American Put Price, V, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	1.84e-11	3.03e-04	3.03e-04	3.03e-04
Duff	1.84e-11	—	3.03e-04	3.03e-04	3.03e-04
CN	3.02e-04	3.02e-04	—	1.85e-11	8.02e-11
FittCN	3.02e-04	3.02e-04	1.85e-11	—	7.97e-11
VanLeer	3.03e-04	3.03e-04	8.02e-11	7.97e-11	—

Table 5.36: Error table for American Put Delta, Δ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	9.33e-11	1.52e-04	1.52e-04	1.52e-04
Duff	9.33e-11	—	1.52e-04	1.52e-04	1.52e-04
CN	1.52e-04	1.52e-04	—	9.37e-11	3.68e-10
FittCN	1.52e-04	1.52e-04	9.37e-11	—	3.67e-10
VanLeer	1.52e-04	1.52e-04	3.68e-10	3.67e-10	—

Table 5.37: Error table for American Put Gamma, Γ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

	Imp	Duff	CN	FittCN	VanLeer
Imp	—	9.31e-11	2.80e-02	2.80e-02	2.80e-02
Duff	9.31e-11	—	2.80e-02	2.80e-02	2.80e-02
CN	3.02e-02	3.02e-02	—	8.61e-11	4.35e-10
FittCN	3.02e-02	3.02e-02	8.61e-11	—	4.33e-10
VanLeer	3.02e-02	3.02e-02	4.35e-10	4.33e-10	—

Table 5.38: Error table for American Put Theta, Θ, with $K = 40$, $T = 2$, $r = 6\%$, $\sigma = 0.4$ using 50 timesteps per year ($\delta t = 0.02$) and the Max method

Single Scheme Analysis

Another common numerical analysis method to estimate the errors in these schemes is to compare solutions of a scheme using different mesh sizes. Since one solution has more datapoints than the other, interpolation is used to enable comparisons at the missing points. It is usual to use the Max Norm for this analysis.

Our American option solutions were compared using a fixed timestep of $\delta t = 0.02$ (or 50 timesteps per year), and 10,000 meshpoints in the S-direction as our benchmark solution.

We also calculated the solutions using 5000, 2500 and 1250 meshpoints in the S-direction and compared them to our benchmark solution of 10000 meshpoints in the error analysis.

The results are shown in Tables 5.39 - 5.46. As would be expected from convergent discretisation schemes, the errors get smaller as the step size decreases. Again, the errors are smallest for the option price, V, and largest for the Gamma, Γ.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	5.88e-03	2.71e-03	1.39e-03	3.69e-03	1.71e-03	9.60e-04
Duff	5.89e-03	2.72e-03	1.39e-03	3.69e-03	1.71e-03	9.60e-04
CN	3.23e-02	1.41e-02	4.97e-03	3.17e-02	1.41e-02	4.62e-03
FittCN	3.23e-02	1.41e-02	4.97e-03	3.17e-02	1.41e-02	4.62e-03
VanLeer	3.18e-02	1.40e-02	4.95e-03	3.16e-02	1.40e-02	4.61e-03

Table 5.39: Error table for American Put Price, V, with $K = 40$, $r = 6\%$ and $T = 1$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	6.22e-03	2.89e-03	1.43e-03	4.40e-03	2.04e-03	9.95e-04
Duff	6.23e-03	2.89e-03	1.43e-03	4.40e-03	2.04e-03	9.96e-04
CN	3.23e-02	1.41e-02	4.97e-03	3.17e-02	1.41e-02	4.86e-03
FittCN	3.23e-02	1.41e-02	4.97e-03	3.17e-02	1.41e-02	4.86e-03
VanLeer	3.18e-02	1.40e-02	4.95e-03	3.16e-02	1.40e-02	4.86e-03

Table 5.40: Error table for American Put Price, V, with $K = 40$, $r = 6\%$ and $T = 2$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	3.76e-02	1.75e-01	1.83e-02	2.41e-02	1.75e-01	1.21e-02
Duff	3.77e-02	1.75e-01	1.83e-02	2.41e-02	1.75e-01	1.21e-02
CN	4.18e-01	3.64e-01	2.45e-01	4.29e-01	3.69e-01	2.47e-01
FittCN	4.18e-01	3.64e-01	2.45e-01	4.29e-01	3.69e-01	2.47e-01
VanLeer	4.16e-01	3.63e-01	2.44e-01	4.28e-01	3.69e-01	2.47e-01

Table 5.41: Error table for American Put Delta, Δ, with $K = 40$, $r = 6\%$ and $T = 1$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	3.90e-02	1.75e-01	1.89e-02	2.66e-02	1.75e-01	1.37e-02
Duff	3.90e-02	1.75e-01	1.89e-02	2.66e-02	1.75e-01	1.37e-02
CN	4.18e-01	3.64e-01	2.45e-01	4.29e-01	3.69e-01	2.47e-01
FittCN	4.18e-01	3.64e-01	2.45e-01	4.29e-01	3.69e-01	2.47e-01
VanLeer	4.16e-01	3.63e-01	2.44e-01	4.28e-01	3.69e-01	2.47e-01

Table 5.42: Error table for American Put Delta, Δ, with $K = 40$, $r = 6\%$, and $T = 2$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	1.60e+00	2.19e+00	1.30e+00	1.05e+00	2.19e+00	8.84e-01
Duff	1.60e+00	2.19e+00	1.30e+00	1.05e+00	2.19e+00	8.84e-01
CN	1.97e+01	1.72e+01	1.17e+01	1.99e+01	1.72e+01	1.17e+01
FittCN	1.97e+01	1.72e+01	1.17e+01	1.99e+01	1.72e+01	1.17e+01
VanLeer	1.97e+01	1.72e+01	1.17e+01	1.99e+01	1.72e+01	1.17e+01

Table 5.43: Error table for American Put Gamma, Γ, with $K = 40$, $r = 6\%$, and $T = 1$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	1.62e+00	2.19e+00	1.31e+00	1.17e+00	2.19e+00	9.07e-01
Duff	1.62e+00	2.19e+00	1.31e+00	1.17e+00	2.19e+00	9.07e-01
CN	1.97e+01	1.72e+01	1.17e+01	1.99e+01	1.72e+01	1.17e+01
FittCN	1.97e+01	1.72e+01	1.17e+01	1.99e+01	1.72e+01	1.17e+01
VanLeer	1.97e+01	1.72e+01	1.17e+01	1.99e+01	1.72e+01	1.17e+01

Table 5.44: Error table for American Put Gamma, Γ, with $K = 40$, $r = 6\%$, and $T = 2$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	1.38e+00	5.72e-01	1.87e-01	1.32e+00	5.56e-01	1.84e-01
Duff	1.38e+00	5.72e-01	1.87e-01	1.32e+00	5.56e-01	1.84e-01
CN	3.15e+00	1.40e+00	4.96e-01	3.15e+00	1.40e+00	4.62e-01
FittCN	3.15e+00	1.40e+00	4.96e-01	3.15e+00	1.40e+00	4.62e-01
VanLeer	3.11e+00	1.40e+00	4.94e-01	3.14e+00	1.40e+00	4.61e-01

Table 5.45: Error table for American Put Theta, Θ, with $K = 40$, $r = 6\%$, and $T = 1$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

	$\sigma = 0.2$			$\sigma = 0.4$		
	1250	2500	5000	1250	2500	5000
Imp	1.38e+00	5.72e-01	1.87e-01	1.32e+00	5.56e-01	1.84e-01
Duff	1.38e+00	5.72e-01	1.87e-01	1.32e+00	5.56e-01	1.84e-01
CN	3.15e+00	1.40e+00	4.96e-01	3.15e+00	1.40e+00	4.86e-01
FittCN	3.15e+00	1.40e+00	4.96e-01	3.15e+00	1.40e+00	4.86e-01
VanLeer	3.11e+00	1.40e+00	4.94e-01	3.14e+00	1.40e+00	4.86e-01

Table 5.46: Error table for American Put Theta, Θ, with $K = 40$, $r = 6\%$, and $T = 2$. The other parameters are as shown in the table headings. The stock stepsize is $10/N$, where N is the size of the mesh.

The error scales well with step-size, yielding a roughly linear relationship between the log of the error and the log of the step-size δS (see Figure 5-16).

As before, the schemes have split into two groups: the Implicit and Duffy schemes in one group and the CN, the Fitted CN and the Van Leer flux-limiting scheme in the other.

The slope of the two groups appear to be roughly equal, and this is surprising, since the CN, Fitted CN and Van Leer are higher order schemes than the Implicit and Duffy schemes.

Also, as in the case for the European options, the error scaling for the Greeks are not as well-behaved as for the option price. Again, this is not surprising, and is to be expected.

Once again, the graphs form into two groups, one for the CN, Fitted CN and Van Leer, and one for the Implicit and Duffy schemes.

The first groups (with the $O(h^2)$ schemes) behave remarkably well for the Greeks. The scaling line is almost linear, and bears a similar resemblance to the scaling of the prices.

Please note again that the lines are a simple joining of the datapoints, there was no regression line or fitting done on the datapoints.

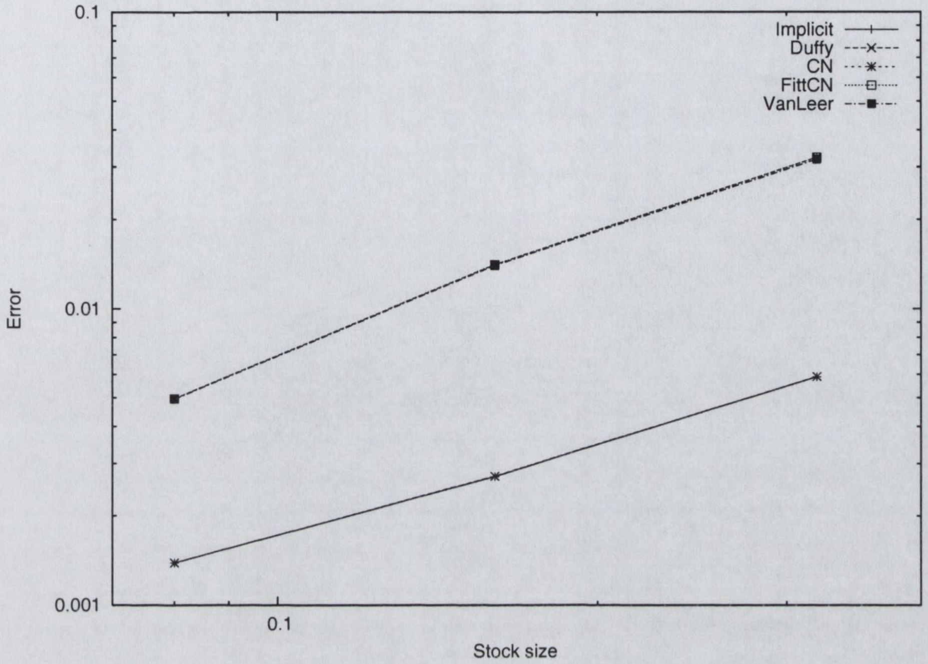

Figure 5-16: Scaling Analysis of an American Put option Price, V, with $K = 40$, $T = 1$, $\sigma = 0.2$, $r = 6\%$, 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

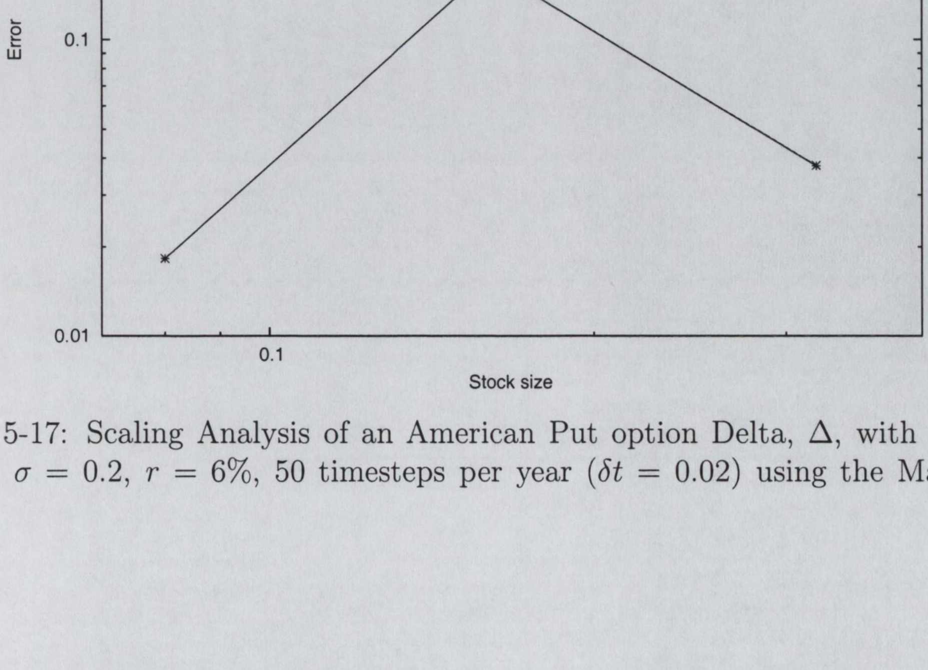

Figure 5-17: Scaling Analysis of an American Put option Delta, Δ, with $K = 40$, $T = 1$, $\sigma = 0.2$, $r = 6\%$, 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

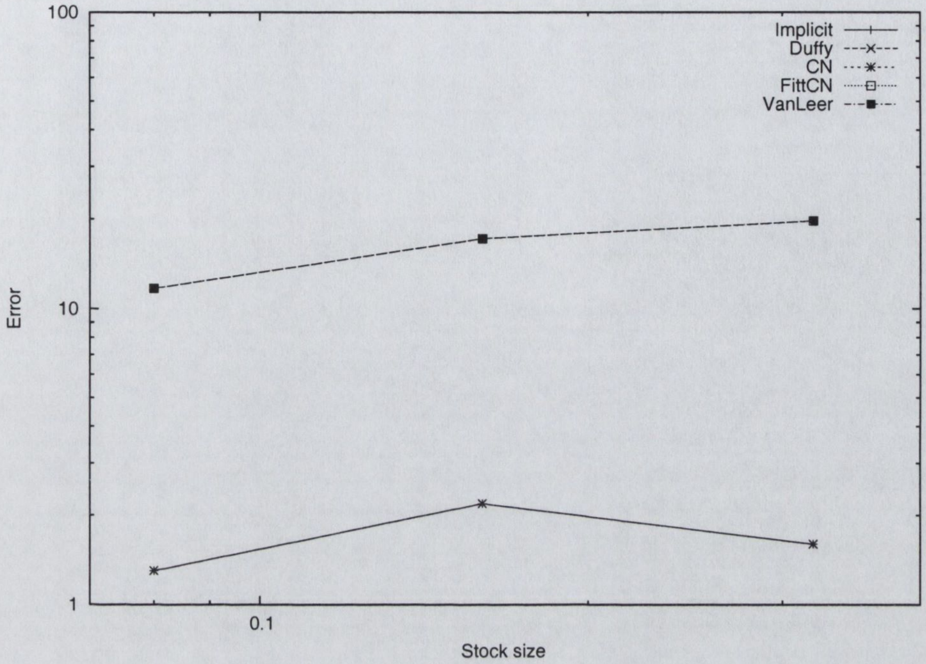

Figure 5-18: Scaling Analysis of an American Put option Gamma, Γ, with $K = 40$, $T = 1$, $\sigma = 0.2$, $r = 6\%$, 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

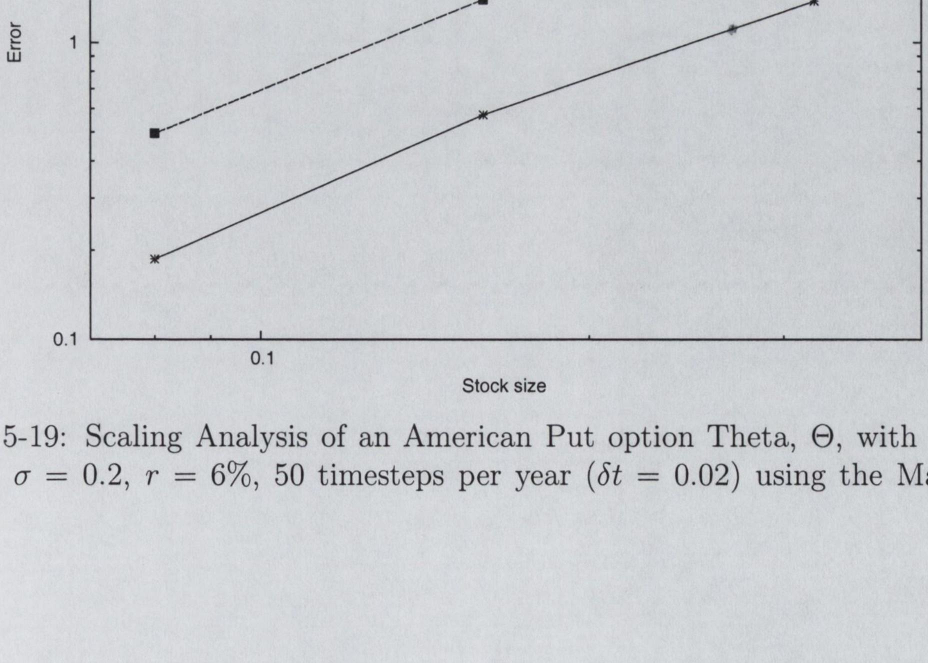

Figure 5-19: Scaling Analysis of an American Put option Theta, Θ, with $K = 40$, $T = 1$, $\sigma = 0.2$, $r = 6\%$, 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

Analysis of Results for American Options with Extreme Parameters

The fitted schemes were designed primarily for situations where the second-order equation becomes singly-perturbed, i.e. the coefficient of diffusion becomes very small in comparison to the other terms.

A similar comparison of the schemes for the extreme values of r, S and σ that were used for the European options was performed. The parameters used were as follows:

$$\begin{aligned} T &= 1, \\ K &= 1, \\ S_{\max} &= 10, \\ \sigma &= 0.01, \\ r &= 15\%. \end{aligned}$$

The error analysis data are shown in Tables 5.47 – 5.50.

	1250	2500	5000
Imp	6.68e-04	3.18e-04	1.12e-04
Duff	6.29e-04	2.70e-04	9.00e-05
CN	6.27e-04	2.87e-04	1.08e-04
FittCN	6.29e-04	2.70e-04	9.00e-05
VanLeer	6.30e-04	2.71e-04	9.08e-05

Table 5.47: Error table for American Put option Price, V, with $K = 40$, $T = 1$, $r = 15\%$, $\sigma = 0.01$, and 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

	1250	2500	5000
Imp	4.47e-01	4.10e-01	2.48e-01
Duff	4.36e-01	3.94e-01	2.50e-01
CN	4.34e-01	3.99e-01	2.48e-01
FittCN	4.36e-01	3.94e-01	2.50e-01
VanLeer	4.37e-01	3.95e-01	2.50e-01

Table 5.48: Error table for American Put option Delta, Δ, with $K = 40$, $T = 1$, $r = 15\%$, $\sigma = 0.01$, and 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

	1250	2500	5000
Imp	7.97e+02	6.99e+02	4.45e+02
Duff	7.99e+02	7.02e+02	4.52e+02
CN	7.98e+02	6.99e+02	4.44e+02
FittCN	8.00e+02	7.03e+02	4.53e+02
VanLeer	8.19e+02	7.21e+02	4.68e+02

Table 5.49: Error table for American Put option Gamma, Γ, with $K = 40$, $T = 1$, $r = 15\%$, $\sigma = 0.01$, and 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

	1250	2500	5000
Imp	2.59e-03	1.00e-02	5.58e-03
Duff	2.78e-03	1.00e-02	5.00e-03
CN	2.91e-03	1.00e-02	5.62e-03
FittCN	2.78e-03	1.00e-02	5.00e-03
VanLeer	2.74e-03	1.00e-02	5.00e-03

Table 5.50: Error table for American Put option Theta, Θ, with $K = 40$, $T = 1$, $r = 15\%$, $\sigma = 0.01$, and 50 timesteps per year ($\delta t = 0.02$) using the Max Norm metric

A graph of cross-section of the option price and the Greeks at a single timestep are shown in Figs 5-20 – 5-27.

The Crank-Nicolson and Implicit schemes exhibit the numerical oscillations in the American option solution also, just like for the European option solution. This is shown in Figs. 5-20 - 5-23. Once again, the Greeks oscillate signficantly around the true value.

This effect disappears with the use of a finer mesh, and this is shown in Figs. 5-24 - 5-27. In these plots, the oscillations have been completely eliminated, and all the schemes closely agree with one another.

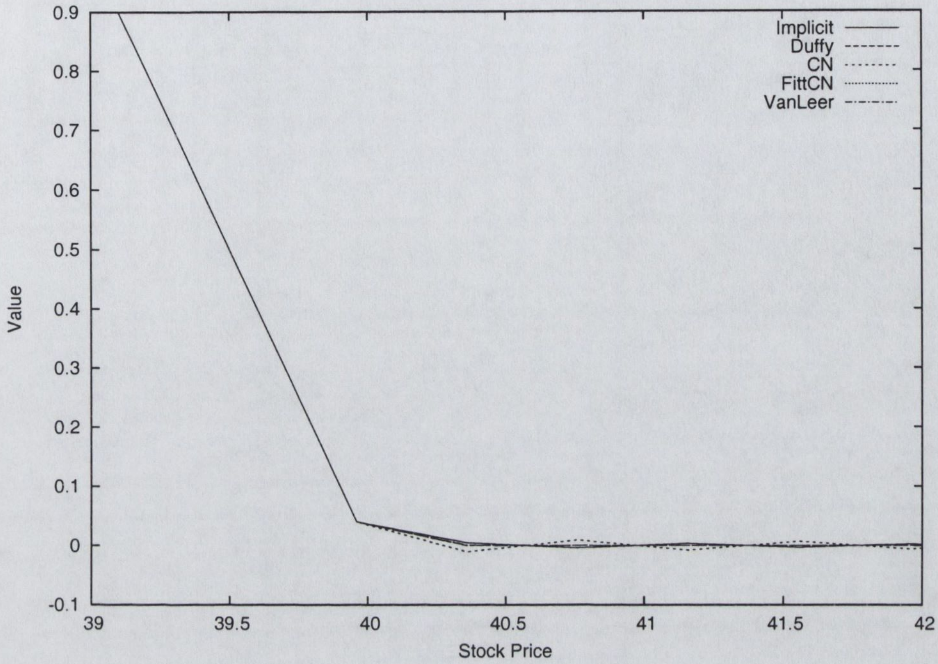

Figure 5-20: Plots Around the Money for American Put option Price, V, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 1000 point mesh ($\delta S = 0.4$)

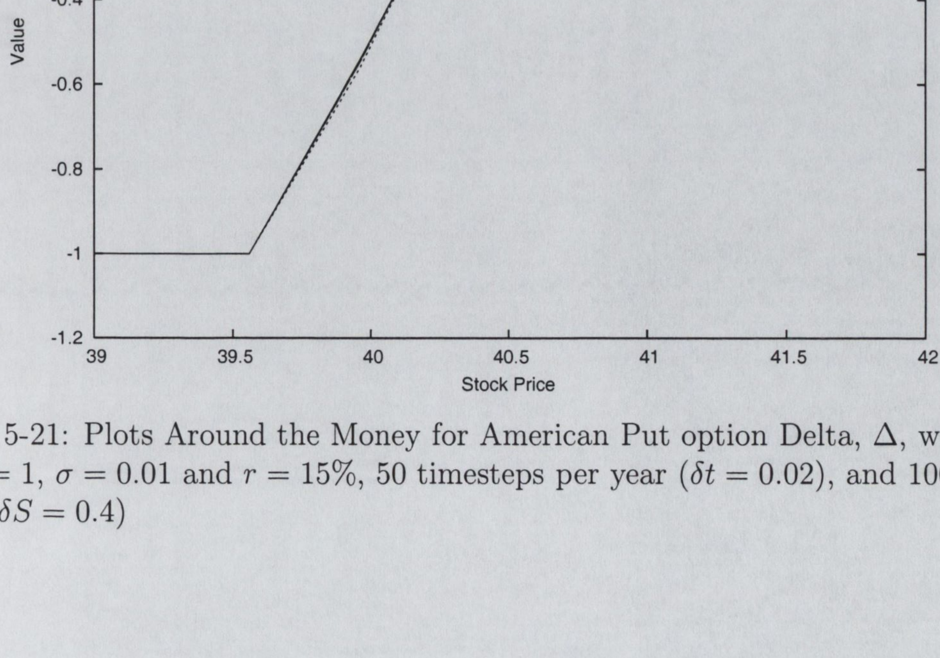

Figure 5-21: Plots Around the Money for American Put option Delta, Δ, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 1000 point mesh ($\delta S = 0.4$)

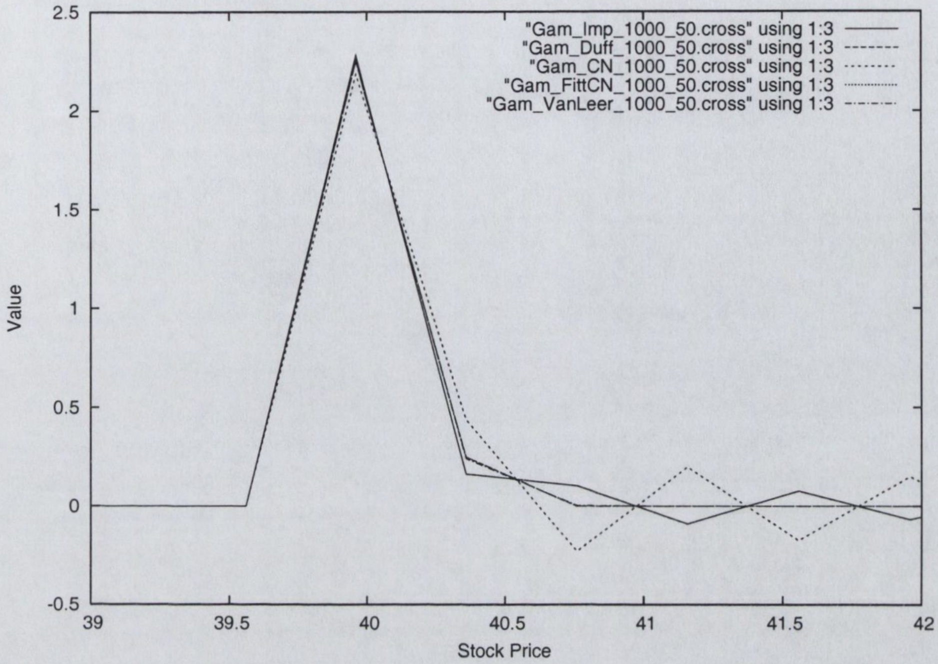

Figure 5-22: Plots Around the Money for American Put option Gamma, Γ, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 1000 point mesh ($\delta S = 0.4$)

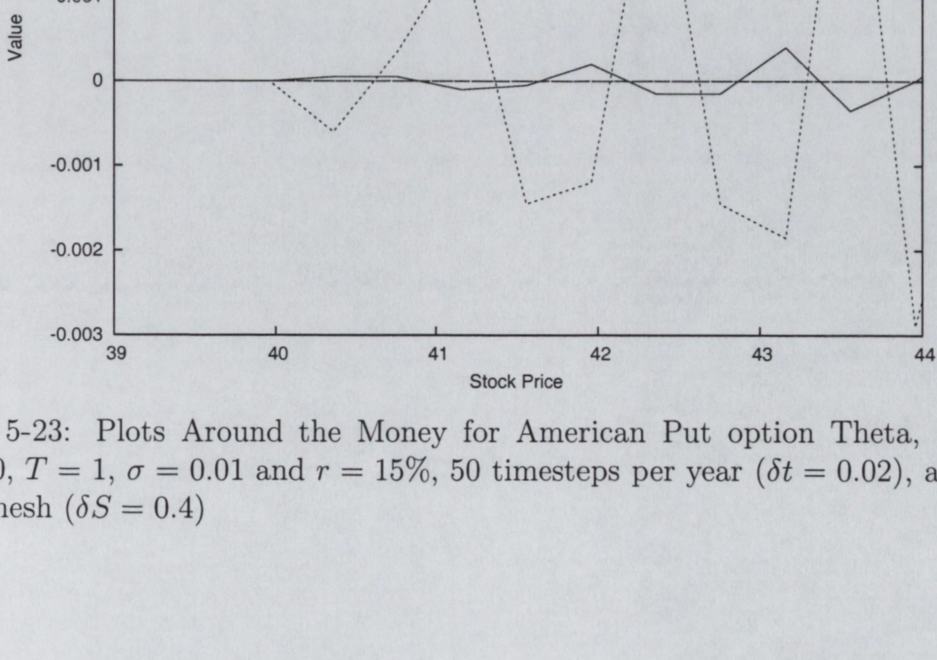

Figure 5-23: Plots Around the Money for American Put option Theta, Θ, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 1000 point mesh ($\delta S = 0.4$)

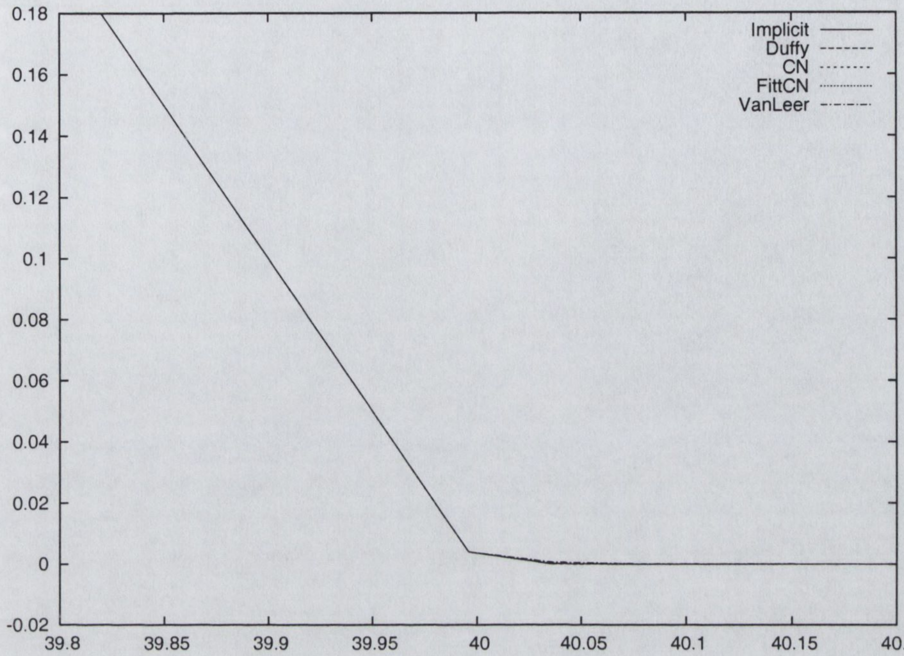

Figure 5-24: Plots Around the Money for American Put option Price, V, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 10000 point mesh ($\delta S = 0.04$)

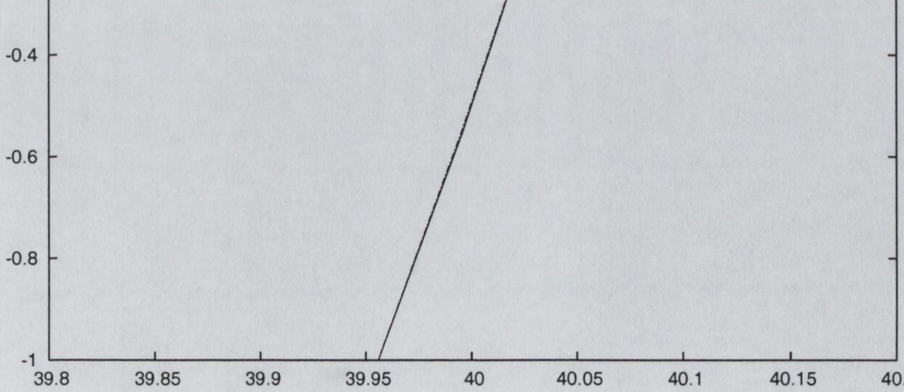

Figure 5-25: Plots Around the Money for American Put option Delta, Δ, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 10000 point mesh ($\delta S = 0.04$)

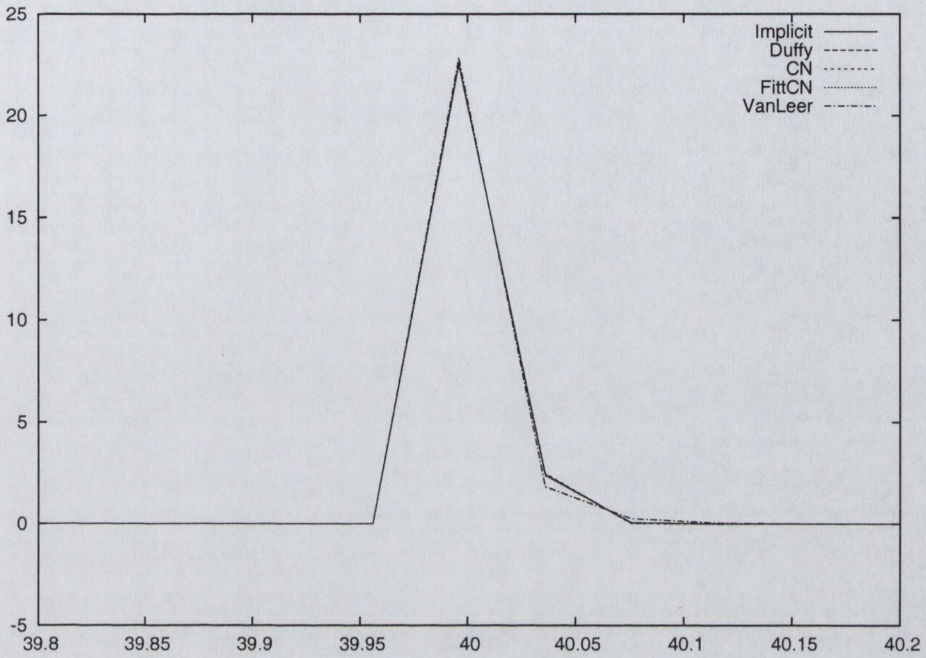

Figure 5-26: Plots Around the Money for American Put option Gamma, Γ, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 10000 point mesh ($\delta S = 0.04$)

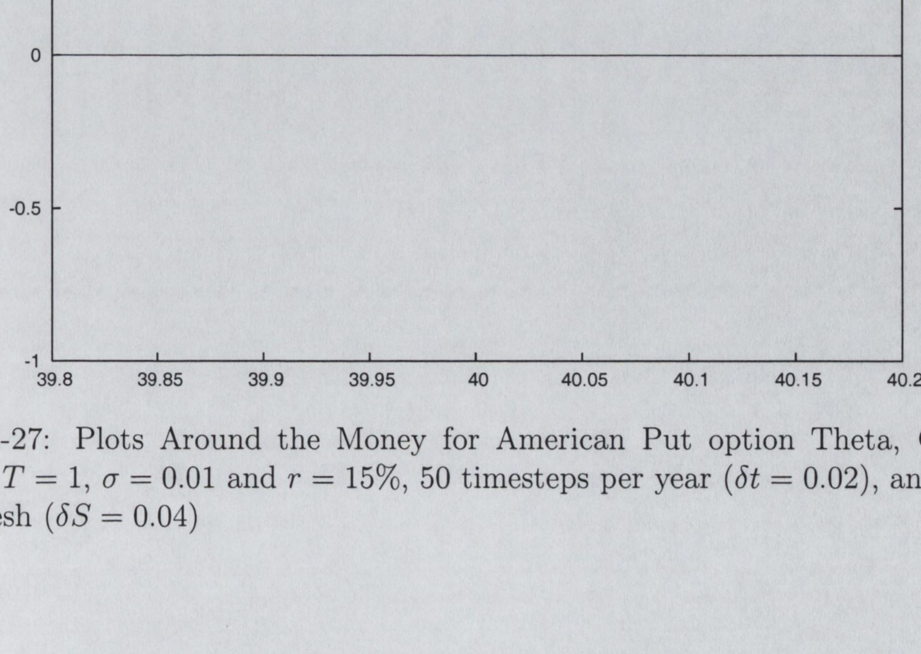

Figure 5-27: Plots Around the Money for American Put option Theta, Θ, with $K = 40$, $T = 1$, $\sigma = 0.01$ and $r = 15\%$, 50 timesteps per year ($\delta t = 0.02$), and 10000 point mesh ($\delta S = 0.04$)

5.6.2 Summary

Overall, the scheme reasonably well. The errors scaled with the step-size as would be expected for their order of convergence, yielding lines that were approximately linear for $\ln \delta S$ vs ln Error.

Overall, under the conditions of high interest rate and low volatility, there is definite room for improvement in terms of calculation of the Greeks, and perhaps other methods should also be considered in this case (such as the Penalty method, used by Forsyth[14]).

Chapter 6

The Stochastic Approach

This chapter will define various stochastic pricing methods and utilise these methods to price American options.

6.1 Monte Carlo Methods

Monte Carlo methods are a widely used class of computational algorithms for simulating the behaviour of various physical and mathematical systems. They are distinguished from other simulation methods (such as molecular dynamics) by being **stochastic**, i.e. non-deterministic in some manner - as opposed to deterministic algorithms.

Monte Carlo methods are especially useful in studying systems with a large number of coupled degrees of freedom, such as liquids, disordered materials, and strongly coupled solids. More broadly, Monte Carlo methods are useful for modeling phenomena with significant uncertainty in inputs, such as the calculation of risk in business (so called **Value at Risk** or VaR). A classic use is for the evaluation of definite integrals, particularly multidimensional integrals with complicated boundary conditions.

Use of Monte Carlo methods require large numbers of random numbers, and it was their use that spurred the development of psuedo-random number generators, which were far quicker to use than the tables of random numbers previously used for statistical sampling.

6.2 Monte Carlo Methods and the Black-Scholes Model

The value of an option is defined to be the present value of the expected payoff of the option at expiry under a **risk-neutral** random walk for the underlying.

This definition for the option price lends itself perfectly to the Monte Carlo technique, and the resulting algorithm follows intuitively:

Algorithm 6-1 The Monte Carlo Method

1 – Simulate the risk-neutral random-walk, starting at the initial value of the asset S_0, over the whole life-time of the option. This gives a single realisation of the underlying price path.

2 – For this realisation, calculate the option payoff.

3 – Perform N such realisations (where N is a large number).

4 – Calculate the average payoff over all realisations.

5 – Take the present value of this average payoff.

One of the assumptions of the Black-Scholes differential equation is that the underlying asset follows a lognormal random walk. Thus, the random walk for S is given by:

$$dS = rS\,dt + \sigma S\,dX. \tag{6.1}$$

The option price is therefore defined to be:

$$V = e^{-r(T-t)}\,E[H(S)]. \tag{6.2}$$

where $H(S)$ is the payoff function of the option, as this is just the discounted value of the expected payoff.

To simulate the random-walk of the underlying we need to decide how exactly to update S at every time-step. This can be derived from the random walk (Eqn 6.1):

$$\delta S = rS\,\delta t + \sigma S\sqrt{\delta t}\,\phi, \tag{6.3}$$

where ϕ is a random number drawn from a standardised Normal distribution.

In most circumstances, this update sequence would be simulated in a discrete way, using some approximate method to calculate δS, and using this new value to update the value of S.

For lognormal random walks it is possible to integrate the stochastic differential equation exactly, producing an exact time-stepping algorithm. This means that there will be no error associated with the number of time-steps used in each simulation.

Once the integration has been done, our time-stepping algorithm becomes

$$S(t+\delta t) = S(t)\ \exp\left((r-\tfrac{1}{2}\sigma^2)\,\delta t + \sigma\sqrt{\delta t}\ \phi\right). \tag{6.4}$$

Thus, all that is required to make good use of the Monte Carlo technique is the capability to produce a large number of normally distributed, independent random variables.

Note that the accuracy of the algorithm is dependent only on the number of simulations performed, N, but this accuracy is of $O(\sqrt{N})$, so that to get a ten-fold accuracy increase it is necessary to multiply the number of simulations by a factor of 100.

It is also important to remember that the payoff can never be below zero (since the option will not be exercised). Should the final price be such that the payoff would be negative, then the option is not exercised and the payoff is zero.

Monte Carlo methods have the big advantage that they are very powerful for solving problems involving more than one asset, and so are very useful in portfolio calculations. Once more than one asset needs to be modeled, other methods such as finite difference become prohibitively expensive to run, as the problem becomes multi-dimensional.

6.3 Monte Carlo Methods and American Options

American options are problematic for Monte Carlo methods due to the difficulty of incorporating the early exercise aspect into the technique. The optimal exercise boundary cannot be calculated easily and so it is very difficult to know whether it is optimal to exercise early or not.

There are a number of methods for incorporating this early-exercise into the Monte Carlo algorithms and we shall discuss the different techniques separately.

6.3.1 Simple Maximal Calculation

The value of an American option can be written as

$$V_{\text{Amer}} = \max_{\tau<T} \left(E[H_{t=\tau}(S_\tau)] \right) . \tag{6.5}$$

which implies a very simple algorithm for calculating the value of an option (Algorithm 6-2).

Algorithm 6-2 The Simple Maximal Calculation

1 – Generate the required number of paths for the option price.

2 – Calculate the discounted payoff at each time-step along each generated path.

3 – Average these discounted payoffs at each time-step.

4 – Take the maximum discounted payoff as the option value.

Although this method has the benefit of simplicity, it has the disadvantage that it gives an estimate of the **perfect foresight solution** and so has a tendency to over-estimate the value of the option.

6.3.2 Regression Estimation of the Expected Payoff

Another method for simulating the early exercise is the "Least-Squares" method, which uses a least squares regression technique to estimate the expected payoff at each time-step.

It was first developed by Longstaff and Schwartz [24] and is applicable to a variety of early exercise instruments.

An expected payoff is calculated at each time-step based on the asset path and is compared to the current instantaneous payoff. This data is then used to decide whether or not it is optimal to exercise early.

The details of this technique require that each time-step is examined backwards, starting at the final time-step and moving backwards towards the initial date of the option. This introduces the obvious disadvantage that it is much more computationally expensive to produce a value, and so takes longer to calculate.

Algorithm 6-3 Regression Estimation of the Expected Payoff

1 – Generate the required number of paths for the option price, N say, each containing M time-steps per path.

2 – For each path, do the following:

   a) Inspect the payoffs at time $t = \tau_{M-1}$, and only consider those paths with non-zero payoffs.
   b) Calculate the payoffs on each path at time $t = \tau_M = T$, and discount them by one time-step to time $t = \tau_{M-1}$.
   c) Using the asset prices at $t = \tau_{M-1}$ as one variable, and the discounted payoffs as the other, calculate least squares regression coefficients.
   d) Use these regression coefficients to calculate an expected payoff at time $t = \tau_{M-1}$ based on the instantaneous payoff.
   e) If the current payoff is larger than the expected payoff, then exercise the option at this point.

3 – Repeat this process on successive time-steps, moving backwards each time, revising for each path the time when exercise occurs.

4 – When this process is complete, each path should have a payoff and a time for when that payoff occurs.

5 – The option value is calculated from taking the average discounted payoff at time $t = 0$.

The above algorithm probably appears to be quite complex, but its working should be made clear through a reading of the example given by the authors of the method.

6.4 Monte Carlo Methods using Optimal Exercise Estimation

The methods of the previous section have proved to be very successful in solving American options using some sort of Monte Carlo method for generating the random walk of the underlying asset. Unfortunately, they have a few drawbacks in that they require a large amount of memory storage for their calculation or need to generate all their asset paths first, then work backwards to calculate the optimum exercise point.

6.4.1 The Brute Force Calculation Method

This method is very straightforward but is totally impractical for calculating the price of an option, due to its hugely inefficient performance, which is $O(M^N)$, where M is the number of iterations performed and N is the number of discrete time-steps per iteration.

The idea is simple. At any point in time, if the immediate payoff obtainable by exercising the option is greater than the discounted expected payoff obtainable by holding the option, then the option should be exercised immediately.

Since the problem is to be solved numerically, we have discretised the time dimension, and so we need to make the decision on early exercise at every time-step.

The exercise price is immediately obtainable from the payoff function of the option, so we need to calculate the discounted expected payoff of the option.

Since the fair price for an option is just the discounted expected payoff of the option over the lifetime of the option, the problem of calculating the discounted expected payoff is simply that of recalculating the option price.

This argument immediately seems circular, as we are trying to calculate this value in the process. This is not the case, as the decision is occuring at some time $t > 0$, and so we are calculating the price of option with $T_{\text{new}} < T$ and the asset price for this 'virtual' option is S_{current}, which is, in all probability, different from S. Note however, that the size of the time discretization is remaining the same, so the number of discrete time-steps in this virtual calculation is getting smaller.

Calculating the price of this new, virtual option repeats this process, which will in turn spawn new virtual price calculations, until we are left with a calculation which only involves a single time-step. With a single time-step, the expected payoff is readily calculable since it is just the mean payoff after a single time-step, discounted by the time step-size, δt.

Having found the option price for one time-step, we can then work backwards, filling in the just-calculated option prices. Note that this backward substitution will occur naturally through the recursive nature of the algorithm.

Algorithm 6-4 The Brute Force Calculation

1 – For each iteration, initialise $S = S_0$, $t = 0$.

2 – For each time-step:

a) Advance a single time-step $t \rightarrow t + \delta t$.

b) Update $S \rightarrow S + dS$, where

$$dS = rS\,dt + \sigma S\,dX$$

where dX is a random variable drawn from a standardised normal distribution.

c) Calculate P_{exercise}, the value of option if it were exercised immediately.

d) Calculate P_{expected}, the value of the option with the same settings as the current value but with $T \rightarrow T - t$ and $N \rightarrow N - n$, where $n = t/\delta t$, the number of time-steps taken since $t = 0$. Essentially, this is a recursive step, with the recursion terminating when the iteration contains only a single time-step, as the option must be exercised at this time-step.

e) Compare P_{exercise} to P_{expected}. If P_{exercise} is larger, then the option is exercised immediately, otherwise move onto the next time-step.

3 – Discount the resulting payoff using the time at which the option was exercised. For early exercise scenarios, this time will be less than the contract duration, T.

4 – Record this result, and start a new iteration.

5 – Average all the results over each iteration to find the option price.

A Brief Worked Example

We will illustrate the Brute Force algorithm by way of a simple example.

Suppose we are calculating the price of an American put with $K = \$40$. We have a starting asset price of \$30, and we want to run two timesteps. With $T = 1$ we have $\delta t = 0.5$. At each timestep, we are going to run two simulations.

So, we start at $S = \$30$. Suppose our stochastic algorithm gives us a new value $S = \$40$. We now need to know what the value of the expected payoff from this point is.

Thus, we start a new calculation with $S = \$40$, $T = 0.5$, and 1 timestep in the simulation. Our first simulation step gives us $S = \$50$, yielding a payoff of

$$\text{Max}(K - S, 0) = \text{Max}(\$40 - \$50, \$0) = \$0.$$

Our second simulation step gives us $S = \$30$, so the payoff is

$$\text{Max}(K - S, 0) = \text{Max}(\$40 - \$30, \$0) = \$10.$$

Thus, the average payoff is \$5. This payoff needs to be discounted by a timestep. Suppose the interest rate is such that the discounted value is \$4. Thus, our expected payoff value is \$4.

Our original asset walk gave us a value of $S = \$40$ after one timestep. The instantaneous payoff for this is

$$\text{Max}(K - S, 0) = \text{Max}(\$40 - \$40, \$0) = \$0.$$

Thus, should we exercise now our payoff is zero.

Since the instantantaneous payoff is lower than the expected payoff, it is not optimal to exercise at this point, and so we continue on to the next timestep.

6.4.2 Expected Payoff Interpolation

A second method for estimating the option price is derived directly from the previous method.

The brute force method is so appallingly inefficient due to the fact that it recalculates the expected payoffs each time a value is required.

This immediately suggests an optimisation. If possible, calculate some of these expectations in advance and then use an interpolation method to calculate the expected payoffs when we need them in the simulation.

So, the algorithm is now split into two stages: first we need to generate a list of values from which we can perform a payoff interpolation. Once that is completed, we perform a similar calculation algorithm to that of the Brute Force method, the primary distinction between the two being the fact that the expected payoff values are determined using an interpolation from the data calculated in the first stage.

A number of benefits of this new algorithm are immediately apparent. The immediate benefit is that there is no recursion in this method and so the performance of this algorithm is $O(MNK)$, where K is the number of asset price iterations over M interpolation points at each of the N time-steps, an immense improvement over the brute force method.

Secondly, in the event of a number of different calculations being required, we can simply re-use the previously calculated interpolation points – allowing us to skip the first stage completely. This results in a computation performance of $O(NM)$, yet another improvement. It is important to note that this can only be done in the case where the strike price, interest rate, and volatility of the underlying all remain the same.

Thus, interpolation data for different scenarios can be pre-generated and stored for future use, which should lead to significant time-cost savings (though the trade-off is that all the interpolation data needs to be stored somewhere, so that the time cost is traded for a storage cost).

A Brief Worked Example - Stage One

Again, we will illustrate the calculation of the interpolation mesh by way of a simple example. Again, we have an American put option with $T = 1$, $K = \$40$, our starting asset price is \$30 and we shall use two timesteps in our algorithm (so that $\delta t = 0.5$). Finally, two iterations are used at each timestep.

First we need to calculate all the S-values that we will use in our interpolation mesh. Suppose for simplicity we will use \$30, \$40, and \$50. We now wish to calculate the expected payoffs of an option when the asset price has these values at each timestep.

Calculating the expected payoffs at the $T = 1$ is very simple – it is just the payoffs of the option. Thus, our payoff mesh looks like this:

	$S = \$30$	$S = \$40$	$S = \$50$
$t = 0.0$	****	****	****
$t = 0.5$	****	****	****
$t = 1.0$	\$10.00	\$0.00	\$0.00

Now move back a timestep, so that $t = 0.5$. The expected payoffs for the option are now needed for this timestep for each of these prices.

Start for the \$40 price, we evolve the asset forward to expiry a number of times equal to the iteration count (in this case, twice).

Suppose our two new prices are \$35 and \$45. We now need to use linear interpolation to find out what our expected payoff is for these two asset prices. The linear interpolation is based on the 3 points of bivariate data we have in the $t = 1.0$ row of the interpolation mesh.

Interpolating for \$35, we get an expected payoff of \$5. Since we are now at $t = 1$, we cannot step any further and are finished with this iteration. We discount this payoff by a timestep (giving us \$4 say), and store this value.

Interpolating for \$45, we get an expected payoff of \$0. Again, we are now at $t = 1$ and are finished. Thus, our payoff for this iteration is \$0.

Having done two iterations, our calculation for this asset price is now complete, and the payoff for this asset price and timestep is the mean of the calculated payoffs.

Thus, we store $(\$4 + \$0)/2 = \$2$ for this mesh point.

Repeating for \$30 and \$50, let us assume we calculate expected payoffs of \$11 and \$0 respectively, giving us an updated interpolation mesh of

	$S = \$30$	$S = \$40$	$S = \$50$
$t = 0.0$	****	****	****
$t = 0.5$	\$11.00	\$2.00	\$0.00
$t = 1.0$	\$10.00	\$0.00	\$0.00

Finally, there is the $t = 0$ line. Again, we will use the \$40 asset price.

Suppose the two asset paths generated are

$$\$40.00 \rightarrow \$35.00 \rightarrow \$30.00$$

$$\$40.00 \rightarrow \$45.00 \rightarrow \$50.00$$

The asset price at $t = 0.5$ is \$35. We use linear interpolation to find that the expected payoff at this asset price is \$6.50. The instantaneous payoff for this asset price is \$5, and since this is less than the expected payoff at this point, we continue on to $t = 1$. This results in an asset price of \$30, giving us a payoff of \$10. Our payoff is \$10 and discount this to $t = 0$ to give us \$6 (say). Thus, the payoff for this iteration is \$6.

Proceed to the next iteration. The new asset path calculates an asset price of \$45 at $t = 0.5$. Using the interpolation mesh, this gives a payoff of \$1. The instantaneous payoff for this asset price is \$0 and so we again proceed to $t = 1$. The next asset price is \$50, giving a payoff of \$0. Thus, the payoff for the second interation is \$0.

We now find the mean of these two payoffs to find the expected payoff for \$40 at $t = 0$. The mean of the two values is \$3.

Suppose a similar procedure gives expected payoffs of \$11.50 for $S = \$30$ and \$0 for $S = \$50$. This gives us a final interpolation mesh that looks like

	$S = \$30$	$S = \$40$	$S = \$50$
$t = 0.0$	\$11.50	\$3.00	\$0.00
$t = 0.5$	\$11.00	\$2.00	\$0.00
$t = 1.0$	\$10.00	\$0.00	\$0.00

A Brief Worked Example - Stage Two

To illustrate Stage Two of the algorithm, the same parameters are used as before, and we shall use the interpolation mesh calculated in the Stage One illustration.

Thus, we have an American put option with $T = 1$, $K = \$40$, our starting asset price is \$40 and we shall use two timesteps in our algorithm (so that $\delta t = 0.5$). Finally, two iterations are used in the algorithm.

Suppose our two asset paths are:

$$\$40.00 \rightarrow \$35.00 \rightarrow \$40.00$$

$$\$40.00 \rightarrow \$30.00 \rightarrow \$35.00$$

Starting with an asset value of \$40, the price at $t = 0.5$ is \$35. Thus, the instantaneous payoff is \$5. Our expected payoff is interpolated from the bivariate data taken from the $t = 0.5$ line of the interpolation mesh. In this case, our interpolated value is \$6.50 and so the we do not exercise at this timestep. At the next timestep, the asset price is \$40, so the payoff is \$0 for this iteration.

On the next iteration, the asset price at $t = 0.5$ is \$30. In this case, the instantaneous payoff is \$10 and the interpolation of the expected payoff is \$11, and so we hold the option. The next price is \$35. Thus, the payoff is \$5 and we discount this back to $t = 0$ to give us \$3 (say). Thus, the payoff is \$3 for this iteration.

Finally, the value for the option is the mean of the two calculated values, which is $(\$3 + \$0)/2 = \$1.50$.

Algorithm 6-5 Expected Payoff Interpolation - Generating the Interpolation Mesh

1 – Determine the spread of asset values for which we require the interpolation data to be calculated, S_{Ii}.

2 – Starting at $t = T$, the expected payoffs are simply the payoffs given by Payoff(S, K).

3 – For each time-step:

a) Step back a time-step $t \to t - \delta t$.

b) For each interpolation mesh-point we perform the following iteration a number of times defined by the simulation parameters:

I. Using $S = S_{Ii}$ as a starting point, evolve the asset back to $t = T$ using time-steps of the same size δt.

II. At each of these new time-steps:

i. Calculate the instantaneous payoff, P_{exercise}, using the payoff formula.

ii. Calculate the expected payoff, P_{expected}, by interpolating from the expected payoff data at this time-step. This data has already been calculated since the overall algorithm is stepping backwards through time.

iii. If $P_{\text{exercise}} > P_{\text{expected}}$, then exercise at this time-step, taking a note of the time t_{exercise}.

iv. Discount the payoff from t_{exercise} to t.

III. Repeat for each iteration.

c) Repeat this process for all the other interpolation mesh values.

d) Store all the expected payoff values at this time-step. These values will be used to interpolate all payoffs at this time-step for the rest of the simulation.

4 – Repeat for each time-step all the way back to $t = 0$.

Algorithm 6-6 Expected Payoff Interpolation - Calculating the Option Price

1 – For each iteration, initialise $S = S_0$, $t = 0$.

2 – For each time-step:

a) Advance a single time-step $t \rightarrow t + \delta t$.

b) Update $S \rightarrow S + dS$, where

$$dS = rS\,dt + \sigma S\,dX$$

where dX is a random variable drawn from a standardised normal distribution.

c) Calculate P_{exercise}, the value of option if it were exercised immediately.

d) Calculate P_{expected} using an interpolation of the values at time t generated in Stage One.

e) Compare P_{exercise} to P_{expected}. If P_{exercise} is larger, then the option is exercised immediately, otherwise move onto the next time-step.

3 – Discount the resulting payoff using the time at which the option was exercised. For early exercise scenarios, this time will be less than the contract duration, T.

4 – Record this result, and start a new iteration.

5 – Average all the results over each iteration to find the option price.

6.4.3 The Iterative Ito Method

This method relys heavily on the fact that the price of an option V can be viewed as arising from two elements: the **intrinsic value** and the **time value**. The option can be exercised immediately, and so has an intrinsic value, V_{int}, the payoff obtained. Also, since the value of this payoff may increase in the future, the option also has a time value, V_{time}. Thus, we have

$$V = V_{\text{int}} + V_{\text{time}}. \tag{6.8}$$

In theory, the optimal exercise point is when the time value of the option drops to zero (and so there is nothing to be gained by holding onto the option). The difficulty is that finding the time value at any given instant is non-trivial.

The value of the option is constantly changing as the asset evolves, causing corresponding changes in both V_{int} and V_{time}, and so it is necessary to be able to first calculate this evolution of the option price, and then split this value into its time and intrinsic parts.

Thus, at every point on the asset path, the value of the option is calculated and compared to the current intrinsic value. If there is no difference between the two, the option price has no time component and so can be exercised.

We cannot do this for continuous time, and so we break up the lifetime of the option into N discrete time-steps of length δt, and so we can use Algorithm 6-7.

Obviously, this is not a practical solution as it requires a full knowledge of the option price as the asset evolves, even allowing for the discretisation of time, and so traders use close approximations of prices for deciding whether or not to exercise.

So, now that we know how an option is used in the financial world, how do we translate this to calculating its price?

Although the above algorithm is useful for providing insight into what is required from our price-calculation algorithm, it does not really help us find that algorithm, as it requires full knowledge of the option price to be able to determine the optimal exercise point. Thus, we have a circular algorithm.

Algorithm 6-7 Trader's Method for Exercising an Option

1 – Start at $t = 0$.

2 – While $t < T$, repeat:

a) Add δt to t.

b) Update the asset value at time t, S_t, using

$$dS = rS\,dt + \sigma S\,dX$$

c) Calculate $V(S_t, T - t)$.

d) Calculate $V_{\text{int}} = \text{Payoff}(S_t, K, t)$ and $V_{\text{time}} = V - V_{\text{int}}$.

e) If $V_{\text{time}} = 0$, then exercise.

3 – Exercise at $t = T$ if $\text{Payoff}(S_T, K, T) > 0$.

Fortunately, such circular calculations occur frequently in numerical problems, so all is not lost.

Usually, we can get a good approximation of the solution using an iterative process, which can be summarised as follows:

Algorithm 6-8 Standard Algorithm for Iterative Techniques

1 – Take an inital estimate of our unknown quantity X, X_0 say.

2 – Using X_0, proceed with the calculation.

3 – This calculation will give us a new value for X, X_1.

4 – Use X_1 as our new value and start the process again.

5 – Repeat these steps until the changes in successive values of X go below a given tolerance level (Note that some iterative algorithm use different methods of calculating accuracy).

In principle, this solves the problem of not knowing the initial option price in advance, but it still does not allow us to know how the option prices changes over the evolution of the asset, $0 \leq t \leq T$.

None of the above discussion has taken performance of the algorithms into account, and the above iterative method may prove to be far too expensive, but it is fair to say that using an iterative method again at this point would not be too expensive.

In any case, it can be readily seen that over use of iterative techniques to calculate unknown values would make the algorithms extremely complex immediately, which can only lead to problems during implementation.

Thus, another method is required.

One option is to use Ito's Lemma (Eqn 1.5) to approximately calculate the changes in V as the asset price S and time t evolve. To recap, we have

$$dV = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S}dS + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt. \tag{6.10}$$

Once found, we can add dV to V to estimate the current value of the option at each time-step.

The intrinsic value of the option V_{int} is readily found from the payoff function of the option, and so we can also calculate the time-value V_{time} quite easily.

Now we need to calculate dV using Eqn (1.5).

An immediate problem becomes apparent: the equation uses the Greeks to calculate dV, but how can we know what the derivatives are if we don't even know what V is?

To find the Greeks, we need to be able to calculate V and then use a finite differencing approximation or something similar, but to find V we need to know the value of the greeks.

Again, an iterative technique could work here, and work has been done to show that it will work in this case (though there is a systematic error involved).

We evolve a differential volume of the phase space of a option price with known greeks in the ranges $(S - \delta S, S + \Delta S)$ and $(t, t + \Delta t)$. Note that the small quantities ΔS and Δt are used purely for calculating the greeks.

Having evolved this differential volume by a few time-steps, we then calculate the new values for the Greeks and plug these new values back into the algorithm, repeating the process until the changes in all the Greeks fall below a given tolerance level.

We can then compare this value to our known values at each time-step and see if our results are accurate.

Verification of the Iterative Algorithm

This entire algorithm hangs on the assumption that an iterative process will converge on the final solution.

It is immediately obvious that there is no clear, logical basis upon which to justify this initial assumption, so some verification is required.

Fortunately, we can again fall back on the fact that we have a complete solution for the European option problem. A good test for the method would be to use it in this case and see how closely our algorithmic solution can match the analytic solution.

The algorithm is shown in Algorithm 6-9.

The results of this should be that the iterative algorithm produces good approximations to the analytically calculated ones.

Results from this algorithm are discussed at the end of this chapter.

The Iterative Ito Algorithm for American Options

Using an iterative technique to estimate the greeks, and hence dV, is acceptable here since the algorithm is efficient and does not provoke cascading levels of iteration to get a result.

Summary

At this point it is probably worth pausing and summarising what we have done so far, since it is very easy to lose track of the goals through all the details.

- We are trying to develop a Monte Carlo method for calculating prices for options that have early exercise features, and we are trying to do so by developing a method for reliably determining the optimal exercise time, as it would appear to a holder of the contract.

- To do this, we use the fact that the option price consists of an **intrinsic** and **time** value and it is optimal to exercise the option when the time value, V_{time} is zero.

- The intrinsic value of the option, V_{int}, is simply the amount obtained from immediate exercise, and so is readily calculated. To find V_{time}, we need the current price of the option V, and use $V_{\text{time}} = V - V_{\text{int}}$.

- To calculate V we use an iterative technique, continually performing the Monte Carlo simulation to refine our initial value of V, which is our final solution.

- The Monte Carlo simulation requires that we know the values of V during the simulation, so we calculate changes in V using Ito's Lemma:

$$dV = \frac{\partial V}{\partial t} dt + \frac{\partial V}{\partial S} dS + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} dt. \tag{6.15}$$

- At each time-step calculation, we use an iterative method to calculate the values of the greeks for the above equation.

Implementation

With all the problems solved, we now have an algorithm which should, at least in principle, calculate the value of an American option.

In the discussion above, very little consideration was given to algorithm efficiency or execution times.

One major drawback of the above technique is the fact that the main focus of the iterative convergence, the price of the option, requires a lot of work to calculate.

Each price calculation requires M iterations, each iteration requires N time-steps, and each time-step requires at most K subiterations to calculate the greeks.

Thus, each calculation is $O(NMK)$.

This is for just a single calculation of the price and so this algorithm could prove expensive if high accuracy is important, since it is not known in advance how many iterations will be required for convergence.

We now need to investigate the possibility of calculating the option price using the principles outlined above without having to resort to an iterative technique to converge on an answer.

The simplest possibility is to remove the iterative part of the technique, i.e. we take an initial estimate for the option price, perform the calculation and use our output as our final answer.

For this to be successful, our initial guess becomes very important for calculating our final price, and so must be chosen with care. This method will be far from ideal and its dependency on the initial guess renders it largely impractical for regular use.

Algorithm 6-9 Verification of the Iterative Algorithm to Calculate the Greeks

1 – Set the stockstep and time-step parameters h and k.

2 – Set $t = 0$.

3 – While $t < T$, repeat:

a) Update $t \to t + dt$.

b) Initialise solution estimates $\frac{\partial V}{\partial t}_{\text{guess}}$, $\frac{\partial V}{\partial S}_{\text{guess}}$, $\frac{\partial^2 V}{\partial S^2}_{\text{guess}}$.

c) Evolve S using

$$S = S \exp\left(r - \frac{1}{2}\sigma^2\right) dt + (\sigma\sqrt{dt}\ \phi)$$

and calculate the corresponding dS.

d) Repeat for each iteration until (error < TOL):

I. Using the analytic solution, initialise the volume values $V(S,t)$, $V(S+\Delta S,t)$, $V(S-\Delta S,t)$, $V(S,t+\Delta t)$.

II. Calculate the Ito updates for these values:

$$dV = \frac{\partial V}{\partial t}_{\text{guess}} dt + \frac{\partial V}{\partial S}_{\text{guess}} dS + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}_{\text{guess}} dt.$$

III. Update the volume values $V(S,t)$, $V(S+\Delta S,t)$, $V(S-\Delta S,t)$, $V(S,t+\Delta t)$.

IV. Update the value of the Greeks:

$$\begin{aligned}
\Delta = \frac{\partial V}{\partial S} &= \frac{V(S+\Delta S,t) - V(S-\Delta S,t)}{2\Delta S}, \\
\Gamma = \frac{\partial^2 V}{\partial S^2} &= \frac{V(S+\Delta S,t) - 2V(S,t) + V(S-\Delta S,t)}{(\Delta S)^2} \\
\Theta = \frac{\partial V}{\partial t} &= \frac{V(S+\Delta S,t) - V(S-\Delta S,t)}{2\Delta t}.
\end{aligned}$$

V. Calculate the error ratios:

$$\begin{aligned}
\text{ratio1} &= \frac{\Delta - \Delta_{\text{old}}}{\Delta_{\text{old}}} \\
\text{ratio2} &= \frac{\Gamma - \Gamma_{\text{old}}}{\Gamma_{\text{old}}} \\
\text{ratio3} &= \frac{\Theta - \Theta_{\text{old}}}{\Theta_{\text{old}}}
\end{aligned}$$

VI. error = Max(ratio1, ratio2, ratio3)

e) Output the calculated values for $\frac{\partial V}{\partial S}$, $\frac{\partial V}{\partial t}$, $\frac{\partial^2 V}{\partial S^2}$ at this time-step and compare to the values from the analytical solution.

Algorithm 6-10 The Iterative Holding Method

1 – Choose an initial guess for the option price, V_{guess}.

2 – Set $V = V_{\text{guess}}$, $V_{\text{prev}} = V_{\text{guess}}$.

3 – Repeat the following steps while $\frac{V - V_{\text{prev}}}{V_{\text{prev}}} > V$-*tolerance*:

a) Start at $t = 0$.

b) Repeat for each Monte Carlo iteration:

I. While $t < T$, repeat:

i. Add δt to t.

ii. Update the asset value at time t, $S(t)$, using

$$S(t) = S \exp\left((r - \frac{1}{2}\sigma^2)\, dt + (\sigma\phi)\sqrt{dt}\right),$$

where ϕ is a random variable drawn from a standardised normal distribution.

iii. Calculate $dS = S(t) - S(t - \delta t)$.

iv. Iteratively calculate dV using algorithm 6-11.

v. Update V using $V(t) = V(t - \delta t) + dV$.

vi. Calculate $V_{\text{int}} = \text{Payoff}(S_t, K, t)$.

vii. Calculate $V_{\text{time}} = V - V_{\text{int}}$.

viii. If $V_{\text{time}} = 0$, then set $t_{\text{expiry}} = t$ and exercise.

II. If $t = T$ then set $t_{\text{expiry}} = T$ and exercise.

III. Calculate the payoff and discount it back to $t = 0$.

c) Average over each Monte Carlo iteration to find a new value for the option price V.

4 – The final value V from this iterative calculation is the final result.

Algorithm 6-11 Iterative Calculation for dV

1 – Set initial values for iterative dV calculation:

$$\begin{aligned} dV &= 1, \\ \frac{\partial V}{\partial t} &= 1, \\ \frac{\partial V}{\partial S} &= 0, \text{ and} \\ \frac{\partial^2 V}{\partial S^2} &= 1. \end{aligned}$$

2 – Set internal stepping values for S and t, h and k respectively.

3 – Set $dV_{\text{prev}} = dV$.

4 – Repeat while $\frac{dV - dV_{\text{prev}}}{dV_{\text{prev}}} > dV\text{-}tolerance$:

a) Calculate:

$$\begin{aligned} V &= V(S,t), \\ V_{\text{pS}} &= V(S+h,t), \\ V_{\text{mS}} &= V(S-h,t), \\ V_{\text{pT}} &= V(S,t+k), \text{ and} \\ V_{\text{mT}} &= V(S,t-k). \end{aligned}$$

b) Calculate the derivatives of V using:

$$\begin{aligned} \frac{\partial V}{\partial t} &= \frac{(V_{\text{pT}} - V_{\text{mT}})}{2k}, \\ \frac{\partial V}{\partial S} &= \frac{(V_{\text{pS}} - V_{\text{mS}})}{2h}, \text{ and} \\ \frac{\partial^2 V}{\partial S^2} &= \frac{(V_{\text{pS}} - 2V + V_{\text{mS}})}{h^2}. \end{aligned}$$

c) Calculate a new value of dV using Ito's Lemma,

$$dV = \frac{\partial V}{\partial t} dt + \frac{\partial V}{\partial S} dS + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} dt.$$

5 – The final value dV from this iterative calculation is the final result.

6.5 Analysis of the Monte Carlo Techniques for European Options

Once we have obtained some results from the Monte Carlo techniques, it is very important to analyse the results so that we can draw conclusions both on the data itself and on the various techniques used to solve the problem.

The basic idea is that of a **Simulation Ensemble**. A Simulation Ensemble is a fixed number of iterations of the basic Monte Carlo simulations, with each ensemble having a variation of one or more of the simulation parameters.

With a fixed number of ensemble iterations, a comparison on the dependency of the different parameters on the option price could be performed.

This comparison is done by placing the output data from the ensemble into a frequency histogram and observing how the histogram changes for different values of the parameters in the simulation.

For the current problem, the only parameters varied were the number of time-steps used and the number of random-walk simulations performed per execution.

It is expected that the histogram produced will be of roughly normal shape, and the goodness of fit of this histogram to the standard normal distribution can be used to determine information on the dependency of the calculated price to the number of walk simulations and time-stepping count.

The results of the Monte Carlo solutions are shown in Fig. 6-1 and Fig. 6-2.

The first two histograms are for the European Call option price, and the single vertical line represents the price calculated using the exact solution.

The parameters used in this calculation were

$$\begin{aligned} S &= 1 \\ K &= 1 \\ T &= 1 \\ \sigma &= 0.5 \\ r &= 5\% \end{aligned}$$

It is immediately obvious from Fig 6-1 that as the number of random-walk realisations increase, the histogram gets narrower around the actual value (note that the diagrams are not on the same scale), showing that the algorithm gets more accurate with the number of simulations per price calculation, as expected.

Conversely, from Fig 6-2, the increasing number of time-steps per realisation has no real effect on the accuracy of the option price, as the histograms do not change in any significant way. Again, this is as expected since the time-stepping formula was exact and so will not be improved by increasing the number of time-steps used.

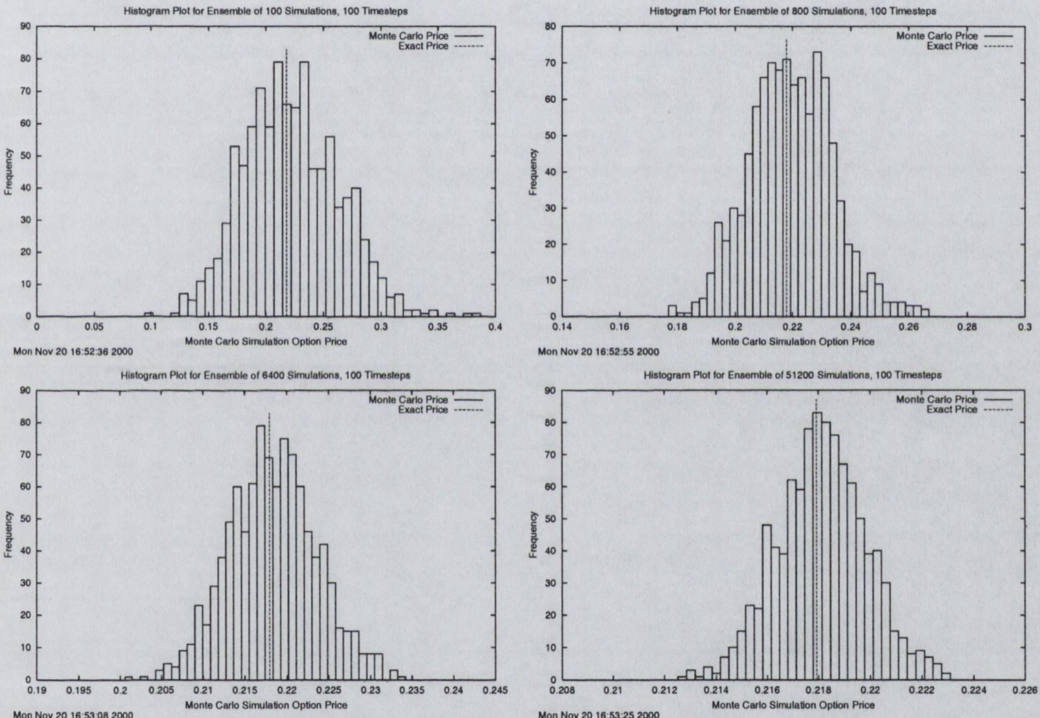

Figure 6-1: Variation of the European Call option price wrt the Number of Random-Walk Realisations

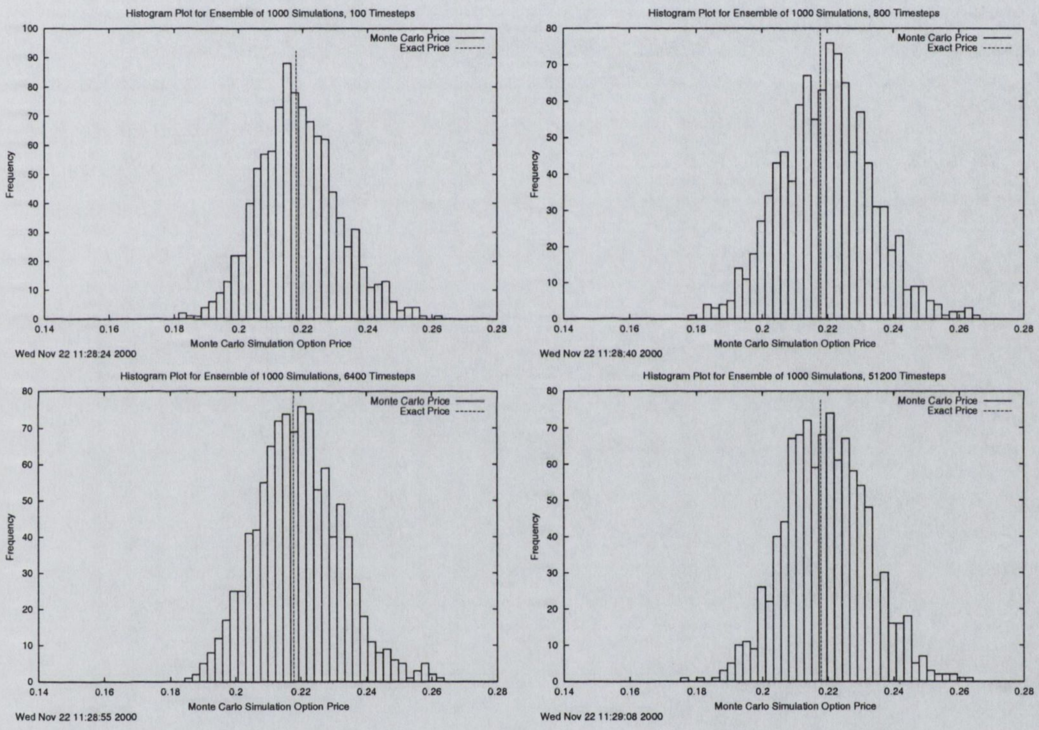

Figure 6-2: Variation of the European Call option price wrt the Number of Time-Steps per Random-Walk

6.5.1 Analysis of the Monte Carlo Techniques for American Options

The Interpolated Expectation Algorithm

Table 6.1 shows the results of the Interpolated Expectation algorithm. The prices calculated agree very strongly with both the finite difference solution as well as the Regressed Expectation algorithms. The errors are in the region of a few cents. On some exchanges, option prices are still quoted as fractions of a dollar (or local currency), and so this is accurate enough. Even on electronic exchanges, the errors below are within the usual bid-ask spreads quoted for an option.

S	T	σ	**Fin Diff**	**Regression**	**Interpolation**
36	1	0.2	4.478	4.472 ± 0.010	4.485 ± 0.010
36	1	0.4	7.101	7.091 ± 0.020	7.090 ± 0.020
36	2	0.2	4.840	4.821 ± 0.012	4.826 ± 0.012
36	2	0.4	8.508	8.488 ± 0.024	8.458 ± 0.024
38	1	0.2	3.250	3.244 ± 0.009	3.236 ± 0.010
38	1	0.4	6.148	6.139 ± 0.019	6.183 ± 0.019
38	2	0.2	3.745	3.735 ± 0.011	3.737 ± 0.011
38	2	0.4	7.670	7.669 ± 0.022	7.644 ± 0.023
40	1	0.2	2.314	2.313 ± 0.009	2.307 ± 0.009
40	1	0.4	5.312	5.308 ± 0.018	5.311 ± 0.018
40	2	0.2	2.885	2.879 ± 0.010	2.884 ± 0.011
40	2	0.4	6.920	6.921 ± 0.022	6.863 ± 0.024
42	1	0.2	1.617	1.617 ± 0.007	1.607 ± 0.008
42	1	0.4	4.852	4.588 ± 0.017	4.564 ± 0.018
42	2	0.2	2.212	2.206 ± 0.010	2.201 ± 0.010
42	2	0.4	6.248	6.243 ± 0.021	6.183 ± 0.022
44	1	0.2	1.110	1.118 ± 0.007	1.102 ± 0.007
44	1	0.4	3.948	3.957 ± 0.017	3.965 ± 0.017
44	2	0.2	1.690	1.675 ± 0.009	1.668 ± 0.009
44	2	0.4	5.647	5.622 ± 0.021	5.630 ± 0.021

Table 6.1: Interpolated Expectation Results for an American put option with $K = 40$, $r = 6\%$, 50 timesteps per year, 100 interpolation points per timestep, and 10000 iterations at each interpolation node

Tables 6.2 and 6.3 show the results obtained from the mesh generation stage of the Interpolated Expectation algorithm (Algorithm 6-5) i.e. the values calculated for the expected payoff of the option at each timestep.

S	T	σ	**Fin Diff**	**Regression**	**Interp. Mesh**	**Rel Error**
36	1	0.2	4.478	4.472 ± 0.010	4.565 ± 0.010	0.019
36	1	0.4	7.101	7.091 ± 0.020	7.334 ± 0.021	0.033
36	2	0.2	4.840	4.821 ± 0.012	5.002 ± 0.012	0.033
36	2	0.4	8.508	8.488 ± 0.024	9.028 ± 0.025	0.061
38	1	0.2	3.250	3.244 ± 0.009	3.325 ± 0.010	0.023
38	1	0.4	6.148	6.139 ± 0.019	6.366 ± 0.020	0.035
38	2	0.2	3.745	3.735 ± 0.011	3.934 ± 0.012	0.050
38	2	0.4	7.670	7.669 ± 0.022	8.182 ± 0.025	0.067
40	1	0.2	2.314	2.313 ± 0.009	2.390 ± 0.009	0.033
40	1	0.4	5.312	5.308 ± 0.018	5.507 ± 0.019	0.037
40	2	0.2	2.885	2.879 ± 0.010	3.051 ± 0.011	0.058
40	2	0.4	6.920	6.921 ± 0.022	7.395 ± 0.024	0.069
42	1	0.2	1.617	1.617 ± 0.007	1.669 ± 0.008	0.032
42	1	0.4	4.852	4.588 ± 0.017	4.775 ± 0.018	0.016
42	2	0.2	2.212	2.206 ± 0.010	2.325 ± 0.010	0.051
42	2	0.4	6.248	6.243 ± 0.021	6.710 ± 0.024	0.074
44	1	0.2	1.110	1.118 ± 0.007	1.160 ± 0.007	0.045
44	1	0.4	3.948	3.957 ± 0.017	4.133 ± 0.017	0.047
44	2	0.2	1.690	1.675 ± 0.009	1.813 ± 0.009	0.073
44	2	0.4	5.647	5.622 ± 0.021	6.077 ± 0.023	0.076

Table 6.2: Interpolated Expectation interpolation values for an American put option with $K = 40$, $r = 6\%$, 50 timesteps per year, 100 interpolation points per timestep, and 10000 iterations at each interpolation node

It is immediately obvious that the algorithm is over-estimating the value of the expected payoff. The relative error is small but significant (ranging from just under 2% to just under 8%), and is always larger than the values calculated for the option price by either the finite difference method or the Regression Estimation of the Expected Payoff algorithm (Algorithm 6-3).

This overestimation bias is a systematic error. Since the algorithm works backward in time, it allows the holder to exercise at a more optimal time as he has foreknowledge of the expectation at future points.

Table 6.2 shows the values generated by the interpolation mesh where 100 values are calculated at each timestep for the purposes of the expectation interpolation.

Table 6.3 shows the corresponding values generated using 200 values on the interpolation. As can be seen, the effect is relatively small, with all the differences lying with the error margins. There are no results for the $T = 2$, $\sigma = 0.4$ configuration because the calculation took too long on the cluster being used. The time limit was 96 hours.

Linear interpolation was used in all of the results shown.

S	T	σ	**Fin Diff**	**Regression**	**Interp. Mesh**	**Rel Error**
36	1	0.2	4.478	4.472 ± 0.010	4.572 ± 0.010	0.021
36	1	0.4	7.101	7.091 ± 0.020	7.362 ± 0.021	0.037
36	2	0.2	4.840	4.821 ± 0.012	5.020 ± 0.012	0.037
36	2	0.4	8.508	8.488 ± 0.024	***	***
38	1	0.2	3.250	3.244 ± 0.009	3.329 ± 0.010	0.024
38	1	0.4	6.148	6.139 ± 0.019	6.378 ± 0.020	0.037
38	2	0.2	3.745	3.735 ± 0.011	3.921 ± 0.012	0.047
38	2	0.4	7.670	7.669 ± 0.022	***	***
40	1	0.2	2.314	2.313 ± 0.009	2.396 ± 0.009	0.035
40	1	0.4	5.312	5.308 ± 0.018	5.523 ± 0.019	0.040
40	2	0.2	2.885	2.879 ± 0.010	3.055 ± 0.011	0.059
40	2	0.4	6.920	6.921 ± 0.022	***	***
42	1	0.2	1.617	1.617 ± 0.007	1.678 ± 0.008	0.038
42	1	0.4	4.852	4.588 ± 0.017	4.762 ± 0.018	0.019
42	2	0.2	2.212	2.206 ± 0.010	2.333 ± 0.010	0.055
42	2	0.4	6.248	6.243 ± 0.021	***	***
44	1	0.2	1.110	1.118 ± 0.007	1.146 ± 0.007	0.032
44	1	0.4	3.948	3.957 ± 0.017	4.126 ± 0.018	0.045
44	2	0.2	1.690	1.675 ± 0.009	1.814 ± 0.009	0.073
44	2	0.4	5.647	5.622 ± 0.021	***	***

Table 6.3: Interpolated Expectation interpolation values for an American put option with $K = 40$, $r = 6\%$, 50 timesteps per year, 200 interpolation points per timestep, and 10000 iterations at each interpolation node

In terms of timing, an overwhelming amount of the calculation time is spent in calculating the interpolation mesh, and this has been shown through timing the code.

The code was ran for 100,000 iterations, with 10,000 iterations used at each of the 50 mesh interpolation nodes, for the standard set of parameters ($S = \$36$, $K = \$40$, $T = 1$, $\sigma = 0.2$, $r = 6\%$). Again, 50 timesteps were used per year (giving a timestep size of $\delta t = 0.02$).

Total	Computation	System
693.91	692.74	1.07
7.21	7.02	0.16

Table 6.4: Timings for execution of the Interpolated Expectation algorithm with 100,000 iterations, 10,000 'mini' iterations at each interpolation node, 50 interpolation nodes, using 50 timesteps per year. The option parameters were $S = \$36$, $K = \$40$, $T = 1$, $\sigma = 0.2$, and $r = 6\%$.

The main column is the second one ("Computation") as this is the number of seconds that the processor used to execute the code. The "System" column is less useful, since it includes time spend on other tasks, file I/O, and others.

In any case, having the interpolation mesh speeds up the calculation by a factor of almost 100, which is a very significant speed up. Thus, should a number of option prices be required for different stock prices, re-using the interpolation mesh will substantially reduce the amount of computation overhead required.

The major downside to the algorithm is that it does not converge on the correct solution. While this is usually fine for practical purposes, since the result lies within the usual bid/ask spread for an option, it is hardly ideal for a computational point of view.

More work is definitely required to improve this.

The Iterated Ito Algorithm

We now turn our attention to the Iterated Ito algorithm (Algorithm 6-10).

We first must check the verification algorithm for the European option Greeks to see if we can use this kind of iterative technique to get good approximations in the first place.

An immediate problem becomes apparent: the Ito update,

$$dV = \frac{\partial V}{\partial t} dt + \frac{\partial V}{\partial S} dS + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \tag{6.16}$$

does not have an explicit t-dependance.

As the dV does not depend on t explicitly, it will update both $V(S,t)$ and $V(S,t+\delta t)$ by the same amount. Since we then take the difference of these quantities when calculating the updated value of Theta we end up with the same value with which we started i.e. the value of Theta, Θ, at the beginning of each iteration is identical to that at the end.

The output of the algorithm illustrates this problem. For the Delta and Gamma it works much better. The results are shown in Table 6.5 for $t \in (0, 0.5)$. The full simulation ran from $t = 0$ to $t = T = 1$, but the results shown are representative of the whole, the rest being removed for space considerations.

Although the iterative technique does not calculate the Greeks precisely, it is giving close approximations for the Delta and Gamma (the first and second order partial derivatives of the option price V with respect to the asset price S, important quantities in dynamic hedging).

As predicted, the value for Theta is wildly inaccurate, and does not change throughout the iterations.

This poses an immediate problem for the Iterated Ito algorithm.

The algorithm may not necessarily require very accurate values for the Greeks (since the Ito updates are just being used to determine when the option should be exercised early), but it does require them to give a value which approximates the true value. This is simply not the case for the Theta.

With this in mind, we found a consistent and strong underestimation bias in the results from our Iterated Ito algorithm, the results of which are shown in Table 6.6. The number of iterations required for convergence is also shown.

There are a number of reasons for bias, the first being that the Ito updates are not accurate as they cannot approximate the Theta, Θ. While the contribution of the Theta-term to the Ito update is small, this error tends to cause the algorithm to

t	Δ_{calc}	Δ_{actual}	Γ_{calc}	Γ_{actual}	Θ_{calc}	Θ_{actual}
$t = 0.000$	-0.6585	-0.7254	0.0547	0.0489	0.4659	-1.6163
$t = 0.020$	-0.6434	-0.7113	0.0555	0.0501	0.3908	-1.6979
$t = 0.040$	-0.5871	-0.6592	0.0569	0.0527	0.1645	-1.9462
$t = 0.060$	-0.6867	-0.7488	0.0548	0.0486	0.5346	-1.5509
$t = 0.080$	-0.6574	-0.7222	0.0564	0.0508	0.3964	-1.6976
$t = 0.100$	-0.6159	-0.6838	0.0580	0.0532	0.2147	-1.8945
$t = 0.120$	-0.5941	-0.6631	0.0588	0.0545	0.1140	-2.0067
$t = 0.140$	-0.6257	-0.6916	0.0588	0.0539	0.2134	-1.8987
$t = 0.160$	-0.6760	-0.7364	0.0577	0.0517	0.4008	-1.7007
$t = 0.180$	-0.7360	-0.7889	0.0546	0.0477	0.6576	-1.4399
$t = 0.200$	-0.7671	-0.8154	0.0524	0.0452	0.7958	-1.3053
$t = 0.220$	-0.7630	-0.8114	0.0532	0.0461	0.7574	-1.3459
$t = 0.240$	-0.7512	-0.8007	0.0548	0.0478	0.6783	-1.4270
$t = 0.260$	-0.7371	-0.7878	0.0566	0.0498	0.5863	-1.5217
$t = 0.280$	-0.7137	-0.7665	0.0590	0.0525	0.4480	-1.6644
$t = 0.300$	-0.8050	-0.8456	0.0506	0.0433	0.9154	-1.2037
$t = 0.320$	-0.8689	-0.8987	0.0411	0.0340	1.2949	-0.8505
$t = 0.340$	-0.8780	-0.9059	0.0398	0.0328	1.3423	-0.8104
$t = 0.360$	-0.8425	-0.8761	0.0467	0.0395	1.0910	-1.0463
$t = 0.380$	-0.8321	-0.8669	0.0489	0.0417	1.0049	-1.1309
$t = 0.400$	-0.7195	-0.7680	0.0629	0.0565	0.3213	-1.8073
$t = 0.420$	-0.6301	-0.6855	0.0689	0.0642	-0.1451	-2.3011
$t = 0.440$	-0.6636	-0.7157	0.0685	0.0631	-0.0273	-2.1746
$t = 0.460$	-0.6499	-0.7023	0.0702	0.0651	-0.1301	-2.2853
$t = 0.480$	-0.6694	-0.7195	0.0703	0.0648	-0.0757	-2.2277
$t = 0.500$	-0.5936	-0.6478	0.0745	0.0706	-0.4659	-2.6527

Table 6.5: Comparison of Greeks from Iterative Algorithm and Analytic Values

optimise at a time that is sub-optimal, thus giving a value for the option which is lower than otherwise expected.

Another major problem, as it stands, is the convergence of the algorithm. A sample output of the Iterative Ito code is shown in Table 6.7. The convergence of each iteration is not smooth, with the value bouncing around and finally settling on a final solution. This can make for very erratic convergence counts, ranging from 7 iterations to 74 iterations across sixteen different configurations.

Thus, the results of the algorithm can be skewed by two successive iterations occuring close to one another by pure random chance. The algorithm will then terminate, despite there being no guarantee of the next iteration also being within the tolerance level.

With all these factors, it is not surprising that the algorithm does not produce accurate results.

S	T	σ	**Fin Diff**	**Regression**	**Iterative Ito**	
36	1	0.2	4.478	4.472 ± 0.010	4.26	(39 iters)
36	1	0.4	7.101	7.091 ± 0.020	5.79	(9 iters)
36	2	0.2	4.840	4.821 ± 0.012	4.48	(33 iters)
36	2	0.4	8.508	8.488 ± 0.024	6.46	(41 iters)
38	1	0.2	3.250	3.244 ± 0.009	2.77	(15 iters)
38	1	0.4	6.148	6.139 ± 0.019	4.61	(22 iters)
38	2	0.2	3.745	3.735 ± 0.011	2.97	(53 iters)
38	2	0.4	7.670	7.669 ± 0.022	5.28	(12 iters)
40	1	0.2	2.314	2.313 ± 0.009	1.61	(53 iters)
40	1	0.4	5.312	5.308 ± 0.018	3.58	(57 iters)
40	2	0.2	2.885	2.879 ± 0.010	1.88	(10 iters)
40	2	0.4	6.920	6.921 ± 0.022	4.30	(33 iters)
42	1	0.2	1.617	1.617 ± 0.007	0.90	(35 iters)
42	1	0.4	4.852	4.588 ± 0.017	2.92	(65 iters)
42	2	0.2	2.212	2.206 ± 0.010	1.15	(74 iters)
42	2	0.4	6.248	6.243 ± 0.021	3.54	(47 iters)
44	1	0.2	1.110	1.118 ± 0.007	0.52	(23 iters)
44	1	0.4	3.948	3.957 ± 0.017	2.21	(7 iters)
44	2	0.2	1.690	1.675 ± 0.009	0.74	(18 iters)
44	2	0.4	5.647	5.622 ± 0.021	3.01	(16 iters)

Table 6.6: Iterative Ito results for 50 time-steps per year, $r = 0.06$ with an iteration tolerance of 0.001

```
Initial Guess:   1.000000

                 V                V_prev           Rel change
V Iteration 1    3.928396         1.000000         2.928396
V Iteration 2    4.230936         3.928396         0.077014
V Iteration 3    4.331693         4.230936         0.023814
V Iteration 4    4.258384         4.331693         0.016924
V Iteration 5    4.233188         4.258384         0.005917
V Iteration 6    4.286959         4.233188         0.012702
V Iteration 7    4.229965         4.286959         0.013295
V Iteration 8    4.292180         4.229965         0.014708
V Iteration 9    4.248276         4.292180         0.010229
V Iteration 10   4.289446         4.248276         0.009691
V Iteration 11   4.251653         4.289446         0.008811
V Iteration 12   4.256610         4.251653         0.001166
V Iteration 13   4.320329         4.256610         0.014970
V Iteration 14   4.269716         4.320329         0.011715
V Iteration 15   4.270077         4.269716         0.000085
Final solution:  4.270077
```

Table 6.7: Output of the Iterative Ito code

Chapter 7

The Tree Model Approach

7.1 Introduction

Up to this point, we have been primarily concerned with the Black-Scholes model for pricing options. It is worth that there are other methods for pricing options, the most commonly-known of which are Tree Methods.

Tree methods take a different, but related, approach. Rather than using the initial price of the asset and evolving this price through time, tree methods construct a tree of prices, with the starting price at the root.

Various algorithms are then used to calculate the asset price using the values at each node of the tree.

7.2 The Binomial Model

Probably the most famous of the tree methods is the Binomial Model.

The Binomial Model is popular because it can be used and understood without any knowledge of higher mathematics (such a Stochastic Mathematics and Differential Calculus) and does not use partial differential equations.

Although this has the benefit of simplicity, it does have a limiting effect on the application of the model.

We assume that the asset, with initial value S, can either rise to a value uS or fall to a value vS over a time interval δt, where $0 < v < 1$ and $u > 1$. We assign a probability p to the event $S \to uS$, and hence the probability of $S \to vS$ is $1 - p$.

So, we have three unknowns, u, v and p, requiring three constraints to fix these values.

Two of these constraints come from the fact that it is necessary for these transitions to result in the same drift μ and variance σ^2 as found in the asset price model Eq. (3.19):

$$dS = \mu S\, dt + \sigma S\, dX$$

For a random walk to have the correct drift we need:

$$pu + (1-p)v = e^{\mu\, \delta t}$$

Rearranging, we get

$$p = \frac{e^{\mu\, dt} - v}{u - v} \tag{7.3}$$

To ensure the random walk has the correct variance, we need

$$pu^2 + (1-p)v^2 = e^{(2\mu+\sigma^2)\delta t}. \tag{7.4}$$

We now have two constraints on three unknowns, giving us a single degree of freedom.

Thus we can arbitrarily choose a third constraint to tailor our model depending on our preferences.

Also, as before, the drift is assumed to be the risk-free interest rate, and so we have $\mu = r$.

As an example, we could desire that a rise and fall in the asset price be equally likely, giving us the constraint

$$p = \frac{1}{2}.$$

However, the standard constraint is to allow the asset price to return to its original value after two time-steps, i.e. $uvS = S$.

$$u = \frac{1}{v}. \tag{7.6}$$

Solving for these constraints Eq. (7.3, Eq. (7.4), and Eq. (7.6), we have

$$u = \frac{1}{2}(e^{-\mu\,\delta t} + e^{(\mu+\sigma^2)\,\delta t}) + \frac{1}{2}\sqrt{(e^{-\mu\,\delta t} + e^{(\mu+\sigma^2)\,\delta t})^2 - 4}. \tag{7.7}$$

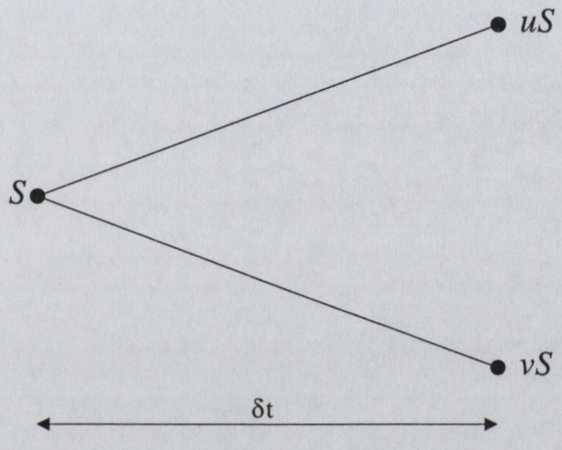

Figure 7-1: A Time-Step in the Binomial Model

So, our Binomial Model allows the asset value to move up or down by a prescribed amount over a single time-step. If we fix the starting asset value to be S at time $t = 0$, then the new value is uS or vS at time $t = \delta t$. Extending onto the next time-step, our new possible values are u^2S, uvS or v^2S.

It should be immediately obvious that we have a binomial situation here, and this is shown diagrammatically in Figure 7-2, the resulting tree structure. Combinatorics tells us that there are $\binom{n}{r}$ possible paths to the asset value $u^r v^{n-r} S$ at time-step $t = n\,\delta t$.

Thus, as there are more paths to the middle nodes in the tree, it is more likely that any particular asset evolution will attain this non-extremal values. In this way, the binomial tree gives results which approximate those obtained from using the gaussian density function for the lognormal asset walk.

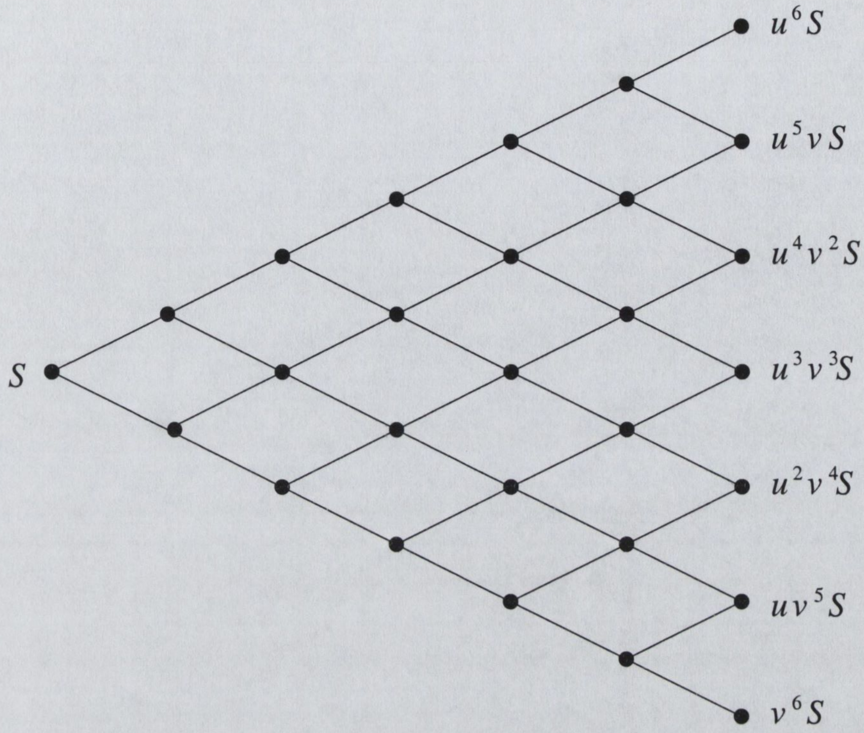

Figure 7-2: The Binomial Tree

7.3 The Binomial Model and Option Pricing

We now need to show how we can use the Binomial Model to price an option.

The basic concept is that we know the value of the option at expiry, and so develop an algorithm that allows us to calculate the value of the option at time t using the value at time $t+\delta t$ i.e. we calculate backwards in time. We can do this at every level of the tree, eventually giving us a value at the root node, which is the option price V.

The reader should note that the argument below is similar to that of the derivation of the Black-Scholes equation given in Chapter 1.

Suppose then that we know the value of the option at $t + \delta t$.

Consider a portfolio consisting of an option and a short position in a quantity Δ of the underlying asset. Then, at time t this portfolio has the value

$$\Pi = V - \Delta S, \tag{7.8}$$

where V is unknown.

It follows that the value of the portfolio at time $t + \delta t$ will be either $V^+ - \Delta u S$ or $V^- - \Delta v S$.

Our choice of model determines u and v, the asset price S is known, and we have made no assumptions about the quantity Δ and so it is arbitrary at this point. Finally, we know the value of the option at time $t + \delta t$, and so V^+ and V^- are also known. Thus, both the above values are known and depend on the value Δ.

We constrain the value of Δ by choosing that the value of the portfolio should be independent of movement in the asset price, giving us

$$V^+ - \Delta u S = V^- - \Delta v S. \tag{7.9}$$

giving us

$$\Delta = \frac{V^+ - V^-}{(u - v)S}. \tag{7.10}$$

The value of the portfolio at time-step $t + \delta t$ is

$$\Pi + \delta\Pi = V^+ - \frac{u(V^+ - V^-)}{(u - v)} = V^- - \frac{v(V^+ - V^-)}{(u - v)}. \tag{7.11}$$

Since the value of the portfolio is independent of movements in the underlying asset, the no arbitrage condition means that its value must be equal to the value of the original portfolio along with any interest earned at a risk-free rate, so

$$\delta\Pi = r\Pi\,\delta t. \tag{7.12}$$

After some manipulation this equation becomes

$$V = \frac{V^+ - V^-}{u - v} + \frac{uV^- - vV^+}{(1 + r\,\delta t)(u - v)}. \tag{7.13}$$

This equation can be approximated to the first order as

$$e^{r\,\delta t}V = p'V^+ + (1 - p')V^-, \tag{7.14}$$

where

$$p' = \frac{e^{r\,\delta t} - v}{u - v}. \tag{7.15}$$

7.3.1 Binomial Model Calculation

The algorithm for calculating an option price using the binomial method is straightforward and is shown in Algorithm 7-1. The algorithm requires two trees of equal size, one to hold the asset values and one to calculate the corresponding option prices:

Algorithm 7-1 The Binomial Model

Phase 1: Filling the Asset Tree

1 – At the root node, the asset value is simply the starting price for the asset S.

2 – Calculate the two child nodes using the given values for u and v in Eq. (7.7).

$$\begin{aligned} u &= \frac{1}{2}(e^{-\mu\,\delta t} + e^{(\mu+\sigma^2)\,\delta t}) + \frac{1}{2}\sqrt{(e^{-\mu\,\delta t} + e^{(\mu+\sigma^2)\,\delta t})^2 - 4}, \\ v &= \frac{1}{u}. \end{aligned}$$

3 – Repeat for all the successive child nodes until the tree has been filled.

Phase 2: Calculating the Option Prices

1 – Fill in the option prices for bottom tree level using the payoff function of the option and the corresponding asset values given in the asset tree.

2 – Using Eq. (7.13):

$$V = \frac{V^+ - V^-}{u - v} + \frac{uV^- - vV^+}{(1 + r\,\delta t)(u - v)},$$

and the two child values on the tree (V^+ and V^- in the above formula), calculate the option prices on the parent nodes.

3 – Repeat from all higher parent nodes, the calculated value at the root node of the tree is the calculated option price.

7.4 Tree Methods and American Options

7.4.1 Converging Bounds Method

This method was first developed by Broadie and Glasserman[5].

It uses two tree/node asset price structures very similar to those used in the binomial method for pricing options to calculate an upper and lower bound for the American option, which converge as the accuracy of the algorithm is increased.

Algorithm 7-2 The Converging Bounds Tree Algorithm - Building the Asset Tree

---

Building the Asset Tree

1 – The root-node of the tree contains the initial stock value S_0.

2 – Generate b values for each branch, using the stochastic formula

$$S_{i+1} = S_i \exp\left(r - \frac{1}{2}\sigma^2\right) dt + (\sigma\phi\sqrt{dt})$$

where ϕ is a random variable drawn for a normalised Gaussian distribution, and $N\,dt = T$, where N is the number of time-steps taken throughout the option lifetime.

3 – Repeat this process at all the children nodes just created, and repeat until the whole tree has been filled.

Algorithm 7-3 The Converging Bounds Tree Algorithm

Calculating the High Estimate

1 – Working from the $t = T$ nodes, the value at each leaf-node is defined to be the exercise value, using the asset prices from the corresponding node in the asset tree.

2 – Working backwards, the value at each node is the maximum of the exercise value (using the corresponding asset price from the asset tree) and the average of the discounted expected payoffs from all the child-nodes.

3 – Work back to the root node.

4 – The high estimate is the calculated value at the root node.

Calculating the Low Estimate

1 – Working from the $t = T$ nodes, the value at each leaf-node is defined to be the exercise value, using the asset prices from the corresponding node in the asset tree.

2 – Working backwards, the estimate at the node is calculated as follows:

a) Assume each branch is labelled, B_i, with i going from 1 to b.

b) At B_1, we decide whether or not to exercise by calculating the expected payoff of the other $b - 1$ child nodes, B_2, B_3, ..., B_b.

c) If this expected payoff is larger than the exercise value, the value at the node is the continuation value calculated from the B_1 node, discounted for time.

d) Otherwise, the exercise value is used.

e) Repeat this calculation using B_2, B_3, ... B_b as the continuation branch.

f) Average the results.

3 – Work back to the root node.

4 – The low estimate is the calculated value at the root node.

Chapter 8

Conclusion

During the course of our research, we looked at different methods for solving the Black-Scholes equation for vanilla options, both European and American.

Our primary goal was to find solutions for the American problem, as this problem does not have any analytical solutions.

Our first method was to use finite difference schemes to solve the problem.

The use of the Implicit, Crank-Nicholson and Van Leer Flux-limiting schemes are known in numerical finance, and we tested two new schemes, the fitted Duffy scheme, and a fitted version of the Crank-Nicholson scheme, showing that they performed well in situations where the Implicit and Crank-Nicholson have known limitations, i.e. when the coefficient of diffusion is small and the problem becomes singly-perturbed.

Secondly, we looked at stochastic methods, specifically variations on the well-known MonteCarlo approach.

Two new algorithms were discussed: the Interpolated Expectation algorithm and the Iterative Ito algorithm.

The Interpolated Expectation algorithm has a slight high bias on the interpolation mesh calculation but the full algorithm produces results which agree with the other methods. It has the benefit that once it is used, subsequent simulations are much faster, as the intepolation can be reused in different simulations.

This interpolation mesh can also be used to approximate the Greeks directly, avoiding the messy methods normally required from MonteCarlo methods in these situations, where values on volumes around the calculation points are required (all of them requiring the same sequence of random numbers).

The mesh generation stage is computationally expensive. Each point on the mesh requires a separate simulation to calculate its value, and this is expensive on large meshes. It was shown that if this mesh can be reused, substantial improvements can be made to the performance of this algorithm.

Overall though, the fact that this algorithm does not converge on the solution indicated by the Longstaff-Schwartz method and the finite difference solution indicates that this algorithm is in need of improvement. Also, the initial generation of the interpolation mesh can take some time, and so the algorithm is best used when a number of prices are required at different stock prices.

The iterative algorithm was shown to calculate approximate values for the Delta, Δ, and Gamma, Γ, and I find this approach encouraging.

The Iterative Ito algorithm itself produces results with a low bias. This is due to the inaccurate estimation of the Ito updates (dV), resulting in sub-optimal early exercise of the option. As the option is being exercised at a time that is sub-optimal, the calculated price is lower than it should be, as the payoff is less than is achievable with a more optimal early-exercise mechanism.

Again, quite a lot of work is required to make this algorithm competitive with existing methods for the calculation of American option prices.

Tree methods were also briefly discussed, but were primarily used as a reference for other solutions and no new research was done into these methods. A few techniques using this method were illustrated and described.

Overall, the finite difference schemes were the most successful at calculating solutions for the American options, both in terms of accuracy and computation time. However, our work was primarily based on looking at vanilla options, which have only a single space dimension S and time dimension t.

MonteCarlo methods become much more useful for calculating the values of more exotic options, where the payoff is path-dependant and involves multiple variables.

In these cases, the use of finite differencing can quickly become expensive, both in terms of memory requirements and computation times, since these increase exponentially as more dimensions are added to the problem space.

Also, when trying to calculate the value of portfolios, which involve a number of different financial derivatives, the finite difference approach quickly becomes unusable, due the the high-dimensionality of the resulting problem. In this case, Monte Carlo methods really come to the fore as they are much better equipped to deal with the problems introduced by having multiple financial derivates that need to be priced.

8.1 Further Work

A number of problems and further work came to light as a result of this work:

- For the finite difference approach, the requirement of having a boundary condition at $S \to \infty$ means that when using a uniform mesh, large areas of the solution mesh are wasted. The use of non-uniform meshes would probably suit the problem better. In particular, I would like to take a look at the possibility of using a Shishkin meshes on this problem.

- For the finite difference approach, only vanilla options were researched. Asian options, which depend on averages of the stock price during the lifetime of the option, are very common in finance and need to be looked at, especially for both fitted schemes, the fitted Duffy Implicit scheme and the fitted Crank-Nicholson.

- The Interpolated Expectation algorithm spends most of the time generating the mesh of interpolation data. Our algorithms used 100 and 200 interpolation points, but it is worth seeing how small the mesh can be and still produce reasonable results, as this would greatly reduce execution times.

- For the stochastic approach, only linear interpolation was used in the Interpolated Expectation algorithm. It would be worthwhile investigating the use of cubic spline interpolation, or some similar higher order interpolation that uses more of the data points to calculate the values.

- The Iterative Ito algorithm cannot calculate a good approximation for Theta, Θ, and this needs to be addressed.

- One of the underlying assumptions of the Black-Scholes model is that the asset price moves according to Brownian motion (i.e. the underlying random process obeys a Normal distribution). Many studies have proved that this is false, using numerous approaches. One of the reasons for using the Iterative Ito algorithm was that it uses Ito's Lemma directly. We would wish to look at non-gaussian distribution and try to calculate option prices using these distributions, as they more accurately reflect real-world movements.

Appendix A

Option Pricing Problems

A.1 Statement of the Pricing Problems

A.1.1 European Options

For European problems, we must find $V(S,t)$ satisfying

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \qquad \text{(A.1.1)}$$

subject to the boundary conditions:

$$\textbf{European Call} \begin{cases} V(S,T) & = \max(S-K,0) \\ V(0,t) & = 0 \\ V(S,t) & \to S - Ke^{-r(T-t)} \text{ as } S \to \infty \end{cases} \qquad \text{(A.1.2)}$$

$$\textbf{European Put} \begin{cases} V(S,T) & = \max(K-S,0) \\ V(0,t) & = Ke^{-r(T-t)} \\ V(S,t) & \to 0 \text{ as } S \to \infty \end{cases} \qquad \text{(A.1.3)}$$

A.1.2 American Options

For American problems without dividends, the price of an American call is identical to that of a European Call. To find the value for an American Put, we find $P(S,t)$ satisfying

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS\frac{\partial P}{\partial S} - rP = 0 \tag{A.1.4}$$

subject to the conditions:

$$\textbf{American Put} \begin{cases} P(S,T) = \max(K - S, 0) \\ P(0,t) = Ke^{-r(T-t)} \\ P(S,t) \to 0 \text{ as } S \to \infty \\ P(S_f,t) = \max(K - S_f, 0) \\ \frac{\partial P}{\partial S}(S_f,t) = 1 \\ P(S,t) \geq \max(K - S, 0) \end{cases} \tag{A.1.5}$$

A.1.3 Barrier Options

The price function for a barrier option still satisfies the standard Black-Scholes PDE as in Eq. (A.1.1).

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

The effect of the barrier manifests in the boundary conditions, and depends on whether the option is an 'In' or 'Out' barrier option.

$$\textbf{Out Barrier} \begin{cases} V(S_u,t) & = 0 \text{ for } t < T \\ V(0,t) & \textit{same as for vanilla option} \\ V(S,T) & \textit{same as for vanilla option} \end{cases} \tag{A.1.6}$$

$$\textbf{In Barrier} \begin{cases} V(S_b,t) = V_v(S_b,t) & \text{barrier reached} \\ V(S,t) = 0 & \text{barrier not reached} \end{cases} \tag{A.1.7}$$

where S_b is the barrier value, and V_v is the price of a vanilla option of the same type as the barrier option.

A.1.4 Asian Options

The value of an Asian option is given by

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + f(S,t)\frac{\partial V}{\partial I} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{A.1.8}$$

where

$$I = \int_0^t f(S,\tau)\, d\tau \tag{A.1.9}$$

The boundary conditions depend on the details of the Asian option. Since we generally do not get conditions for I, we assume they are the same as for S.

Arithmetic Average Strike Call Options

The problem is to solve

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + S\frac{\partial C}{\partial I} + rS\frac{\partial C}{\partial S} - rC = 0, \tag{A.1.10}$$

where

$$\textbf{Arithmetic Average Strike Call} \begin{cases} C(S,I,T) = \max(S-A,0) \\ C(0,I,t) = 0 \\ C(S,0,t) = 0 \\ C(S,I,t) \to S - Ae^{-r(T-t)} \text{ as } S \to \infty \\ C(S,I,t) \to S - Ae^{-r(T-t)} \text{ as } I \to \infty \end{cases} \tag{A.1.11}$$

In the above equation A is defined to be the average value of I, so we have

$$A = \frac{I}{T}. \tag{A.1.12}$$

Geometric Average Strike Call Option

The problem is to solve

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \log S \frac{\partial C}{\partial I} + rS \frac{\partial C}{\partial S} - rC = 0 \tag{A.1.13}$$

where

$$\textbf{Geometric Average Strike Call} \begin{cases} C(S,I,T) = \max(S-A,0) \\ C(0,I,t) = 0 \\ C(S,0,t) = 0 \\ C(S,I,t) \to S - Ae^{-r(T-t)} \quad \text{as } S \to \infty \\ C(S,I,t) \to S - Ae^{-r(T-t)} \quad \text{as } I \to \infty \end{cases} \tag{A.1.14}$$

In the above equation A is defined to be the average value of I, so we have

$$A = \exp\left(\frac{I}{T}\right) \tag{A.1.15}$$

Arithmetic Average Rate Call Options

The problem is to solve

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + S \frac{\partial C}{\partial I} + rS \frac{\partial C}{\partial S} - rC = 0 \tag{A.1.16}$$

where

$$\textbf{Arithmetic Average Rate Call}\begin{cases} C(0,I,t) = \max(A-K,0) \\ C(S,0,T) = 0 \\ C(0,I,T) = 0 \\ C(S,I,t) \to S - Ae^{-r(T-t)} \quad \text{as } S \to \infty \\ C(S,I,t) \to S - Ae^{-r(T-t)} \quad \text{as } I \to \infty \end{cases} \tag{A.1.17}$$

Geometric Average Rate Call Options

The problem is to solve

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \log S\ \frac{\partial C}{\partial I} + rS\frac{\partial C}{\partial S} - rC = 0, \tag{A.1.18}$$

where

$$\textbf{Geometric Average Rate Call}\begin{cases} C(S,I,T) = \max(A-K,0) \\ C(S,0,T) = 0 \\ C(0,I,T) = 0 \\ C(S,I,t) \to S - Ae^{-r(T-t)} \quad \text{as } S \to \infty \\ C(S,I,t) \to S - Ae^{-r(T-t)} \quad \text{as } I \to \infty \end{cases} \tag{A.1.19}$$

A.1.5 Lookback Options

For a look-back option, we find $V(S,t)$ satisfying

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{A.1.20}$$

with the conditions

$$\textbf{Lookback Option} \begin{cases} V(S,J,T) = \text{Payoff}(S,J) \\ V(0,J,t) = 0 \\ \frac{\partial V}{\partial J}(J,J,t) = 0 \end{cases} \tag{A.1.21}$$

where J is defined by

$$J = \max_{0<\tau<t} S(\tau) \tag{A.1.22}$$

Appendix B

Glossary

A.1 Financial Instruments

This section explains the terminology used in this field, and defines the various types of financial instruments dealt with in this work.

A.1.1 Option Contracts

- Vanilla Options

Option Contract An *Option Contract* (or just *option*) is a financial contract which confers upon the holder the right, but not the obligation, to buy or sell a particular asset for an agreed price on an agreed date. The price and date are fixed and considered to be part of the contract.

Strike Price The *Strike Price* is the agreed price for the asset in an option contract.

Expiry Date The *Expiry Date* is the agreed date in the option contract when the right to buy or sell must be exercised.

Payoff The *payoff* of an option contract is the value of the option at exercise.

Call Option A *Call Option* is an option which confers the right to buy upon the holder.

Put Option A *Put Option* is an option which confers the right to sell upon the holder.

European Option A *European Option* is an option which may only be exercised on the expiry date.

American Option An *American Option* is an option which may be exercised throughout the lifetime of the contract until the expiry date.

- Asian Options

Path-dependent Option A *path-dependent* option is an option in which the payoff of the option is dependent on the history of the underlying asset during the lifetime of the option.

Asian Option An *asian option* is a path-dependent option in which the payoff at exercise is dependent on the history of the underlying price through an average of some kind.

Average Rate Asian Option An *average rate asian option* is an asian option in which the dependancy of the underlying in the option payoff is replaced with the average of the underlying value throughout the lifetime of the option.

Average Strike Asian Option An *average strike asian option* is an asian option in which the dependancy of the strike price in the option payoff is replaced with the average of the underlying value throughout the lifetime of the option.

Continuously Sampled Average A *continuously sampled average* is an average where the value being averaged is measured continuously throughout the averaging interval.

Discretely Sampled Average A *discretely sampled average* is an average where the value being averaged is measured at a finite number of points throughout the averaging interval.

- Barrier Options

Barrier Option A *Barrier Option* is a weakly path-dependent option whose value depends on whether or not the price of the underlying crossed some fixed point during the life-time of the option. In general, the actual payoff of the option is only dependent of the asset price at expiry.

Out Barrier Option An *Out barrier option* is an option which expires worthless if a defined barrier is reached.

In Barrier Option A *In barrier option* is an option which expires worthless if a defined barrier is not reached.

Up Barrier Option An *Up barrier option* is an option where the defined barrier is above the initial value of the underlying asset.

Down Barrier Option A *Down barrier option* is an option where the defined barrier is below the initial value of the underlying asset.

Barrier Option Rebate A *Rebate* on a barrier option is the amount paid to holder when a barrier is reached on a barrier option.

- Lookback Options

Lookback Option A *Lookback option* is a path-dependent option where the payoff of the option depends on the maximum or minimum value of the underlying asset attained during the lifetime of the option contract.

A.1.2 Bond Contracts

Bond Contract A *bond contract* is a contract, paid for up front, which pays a fixed amount at some date in the future. The amount and date of payment are fixed and considered part of the contract.

Maturity Date The *maturity date* of a bond contract is the date at which the seller of the bond must pay the holder of the bond the agreed amount in the contract.

Coupon A *coupon* is a known cash dividend paid out to holders bond contracts at fixed dates during the lifetime of the bond. Coupons are considered part of the bond contract.

Zero-Coupon Bond A *zero-coupon bond* is a bond contract which does not pay out any dividends during its lifetime.

Bibliography

[1] Louis Bachelier. Théorie de la spéculation. *Annales Scientifique de l'École Normale Supérieure*, 3(17):21–86, 1900.

[2] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 8(3):637–654, 1973.

[3] Richard Breen. The accelerated binomial pricing model. *Journal of Financial and Quantitative Analysis*, 26(2):153–164, June 1991.

[4] Michael J. Brennan and Eduardo S. Schwartz. The valuation of american put options. *Journal of Finance*, 32(2):449–462, May 1977.

[5] Mark Broadie and Paul Glasserman. Pricing american-style securities using simulation. *Journal of Economic Dynamics and Control*, 21:1323–1352, 1997.

[6] Robert Brooks. *Building Financial Derivatives Applications with C++*. Quorom Books, 2000.

[7] David S. Bunch and Herb Johnson. A simple and numerically efficient valuation method for american puts using a modified geske-johnson approach. *Journal of Finance*, 47(2):809–816, June 1992.

[8] John Y. Campbell, Andrew W. Lo, and A. Craig MacKinlay. *The Econometrics of Financial Markets.* Princeton University Press, 1997.

[9] John C. Cox and Mark Rubinstein. *Options Markets.* Prentice-Hall, 1985.

[10] Keith Cuthbertson. *Quantitative Financial Economics - Stocks, Bonds and Foreign Exchange.* Wiley, 1996.

[11] Daniel J. Duffy. *Uniformly convergent difference schemes for problems with a small parameter in the leading derivative.* PhD Thesis, Trinity College, Dublin, 1980.

[12] (editors) L. C. G. Rogers and D. Talay. *Numerical Methods in Finance.* Cambridge University Press, 1997.

[13] Robert J. Elliott. *Mathematics of Financial Markets.* Springer, 1999.

[14] P.A. Forsyth and K. R. Vetzal. Quadratic convergence for valuing american options using a penalty method. *SIAM Journal of Scientific Computing*, 23:2095–2122, 2002.

[15] George E. Forsythe and Wolfgang R. Wasow. *Finite-Difference Methods for Partial Differential Equations.* Wiley, 1960.

[16] Robert Geske and H. E. Johnson. The american put option valued analytically. *Journal of Finance*, 39(5):1511–1524, December 1984.

[17] Gene H. Golub and Charles F. van Loan. *Matrix Computations.* John Hopkins University Press, 1993.

[18] John C. Hull. *Options, Futures, and Other Derivatives, 3rd Edition.* Prentice-Hall, 1997.

[19] H. E. Johnson. An analytic approximation for the american put price. *Journal of Financial and Quantitative Analysis*, 18(1):141–148, March 1983.

[20] Richard J. Rendleman Jr. and Brit J. Bartter. Two-state option pricing. *Journal of Finance*, 34(5):1093–1110, December 1979.

[21] Ioannis Karatzas and Steven E. Shreve. *Methods of Mathematical Finance.* Springer, 1998.

[22] Ito Kiyoshi. On stochastic differential equations. *Memoirs, American Mathematical Society*, (4):1–51, 1951.

[23] Y.K. Kwok. *Mathematical Models of Financial Derivatives.* Springer, 1998.

[24] Francis A. Longstaff and Eduardo S. Schwartz. Valuing american options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14(1):113–147, Spring 2001.

[25] Robert C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

[26] Robert C. Merton. *Continuous Time Finance.* Basil Blackwell, 1990.

[27] G. H. Meyer. One-dimensional free-boundary problems. *SIAM Review*, 19(1):17–34, January 1977.

[28] Ravi Myneni. The pricing of the american option. *The Annals of Applied Probability*, 2(1):1–23, February 1992.

[29] Bernt Oksendal. *Stochastic Differential Equations: An Introduction with Applications.* Springer, 1998.

[30] Michael Parkinson. Option pricing: The american put. *Journal of Business*, 50(1):21–36, January 1977.

[31] Roger Peyret and Thomas D. Taylor. *Computational Methods for Fluid Flow.* Springer, 1983.

[32] William H Press, William T Vetterling, Saul A Teukolsky, and Brian P Flannery. *Numerical Recipes in C: The Art of Scientific Computing, 2nd Edition.* Cambridge University Press, 1992.

[33] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numerische Mathematik*, 43:309–327, 1984.

[34] G. D. Smith. *Numerical Solution of partial differential equations: Finite Difference methods, 3rd Edition.* Clarendon Press, 1985.

[35] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* Wadsworth and Brooks/Cole, 1989.

[36] Domingo Tavella and Curt Randall. *Pricing Financial Instruments: The Finite Difference Method.* Wiley, 2000.

[37] Michael E. Taylor. *Partial Differential Equations: Basic Theory.* Springer, 1996.

[38] J. W. Thomas. *Numerical Partial Differential Equations.* Springer, 1995.

[39] various. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, 1994.

[40] Paul Wilmott. *Derivatives: The Theory and Practice of Financial Engineering, University Edition.* Wiley, 1998.

[41] Paul Wilmott, Jeff Dewynne, and Sam Howison. *Option Pricing: Mathematical models and computation.* Oxford Financial Press, 1993.

[42] Erich Zauderer. *Partial Differential Equations of Applied Mathematics, 2nd Edition.* Wiley, 1989.

[43] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Robust numerical methods for pde models of asian options. *Journal of Computational Finance*, 1(2):39–78, 1998.