Aalto University
School of Business

Numerical Methods in Real Option Analysis

Economics

Master's thesis

Arto Sorsimo

2015

Department of Economics
Aalto University
School of Business

Aalto University
School of Business

Aalto University
P.O. BOX 11000, 00076 AALTO
www.aalto.fi

ABSTRACT OF MASTER'S THESIS

Author Arto Sorsimo	
Title of thesis Numerical Methods in Real Option Analysis	
Degree Master of Science in Economics and Business Administration	
Degree programme Economics	
Thesis advisor Associate Professor Pauli Murto	**Pages** v + 82
Year of approval 2015	**Language** English

In this study we examine different numerical solution methods that can be used to solve differential equations arising from real options analysis and present two case studies that are solved numerically.

First we examine commonly used methods in valuating investments with uncertainty. The most suitable method for long-term investments with high uncertainty is the real options analysis, which uses an underlying stochastic variable in valuation.

We introduce the framework for real options and examine the differences between infinite and finite time horizon real options. Short literature review reveals that there are several problems within real options theory for which a closed-form solution does not exist and hence numerical methods should be applied. We introduce three numerical methods commonly used in real options analysis: the Monte Carlo (MC) method, binomial lattice (BL) method, and finite difference method (FDM) with explicit and implicit solution scheme. Then we present two case studies, investment option that is used to benchmark numerical solutions, and abandonment option which cannot be solved analytically.

Comparison of numerical methods reveals that even though the MC method is stable, it is inaccurate and slow in comparison to other methods. The implicit FDM is superior to the explicit method as the latter is very unstable to grid parameters. Even though the BL method outperforms other methods with respect to simulation time and accuracy, the implicit FDM is the most advantageous method as it provides always convergent solution in the whole time domain at once. Finally, we apply BL method and FDM to solve the abandonment option case the option to abandon can be exercised at any point of time during the project.

On the grounds of the study, we suggest using the implicit FDM in further real option applications due to the output, convergence and stability properties, and the flexibility over the boundary conditions. We recommend investigating additional case studies with the presented numerical methods along with their extensions, as well as completely new approaches such as the finite element method.

Keywords Real options, finite time horizon, abandonment option, numerical methods, Monte Carlo, binomial lattice, finite difference method.

Acknowledgements

First, I would like to thank my thesis advisor Pauli for helpful comments and insights on my work, which helped me to complete the thesis within a tight schedule.

Studying and working effectively is very difficult without taking a break once in a while. Thus, special thanks to my dearest friends at Kauppis – David, Timo and Roope – for enjoyable conversations on economics and life, and for all the fun moments outside the studies. I would also like to show my gratitude for all my friends that have helped me to balance life each in their own way.

Finally, but not certainly least, I would like to thank my loving parents for their everlasting support and encouragement.

Helsinki, April 2, 2015

Arto Sorsimo

Contents

Chapter 1

Introduction

"A ship is safe in harbor, but that's not what ships are for."
-William G.T. Shedd

From the beginning of the human history, people have always taken risks. Relocating a tribe from one place to another in order to find food and hunting dangerous animals have been daily decisions in the prehistoric era. The decisions at that time have been most probably based on historic data and gut feeling. This immeasurable justification on decisions stayed mostly the same also on the historic era until the concept of probability was invented[1].

While the gamblers that lived in Ancient Greece had the concept of numerals and could determine the number of possible outcomes, they strongly believed that the outcome of games was determined by gods. Concept of modern arithmetic, i.e. numbers and symbols, came from Hindus during the Dark Ages and made the analysis of games possible in the 16th century. Italian mathematician Geralamo Gardano (1501–1576) was the first one to determine the theoretical probability correctly by dividing the number of required outcomes with the number of possible outcomes. However, Gardano's ideas were not that rigorously presented or proven. Famous scientists such as Galileo Galilei (1581–1585), Blaise Pascal (1623–1662) and Pierre de Fermat (1607–1665) studied and extended Gardano's ideas and finally Christianus Huygens (1629–1695) published a mathematical formulation of expectations and probabilities. More extensive and in-depth analysis was shortly published in Abraham de Moivre's (1667–1754) famous book The Doctrine of Chances, which is considered to be the cornerstone of probability. This led

[1]Naturally there has been some common rules, i.e. strategies, for games played before the emergence of probability theory, e.g. Roman emperor Claudius (10 BC – 54 AD) wrote a book on how to win at dice, that didn't unfortunately survive to this date [18].

to explosion of mathematical texts on probability and consequently to the birth of probability theory as a branch of mathematics. [18][2]

From these days forwards, the concept of risk became measurable. Part of the decisions under a risk could be validated through mathematics and thus the dice seemed to be more favorable for those enlightened of the underlying mathematics. This competitive edge has intrigued every decision maker ever since the time of the invention. Naturally it has also been a great interest for companies to this date in their mission of seeking endless profits.

1.1 Decisons under uncertainty

Companies face several tough decisions throughout their existence. Some of them might be easier, such as firing a bad employee, but many decisions can essentially determine the fate of a company. The most crucial decisions are usually investments, which have visible and direct effect on company's balance sheet. In addition, the difficulty of investment decision usually increases as the uncertainty on the outcome increases. Some examples of difficult questions that are typically asked before initiating a long investment project are: should we invest into project A or B, when we will see some results from the project, what will be the market demand after the project is complete, when the first competitors will arrive to the designated market, and should we exercise the patent before that? Before answering to these questions, let us take a step back and consider what are some of the frameworks companies use when making investment decisions.

Consider a company that has an option to invest into a project. Aside from underlying strategic aspects, the company should choose the project for which the expected return is the highest. One commonly used metric is return on investment (ROI), which is simply calculated by dividing the net profit from the investment with the cost of investment. However, the problem with ROI and similar fixed metrics is that they do not take account the time value of the money as the net present value is not contained in the metric. This arises serious problems when the time range of the investment is longer then one year.

If the project duration and consequently the payback time is long, one might consider using discounted cash flow analysis (DCF), which takes account the time value of the money. In DCF the possible future cash flows are discounted with chosen interest rate to the present value. DCF is very popular method throughout the industries and especially in the finance sector but

[2]In this study, a citation mark outside the dot at the end of the paragraph denotes that the citation refers to the whole paragraph.

it has several drawbacks. First of all, the main parameters, the interest rate and cash flows, are static estimates and thus vulnerable to bias. In addition, the underlying parameters in the model are similarly static estimates relying on highly questionable and sensitive models, which decreases the credibility of the model, cf. Figure 1.1 for a representation of the weaknesses.

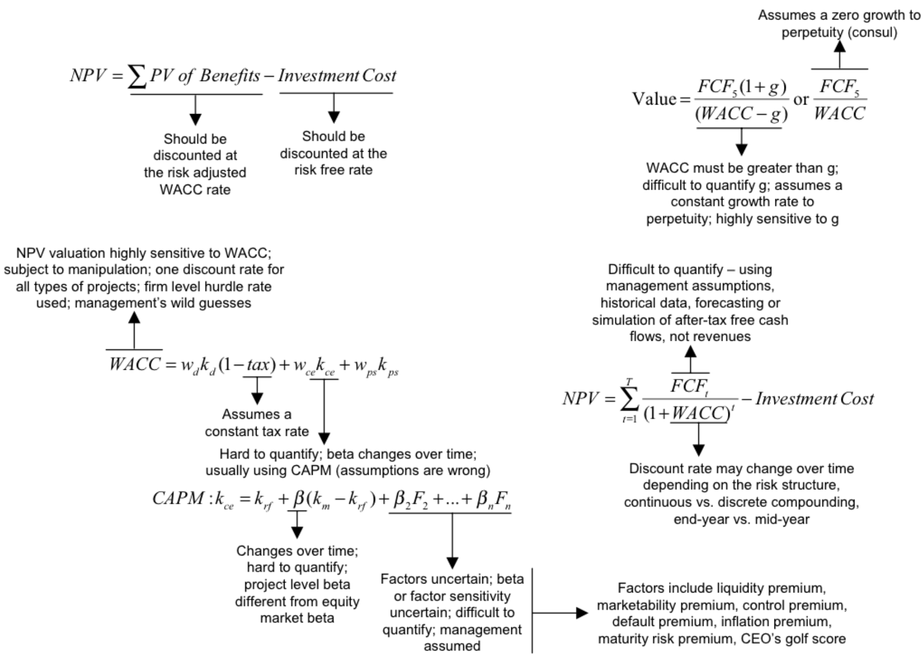


Figure 1.1: Some drawbacks of parameters with the discounted cash flow analysis [55].

Moreover, the general decision rule for DCF is that the option to invest should be executed *now* if the net present value is positive. All the possible projects are not that straightforward as they might consist several different phases or there might be an underlying uncertainty within the possible outcomes.

Typically used method to model such projects and to determine the strategic decisions is decision tree analysis, which is illustrated in Figure 1.2. The main idea is to determine the probabilities on all the possible outcomes branching from a single event and calculate the expected final outcomes. Hence a validation for a decision can be derived from the final outcomes, which are usually classified to different scenarios, such as optimistic, neutral and pessimistic.

Decision tree analysis is widely used in decision-making as it is a simple and understandable framework suitable in multiple situations. However, as

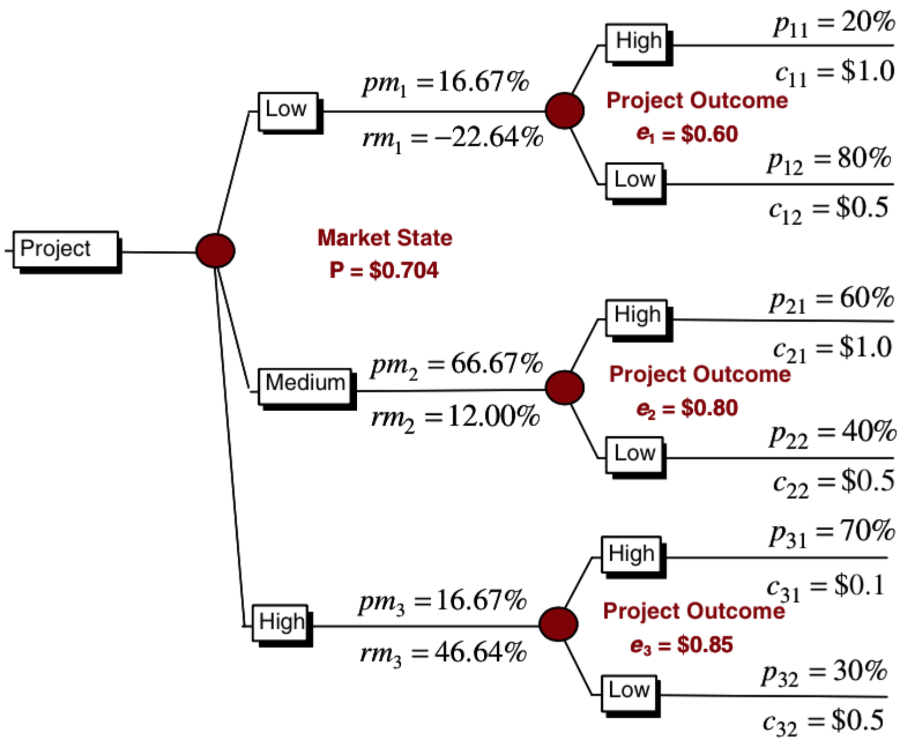


Figure 1.2: Example of a decision tree analysis scheme applied to product development project with three possible market states and two possible outcomes for cash flow [19].

most of the models are extremely simple, they lack many properties, such as modeling stochastic processes of continuous time and, once again, the option to delay the investment.

Decision-makers throughout the world use the frameworks presented above nearly everyday, even though the frameworks lack the option for delaying. Possible reasons for the wide usage could be their simplicity, wide acceptance within the business scene and assumed suitability to a given situation. While some or even all of these reasons might be true for some cases, there are still several cases where the frameworks presented are just highly unusable.

Moreover, the economy has changed dramatically over the last few decades. Prior to the globalization and Internet Revolution there were a handful of investment options and the risk of expanding to new areas was relatively low. In the current business world the companies are surrounded with a blinding amount of different investment options to choose from. Furthermore, the underlying risk has been increasing due to fierce, global competition.

This paradigm shift in business environment has arisen the need for new approaches when making investment decisions under uncertainty. One possible and highly embraced remedy is real options analysis.

1.2 Real options analysis

Real options analysis is a study of different kind of options that a business can take in its endeavors. The "real" word signals that real options are mainly focused on options on tangible assets, contrary to financial options. To put it short, real option analysis is extension of financial option theory to real assets. The main idea, described more technically, is that the value of the real option is defined as a function of a stochastic variable, and by using some common tools of stochastic calculus, the value of the option can be determined from a differential equation.

1.2.1 History of real options

Staring point for real options analysis emerged from the notorious Black-Scholes[3] equation that revolutionized the Wall Street, both positively and negatively. In 1973 Fischer Black and Myron Scholes published a famous paper that described a theoretical valuation formula for options. The main idea of the paper was to derive the price of an option by delta hedging, that is, mitigating the risk by taking both the long and short position of the underlying stock [7]. During the same year, Robert C. Merton completed and extended the mathematical theory behind the Black-Scholes by deriving the equation with a "replication method" [53]. Both Scholes and Merton received a Nobel prize in 1997 for their contribution, two years after Black deceased[4].

The consequences of the Black-Scholes equation were tremendous – bankers on the Wall Street could construct nearly arbitrary exotic options with the Black-Scholes equation and offer them to several different customer, ranging from gambling individuals to companies hedging their mainstream of revenues against market risks. Furthermore, it gave ideas and concepts for fields outside financial industry.

---

[3]Also sometimes referred as Black-Scholes-Merton equation to honor the work done by Robert C. Merton.

[4]Coincidence or not? One reasonable argument for the latter is that the Nobel prize committee was reluctant to give a prize to someone working in financial industry – Black left the academics in 1984 and joined investment bank Goldman Sachs.

Dan Galai and Ronald W. Masuilis were the first ones to suggest using option pricing in corporate investment decisions in 1976 [29]. One year later Stewart C. Myers published a paper where he discussed using the concepts of call options with corporate assets and referred them as "real options" [56]. In 1983 Paddock, Siegel and Smith introduced an option pricing method for real assets, using an offshore petroleum lease as an example [60]. During the same year Myers and Saman Majd presented a model for abandoning a project by using the similarities found in American put option [58]. Several other applications and extensions for real options were arising in the mid 1980s and simultaneously the criticism over the other methods accelerated.

In 1984 Myers pointed out several inconsistencies in use of discounted cash flow analysis in strategic planning and applications [57] and emphasized the positive aspects of using real options within corporate finance. The main statement was that DCF techniques, that were widely used at that time, underestimate the option value and does not work well with businesses with high growth opportunities or intangible assets. Further inconsistencies with the DCF technique were pointed out in 1985 by Hodder et al. [36], arguing that the method is shortsighted and produces excessive risk aversion on biased perceptions mainly due to the fixed discount rate over long period of time. Trigeorgis et al. [73] joined the chorus two years later underlining the lack of flexibility on decision time for DCF techniques which ultimately produces biased results. This insight brought upon the fact that information itself can actually have a real, measurable value.

The pace of uprising number of applications for real options and criticism for commonly used methods inspired part of the industry to use real options as a valuation method. Consequently real options analysis rose from the academic circles to the everyday use of practitioners in the early 1990s, and the interest towards the method has been increasing ever since. This is not a tremendous shock, as the real options analysis is highly applicable in many different fields of industry.

1.2.2 Applications for real options

Naturally, there exists a massive number of articles on real options to this date and consequently different applications can be classified with multiple ways. Lander et al. [45] discussed the challenges of practical implementation for real options in an article published in 1998. The classification among different areas where real options have been applied were the following; natural resources, competition and corporate strategies, manufacturing, real estate, international, R&D, regulated firms and utilities, M&A and corporate governance, interest rates, inventory, labor force, venture capital, advertising, law, environmental compliance and conservation. We note that the number of different applications is vast, even though the first educational book that deals exclusively with real options was published in 1994 by Dixit and Pindyck [22].

Following more recent discussion [55], some industries where real options analysis has been used successfully applied are automobile and manufacturing, computer, airline, oil and gas, telecommunications, utilities, real estate, pharmaceutical and high-tech industry. More specific examples within these industry sectors are General Motors Company's usage of material switching options with different vendors in producing new cars, and the application of growth option by Sprint Corporation to justify the enormous investments into telecommunication infrastructure for which the technology did not yet exist at the time.

From the long list of different applications, we will discuss the energy and R&D sector more in detail as they provide interesting problems for further examination.

1.2.2.1 Energy sector

The energy sector has experienced large economic shifts from the 1970s to this date. The changes are mainly due to the significant technological and regulatory changes within the sector. Overall, the sector has transformed from highly regulated and monopolistic sector to deregulated and highly competitive sector with high uncertainty. This has led to inaccurate valuations with traditional capital budgeting methods such as the net present value method. Consequently, the search of other methods that provide an option to wait, such as real options theory have gained ground. [3]

One of the first applications of real options theory to the energy sector was the research done in 1979 by Tourinho [72], where a natural reserve was valuated with uncertainty in the future price of the resource. Similar applications with some extensions were made in 1985 by Brennen and Schwartz [13] on the decision whether to open or close a copper mine with uncertainty in the price of copper. Applications within the oil industry were published few years later by Siegel et al. [67] on the valuation of offshore oil properties and Paddock et al. [60] on valuating offshore petroleum leases, where in both the price of oil was uncertain.

Most of the research presented by the end of the 1980s were considered as extensions of the financial theory. This reasoning was expected to be even more rational as the electric utility industry was becoming progressively deregulated in the mid 1990s [25]. Several articles were published on the real options related to the different types of financial options on the electricity market, such as [38], [20] and later [1]. Further applications of real options theory can be found e.g. within the field of power generation and environmental policy, such as optimizing the usage of Brazilian power plants [51] and analyzing the effects of emission regulation policy in Finland [47].

1.2.2.2 R&D sector

The investment problems within the R&D sector suit perfectly into the real options framework as the investment period is typically long and uncertainty is high. Examples of possible real option models applied to R&D are optimal timing and amount of investment, sequential choice over continuation and abandonment, and option to exercise a project through a patent. In other words, a decision-maker may seek an answer from the real options theory to questions such as when and how much should one invest into R&D projects, at what level one should abandon a sinking R&D project, and should one patent a product or not in order to keep the competition away?

As most of the applications of the real options theory within the R&D sector usually deal with an uncertainty in an investment project, we will focus our discussion on the different methods and possible challenges, contrast to the discussion of the energy sector where different applications were examined.

The uncertain variable, i.e. the stochastic variable, is typically the value [34], [21] or cost [65], [62] of the R&D project. Further complexity to a model can be introduced by including additional stochastic variables, such as assuming that the success of the project is probabilistic and the value of the patent is stochastic [75].

The stochastic variables in real option models can be either static or dynamic. In static models the parametric values of the stochastic variables are constant over the time while in the dynamic models the parameters can change over time, e.g. due to updated beliefs or competitive actions. Examples of these dynamic multiple-stage model are [30] and [66].

One of the main challenges with real R&D options is the choice of parameters. Most authors simply set some parameter, e.g. volatility of a R&D project, that seems reasonable but may not have been that fully verified. Instead of assigning a nearly arbitrary value as a parameter, one can seek some validation from the historical data, such as comparing the stock price with the company's announcements on R&D breakthroughs. However, a suitable value for a parameter might not always be available as companies tend to exclude the full specifications of R&D investments from the public. Needless to say, the choice of parameter is dependent on the model examined but always highly relevant. Many authors have pointed out that the determination of valid parameters is highly important especially in the field of R&D [61].

1.3 Motivation for the study

As noted from the brief outlook on the history of the real option analysis, the field is still remarkably young in comparison to other fields within science and economics. The number of articles has risen progressively year by year since the early 1990s. There are still several applications yet to be covered and also improvements to be made within the existing applications.

One of the main challenges within the real option analysis is that as the models become more complex, finding a closed-form solution to a differential equation describing the model becomes even more difficult. However, before delving into these issues, we need to define real options mathematically.

Chapter 2

Theory

In this chapter we examine the mathematical theory behind real option models. First we introduce real options models starting from a simple deterministic case without any uncertainty. Then we extend the basic model to stochastic case and examine real options with finite and infinite decision horizon. As we will notice, there is a vast difference between these two settings. In deriving the models, we mostly follow [22], which we suggest to refer for further details on real option models. Finally, we examine different cases within real options theory that lead to problems for which a closed-form solution does not exist.

2.1 Infinite horizon investment

Suppose that a decision-maker has an opportunity to invest into a project for which the value $V(x,t)$ is dependent on some stochastic variable x over the time period $t \in [0,\infty)$. Let $W(t)$ be a Wiener process. We assume that the stochastic variable x follows geometric Brownian motion with the increment $dW(t)$ and that the value of the project is determined from the equation

$$dx(t) = \alpha x(t)dt + \sigma x(t)dW(t), \tag{2.1}$$

where α, σ are some constant parameters of the model.

Few remarks about the stochastic variable we introduced. As $x(t)$ follows Brownian motion, it is described mathematically by the Wiener process[1]. Thus for $x(t)$ that follows a Wiener process has three crucial properties. First, the Wiener process is a Markov process, which implies that the probability

[1] *Wiener process.* Then it has the following properties; $W(0) = 0$, the mapping $t \to W(t)$ is almost surely continuous, and it has independent increments for which $W_t - W_s \sim N(0, t-s)$ for $0 \le s \le t$.

distribution for all future values of the process are independent on the history of values. Second, the increments of the Wiener process are independent. Third, the changes in the process are normally distributed over arbitrary finite interval of time. Consequently, the increment of a Wiener process can be described as a function of time t and it is given as

$$dW(t) = \epsilon\sqrt{dt}, \tag{2.2}$$

where ϵ is a normally distributed random variable with the properties $\epsilon \sim N(0, 1)$.

Now suppose that there is a investment cost I involved. Then the problem that a decision-maker has is to maximize the value of the investment opportunity, that is

$$V(x, t) = \max \mathbb{E}\left[(x(t) - I)e^{-\rho t}\right], \tag{2.3}$$

where the payoff for the investment, $x_t - I$, is discounted to the present value with a discount rate ρ from the time t when the investment will be exercised. Note that we must assume that $\alpha < \rho$ as otherwise the value of the project would grow indefinitely larger as the time t advances. Therefore we define a variable $\delta := \rho - \alpha > 0$, that we will use later in the discussion.

2.1.0.3 Deterministic case

We start by deriving the value of the investment in the deterministic case, that is, assuming that there is no uncertainty. Hence we set $\sigma = 0$ and the stochastic variable is given as

$$dx = \alpha x(t)dt \;\Rightarrow\; x(t) = x_0 e^{\alpha t},$$

where $x_0 = x(0)$. From this follows that the value of the investment opportunity is given as

$$V(x^*, t) = (x_0 e^{\alpha t} - I)e^{-\rho t}. \tag{2.4}$$

Note that Equation 2.4 is still dependent on the values of parameters α and ρ. Suppose first that $\alpha \leq 0$. Then $x(t) = x_0 e^{\alpha t}$ is decreasing or constant as the time passes and thus one should invest immediately if $x_0 e^{\alpha t} > I$. Hence the solution for the case $\alpha \leq 0$ is the following:

$$V(x^*, t) = \max\{x_0 e^{\alpha t} - I, 0\}.$$

Suppose then that $0 < \alpha < \rho$. In this case there might be a point of time when it is more optimal to invest than in the beginning. To obtain the optimal point of time, we differentiate Equation 2.4 to obtain

$$\frac{dV(x^*, t)}{dt} = (\alpha - \rho)x_0 e^{(\alpha-\rho)t} + \rho I e^{-\rho t} = 0$$

and hence

$$t = \frac{1}{\alpha}\ln\frac{\rho I}{(\rho-\alpha)x_0} \;\Rightarrow\; t^* = \max\left\{\frac{1}{\alpha}\ln\frac{\rho I}{(\rho-\alpha)x_0}, 0\right\}, \tag{2.5}$$

which describes the optimal time for the investment. Clearly, since we had $\alpha > 0$, one should invest immediately if

$$\ln\frac{\rho I}{(\rho-\alpha)x_0} > 0 \;\Rightarrow\; \frac{\rho I}{\rho-\alpha} > x_0,$$

since Equation 2.5 gives $t^* = 0$. If $x_0 > \frac{\rho I}{\rho-\alpha}$, the immediate investment is not the best response as one should wait for another opportunity for which the value is derived by substituting Equation 2.5 into Equation 2.4. Hence we obtain the best response strategy for the investment opportunity when $0 < \alpha < \rho$ holds:

$$V(x^*, t^*) = \begin{cases} \left(\frac{I\alpha}{\rho-\alpha}\right)\left(\frac{(\rho-\alpha)x_0}{\rho I}\right)^{\frac{\rho}{\alpha}} & \text{if } x_0 \le \frac{\rho I}{\rho-\alpha} \\ x_0 - I & \text{if } x_0 > \frac{\rho I}{\rho-\alpha} \end{cases}$$

Next we derive the same case as above but with a positive stochastic component, that is $\sigma > 0$. There are essentially two ways to derive the solution, with dynamic programming or contingent claims. We will present the both methods to obtain the solution.

2.1.1 Solution by dynamic programming

The main idea behind dynamic programming[2] is to break down a larger problem into a set of smaller, overlapping subproblems that are more easily solvable and then construct the solution to the initial problem from these subproblems. Describing the idea within the context of mathematical optimization, it usually means that a function that is defined over all time periods t is divided into discrete steps Δt and the full solution is formed by solving the value of the function one time step at the time.

Let $\pi(x(t), t)$ be the rate of the profit flow from the investment. Hence the total profit is given by $\pi(x(t), t)\Delta t$ and the total discounting over a discrete

[2] The grandfather of the dynamic programming, Richard Bellman (1920-1984) coined the term in the 1950s while working for RAND Corporation to hide out the research content in his study on optimization problems – the term "programming" was more suitable within military purposes [5].

step is $\frac{1}{1+\rho\Delta t}$. Consequently the value of the investment at continuous time t is given as

$$V(x(t),t) = \max\left\{\pi(x(t),t) + \frac{1}{1+\rho}\mathbb{E}\left[V(x(t+1),t+1)\right]\right\}. \tag{2.6}$$

This is commonly referred as Bellman equation in continuous time. Then on every discrete time step Δt we have the following equation

$$V(x(t),t) = \max\left\{\pi(x(t),t)\Delta t + \frac{1}{1+\rho\Delta t}\mathbb{E}\left[V(x(t+1),t+\Delta t)|x(t)\right]\right\}$$

from which we obtain

$$\begin{aligned}\rho\Delta t V(x(t),t) &= \max\{\pi(x(t),t)(1+\rho\Delta t)\Delta t \\ &\qquad + \mathbb{E}\left[V(x(t+1),t+\Delta t) - V(x(t),t)\right]\} \\ &= \max\{\pi(x(t),t)(1+\rho\Delta t)\Delta t + \mathbb{E}\left[\Delta V(x(t),t)\right]\}.\end{aligned}$$

Dividing by Δt and letting $\Delta t \to 0$ results in

$$\rho F(x,t) = \max\left\{\pi(x,t) + \frac{1}{dt}\mathbb{E}\left[dV(x,t)\right]\right\},$$

where we have denoted $x := x(t)$ for simplicity.

Following the assumptions we set in the beginning of this example, there are no profits as the investment project generates cash flows only at the time when the investment is undertaken, i.e. $\pi(x,t) = 0$. Thus the Bellman equation reduces to

$$\rho V(x,t)dt = \mathbb{E}[dV(x,t)].$$

Note that x follows Brownian motion as assumed above and thus is given by Equation 2.1. Hence by using the Itô's Lemma[3], the properties of Wiener process (2.2), and the fact that $\mathbb{E}[dW(t)] = 0$, we obtain

$$\begin{aligned}\mathbb{E}[dV(x,t)] &= \mathbb{E}\left[\frac{\partial V(x,t)}{\partial x}(\alpha x dt + \sigma x dW(t)) + \frac{1}{2}\frac{\partial^2 V(x,t)}{\partial x^2}(\alpha x dt + \sigma x dW(t))^2 + \dots\right] \\ &= \mathbb{E}\left[\frac{\partial V(x,t)}{\partial x}\alpha x dt + \frac{\partial V(x,t)}{\partial x}\sigma x dW(t) + \frac{1}{2}\frac{\partial^2 V(x,t)}{\partial x^2}\sigma^2 x^2 dt\right] \\ &= \frac{\partial V(x,t)}{\partial x}\alpha x dt + \frac{1}{2}\frac{\partial^2 V(x,t)}{\partial x^2}\sigma^2 x^2 dt = \rho V(x,t)dt,\end{aligned}$$

[3] *Itô's Lemma.* Let $x(t)$ be a Itô's drift-diffusion process and let the derivative be defined as $dx(t) := a(x,t)dt + b(x,t)dW(t)$, where $W(t)$ is a Wiener process. Then the total derivative on arbitrary function $F(x,t)$ is given as $dF(x,t) = \frac{\partial F(x,t)}{\partial t}dt + \frac{\partial F(x,t)}{\partial x}dx + \frac{1}{2}\frac{\partial^2 F(x,t)}{\partial x^2}(dx)^2$.

where we assumed that $\frac{\partial V(x,t)}{\partial t} = 0$ since we are dealing with infinite time horizon case where the value of the project may remain constant for long periods of time. Diving by dt and substituting the definition $\delta = \rho - \alpha$, we obtain the following differential equation:

$$\frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} + (\rho - \delta)x\frac{\partial V(x,t)}{\partial x} - \rho V(x,t) = 0. \tag{2.7}$$

We note that there is no dependence on time t and thus we will denote $V(x) := V(x,t)$ for simplicity.

We note that the Equation 2.7 is a second order homogeneous nonlinear differential equation. To solve the equation we need some boundary conditions. One boundary condition arises from the properties of stochastic processes, i.e. if x hits zero, it will stay there due to the independent increments. Thus we have a boundary condition

$$V(0) = 0. \tag{2.8}$$

In addition, we have two optimality conditions for the solution:

$$V(x^*) = x^* - I, \tag{2.9}$$

$$\frac{dV(x^*)}{dx} = 1. \tag{2.10}$$

The first optimality condition (2.9) determines valid payoff at the optimal stopping point and the second (2.10) determines unique stopping point as other conditions would break the continuity condition or contradict the definition of optimal point. The latter is commonly referred as "smooth-pasting" condition, for further details, see [22].

Note that as we have no dependence on time, we are seeking a solution boundary where one should invest at all periods of time. The solution boundary for Equation 2.7 that satisfies the boundary and optimality conditions can be derived analytically. We make a sophisticated guess that the solution must take the form $V(x) = Ax^\beta$, where A is a constant. Substituting this into Equation 2.7 gives

$$\frac{1}{2}\sigma^2\beta(\beta - 1) + (\rho - \delta)\beta - \rho = 0$$

from which we obtain two possible values for β, given as

$$\beta_{1,2} = \frac{1}{2} - \frac{\rho - \delta}{\sigma^2} \pm \sqrt{(\frac{\rho - \delta}{\sigma^2} - \frac{1}{2})^2 + 2\frac{\rho}{\sigma^2}}.$$

Since the differential equation (2.7) is linear in the dependent variable and its derivatives, the general solution can be presented as a linear combination. Hence the solution can be written as

$$V(x) = A_1 x^{\beta_1} + A_2 x^{\beta_2}. \tag{2.11}$$

However, the first boundary condition (2.8) implies that $A_2 = 0$ as $\beta_2 < 0$ and thus the solution is in from

$$V(x) = A_1 x^{\beta_1}. \tag{2.12}$$

Substituting Equation 2.12 into other two boundary conditions (2.9, 2.10) we obtain the critical value for the stochastic variable x when one should invest, giving

$$x^* = \frac{\beta_1}{\beta_1 - 1} I,$$

and the value for constant

$$A_1 = \left(\frac{\beta_1 - 1}{\beta_1 I}\right)^{\beta_1} \frac{I}{\beta_1 - 1}, \tag{2.13}$$

where

$$\beta_1 = \frac{1}{2} - \frac{\rho - \delta}{\sigma^2} + \sqrt{\left(\frac{\rho - \delta}{\sigma^2} - \frac{1}{2}\right)^2 + 2\frac{\rho}{\sigma^2}}.$$

Hence we have derived an analytical solution to the investment problem with uncertainty. Using some reasonable values as constants gives us the value of the investment as a function of the underlying stochastic variable x. Thus we may for example investigate how the investment opportunity is dependent on the parameter values with a sensitivity analysis.

However, there is one problem with the solution derived with the dynamic programming. The discount rate ρ that we assumed to be constant, is up to the decision-makers to decide without greater justification. Next we will derive the solution with contingent claims analysis, which will exempt the decision-maker from setting a value for the discount rate.

2.1.2 Solution by contingent claims analysis

The crucial assumption that differs the contingent claims analysis from dynamic programming is that the stochastic variable x must be spanned by a variable in the economy. This is essential assumption of the contingent claims analysis, implying that there must exist a variable within the economy that can be used to replicate the properties of the stochastic variable.

Note that the assumption is very convenient in some cases, e.g. when using publicly traded commodities such as price of oil and electricity as the stochastic variable. However, in some cases such as modeling an innovation rate as a stochastic variable, it may be rather hard to find a replicate for the stochastic variable within an economy.

Let $\hat{x}$ be a replicate for the stochastic variable x. We assume that the replicate $\hat{x}$ is perfectly correlated with the stochastic variable x, and that the return of the replicate variable, $r_{\hat{x}}$, is correlated with the return of the market portfolio r_m with some value $corr(r_{\hat{x}}, r_m)$. Hence the movements of $\hat{x}$ are given as

$$d\hat{x} = \mu\hat{x}dt + \sigma\hat{x}dW(t),$$

where μ is the drift rate that determines the expected rate of return for the stochastic replicate variable. According to the capital asset pricing model, this parameter should reflect the asset's systematic, nondiversifiable risk. Hence the drift rate can be defined as

$$\mu = r + \frac{r_m - r}{\sigma_m} corr(r_{\hat{x}}, r_m)\sigma,$$

where σ_m is the standard deviation of the market, and r is the risk-free interest rate. The interpretation of the drift rate μ is that it determines the rate of return for the project that investors would require. We assume that the expected percentage rate of change of x, given by α, is less than the risk-adjusted return μ, that is $\alpha < \mu$. Otherwise an investor would always rather wait and than invest. We denote $\delta := \mu - \alpha > 0$ and thus the parameter δ plays the same role as in the dynamic programming problem. Note that with our definition δ represents the opportunity cost of delaying the investment.

Now suppose that the value of the investment option is $V(x, t)$ as in the previous section. We construct a risk-free investment by shorting $n = \frac{\partial V(x,t)}{\partial x}$ units of the investment option, which is equivalent to shorting the replicator variable $\hat{x}$ as they are perfectly correlated. Hence the value of the investment option is $\Phi = V(x, t) - nx = V(x, t) - \frac{\partial V(x,t)}{\partial x}x$.

The excess rate of return from the investment in comparison to the replicate is $\delta = \mu - \alpha$ and thus the total return from the investment is δx. However, taking a short position has a natural cost related to it. The payment from taking a short position must be equal to the excess returns of the investment as otherwise nobody would take a long position on this portfolio. Since the investment was shorted $n = \frac{\partial V(x,t)}{\partial x}$ units, the total cost of shorting the investment is $\delta xn = \delta x\frac{\partial V(x,t)}{\partial x}$ per time period to keep things rational.

The risk-free return $R_{rf,i}$ of the investment over one time period is given as the total change in value of the investment minus the shorting payments over one time period, that is

$$R_{rf,i} = d\Phi - \delta x \frac{\partial V(x,t)}{\partial x} dt,$$

which gives us

$$\begin{aligned} R_{rf,i} &= dV(x,t) - \frac{\partial V(x,t)}{\partial x} dx - \frac{\partial^2 V(x,t)}{\partial x^2} x - \delta x \frac{\partial V(x,t)}{\partial x} dt \\ &= dV(x,t) - \frac{\partial V(x,t)}{\partial x} dx - \delta x \frac{\partial V(x,t)}{\partial x} dt, \end{aligned}$$

since $\frac{\partial^2 V(x,t)}{\partial x^2} = 0$ as the number of short positions are held fixed over the interval. Once again, by using the Itô's Lemma and the properties of a Wiener process, we obtain

$$R_{rf,i} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} dt - \delta x \frac{\partial V(x,t)}{\partial x} dt,$$

where we neglected the partial derivative over time as we are dealing with infinite time horizon. Note that this is the risk-free return of the investment that can be obtained within a time period dt. To avoid arbitrage possibilities, it must equal the risk-free return from the market during the same time period, that is $R_{rf,i} = R_{rf,m} := r\Phi dt$. This equality results in the equation

$$\frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} dt - \delta x \frac{\partial V(x,t)}{\partial x} dt = r\left(V(x,t) - \frac{\partial V(x,t)}{\partial x} x\right) dt$$

from which we obtain

$$\frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} + (r-\delta) x \frac{\partial V(x,t)}{\partial x} - rV(x,t) = 0. \tag{2.14}$$

Note once again that Equation 2.14 is time-independent. In addition, we note that Equation 2.14 identical to Equation 2.7 with the exception that the risk-free interest rate r equals the discount rate ρ used in the dynamic programming.

The same boundary and optimality conditions used previously apply also here, and thus the solution has the form

$$V(x) = Ax^{\beta_1}, \tag{2.15}$$

where the parameter β_1 is defined as

$$\beta_1 = \frac{1}{2} - \frac{r-\delta}{\sigma^2} \pm \sqrt{(\frac{r-\delta}{\sigma^2} - \frac{1}{2})^2 + 2\frac{r}{\sigma^2}} \tag{2.16}$$

and A is similar as in Equation 2.13.

2.1.3 Comparison of the derivation methods

We demonstrated above how to calculate the optimal stopping value for an investment with a stochastic variable that follows Brownian motion. When the contingent claims analysis was used to derive the value of the investment, the stochastic variable was linked to a stochastic replicator variable within a economy, while in the dynamic programming case the stochastic variable represented a direct stochastic variable related to the investment.

The differential equation derived was similar in both cases. However, the equation derived with the dynamic programming method included a discount rate parameter that has to be set by a decision-maker, while in the contingent claims method the same parameter was the risk-free interest rate. While both of the methods are valid for deriving the set of equations to be solved, the difference on the parameters sets limitations to the methods.

When choosing a method to derive the equations, one should consider does there exist a credible replicator within an economy for the stochastic variable in the model so that a risk-free investment can be constructed. In case there exist a credible replicator, the contingent claims method should be used as it exempts fixing one parameter. However, one should bear in mind that the contingent claims method is based on the capital asset pricing model that has been under heavy criticism lately, see for example [24].

2.2 Finite horizon investment

In the previous chapter we solved the value of the investment with infinite time horizon. The differential equation derived along with the boundary and optimality conditions were such that a closed-form solution could be derived. This was mainly due to the choice of infinite time horizon for which the dependence on time t vanished and thus the solution was static over the time of the investment option. While the infinite time horizon simplifies the analysis and produces usually an analytical solution to the problem, it may not be the most realistic assumption in all the cases.

Major part of the real option models presented in the literature are based on infinite horizon case, especially in the beginning of real option analysis era. One could argue that this is mainly due to the fact that assuming infinite time horizon conditions, the model usually reduces to one dimension which simplifies the analysis substantially and thus often leads to simple analytical solutions. Nowadays there are also several articles on finite horizon investments, but for example within real option games, which combine real options with game theoretic concepts, infinite time horizon is assumed essentially in every article, cf. [4]. Choosing infinite time horizon for a model might ease the analysis but poses some restrictions to the model. Consequently, using infinite time horizon in cases where there exist a clear ending date, i.e. terminal time, is highly questionable. Some examples of situations where finite time horizon has been used successfully are for example the offshore petroleum leases [60] and investment decision on nuclear power plant before the competition arrives [39]. In the latter article it was showed that the investment rules between using infinite and finite time horizon differ substantially, which points out that using infinite time horizon recklessly in improper settings leads to false results. Taking a step back and considering the characteristics of investments in real business world, de facto all the investments are limited with finite time horizon. Thus one should really consider carefully when using infinite time horizon over finite time horizon solely for simplification purposes.

Let us examine what are the effects of constraining time frame from infinite to finite. Assume that there exists some finite time T after which the option for investment is expired. Thus the time domain for the problem is $t \in [0, T]$. Adding a constrain to the time horizon has substantial effects on the problem setting. With the infinite time horizon the problem looks the same at every time period t as there is no dependence on time. However, with the finite time horizon the problem varies between the points of time, that is, when the time passes the possible investment time decreases as $T - t$ where t increases.

We consider similar risk-free investment as in the contingent claims analysis by shorting the investment with associated cost. The initial steps are the same as with the infinite horizon case and thus the risk-free return of the investment is given as

$$R_{rf,i} = d\Phi - \delta x \frac{\partial V(x,t)}{\partial x} dt = dV(x,t) - \frac{\partial V(x,t)}{\partial x} dx - \delta x \frac{\partial V(x,t)}{\partial x} dt.$$

However, when Itô's Lemma is used, the term with derivative over time does not vanish. Consequently along with the properties of a Wiener process, the risk-free return of the investment is

$$R_{rf,i} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} dt + \frac{\partial V(x,t)}{\partial t} dt - \delta x \frac{\partial V(x,t)}{\partial x} dt.$$

Once again, to avoid the possibility for arbitrage, this must equal the risk-free return of market, $R_{rf,m}$, and hence

$$\frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} + (r - \delta) x \frac{\partial V(x,t)}{\partial x} - rV(x,t) = -\frac{\partial V(x,t)}{\partial t}. \tag{2.17}$$

Note that the partial derivative with respect to time did not vanish and we are dealing with a parabolic partial differential equation, whereas the differential equation in the infinite horizon case (2.14) was merely a simple homogeneous differential equation.

To derive a solution for Equation 2.17, we need some boundary conditions. A reasonable set of boundary and optimality conditions csould be

$$V(0,t) = 0, \tag{2.18}$$

$$V(x^*,t) = x^* - S, \tag{2.19}$$

$$\frac{\partial V(x^*,t)}{\partial x} = 1. \tag{2.20}$$

$$V(x,T) = \max\{x - S, 0\}, \tag{2.21}$$

Note that the time-independent boundary conditions (2.18-2.20) are similar to ones used in the infinite time horizon case, cf. Equations 2.8-2.10. In addition we need to define a boundary condition for the value of the option when the finite time horizon ends. One reasonable boundary condition is given with Equation 2.21, which implies that the investment option will be exercised if the value of the stochastic variable is greater than some fixed value S, that is $x > S$.

Contrast to the infinite horizon case, Equation 2.17 along with the boundary conditions (2.18-2.20) cannot be solved analytically and thus numerical methods are required. In addition to the problem described above, there are several other problems within real options theory that does not have a closed-form solution.

2.3 Review on real option valuation methods

Real option valuation methods can be categorized into two sections - analytical and numerical methods. They can be further divided into subsections as represented in Figure 2.1.

Analytical solutions can divided into two types of solutions – closed-form and approximative solutions. For some problems, such as the infinite horizon investment problem presented in the previous section, a closed-form solution exists. Additional examples of problems that have a closed-form solution are valuation of two risky assets with restricted option to exercise [68], valuation of options with possibility to exchange one asset for another [50], [16] and valuation of compound options [31]. In some cases a closed-form solution does not exist but an approximative solution with reasonable accuracy can be derived, for example by using polynomial approximations [32], linear

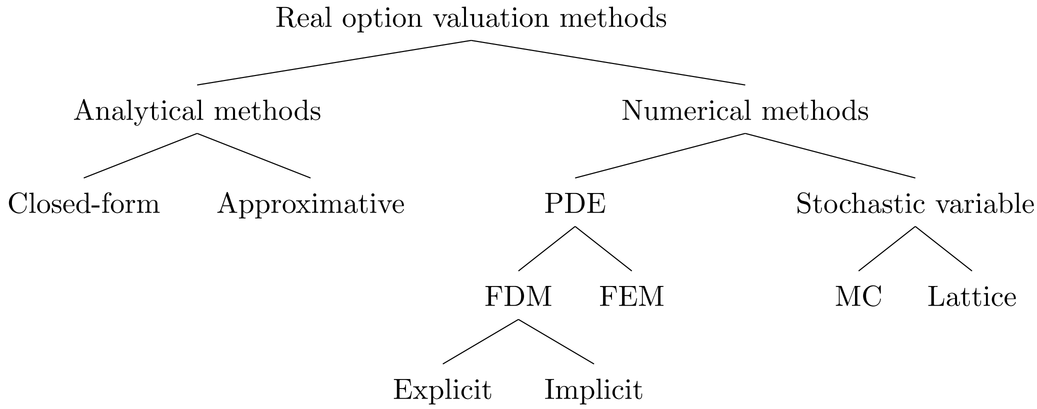


Figure 2.1: Classification of real option valuation methods.

linearpolation [27] or posing restrictions on feasible exercise strategies [6], [41].

However, there are several types of problems for which cannot be solved analytically. Some of the key features that break the solvability with analytical methods, in addition to previously discussed finite horizon, are increasing number of variables and non-constant variables. Needless to say, there are highly important features when modeling the real world. Limiting the number of variables or assuming constant features are major restrictive elements in real options models, which are typically implemented merely for academic reasons or to obtain some solution to problem.

Numerical methods can be used to derive a solution by approximating the underlying stochastic process or partial differential equation (PDE) of the model. Commonly used methods for approximating a stochastic process are Monte Carlo method (MC) and different types of lattice methods. In Monte Carlo method the possible outcome is simulated by repeated random sampling, while in the lattice methods the underlying stochastic process modeled by dividing the process into discrete steps. Monte Carlo method is typically used on relatively simple problems, e.g. real option valuation problems with no possibility to early exercise [9]. Lately the Monte Carlo method has also been used with more complex problems with an early exercise option [15]. Lattice methods, such as binomial and trinomial lattice tree methods, are commonly used in various types of problems due to their intuitive approach and usability [17], and in some cases they are easily extendable [10].

Numerical solution can also be derived by approximating a partial differential equation of the real option model. This can be done with methods such as finite difference method (FDM) and finite element method (FEM). The underlying idea in both methods is to discretize the solution domain and calculate an approximative solution of the differential equation in the whole domain. Finite difference method and its two solution schemes, explicit and implicit schemes, are more commonly used within real options theory. Examples of usages vary from using the explicit finite difference method to optimize the cut-off policy of mines [70] to modeling of generalized jump process with implicit finite difference method [12]. Finite element method has not been used that extensively the options theory probably due to its relatively high level of complexity, but there are several examples within financial options, such as combined penalty method for American options [**?**] and at least one example within real options theory [2].

Comparing the two real option valuation method approaches used in the corresponding literature, we note that there are three major disadvantages with analytical methods. First, the problem at hand should be simple enough so that an analytical solution can be derived. This widely restricts the problems to a small subset of problems for which a closed-from solution exists. These problems are rarely an accurate description of reality as several simplifying assumption are usually made to obtain an analytical solution, such as assuming infinite horizon, constant risk-free rate or restricting the exercise of an option. This fact cannot be stressed highly enough – in practice it is essentially the same thing as designing an airplane with approximative solutions for Navier-Stokes equation describing fluid flow on a wing. Even though the required accuracy with real options is not at the same level, the result would be devastating. Secondly, a considerable amount of work is usually required to work out an analytical solution for even slightly realistic problems. The level of mathematics involved with deriving an analytical solution can be at high level and hence out of reach for common practitioners. In addition, it also takes a considerable amount of time to derive a closed-form solution for some problems that could be solved instantly with numerical methods. Thirdly, the derived analytical solutions are not anymore viable when the complexity of the problem increases. For example, introducing an additional stochastic variable to the model breaks up the analytical solution totally. Consequently, the flexibility of analytical solutions is non-existent. This is a major problem considering the need for minor adjustments and the increasing complexity in the economic world.

Large part of the problems described above can be solved by using numerical methods. Numerical solution can be derived essentially for any type of problem with arbitrary level of complexity. Discretizing a partial differential equation simplifies the problem at hand by definition without neglecting the key items in the model, and the accuracy of the problem can be enhanced by increasing the simulation time, e.g. by choosing more dense grid. Once a numerical method has been implemented, it is very flexible to various types of minor and major adjustments. For example, the Monte Carlo method is essentially negligible to number of variables and arbitrary type of differential equation can be used with FDM and FEM. In addition, the level of detail numerical methods provide is exhaustive to a single solution provided by analytical methods. Furthermore, the numerical methods are relatively easy for practitioners to use as they can be easily implemented e.g. into a spreadsheet software.

Summarizing, even though analytical methods play a key role in developing the theoretical framework of real options, they are very restricted to simple problems that have merely an academic interest and does not have capabilities to meet the demand of flexibility and ability to model even increasing complexity of the real world. In the next chapter we delve more deeply into different types of numerical methods by presenting the mathematical framework behind the methods and introducing two case studies.

Chapter 3

Numerical methods

In this chapter we give a short introduction to the most commonly used numerical methods in the field of real options, and present two case studies that we will solve with the numerical methods discussed.

3.1 Introduction

3.1.1 Short history of numerical methods

We noted in the previous chapter that raising the complexity of a real option model even a little bit usually leads to a situation where a solution cannot be derived analytically. This situation is actually quite usual in the world of mathematics. One of the oldest mathematical problems that we can prove to be approximated numerically, was the length of the diagonal in a unit square that was approximated with the Old Babylonian tablet YBC 7289 [28] around 1800 BC. The first person that was credited with the proof was naturally Pythagoras over a millennium later. The same trend has followed throughout the history from linear interpolation to Archimedes' numerical integration and to Newton's and Euler method up until the modern days methods of deriving a numerical approximations for complex partial differential equations.

Shortly before the time of computers numerical approximations of functions were looked up from gathered tables or calculated by hand one iteration at time. This was a daunting task for some of the mathematical problems, but it was soon to change around the time of the World War II. The Axis powers used the Enigma machine to send encrypted messages to their peers. The Allied forces were puzzled how to break the encryption until pioneering but misunderstood computer scientist Alan Turing developed cipher machines that were used to break the code[1]. The cipher machines can be regarded as first computing machines, so called Turing machines, that led to the invention of the modern computer.

The study of numerical methods exploded with the invention of the modern computer after the World War II. Brilliant minds such as John von Neumann devoted part of their precious time to the development of numerical methods, such as the Monte Carlo method and finite difference method (FDM).

The underlying concept behind the Monte Carlo method was used the first time by Enrico Fermi in the 1930s in his study of neutron energies. Around fifteen years later von Neumann alongside with his colleague Stanislaw Ulam formulated the modern version of the Monte Carlo[2] method for the study of nuclear fission at Los Alamos Scientific Laboratory. Thus the Monte Carlo method played a significant role in the Manhattan Project that eventually led to the invention of the first atomic bomb and end of the World War II. [54]

While the concept beneath the FDM dates back to the time of Isaac Newton, the first idea of using FDM in solving problems within the field of physics can be considered to be given in 1928. More rigorous and extended formulations were given by von Neumann, e.g. on the stability conditions of finite difference schemes. The analysis of the finite difference methods broke out thereafter and also other numerical methods gained ground. [69]

3.1.1.1 Numerical methods in real option analysis

Considering the field of real options analysis, several different numerical methods have been used to obtain a solution for a differential equation that usually arises from the real option analysis. The most commonly used methods are different types of stochastic simulation methods, lattice methods, and finite difference methods.

In addition to basic usage of the methods, the methods are typically tailored to fit into a specific problem at hand. Some examples of these extensions are stretched trinomial lattices for valuation of public sector R&D investments [74], least-square Monte Carlo method for valuation of power

[1] Polish mathematician Marian Rejewski was the first one to solve the encryption on the first versions of the Enigma machine in 1932 by using the permutation and group theory. However, this effort was soon to be fruitless as the Axis powers made continuous improvements to the Enigma machine. [37]

[2] The name "Monte Carlo" was suggested by Nicholas Metropolis, a colleague of von Neumann and Ulam at Los Alamos, according to Ulam's uncle that would borrow money from his relatives because he "just had to go to Monte Carlo" [54].

transmission investments [63] and flux-limited upwind scheme for finite difference method to determine the optimal cut-off policy for mines [70]. Furthermore, additional numerical methods have been examined for use in real option analysis, such as finite element method [2] which is similar to the finite difference method.

However, in this study we will focus on comparing the basic versions of the three numerical methods most commonly used in real options analysis; the Monte Carlo method, the binomial lattice method and the finite difference method.

3.1.2 Monte Carlo method

The main idea of the Monte Carlo method is to approximate the probability of some outcome by repeated random sampling. The method as such does not have any rigorous mathematical definition. It is based on the classical definition of probability, which is defined as the number of desired events divided by the number of possible outcomes. These sets of numbers can be regarded as areas or volumes, and distributing a random sampling over the whole domain of possible outcomes ultimately produces the desired outcome.

One classic example on the intuition of Monte Carlo method is the estimation of π. Suppose we have a circle with radius R inside a square with side length $2R$. The ratio between the area of circle and square equals $\pi/4$. Thus the estimation for the value of π, which we denote with $\tilde{\pi}$, can be estimated with the Monte Carlo method by generating random points over the whole domain and counting the number of points N inside the circle and square, that is $\pi \approx \tilde{\pi} = 4\frac{N_{circle}}{N_{square}}$. See Figure 3.1 for an illustrative simulation. Clearly as the random sample size approaches infinity, the whole domain will be filled and thus the approximation $\tilde{\pi} \to \pi$.

The Monte Carlo method is highly convenient in the numerical integration schemes. Consider a simple integral

$$I = \int_{[0,1]^d} f(x)dx,$$

where $f(x)$ is assumed to be integrable over the defined domain. The integral can be defined with expected value $\mathbb{E}[f(X)] = I$, where $X \sim unif[0,1]^d$. Evaluating the value of the function $f(X_i)$ at $n \geq i \geq 1$ random points and gives the Monte Carlo estimate

$$\tilde{I} = \frac{1}{n}\sum_{i=1}^{n} f(X_i).$$

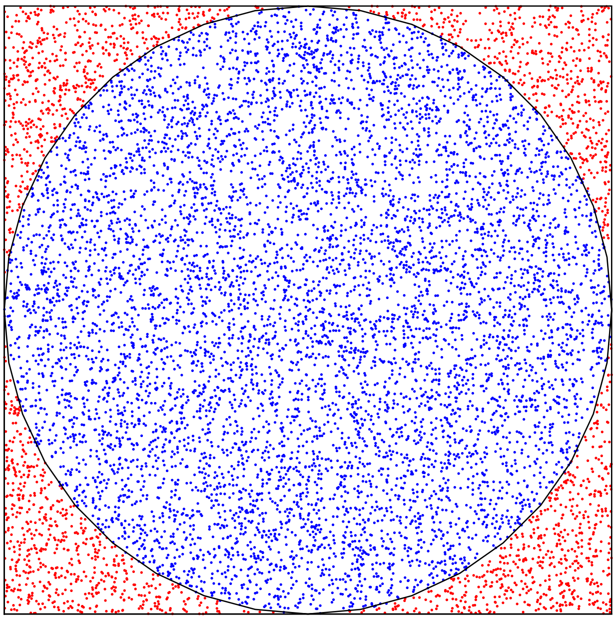


Figure 3.1: Monte Carlo simulation for approximating the value of π. Here we used sample size $n = 10000$ which resulted in $\tilde{\pi} = 3.1356$.

Consequently, by the strong law of large numbers[3] we note that $\tilde{I} \to I$ as $n \to \infty$. Thus we obtained an estimate for arbitrary integrable function with the Monte Carlo method. [33]

The increasing sample size is the essence of the Monte Carlo method which ultimately allows the convergence to the exact value. The number of simulations required to obtain a reasonable value can be investigated by examining the rate of convergence. For the Monte Carlo method, the convergence rate is obtained from the central limit theorem and it is of order $\mathcal{O}(\frac{1}{\sqrt{n}})$, where n denotes the sample size. While this is not the most efficient rate of convergence in comparison to other numerical integration schemes, the convergence rate holds also in higher dimensions, that is, when $d > 1$. Thus evaluating integrals in higher dimensions is relatively effortless with the Monte Carlo method as with other deterministic methods the rate of convergence decreases as the dimension increases. [33]

The Monte Carlo method has various applications throughout different fields due to its applicability to higher dimension problems with uncertainty. It is used heavily in computational physics, chemistry and biology, and in several everyday applications such as weather forecasting and artificial in-

[3] *Strong law of large numbers.* Let $(X_i : n \geq 1)$ be independent identically distributed random variables with $\mathbb{E}|X_i| < \infty$. Then $\frac{\sum_{i=1}^{n} X_i}{n} \to \mathbb{E}[X_i]$ almost surely as $n \to \infty$.

telligence, just to give few examples. Taking into account that the field of finance and economics usually involves expected values, high dimensions and uncertainty, the Monte Carlo method is also highly attractive method for our study.

3.1.3 Binomial lattice model

The general idea of behind different types of lattice models is to simulate the continuous stochastic process of a stochastic variable with discrete steps. This is obtained by dividing the time domain into discrete steps where the value of a stochastic variable is evaluated. In the binomial lattice model the value may go either up or down with a given probability, while in the trinomial lattice model the value has three different states in the next step. However, in this study we will focus only to the binomial lattice model, see Figure 3.2 for an illustration of the binomial grid. The binomial lattice model was originally introduced by Cox et al. in 1979 [17] as a framework to price options. The model is known as the CRR model in the literature, which abbreviation we will also use.

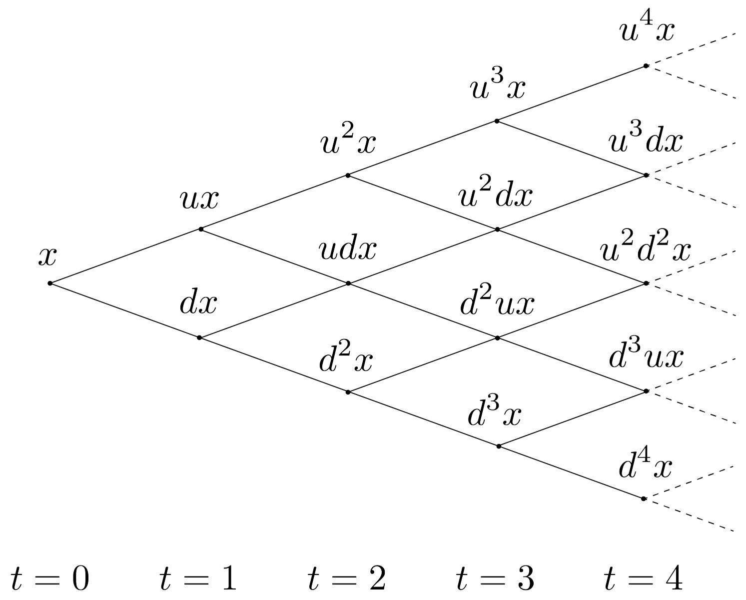


Figure 3.2: Schematic illustration of a binomial lattice grid where the stochastic variable x can either increase or decrease in the discrete time grid.

Suppose that we have a continuous stochastic process x, which is divided into $n \geq i \geq 0$ discrete steps. Thus the value of the stochastic variable at each node is given as x_i. Advancing from a node forwards, the value can either increase by the factor u with probability p, or decrease by the factor $d < u$ with probability $(1-p)$. Let U_n denote the number of movements upwards and thus the movements U_n follow Binomial distribution with n trials with probability p, that is $U_n \sim B(n, q)$. Consequently, for the binomial model at the terminal time $i = n$, we have

$$x_n = pu^{U_n} d^{n-U_n} = pe^{U_n \ln\left(\frac{u}{d}\right) + n\ln(d)}.$$

We assume that the probability for the up and down is the same, that is $p = \frac{1}{2}$, and that

$$u = e^{b\Delta t + \sigma\sqrt{\Delta t}}, \quad d = e^{b\Delta t - \sigma\sqrt{\Delta t}},$$

where we have

$$b = \frac{1}{2}\frac{\ln(u) + \ln(d)}{\Delta t}, \quad \sigma = \frac{1}{2} = \frac{\ln(u) - \ln(d)}{\sqrt{\Delta t}}.$$

Consequently, with the given assumptions the value of the stochastic variable at the terminal time $T = n\Delta t$ is given as

$$x_n = pe^{bT + \sigma\sqrt{T}\left(\frac{2U_n - n}{\sqrt{n}}\right)}.$$

Now, using the central limit theorem[4], we note that

$$x_n \to pe^{bT + \sigma W(T)} = x(T) \quad \text{as} \quad n \to \infty.$$

Thus with given parameters the terminal value of the binomial model converges to the exact value of the continuous stochastic process. [43]

Few remarks about the convergence. Note that we made some assumptions about the parameters and assumed that we know the terminal value of the stochastic variable. Regarding the convergence of the binomial lattice model, it is unrelevant how the probability p is chosen. However, if the stochastic variable is not dependent on the terminal value, the proof presented above does not guarantee the convergence to the exact value. To prove that the binomial lattice method also applies in cases where the value of the stochastic variable is dependent on the whole time domain, more rigorous methods should be applied. Using Donsker's Theorem, which is a extension of the central limit theorem, it can be proven that the binomial lattice method converges weakly to the geometric Brownian motion that is

[4] *Central limit theorem (Lindeberg-Lèvy).* Let $(X_i : n \geq 1)$ be independent identically distributed random variables with $\mathbb{E}[X_i] = \mu$ and $Var[X_i] = \sigma^2 < \infty$. Then $\frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}} \to N(0, \sigma^2)$ almost surely as $n \to \infty$.

the essence of real option models. However, we will not go through the prove here as it is slightly out of the scope of this study. [44]

Note that the binomial lattice model has similarities with the decision tree analysis that was discussed earlier, but it has two distinct differences. First, the shape of the lattice, i.e. the tree, has a fixed symmetry while in decision tree analysis there can be an arbitrary number of outcomes at every branch. Second, the nodes in the lattice describe the current state of a variable that evolves discretely through time. Thus the binomial lattice method only provides information on the value of a stochastic variable and not the decision itself.

The binomial lattice model introduced above takes us one step closer to finite difference method, which can be regarded as an extension of the lattice model.

3.1.4 Finite difference method

The main idea of finite difference method is to create a discrete grid over the domain of a differential equation, calculate an initial value at some grid point, and move along the grid by calculating an approximation of the derivative at the next grid point. Hence a discrete image of the domain will be formed as all the nodes in the grid are calculated.

Let us start with a simple example. Assume that we are seeking a solution in domain $x \in [x_0, x_n]$ for a first order ordinary differential equation described by the following set of equations:

$$\begin{cases} \dfrac{dy}{dx} = f(x, y), \\ y(x_0) = y_0. \end{cases}$$

The first step is to form a discrete grid by dividing the continuous domain to non-overlapping intervals. Hence set up a discrete grid $x_0 < x_1 < \ldots < x_n$. For simplicity, we assume that the grid is regular, that is, the length of interval is given by a constant h and thus $x_i = x_0 + ih$, see Figure 3.3 for illustration.

Next we will start at some initial value and move along the grid to determine the discrete image of the domain. There are several interpolation methods to move along the grid. The most simplest is the classical Euler's method, where the approximation of the differential equation at the neighboring grid point is determined from equation

$$y_{i+1} = y_i + hf(x_i, y_i). \tag{3.1}$$

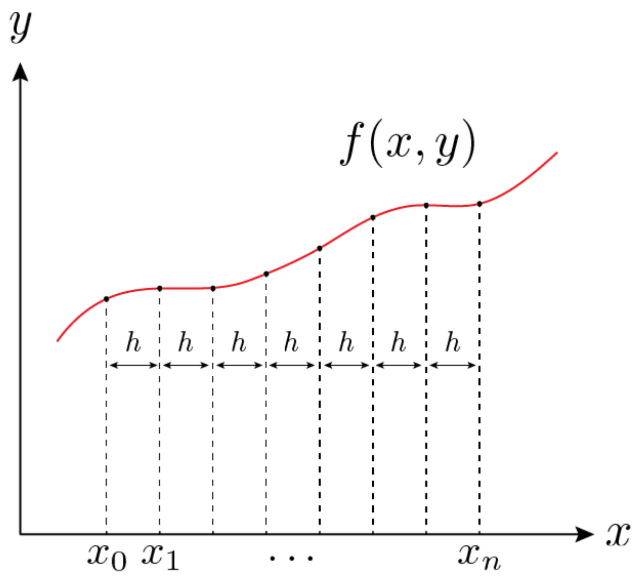


Figure 3.3: Illustration of one dimensional discrete grid.

Hence starting from the initial value y_0, the next grid point can be calculated according to Equation 3.1. Consequently, moving a single step forwards through the discrete grid until point x_n, a numerical approximation of the differential equation at every grid point can be determined. However, Euler's method is rarely used in practice as the local and global truncation errors[5] are of order $\mathcal{O}(h^2)$ and $\mathcal{O}(h)$, respectively, which imply that the Euler's method is rather ineffective. However, based on the same single step forward -principle, there other more effective methods available. One example is the fourth order Runge-Kutta method for which the local and global truncation errors are of order $\mathcal{O}(h^5)$ and $\mathcal{O}(h^4)$.

Both of the two methods are *explicit methods* as the approximation of the differential equation is calculated by taking account only the previous value of equation. Contrast to the explicit methods, in *implicit methods* the current value of the equation is also taken into account. Classical example of an implicit method is the backward Euler method, for which the numerical scheme is the following:

$$y_{i+1} = y_i + hf(y_{i+1}, x_{i+1}). \tag{3.2}$$

We note from Equation 3.2 that the state of the system, y_{i+1}, is included on

[5] The local truncation error measures the error made at a single step, while the global truncation error measures the sum of local errors for multiple steps. For example in Euler's method, reducing the step size to a half the local and global discretization errors are reduced by a quarter and half, respectively.

the both sides of the equation, which indicates an implicit method. If the differential equation is linear and an initial value is assigned, it is possible to solve Equation 3.2 from the previous step. However, if the differential equation is nonlinear, it is usually necessary to calculate all the steps to obtain a solution.

As noted in the beginning of this section, there are multiple interpolation methods to consider when solving a problem with finite difference method. Considering differential equations outside the illustrative example we presented, the basic idea of finite difference method stays the same as the derivatives of a function are simply replaced with approximations and the discrete values within the grid are calculated one by one. For further details, see for example [49], [23].

3.2 Applications

Next we present two possible applications for the numerical methods discussed above. In the first case we consider an option to invest into a project where the cash flows are uncertain. We assume that the payoff from the project will be given at the terminal time when the project is complete. For this type of investment problem a closed-form solution exists, which we will use as a benchmark for the numerical methods. In the second case we consider an abandonment option for an ongoing project with uncertainty in cash flows. We assume that project has some salvage value and thus a decision-maker has to decide at every point of time whether to continue the project or abandon the project with a salvage value. Thus the second problem is a optimal stopping problem for which a closed-form solution does not exist and hence numerical methods are required to obtain a solution.

3.2.1 Investment option

Suppose that we are seeking a valuation to a project with a finite lifetime $t \in [0, T]$. The cash flows x from the investment are stochastic with a standard deviation σ. Hence the evolution of cash flows over time are described as

$$\frac{dx(t)}{x(t)} = rdt + \sigma dW(t), \tag{3.3}$$

where r is the risk-free interest rate. No initial investment is required at $t = 0$, but if the investment to the project is executed it has a fixed investment cost I.

Consequently, we are dealing with a finite time horizon and thus by following the discussion in Chapter 2.2 and assuming that there are no dividends, the set of equations to be solved are given as

$$\begin{cases} \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} + rx\frac{\partial V(x,t)}{\partial x} - rV(x,t) = -\frac{\partial V(x,t)}{\partial t}, & (3.4a) \\ V(0,t) = 0, & (3.4b) \\ V(x,T) = \max\{x - I, 0\}. & (3.4c) \end{cases}$$

The first boundary condition (3.4b) is a typical Dirichlet boundary condition that follows from the properties of the stochastic process for the stochastic variable x. The second boundary condition (3.4c) determines the optimal investment strategy at the terminal time – the value of the investment option has no value if the fixed investment cost I is greater than the cash flows x at the terminal time. The boundary conditions can be visualized on the (x,t)-domain, which is illustrated in Figure 3.4. Note that there are two fixed boundaries, $V(x = 0, t)$ and $V(x, t = T)$, otherwise the boundaries are free. In addition, note that if a solution in the whole domain is calculated, the maximum value for x has to be fixed to some value $x = X$.

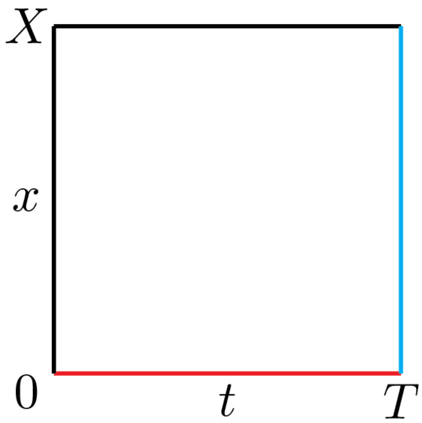


Figure 3.4: Domain of the case problem, where the fixed boundaries $V(x = 0, t)$ and $V(x, t = T)$ are marked with red and blue color, respectively.

Note that the case study on investment problem described with set of equations (3.4a-3.4c) has a close resemblance to the pricing of an European call option[6] with the Black-Scholes equation. The analogies between the two approaches are listed in Table 3.1.

[6] In finance, a European option can be exercised only at the expiration time of the option, while an American option can be exercised at any point of time during the option lifetime. Given the price of underlying security P and the strike price S, the payoff for a call option is defined as $\max\{P - S, 0\}$ and for a put option as $\max\{S - P, 0\}$.

Table 3.1: Analogies of the investment problem with a European call option.

Call option	Investment problem
Stock price	Investment cash flows (x)
Exercise price	Fixed investment cost (I)
Time to expiration	Time to invest (T)
Volatility of stock price	S.d. of cash flows (σ)
Risk-free interest rate	Risk-free interest rate (r)

Contrast to several other real option problems with a finite time horizon, a closed-form solution has been derived to the set of equations (3.4a-3.4c) describing the case problem. The value of the investment is given as

$$\begin{aligned} V(x,t) &= \Phi(d_1)x - \Phi(d_2)Ie^{-r(T-t)}, \\ d_1 &= \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{x}{I}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right], \\ d_2 &= d_1 - \sigma\sqrt{T-t}, \end{aligned} \tag{3.5}$$

where $\Phi(\cdot)$ is the standard normal distribution cumulative distribution function[7]. The reasoning behind choosing a case study with a closed-form solution is that we can use the analytical solution as a benchmark when comparing different numerical methods.

Next we will apply the numerical methods presented in Chapter 3.1 to the investment case study.

3.2.1.1 Monte Carlo method

In this section we derive a valuation for the investment case study problem (3.4a-3.4c) by using the Monte Carlo method. We follow the steps given by Glasserman in [33] and advise the reader to refer it for further details.

Given the evolution of cash flows (3.15) and the properties of a Wiener process (2.2), we may solve the cash flows at terminal time T, giving us

$$x_i(T) = x(0)e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}Z_i},$$

[7] The value for the standard normal distribution cumulative distribution function can be calculated from equation $\Phi(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$, where the error function is given as $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$.

where $(Z_i : n \geq i)$ is a random vector containing pseudorandom values drawn from the standard normal distribution. We have a boundary condition (3.4c) that holds at the terminal time T, which we can use to solve the value of the investment. For all the randomly generated values in the random vector Z_i, the value of the investment option given as a vector is

$$V_i = e^{-rT} \max\{x_i(T) - I, 0\},$$

where the cash flow at the terminal time has been discounted with a factor e^{-rT} to obtain the present value. Hence the estimate for the value of the investment can be derived from the mean of the investment option vector, that is

$$\tilde{V} = \frac{\sum_{i=1}^{n} V_i}{n}.$$

The estimate $\tilde{V}$ is unbiased for any $n \geq 1$ and strongly consistent as $\tilde{V} \to V$ with probability 1 when $n \to \infty$. Clearly a finite sample size is sufficient for an approximative value. The asymptotic $(1 - \omega)$ confidence interval can be calculated from equation

$$\tilde{V} \pm z_{\omega/2} \frac{\tilde{V}_{s.d.}}{\sqrt{n}}, \tag{3.6}$$

where $\tilde{V}_{s.d.}$ is the standard deviation of investment option vector V_i, that is

$$\tilde{V}_{s.d.} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (V_i - \tilde{V})^2}.$$

Consequently for example with 95% confidence interval we have $\omega = 0.05$ and $z_{\omega/2} \approx 1.96$.

Note that with the given variant of the Monte Carlo method, the value of the investment option was independent on the path between 0 and T. Thus the procedures given above are not applicable with options that are path-dependent, such as an American option in finance where the option can be exercised at any time during the lifespan of the option.

3.2.1.2 Binomial lattice method

The binomial lattice method we present below is originally the work of Cox et al. [17], which we will follow here on the implementation to the investment case study.

The method consists of two phases. In the first phase the lattice with n states and consequently $n - 1$ discrete time steps is created by starting from the current value of cash flows $x(0)$ and moving either up or down on every time step. The up and down factors given as

$$u = e^{\sigma\sqrt{\Delta t}}, \tag{3.7}$$

$$d = e^{-\sigma\sqrt{\Delta t}}, \tag{3.8}$$

where $\Delta t = \frac{T}{n}$ is the length of the discrete time step. Thus the value of cash flow at every node is given by $x_{i,j} := (u^i d^j x(0) : n-1 \geq i+j)$. After deriving the values for all the nodes and consequently the investment value at the terminal time, boundary conditions (3.4c) are applied at the terminal nodes according to equation

$$V_{i,j} = \max\{x_{i,j} - I\} \quad \forall (i+j) = n-1.$$

See Figure 3.5 for a illustration with two time steps.

In the second phase the investment values at intermediate nodes are calculated according to equation

$$V^{int}_{i-1,j-1} = \left(pV^{up}_{i,j-1} + (1-p)V^{down}_{i-1,j}\right) e^{-r\Delta t} \quad \forall (i+j) \leq n-1, \tag{3.9}$$

where p is the probability for the up-movement given as

$$p = \frac{e^{r\Delta t} - d}{u - d}.$$

Note that Equation 3.9 is similar to the backward Euler scheme which utilizes the fact that terminal value is defined. Consequently moving backwards in time according to Equation 3.9 the value of the investment at $t = 0$ can be obtained.

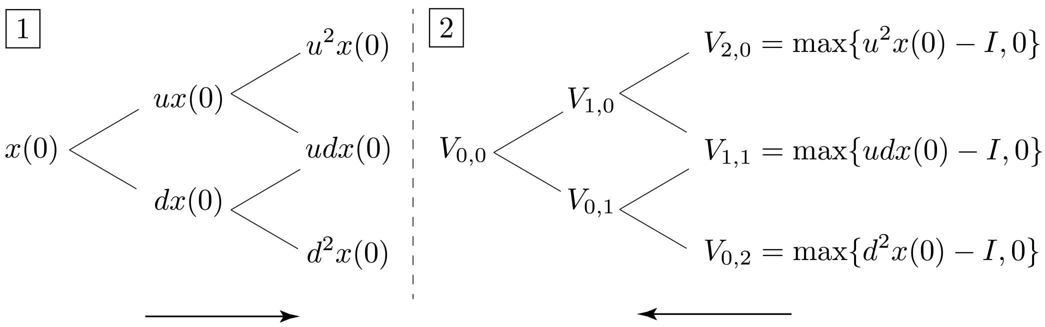


Figure 3.5: Illustration of the binomial lattice method for the case study with two time steps.

3.2.1.3 Finite difference method

Finally, we present a finite difference implementation for the investment case study. Contrast to other numerical methods presented so far, we will use two different schemes, explicit and implicit, within the finite difference method to fully compare the efficiency of the method. For a practical reference on finite difference methods, see [77], and for more theoretical discussion, see [46].

In both cases, we begin by setting up the discrete grid which remains the same for both schemes. Since we are calculating the value of the investment in the whole domain, we have to set some upper boundary for the cash flows x given as X, cf. 3.4. Consequently we discretize the cash flows into a grid by setting $0 = x_0 < x_1 < \ldots < x_N = X$ with $x_i = i\Delta x$, where $\Delta x = \frac{X}{N}$. The continuous time t is discretized similarly in similar manner by setting $0 = t_0 < t_1 < \ldots < t_M = T$ with $t_j = j\Delta t$, where $\Delta t = \frac{T}{M}$. Note that in this case we have a regular grid for both variables, see an illustration of the grid in Figure 3.6. Hence the discrete approximation of every value is determined uniquely from the grid points, that is $\tilde{V}_{i,j} \approx V(x_i, t_j)$.

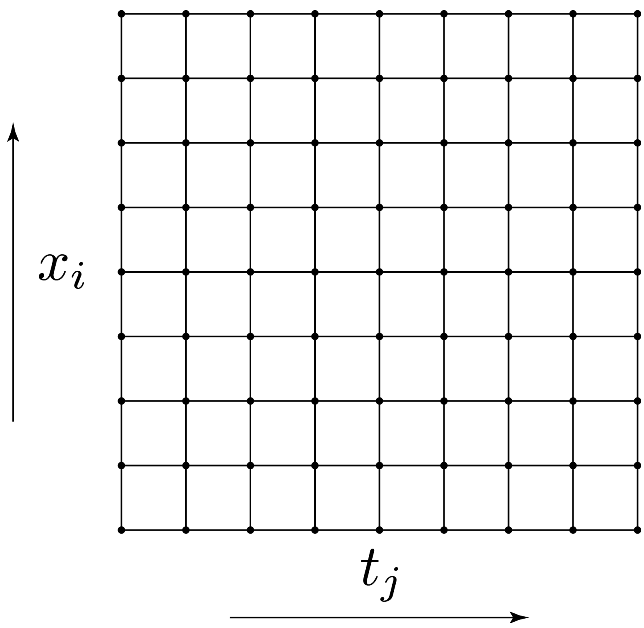


Figure 3.6: Schematic illustration of a regular grid used in the case study.

Next we apply the boundary conditions of the case study to the generated grid. The boundary conditions given in Equations (3.4b-3.4c) are given in grid notation as

$$\begin{aligned}\tilde{V}_{0,j} &= 0 \quad j = 0, 1, \ldots, M\\ \tilde{V}_{i,M} &= \max\{x_i - I, 0\} \quad i = 0, 1, \ldots, N\\ \tilde{V}_{N,j} &= X - Ie^{-r(T-t_j)} \quad j = 0, 1, \ldots, M.\end{aligned}$$

Note that as the domain was also limited with respect to cash flows x in addition to the finite time horizon, we have to set a boundary condition at the boundary $V(x = X, t) = \tilde{V_{N,j}} = X - Ie^{-r(T-t_j)}$. The reasoning behind this is that the value of the investment option asymptotes to $x - Ie^{-r(T-t_j)}$ with large values for x, where the fixed investment I is discounted to present value.

The main idea of finite difference method is to estimate an unknown value of the grid from neighboring grid points. The estimation is performed with approximative values for differentials. The approximations vary slightly depending on the direction of movement. For the forward difference, we have equations

$$\frac{\partial V}{\partial x} \approx \frac{\tilde{V}_{i+1,j} - \tilde{V}_{i,j}}{\Delta x}, \qquad \frac{\partial V}{\partial t} \approx \frac{\tilde{V}_{i,j+1} - \tilde{V}_{i,j}}{\Delta t},$$

and for the backward difference, we have

$$\frac{\partial V}{\partial x} \approx \frac{\tilde{V}_{i,j} - \tilde{V}_{i-1,j}}{\Delta x}, \qquad \frac{\partial V}{\partial t} \approx \frac{\tilde{V}_{i,j} - \tilde{V}_{i,j-1}}{\Delta t}.$$

Central difference is more accurate than the forward and backward difference, and it is given as

$$\frac{\partial V}{\partial x} \approx \frac{\tilde{V}_{i+1,j} - \tilde{V}_{i-1,j}}{2\Delta x}, \qquad \frac{\partial V}{\partial t} \approx \frac{\tilde{V}_{i,j+1} - \tilde{V}_{i,j-1}}{2\Delta t}.$$

The difference for the second derivative is obtained from the difference in forward and backward difference, which is

$$\frac{\partial^2 V}{\partial x^2} \approx \frac{\tilde{V}_{i+1,j} - 2\tilde{V}_{i,j} + \tilde{V}_{i-1,j}}{(\Delta x)^2}.$$

In *explicit difference scheme*, steps are taken backwards in time. Thus we use the backward difference for time t and the central difference for x. Substituting the differences into Equation 3.4a along with $x \approx i\Delta x$ gives us

$$\begin{aligned}\frac{1}{2}\sigma^2 i^2 (\Delta x)^2 \frac{\tilde{V}_{i+1,j} - 2\tilde{V}_{i,j} + \tilde{V}_{i-1,j}}{(\Delta x)^2} - r\tilde{V}_{i,j}\\ + ri\Delta x \frac{\tilde{V}_{i+1,j} - \tilde{V}_{i-1,j}}{2\Delta x} = -\frac{\tilde{V}_{i,j} - \tilde{V}_{i,j-1}}{\Delta t},\end{aligned}$$

and hence by using some basic algebra we obtain

$$\tilde{V}_{i,j-1} = a_i\tilde{V}_{i-1,j} + b_i\tilde{V}_{i,j} + c_i\tilde{V}_{i+1,j} \tag{3.10}$$

with $i = 0, \ldots, N-1$, and $j = 1, \ldots, M-1$. That is, we are calculating the values at the inner nodes which are inside the boundaries. The coefficients in Equation 3.10 are given as

$$\begin{aligned} a_i &= \frac{1}{2}(\sigma^2 i^2 - ri)\Delta t, \\ b_i &= 1 - (\sigma^2 i^2 + r)\Delta t, \\ c_i &= \frac{1}{2}(\sigma^2 i^2 + ri)\Delta t. \end{aligned}$$

Hence we can calculate an approximation for the current value of the option, $\tilde{V}_{i,0}$, starting from the boundaries at the terminal time and working our way back to $t = 0$. Note that the method derives the option value at every grid point. To obtain the option value at some other data point, e.g. between grid points, some reasonable interpolation method should be used such as bilinear interpolation.

A simple way to perform the calculations is to transfer the set of equations (3.10) into matrix form

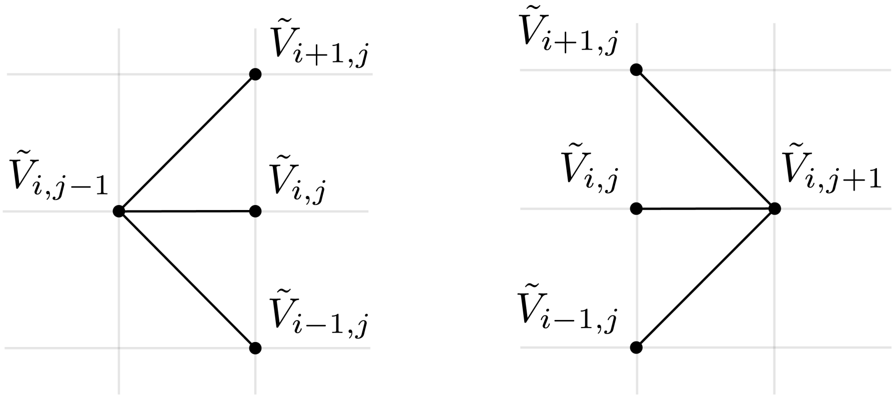
$$\tilde{\mathbf{v}}_{j-1} = \mathbf{A}\tilde{\mathbf{v}}_j + \mathbf{k}_j \quad j = M, \ldots, 1, \tag{3.11}$$

where the system of equations at time j are given as

$$\begin{bmatrix} \tilde{V}_{1,j-1} \\ \tilde{V}_{2,j-1} \\ \vdots \\ \vdots \\ \tilde{V}_{N-2,j-1} \\ \tilde{V}_{N-1,j-1} \end{bmatrix} = \begin{bmatrix} b_1 & c_1 & 0 & \cdots & \cdots & 0 \\ a_2 & b_2 & c_2 & \ddots & & \vdots \\ 0 & a_3 & b_3 & c_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & a_{N-2} & b_{N-2} & c_{N-2} \\ 0 & \cdots & \cdots & 0 & a_{N-1} & b_{N-1} \end{bmatrix} \begin{bmatrix} \tilde{V}_{1,j} \\ \tilde{V}_{2,j} \\ \vdots \\ \vdots \\ \tilde{V}_{N-2,j} \\ \tilde{V}_{N-1,j} \end{bmatrix} + \begin{bmatrix} a_1\tilde{V}_{0,j} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ c_{N-1}\tilde{V}_{N,j} \end{bmatrix}.$$

The solution scheme with the matrix formulation presented above is relatively easy to implement into a computer environment.

In *implicit difference scheme*, steps are taken forward in time. The difference of the explicit and implicit scheme is illustrated in Figure 3.7. In both methods the value is calculated from the three values before or after with respect to time depending on the scheme. There are also other methods available, such as Crank-Nicholson scheme where the nodal value is calculated implicitly from five neighboring grid points. However, in this study we will present only the typical explicit and implicit scheme.



(a) Explicit (b) Implicit

Figure 3.7: Determining the nodal value with the explicit and implicit method.

As noted from Figure 3.7, the desired node value in implicit scheme is derived from the nodes prior in time. Thus we use forward difference to approximate the time differential and central difference for the other variable. Plugging the expressions into 3.4a gives us

$$\begin{aligned}\frac{1}{2}\sigma^2 i^2(\Delta x)^2 \frac{\tilde{V}_{i+1,j} - 2\tilde{V}_{i,j} + \tilde{V}_{i-1,j}}{(\Delta x)^2} - r\tilde{V}_{i,j} \\ + ri\Delta x \frac{\tilde{V}_{i+1,j} - \tilde{V}_{i-1,j}}{2\Delta x} = -\frac{\tilde{V}_{i,j+1} - \tilde{V}_{i,j}}{\Delta t},\end{aligned} \tag{3.12}$$

which can be rewritten as

$$a_i\tilde{V}_{i-1,j} + b_i\tilde{V}_{i,j} + c_i\tilde{V}_{i+1,j} = \tilde{V}_{i,j+1} \tag{3.13}$$

with $i = 0, \ldots, N-1$, and $j = 1, \ldots, M-1$, where the coefficients are given as

$$\begin{aligned} a_i &= \frac{1}{2}(ri - \sigma^2 i^2)\Delta t, \\ b_i &= 1 + (\sigma^2 i^2 + r)\Delta t, \\ c_i &= -\frac{1}{2}(ri + \sigma^2 i^2)\Delta t. \end{aligned}$$

Hence we have a set of equations where the time-zero vector can be solved. The calculation scheme (3.13) can be presented as matrix formulation:

$$\mathbf{B}\tilde{\mathbf{v}}_j = \tilde{\mathbf{v}}_{j+1} - \mathbf{k}_{j+1} \quad j = M-1, \ldots, 0, \tag{3.14}$$

where the system of equations at time j is the following:

$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & \cdots & 0 \\ a_2 & b_2 & c_2 & \ddots & & \vdots \\ 0 & a_3 & b_3 & c_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & a_{N-2} & b_{N-2} & c_{N-2} \\ 0 & \cdots & \cdots & 0 & a_{N-1} & b_{N-1} \end{bmatrix} \begin{bmatrix} \tilde{V}_{1,j} \\ \tilde{V}_{2,j} \\ \vdots \\ \vdots \\ \tilde{V}_{N-2,j} \\ \tilde{V}_{N-1,j} \end{bmatrix} = \begin{bmatrix} \tilde{V}_{1,j+1} \\ \tilde{V}_{2,j+1} \\ \vdots \\ \vdots \\ \tilde{V}_{N-2,j+1} \\ \tilde{V}_{N-1,j+1} \end{bmatrix} - \begin{bmatrix} a_1\tilde{V}_{0,j} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ c_{N-1}\tilde{V}_{N,j} \end{bmatrix}.$$

Note that even though we used forward difference to approximate the value at the next grid point, the system of equations (3.14) is such that it can be solved by moving backwards in time. To carry out this, we need to compute the inverse of matrix $\mathbf{B}$ to derive the solution vector $\tilde{\mathbf{v}}_j$. However, as the matrix $\mathbf{B}$ tridiagonal and constant in time, this is not a major problem. In addition, due to given properties of the matrix, some factorization method such as LU decomposition can be used to speed up the calculation time.

3.2.2 Abandonment option

Next we present our second case study in which we consider an option to abandon a project under finite lifetime $t \in [0, T]$. Let the project generate uncertain cash flow x at the terminal time T and thus the cash flow is described with Brownian motion

$$\frac{dx(t)}{x(t)} = rdt + \sigma dW(t), \tag{3.15}$$

where r is the risk-free interest rate and σ is the standard deviation of cash flow. The project can be abandoned at any time with a fixed salvage value S.

The possibility for abandonment at any given time causes substantial changes to the set of equations derived for the finite time horizon investments. We partly follow [76] in the derivation of the set of equations and in the implementation of numerical methods.

In Chapter 2.2 it was assumed that the risk-free return of the investment $R_{rf,i}$ and risk-free return of market $R_{rf,m}$ were equal to avoid arbitrage. However, as it is up to the holder of the project to decide if the project should be abandoned or not, it is possible that the decision-maker makes wrong decision and does not abandon the project at optimal time. In this case the value of project is less than the value obtained from the market and if the decision-maker acts optimally, the value of the project equals the market value. Hence we note that it must hold $R_{rf,i} \leq R_{rf,m}$ at all periods of time and thus we have the following condition:

$$\mathcal{L}(x,t) := \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V(x,t)}{\partial x^2} + rx\frac{\partial V(x,t)}{\partial x} - rV(x,t) + \frac{\partial V(x,t)}{\partial t} \leq 0,$$

The payoff from the abandonment option is clearly $\Psi(x,t) := \max\{S - x(t), 0\}$. Now suppose that $V(x,t) < \Psi(x,t)$. Then the underlying replicate variable for the cash flows x can be bought from the market to cover the project value and risk-free profit is obtained. Hence to avoid this arbitrage opportunity, we must have the inequality condition $V(x,t) \geq \Psi(x,t)$. Here clearly if the abandonment value is larger than the payoff, that is, $V(x,t) > \Psi(x,t)$, it makes no sense to abandon the project and if the abandonment value equals the payoff, $V(x,t) = \Psi(x,t)$, the project should be abandoned immediately.

Combining the two inequalities we obtain two regions where the project should be continued or abandoned together define the boundary where the project should be abandoned. In the abandoning region the value of project equals the payoff and the rate of return from the project is less than from the market, giving us

$$V(x,t) = \Psi(x,t), \quad \mathcal{L}(x,t) < 0,$$

and then in the holding region the value of the project is larger than the payoff and the rate of return from the project equals the rate of return from the market, that is

$$V(x,t) > \Psi(x,t), \quad \mathcal{L}(x,t) = 0.$$

In addition to the two regions, we need to define the boundary conditions for the abandonment option. Since we are dealing with finite time horizon, the option to abandon the project ends at terminal time where the option value is $V(x,T) = \Psi(x,T)$. Then, clearly if the cash flow from the project is sufficiently large the option has no value and thus $\lim_{x\to\infty} V(x,t) = 0$.

Hence given all the required conditions, we may formulate the abandonment problem as the following free boundary problem:

$$\begin{cases} \mathcal{L}(x,t) \leq 0, & (3.16a) \\ V(x,t) \geq \Psi(x,t), & (3.16b) \\ \mathcal{L}(x,t)(V(x,t) - \Psi(x,t)) = 0, & (3.16c) \\ V(x,T) = \Psi(x,T), & (3.16d) \\ \lim_{x\to\infty} V(x,t) = 0. & (3.16e) \end{cases}$$

Finding a solution for the free boundary problem involves finding the optimal abandonment boundary $x^f(t)$, see Figure 3.8, which cannot be determined a priori. We note that the abandonment problem is similar to finding an optimal price for an American put option, see Table 3.2 for analogies between the approaches.

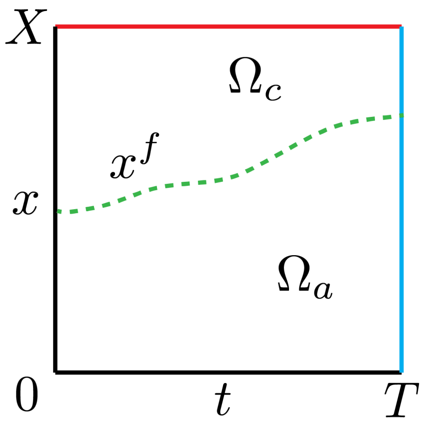


Figure 3.8: Finite domain for the abandonment problem, where the fixed boundaries $V(x = X, t)$ and $V(x, t = T)$ are marked with red and blue color, respectively, and the whole domain Ω is divided with the abandonment boundary x^f into the continuation region $\Omega_c \subset \Omega$ and the abandonment region $\Omega_a \subset \Omega$.

Table 3.2: Analogies of the abandonment problem with an American put option.

Put option	Abandonment problem
Stock price	Project cash flows (x)
Exercise price	Salvage value (S)
Time to expiration	Time to abandon (T)
Volatility of stock price	S.d. of cash flows (σ)
Risk-free interest rate	Risk-free interest rate (r)

Contrast to the first case, the free boundary problem does not have a closed form solution[8] and hence must be solved numerically. Naturally, there

[8]Note that if we would have similar boundary conditions as in the first case study but

are several numerical methods that can be applied to obtain a solution. Considering the numerical methods presented in the previous discussion, deriving a solution with the Monte Carlo method is not that straightforward due to the path-independence feature. It is nevertheless possible for example with variance reduction methods [8] and Lagrangian martingales [64]. However, instead of delving into these extensions of the Monte Carlo method, we present how to obtain a solution for the abandonment option with binomial lattice and finite difference method for which the procedures are very similar to the implementations presented in Chapter 3.2.1.

3.2.2.1 Binomial lattice method

Next we apply the binomial lattice method into the abandonment problem with option for early exercise. The basic idea and steps are similar to the first case study with a small adjustments.

We initialize the lattice with n states according to the future value of cash flows from the project $x(0)$ where the value can go either up (3.7) or down 3.8) between discrete time steps $n-1$ with length $\Delta t = \frac{T}{n}$. Next the boundary condition (3.16d) at the terminal nodes is applied, given for the abandonment problem as

$$V_{i,j} = \max\{S - x_{i,j}\} \quad \forall (i+j) = n-1.$$

The second phase follows the principles of dynamic programming, where the optimal stopping time is determined by starting from the terminal time and advancing one step at time to the initial time by solving a subproblem at every step. The subproblem is the choice between continuing and abandoning the project. If the project is continued, the value is the same as in the first case study, that is, given by Equation 3.9. Then, if the project is abandoned, the project holder receives the payoff $\max\{S - V_{i-1,j-1}\}$. Hence the subproblem that is solved at every time step for the value maximizing decision-maker is given by

$$V^{int}_{i-1,j-1} = \max\left\{\max\left\{S - V_{i-1,j-1}\right\}, e^{-r\Delta t}\left(pV^{up}_{i,j-1} + (1-p)V^{down}_{i-1,j}\right)\right\},$$

which holds for all $(i+j) \leq n-1$. Consequently by moving backwards to the initial time $t = 0$ the value of the abandonment option can be determined. Note that even though the optimal price for the abandonment option is determined at the initial time, this implementation method does not naturally

the possibility to exercise the option at any given time, it would not be reasonable to exercise the option until the terminal time. Hence the set of equations would reduce to Equations 3.4a-3.4c, and thus a closed-form solution would exist. However, this would not hold if payments would be given during the life of an option. [76]

determine the free boundary in a strict sense. However, it can be constructed manually with different methods by examining the state values. For further details on this, see for example [42].

3.2.2.2 Finite difference method

Starting from the free boundary value problem (3.16a-3.16e), there are several ways to solve the problem. One possibility is to use front-fixing method where a change of variables is executed to convert the free boundary into a fixed boundary or penalty method where the free boundary eliminated by introducing a penalty term that essentially converts the inequalities into equalities [59]. Another option is to take the dynamic programming approach and neglect the free boundary for a moment. This will impose only few adjustments to the finite difference method implementation of the first case study and thus we will use it here.

We begin by initializing a discrete grid on a fixed domain where the cash flows are discretized with $0 = x_0 < x_1 < \ldots < x_N = X$ and time discretized with $0 = t_0 < t_1 < \ldots < t_M = T$. The discrete steps are given by $x_i = i\Delta x = i\frac{X}{N}$ and $t_j = j\Delta t = j\frac{T}{M}$. Hence we have a regular grid with discrete approximations $\tilde{V}_{i,j} \approx V(x_i, t_j)$. The boundary conditions for the abandonment problem in grid notation are given by

$$\begin{aligned}
\tilde{V}_{0,j} &= Se^{-r(T-t_j)} \quad j = 0, 1, \ldots, M \\
\tilde{V}_{i,M} &= \max\{S - x_i, 0\} \quad i = 0, 1, \ldots, N \\
\tilde{V}_{N,j} &= 0 \quad j = 0, 1, \ldots, M,
\end{aligned}$$

which are applied to the initialized grid.

We will employ only the implicit finite difference method for the abandonment problem. For the explicit method the concept is exactly the same as for the binomial lattice method, but with the implicit few additional modifications are required. At every time step we solve the subproblem whether to continue or abandon the project. If the project is continued, the value is given by Equation 3.13 and if the project is abandoned the payoff $\max\{S - x_i, 0\}$ is obtained. Hence, following the matrix notation given in Equation 3.14, we solve at every time step j the following equation:

$$\tilde{\mathbf{v}}_j = \max\left\{\max\left\{S - \mathbf{x}, 0\right\}, \mathbf{B}^{-1}(\tilde{\mathbf{v}}_{j+1} - \mathbf{k}_{j+1})\right\} \quad j = M-1, \ldots, 0.$$

Consequently we obtain the abandonment option value at the whole fixed domain.

Note that this requires that the matrix $\mathbf{B}$ is invertible, which is true for our case as the matrix $\mathbf{B}$ is tridiagonal. More robust and accurate method would be to use iterative methods such as projected successive over-relaxation (PSOR), which is commonly used in the implicit finite difference method, cf. [76].

Furthermore, note that by taking the dynamic programming approach to solve the option value, the abandonment boundary is not solved directly. We will construct an approximation $\tilde{x}^f$ for the abandonment boundary x^f a posteriori by solving the following equation:

$$\tilde{x}_j^f = \|\tilde{\mathbf{v}}_j - (S - \mathbf{x})\| < \varepsilon \quad \forall j$$

where the first element of $\tilde{\mathbf{v}}_j$ is chosen for which the condition holds. Here $\varepsilon > 0$ is used as the relaxation parameter that gives a reasonable approximation of the abandonment boundary when the parameter value is very small.

Chapter 4

Results

In this chapter we present numerical solutions for the case studies and analyze the results. In addition a comparison of numerical methods is performed. All of the numerical methods were implemented in MATLAB R2014b [52] numerical computing environment, see Appendix A for detailed code on the implementation procedures.

4.1 Investment option

We begin by choosing a reasonable set of parameters for the investment option. We assume that the investment period is ten years ($T = 10$) during which the risk-free interest rate is assumed to be low, three percent on average ($r = 0.03$), reflecting weak economic conditions. The future cash flows are assumed to be highly uncertain, and thus we adjust standard deviation to 35 percent ($\sigma = 0.35$). The current value of cash flows is assumed to be hundred million euros ($x(0) = 100$). The parameters are summarized in Table 4.1.

Table 4.1: Parameters used in the investment option case.

Parameter	Symbol	Value	Unit
Current CF from investment	$x(0)$	100	EURm
Fixed investment cost	I	150	EURm
Time to invest	T	10	years
S.d. of cash flows	σ	0.35	-
Risk-free interest rate	r	0.03	-

Plugging the given parameters into Equation 3.5 yields $V_{exact} \approx 38.9566$ as the closed-form solution, which we will use to benchmark the numerical results. Clear interpretation of this result is that a company should not pursue the project if there is an opportunity available with a greater value.

4.1.1 Monte Carlo simulation

We begin the analysis by running the Monte Carlo simulation with the parameters given in Table 4.1. Using a sample size of $n_{max} = 1.5 \cdot 10^6$ and 95% confidence level, the simulation yields

$$\tilde{V}_{MC}^{inv} = 39.1506 \pm 0.2207$$

as the value of investment option. We note that the value is reasonably close to the exact value. To investigate the convergence properties we run the simulation with smaller sample sizes, descending evenly to $n_{min} = 5 \cdot 10^4$. The results of the simulation are presented in Figure 4.1 along with the 95% confidence level.

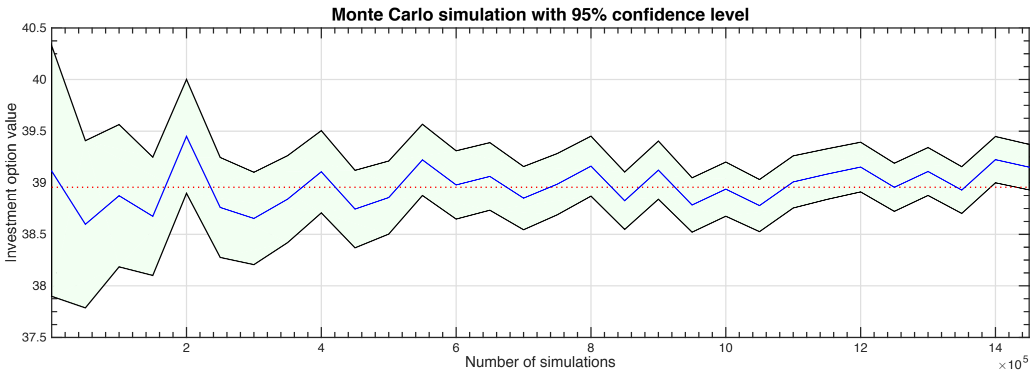


Figure 4.1: The value of the investment option (blue) and the 95% confidence level (light green area) with a Monte Carlo simulation in comparison to the analytical solution (red).

Examining Figure 4.1, we note that the 95% confidence level contracts as the number of simulations increases as expected, cf. Equation 3.6. Clearly, $\tilde{V}_{MC}^{inv} \to V_{exact}$ as $n_{max} \to \infty$. However, notice that the sample size for the Monte Carlo simulation should be sufficiently large to obtain accurate values as the rate of convergence is of order $\mathcal{O}\left(n^{-1/2}\right)$. The slow convergence effect is also clearly visible in Figure 4.1. Hence, even though the Monte Carlo simulation returns always reasonable values with absolutely contracting confidence level, the slow convergence can also be disadvantageous as in the investment option case where the dimension number is low.

4.1.2 Binomial lattice

Next we valuate the investment option with the binomial lattice model. Dividing the finite time horizon into $n_{max} = 150$ discrete time steps yields

$$\tilde{V}_{BL}^{inv} = 38.9688.$$

We examine the convergence properties by running simulations with different grid sizes $1 \leq n_{max}$. The simulation results are given in Figure 4.2, where the grid size was increased in every simulation by one.

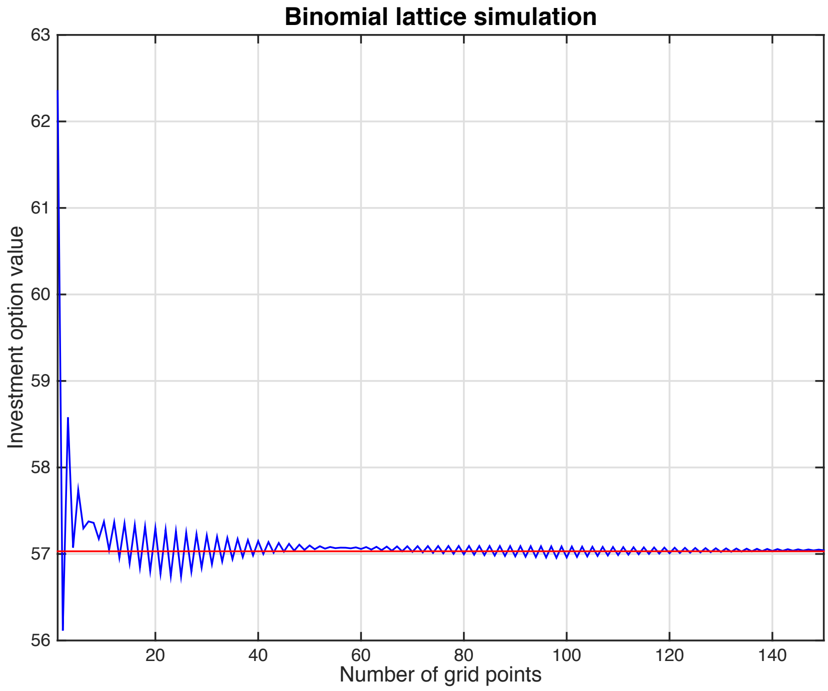


Figure 4.2: Value of the investment option (blue) in comparison to the analytical solution (red line) with different number of grid points.

We notice two distinctive trends in the convergence; a sawtooth and oscillatory effect. That is, the convergence is oscillatory and non-monotonic. The sawtooth effect is commonly referred as distribution error in corresponding literature. This arises due to the fact that a discrete binomial probability is used to approximate the log-normal distribution of the stochastic diffusion process [26].

The oscillatory effect, more commonly known as the non-linearity error, can be seen as a periodic expansion of the investment option value which diminishes as number of grid points are added. The effect arises only when the current position differs from the at the money position, that is $x(0) \neq I$. Hence the non-linearity error emerges due to large discontinuity in the terminal region between the current value and investment cost [26].

Thus by comparing the convergence effects, we note that the distribution error is not that harmful as by increasing the number of steps the error diminishes. However, the non-linearity error is quite negative aspect of the method as the derived value of the option might be far away from the exact value even with a large number of steps. Leisen et al. [48] showed that the convergence rate for the CRR model is $\mathcal{O}(n^{-1})$, which is quite poor. However, several different solutions have been proposed, e.g. replacing values at the end nodes with the investment cost values [35], smoothing out the payoff functions to maturity [35], or by using adaptive meshes [26].

4.1.3 Finite difference method

Finally, we solve the investment option case with finite difference method. We derive a numerical solution both with the explicit and implicit interpolation scheme to compare the efficiency of the schemes. In addition to parameters listed in Table 4.1, we have to set additional parameters for the grid. Limiting the domain to $X = 900$ with $N = 250$ nodes and using $M = 10^5$ time steps, we obtain:

$$\tilde{V}^{inv}_{FDM,exp} = 38.9567, \quad \tilde{V}^{inv}_{FDM,imp} = 38.9565.$$

Comparing the values to the exact solution, we note that the values are very close to the exact solution with both methods. Moreover, note that the investment is valuated naturally in the whole domain with finite difference method, which is not possible for example with the Monte Carlo method due to path independency. Examples of resulting solution surfaces for the investment option are given in Figures 4.4.

Adjusting the grid size has a significant effect on the accuracy as we note from Table 4.2, which shows the results from additional simulations with different grid sizes. Comparing the option values, we note that the error in both methods is approximately the same and decreases rapidly as the length of the time step decreases. Corresponding error plot of the values in log-log scale is given in Figure 4.3. We note that the convergence of both methods follows approximately the expected the convergence rate, which is determined from the difference approximations of the partial derivatives, giving $\mathcal{O}(\Delta t)$ and $\mathcal{O}(\Delta x^2)$ for both methods [46]. However, by running simulations with various combinations of grid parameters we notice that the explicit method is very sensitive to chosen grid parameters as the explicit method does not convergence constantly.

Table 4.2: Comparison of the explicit and implicit FDM with respect to the absolute error for the case study.

	FDM Explicit		FDM Implicit	
Δt	Value	Error	Value	Error
0.1	39.691497763	0.734869664	39.545121521	0.588493422
0.01	39.124475131	0.167847031	39.106936177	0.150308077
0.001	38.937438616	0.019189482	38.935679028	0.020949070
0.0001	38.956664454	0.000036355	38.956488743	0.000139355

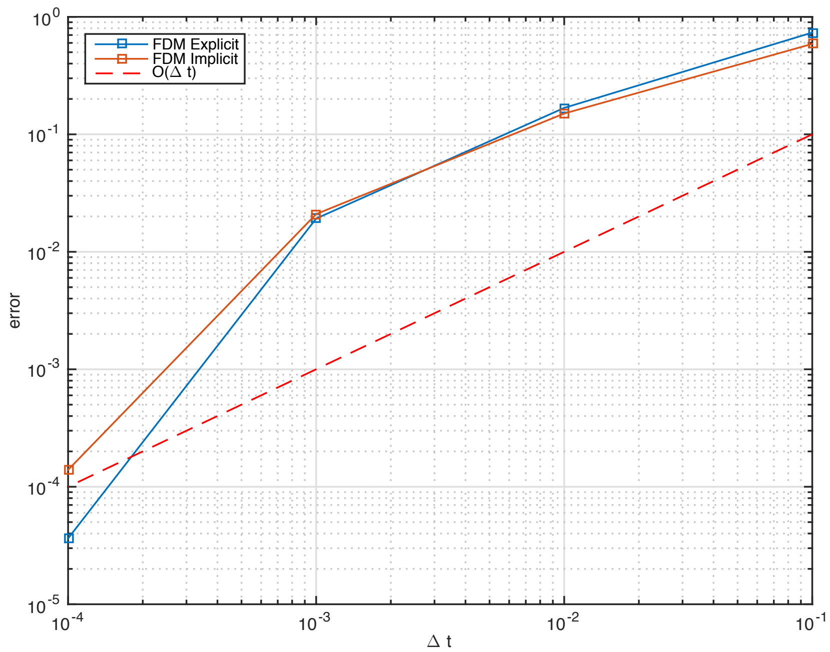


Figure 4.3: Absolute error for the case study in log-log scale for the explicit and implicit FDM according to Table 4.2.

Examining the matrix formulations for the explicit and implicit method (3.11, 3.14), we note that the stability and convergent requirements are

$$\|\mathbf{A}\|_\infty \leq 1, \quad \|\mathbf{B}^{-1}\|_\infty \leq 1$$

for the explicit and implicit scheme, respectively, where we denote $\|\cdot\|_\infty$ as the maximum norm[1]. Note that these conditions hold for all the possible differential equations that are approximated with the explicit or implicit finite difference method. Considering the investment option case, the stability condition for the differential equation of the case equation (3.4a) can be written as

$$\Delta t \leq \frac{1}{\sigma^2 N^2}.$$

This poses serious limitations to the explicit method as with small values of volatility σ the time step size should be very small. Unlike the explicit method, the implicit method is always stable and convergent as the maximum eigenvalue is always less than one. For further details on the stability and convergence of finite difference method, see for example [11], [46].

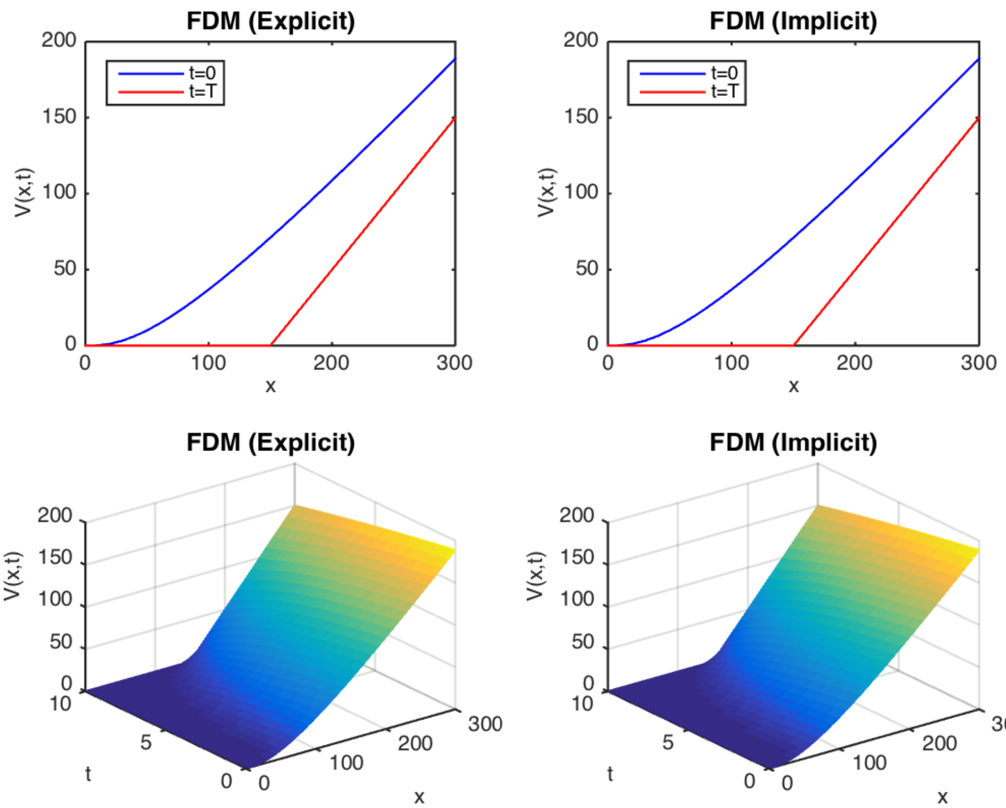


Figure 4.4: Solution surface for the investment option with the explicit and implicit FDM with given case parameters in Table 4.1 and grid parameters $X = 300, N = 30, M = 1000$.

[1]For $\mathbf{x} \in \mathbb{R}^N$, the maximum norm is defined as $\|\mathbf{x}\|_\infty = \max_{i \leq i \leq N} |x_i|$ and for $\mathbf{X} \in \mathbb{R}^{NxM}$, we have $\|\mathbf{X}\|_\infty = \max_{i \leq i \leq N} \sum_{j=1}^{n} |x_{ij}|$.

4.1.4 Comparison of numerical methods

Generally, it is rather difficult to rigorously compare different types of numerical methods as they are constructed in a different way. Using the accuracy of the numerical solution as only metric is problematic since by increasing the number of iterations by one step does not equal increasing the grid size by one node. In addition, the requirements for a numerical method vary by user, e.g. a company might seek fast and easily implementable solution while a researcher may seek as accurate solution as possible.

To investigate these two dimensions, we run additional simulations with the case study parameters given in Table 4.1 with varying requirements for the simulation. First we seek solutions with a condition that the absolute error is smaller than 0.1 percent. Then we examine what is the accuracy of the numerical methods when the wall-clock time on simulation is fixed approximately to one second. The results of the simulations are given in Table 4.3 and illustrated graphically in Figure 4.5.

Table 4.3: Comparison of the numerical methods for the investment case with respect to fixed absolute error and wall-clock time in seconds. The inputs correspond the random sample size for the Monte Carlo method (MC), number of time steps for binomial lattice method (BL) and the grid size with notation (N, M) for fixed $X = 900$ for both finite difference methods (FDM). The upper value from 95% confidence level was used in the MC comparison.

	MC	BL	FDM Exp.	FDM Impl.
$\frac{\lVert\tilde{V}-V\rVert}{V} \approx 0.001$				
Input	$100 \cdot 10^6$	88	$(80, 9000)$	$(80, 9000)$
$\tilde{V}$	38.99119	38.99474	38.97428	38.97232
Clock time	46.86585	0.001758	0.368636	0.397352
Clock time ≈ 1				
Input	$13 \cdot 10^6$	1800	$(90, 15000)$	$(80, 9000)$
$\tilde{V}$	39.02310	38.95732	38.93714	38.93597
$\frac{\lVert\tilde{V}-V\rVert}{V}$	0.001706	0.000018	0.000500	0.000500

We notice that the binomial lattice method outperforms all the other methods in both scenarios in the case study with given parameters. With the fixed limit for the absolute error, the performance for the Monte Carlo

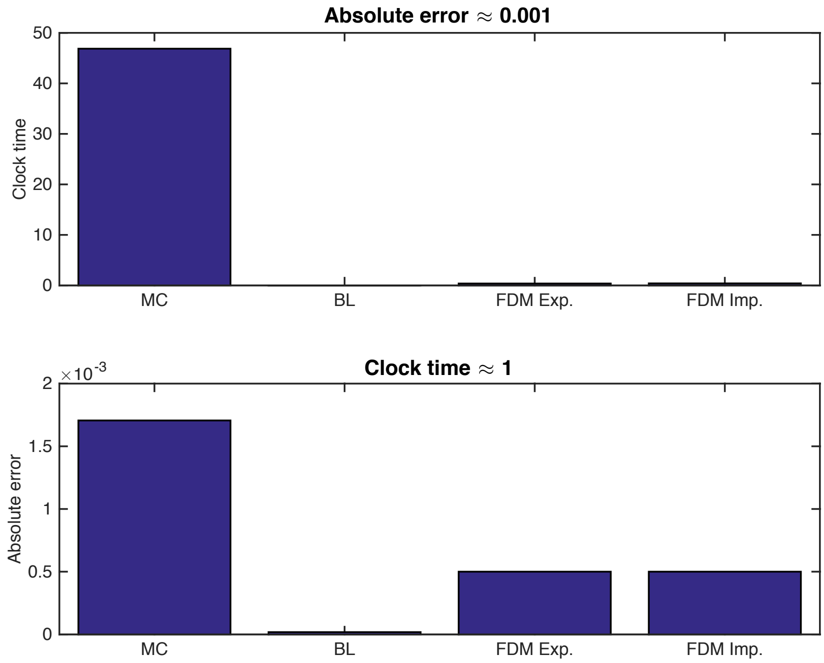


Figure 4.5: Comparison of the performance of numerical methods in the case study according to Table 4.3, where unit of the simulation clock time is seconds.

method is clearly the worse as the clock time is close to a minute while other are able to perform calculation under half a second. When the clock time is fixed to one second, the absolute error is very low with all the methods but once again the performance of the Monte Carlo method is the weakest. Overall, there does not seem to be that significant difference between FDM with the explicit and implicit scheme. This does not however remove the weak convergence properties for the explicit method.

In addition, note that the comparison given in Table 4.3 is not by any means complete comparison of the numerical methods as only one set of equations derived from the investment case was examined. Note also that the outputs for the simulation methods are different, e.g. finite difference methods calculate the solution in the whole domain while the output for the Monte Carlo method and binomial lattice method is merely a scalar value for the option. Furthermore, there is plenty of room for improving the numerical procedures implemented into computer environment as the code is not fully optimized for fast calculation.

Summary of the key features of the numerical methods is given in Table 4.4. The performance estimates are based on the previous discussion within this study. Note that most of the features examined in the case study can be considered global, as they are mostly independent on the set of equations to be solved.

Table 4.4: General overview of numerical methods' capabilities, where the excellence of a feature is directly proportional to the filled area of a pie. The features that are also applicable outside the investment case are denoted with a dagger symbol.

	MC	BL	FDM Exp.	FDM Imp.
Case study				
Implementability†	7/8	3/4	3/8	1/4
Clock time†	1/8	7/8	5/8	5/8
Accuracy	1/8	7/8	3/4	3/4
Convergence†	1/8	3/8	3/4	3/4
Stability†	1	1	1/8	1
Extensions				
Inequalities	1/8	5/8	3/4	3/4
High dimensions	1	1/4	1/4	1/4
Boundary conditions	1/8	3/8	1/2	1/2

In addition to the case study comparison in Table 4.4, we consider the competence of the numerical methods with possible extensions outside the investment case. Inequalities can be used to allow an early exercise for an investment option, which we will examine more closely with the following case study. Allowing an early exercise for an option is particularly difficult in the Monte Carlo method due to path independency in comparison to other methods where the condition can be simply applied at every node of lattice or grid. However, in higher dimensions the Monte Carlo method excels over the other methods as the rate of convergence is independent on the dimension number. The difficulty of setting a different boundary condition is hard to estimate as it is clearly highly dependent on the boundary condition to be set. For example, changing the option to buy an investment to sell is very easily implementable with all the methods but implementing more free boundaries on the domain can be a daunting task especially for the basic Monte Carlo method due to path independency. Comparing the binomial lattice and the finite difference method, the latter is more flexible on different types of boundary conditions due to mathematical nature of deriving the numerical scheme. However, some boundaries, such as an infinite boundary condition, are still very difficult implement with the finite difference method.

Naturally, there are also other numerical methods available for solving a differential equation arising from the real options analysis in addition the numerical methods presented so far in this study. One particularly important is finite element method (FEM), which has been used very successively within the field of engineering for decades. The main idea for FEM is the same as for the FDM, that is, approximate the solution of a differential equation with a discrete solution in the whole domain. However, the method of obtaining the discretization and the discrete solution are a bit more elegant[2] in FEM. Even though the methods are similar with regards to idea, there are several advantages and disadvantages between the FDM and FEM. Some advantages in FEM are arbitrary and adaptive geometry of the grid, nearly arbitrary boundary conditions that are easily implemented, reduced requirements on the regularity of the solution and more robust convergence and error estimates. However, there is no free lunch in numerical methods. Comparing to the FDM and other numerical methods presented in this study, the implementation of FEM is rather difficult task and it can be regarded as an overkill to some of the problems within real options analysis. This is probably the main reason why FEM has not been used that often in the field of finance and economics. However, the flexibility it provides through nearly arbitrary grid and boundary conditions is enormous. This cannot be underestimated when considering the increasing complexity of the economic world.

4.2 Abandonment option

Next we turn into our second case study and present the results for the abandonment option. Let us suppose that the projected cash flows from a project are three hundred million euros ($x(0) = 300$) with a standard deviation of 30 percent ($\sigma = 0.3$). We assume that option to abandon the

[2] In FEM the solution of a differential equation is approximated with a simple algebraic equations. To obtain this, three steps are usually required. First, the differential equation has to be transformed into a variational formulation, i.e. the weak form of the boundary value problem. Then the initial domain is discretized into a mesh consisting of elements, such as triangles or cubes, and nodes connecting the elements. Finally the approximation of the infinite dimensional differential equation is calculated for the discretized variational formulation in a finite dimensional subspace consisting of piecewice polynomials that are connected over the elements. For further details on FEM, see for example [71], [40] and [14].

project is valid for five years ($T = 5$) with a fixed salvage value two hundred million euros ($S = 200$). Finally, let the risk-free interest rate from the market be three percent ($r = 0.03$). Summary of the parameters is given in Table 4.5.

Table 4.5: Parameters used in the abandonment option case.

Parameter	Symbol	Value	Unit
Projected CF from project	$x(0)$	300	EURm
Salvage value	S	200	EURm
Time to abandon	T	5	years
S.d. of cash flows	σ	0.3	-
Risk-free interest rate	r	0.03	-

As noted in Chapter 3, a closed-form solution for the abandonment problem does not exist and thus numerical methods are required to derive a solution.

4.2.1 Numerical results

Following the discussion and results in the previous section where numerical methods were compared, we derive a solution with the binomial lattice and implicit finite difference method. The reasoning behind choosing two methods instead of one is that we are able to compare and verify the numerical results as analytical solution is not available.

Guided by the previous discussion and results, we choose grid size of $n_{max} = 500$ for the binomial lattice method that should be sufficient for the problem at hand. Along with the given case parameters in Table 4.5, the binomial lattice method valuates the abandonment option to

$$\tilde{V}_{BL}^{Ab} = 18.1683.$$

Note that the derived value represents the value of the abandonment option at the beginning of the project and thus does not provide relevant information during the project. That is, it can be used only to verify is it reasonable to sign a contract with the abandonment option. Clearly, if the cost of option exceeds the value of the option one should not sign the contract. The convergence of the binomial lattice with increasing number of

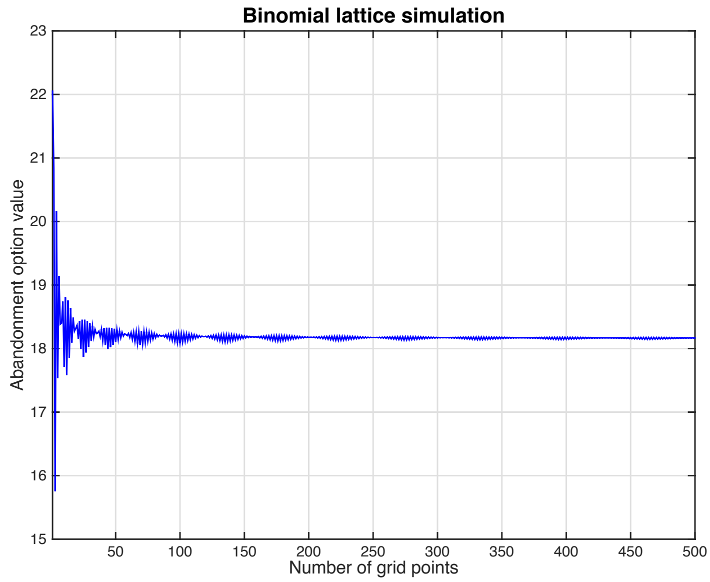


Figure 4.6: Convergence of the binomial lattice method on the abandonment option.

discrete time steps is given in Figure 4.6 to verify that the derived value is within the ballpark.

To further verify the result and obtain additional information on the abandonment option, we solve the corresponding problem with the implicit finite difference method, which was chosen over the explicit method due to superior convergence properties. Limiting the domain to $X = 900$ with $N = 200$ nodes and using $M = 300$ time steps yields

$$\tilde{V}^{Ab}_{FDM,imp} = 18.1240.$$

First of all, we note the value is relatively close to the value derived with the binomial lattice method verifying the both results to some extent. In addition to scalar value of the option at the beginning of project, we are able to examine the corresponding solution surface which is given in Figure 4.7. Note that as the projected cash flow from the project (x) increases, the value of the option decreases as the salvage value is constant.

Moreover, the three dimensional graph reveals the value of the option at every given point of time. This feature can be used to determine the optimal abandonment boundary, which is given in Figure 4.8 with relaxation parameter $\epsilon = 0.001$. The graph reveals directly whether the ongoing project should be abandoned or not, e.g. if the expected cash flows from the project decline from 150 to 130 million euros after three years due to external conditions, such as emerging of new competitors to the market, then the optimal decision is to abandon the project. The information obtained from the optimal abandonment boundary is crucial for any decision-maker holding an option to abandon a project.

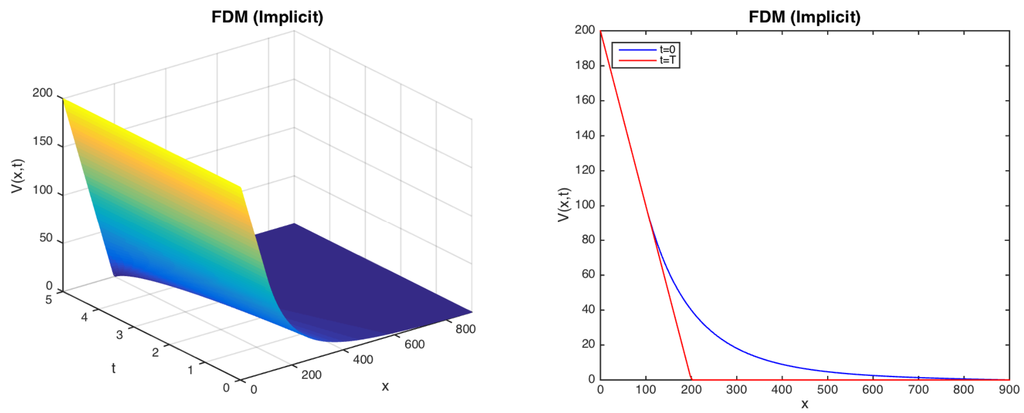


Figure 4.7: Solution surfaces for the abandonment option derived with the implicit finite difference method with case parameters given in Table 4.5 and grid parameters $X = 900$, $N = 200$ and $M = 300$.

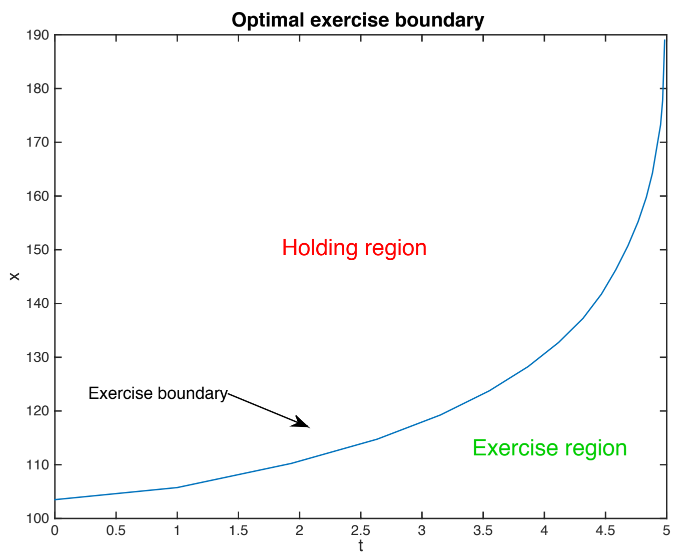


Figure 4.8: Optimal abandonment boundary for the abandonment option with relaxation parameter $\epsilon = 0.001$ and grid parameters $X = 900$, $N = 200$ and $M = 300$.

Chapter 5

Conclusions

In this study we examined different numerical solution methods that can be used to solve differential equations arising from real options analysis and presented two case studies where the discussed numerical methods were applied. a solution to a case study within real options theory for which a closed-form solution does not exist.

We begun by investigating different methods that are commonly used to validate decisions under uncertainty. All of the discussed methods lack one crucial property – an option delay the investment. This feature is especially important in the current economy, where the investment periods are typically long and highly uncertain. One possible method that suits perfectly into the situation is real options analysis, which can be considered as an extension of financial option theory into real assets. Technically the idea is to use a stochastic variable and determine the real option value from a differential equation.

Following a brief literature review on real options, we investigated the differences in real option formulations between an infinite and finite time horizon problems. While the infinite time horizon cases can usually be solved analytically, the suitability to different real-life situations is highly questionable. Investment problems with finite time horizon typically lead to a set of equations for which a closed-form does not exist. Examining the literature on real options, we concluded that there are several other problems within real options theory that can not be solved analytically in addition to finite time horizon problems. Hence numerical methods are required to obtain a solution.

We introduced three numerical methods that are commonly used within real options theory: the Monte Carlo method, binomial lattice method and finite difference method with explicit and implicit scheme. Then we presented two case studies: investment and abandonment option. In both cases the cash flows from the project were uncertain and the time horizon was finite, but in the latter case there was an option to abandon the project at any given time during the project. All numerical procedures were implemented into MATLAB computing environment.

Investment option case was used to compare the numerical methods presented and benchmark the solutions against the closed-form solution which existed for the first case study. All of the numerical methods were suitable for the case study as the numerical solutions were sufficiently close to the exact solution. However, there were clear differences between the methods. Obtaining the solution with the Monte Carlo method was inaccurate and very slow with respect to the wall-clock time in comparison to other numerical methods. The performance of the explicit and implicit finite difference method was similar, but the stability of the explicit method was extremely sensitive to the grid parameters. Considering only the value of the option at the beginning of project as the only requirement, the binomial lattice model outperformed the other numerical methods in every aspect excluding the convergence, which was relatively slow and presented some negative features. Remark that with finite difference method the whole time domain is solved at once in roughly the same time as only single value with binomial lattice and thus finite difference method provides substantial benefits over the binomial lattice model. Considering extensions outside the investment option case, the Monte Carlo methods suits perfectly into problems with higher dimensions since the rate of convergence stays at the same level. With other extensions, such as inequalities and more complex boundary conditions, we suggest using the implicit finite difference method from the numerical methods examined due to the stability and convergence properties and the reasonable flexibility over the boundary conditions.

Abandonment option case was used to illustrate the effectiveness and usability of numerical methods in solving a differential equation for which a closed-form solution does not exist. The abandonment option was solved with both the binomial lattice and implicit finite difference method to verify the results against each other. In addition, optimal abandonment boundary was derived from the finite difference method solution as the solution is directly calculated in the whole time domain. The derived boundary is extremely useful for any decision-maker involved in a project with an option to abandon the project.

Overall, the implicit finite difference methods proved to be the most suitable for the discussed case studies and further possible extensions. For further research, we recommend to investigate the performance of the three numerical methods along with their extensions with additional case studies for which a closed-form solution does not exist. Particularly interesting would be to examine the feasibility of different boundary conditions with the given methods. In addition, we suggest looking also into completely new approaches, such as the finite element method.

Bibliography

[1] Abadie, L. M. Valuation of long-term investments in energy assets under uncertainty. *Energies 2*, 3 (2009), 738–768.

[2] Andalaft-Chacur, A., Montaz Ali, M., and González Salazar, J. Real options pricing by the finite element method. *Computers & Mathematics with Applications 61*, 9 (2011), 2863–2873.

[3] Awerbuch, S., Dillard, J., Mouck, T., and Preston, A. Capital budgeting, technological innovation and the emerging competitive environment of the electric power industry. *Energy Policy 24*, 2 (1996), 195–202.

[4] Azevedo, A., and Paxson, D. Developing real option game models. *European Journal of Operational Research 237*, 3 (2014), 909–920.

[5] Bellman, R. E. *Eye of the Hurricane: an autobiography.* World Scientific Singapore, 1984.

[6] Bjerksund, P., and Stensland, G. Closed-form approximation of american options. *Scandinavian Journal of Management 9* (1993), S87–S99.

[7] Black, F., and Scholes, M. The pricing of options and corporate liabilities. *The journal of political economy* (1973), 637–654.

[8] Boyle, P., Broadie, M., and Glasserman, P. Monte carlo methods for security pricing. *Journal of economic dynamics and control 21*, 8 (1997), 1267–1321.

[9] Boyle, P. P. Options: A monte carlo approach. *Journal of financial economics 4*, 3 (1977), 323–338.

[10] Boyle, P. P. A lattice framework for option pricing with two state variables. *Journal of Financial and Quantitative Analysis 23*, 01 (1988), 1–12.

[11] Brandimarte, P. *Numerical methods in finance and economics: a MATLAB-based introduction*. John Wiley & Sons, 2013.

[12] Brennan, M. J., and Schwartz, E. S. Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial and Quantitative Analysis 13*, 03 (1978), 461–474.

[13] Brennan, M. J., and Schwartz, E. S. Evaluating natural resource investments. *Journal of business* (1985), 135–157.

[14] Brenner, S. C., and Scott, R. *The mathematical theory of finite element methods*, vol. 15. Springer, 2008.

[15] Broadie, M., and Glasserman, P. Pricing american-style securities using simulation. *Journal of Economic Dynamics and Control 21*, 8 (1997), 1323–1352.

[16] Carr, P. The valuation of american exchange options with application to real options. *Real options in capital investment: models, strategies and applications* (1995), 109–120.

[17] Cox, J. C., Ross, S. A., and Rubinstein, M. Option pricing: A simplified approach. *Journal of financial Economics 7*, 3 (1979), 229–263.

[18] David, F. N. Games, gods and gambling: The origins and history of probability and statistical ideas from the earliest times to the newtonian era.

[19] De Reyck, B., Degraeve, Z., and Vandenborre, R. Project options valuation with net present value and decision tree analysis. *European Journal of Operational Research 184*, 1 (2008), 341–355.

[20] Deng, S.-J., Johnson, B., and Sogomonian, A. Exotic electricity options and the valuation of electricity generation and transmission assets. *Decision Support Systems 30*, 3 (2001), 383–392.

[21] Dixit, A. Entry and exit decisions under uncertainty. *Journal of political Economy* (1989), 620–638.

[22] Dixit, A., and Pindyck, R. *Investment Under Uncertainty*. Princeton University Press, 1994.

[23] Duffy, D. J. *Finite Difference methods in financial engineering: a Partial Differential Equation approach*. John Wiley & Sons, 2006.

[24] Fama, E. F., and French, K. R. The capital asset pricing model: theory and evidence. *Journal of Economic Perspectives* (2004), 25–46.

[25] Felder, F. A. Integrating financial theory and methods in electricity resource planning. *Energy policy 24*, 2 (1996), 149–154.

[26] Figlewski, S., and Gao, B. The adaptive mesh model: a new approach to efficient option pricing. *Journal of Financial Economics 53*, 3 (1999), 313–351.

[27] Fischer, E. O. Analytic approximation for the valuation of american put options on stocks with known dividends. *International Review of Economics & Finance 2*, 2 (1993), 115–127.

[28] Fowler, D., and Robson, E. Square root approximations in old babylonian mathematics: Ybc 7289 in context. *Historia Mathematica 25*, 4 (1998), 366–378.

[29] Galai, D., and Masulis, R. W. The option pricing model and the risk factor of stock. *Journal of Financial economics 3*, 1 (1976), 53–81.

[30] Garlappi, L. Risk premia and preemption in r&d ventures. *Journal of Financial and Quantitative Analysis 39*, 04 (2004), 843–872.

[31] Geske, R. The valuation of compound options. *Journal of financial economics 7*, 1 (1979), 63–81.

[32] Geske, R., and Johnson, H. E. The american put option valued analytically. *The Journal of Finance 39*, 5 (1984), 1511–1524.

[33] Glasserman, P. *Monte Carlo methods in financial engineering*, vol. 53. Springer, 2004.

[34] Grenadier, S. R., and Weiss, A. M. Investment in technological innovations: An option pricing approach. *Journal of financial Economics 44*, 3 (1997), 397–416.

[35] Heston, S., and Zhou, G. On the rate of convergence of discrete-time contingent claims. *Mathematical Finance 10*, 1 (2000), 53–75.

[36] Hodder, J. E., and Riggs, H. E. Pitfalls in evaluating risky projects. *Harvard Business Review 63*, 1 (1985), 128–135.

[37] Hodges, A. *Alan Turing: The Enigma.* Random House, 1992.

[38] Hsu, M. Spark spread options are hot! *The Electricity Journal 11*, 2 (1998), 28–39.

[39] Jain, S., Roelofs, F., and Oosterlee, C. W. Valuing modular nuclear power plants in finite time decision horizon. *Energy Economics 36* (2013), 625–636.

[40] Johnson, C. *Numerical solution of partial differential equations by the finite element method.* Courier Dover Publications, 2012.

[41] Jun, D., and Ku, H. Analytic solution for american barrier options with two barriers. *Journal of Mathematical Analysis and Applications 422*, 1 (2015), 408–423.

[42] Kim, I. J., and Byoon, S. J. Optimal exercise boundary in a binomial option pricing model. *Available at SSRN 5373* (1994).

[43] Korn, R., and Korn, E. Option pricing and portfolio optimization: Modern methods of financial mathematics, graduate studies in mathematics. In *Amer. Math. Soc* (2001).

[44] Korn, R., and Müller, S. Binomial trees in option pricing – history, practical applications and recent developments. In *Recent Developments in Applied Probability and Statistics.* Springer, 2010, pp. 59–77.

[45] Lander, D. M., and Pinches, G. E. Challenges to the practical implementation of modeling and valuing real options. *The Quarterly Review of Economics and Finance 38*, 3 (1998), 537–567.

[46] Larsson, S., and Thomée, V. *Partial differential equations with numerical methods*, vol. 45. Springer Science & Business, 2008.

[47] Laurikka, H., and Koljonen, T. Emissions trading and investment decisions in the power sector – a case study in finland. *Energy Policy 34*, 9 (2006), 1063–1074.

[48] Leisen, D. P., and Reimer, M. Binomial models for option valuation-examining and improving convergence. *Applied Mathematical Finance 3*, 4 (1996), 319–346.

[49] LeVeque, R. J. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, vol. 98. Siam, 2007.

[50] Margrabe, W. The value of an option to exchange one asset for another. *The journal of finance 33*, 1 (1978), 177–186.

[51] Marreco, J. d. M., and Carpio, L. G. T. Flexibility valuation in the brazilian power system: A real options approach. *Energy Policy 34*, 18 (2006), 3749–3756.

[52] MATLAB. *version 8.4.0 (R2014b)*. The MathWorks Inc., Natick, Massachusetts, 2014.

[53] Merton, R. C. Theory of rational option pricing. *The Bell Journal of economics and management science* (1973), 141–183.

[54] Metropolis, N. The beginning of the monte carlo method. *Los Alamos Science 15*, 584 (1987), 125–130.

[55] Mun, J. *Real Options Analysis: Tools and Techniques for Valuing Strategic Investments and Decisions*, vol. 137. John Wiley & Sons, 2002.

[56] Myers, S. C. Determinants of corporate borrowing. *Journal of financial economics 5*, 2 (1977), 147–175.

[57] Myers, S. C. Finance theory and financial strategy. *Interfaces 14*, 1 (1984), 126–137.

[58] Myers, S. C., Majd, S., et al. Calculating abandonment value using option pricing theory.

[59] Nielsen, B. F., Skavhaug, O., and Tveito, A. Penalty and front-fixing methods for the numerical solution of american option problems.

[60] Paddock, J. L., Siegel, D. R., and Smith, J. L. Option Valuation of Claims on Real Assets: The Case of Offshore Petroleum Leases. *The Quarterly Journal of Economics 103*, 3 (August 1988), 479–508.

[61] Paxson, D. *Real R & D Options*. Butterworth-Heinemann, 2002.

[62] Pindyck, R. S. Investments of uncertain cost. *Journal of financial Economics 34*, 1 (1993), 53–76.

[63] Pringles, R., Olsina, F., and Garcés, F. Real option valuation of power transmission investments by stochastic simulation. *Energy Economics* (2014).

[64] Rogers, L. C. Monte carlo valuation of american options. *Mathematical Finance 12*, 3 (2002), 271–286.

[65] Schwartz, E. S., and Moon, M. Rational pricing of internet companies revisited. *Financial Review 36*, 4 (2001), 7–26.

[66] Siddiqui, A., and Takashima, R. Capacity switching options under rivalry and uncertainty. *European Journal of Operational Research 222*, 3 (2012), 583–595.

[67] Siegel, D. R., Smith, J. L., and Paddock, J. L. Valuing offshore oil properties with option pricing models. *Midland Corporate Finance Journal 5*, 1 (1987), 22–30.

[68] Stulz, R. Options on the minimum or the maximum of two risky assets: analysis and applications. *Journal of Financial Economics 10*, 2 (1982), 161–185.

[69] Thomée, V. From finite differences to finite elements: A short history of numerical analysis of partial differential equations. *Journal of Computational and Applied Mathematics 128*, 1 (2001), 1–54.

[70] Thompson, M., and Barr, D. Cut-off grade: A real options analysis. *Resources Policy 42* (2014), 83–92.

[71] Topper, J. *Financial engineering with finite elements*. Wiley finance series. Wiley, 2005.

[72] Tourinho, O. A. The valuation of reserves of natural resources: an option pricing approach.

[73] Trigeorgis, L., and Mason, S. P. Valuing managerial flexibility. *Midland corporate finance journal 5*, 1 (1987), 14–21.

[74] van Zee, R. D., and Spinler, S. Real option valuation of public sector r&d investments with a down-and-out barrier option. *Technovation* (2013).

[75] Weeds, H. Strategic delay in a real options model of r&d competition. *The Review of Economic Studies 69*, 3 (2002), 729–747.

[76] Wilmott, P. *The mathematics of financial derivatives: a student introduction.* Cambridge University Press, 1995.

[77] Wilmott, P. *Paul Wilmott on Quantitative Finance, 3 Volume Set.* John Wiley & Sons, 2007.

Appendix A

MATLAB Code

Here we present the MATLAB code that was used for solving both the investment and abandonment option numerically. The code includes the Monte Carlo method, binomial lattice method, and finite difference method with both the explicit and implicit scheme for the investment option case and the binomial lattice and implicit finite difference method for the abandonment option case.

```
% Arto Sorsimo | arto.sorsimo@aalto.fi
% Aalto University School of Business
% Helsinki, 19.1.2015
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % COMPARISON OF NUMERICAL METHODS IN ROA   %
% % Case study #1                             %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This file solves the case study equation with the Monte Carlo,
% binomial lattice and finite difference method.
% The case problem is the following:
%
% { 1/2*sigma^2*x^2*V_xx + rxV_x - rV + V_t = 0,
% { V(0,t) = 0,
% { V(x,T) = max{x-I,0},
%
% where
%
% V: value of the investment option,
% x: stochastic variable (e.g. cash flows),
% I: fixed investment cost,
% T: terminal time,
% r: risk-free interest rate,
% sigma: standard deviation of stochastic variable.
```

```
%
% The problem is similar to European call option.
%
% This code is not fully optimized to be efficient.
% Use with care.

clear all;
close all;

%% Global parameters
x0 = 100;
I = 150;
r = 0.03;
sigma = 0.35;
T = 10;

%% Closed-form solution
t=0;

d_1 = 1/(sigma*sqrt(T-t)) * (log(x0/I)+(r+sigma^2/2)*(T-t));
d_2 = d_1 - sigma*sqrt(T-t);

Phi_1 = 1/2 * (1+erf(d_1/sqrt(2)));
Phi_2 = 1/2 * (1+erf(d_2/sqrt(2)));

V_exact = Phi_1*x0 - Phi_2*I*exp(-r*(T-t));

display(V_exact);

%% MONTE CARLO METHOD
nMC = 1e6;
data_interval = 50000;

% Choose whether to plot or not (yes=1, no=0)
PLOTTING_MC = 1;

if PLOTTING_MC == 0
    data_interval = nMC;
end

sim_vec = zeros(4,nMC/data_interval);
sim_vec(1,:) = (1:data_interval:nMC);

j=1;
for k=data_interval:data_interval:nMC
    RandVec = randn(k,1);

    % Calculate terminal value
    x_term = x0*exp((r-0.5*sigma^2)*T + sigma*sqrt(T)*RandVec);
```

```
    % Apply the BCs, calculate the mean of the sample
    % and boundaries for 95% confidence level
    V_i = exp(-r*T)*max(x_term-I,0);
    V_est = mean(V_i);

    V_sd = std(V_i);
    V_conf_up = V_est + 1.96*V_sd/sqrt(k);
    V_conf_low = V_est - 1.96*V_sd/sqrt(k);

    % Save the data to simulation vector
    sim_vec(2,j) = V_est;
    sim_vec(3,j) = V_conf_up;
    sim_vec(4,j) = V_conf_low;

    j = j+1;
end

if PLOTTING_MC == 1
    hFig = figure;
    set(hFig, 'Position', [300 300 1000 300])
    tempX = [sim_vec(1,:), fliplr(sim_vec(1,:))];
    tempY = [sim_vec(4,:),fliplr(sim_vec(3,:))];
    fill(tempX,tempY,[0 1 0], 'FaceAlpha', 0.05);

    xlim([sim_vec(1) sim_vec(1, end)]);

    set(gca,'XMinorTick','on','YMinorTick','on')
    grid on;

    hTitle = title(['Monte Carlo simulation with ' ...
                                 '95% confidence level']);
    hXLabel = xlabel('Number of simulations');
    hYLabel = ylabel('Investment option value');

    set(gca,'FontName','Helvetica');
    set([hTitle, hXLabel, hYLabel],'FontName','Helvetica');
    set([hXLabel, hYLabel],'FontSize',12);
    set( hTitle,'FontSize',14,'FontWeight','bold');

    hold on;
    plot(sim_vec(1,:),V_exact*ones(nMC/data_interval,1), ':r');

    hold on;
    plot(sim_vec(1,:),sim_vec(2,:),'b');
end

V_MonteCarlo = [V_conf_low V_est V_conf_up];
display(V_MonteCarlo);
```

```
relerror_mc = (V_conf_up - V_exact)/V_exact;
display(relerror_mc);

%% BINOMIAL LATTICE METHOD
nBL = 180;
data_interval = 1;

% Choose whether to plot or not (yes=1, no=0)
PLOTTING_BL = 1;

if PLOTTING_BL == 0
    data_interval = nBL;
end

sim_vec = zeros(3,nBL/data_interval);
sim_vec(1,:) = (1:data_interval:nBL);

count = 1;
for m=data_interval:data_interval:nBL

    dt = T/m;
    u = exp(sigma*sqrt(dt));
    d = exp(-sigma*sqrt(dt));
    p = (exp(r*dt)-d)/(u-d);

    % Initialize the lattice matrix and fill the first row
    % with up values
    V_lattice_fw = zeros(m);
    for k=1:m+1
        V_lattice_fw(1,k) = x0*u^(k-1);
    end

    % Calculate down values column-by-column
    for k=1:m
        for j=1:m
            V_lattice_fw(1+k,j+1) = V_lattice_fw(k,j)*d;
        end
    end

    % Apply call/put boundary condition at the terminal time
    V_lattice_bw = zeros(size(V_lattice_fw));
    V_lattice_bw(:,m+1) = max(V_lattice_fw(:,m+1)-I,0);

    % Calculate previous values with backward induction
    for j=1:m
        for k=1:m
            V_lattice_bw(k,m+1-j) = (p*V_lattice_bw(k,m+2-j) ...
                +(1-p)*V_lattice_bw(k+1,m+2-j))*exp(-r*dt);
        end
```

```
    end

    V_BinomialLattice = V_lattice_bw(1,1);
    sim_vec(2,count) = V_BinomialLattice;

    count = count + 1;
end

if PLOTTING_BL == 1
    hFig = figure;
    set(hFig, 'Position', [100 700 600 500])
    plot(sim_vec(1,:),sim_vec(2,:),'b');
    hold on;
    plot(sim_vec(1,:),V_exact*ones(nBL/data_interval,1), 'r');

    xlim([sim_vec(1) sim_vec(1, end)]);
    grid on;

    hTitle = title('Binomial lattice simulation');
    hXLabel = xlabel('Number of grid points');
    hYLabel = ylabel('Investment option value');

    set(gca,'FontName','Helvetica');
    set([hTitle, hXLabel, hYLabel],'FontName','Helvetica');
    set([hXLabel, hYLabel],'FontSize',12);
    set( hTitle,'FontSize',14,'FontWeight','bold');
end

display(V_BinomialLattice);
rel_error_bl = abs(V_BinomialLattice - V_exact)/V_exact;
display(rel_error_bl);

%% FINITE DIFFERENCE METHOD WITH EXPLICIT AND IMPLICIT SCHEME
%  Grid_ik = Grid(x,t)
%
%  (x=0,t=0)  -----------------  (x=0,t=T)
%            |                 |
%            |                 |  x
%            |                 |  |
%            |                 |  v
%            |                 |
%  (x=X,t=0)  -----------------  (x=X,t=T)
%                  t -->

Nt = 10000;
Nx = 90;
xMax = 900;
xMin = 0;
```

```
dt = T/Nt;
dx = (xMax-xMin)/Nx;

% Create a regular mesh
V_grid = nan(Nx+1,Nt+1);
xAxis = xMin:dx:xMax;
tAxis = 0:dt:T;

% Set the BCs
V_grid(:,Nt+1) = max(xAxis-I,0);                   %F(x,T)=max(x-S,0)
V_grid(Nx+1,:) = xAxis(Nx+1)-I*exp(-r*(T-tAxis)); %F(X,t)=x-Se^(-r(T-t))
V_grid(1,:) = 0;                                   %F(0,t)=0

% Choose which method to use (0=no, 1=yes)
METHOD_EXPLICIT = 1;
METHOD_IMPLICIT = 1;

if METHOD_EXPLICIT == 1
    V_grid_exp = V_grid;
    % Calculate tri-diagonal coefficient matrix
    abc_exp = zeros(Nx-1,3);

    for i=1:Nx-1
        abc_exp(i,1) = 0.5*(sigma^2*i^2-r*i)*dt;
        abc_exp(i,2) = 1-(sigma^2*i^2+r)*dt;
        abc_exp(i,3) = 0.5*(sigma^2*i^2+r*i)*dt;
    end

    A_exp = gallery('tridiag',abc_exp(2:end,1), ...
                abc_exp(:,2),abc_exp(1:end-1,3));

    % Calculate values at inner nodes starting from terminal time
    % and going backwards into t=0
    for k=Nt+1:-1:2
        V_grid_exp(2:Nx,k-1) = A_exp*V_grid_exp(2:Nx,k);
        V_grid_exp(2,k-1) = V_grid_exp(2,k-1) ...
                            + abc_exp(1,1)*V_grid_exp(1,k);
        V_grid_exp(Nx,k-1) = V_grid_exp(Nx,k-1) ...
                            + abc_exp(end,3)*V_grid_exp(Nx+1,k);
    end

    % Display figures
    hFig = figure;
    set(hFig, 'Position', [1000 400 1000 850])
    subplot(2,2,1);
    plot(xAxis,V_grid_exp(:,1),'b-',xAxis,V_grid_exp(:,end),'r-');

    hTitle = title('FDM (Explicit)');
    hXLabel = xlabel('x');
```

```
    hYLabel = ylabel('V(x,t)');
    legend('t=0','t=T','Location','northwest')

    set(gca,'FontName','Helvetica');
    set([hTitle, hXLabel, hYLabel],'FontName','Helvetica');
    set([hXLabel, hYLabel],'FontSize',10);
    set( hTitle,'FontSize',12,'FontWeight','bold');

    subplot(2,2,3);
    mesh(xAxis,tAxis,V_grid_exp');
    xlim([0 xAxis(end)]);

    hTitle = title('FDM (Explicit)');
    hXLabel = xlabel('x');
    hYLabel = ylabel('t');
    hZLabel = zlabel('V(x,t)');

    set(gca,'FontName','Helvetica');
    set([hTitle, hXLabel, hYLabel, hZLabel],'FontName','Helvetica');
    set([hXLabel, hYLabel, hZLabel],'FontSize',10);
    set( hTitle,'FontSize',12,'FontWeight','bold');

    % Interpolate the values
    V_FDM_Exp = interp1(xAxis,V_grid_exp(:,1),x0);
    error_exp = V_FDM_Exp - V_exact;
    relerror_exp = (V_FDM_Exp - V_exact)/V_exact;
    display(V_FDM_Exp);
    display(error_exp);
    display(relerror_exp);
end

if METHOD_IMPLICIT == 1
    V_grid_imp = V_grid;
    % Calculate tri-diagonal coefficient matrix
    abc_imp = zeros(Nx-1,3);

    for i=1:Nx-1
        abc_imp(i,1) = 0.5*(r*i-sigma^2*i^2)*dt;
        abc_imp(i,2) = 1+(sigma^2*i^2+r)*dt;
        abc_imp(i,3) = 0.5*(-r*i-sigma^2*i^2)*dt;
    end

    A_imp = gallery('tridiag',abc_imp(2:end,1), ...
                abc_imp(:,2),abc_imp(1:end-1,3));

    % Calculate values at inner nodes starting from terminal time
    % and going backwards into t=0
    leftover_const = zeros(size(abc_imp(:,2)));
```

```
    for k=Nt:-1:1
        leftover_const(1) = abc_imp(1,1)*V_grid_imp(1,k);
        leftover_const(end) = abc_imp(end,3)*V_grid_imp(Nx+1,k);
        V_grid_imp(2:Nx,k) = A_imp\(V_grid_imp(2:Nx,k+1) ...
                                    - leftover_const);
    end

    % Display figures
    %hFig = figure;
    %set(hFig, 'Position', [1400 400 1000 850])
    subplot(2,2,2);
    plot(xAxis,V_grid_imp(:,1),'b-',xAxis,V_grid_imp(:,end),'r-');

    hTitle = title('FDM (Implicit)');
    hXLabel = xlabel('x');
    hYLabel = ylabel('V(x,t)');
    legend('t=0','t=T','Location','northwest')

    set(gca,'FontName','Helvetica');
    set([hTitle, hXLabel, hYLabel],'FontName','Helvetica');
    set([hXLabel, hYLabel],'FontSize',10);
    set( hTitle,'FontSize',12,'FontWeight','bold');

    subplot(2,2,4);
    mesh(xAxis,tAxis,V_grid_imp');
    xlim([0 xAxis(end)]);

    hTitle = title('FDM (Implicit)');
    hXLabel = xlabel('x');
    hYLabel = ylabel('t');
    hZLabel = zlabel('V(x,t)');

    set(gca,'FontName','Helvetica');
    set([hTitle, hXLabel, hYLabel, hZLabel],'FontName','Helvetica');
    set([hXLabel, hYLabel, hZLabel],'FontSize',10);
    set( hTitle,'FontSize',12,'FontWeight','bold');

    % Interpolate the values
    V_FDM_Imp = interp1(xAxis,V_grid_imp(:,1),x0);
    error_imp = V_FDM_Imp - V_exact;
    relerror_imp = (V_FDM_Exp - V_exact)/V_exact;
    display(V_FDM_Imp);
    display(error_imp);
    display(relerror_imp);
end

% Copyright 2015, Arto Sorsimo, All rights reserved.
```

```
% Arto Sorsimo | arto.sorsimo@aalto.fi
% Aalto University School of Business
% Helsinki, 16.3.2015
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % COMPARISON OF NUMERICAL METHODS IN ROA  %
% % Case study #2                           %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This file uses two different numerical
% solution methods (binomial lattice and FDM)
% for solving the equation
%
% { L(x,t) := 1/2*sigma^2*x^2*V_xx + rxV_x - rV + V_t <= 0,
% { V(x,t) >= max{S-x,0},
% { L(x,t)(V(x,t) - max{S-x,0}) = 0,
% { V(x,T) = max{S-x,0},
% { lim x->inf V(x,t) = 0,
%
% where
%
% V: value of the abandonment option,
% x: stochastic variable (e.g. cash flows),
% S: salvage value of the project,
% T: terminal time (time to abandon),
% r: risk-free interest rate,
% sigma: standard deviation of stochastic variable.
%
% The problem is similar to American put option.
%
% This code is not fully optimized to be efficient.
% Use with care.

clear all;
close all;

%% Global parameters
x0 = 300;
S = 200;
r = 0.03;
sigma = 0.30;
T = 5;

%% Binomial lattice
tic;
nBL = 500;
data_interval = 1;

% Choose whether to plot or not (yes=1, no=0)
```

```
PLOTTING_BL = 1;

if PLOTTING_BL == 0
    data_interval = nBL;
end

sim_vec = zeros(3,nBL/data_interval);
sim_vec(1,:) = (1:data_interval:nBL);

count = 1;
for m=data_interval:data_interval:nBL

    dt = T/m;
    u = exp(sigma*sqrt(dt));
    d = exp(-sigma*sqrt(dt));
    p = (exp(r*dt)-d)/(u-d);

    % Initialize the lattice matrix and
    % fill the first row with up values
    V_lat_fw = zeros(m);
    for k=1:m+1
        V_lat_fw(1,k) = x0*u^(k-1);
    end

    % Calculate down values column-by-column
    for k=1:m
        for j=1:m
            V_lat_fw(1+k,j+1) = V_lat_fw(k,j)*d;
        end
    end

    % Apply call/put BCs at the terminal time
    V_lat_bw = zeros(size(V_lat_fw));
    V_lat_bw(:,m+1) = max(S-V_lat_fw(:,m+1),0);

   % Calculate value for the open exercise option
   for j=1:m
        for i=1:m
            hold_value = (p*V_lat_bw(i,m+2-j) ...
                +(1-p)*V_lat_bw(i+1,m+2-j))*exp(-r*dt);
            V_lat_bw(i,m+1-j) = max(hold_value, ...
                                    S-V_lat_fw(i,m+1-j));
        end
    end

    V_BinomialLattice = V_lat_bw(1,1);
    sim_vec(2,count) = V_BinomialLattice;

    count = count + 1;
```

```
end

timer_bl = toc;
display(timer_bl);

if PLOTTING_BL == 1
    plot(sim_vec(1,:),sim_vec(2,:),'b');
    xlim([sim_vec(1) sim_vec(1, end)]);
    grid on;

    hTitle = title('Binomial lattice simulation');
    hXLabel = xlabel('Number of grid points');
    hYLabel = ylabel('Abandonment option value');

    set(gca,'FontName','Helvetica');
    set([hTitle, hXLabel, hYLabel],'FontName','Helvetica');
    set([hXLabel, hYLabel],'FontSize',12);
    set( hTitle,'FontSize',14,'FontWeight','bold');
end

display(V_BinomialLattice);

%% Finite difference method with implicit method
%  Grid_ik = Grid(x,t)
%
%  (x=0,t=0)  -----------------  (x=0,t=T)
%            |                 |
%            |                 |  x
%            |                 |  |
%            |                 |  v
%            |                 |
%  (x=X,t=0)  -----------------  (x=X,t=T)
%                   t -->
tic;
Nt = 300;
Nx = 200;
xMax = 900;
xMin = 0;

dt = T/Nt;
dx = (xMax-xMin)/Nx;

% Create a regular mesh
V_grid = nan(Nx+1,Nt+1);
xAxis = xMin:dx:xMax;
tAxis = 0:dt:T;

% Set the BCs
V_grid(:,Nt+1) = max(S-xAxis,0);   %F(x,T)=max(S-x,0)
```

```
V_grid(1,:) = S*exp(-r*(T-tAxis)); %F(0,t)=Se^(-r(T-t))
V_grid(Nx+1,:) = 0;                %F(X,t)=0

% Calculate tri-diagonal coefficient matrix
abc = zeros(Nx-1,3);

for i=1:Nx-1
    abc(i,1) = 0.5*(r*i-sigma^2*i^2)*dt;
    abc(i,2) = 1+(sigma^2*i^2+r)*dt;
    abc(i,3) = 0.5*(-r*i-sigma^2*i^2)*dt;
end

A = gallery('tridiag',abc(2:end,1), ...
            abc(:,2),abc(1:end-1,3));

% Calculate values at inner nodes starting from
% terminal time and going backwards into t=0
leftover_const = zeros(size(abc(:,2)));

for k=Nt:-1:1
    leftover_const(1) = abc(1,1)*V_grid(1,k);
    leftover_const(end) = abc(end,3)*V_grid(Nx+1,k);

    holding = A\(V_grid(2:Nx,k+1) - leftover_const);
    V_grid(2:Nx,k) = max(holding,S-xAxis(2:end-1)');
    V_grid(1,k) = max(S,S*exp(-r*(T-tAxis(k))));
end

V_grid(2:Nx,1) = max(holding,S-xAxis(2:end-1)');

% Display figures
hFig = figure;
set(hFig, 'Position', [400 400 1000 350])
subplot(1,2,2);
plot(xAxis,V_grid(:,1),'b-',xAxis,V_grid(:,end),'r-');

hTitle = title('FDM (Implicit)');
hXLabel = xlabel('x');
hYLabel = ylabel('V(x,t)');
legend('t=0','t=T','Location','northwest')

set(gca,'FontName','Helvetica');
set([hTitle, hXLabel, hYLabel],'FontName','Helvetica');
set([hXLabel, hYLabel],'FontSize',12);
set( hTitle,'FontSize',14,'FontWeight','bold');

subplot(1,2,1);
mesh(xAxis,tAxis,V_grid');
xlim([0 xAxis(end)]);
```

```
hTitle = title('FDM (Implicit)');
hXLabel = xlabel('x');
hYLabel = ylabel('t');
hZLabel = zlabel('V(x,t)');

set(gca,'FontName','Helvetica');
set([hTitle, hXLabel, hYLabel, hZLabel],'FontName','Helvetica');
set([hXLabel, hYLabel, hZLabel],'FontSize',12);
set( hTitle,'FontSize',14,'FontWeight','bold');

% Interpolate the values
V_FDM_Imp = interp1(xAxis,V_grid(:,1),x0);
display(V_FDM_Imp);
timer_fdm_imp = toc;
display(timer_fdm_imp);

% Find the optimal abandonment boundary
x_f_index = zeros(1,length(tAxis));
epsilon = 0.001;  % relaxation parameter for the search

% Find the indexes in x-vector for boundary at all time periods
for j = 1:length(tAxis)
    x_f_index(1,j) = find(abs(V_grid(:,j)- ...
                        (S-xAxis)') < epsilon,1,'last');
end

% Find time indexes when the value changes
time_indexes = find(diff(xAxis(x_f_index(1,:))) > epsilon);
time_indexes = [1 time_indexes Nt];

% Calculate the mean from the jump
meanvalue = (xAxis(x_f_index(1,time_indexes+1))+ ...
                xAxis(x_f_index(1,time_indexes)))/2;

% Plot the curve
figure;
plot(tAxis(time_indexes),meanvalue);

hTitle = title('Optimal exercise boundary ');
hXLabel = xlabel('t');
hYLabel = ylabel('x');

set(gca,'FontName','Helvetica');
set([hTitle, hXLabel, hYLabel],'FontName','Helvetica');
set([hXLabel, hYLabel],'FontSize',12);
set( hTitle,'FontSize',14,'FontWeight','bold');

annotation('textarrow',[0.35,0.45],[0.32,0.265], ...
```

```
            'String','Exercise boundary', ...
            'FontSize', 12 ...
             );

annotation('textbox', [0.4,0.5,0.1,0.1], ...
            'String', 'Holding region', ...
            'FontSize', 16, ...
            'Color', [1 0 0], ...
            'LineStyle','none',...
            'Margin', 10 ...
             );

annotation('textbox', [0.65,0.15,0.1,0.1], ...
            'String', 'Exercise region', ...
            'FontSize', 16, ...
            'LineStyle','none',...
            'Color', [0 0.8 0] ...
             );

% Copyright 2015, Arto Sorsimo, All rights reserved.
```