Erasmus Mundus Consortium

MathMods

Joint Degree of Master of Science in
Mathematical Modelling in Engineering: Theory, Numerics, Applications

In the framework of the
Consortium Agreement and Award of a Joint/Multiple Degree 2013–2019

Master's Thesis

The Singular Points method for pricing exotic path–dependent options

Supervisor:
PROF. FABIO ANTONELLI

Candidate:
SUDIP SINHA
Matricola: 228435

2015-10-07

Contents

Chapter .

Preface

A path-dependent option is a type of exotic option in which the payoff depends not only on the price of the underlying asset at maturity, but also on the history of the underlying's price till that point. Typical examples of popular exotic options are Asian options, lookback options, barrier options, and digital options.

In the Black-Scholes market model, it is not possible to find closed-form analytical formulae for the payoffs of most exotic options. This inspires the requirement of fast numerical algorithms to determine the fair value of such options. Such algorithms may be clustered into categories. Some algorithms utilise the convergence of prices calculated using a discrete model to those of a continuous model, as a limit when the time step is reduced. Other approaches include use of numerical methods to solve partial differential equation, or simulation using Monte Carlo methods.

Under the Cox-Ross-Rubinstein model, Gaudenzi[1], Zanette[2] and Lepellere[3] introduced the singular points method. It is a numerical method to price Asian and lookback options. A modification enables pricing of cliquet options. In the method, in each node of the binomial tree of the underlying risky asset, the price is represented as a continuous function of the path-dependent parameter. An advantage of this method over pre-existing methods is its low order of computational and space complexity. It is convergent, and allows us to set *a priori* upper and lower bounds on the error.

In the master's thesis, we present an exposition on the *singular points* method and how it may be used to price exotic options. We also explore the extensibility of the method to similar types of options, like Asian options with geometric mean as opposed to arithmetic mean, and whether it may be used for variable local volatilities and interest rates.

Prerequisites   The reader is expected to be familiar about basic Probability Theory and Stochastic Processes. In particular, (s)he should be comfortable with probability spaces and measures, filtrations, random variables, stochastic processes, martingales, Brownian

---

[1] http://people.uniud.it/page/marcellino.gaudenzi
[2] http://people.uniud.it/page/antonino.zanette
[3] https://www.researchgate.net/profile/Maria_Lepellere

motion, and elementary stochastic calculus. There is no strict requirement of knowledge of financial concepts, as we introduce the required terminologies in the introductory chapters.

Note to the reader   The theory part of the thesis borrows heavily from the book titled Introduction to Stochastic Calculus Applied to Finance by Damien Lamberton and Bernard Lapeyre [**Lamberton1996**], as well as from the lecture notes of Mathematical Finance authored by Prof. Fabio Antonelli. This course was offered in the Fall semester of 2014-15 in Università degli Studi dell'Aquila, Italy.

The chapters on the singular point method has been motivated by the series of papers published by Gaudenzi, Zanette and Lepellere on the same topic.

Structure of the thesis   In Chapter **??**, we briefly discuss financial assets and financial markets. In Chapter **??**, we introduce the Cox-Ross-Rubinstein model and the Black-Scholes model. In chapters **??** and **??**, we see how the singular points method may be used to price Asian and cliquet options, respectively.

Notations

$[n]$: $\{0, 1, 2, \ldots, n\}$

$\mathbb{N}$: $\{1, 2, 3, \ldots\}$

$\mathbb{N}_0$: $\mathbb{N} \setminus \{0\}$

$\mu\mathcal{F}$: The class of all $\mathcal{F}$-measurable functions.

$\lfloor c \rfloor$: The floor function. $\lfloor x \rfloor$ gives the largest integer smaller or equal to x.

$\lceil e \rceil$: The ceiling function. $\lceil x \rceil$ gives the smallest integer larger or equal to x.

$\sim$: Represents the discounted value when used on top of a quantity. For example, $\tilde{S}_t = e^{-rt} S_t$

Units of prices   Throughout the text, we have not specified any monetary unit for the prices of assets. This is because the theory and computations hold no matter what unit is chosen. The only care that needs to be taken is to have all quantities in the same unit; in case not, they should be converted to a uniform unit system.

Chapter 1

Prologue

Mathematics and finance have been intricately linked since the inception of human civilisations. In the span of the last two millennia, many mathematicians – from Thales of Greece to Fibonacci, Pascal, Fermat and Bernoulli – have tried to apply their knowledge to the domain of finance[1]. However, Louis Bachelier's thesis "Théorie de la Spéculation" (Theory of Speculation) in 1900 arguably marked the birth of modern mathematical finance. Bachelier is credited with being the first person to derive the mathematics of Brownian motion and to apply its trajectories for modelling stock price dynamics and calculating option prices.

One of the most exciting innovations in the modern history of mathematical finance is rooted in the discovery of the Brownian motion by Scottish botanist Robert Brown. In 1827, he observed rapid oscillatory motion of microscopic particles in a fluid resulting from their collision with atoms or molecules in the fluid. However, Bachelier was the first to define Brownian motion mathematically, and used the one dimensional version $t \mapsto B_t, t \geqslant 0$ to model stock price dynamics. Norbert Wiener laid down the rigorous mathematical foundation of Browian motion, hence the parallel name of Wiener process is also used for Brownian motion. He proved the existence of Brownian motion, and constructed the Wiener measure which describes its probability distribution.

In 1933, in the seminal book "Foundations of the Theory of Probability", Kolmogorov laid the foundation of modern Probability theory from fundamental axioms. His work relied heavily on Lebesgue's Measure Theory, and united the previously disparate areas of probability – discrete and continuous – into a unified theory that could be applied to both. It is in this book that he introduced the ideas of conditional expectation and equivalent measures.

Another breakthrough in Probability theory came in 1941, when in his attempts to model Markov processes, Kiyosi Itô (in his famous 1942 paper "On stochastic processes") constructed stochastic differential equations of the form $dX_t = \mu(X_t)\,dt + \sigma(X_t)\,dB_t$, where B_t is a standard Brownian motion. He introduced the theory of integration over a Brownian motion, and in 1951 gave a formula for constructing new stochastic differntial equations from existing ones (Itô's lemma), giving birth to Itô calculus. Itô's lemma is one of the most widely used mathematical formula by financial engineers today.

---

[1]Significant portions in this introductory section have been taken verbatim from the paper [**Akyildirim2014**]

At this crucial juncture, when the probability theory was ripe enough, in 1973, Fischer Black and Myron Scholes published the paper "The Pricing of Options and Corporate Liabilities", and Robert Merton published the paper "On the pricing of corporate debt: the risk structure of interest rates". These papers together introduced a new methodology for the valuation of financial instruments and in particular developed the Black–Scholes model for pricing European call and put options. At the same time, another breakthrough on the industry side was the foundation of the Chicago Board Options Exchange to become the first marketplace for trading listed options. Even beyond the imagination of the celebrated authors above, the market was so quick to adapt these models. By 1975, almost all traders were valuing and hedging option portfolios by using the Black–Scholes model built in their hand calculators.

But there were problems with the Black and Scholes model. A primary disadvantage was that the esoteric theory was not understandable by most practitioners in the field of finance. Moreover, it could not be used to price all kinds of options. In 1979, Cox, Ross and Rubinstein introduced a discrete model, which is much more approachable, and yet has the flexibility to price almost all kinds of derivatives. Yet, the exponential nature of the complexity forbids scaling up the number of levels of the tree to any significant value, reducing its utility.

In this chapter, we briefly discuss some fundamental ideas in finance. We shall follow this up with a discussion on the two models mentioned.

1.1   Financial instruments

A *financial instrument* or a *financial asset* is an intangible asset whose value is derived from a contractual claim, such as bank deposits, bonds, stocks and derivatives. Financial assets are usually more *liquid* than other tangible assets, such as commodities or real estate, and may be traded on financial markets. Every financial asset is characterised by its return. When the return is deterministic, we call it a *risk-free* or *riskless* asset. When the return is contingent on the market and external conditions, it is called *risky*. It must be kept in mind that no instrument is fundamentally risk-free, it has only negligible risk compared to its risky counterparts.

1.1.1   Return on an asset

Return on a riskless asset – Compounding   Compounding is the first idea that we must be familiar with. Essentially, a riskless asset will increase in monetary value in a deterministic manner if we keep it in the market. The increase depends on the compounding frequency and the duration of investment. The term compounding is used because the interest earned in each period also contributes to the principal in the successive periods.

Let the compounding frequency is n times per year, the total time is t, and the annual rate of interest is r. Then

(1.1) $$B_t = B_0 \left(1 + \frac{r}{n}\right)^{\lfloor nt \rfloor},$$

where B_0 is the starting value of the asset, and B_t is the value of the asset at time t, and $\lfloor\ \rfloor$ represents the floor function.

If the compounding is continuous, we let $n \nearrow \infty$ to obtain

(1.2) $$B_t = B_0 e^{rt}.$$

Return on a risky asset   At any point of time in the future, the value of a risky asset is not known with certainty, so it is a random variable taking values in $[0, \infty)$. Since the values change with time, we denote the *spot price* of a risky asset by the stochastic process[2] $(S_t)_t$, where $t \in [0, T]$ denotes the time and T is the maturity. Since the future value of the asset is adventitious, we use the following quantities to measure the return of the risky asset in the given time interval.

DEFINITION 1.1 (absolute and relative returns). The absolute return on an asset for the time interval $[0, t]$, $t \in [0, T]$ is given by

$$\tilde{R}_t = S_t - S_0$$

The relative return on the asset is given by

$$R_t = \frac{S_t - S_0}{S_0}$$

It must be noted that relative return is usually a better indicator of the return on an asset as compared to its absolute equivalent. This may be illustrated better by the following example.

EXAMPLE 1.1 (relative vs absolute return). Let A and B be two assets. Let their values at $t = 0$ be $S_0^A = 5$ and $S_0^B = 10$, respectively. At time $t = 1$ one observes that the prices have become $S_1^A = 10$ and $S_1^B = 15$.

The absolute return on both assets are equal to 5 here. On the other hand, the relative returns are:

$$R_1^A = \frac{S_1^A - S_0^A}{S_0^A} = \frac{5}{5} = 1$$

$$R_1^B = \frac{S_1^B - S_0^B}{S_0^B} = \frac{5}{10} = \frac{1}{2}$$

Thus, the asset A is more interesting to an investor, as it gives more return on investment. If an investor were to invest an amount of 10 at $t = 0$, it is more profitable for him to buy two quantities of asset A than one quantity of asset B.

---

[2]https://en.wikipedia.org/wiki/Stochastic_process

1.1.2 Common asset types

Bonds   A *bond* is an instrument of indebtedness of the bond issuer to the holders. It is a *debt security*, under which the issuer owes the holders a debt and, depending on the terms of the bond, is obliged to pay them interest (the coupon) and/or to repay the principal at a later date, termed the maturity date. Bonds can also be thought of as a *loan* given to the issuer of the bond by the holder. A bond primarily has two kinds of risks, *credit default risk* and *interest rate risk*. A bond issued by a reliable institution like the United States government is a good illustration of a risk-free asset. This is because the probability of such an organisation defaulting is close to zero, or in other words, the bond has *negligible credit default risk*. Such bonds are only subject to fluctuations of the current interest rate, called *interest rate risk*. The interest rate risk may also be nullified if the bond is held till maturity. If we assume that the interest rate is deterministic (the fluctuations are not random), the value of the bond is computable at any given future date, making it riskless. Such an assumption is quite reasonable in short periods of time and for institutions with a low default risk rating.

Stocks   A *stock* of a corporation is an ownership certificate, and constitutes the equity stake of its owners. It represents the residual assets of the company that would be due to stockholders after discharge of all senior claims such as secured and unsecured debt. A *share* of a stock is a unit of ownership of the organisation. Stocks are inherently risky, since the value of the organisation may change from time to time due to various internal and external factors. As noted earlier, since stocks are risky, the value of a stock in time is represented by a stochastic process $(S_t)_t$.

Derivatives   A *derivative* is a contract between two parties that specify conditions (starting and termination dates, resulting values and definitions of the underlying variables, the parties' contractual obligations, and the notional amount) under which transactions are to be made between the parties. The most common underlying assets include commodities, stocks, bonds, interest rates and currencies, but they can also be other derivatives, which adds another layer of complexity to proper valuation. Essentially, the value of a derivative is a function of the value of the *underlying* asset(s). Derivatives are traded in their own right, and a fair price must be found for a derivative at each time of its existence. This problem is known as the *pricing problem*. One of the primary motivations for creation of derivatives was to hedge one's position from fluctuations in the market. A *hedge* is an investment strategy intended to offset potential losses or gains that may be incurred by a companion investment. Finding a hedging strategy is called the *hedging problem*. These are the two problems that must be looked at when defining a market model. In this thesis, our main focus shall be the pricing problem of a particular class of derivatives, called *exotic options*.

1.1.3 Classification of derivatives

Derivatives may be classified on the basis of various factors. One important factor is whether the risk is shared, or taken up by only one party. Another factor is the nature of the function (of the underlying) that the derivative depends on. This function may either be dependent only on the final value of the underlying (*path-independent*), or on the path that it took to reach this final value (*path-dependent*). The function may be discrete (*digital* or *binary*), or continuous In this section, we briefly look at some of the more important types of derivatives. [3]

Futures and forwards

DEFINITION 1.2 (Futures and forwards). Futures and forwards are contracts between two parties, the seller and the buyer, to exchange a certain asset at a predetermined future time at a agreed upon price. Futures are *exchange-traded derivatives* (ETDs), whereas forwards are traded *over-the-counter* (OTC).

Such derivatives obligate the contractual parties to the terms over the life of the contract. Futures are in some sense 'safer' compared to forwards, since the involved parties must go through standard protocols of the exchange. The contract contains the following details.

T: The maturity, or the duration of the contract

F_0: The delivery price, or the price prefixed (at the initial time) at which trades must take place at maturity

r: The rate of interest

underlying: The asset(s) of trade at maturity

S_0: The initial value of the underlying asset(s)

There are, of course, other possibilities, for instance a variable interest rate, dividends yielded by the underlying, but these may be viewed as generalisations of this simple case.

Let us assume that the compounding is continuous. We may show that under the condition of a *viable market*[4], the fair delivery price of a future with underlying prices $(S_t)_{t\in[0,T]}$ at any time $t \in [0, T]$ is given by the following equation.

(1.3)
$$F_t = S_t e^{r(T-t)}$$

Options

DEFINITION 1.3 (option). An *option* or a *contingent claim* is a derivative which provides the buyer *the right, but not the obligation* to enter the contract under the specified terms.

---

[3]A more interested reader should consult the following extensive Wikipedia articles.

https://en.wikipedia.org/wiki/Option_(finance)#Types
https://en.wikipedia.org/wiki/Option_style

[4]see Section ?? for definitions of the term

Thus, the owner of the option may choose whether to exercise his right or not. Thus, on the one hand, the owner of the option bears no risk, since all the choice is his. On the other hand, the seller of the option is *obligated* to honour the terms of the contract – whether it benefits him or not – essentially making him bear all the risks. This asymmetry is primarily what sets options apart from the futures and forwards discussed earlier. The contract contains the following details.

T: The maturity, or the duration of the contract

K: The strike price, or the prefixed price at which trades may take place at maturity

r: The rate of interest

underlying: The asset(s) which may be traded at maturity

S_0: The initial value of the underlying asset at the initial time

right: The exact right that the owner of the options has (see below)

exercise time: European or American (see below)

Options may be categorised by the right of the owner and the exercise time.

According to the right of the owner   Options may be of two main types.

call: The owner has the right to buy. In this case, the price of the option at maturity is given by $c_T = (S_T - K)_+$, where $(x)_+ := \max\{0, x\}$.

put: The owner has the right to sell. In this case, the price of the option at maturity is given by $p_T = (K - S_T)_+$.

Of course, other complicated ownership rights may be constructed, but we shall not go into the details of those.

According to the time at which the option may be exercised   Again, option may be of two main types.

European: The owner may exercise the option only at maturity

American: The owner may exercise the option at any time up to the maturity

Again, more complicated options exist, which allow exercising rights only at certain time points, but we exclude them from out discussion.

Since American options allow for more flexibility for the owner, and thus more risk for the seller, they are more expensive as compared to their European counterparts. Let c_t, p_t denote the prices of an European call and put, and C_t, P_t denote the prices of an American call and put, respectively. Then, we must have $C_t \geqslant c_t$ and $P_t \geqslant p_t$.

Call-put parity   Call and put prices are connected to each other. We need the following proposition to explore the relationship.

PROPOSITION 1.1 (Equality of portfolios). *In a* viable *and* frictionless market[5], *if the values of two portfolios coincide at a time* T, *they have to coincide at* 0 *(or any other intermediate time* t*).*

PROOF. Let us denote by $\mathcal{P}_1$ and $\mathcal{P}_2$ the two portfolios and by $v_t(\mathcal{P})$ the value of a portfolio $\mathcal{P}$ at time t. By assumption $v_T(\mathcal{P}_1) = v_T(\mathcal{P}_2)$, so we assume by contradiction that $v_0(\mathcal{P}_1) > v_0(\mathcal{P}_2)$.

Under this hypothesis it is possible to construct the following arbitrage strategy. At time 0, one can borrow the portfolio $\mathcal{P}_1$ and sell it right away to buy portfolio $\mathcal{P}_2$. One can pocket the difference $v_0(\mathcal{P}_1) - v_0(\mathcal{P}_2) > 0$. At $t = T$ the values of the two portfolios coincide, so selling $\mathcal{P}_2$ one gets the exact money to buy $\mathcal{P}_1$ to be returned to the original lender. An profit is achieved, without investing any money, implying an arbitrage and violating the viable market hypothesis.

Similarly, we can show that $v_0(\mathcal{P}_1) < v_0(\mathcal{P}_2)$ would also enable an arbitrage opportunity. Generalising the argument to all intermediate times t $\in$ [0, T], we get the result.   □

We now look at the relationship. $S_T - K = (S_T - K)_+ + (S_T - K)_- = (S_T - K)_+ - (K - S_T)_+ = c_T - p_T$. For any general time t, using Proposition **??**, it holds that $c_t - p_t = S_t - Ke^{-r(T-t)}$. This is known as the *call-put parity*.

Exotic options   European options are path-independent and the simplest type of options available. Hence, they are popularly known as *vanilla options*. The American options are path-dependent. Typically, other options which are more complex in nature are collectively called *exotic options*. These are usually path-dependent, and may be either European, American or have more complex exercise times. A few such options are described in brief.

    Asian: The payoff depends on the average of the underlying's prices

    lookback: The payoff depends on one of the extrema of the underlying's prices

    cliquet or ratchet: A series of globally or locally, capped or floored, at-the-money options, but where the total premium is determined in advance.

    barrier: The price of the underlying reaching the pre-set barrier level either springs the option into existence (*knock-in*) or extinguishes an already existing option (*knock-out*).

1.2   Financial Markets

The idea of financial markets is intricately linked to that of financial transactions. Analogous to the ordinary markets, a financial market is a human construct to allow transaction between investors. The assets in the financial market are typically financial instruments such as bonds, stocks and derivatives discussed in the previous section. In this section we will primarily concern ourselves with the nature of financial markets and the assumptions we

---

[5]see Section **??** for definitions of the terms

make while modelling them. Some of the jargon used in the previous section will become clear after this section.

Pricing of financial assets is one of the most pressing aims of the subject of Financial Mathematics. In order to do so, we need to understand and characterise the fundamental mechanisms of the market that shape the pricing of assets. In doing so, we must note which dynamics of the market are most fundamental and must be incorporated in every model, and which are more debatable may be excluded from simpler models.

Viable market   The term viability here refers to the fairness of a market. To interpret viability, we need to familiarise ourselves with the following definitions.

In what follows, we assume the following.

> There is a probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_t, P)$, endowed with the filtration $(\mathcal{F}_t)_t$.
>
> $\forall t \in [0, T], T \in [0, \infty)$, there is one riskless asset worth $S_t^0 = e^{rt}$ (Take $S_0^0 = 1$).
>
> $\forall t \in [0, T], T \in [0, \infty)$, there are d risky assets each worth S_t^i, where $i \in \{1, 2, \ldots, d\}$ is the index of the risky asset. These may be represented as a d–dimensional (vector) stochastic process $(S_t^1, S_t^2, \ldots, S_t^d)$.

Also, any quantity with a '$\sim$' on top denotes the discounted value of that quantity.

DEFINITION 1.4 (investment strategy). A $d + 1$ dimensional (vector) stochastic process $\Phi = (\boldsymbol{\phi_t})_t = (\phi_t^0, \phi_t^1, \ldots, \phi_t^d)_t$ is called an *investment strategy* or *trading strategy* if ϕ_t^i is $\mathcal{F}_t$–measurable $\forall i \in [d]$.

This means that there is a procedure to allocate resources within the portfolio at all times. We shall write strategy to mean investment strategy in the rest of the document.

The following definition gives the value of a strategy at a point in time.

DEFINITION 1.5 (value of a strategy). The value of a strategy Φ at a time t is given by $V_t(\Phi) = \boldsymbol{\phi_t} \cdot S_t$.

DEFINITION 1.6 (self–financing strategy). A portfolio is self–financing if its changes in value are only due to changes in prices of the assets. This can be represented as follows.

$$(1.4a) \qquad\qquad dV_t = \boldsymbol{\phi_t} \cdot dS_t \qquad \forall t \in [0, T]$$

$$(1.4b) \qquad \implies \qquad d\tilde{V}_t = \boldsymbol{\phi_t} \cdot d\tilde{S}_t \qquad \forall t \in [0, T]$$

This implies that we do not put in any fresh money in the strategy at any point of time, apart from what is generated due to the change is values of the underlying assets.

DEFINITION 1.7 (admissible strategy). A self–financing strategy Φ is said to be admissible if $V_t(\Phi) \geqslant 0 \ \forall t \in [0, T]$.

This implies that we do not run out of money at any point of time.

DEFINITION 1.8 (arbitrage strategy). An admissible strategy Φ is said to be an arbitrage strategy if $V_0(\Phi) = 0$ and $P(V_T(\Phi) > 0) > 0$.

A arbitrage strategy basically means that we generate value at time T without any initial investment. For the sake of fairness, we do not want a market in which there exist arbitrage opportunities. The next definition addresses this issue.

DEFINITION 1.9 (viable market, no free lunch). A market is called *viable*, or there is *no free lunch*, if there does not exist any arbitrage strategies.

We will see in Theorem ?? of Chapter ?? how the financial concept of viability translates mathematically to the existence of equivalent martingale measures.

Complete market

DEFINITION 1.10 (hedgeable, attainable). A derivative is called hedgeable or attainable if it can be represented as a linear combination of the basic assets at all times. Basic assets include stocks, bonds and commodities whose prices are *not* directly linked to other assets in the market.

DEFINITION 1.11 (complete market). A market is said to be complete if all derivatives are hedgeable.

A market is complete if the traded basic assets represent all the random factors that influence the course of prices. If we are in complete market model, this means that, whatever be the contract, one can always set a hedging strategy that equals the final value of the derivative. This hedging strategy employs, by definition, only the basic assets (riskless and risky) traded on the market. In a sense those derivatives are redundant, they do not introduce any additional risk factor; any randomness source is represented by the basic assets prices and it is tradeable.

This implies that the marginal probability distributions of the derivative price are uniquely determined by the marginal distributions of the prices of the basic assets. Namely, each derivative is replicable, so its price at all times may be written as the value of a portfolio employing basic assets only, that is to say it is a linear combination of the prices of the basic assets. On the other hand, in a viable market the discounted value of a portfolio is a martingale under the risk–neutral probability, so if a contingent claim is attainable, then its price, by the martingale property, belongs to the linear span of the basic assets' prices under this probability. Vice versa, if the market is incomplete, then there must be sources of randomness that cannot be totally represented as linear combinations of the prices of the basic tradeable assets, which means that the basic titles are not sufficient to construct all the necessary hedging strategies.

We will see in Theorem ?? of Chapter ?? how the financial concept of market completeness translates mathematically to the existence of a unique equivalent martingale measure.

Frictionless market   For any transaction (sale or purchase) in the market, one usually pays some *commission*. The commission is a very small fraction of the current value of the traded assets, and it seems reasonable to assume that it is not a factor that affects the dynamics of the prices in a direct fashion. Furthermore, the computational difficulty of including such

transactional costs in the market is quite high. Thus, we choose to ignore such costs in the simple market models that we shall deal with, and call the market as *frictionless*.

DEFINITION 1.12 (frictionless market). A market is called frictionless if there are no transaction costs.

Infinitely divisible assets   In a market, usually only discrete units of assets may be bought or sold. This would pose an additional constraint in the modelling of the market. But it is quite evident that this constraint does in no way affect prices of individual assets. Furthermore, markets are usually so varied that one might think to combine stocks and bonds to a value that is roughly equivalent to a fraction of a different asset. Thus, we ignore this constraint in our market models, and say that we may have *infinitely divisible assets* in our market.

Small investor hypothesis   An investor who has virtually unlimited funds might decide to buy massive quantities of an asset to make its price rise in order to sell it later at a higher price. We shall ignore such cases, and assume that all agents are trifling with respect to the market dimension, meaning that they cannot influence prices uniquely by means of their investing strategies, hence prices are determined only by the combined actions of all agents. This assumption is called the *small investor hypothesis* and it is totally realistic in bigger stock markets like for those in the United States, even though it is less so for much smaller markets.

Borrowing   Lastly, we assume that an investor may borrow assets, whether they are financial instruments such as money, bonds and stocks, or otherwise.

Chapter 2

Market models

Now that we are familiar the basics of financial assets and markets, we may delve into market models under which we may price options. Since continuous models are mathematically more complex than their discrete time counterparts (also known as *lattice models*), we shall discuss the latter first. We shall then show that under certain convergence conditions, the discrete model converges to the continuous one.

2.1 Discrete models – The binomial model

We start our discussion with one of the simplest model used for pricing of assets, the binomial model. This model was first introduced by Cox, Ross and Rubinstein [**Cox1979**] in the paper titled *Option pricing: A simplified approach* in 1979. Even though it is quite simplistic, it does contain all the necessary ingredients to construct a viable market model, and to solve the problems of pricing and hedging of derivatives.

We assume that the following are true.

> The market is frictionless
>
> All assets are infinitely divisible
>
> The small investor hypothesis holds
>
> The annual interest rate is constant and it is applied both for borrowing or lending, hence there are no spreads

In this model, we essentially have

> two time points, $t = 0$ (present) and $t = 1$ (future)
>
> two traded assets
>
> > ⋆ the riskless asset, usually a bond, which is compounded at a constant rate of interest $r > 0$

⋆ the risky asset, usually a stock, which may either go up with a factor u, or down with a factor d

The binomial model is so called because there are two times, two assets and two possible movements of the risky asset.

We denote as S_0 the value of the risky asset at time 0, and by S_1 its value at time 1. Firstly, in order to have no arbitrage opportunities, we must have $d < R < u$, where $R := 1 + r$. Secondly, for fairness, there must exist a probability distribution $p, (1 - p)$ – signifying the probabilities of the up and down movements, respectively – such that the expected value of the asset remains the same as that of the riskless asset given the same time. (See Figure ??.)
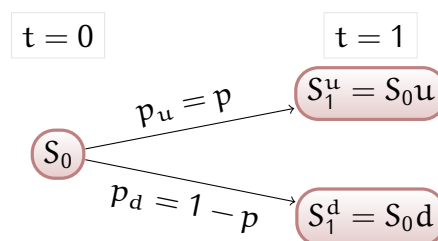

FIGURE 2.1. Binomial tree for the underlying

Thus, we have

$$S_1 = \begin{cases} uS_0 & \text{with probability } p \\ dS_0 & \text{with probability } (1 - p) \end{cases}$$

We may also write $S_1 = T_1 S_0$, where T_1 is a random variable taking values in $\{u, d\}$ with associated probability distribution $(p, 1 - p)$.

$$T_1 = \begin{cases} u & \text{with probability } p \\ d & \text{with probability } (1 - p) \end{cases}$$

We want $p \in [0, 1]$ such that $\mathbb{E}(S_1) = S_0 R$. This would make the system fair because the expected gain from either asset should be the same.

$$(2.1) \qquad\qquad\qquad \mathbb{E}(S_1) = S_0 R$$

$$\implies \ puS_0 + (1 - p)dS_0 = S_0 R$$

$$(2.2) \qquad\qquad\qquad \implies \qquad\qquad p = \frac{R - d}{u - d}$$

The probability thus obtained is called the *risk neutral probability*, because under this probability, on average, it is equivalent for the investor whether he invests in a risky or a riskless asset. Note that this probability is completely objective as it is determined completely by the parameters u, d and r. Since the probability distribution is uniquely determinable from the market parameters, the market is viable and complete (See Theorems ?? and ?? in the next section).

Now we impose the condition $p \in [0,1]$ to obtain

(2.3)
$$d < R < u$$

Volatility   A natural way to express the risk associated with an asset is by its variance. In the case of binomial model, at each time, the variance is associated with the gap between u and d, the up and down factors. It is reasonable to assume that when comparing to assets, an asset with the propensity of a higher up movement will also have a similar disposition towards a large down movement. If it were not so, the investors would invest in the asset with higher up movement potential, driving up the prices, and naturally recalibrating the current value of the asset, so that the assumption is true. Thus we may reduce the requirement of two variables u and d by expressing them as a function of only the variance of the return T_i. This may be achieved by making the model symmetrical by setting $u = d^{-1}$. In this case, the log return is symmetrical w.r.t. the origin since $\log(u) = -\log(d)$.

Moreover it is reasonable to think that this variance stays constant at each time unit, but directly proportional to time.

$$\mathbb{Var}(\log(T_i)) = \sigma^2 \Delta T$$

Here $\sigma > 0$ is called the *volatility* of the asset, and is used as a constant of proportionality.

With this choice it is natural to take

(2.4a)
$$u = e^{\sigma\sqrt{\Delta T}}$$

(2.4b)
$$d = e^{-\sigma\sqrt{\Delta T}}$$

Now we require only one parameter, σ, to get u and d, and subsequently to generate the whole tree. The value of this parameter must be estimated from the market.

Pricing a call   Let us use the above model to price a call. Recall that the pay-off of a call is given by $h(x) = (x - K)_+$, where K is the strike price, a fixed value specified in the contract. Thus, we know the values of the call at maturity. Note that for financial viability, we must have $K \in (S_1^d, S_1^u)$, implying $c_1^u = (S_1^u - K)_+ = (S_1^u - K)$ and $c_1^d = (S_1^d - K)_+ = 0$ (See Figure ??). To ensure fairness, we may again write the following

$$
\begin{aligned}
c_0 &= \mathbb{E}\left(\frac{c_1}{R}\right) \\
&= \frac{1}{R}\left(pc_1^u + (1-p)c_1^d\right) \\
&= p\frac{c_1^u}{R} \\
&= \frac{R-d}{u-d}\frac{uS_0 - K}{R}
\end{aligned}
$$

Thus we have been able to price the call uniquely at all times. This is an implication of the completeness of the market, whose randomness is totally characterized by the unique probability measure p.
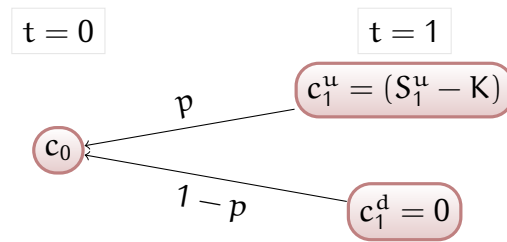
FIGURE 2.2. Binomial tree for the call

For the sake of completeness, we comment here that a call is also completely hedgeable in the binomial model.

2.2   Discrete models – The Cox Ross Rubinstein model

In this section we extend the binomial model introduced in Section ?? to a sequence of integer times $[n] := \{0, 1, \ldots, n\}$, $n \in \mathbb{N}$.

Let $(\Omega, \mathcal{F}, (\mathcal{F}_i)_i, P)$ be a finite probability space ($|\Omega| < \infty$), such that $P(\omega) > 0 \; \forall \omega \in \Omega$, endowed with a filtration $(\mathcal{F}_i)_{i \in [n]}$, such that $\mathcal{F}_0$ is trivial ($\mathcal{F}_0 = \{\emptyset, \Omega\}$).

We assume that the following are true.

The market is frictionless

All assets are infinitely divisible

The small investor hypothesis holds

The annual interest rate is constant and it is applied both for borrowing or lending, hence there are no spreads

The market is viable

The market is complete

Using the above, we may write

$$(2.5) \qquad S_n = s_0 \prod_{j=1}^{i} T_j \qquad \forall i \in [n].$$

Using exactly the same computations as in the binomial case, we again find the risk–neutral probability as $p = \frac{R-d}{u-d}$, such that $d < R < u$. Again, since the probability distribution is uniquely determinable from the market parameters, the market is viable and complete (See Theorems ?? and ?? in the next section).

2.2.1   Fundamental Theorems of Asset Pricing

DEFINITION 2.1 (absolute continuity of measures). Let $(\Omega, \mathcal{F})$ be a measurable space, with associated measures P and Q. The we say that P is absolutely continuous with respect to Q, or $P \ll Q$, if $\forall A \in \mathcal{F}$,

$$Q(A) = 0 \implies P(A) = 0.$$

DEFINITION 2.2 (Equivalence of measures). Let $(\Omega, \mathcal{F})$ be a measurable space, with associated measures P and Q. The we say that P is equivalent to Q, or $P \sim Q$, if $P \ll Q$ and $Q \ll P$.

We now need to invoke two cornerstone theorems, which will allow us to price the option at any time step i.

THEOREM 2.1 (First Fundamental Theorem of Asset Pricing). *The market model is viable if and only if there exists a probability measure* P^* *equivalent to the historic probability measure* P *under which the discounted prices of the basic risky assets are martingales.*

Mathematically, Viable market $(\ )$ $\exists P^* \sim P$ such that $\mathbb{E}^*(\tilde{S}_{i+1}|\mathcal{F}_n) = \tilde{S}_i$, where $\mathbb{E}^*$ is the expectation computed under P^*..

PROOF. See [**Lamberton1996**]   □

Theorem ?? guarantees the existence of an equivalent martingale measure for viable markets. This implies that under this probability measure, we can calculate the fair price of an option, although they may not be unique. To ensure uniqueness, we need the subsequent theorem.

THEOREM 2.2 (Second Fundamental Theorem of Asset Pricing). *The market model is complete if and only if there exists a **unique** probability measure* P^* *equivalent to the historic probability measure* P *under which the discounted prices of the basic risky assets are martingales.*

Mathematically, Complete market $(\ )$ $\exists! P^* \sim P$ such that $\mathbb{E}^*(\tilde{S}_{i+1}|\mathcal{F}_n) = \tilde{S}_i$, where $\mathbb{E}^*$ is the expectation computed under P^*.

PROOF. See [**Lamberton1996**]   □

Theorem ?? guarantees the uniqueness of an equivalent martingale measure for complete markets. This implies that we can uniquely and objectively compute the fair price of an option.

2.2.2 European options

Augmented by these theorems, we may now seek to price path–independent options in this model. Let $\mathbb{E}^*$ be the expectation computed under the risk–neutral martingale measure P^*. If the pay–off of an option at maturity is given by the function $h(x)$, then the price of the option v_n at any time step i is given by the following formula.

$$(2.6) \qquad v_i = \mathbb{E}^*(R^{-(n-i)}h(S_n)|\mathcal{F}_i)$$

In particular, the price a European call at any time i is as follows.

$$(2.7) \qquad c_i = \mathbb{E}^*(R^{-(n-i)}(S_n - K)_+|\mathcal{F}_i)$$
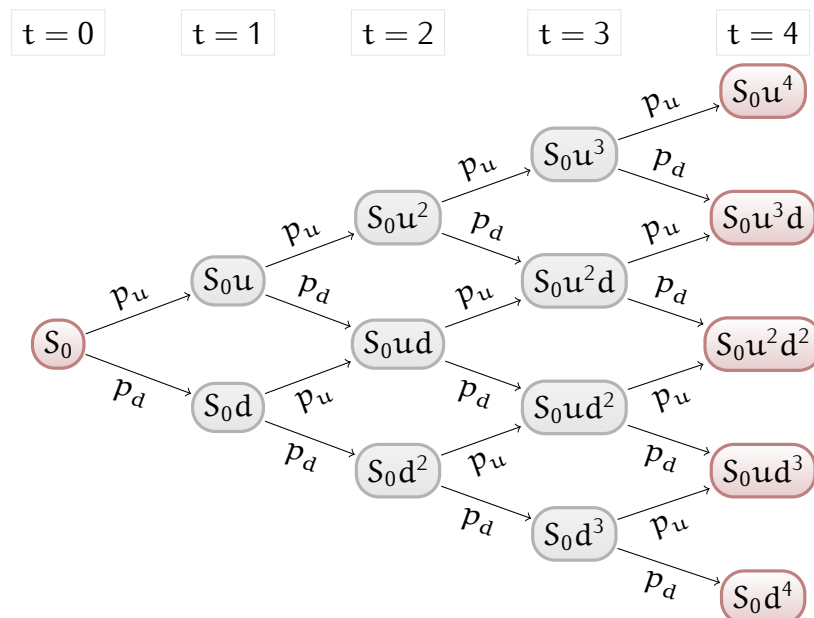

FIGURE 2.3. A 4 step lattice ($n = 4$)

Computational details   If we are given the contingency claim (which is the function of the underlying as stated in the contract), and the option is of the European type, it becomes

very simple to evaluate it value in this model. While going through the algorithm, it is suggested to keep in mind Figure ??.

Algorithm 2.1: Pricing European options in Cox-Ross–Rubinstein model
---
Input: Claim $h(x)$, Number of time steps n, Volatility σ
Output: The price of the option at the initial time
1 **begin**
2   Calculate p using Equation ??.
3   Calculate u and d using Equations ??.
4   Construct the binomial tree.
5   Use the claim to calculate the values that the option might take at maturity.
6   **for** $i \, 2 \, \{n-1, \ldots, 0\}$ **do**
7   **foreach** *node* **do**
8   Calculate the expected value of the option using the weighted mean of the prices at the $(i+1)^{\text{th}}$ time which are connected to this node. The weighting is done by the probabilities p and $1-p$.
9   **return** *Value at time 0*

2.2.3 American options

Since an American option can be exercised at any time $i \, 2 \, [n]$, at any time, the bearer of the option has two options – to exercise or to hold. The exercise value at any time is given by $Z_i = h(S_i)$, which in the case of a call is $Z_i = (S_i - K)_+$. Let the value of the options at a time i be denoted by U_i. We start by backward induction. At maturity, we must have $U_n = Z_n$. The value of holding the option at the time just before the maturity ($i = n-1$) is given by the discounted risk-neutral price $\mathbb{E}^*(R^{-1}h(S_n)|\mathcal{F}_{n-1}) = \mathbb{E}^*(R^{-1}U_n|\mathcal{F}_{n-1})$. The value of the option is thus given by the maximum of these two values, or

$$(2.8) \qquad U_{n-1} = \max\left\{Z_{n-1}, \mathbb{E}^*(R^{-1}U_n|\mathcal{F}_{n-1})\right\}.$$

By induction, we write the price of an American option at any time $i \, 2 \, [n]$.

$$(2.9) \qquad \begin{cases} U_i = \max\left\{Z_i, \mathbb{E}^*(R^{-1}U_{i+1}|\mathcal{F}_i)\right\} & 8i \, 2 \, [n-1] \\ U_n = Z_n := h(S_n). \end{cases}$$

Note that Equation ?? may be equivalently represented using discounted values.

$$(2.10) \qquad \begin{cases} \tilde{U}_i = \max\left\{\tilde{Z}_i, \mathbb{E}^*(\tilde{U}_{i+1}|\mathcal{F}_i)\right\} & 8i \, 2 \, [n-1] \\ \tilde{U}_n = \tilde{Z}_n. \end{cases}$$

This is sufficient to understand Algorithm ??, which tells the mechanism of pricing American options under the Cox–Ross-Rubinstein model. But we will go ahead with some theoretical results regarding the American options, because we will use the results in Chapter ??.

Super-martingales and discounted prices The discounted prices of American options are super-martingales under P^*, in contrast to European options. This is evinced from Equation ??, since $\tilde{U}_i \geqslant \check{Z}_i$ $\forall i$. In fact, this is the smallest super-martingale that dominates $(Z_n)_n$ (see [**Lamberton1996**]). Historically, $\tilde{U}_i$ is called the *Snell envelope* of the process $(\check{Z}_i)_i$.

Stopping time The owner of an American options may exercise his/her right at any point of time. For the decision to be fair, the owner should be able to make the choice with the information available only at till that time i. To model this, we need the definition of a stopping time.

DEFINITION 2.3 (stopping time). A random variable τ, taking values in $[n]$ is called a stopping time, if for any $i \in [n]$,
$$\{\tau = i\} \in \mathcal{F}_i$$

An equivalent characterisation of a stopping time is $\{\tau \leqslant i\} \in \mathcal{F}_i$.

Process stopped at a stopping time A process is $(X_i)_i$ is said to be adapted to a filtration $(\mathcal{F}_i)_i$ if X_i is $\mathcal{F}_i$-measurable $\forall i$.

Let $(X_i)_i$ be a process adapted to the filtration $(\mathcal{F}_i)_i$ and let τ be a stopping time. Then, the process stopped at τ is defined as
$$X_i^\tau(\omega) := X_{\tau(\omega) \wedge i}(\omega).$$

On the set $\{\tau = m\}$, this is equivalent to
$$X_i^\tau(\omega) = \begin{cases} X_i & \forall i < m \\ X_m & \forall i \geqslant m \end{cases}.$$

Below we state some results without proofs. The proofs may be found in [**Lamberton1996**].

PROPOSITION 2.1 (Stopped (super) martingales are (super) martingales). *If $(X_i)_i$ is an adapted process and τ is a stopping time, then $(X_i^\tau)_{i\in[n]}$ is adapted. Furthermore, if $(X_i)_i$ is a (super) martingale, then so is (X_i^τ).*

PROPOSITION 2.2. *The random variable defined by $\tau_0 = \inf\{i \geqslant 0 \mid U_i = Z_i\}$ is a stopping time, and $(U_i^{\tau_0})$ is a martingale.*

We denote by $\mathcal{T}_{i,n}$ the set of stopping times taking values in $\{i, i+1, \dots, n\}$. Since Ω is finite, $\mathcal{T}_{i,n}$ is also finite. The martingale property of the process $(U_i^{\tau_0})$ gives the following result, relating the concepts of Snell envelope to the optimal stopping problem. The idea portrayed is that if Z_n represents the total winnings of a gambler after n games, then stopping at time τ_0 maximises the expected gain given $\mathcal{F}_0$ (the information at time 0).

PROPOSITION 2.3. *The stopping time τ_0 satisfies*
$$U_0 = \mathbb{E}^*(Z_{\tau_0} \mid \mathcal{F}_0) = \sup_{\tau \in \mathcal{T}_{0,n}} \mathbb{E}^*(Z_\tau \mid \mathcal{F}_0)$$

Generalising the above proposition, we get

$$U_i = \sup_{\tau \in \mathcal{T}_{i,n}} \mathbb{E}^*(Z_\tau \mid \mathcal{F}_i) = \mathbb{E}^*(Z_{\tau_i} \mid \mathcal{F}_i), \tag{2.11}$$

where $\tau_i = \inf\{j \geqslant i \mid U_j = Z_j\}$.

DEFINITION 2.4 (optimal stopping time). A stopping time τ is called optimal for the process $(Z_i)_i$ if

$$\mathbb{E}^*(Z_\tau \mid \mathcal{F}_0) = \sup_{\tau \in \mathcal{T}_{0,n}} \mathbb{E}^*(Z_\tau \mid \mathcal{F}_0) \tag{2.12}$$

THEOREM 2.3. *A stopping time τ is optimal if and only if both of the following hold.*

$$\begin{cases} U_\tau = Z_\tau \\ (U_i^\tau)_{i \in [n]} \text{ is a martingale} \end{cases}$$

Computational details   If we are given the contingency claim, and the option is of the American type, it becomes very simple to evaluate it in this model. The only difference with the European options is that at each node, we select the higher value between the prices for immediate exercise and holding the option till the next time. While going through the algorithm, it is suggested to keep in mind Figure **??**.

Algorithm 2.2: Pricing American options in Cox–Ross–Rubinstein model

Input: Claim $h(x)$, Number of time steps n, Volatility σ
Output: The price of the option at the initial time

1 **begin**
2 | Calculate p using Equation **??**.
3 | Calculate u and d using Equations **??**.
4 | Construct the binomial tree.
5 | Use the claim to calculate the values that the option might take at maturity.
6 | **for** $i \in \{n-1, \ldots, 0\}$ **do**
7 | | **foreach** *node* **do**
8 | | | Calculate the expected value of the option using the weighted mean of the prices at the $(i+1)^{\text{th}}$ time which are connected to this node. The weighting is done by the probabilities p and $1-p$.
9 | | | Calculate the value of exercising the option.
10 | | | Select the higher value among the two numbers obtained above.
11 | **return** *Value at time 0*

2.3   Continuous models – the Black-Scholes model

We now consider a continuous model, meaning that the times are continuous and the prices do not have discrete jumps, but are continuous random variables. The most famous model is the Black–Scholes model proposed by Black and Scholes in 1973 [**Black1973**].

As in the previous chapter, we consider two assets – one riskless and one risky.

The value of the riskless asset is proportional to the instantaneous rate of interest, the current value of the asset, and the time. Therefore, the dynamics of the riskless asset is given as follows.

$$(2.13) \qquad\qquad dS_t^0 = rS_t^0\,dt$$

$$(2.14) \qquad\qquad S_0^0 = 1$$

Solving this initial value problem, we get

$$(2.15) \qquad\qquad S_t^0 = e^{rt}.$$

The risky asset is dependent on both deterministic and stochastic factors. The deterministic part is proportional to the current value of the asset and the time. The non–deterministic part is dependent on its current value and a stochastic process, whose mean should be zero, and variance should be proportional to the time. We choose these criteria because we do not expect the process to have a bias on going up or down (mean zero) and we expect that the process has a higher probability of going away from the current value with more time. It so happens that the best candidate for such a process is a standard Brownian motion[1].

Hence, Black and Scholes proposed the solution of the following stochastic differential equation with initial value as a model for the price dynamics.

$$(2.16a) \qquad\qquad dS_t = \mu S_t\,dt + \sigma S_t\,dB_t$$

$$(2.16b) \qquad\qquad S_0 = s_0.$$

Here B_t be a Brownian motion, and μ (drift) and σ (volatility) are proportionality constants. We shall assume that $\sigma > 0$, for if $\sigma = 0$, the process becomes deterministic, and $\sigma < 0$ is equivalent to $\sigma > 0$ because a Brownian motion is symmetric with respect to the time axis.

The solution to this problem is the geometric Brownian motion

$$(2.17) \qquad\qquad S_t = s_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma B_t}.$$

Before we proceed further in this section, we state the Radon–Nikodym Theorem and the Girsanov theorem.

THEOREM 2.4 (Radon-Nikodym). *Let $(\Omega, \mathcal{F})$ be a measurable space, with associated σ-finite measures P and Q such that $Q \ll P$. Then there exists an $\mathcal{F}$-measurable function f such that*

$$Q(A) = \int_A f\,dP \qquad \forall A \in \mathcal{F}$$

DEFINITION 2.5 (Radon-Nikodym derivative). In Theorem **??**, the function f is called the Radon-Nikodym derivative of Q with respect to P, and is denoted as $f = \frac{dQ}{dP}$.

THEOREM 2.5 (Girsanov). *Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_t, P)$ be a filtered probability space, $(B_t)_t$ be a $(\mathcal{F}_t)_t$-Brownian motion on the bounded interval $[0, T]$, and $\theta \in \mathbb{R}$. Let the probability P^* be defined by the Radon-Nikodym derivative*

$$(2.18) \qquad\qquad \frac{dP^*}{dP} = e^{\theta B_T - \frac{\theta^2}{2}T}.$$

Then, the following holds.

---

[1]For more information on Brownian motion, refer to [**Lamberton1996**]

$$P^* \sim P$$

The process $Z_t := \mathbb{E}\left(\frac{dP^*}{dP} \mid \mathcal{F}_t\right)$ *is a* P*-martingale*

The process $W_t := B_t + \theta t$ *is a* P^**-Brownian motion*

Proof. Refer to [**Lamberton1996**]   □

First, we need to show that there exists a probability measure P^* equivalent to P under which the discounted share price $\tilde{S}_t = e^{-rt}S_t$ is a martingale. From Equation ?? and Itô's Lemma (see [**Oksendal2003stochastic**]), we get:

$$
\begin{aligned}
d\tilde{S}_t &= -re^{-rt}S_t\,dt + e^{-rt}\,dS_t \\
&= -r\tilde{S}_t\,dt + e^{-rt}(\mu S_t\,dt + \sigma S_t\,dB_t) \\
&= (-r\tilde{S}_t + \mu\tilde{S}_t)\,dt + \sigma\tilde{S}_t\,dB_t \\
&= \tilde{S}_t((\mu - r)\,dt + \sigma\,dB_t)
\end{aligned}
$$

Let $W_t = B_t + \frac{\mu - r}{\sigma}t$ (since $\sigma \ne 0$). The we derive the following.

$$\text{(2.19a)} \qquad\qquad\qquad W_t = B_t + \frac{\mu - r}{\sigma}t$$

$$\Rightarrow \qquad\qquad dW_t = dB_t + \frac{\mu - r}{\sigma}\,dt$$

$$\Rightarrow \qquad\qquad \sigma\tilde{S}_t\,dW_t = \tilde{S}_t((\mu - r)\,dt + \sigma\,dB_t)$$

$$\text{(2.19b)} \qquad \Rightarrow \qquad\qquad \sigma\tilde{S}_t\,dW_t = d\tilde{S}_t$$

From Theorem ??, with $\theta = \frac{\mu - r}{\sigma}$, we know that there exists probability $P^* \sim P$ given by $\frac{dP^*}{dP} = e^{\theta B_t - \frac{\theta^2}{2}t}$, under which $(W_t)_t$ is a standard Brownian motion. The definition of stochastic integral is invariant by change of equivalent probability. Therefore, from Equation ??b, under the probability P^*, the discounted risky asset prices become

$$\text{(2.20a)} \qquad\qquad\qquad \tilde{S}_t = s_0 e^{\sigma W_t - \frac{\sigma^2}{2}t}$$

$$\text{(2.20b)} \qquad\qquad \Rightarrow \qquad S_t = s_0 e^{(r - \frac{\sigma^2}{2})t + \sigma W_t}.$$

To derive a SDE under the risk–neutral probability, we use Itô's lemma to get

$$\text{(2.21)} \qquad\qquad\qquad \frac{dS_t}{S_t} = r\,dt + \sigma\,dW_t.$$

We verify that $\tilde{S}_t$ is a P^*-martingale. Recall that we denote the set of all $\mathcal{F}$-measurable functions by $\mu\mathcal{F}$.

$$\mathbb{E}^*\left(\tilde{S}_{t+s}\mid\mathcal{F}_t\right) = \mathbb{E}^*\left(s_0 e^{\sigma W_{t+s}-\frac{\sigma^2}{2}(t+s)}\mid\mathcal{F}_t\right)$$

$$= \mathbb{E}^*\left(e^{\sigma\overbrace{(W_{t+s}-W_t)}^{\perp\mathcal{F}_t}}\overbrace{e^{\sigma W_t}}^{\in\mu\mathcal{F}_t}\overbrace{s_0 e^{-\frac{\sigma^2}{2}(t+s)}}^{\in\mu\mathcal{F}_0}\mid\mathcal{F}_t\right)$$

$$= s_0 e^{\sigma W_t-\frac{\sigma^2}{2}t}e^{-\frac{\sigma^2}{2}s}\mathbb{E}^*\left(e^{\sigma(W_{t+s}-W_t)}\right)$$

$$= \tilde{S}_t e^{-\frac{\sigma^2}{2}s}e^{\frac{\sigma^2}{2}s}$$

$$= \tilde{S}_t$$

The last expectation was calculated using the fact that $(W_{t+s}-W_t)\sim\mathcal{N}(0,s)$, and that the moment generating function[2] of $\mathcal{N}(\mu,\sigma)$ is given by $M(t)=e^{\mu t+\frac{\sigma^2 t^2}{2}}$.

REMARK 2.1 (Continuous dividend yield). If the risky asset pays continuous dividends, the only modification that needs to be done is to replace r in all the equations by the effective interest rate $r-q$.

Pricing the European call   From Equation **??**, we have:

$$\frac{S_T}{S_t} = \frac{s_0 e^{(r-\frac{\sigma^2}{2})T+\sigma W_T}}{s_0 e^{(r-\frac{\sigma^2}{2})t+\sigma W_t}}$$

$$= e^{(r-\frac{\sigma^2}{2})\overbrace{(T-t)}^{:=\Delta t}+\sigma\overbrace{(W_T-W_t)}^{:=\Delta W_t}}$$

$$\implies\qquad S_T = S_t e^{(r-\frac{\sigma^2}{2})\Delta t+\sigma\Delta W_t}$$

The value of a contingency claim then becomes

$$v_t = \mathbb{E}^*\left(e^{-r\Delta t}\,h(S_T)\big|\mathcal{F}_t\right)$$

$$= \mathbb{E}^*\left(e^{-r\Delta t}\,h\left(S_t e^{(r-\frac{\sigma^2}{2})\Delta t+\sigma\Delta W_t}\right)\big|\mathcal{F}_t\right).$$

Since the process $(S_t)_t$ is Markovian, we may write $\mathbb{E}^*(\ \mid\mathcal{F}_t)=\mathbb{E}^*(\ \mid S_t)$. By conditional expectation, the latter is a measurable function of t and S_t, denoted by $F(t,S_t)$. Moreover, since S_t is $\mathcal{F}_t$-measurable, we can treat it as a constant x. The value of the call can then be represented by $V_t=F(t,S_t)$, where $F(t,x)=\mathbb{E}^*\left(e^{-r\Delta t}\,h\left(S_t e^{(r-\frac{\sigma^2}{2})\Delta t+\sigma\Delta W_t}\right)\big|S_t=x\right)$.

By independence of increments of the Brownian motion W ($\Delta W_t\ \widetilde{?}\ \sigma(W_s:s\leqslant t)$), the

---

[2] https://en.wikipedia.org/wiki/Moment-generating_function

conditional expectation reduces to a simple expectation, and we write

$$F(t, x) = \mathbb{E}^* \left(e^{-r\Delta t} \, h \left(x e^{(r - \frac{\sigma^2}{2})\Delta t + \sigma \Delta W_t} \right) \right)$$

For the call, we put $h(x) = (x - K)_+$, and denote $F(t, x)$ by $c(t, x)$.

$$c(t, x) = \mathbb{E}^* \left(e^{-r\Delta t} \, \left(x e^{(r - \frac{\sigma^2}{2})\Delta t + \sigma \Delta W_t} - K \right)_+ \right)$$

$$= x \underbrace{\mathbb{E}^* \left(e^{\sigma \Delta W_t - \frac{\sigma^2}{2}\Delta t} \mathbb{1}_{\left\{ x e^{(r - \frac{\sigma^2}{2})\Delta t + \sigma \Delta W_t} > K \right\}} \right)}_{A} + K e^{-r\Delta t} \underbrace{\mathbb{E}^* \left(\mathbb{1}_{\left\{ x e^{(r - \frac{\sigma^2}{2})\Delta t + \sigma \Delta W_t} > K \right\}} \right)}_{B}$$

We deal with B first as it is more manageable.

$$B = \mathbb{E}^* \left(\mathbb{1}_{\left\{ x e^{(r - \frac{\sigma^2}{2})\Delta t + \sigma \Delta W_t} > K \right\}} \right)$$

$$= P^* \left(x e^{(r - \frac{\sigma^2}{2})\Delta t + \sigma \Delta W_t} > K \right)$$

$$= P^* \left(\frac{\Delta W_t}{\sqrt{\Delta t}} > \frac{\log \frac{K}{x} - (r - \frac{\sigma^2}{2})\Delta t}{\sigma \sqrt{\Delta t}} \right)$$

$$= P^* \left(\frac{\Delta W_t}{\sqrt{\Delta t}} < - \frac{\log \frac{K}{x} - (r - \frac{\sigma^2}{2})\Delta t}{\sigma \sqrt{\Delta t}} \right)$$

$$= P^* \left(\frac{\Delta W_t}{\sqrt{\Delta t}} < \frac{\log \frac{x}{K} + (r - \frac{\sigma^2}{2})\Delta t}{\sigma \sqrt{\Delta t}} \right)$$

Now, $\Delta W_t \sim N(0, \Delta t)$, implying $\frac{\Delta W_t}{\sqrt{\Delta t}} \sim N(0, 1)$. Writing

$$d_\ominus(\Delta t, x; K, \sigma, r) = \frac{\log \frac{x}{K} + (r - \frac{\sigma^2}{2})\Delta t}{\sigma \sqrt{\Delta t}},$$

we get:

$$B = P^* \left(N(0, 1) < d_\ominus \right)$$

$$=) \qquad B = \mathcal{N}(d_\ominus(\Delta t, x; K, \sigma, r)),$$

where $\mathcal{N}$ is the (cumulative) distribution function of the standard normal distribution, given by

(2.22) $$\mathcal{N}(x) = \frac{1}{2\pi} \int_{-\infty}^{x} e^{-\frac{z^2}{2}} \, dz.$$

Similarly, we may now evaluate A.

$$A = \mathbb{E}^* \left(e^{\sigma \Delta W_t - \frac{\sigma^2}{2}\Delta t} \mathbb{1}_{\left\{ x e^{(r-\frac{\sigma^2}{2})\Delta t + \sigma \Delta W_t} > K \right\}} \right)$$

$$= \mathbb{E}^* \left(e^{\sigma \sqrt{\Delta t}\frac{\Delta W_t}{\sqrt{\Delta t}} - \frac{\sigma^2}{2}\Delta t} \mathbb{1}_{\left\{ \frac{\Delta W_t}{\sqrt{\Delta t}} > -d_\ominus \right\}} \right)$$

$$= \int_{-d_\ominus}^{\infty} e^{\sigma \sqrt{\Delta t}\, z - \frac{\sigma^2}{2}\Delta t} \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}\, dz$$

$$= \int_{-d_\ominus}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(z - \sigma\sqrt{\Delta t})^2}{2}}\, dz$$

$$= \int_{-d_\oplus}^{\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}\, dy$$

$$= \int_{-\infty}^{d_\oplus} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}\, dy$$

$$= \mathcal{N}(d_\oplus(\Delta t, x; K, \sigma, r))$$

The pre-penultimate step is obtained by substituting $y = z - \sigma\sqrt{\Delta t}$. Then the new lower limit, denoted $-d_\oplus$, is given as follows.

$$-d_\oplus(\Delta t, x; K, \sigma, r) := -\frac{\log \frac{x}{K} + (r - \frac{\sigma^2}{2})\Delta t}{\sigma\sqrt{\Delta t}} - \sigma\sqrt{\Delta t}$$

$$= -\frac{\log \frac{x}{K} + (r + \frac{\sigma^2}{2})\Delta t}{\sigma\sqrt{\Delta t}}$$

$$\implies \quad d_\oplus(\Delta t, x; K, \sigma, r) = \frac{\log \frac{x}{K} + (r + \frac{\sigma^2}{2})\Delta t}{\sigma\sqrt{\Delta t}}$$

Therefore, we have $d_\oplus(\Delta t, x; K, \sigma, r) = \frac{\log \frac{x}{K} + (r \pm \frac{\sigma^2}{2})\Delta t}{\sigma\sqrt{\Delta t}}$.

Henceforth, we shall write $d_\oplus$ to denote $d_\oplus(\Delta t, x; K, \sigma, r)$.

Assembling all the pieces together we obtain the famous Black–Scholes formula for a European call.

(2.23) $$\qquad\qquad c_t = c(t, S_t) = S_t \mathcal{N}(d_\oplus) - K e^{-r\Delta t} \mathcal{N}(d_\ominus)$$

Pricing the put   Employing the call-put parity, we get

(2.24) $$\qquad\qquad p_t = p(t, S_t) = K e^{-r\Delta t} \mathcal{N}(-d_\ominus) - S_t \mathcal{N}(-d_\oplus).$$

REMARK 2.2 (Hedging). Even though we shall not derive the hedging strategy, we comment that in the Black–Scholes model, derivatives are hedgeable, and is determined by the so called *delta*, where $\Delta := \frac{\partial c}{\partial x}(t, S_t) = \mathcal{N}(d_\oplus)$. The hedging strategy which uses the delta is called *delta hedging*. In the real-world, there are other sources of risk which needs different hedging strategies, as highlighted in the limitations of the model.

Limitations

Finding closed–form formulae for prices of American options and most exotic options is not possible.

Security prices in reality do not follow a strict stationary log–normal process.

The risk–free interest rate is not known, and is not constant over time.

The volatility is not constant and depends on current value of the underlying (the curve is called the *volatility smile*), and on the time to maturity (the curve is called the *volatility skew*).

The model underestimates extreme moves, yielding *tail risk*, which can be hedged with out–of–the–money options.

Its assumption of instant, cost-less trading yields *liquidity risk*, which is difficult to hedge.

Its assumption of a stationary process yields volatility risk, which can be hedged with *volatility hedging*.

Its assumption of continuous time and continuous trading yields *gap risk*, which can be hedged with *Gamma hedging*.

Benefits   In spite of its limitations, the Black–Scholes model remain one of the most useful models in Mathematical Finance. Below are some advantages of the model.

computability: The ease of computation is one of the biggest advantages of the model.

useful approximation: It serves as decent approximation of the real world, particularly when analysing the direction in which prices move when crossing critical points.

basis: It serves as a benchmark for more refined models, which try to address its drawbacks.

reversible: The model's original output, price, can be used as an input and one of the other variables solved for; the implied volatility calculated in this way is often used to quote option prices.

2.3.1   Discrete and continuous – different worlds?

Even though it may seem that the Cox-Ross-Rubinstein model and the Black–Scholes models are fundamentally different, it can be shown that as the number of time steps increase to infinity, the former converges to the latter in distribution.

The idea is that we rescale the Cox-Ross-Rubinstein model in time and space and use the central limit theorem. We assume

More trading periods $t \, 2 \, \frac{iT}{n} \quad \mathcal{S}i \, 2 \, [n]$

Smaller price increments: $T_i^n \in \left\{ \frac{d}{\sqrt{n}}, \frac{u}{\sqrt{n}} \right\} \implies \frac{1}{T} \mathbb{V}ar_n(T_i^n) = \frac{\sigma}{\sqrt{n}}$

Continuous compounding

THEOREM 2.6 ([3]). *The distribution of the asset price S_n^n at maturity under P_n^* converges to the distribution of the log-normal random variable $S_T = S_0 e^{(r - \frac{\sigma^2}{2})T + \sigma W_T}$, where $W_T \sim N(0, T)$.*

Moreover, if f is a continuous function on $(0, \infty)$, such that $|f(x)| \leqslant c(1 + x)^q$ for some $c > 0$ and $0 \leqslant q < 2$, then $\mathbb{E}^* \left(f(S_n^n) \right) \to \mathbb{E}^* \left(f(S_T) \right)$.

The proof is quite long, and we shall not prove it here. We refer the interested reader to [**He1989**] for a detailed analysis.

What is more interesting to us is that with the help of this result, we may use the Cox–Ross–Rubinstein model to price an option, and we would know that the result is guaranteed to converge to the Black–Scholes price, even if such a price cannot be calculated directly.

2.3.2   Black-Scholes model – other options

We have seen that it is quite easy to compute the price of European options in the Black–Scholes model. On the other hand, pricing most other options is not so straight forward. For most options, we do not have closed–form solutions in the Black–Scholes model. Examples of such options include American options and most exotic options.

On the other hand, the Cox–Ross–Rubinstein, even though computationally much more demanding, does allow pricing of all options in a straightforward manner. In fact, even though the theoretical foundations are quite different between European and American options, the similarity among the Algorithms **??** and **??** testifies to this fact. This is because unlike in the Black–Scholes model, we do not calculate a closed form formula for the prices in the Cox–Ross–Rubinstein model. Rather, we utilise the recursive procedure that does not call for an explicit evaluation formula. This universal computability is one of the primary advantage of discrete models. Our trust on the prices computed using discrete models is bolstered by the fact that they do converge to the continuous models in certain cases – a prime example being the convergence of the Cox–Ross–Rubinstein model to the Black–Scholes one.

In light of the above, a good balance has been struck by algorithms which develop on the discrete models in order to decrease the computational complexity by allowing for approximations. The sacrifice in accuracy is justified by the significant improvement in computability. One such algorithm is the *singular points method* introduced by Gaudenzi, Zanette and Lepellere.

In the next two chapters, we shall deal with Asian options of the American type, and cliquet options, respectively. These options do not lend themselves to be priced in closed–form formulae in the Black–Scholes model. The singular points method is a viable alternative method for these options, and we focus our study on its theory and extensibility, stating also the numerical results.

---

[3]http://horst.qfl–berlin.de/sites/files/u4/CMM2.pdf

Chapter 3

Asian options

In the last chapter, we saw that exotic options usually cannot be priced in a closed-form formula using the Black-Scholes model. The way out is by using numerical methods. These methods range from using discrete models (which converge to continuous models, as remared in Chapter **??**), to discretisation of the Black-Scholes PDE or using Monte Carlo simulations.

The advantage of using the Cox-Ross-Rubinstein model is that it converges to the Black-Scholes model as the number of time steps increases to infinity. But the exponential number of paths (2^n to be exact, where n is the number of time steps) make the method very slow and memory intensive, making it computably impractical. A logical step would be to modify the basic Cox-Ross-Rubinstein model to allow for approximations. In this direction, Gaudenzi et al[**Gaudenzi2010**] introduced a new method called the *singular points method* for pricing certain path-dependent options in an efficient manner.

In this chapter, we will mainly focus on applying the singular points method to price Asian options. In Asian options, the price is expressed as a function of some form of averaging on the underlying's price. Popular Asian options use the arithmetic or geometric means as the average. Again, Asian options may be exercised only at maturity (European) or at any time before maturity (American). They may give the owner of the option the right to either sell (put) or buy (call). Theoretically, we will only study calls, because the framework for puts one may be derived in the exact same way.

3.1 Literature Review

Before we go into the details of the singular points method, we shall look into the pre-existing methods of pricing Asian options, and discuss their advantages and disadvantages briefly. As we remarked in Section **??**, Asian options with arithmetic mean cannot be valued by closed-form formulae in the Black–Scholes model, and their valuation requires the use of numerical methods. Here we consider a tree method for pricing these types of options.

The main barrier to applying the Cox–Ross–Rubinstein method [**Cox1979**] introduced in Chapter **??** to Asian options with arithmetic averages is the exponential increase in the

number of paths that the underlying may take, and this increases the computational difficulty very quickly as we increase the number of time steps.

Tree methods   Alternative feasible approaches were proposed by Hull and White (1993) [**Hull1993**], and Barraquand and Pudet (1996) [**Barraquand1996**]. The main idea behind their procedures is to restrict the range of all the possible arithmetic averages to a set of representative values. These values are selected in order to span all the possible values of the averages achievable at each node of the tree. The price is then computed by a backward induction procedure in which the prices associated with averages not included in the set of representative values are obtained by interpolation. Both of these methods reduce the computational complexity to $O(n^3)$, n being the number of steps. Nevertheless, the advantage of speed is offset by the fact that it is difficult to control the precision of the approximations and the convergence to the continuous value. This was highlighted by Forsyth *et al* [**Forsyth2002**] in 2002. Forsyth *et al* also proved that a procedure of order $O(n^{\frac{7}{2}})$ is necessary in order to assure the convergence of these algorithms.

Later, Chalasani *et al* (1999) [**Chalasani1999**] proposed a totally different approach, which allowed them to obtain thin upper and lower bounds on the exact Cox–Ross–Rubinstein binomial price for American Asian options. Their method requires a forward procedure and a backward induction. This algorithm significantly increases the precision of the estimates but requires a very large amount of memory and has computational complexity $O(n^4)$.

PDE based methods   All of the above were tree methods. More recently, very efficient partial differential equation (PDE)–based methods have been introduced by Vecer (2001) [**Vecer2001**] and d'Halluin et al (2005) [**dHalluin2005**]. In Vecer's method, the price of the Asian option is characterized by a simple one-dimensional partial differential equation which could be applied to both continuous and discrete average Asian option. The compuatational complexity is $O(n^2)$. This approach cannot be applied to American fixed–strike Asian options, which, on the other hand, can be treated using the semi–lagrangian approach of d'Halluin *et al*.

Table **??** briefly reviews the discussion above.

Table 3.1. Pre-existing methods for Asian options

Method	Type	Complexity	Remarks
Binomial	Tree	$O(2^n)$	simple, accurate, convergence to continuous
Hull & White	Tree	$O(n^3)$	accuracy and convergence problems
Barraquand & Pudet	Tree	$O(n^3)$	accuracy and convergence problems
Chalasani et al	Tree	$O(n^4)$	thin bounds, but very large memory
Vecer	PDE	$O(n^2)$	not universally applicable
d'Halluin	PDE	NA	more general than Vecer

A number of these algorithms has been implemented in Premia 13. Premia is a software designed for option pricing, hedging and financial model calibration. It has been developed

by the 'MathFi' team in INRIA. It is provided with its C/C++ source code and an extensive scientific documentation. More information about Premia can be found at the dedicated website[1].

3.2   The exact binomial algorithm

In what follows, we shall assume that the evolution of the prices of the risky asset $(S_t)_t$ is governed by the Black–Scholes stochastic differential equation as discussed in Equation ?? of Chapter ??. Its solution is given by Equation ??b. Whenever there is a continuous dividend yield, we modify the equation according the remark ??.

Consider the discrete model. If the number of time steps in the binomial tree is n, then the corresponding time step is $\Delta T = \frac{T}{n}$. The lognormal diffusion process $(S_{i\Delta T})_{i\in[n]}$ is approximated by the Cox–Ross–Rubinstein binomial process (refer Equation ??).

$$S_i = s_0 \prod_{j=1}^{i} T_j \qquad \text{\&i } 2 \text{ [n]}.$$

As usual, we represent the risk–neutral probability by $p = \frac{R-d}{u-d}$, where $u = d^{-1} = e^{\sigma\sqrt{\Delta T}}$. We denote the effective rate of interest in each period as $R := e^{r\Delta T}$. We note that the R is not an instantaneous quantity, but one which is constant on an interval of time.

Asian options are dependent on the average prices of the underlying risky asset. The price of an Asian option of the American type with initial time 0 and maturity T is given by the risk–neutral expectation calculated at an optimal stopping time (see Definiton ?? from Chapter ??).

$$(3.1) \qquad P(0, s_0, a_0) = \sup_{\tau \in \mathcal{T}_{[0,T]}} \mathbb{E}^* \left(e^{-r\tau} \, h(S_\tau, A_\tau) \mid S_0 = s_0, A_0 = s_0 \right)$$

The quantities used in the formula are explained below.

  $\mathcal{T}_{[0,T]}$: the set of all stopping times with values in $[0, T]$

  h: the payoff function, dependent on both the underlying's price and its average

  S_τ: the underlying's price at time τ

  A_τ: (some form of) the average of the price of the underlying asset over the period $[0, \tau]$

Let K denote the strike price. The price function may be one of the following

  fixed Asian call: $h(A_T) = (A_T - K)_+$

  fixed Asian put: $h(A_T) = (K - A_T)_+$

  floating Asian call: $h(S_T, A_T) = (S_T - A_T)_+$

  floating Asian put: $h(S_T, A_T) = (A_T - S_T)_+$

---

[1]https://www.rocq.inria.fr/mathfi/Premia/

Definition 3.1 (arithmetic mean). The arithmetic mean of a set of numbers $\{S_i\}_{i \in [n]}$ is given by:

$$(3.2) \qquad\qquad A_n = \frac{\sum_{i=0}^{n} S_i}{n+1}$$

In the rest of the chapter, we will assume that the average means arithmetic mean, unless otherwise stated.

In the Cox–Ross–Rubinstein model, consider a node at time i. Let the price of the underlying be given by x, and the arithmetic mean of the underlying till time i be given by y. There are two possibilities for the asset price in the next time $t = i + 1$, namely xu and xd. The average corresponding to the up and down movements become:

$$A_{i+1}^u = \frac{(i+1)A_i + xu}{i+2} = \frac{(i+1)y + xu}{i+2}$$
$$A_{i+1}^d = \frac{(i+1)A_i + xd}{i+2} = \frac{(i+1)y + xd}{i+2}$$

The price at time 0 of the Asian option of the European type with payoff function $h(x, y)$ is given by $v(0, s_0, s_0)$ (since $a_0 = s_0$), where the functions $v(i, x, y)$ can be computed by the following backward dynamic programming equations, computed by the discounted risk–neutral expectation of the prices at the next time.

$$(3.3a) \qquad v(n, x, y) = h(x, y)$$

$$(3.3b) \qquad v(i, x, y) = \frac{1}{R}\left(pv\left(i+1, xu, \frac{(i+1)y + xu}{i+2}\right)\right.$$
$$\left. + (1-p)v\left(i+1, xd, \frac{(i+1)y + xd}{i+2}\right)\right) \qquad \forall i \, 2 \, [n-1]$$

In case of Asian options of the American type, we modify the equations accordingly.

$$(3.4a) \qquad v(n, x, y) = h(x, y)$$

$$(3.4b) \qquad v(i, x, y) = \max\left\{h(x, y), \frac{1}{R}\left(pv\left(i+1, xu, \frac{(i+1)y + xu}{i+2}\right)\right.\right.$$
$$\left.\left. + (1-p)v\left(i+1, xd, \frac{(i+1)y + xd}{i+2}\right)\right)\right\} \qquad \forall i \, 2 \, [n-1]$$

The payoff is a function of the average, which is clearly path–dependent. Thus, the option is path–dependent, and the corresponding price tree is non–recombinant. This makes the classical binomial method infeasible after a small number of steps. Note that the binomial tree for the underlying is always recombinant for constant volatility.

3.3   The singular points method

The price of an Asian option at each instance is a continuous function of the underlying's average. Since the number of paths to a node in a binomial tree is finite, we have that at each node of the underlying's binomial tree, the option price may be represented as a piecewise-linear, continuous, convex function of the average. We shall develop the theoretical idea in this section. In the subsequent section, we shall see that the nature of the function allows us to make approximations with *a priori* error bounds.

DEFINITION 3.2 (singular points and singular values). Let $P = (P_i)_{i \in [n]} = ((x_i, y_i))_{i \in [n]}$, $n \in \mathbb{N}$ be a sequence of points such that

(3.5a) $a = x_0 < x_1 < \quad < x_{n-1} < x_n = b$

(3.5b) $m_{i+1} := \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i} \leqslant \dfrac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}} = m_{i+2} \qquad \forall i \in \{1, \ldots, n-1\}.$

Let $f : [a, b] \to [0, \infty)$ be the function obtained by linear interpolation of the points in P. From the definition of f and ??, the function is continuous, piecewise-linear and convex.

Then, the2
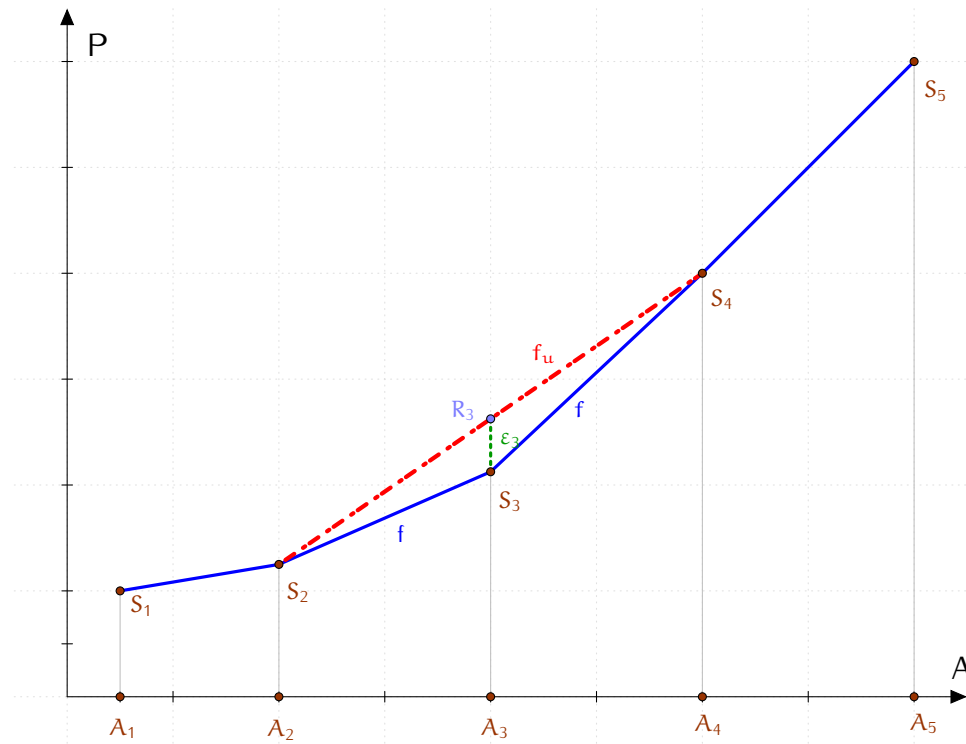
FIGURE 3.1. Illustration of Lemma ?? with $j = 3$

PROOF. By construction, $\forall x \notin (x_{j-1}, x_{j+1})$, we have $f_u(x) = f(x)$.

Again, by construction, $\forall x \in (x_{j-1}, x_{j+1}), f_u(x) = (1-t)f(x_{j-1}) + tf(x_{j+1})$, where $t = \frac{x - x_{j-1}}{x_{j+1} - x_{j-1}}$.

Now, we have:

$$
\begin{aligned}
& x_{j-1} < \quad\quad x \quad\quad < x_{j+1} \\
\Longrightarrow \quad & 0 < \quad x - x_{j-1} \quad < x_{j+1} - x_{j-1} \\
\Longrightarrow \quad & 0 < \frac{x - x_{j-1}}{x_{j+1} - x_{j-1}} < 1 \\
\Longrightarrow \quad & 0 < \quad\quad t \quad\quad < 1
\end{aligned}
$$

f is convex $\Longrightarrow$ $\forall t \in (0,1), f((1-t)x_{j-1} + tx_{j+1}) < (1-t)f(x_{j-1}) + tf(x_{j+1})$.

Thus, $f_u(x) \geqslant f(x) \; \forall x \in [a, b]$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \square$

Refer to Figure ?? for a graphical representation of the above lemma.

3.3.2 Lower estimates

LEMMA 3.2 (Lower estimate). *Let $f : [a, b] \to [0, \infty)$ be a continuous, piecewise-linear, convex function characterised by the singular points $P = ((x_i, y_i))_{i \in [n]}$. Let l_j be the line segment joining points P_{j-1} and P_j. Similarly, let l_{j+2} be the line segment joining points P_{j+1} and P_{j+2}. Denote the intersection of the line segments l_j and l_{j+2} by $\bar{P} = (\bar{x}, \bar{y})$.*
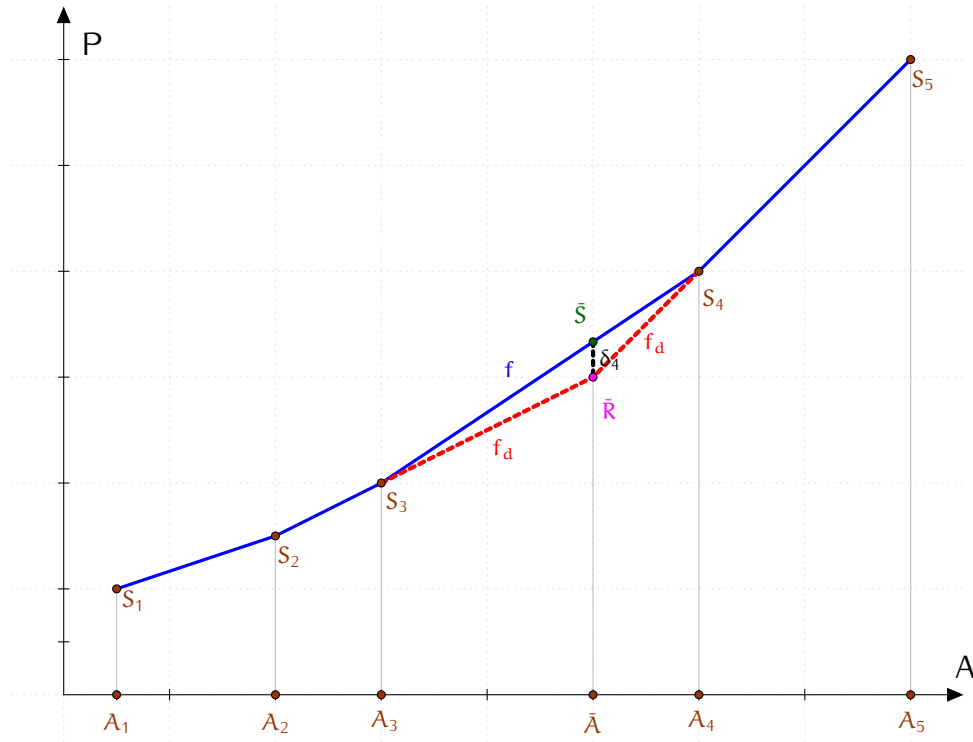
FIGURE 3.2. Illustration of Lemma ?? with $j = 3$

Then the function $f_d : [a, b] \to [0, \infty)$ *characterised by* $(P_0, \ldots, P_{j-1}, \bar{P}, P_{j+2}, \ldots, P_n)$ *is also continuous, piecewise-linear and convex, and*

$$(3.8) \qquad\qquad f_d(x) \leqslant f(x) \qquad \forall x \in [a, b]$$

PROOF. First we show the convexity of f_d. We know that f satisfies the property of increasing slopes, that is $m_i \leqslant m_{i+1} \leqslant m_{i+2}$. Since f_d is obtained from f by removing the line segment l_{j+1}, for f_d we have that $m_i \leqslant m_{i+2}$, which implies that the function f_d is still convex.

Secondly, to prove the inequality, we may look at the convex function f as if it has been obtained by removing point $\bar{P}$ from the convex function f_d. Then, if $\bar{x} \in (x_j, x_{j+1})$, we have, using Lemma ??, that $f_d(x) \leqslant f(x) \qquad \forall x \in [a, b]$. $\qquad\qquad\square$

Refer to Figure ?? for a graphical representation of the above lemma.

The lemmas ?? and ??, will be used later to reduce both the computational complexity and the memory requirement of the algorithm by removing points or edges, effectively simplifying the function.

3.3.3   Notations and conventions

In this and subsequent sections, we shall use the convention that $[n] = \{0, 1, 2, \ldots, n\}$.

Let the number of time steps be n. Let i denote the highlighted time step, and j represent the number of up movements. In this way, we may represent any node by $N_{i,j}$. For example, in Figure ??, the node denoted by $S_0 u^2 d$ would be represented as $N_{3,2}$.

The price of the underlying at each node $N_{i,j}$ is denoted by $S_{i,j}$. Since there are j up movements, there must be $i - j$ down movements, and thus

$$(3.9) \qquad S_{i,j} = S_0 u^j d^{i-j} = S_0 u^j u^{-(i-j)} = S_0 u^{-i+2j} \qquad \forall i \in [n], \ \forall j \in [i]$$

PROPOSITION 3.1. *The number of paths to a node $N_{i,j}$ is $\binom{i}{j}$.*

PROOF. At each point in a path, we may choose either an up movement or a down movement. To reach node $N_{i,j}$, we much choose j up movements among i possibilities. The result follows immediately. □

We denote the number of singular points in a node $N_{i,j}$ by $L_{i,j}$, where $L_{i,j} \in \left[\binom{i}{j}\right]$. The lth average (in ascending order) ($l \in \{1, \ldots, L_{i,j}\}$) is denoted by $A_{i,j}^l$, and the corresponding price by $P_{i,j}^l$. Thus the singular points characterising the price function are $((A_{i,j}^l, P_{i,j}^l))_{l \in \{1, \ldots, L_{i,j}\}}$.

DEFINITION 3.3 (singular average and singular price). In the particular case of Asian options with arithmetic mean, each $A_{i,j}^l$ is called a *singular average* and each $P_{i,j}^l$ is called a *singular price*.

We recall some basic definitions and derive simple results for the maximum and minimum attainable value of the averages on each node.

DEFINITION 3.4 (Path). A path is a sequence $(j_i)_{i \in [n]}$ such that $j_{i+1} \in \{j_i, j_i + 1\}$.

EXAMPLE 3.1. In Figure ??, two paths are shown using red/thicker and blue/thick arrows. The other arrows are in grey/thin. The two paths have the same value at maturity, but give different averages.
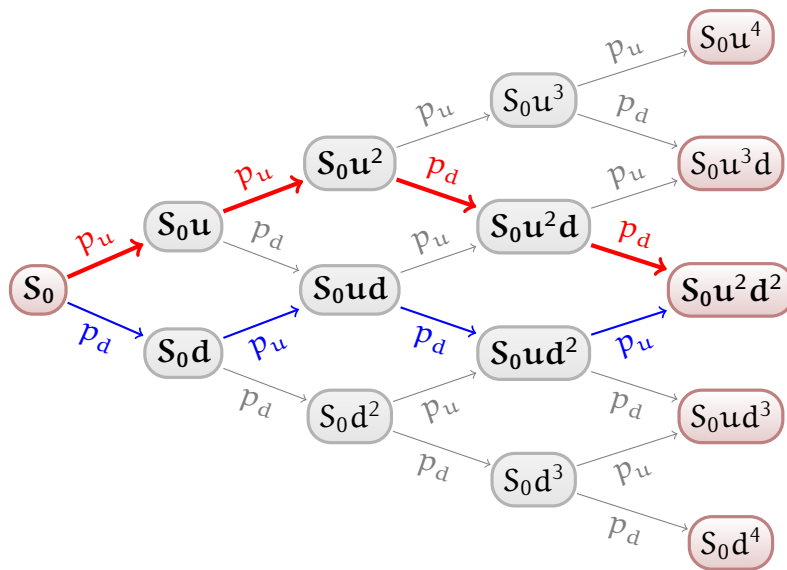

FIGURE 3.3. Different paths leading to a single destination

THEOREM 3.1 (Path inequality). *Let there be two paths α and β, such that $S_{i,j_i^\alpha} \geqslant S_{i,j_i^\beta}$ $\forall i$, where $(j_i^\alpha)_{i \in [n]}$ and $(j_i^\beta)_{i \in [n]}$ denote the paths as defined above. Denote the corresponding averages by A^α and A^β, respectively. Then $A^\alpha \geqslant A^\beta$.*

PROOF. Clearly if $S_{i,j_i^\alpha} = S_{i,j_i^\beta}$ $\forall i$, then $A^\alpha = A^\beta$.

We only need to show the result in the case of strict inequality at one time. Let $S_{i,j_i^\alpha} = S_{i,j_i^\beta}$ $\forall i \in [n] \setminus \{l\}$, and $S_{l,j_l^\alpha} > S_{l,j_l^\beta}$.

Now, from equation ??, we have:

$$(n+1)A_{n,j}^\alpha = \sum_{i=0}^{l-1} S_{i,j_i} + S_{l,j_l^\alpha} + \sum_{i=l+1}^{n} S_{i,j_i}$$

$$(n+1)A_{n,j}^\beta = \sum_{i=0}^{l-1} S_{i,j_i} + S_{l,j_l^\beta} + \sum_{i=l+1}^{n} S_{i,j_i}$$

$$\implies (n+1)\left(A_{n,j}^\alpha - A_{n,j}^\beta\right) = S_{l,j_l^\alpha} - S_{l,j_l^\beta}$$

$$= S_{l-1,j_{l-1}} u_l - S_{l-1,j_{l-1}} d_l$$

$$= S_{l-1,j_{l-1}}(u_l - d_l) > 0 \qquad (u_l > d_l \text{ by definition})$$

$$\implies A_{n,j}^\alpha > A_{n,j}^\beta$$

Iterating this procedure, we obtain the general case. $\qquad\qquad\square$

REMARK 3.3. The path α signifies a path *above* and β a path *below* in the usual depiction of the binomial tree (the up movement shown above the down movement). Thus, a path which never goes below another cannot have a lower arithmetic mean than the other.

COROLLARY 3.1. *At each node $N(i,j)$, the following hold:*

(1) *The minimum average possible $A_{i,j}^{\min}$ is attained by the path corresponding to the path corresponding to the path with $(i-j)$ down movements followed by j up movements, and*

$$(3.10) \qquad A_{i,j}^{\min} = \frac{S_0}{i+1}\left(\frac{1 - d^{i-j+1}}{1-d} + d^{i-j}u\frac{1-u^j}{1-u}\right)$$

(2) *The maximum average possible $A_{i,j}^{\max}$ is attained by the path corresponding to the path with j up movements followed by $(i-j)$ down movements, and*

$$(3.11) \qquad A_{i,j}^{\max} = \frac{S_0}{i+1}\left(\frac{1 - u^{j+1}}{1-u} + u^j d\frac{1-d^{i-j-1}}{1-d}\right)$$

PROOF. We show the proof only for the case of the maximum, since the case of the minimum can be shown by using the exact same argument.

From Theorem ??, the result about path with the maximum average holds directly, since there cannot be a path above the one given by j up movements followed by $(i-j)$ down movements.

The subsequent formula may be derived as follows.

$$(i+1)A_{i,j}^{max} = \underbrace{(S_0 + S_0 u + S_0 u^2 + \quad + S_0 u^j)}_{\text{up movement}} + \underbrace{(S_0 u^j d + S_0 u^j d^2 + \quad + S_0 u^j d^{i-j})}_{\text{down movement}}$$

$$= S_0((1 + u + u^2 + \quad + u^j) + u^j d(1 + d + \quad + d^{i-j-1}))$$

$$= S_0 \left(\sum_{k=0}^{j} u^k + u^j d \sum_{k=0}^{i-j-1} d^k \right)$$

$$= S_0 \left(\frac{1 - u^{j+1}}{1 - u} + u^j d \frac{1 - d^{i-j-1}}{1 - d} \right) \qquad (\text{Geometric series}) = A_{i,j}^{max}$$

piecewise-linear, convex and continuous, and is characterised by these points. In essence, we consider not only averages that are effectively achievable but all the possible averages between the minimum and maximum realized at that point. This gives us the continuous representation of prices. The intuitive idea is that as the time step is reduced to zero, this function converges to the price function of the continuous time model.

We start with the prices at maturity, and proceed using backward iteration. The exact details are explained below.

At maturity ($i = n$)

From equations ?? and ??, putting $i = n$, we get

$$A_{n,j}^{\min} = \frac{S_0}{n+1}\left(\frac{1 - d^{n-j+1}}{1-d} + d^{n-j}u\frac{1-u^j}{1-u}\right)$$

$$A_{n,j}^{\max} = \frac{S_0}{n+1}\left(\frac{1 - u^{j+1}}{1-u} + u^j d\frac{1-d^{n-j-1}}{1-d}\right)$$

In defining the price function, we note that three cases may arise.

$j \in \{0, n\}$
In this case, there can be only one path to these nodes, so there is only one average, implying one price and one singular point.

$j \notin \{0, n\}$ and $K \in (A_{n,j}^{\min}, A_{n,j}^{\max})$
In this case, the price function is characterised by three singular points ($L_{i,j} = 3$), $(A_{n,j}^l, P_{n,j}^l)_{l \in \{1,2,3\}}$, since we need to compare the averages with the fixed strike price K. The points are as follows.

(3.13)
$$\begin{aligned}
(A_{n,j}^1, P_{n,j}^1) &= (A_{n,j}^{\min}, 0) \\
(A_{n,j}^2, P_{n,j}^2) &= (K, 0) \\
(A_{n,j}^3, P_{n,j}^3) &= (A_{n,j}^{\max}, A_{n,j}^{\max} - K)
\end{aligned}$$

$j \notin \{0, n\}$ and $K \notin (A_{n,j}^{\min}, A_{n,j}^{\max})$
In this case, the price function is characterised by only two singular points ($L_{i,j} = 2$), $(A_{n,j}^l, P_{n,j}^l)_{l \in \{1,2\}}$, which are as follows.

(3.14)
$$\begin{aligned}
(A_{n,j}^1, P_{n,j}^1) &= (A_{n,j}^{\min}, (A_{n,j}^{\min} - K)_+) \\
(A_{n,j}^2, P_{n,j}^2) &= (A_{n,j}^{\max}, (A_{n,j}^{\max} - K)_+)
\end{aligned}$$

LEMMA 3.3 (Price function at maturity ($i < n$)). *At each node at maturity, the price function* $v_{n,j} : \left[A_{n,j}^{\min}, A_{n,j}^{\max}\right] \to \left[(A_{n,j}^{\min} - K)_+, (A_{n,j}^{\max} - K)_+\right]$ *defined as* $v_{n,j}(A) = (A - K)_+$ *is continuous, piecewise-linear and convex.*

PROOF. The singular points satisfy the conditions **??**. So for each $A \in \left[A_{n,j}^{\min}, A_{n,j}^{\max} \right]$, the price function $v_{n,j}(A)$ characterised by the singular points is continuous, piecewise–linear and convex by remark **??**. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Before maturity ($i < n$)

LEMMA 3.4 (Price function at any node). *At any node* $N_{i,j}$, *the price function* $v_{i,j}$: $\left[A_{i,j}^{\min}, A_{i,j}^{\max} \right] \to [0, \infty)$ *is continuous, piecewise-linear and convex.*

PROOF. We shall prove this using backward induction, the base case at maturity being true by virtue of Lemma **??**. We now consider step $i = n - 1$. Let A_u and A_d respectively represent the averages after an up and down movement corresponding to an average A. From equation **??**, we get

$$(3.15a) \qquad\qquad A_u = \frac{(i+1)A + S_0 u^{-i+2j+1}}{i+2}$$

$$(3.15b) \qquad\qquad A_d = \frac{(i+1)A + S_0 u^{-i+2j-1}}{i+2}$$

Since the options is of the European type, applying the no–arbitrage condition, the price function $v_{i,j} : \left[A_{i,j}^{\min}, A_{i,j}^{\max} \right] \to [0, \infty)$ is obtained by considering the discounted expectation value.

$$(3.16) \qquad\qquad v_{i,j}(A) = \frac{1}{R} \left[p v_{i+1,j+1}(A_u) + (1-p) v_{i+1,j}(A_d) \right]$$

From equation **??**, we get that A_u and A_d are linear functions of A. Thus, $v_{i+1,j+1}(A_u) = v_{n,j+1}(A_u)$ and $v_{i+1,j}(A_d) = v_{n,j}(A_d)$ are piecewise–linear convex continuous functions of A_u and A_d respectively. Thus, $v_{i+1,j+1}$ and $v_{i+1,j}$ may be seen as a linear combination of the above functions, and is thus piecewise–linear, convex and continuous itself. Again, from equation **??**, we get that $v_{i,j}$ is a convex combination of such functions, and the proof is complete.

We showed that if at time $i + 1$, if the price function is continuous, piecewise–linear and convex, so is it for time i. Since this is true for $i + 1 = n$, the same logic applied iteratively proves that the functions retain the characteristics for all $i \in [n]$. $\qquad\qquad\square$

REMARK 3.4. From Lemma **??**, we see that the price function may be characterised by singular points.

3.4.1   Evaluation of singular points

The evaluation of singular points for any node $N_{i,j}$ is done by the following algorithm, which works in a backward fashion in time, starting from the maturity.

We note that for the only influencing nodes for the node $N_{i,j}$ are $N_{i+1,j+1}$ and $N_{i+1,j}$. Thus we need to calculate the price of the option for each singular average belonging to either of these nodes.

Up movement   First we take each singular average $A_{i+1,j}^l$ belonging to $N_{i+1,j}$ and project it to $N_{i,j}$ via the following relation.

$$(3.17) \qquad B^l = \frac{(i+2)A_{i+1,j}^l - S_0 u^{-i+2j-1}}{i+1}$$

Thus, B^l is that average which after a down movement of the asset gives us the average $A_{i+1,j}^l$.

Next, we note that B^l is an increasing function of l, since a higher average at time step i would yield a higher average at time $i+1$. This in turn implies the following:

$$B^{L_{i+1,j}} = A_{i+i,j}^{\max} \ \forall j$$

$$B^1 \notin \left[A_{i+i,j}^{\min}, A_{i+i,j}^{\max}\right] \ \forall j \in \{1,\dots,i-1\}$$

Each $B^l \in \left[A_{i,j}^{\min}, A_{i,j}^{\max}\right]$ becomes the singular average of $N_{i,j}$.

In this way, we have determined the first coordinate of the singular points. We need to determine the second coordinate, or the prices $v_{i,j}(B^l)$, $\forall B^l \in \left[A_{i,j}^{\min}, A_{i,j}^{\max}\right]$, in order to determine the singular points completely. The idea is to calculate the discounted expected value of the price corresponding to each average B^l at $N_{i,j}$. In order to be able to do this, we need the prices corresponding to the average projected to the node $N_{i+1,j+1}$.

We consider an up movement of the underlying asset from node $N_{i,j}$. In this case, B^l transforms into the average $B_u^l = \left((i+1)B^l + S_0 u^{-i+2j+1}\right)/(i+2)$. Clearly, this average cannot belong to the set of averages associated with the node $N_{i+1,j+1}$. Thus, we need to find the index s such that $B_u^l \in \left[A_{i+1,j+1}^s, A_{i+1,j+1}^{s+1}\right]$. In the intervals the price function is linear, and thus we have

$$(3.18) \qquad v_{i+1,j+1}\left(B_u^l\right) = \frac{P_{i+1,j+1}^{s+1} - P_{i+1,j+1}^s}{A_{i+1,j+1}^{s+1} - A_{i+1,j+1}^s}\left(B_u^l - A_{i+1,j+1}^s\right) + P_{i+1,j+1}^s.$$

We follow this up by calculating the price associated with the singular value B^l by evaluating the discounted expectation value.

$$(3.19) \qquad v_{i,j}(B^l) = \frac{1}{R}\left[p v_{i+1,j+1}\left(B_u^l\right) + (1-p)v_{i+1,j}\left(A_{i+1,j}^l\right)\right].$$
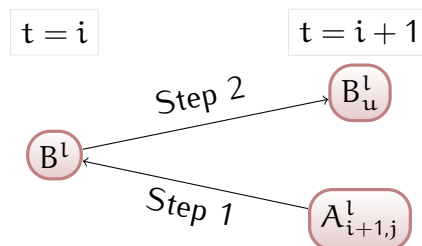
Figure ?? depicts the idea.

FIGURE 3.4. Up movement

Down movement   We now proceed to formulate the theory for the downward movement in the exact same fashion. Define the new average C^l at the node $N_{i,j}$ via the relation

$$(3.20) \qquad C^l = \frac{(i+2)A^l_{i+1,j+1} - S_0 u^{-i+2j+1}}{i+1}$$

Again, we note that

$$C^1 = A^{min}_{i,j} \; \forall j$$

$$C^{L_{i+1,j+1}} \in \left[A^{min}_{i,j}, A^{max}_{i,j}\right] \; \forall j \in \{1,\dots,i-1\}$$

$$C^l_d = \left((i+1)C^l + S_0 u^{-i+2j-1}\right)/(i+2)$$

Each $C^l \in \left[A^{min}_{i,j}, A^{max}_{i,j}\right]$ becomes the singular average of $N_{i,j}$.

For $v_{i,j}(C^l)$, $\forall C^l \in \left[A^{min}_{i,j}, A^{max}_{i,j}\right]$, we now have the following.

$$(3.21) \qquad v_{i+1,j+1}\left(C^l_d\right) = \frac{P^{s+1}_{i+1,j} - P^s_{i+1,j}}{A^{s+1}_{i+1,j} - A^s_{i+1,j}}\left(C^l_d - A^s_{i+1,j}\right) + P^s_{i+1,j}$$

$$(3.22) \qquad v_{i,j}(C^l) = \frac{1}{R}\left[pv_{i+1,j+1}\left(A^l_{i+1,j+1}\right) + (1-p)v_{i+1,j}\left(C^l_d\right)\right]$$

Figure ?? depicts the idea.
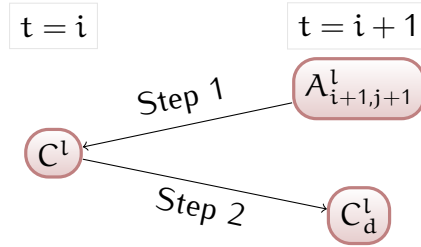
FIGURE 3.5. Down movement

Aggregation   Now we have the singular points for both up and down movements. We sort these points in ascending order of the first coordinate, i.e. the averages B^l and C^l that belong to $\left[A^{min}_{i,j}, A^{max}_{i,j}\right]$. These is an exhaustive list of all the singular points in the node (by construction). We note that $L_{i,j} \leqslant L_{i+1,j} + L_{i+1,j+1} - 2$.

This procedure is applied to all nodes, starting from maturity and proceeding backwards. At the 'edge' nodes $N_{i,0}$ and $N_{i,i}$, there is only one singular point whose price is given as follows

$$(3.23a) \qquad P^1_{i,0} = \frac{1}{R}\left[pP^1_{i+1,0} + (1-p)P^1_{i+1,1}\right]$$

$$(3.23b) \qquad P^1_{i,i} = \frac{1}{R}\left[pP^1_{i+1,i+1} + (1-p)P^{L_{i+1,i}}_{i+1,i}\right]$$

Thus we have a complete description of the price function at each node of the binomial tree. The price $P_{0,0}^1$ is exactly the binomial price relative to the tree with n steps of a fixed-strike European call option.

3.5   Fixed-strike Asian options of the American type

We now consider the American case. At maturity we have the same situation as in the European case. The price function is $v_{n,j}(A) = (A - K)_+$ for $A \in [A_{n,j}^{\min}, A_{n,j}^{\max}]$, and it is characterized by the same singular points.

Consider the step $i = n - 1$. At the node $N_{i,j}$, we first compute, by using the procedure described in the previous section (European case), the singular points associated with this node, thus obtaining the continuation value function $v_{n,j}^c(A)$ (note that this is nothing but $v_{n,j}(A)$ in the European case). But now, we must also account for exercise rights at this time. Thus, we have

(3.24) $$v_{n,j}(A) = \max\{\underbrace{A - K}_{\text{exercise}}, \underbrace{v_{n,j}^c(A)}_{\text{hold}}\}.$$

Note that $v_{i,j}(A)$ is still a piecewise-linear convex function, since maximum is a convex function, and the composition of two convex function is convex. For this reason, we can again characterize it by its singular points. In order to compute the singular points associated with the American case, we first remark that the slopes characterizing the piecewise-linear convex function $v_{i,j}^c(A)$ are all smaller than 1.

PROPOSITION 3.2 (boundedness of slope). *The slopes characterizing the piecewise-linear convex function $v_{i,j}^c(A)$ are all smaller than* 1, $\forall A \in \left(A_{i,j}^l, A_{i,j}^{l+1} \right) \ \forall l \in \{1, 2, \ldots, L_{i,j}\}$.

Of course, at the singular points, due to piecewise linearity, the function is not differentiable and thus admits no slope.

PROOF. First, note that differentiating Equations ?? with respect to A, we get

$$\frac{dA_u}{dA} = \frac{dA_d}{dA} = \frac{i+1}{1+2} < 1.$$

Now, we differentiate $v_{i,j}^c$ with respect to A in Equation ?? to get

$$\frac{dv_{i,j}^c}{dA} = \frac{1}{R} \left(p \frac{dv_{i+1,j+1}}{dA} + (1-p) \frac{dv_{i+1,j}}{dA} \right)$$

$$= \frac{1}{R} \left(p \frac{dv_{i+1,j+1}}{dA_u} \frac{dA_u}{dA} + (1-p) \frac{dv_{i+1,j}}{dA_d} \frac{dA_d}{dA} \right)$$

$$= \frac{1}{R} \frac{i+1}{1+2} \left(p \frac{dv_{i+1,j+1}}{dA_u} + (1-p) \frac{dv_{i+1,j}}{dA_d} \right)$$

Now, let $i = n - 1$. Then $v_{i+1,\cdot} = v_{n,\cdot}$. Clearly, from Section ?? (the price function at maturity), in each of the three cases, the price function can have a slope of either 0 or 1. In the above equation, considering $r > 0$, we have $R > 1$, and consequently, $R^{-1} < 1$.

Also, $p, 1 - p \in [0, 1]$. Putting all the inequalities in the last expression of $\frac{dv_{i,j}^c}{dA}$, we obtain $\frac{dv_{n-1,j}^c}{dA} < 1$.

Clearly, the slope in any previous time $i < n - 1$ can never be equal to unity by the exact same logic. The proof is complete by backward induction.   □

Hence there are two possible cases, as follows.

(1) $A_{i,j}^{\max} - K \leqslant v_{i,j}^c(A_{i,j}^{\max})$. This implies that $v_{i,j} \quad v_{i,j}^c$. The singular points remain the same.

(2) $A_{i,j}^{\max} - K < v_{i,j}^c(A_{i,j}^{\max})$. This case has two subcases, as follows.

(a) $A_{i,j}^{\min} - K \geqslant v_{i,j}^c(A_{i,j}^{\min})$. This implies that $v_{i,j}(A) = (A - K)$, $\forall A \in [A_{i,j}^{\min}, A_{i,j}^{\max}]$, so the only singular points points are $\left(A_{i,j}^{\min}, A_{i,j}^{\min} - K\right)$ and $\left(A_{i,j}^{\max}, A_{i,j}^{\max} - K\right)$.

(b) $A_{i,j}^{\min} - K < v_{i,j}^c(A_{i,j}^{\min})$. This means that there is a unique average $\bar{A}$ at which point the continuation value equals the early exercise. Let j_0 be the largest index such that $A_{i,j}^{j_0} < \bar{A}$. The new set of singular points becomes (see Figure ?? for a graphical representation)

$$\left\{ \left(A_{i,j}^1, P_{i,j}^1\right), \left(A_{i,j}^{j_0}, P_{i,j}^{j_0}\right), \left(\bar{A}, \bar{A} - K\right), \left(A_{i,j}^{\max}, A_{i,j}^{\max} - K\right) \right\}$$

We can repeat the same procedure iteratively for $i = n - 2, \ldots, 0$. At the final step, we get only one singular point, which gives us $P_{0,0}^1$, the exact American binomial price relative to the tree with n steps.

REMARK 3.5 (computational complexity – European vs. American). If we use approximations (as explained in the next section), the number of singular points in the American case can never be greater than the European case (even though right now we cannot determine the theoretical complexity of even the European case). Thus, the American procedure cannot be slower than the European one. In case we also take approximations into account, the American case is expected to be faster heuristically, but cannot be proved in general. Figure ?? illustrates this. A_{bar} is inserted, and all points except for the last, is removed. Thus, if the number of points after A_{bar} is huge, this would remove all those points. This is not possible in the European case.

REMARK 3.6 (put). For an Asian put, the exact same procedure has to be followed.

REMARK 3.7 (floating). In this case, we modify the procedure as follows: at maturity the singular points depend not on the strike K but rather on the underlying value at each node $S_{i,j}$. Therefore the new singular points are obtained by replacing K by $S_{i,j}$. The backward procedure is the same as before, just properly taking into account the new intrinsic values.

REMARK 3.8 (lookback options). Lookback options can also be similarly priced. In fact, in this case the algorithm admits several simplifications. We shall not study lookback options in this thesis. The interested reader should refer to [**Gaudenzi2010**].
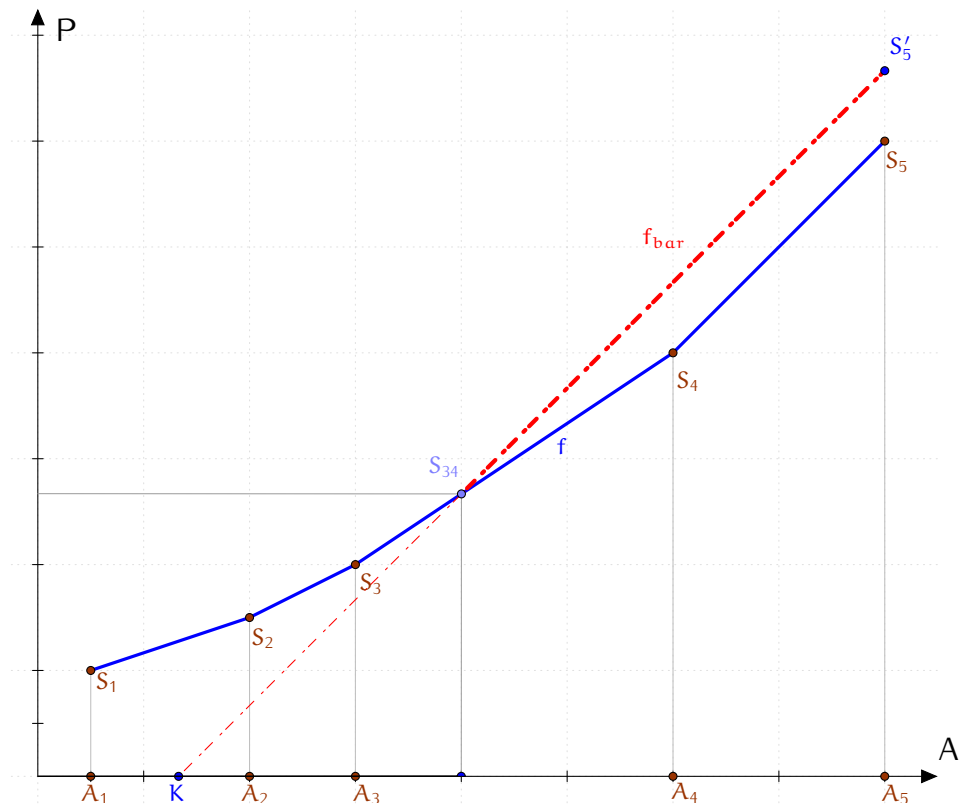
point $\left(A_{i,j}^l, P_{i,j}^l\right)$ from C. We have

(3.25) $\left| v_{i,j}^u(A) - v_{i,j}(A) \right| \leqslant \epsilon_l \qquad 8A \ 2 \left[A_{i,j}^{min}, A_{i,j}^{max} \right],$

where

(3.26) $\epsilon_l = v_{i,j}^u\left(A_{i,j}^l\right) - v_{i,j}\left(A_{i,j}^l\right) = \dfrac{P_{i,j}^{l+1} - P_{i,j}^{l-1}}{A_{i,j}^{l+1} - A_{i,j}^{l-1}}\left(A_{i,j}^l - A_{i,j}^{l-1}\right) + \left(P_{i,j}^{l-1} - P_{i,j}^l\right).$

Therefore, given any tolerance $h > 0$, we may remove a point $\left(A_{i,j}^l, P_{i,j}^l\right)$ only if $\epsilon_l < h$. Repeating this procedure sequentially at each node of the tree, while avoiding the elimination of two consecutive singular points, we can conclude that the upper estimate thus obtained differs from the exact binomial value by at most nh.

3.6.2   Lower bound

The computational procedure is quite similar to the procedure of approximation using upper bounds. The theoretical foundation of this part is given by Lemma ??. If we remove the points $\left(A_{i,j}^{l-1}, P_{i,j}^{l-1}\right)$ and $\left(A_{i,j}^l, P_{i,j}^l\right)$, and add the point $(\bar{x}, \bar{y})$ as described in Lemma ??, the new function is never greater than the original one, and the differences between the values at any point is bounded over by δ_l, where

(3.27) $\delta_l = \dfrac{P_{i,j}^l - P_{i,j}^{l-1}}{A_{i,j}^l - A_{i,j}^{l-1}}\left(\bar{x} - A_{i,j}^{l-1}\right) + \left(P_{i,j}^{l-1} - \bar{y}\right).$

Again, we allow for the operation only if $\delta_l < h$.

An implementation methodology is as follows. At each node N_{ij}, we consider the starting four points $\left\{ \left(A_{i,j}^1, P_{i,j}^1\right), \left(A_{i,j}^2, P_{i,j}^2\right), \left(A_{i,j}^3, P_{i,j}^3\right), \left(A_{i,j}^4, P_{i,j}^4\right) \right\}$. Calculate δ_3. If $\delta_3 < h$, we add the point $(\bar{x}, \bar{y})$ and remove the points corresponding to $\left(A_{i,j}^2, P_{i,j}^2\right)$ and $\left(A_{i,j}^3, P_{i,j}^3\right)$. The procedure will continue by considering the four new points $\left\{ (\bar{x}, \bar{y}), \left(A_{i,j}^4, P_{i,j}^4\right), \left(A_{i,j}^5, P_{i,j}^5\right), \left(A_{i,j}^6, P_{i,j}^6\right) \right\}$ and proceeding in the exact same way. On the other hand, if $\delta_3 \geqslant h$, then we do not remove any points and the procedure will continue by considering the four new points $\left\{ \left(A_{i,j}^2, P_{i,j}^2\right), \left(A_{i,j}^3, P_{i,j}^3\right), \left(A_{i,j}^4, P_{i,j}^4\right), \left(A_{i,j}^5, P_{i,j}^5\right) \right\}$. The procedure is repeated until all the singular points of the node has been taken into account.

REMARK 3.9 (Convergence to the exact binomial value). Jiang and Dai (2005) [**Jiang2004**] proved the convergence of the exact binomial algorithm for European/American path-dependent options. In particular, they proved that the rate of convergence of the exact binomial algorithm to the continuous value is $O(\Delta T)$.

The possibility of obtaining estimates of the exact binomial price with an error control allows us to prove easily the convergence of our method to the continuous value. By choosing h to depend on n so that $nh(n) \ ! \ 0$, we have that the corresponding sequences

of upper and lower estimates converge to the continuous price value. Moreover, the choice $h(n) = O(\frac{1}{n^2})$ guarantees that the order of convergence is $O(\Delta T)$.

REMARK 3.10 (Computational complexity – theoretical considerations). The key issue in assessing the theoretical complexity of the algorithm lies in the upper and lower bound computation. Since the number of singular points eliminated depends on various factors, and we do not control the number of singular points directly, it is difficult to theoretically calculate the order of complexity of the algorithm. The dependence of number of singular points and their nature (closeness, convexity) on the initial data might be a fruitful area of research, and this in turn might give us clues as to how the problem of finding complexity theoretically might be tackled. Nevertheless, the numerical results indicate that the method is quite competitive in practice.

3.7 The program

3.7.1 Algorithm

Algorithm 3.1: Pricing cliquet options using the singular points method

---

Input:

    Contract details

        time to maturity: T, strike price: K

        type: arithmetic/geometric, call/put, European/American, fixed/floating

    Details of the underlying asset

        initial price: s_0, volatility: σ, continuous dividend rate: q

    Market parameters – spot interest rate: r

    Computational parameters – time steps: n, error bound: h

Output: The price of the option at the initial time

1   **begin**

2   Set $\Delta T, u, p$ from the formulae in Section ??.

3   Compute the singular points at maturity using Equation ?? and ??.

4   **for** $i \in \{N-1, \ldots, 0\}$ **do**

5   Evaluate $P_{i,0}^1$ and $P_{i,i}^1$ by Equation ?? with the early exercise.

6   **forall the** $N_{i,j}$, $j \in \{1, \ldots, i-1\}$ **do**

7   Using Equation ??, $\forall A_{i+1,j}^l, \forall l \in \{1, \ldots, L_{i+1,j}\}$, compute B^l.

8              $\forall B^l \in \left[A_{i,j}^{\min}, A_{i,j}^{\max}\right]$, compute $v_{i,j}^c(B^l)$ by Equations ?? and ??.

9   Using Equation ??, $\forall A_{i+1,j+1}^l, \forall l \in \{1, \ldots, L_{i+1,j+1}\}$, compute C^l.

10             $\forall C^l \in \left[A_{i,j}^{\min}, A_{i,j}^{\max}\right]$, compute $v_{i,j}^c(C^l)$ by Equations ?? and ??.

11   Sort the averages $\{B^l\}_l \cup \{C^l\}_l \in \left[A_{i,j}^{\min}, A_{i,j}^{\max}\right]$ to obtain the set of $L_{i,j}$ singular points.

12   Compute the American price according to Section ?? and obtain a new set of singular points with a new cardinality denoted, for simplicity, by $L_{i,j}$ again.

13   **switch** *upper or lower* **do** Approximation

14   **case** *upper bound*

15   Follow Section ??

16   **case** *lower bound*

17   Follow Section ??

18   **return** $P_{0,0}^1$ // the upper or lower estimate of the exact binomial price with error smaller that nh.

3.7.2 Implementation

The algorithm was implemented in Python 3.5.0 (2015-09-13).

```python
# ================================================================= #
__author__ = "Sudip Sinha"

from math import exp, sqrt, isclose



# ----------------------------------------------------------------- #

# @profile
def asian_call_sp( r: float,  # Market
                   s0: float, sigma: float, q: float,  # Underlying
                   k: float, t: float, am: bool = True,  # Derivative
                   n: int = 25, h: float = 0., ub: bool = True
                   # Computation
                   ) -> list:
  """Prices of an Asian call option using the singular point method"""

  # Time period
  dt = t / n
  # Effective interest rate
  R = exp( (r - q) * dt )

  # Up and down factors
  u = exp( sigma * sqrt( dt ) )
  d = 1. / u

  # Risk neutral probability, discounted by R
  p_u = (R * u - 1) / (u * u - 1) / R
  p_d = 1. / R - p_u
  assert p_u >= 0. and R * p_u <= 1., "Probability!"

  # Singular points for nodes N(i+1,*)

  # Start from maturity
  now = [[]] * (n + 1)

  # Singular points for N(n,0)
  a = s0 / (n + 1) * (1 - d ** (n + 1)) / (1 - d)
  now[0] = [(a, max( a - k, 0. ))]

  # Singular points for N(n,n)
  a = s0 / (n + 1) * (1 - u ** (n + 1)) / (1 - u)
```

```python
now[-1] = [(a, max( a - k, 0. ))]

# Singular points for N(n,j)
for j in range( 1, n ):
  a_min = s0 / (n + 1) * ((1 - d ** (n - j + 1)) / (1 - d)
                          + d ** (n - j - 1) * (1 - u ** j) /
                          (1 - u))
  a_max = s0 / (n + 1) * ((1 - u ** (j + 1)) / (1 - u)
                          + u ** (j - 1) * (1 - d ** (n - j)) /
                          (1 - d))
  if k < a_min:
    now[j] = [(a_min, a_min - k), (a_max, a_max - k)]
  elif k > a_max:
    now[j] = [(a_min, 0), (a_max, 0)]
  else:
    now[j] = [(a_min, 0), (k, 0), (a_max, a_max - k)]


# Singular points for N(i,j) i != n
for i in range( n - 1, -1, -1 ):
  nxt = now
  now = [[]] * (i + 1)

  for j in range( i + 1 ):

    a_min = s0 / (i + 1) * ((1 - d ** (i - j + 1)) / (1 - d) +
                            d ** (i - j - 1) * (1 - u ** j) /
                            (1 - u))

    a_max = s0 / (i + 1) * ((1 - u ** (j + 1)) / (1 - u) +
                            u ** (j - 1) *
                            (1 - d ** (i - j)) / (1 - d))

    if a_min > a_max:
      (a_min, a_max) = (a_max, a_min)  # Jugaad

    # 'B'
    sp_b = list( )  # Initialize sp_b
    for (a, v_a) in nxt[j]:
      b = ((i + 2) * a - s0 * u ** (-i + 2 * j - 1)) / (i + 1)

      assert len( nxt[j + 1] ) > 0, "len( nxt[j+1] ) == 0"

      # Get b_up for b in [a_min, a_max]
      if a_min <= b <= a_max \
```

```python
                or isclose( b, a_min ) \
                or isclose( b, a_max ):
            b_up = ((
                    (i + 1) * b + s0 * u ** (-i + 2 * j + 1)) /
                    (i + 2))
            nxt_nd = nxt[j + 1]

            v_b_up = 0
            if isclose( b_up, nxt_nd[-1][0] ):
                v_b_up = nxt_nd[-1][1]
            elif isclose( b_up, nxt_nd[0][0] ):
                v_b_up = nxt_nd[0][1]
            else:
                for kb in range( len( nxt_nd ) - 1 ):
                    if nxt_nd[kb][0] <= b_up < nxt_nd[kb + 1][0]:
                        v_b_up = (
                        (nxt_nd[kb + 1][1] - nxt_nd[kb][1]) /
                        (nxt_nd[kb + 1][0] - nxt_nd[kb][0]) *
                        (b_up - nxt_nd[kb][0]) + nxt_nd[kb][1])
                        break
            sp_b.append( (b, p_u * v_b_up + p_d * v_a) )

    # 'C'
    sp_c = list( )  # Initialize sp_c
    for (a, v_a) in nxt[j + 1]:
        c = ((i + 2) * a - s0 * u ** (-i + 2 * j + 1)) / (i + 1)

        assert len( nxt[j] ) > 0, "len( nxt[j] ) == 0"

        # Uniqueness check: Check if 'C' is not in the set of 'B'.
        # Not required since we eliminate close points later.

        # Get c_dn for c in [a_min, a_max]
        if a_min <= c <= a_max \
                or isclose( c, a_min ) \
                or isclose( c, a_max ):
            c_dn = ((
                    (i + 1) * c + s0 * u ** (-i + 2 * j - 1)) /
                    (i + 2))
            nxt_nd = nxt[j]

            v_c_dn = 0
            if isclose( c_dn, nxt_nd[-1][0] ):
                v_c_dn = nxt_nd[-1][1]
            elif isclose( c_dn, nxt_nd[0][0] ):
```

```
          v_c_dn = nxt_nd[0][1]
        else:
          for kc in range( len( nxt_nd ) - 1 ):
            if nxt_nd[kc][0] <= c_dn < nxt_nd[kc + 1][0]:
              v_c_dn = (
              (nxt_nd[kc + 1][1] - nxt_nd[kc][1]) /
              (nxt_nd[kc + 1][0] - nxt_nd[kc][0]) *
              (c_dn - nxt_nd[kc][0]) + nxt_nd[kc][1])
              break
        sp_c.append( (c, p_u * v_a + p_d * v_c_dn) )

    # Aggregate 'B's and 'C's.
    now_nd = sorted( sp_b + sp_c, key = lambda sp_l:sp_l[0] )

    # Remove singular points very close to each other.
    l = 0
    while l < len( now_nd ) - 1:
      if isclose( now_nd[l + 1][0], now_nd[l][0] ):
        if (now_nd[l][0] - s0) == min(now_nd[l][0] - s0,
                                      now_nd[l + 1][0] - s0):
          del now_nd[l + 1]
        else:
          del now_nd[l]
      l += 1

    # American
    if am:
      if a_max - k <= now_nd[-1][1]:
        pass   # Same as the European case
      elif now_nd[0][1] <= a_min - k:
        now_nd = [(a_min, a_min - k), (a_max, a_max - k)]
      else:
        for l in range( len( now_nd ) - 1 ):
          if isclose( now_nd[l][0], now_nd[l][1] ):
            now_nd = now_nd[0:l + 1]
            now_nd.append( (a_max, a_max - k) )
            break
          if (now_nd[l][0] - k <= now_nd[l][1]) \
              and (now_nd[l + 1][1] <=
                        now_nd[l + 1][0] - k):
            # Find the point of intersection
            # x = ((a2 - a1) * k - (a2 * v1 - a1 * v2)) /
            #                     ((A2 - v2) - (A1 - v1))
            a_int = (
              ((now_nd[l + 1][0] - now_nd[l][0]) * k +
```

```
            (now_nd[l + 1][0] * now_nd[l][1] -
             now_nd[l][0] * now_nd[l + 1][1])) /
            ((now_nd[l + 1][0] - now_nd[l][0]) -
             (now_nd[l + 1][1] - now_nd[l][1])))
        now_nd = now_nd[0:l + 1]
        now_nd.extend(
          [(a_int, a_int - k),
           (a_max, a_max - k)] )
        break

# Approximations
if h > 0.:  # and (i < n - 1):
  if ub:  # Lemma 1
    l = 1
    last_point_removed = False
    while l < len( now_nd ) - 1:
      if last_point_removed:
        last_point_removed = False
        l += 1
        continue
      eps = ((now_nd[l + 1][1] - now_nd[l - 1][1]) /
             (now_nd[l + 1][0] - now_nd[l - 1][0]) *
             (now_nd[l][0] - now_nd[l - 1][0]) +
             (now_nd[l - 1][1] - now_nd[l][1]))
      if eps < h:  # Remove this point
        del now_nd[l]
        last_point_removed = True
      else:
        l += 1

  if not ub:  # Lemma 2
    l = 1
    while l < len( now_nd ) - 2:
      m1 = ((now_nd[l][1] - now_nd[l - 1][1]) /
            (now_nd[l][0] - now_nd[l - 1][0]))
      m2 = ((now_nd[l + 2][1] - now_nd[l + 1][1]) /
            (now_nd[l + 2][0] - now_nd[l + 1][0]))
      x_bar = (((m2 * now_nd[l + 1][0] -
                 m1 * now_nd[l - 1][0]) -
                (now_nd[l + 1][1] - now_nd[l - 1][
                   1])) /
               (m2 - m1))
      y_bar = (m1 * (x_bar - now_nd[l - 1][0]) +
               now_nd[l - 1][1])
      dlt = ((now_nd[l + 1][1] - now_nd[l][1]) /
```

```
                   (now_nd[l + 1][0] - now_nd[l][0]) *
                   (x_bar - now_nd[l][0]) +
                   now_nd[l][1] - y_bar)
             if dlt < h:   # Remove l-1 and l and add x_bar
               now_nd[l] = (x_bar, y_bar)
               del now_nd[l + 1]
             l += 1

       now[j] = now_nd

  return now[0][0][1]


# ===================================================================== #
```

3.8   Extensibility

Let us recapitulate the conditions required for the singular points method to work in the case of Asian options with arithmetic mean.

> The ability to calculate the upper and lower bounds of the mean for all nodes of the tree.

> The recombinant nature of the tree for the underlying. Note that the tree for the option prices are *not* recombinant.

> Convexity and piecewise-linearity of the price function on the mean of the underlying.

> Constant volatility

Keeping these in mind, let us look at the possibility of extending the singular points method to the following cases:

(1) Asian options with geometric mean and fixed volatility.

(2) Asian options with arithmetic mean and local volatility.

Geometric mean and fixed volatility   In the case of geometric options, we have a closed form formula under the Black–Scholes market model. Let us try to extend the singular points method to this case.

Firstly, we show that the result about the maximum and minimum paths still hold in the geometric case.

DEFINITION 3.5 (Geometric mean). The geometric mean of the risky asset's prices $(S_i)_{i \in [n]}$ is given by:

$$(3.28) \qquad G_n = \left(\prod_{i=0}^{n} S_i \right)^{\frac{1}{n+1}}$$

LEMMA 3.5. *At each node* $N(i,j)$, *the following hold:*

(1) *The maximum average possible* $G_{i,j}^{\max}$ *is attained by the path corresponding to the path with* j *up movements followed by* $(i-j)$ *down movements.*

(2) *The minimum average possible* $G_{i,j}^{\min}$ *is attained by the path corresponding to the path corresponding to the path with* $(i-j)$ *down movements followed by* j *up movements.*

PROOF. The proof is the same as Corollary **??**, with A replaced by G and relevant modifications.   □

One of the central ideas behind the singular points method is that the price of the option is a convex, piecewise-linear function of the average A. But in the geometric case, this no

longer holds true. For example, take a node $N_{i,j}$ with $i = n - 1$. The price function given by $v_{i,j}(G)$, with $G \in [G^{min}, G^{max}]$, can be calculated by the discounted expectation value.

$$(3.29) \qquad v_{i,j}(G) = \frac{1}{R} \left[p v_{i+1,j+1}(G_u) + (1-p) v_{i+1,j}(G_d) \right]$$

$$(3.30) \qquad G_u = \left(G^{i+1} S_0 u^{-i+2j+1} \right)^{\frac{1}{i+2}} \Big/ G^{\frac{i+1}{i+2}}$$

$$(3.31) \qquad G_d = \left(G^{i+1} S_0 u^{-i+2j-1} \right)^{\frac{1}{i+2}} \Big/ G^{\frac{i+1}{i+2}}$$

Clearly, the final function $v_{i,j}$ is not linear in G. Rather it is piecewise-concave. Thus we cannot use the singular points method in this case.

Arithmetic mean with local volatility   In this case, the tree for the underlying is not recombinant, so we do not have more than one singular point in one (non-recombining) node. Essentially, we cannot use the singular points method for local volatility models.

3.9   Results and conclusion

We assume that the initial value of the stock price is $s_0 = 100$, the maturity is $T = 1$, the force of interest rate is $r = 0.1$ and the continuous dividend yield is $q = 0.03$. We will consider two choices for volatility, $\sigma = 0.2$ and $\sigma = 0.4$, and two choices for the strike, $K = 90$ and $K = 110$. The time steps (n) taken into consideration are 10, 25, 50, 100, 200 and 400.

All simulations were run on a computer with the following specifications.

TABLE 3.3. Computer specifications

Item	Details
Processor	Intel R Celeron R N2840 @ 2.16GHz (2 CPUs)
Architecture	64-bit
Memory	4 GB
Operating system	Arch Linux
Python	v3.5.0 (2015-09-13)

Table ?? highlights the prices obtained using the binomial method ($n = 10$ and $n = 25$), and the singular points method for the aforementioned time steps.

By comparing our results with the exact binomial value obtained for $n = 10$, we can confirm that the method gives correct results. In fact, from Table ?? and also [**Gaudenzi2010**], we observe that the prices are monotonically increasing with respect to the number of time steps, and seem to gradually converge towards a limiting value.

TABLE 3.4. Results for Asian options

| | n | K = 90 | | K = 110 | |
		$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.2$	$\sigma = 0.4$
Bin	10	14.5912	17.8033	2.5100	6.6523
	25	15.1535	18.6786	2.6270	7.3451
SP	10	14.5925	17.8068	2.5090	6.6511
	25	15.1535	18.6785	2.6270	7.3449
	50	15.3524	19.0420	2.6673	7.4563
	100	15.4732	19.2696	2.6886	7.5174
	200	15.5453	19.4065	2.6996	7.5502
	400	15.5861	19.4845	2.7053	7.5674

Concluding remarks   Thus, we have seen that the singular points method fails to be generalised, its shortcomings being the inability to deal with Asian options with geometric mean and to deal with local volatility cases. Nevertheless, it is quite efficient in its domain. It is comparatively fast, having experimental order of complexity $O(n^3)$. It also allows specification of *a priori* error bounds, both upper and lower. Numerical results reported by [**Gaudenzi2010**] shows that the method outperforms alternative algorithms, and may be viewed as an improvement on previous tree methods.

Chapter 4

Cliquet options

A forward start option is an option that starts at a specified future date (called the *determination date*), with an expiration date set further in the future[1]. A forward start option starts at a specified date in the future; however, the premium is paid in advance, and the time of expiration is established at the time the forward start option is purchased. Since the asset price at the start of this option is not known a priori, it is common to specify that the strike price will be set in the future so that the option is initially at the money or a certain percentage in the money or out of the money.

The payoff of a forward start call option with time to expiration T and such that the strike is determined at the money at time u is given by $(S_T - S_u)_+$[2]

A *cliquet option* or a *ratchet option* is an exotic option consisting of a series of consecutive forward start options. The first such option is active immediately, and once it expires the second comes into existence, and so on. Each option is struck *at-the-money* when it becomes active. Therefore, such an option periodically settles and resets its strike price at the level of the underlying during the time of settlement. Investors can opt to receive their payout either when each option expires or wait until the entire time has elapsed.

Usually, the return on a cliquet option is capped and floored. The capping and flooring may be local or global (or both). The motivation behind bounding the return is to provide the investor safety against downside risks, yet allowing significant upside potential. Consequently, the investor is also constrained from having unbounded gains. Capping the maximum ensures that the payoff is never too extreme and therefore that the value of the contract is not too outrageous. Some variants of cliquet options are as follows:

Reverse cliquet: Amounts to a cash flow minus a capped cliquet of puts.

Digital cliquet: The forward–starting options are digital options.

The following example, copied verbatim from Investopedia[3], provides a good illustration.

---

[1] http://www.math.umn.edu/~spirn/5076/Lecture16.pdf (Page 5)
[2] http://www.stat.nus.edu.sg/~stalimtw/MFE5010/PDF/L2forward.pdf
[3] http://www.investopedia.com/terms/c/cliquet.asp

EXAMPLE 4.1. A three-year cliquet option with a strike of 1000 would expire worthless on the first year if the underlying was to be 900. This value (900) would then be the new strike price for the following year and should the underlying on the settlement be 1200, the contract holder would receive a payout and the strike would reset to this new level. Higher volatility provides better conditions for investors to earn profits.

Literature review   The literature for pricing of cliquet options is primarily based on partial differential equations (PDE) techniques. Notable mentions include Wilmott's finite difference (FD) approach in a non-linear uncertain volatility model (UVM) in 2002 [**Wilmott2002**]. Later in 2006, Windcliff et al.[**Windcliff2006**] explored a variety of modelling alternatives, including jump diffusion models, local volatility and UVM models, again using finite differences methods. They compared the use of a running sum of returns formulation to an average return formulation. Methods for grid construction, interpolation of jump conditions, and application of boundary conditions were also compared.

Lattice method   Gaudenzi *et al*[**Gaudenzi2011**] introduced a discrete method for pricing cliquet options in the Cox Ross Rubinstein model [**Cox1979**] introduced in Chapter **??**. This singular points method is faster than the alternative lattice based approaches available, while retaining a fair bit of flexibility to handle varying volatilities and rates of interest. Most of the chapter is inspired by this paper.

The central idea of the method, as we saw in Chapter **??**, is to give, at every monitoring date, a continuous representation of the cliquet option price as a piecewise-linear function of the sum of the returns over the corresponding period. This function is characterized only by a set of points, called *singular points*, which can easily be computed recursively by backward induction. Although the number of singular points grows rapidly at every monitoring date, it is possible to reduce their number drastically in a straightforward way, controlling the error involved by the elimination procedure at the same time. Moreover, the error control process leads automatically to the convergence of the approximations to the continuous value.

4.1   Cliquet contracts and models

Just as in the case of Asian options, we shall assume that the evolution of the prices of the risky asset $(S_t)_t$ is governed by the Black–Scholes stochastic differential equation as discussed in Equation **??** of Chapter **??**. Its solution is given by Equation **??**b. Whenever there is a continuous divident yield, we modify the equation according the remark **??**. Moreover, we shall consider the discrete model setup exactly as described in the beginning of Section **??** of Chapter **??**.

Let T be the maturity of the cliquet contract. Let the payoffs depend on the N preordained observation times $t_1, t_2, \ldots, t_N$ ($t_0 = 0$). At these observation times, the value of the underlying are $(S_i)_i, S_i = S_{t_i}, i \; 2 \; [N]$. The returns for the time interval $(t_{i-1}, t_i]$ are given

by

(4.1)
$$R_i = \frac{S_i - S_{i-1}}{S_{i-1}} = \frac{S_i}{S_{i-1}} - 1 \; .$$

During each time interval, the return is capped and floored locally by the quantities C_{loc} and F_{loc}. In other words, we consider the quantity $\max\{F_{loc}, \min\{C_{loc}, R_i\}\}$ rather than the return itself. The sum of these quantities till time t_i is called the 'running sum' and is given by

(4.2)
$$Z_i = \sum_{k=1}^{i} \max\{F_{loc}, \min\{C_{loc}, R_k\}\}.$$

We also consider a global cap C_{glob} and floor F_{glob}. Thus, the expression for the payoff finally becomes

(4.3)
$$\text{payoff} = \text{notional} \;\; \max\{F_{glob}, \min\{C_{glob}, Z_N\}\}.$$

For ease of notation, we take notional $= 1$.

We note that the case $C_{glob} > NC_{loc}$ is equivalent to $C_{glob} = NC_{loc}$. Similarly, the case $F_{glob} < NF_{loc}$ is equivalent to $F_{glob} = NF_{loc}$. In general, we may write the following.

$$F_{glob} = \max\{NF_{loc}, F_{glob}\}$$
$$C_{glob} = \min\{NC_{loc}, C_{glob}\}$$

We assume that the difference between two observation times is constant and we denote by m the number of steps of the binomial tree in every period (so that the total number of steps of the binomial tree is $n = mN$).

4.2   The singular points method for cliquet options

The binomial method may always be used to price any option, including path–dependent ones. The binomial method looks through all possible paths of the underlying in order to price the option. The number of possible paths are 2^{mN}. Thus, the method is inherently extremely computationally expensive due to the exponential dependence of the number of paths on m and N. The theoretical computational complexity is $O(m^N)$, as in [**Gaudenzi2011**].

A modification of the singular points method described in the previous chapter solves this problem for cliquet options by the process of approximation. The method of approximation selectively removes certain paths that would be normally considered, but which do not affect the result in a significant manner. This may be done by putting an *a priori* error bound while removing points. The method turns out to be significantly faster and memory efficient compared to known binomial techniques. Moreover, its flexible is evinced by the fact that it is adaptable for varying interest rate and volatility in each observational period.

The price function in the cliquet case is not necessarily convex due to the presence of a global cap (see Figure **??**), as opposed to the Asian case. Since the algorithm starts from maturity, it follows that we cannot assume convexity at any point of time. It was primarily

the convexity of the price functions in the Asian case that allowed us to obtain simple upper and lower bounds of the exact binomial price. Nevertheless, in the case of cliquet options, the singular points approach still provides an efficient binomial framework, even in the absence of convexity of the price functions, as we shall see.

We redefine singular points to exclude the constraint of convexity. The reader is advised to keep in mind the ideas introduced in Section ?? of Chapter ??.

DEFINITION 4.1 (singular points and singular values). Let $P = (P_i)_{i \in [n]} = ((x_i, y_i))_{i \in [n]}$, $n \in \mathbb{N}$ be a sequence of points such that $a = x_0 < x_1 < \quad < x_{n-1} < x_n = b \, \& \, i \in [n]$

Let $f : [a, b] \to [0, \infty)$ be the function obtained by linear interpolation of the points in P. The definition of f ensures that the function is continuous and piecewise-linear.

Then, the elements of P are called *singular points of* f and the abscissae $\{x_i\}_{i \in [n]}$ are called *singular values of* f.

REMARK 4.1 (characterisation). Note that the singular points characterise such a function completely, even without the requirement of convexity. This is clear from ??.

4.2.1   The method

Our aim is to look at every possible value taken by the running sum Z within the bounds $[F_{loc}, C_{loc}]$ for each time interval $t_1, \ldots, t_N$. If we know the price function at maturity, we may use a backward procedure (in time) in order to obtain a continuous representation of the cliquet price as a piecewise-linear function of the running sum Z. Since it is piecewise-linear and continuous, the function may be represented using its singular points. Thus, we see an evolution of singular points as we go back in time. Since the number of singular points may be significantly high for any computer, we shall introduce an error controlled approximation procedure to reduce the number of singular points.

Let the number of singular points at each observational time t_i be L_i, where $i \in \{1, \ldots, N\}$. For each singular point $l \in \{1, \ldots, L_i\}$, the abscissa is called the *singular running sum* Z_i^l and the ordinate is called the *singular price* P_i^l. Thus, the singular points are denoted by

$$(Z_i^l, P_i^l) \qquad \& \, l \in \{1, \ldots, L_i\}$$

At maturity   At maturity ($t_{Nobs} = T$), for every running sum Z, the price of the cliquet option $V_N(Z)$ as function of Z, is given by

(4.5) $$V_N(Z) = \max\{Fglob, \min\{Cglob, Z\}\}$$

We note the following.

(1) $V_N(Z)$ is a continuous and piecewise-linear function

(2) $V_N(Z)$ is defined in the interval $[NF_{loc}, NC_{loc}]$

(3) There are only four points where the function changes slope, namely NF_{loc}, F_{glob}, C_{glob} and NC_{loc}. This is because if $Z < F_{glob}$, the price is constant. Same can

be said about $Z > C_{glob}$. Thus the four numbers enumerated above form the complete set of singular running sums at maturity ($L_N = 4$).

Focusing further on point 3 above, we note down the singular running sums and corresponding prices.

TABLE 4.1. Singular points at maturity

l	Z_N^l	P_N^l
1	NF_{loc}	F_{glob}
2	F_{glob}	F_{glob}
3	C_{glob}	C_{glob}
4	NC_{loc}	C_{glob}

A more visual representation of the price function at maturity is given in Figure ??.
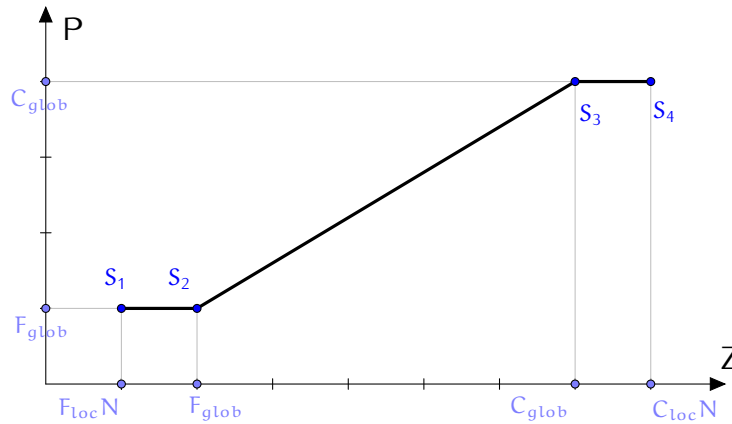
FIGURE 4.1. The price function at maturity

The penultimate time step   We consider the time step $N - 1$. If the running sum at time t_{N-1} is denoted by Z, then the corresponding price depends on the possible returns of the underlying asset during the time interval $[t_{N-1}, T]$.

We revisit equation ?? once more. Now we note that since the number of time steps in each interval is m, there will be m up movements or down movements of the asset. Thus, there are $m + 1$ possible outcomes, given by $S_i = u^{-m+2j}S_{i-1}, j \, \mathcal{2} \, [m]$. Corresponding to these cases, there are also $m + 1$ returns, given by

$$(4.6) \qquad\qquad R_j = u^{-m+2j} - 1 \qquad j \, \mathcal{2} \, [m]$$

Since the probability of an up movement in time ΔT is p, we have that the probability of each return is distributed binomially, and is given by

$$(4.7) \qquad\qquad p_j = \binom{m}{j} p^j (1-p)^{m-j}$$

The above derivations assume that there has been no local flooring or capping of the returns. In the case of such bounds, the actual possibilities are fewer in number, and we need to put bounds on j. We can do this in the following fashion.

In the case of a local floor, we must have

$$
\begin{aligned}
F_{loc} &\geqslant u^{-m+2j_{min}} - 1 \\
\Rightarrow \quad \log(F_{loc} + 1) &\geqslant (-m + 2j_{min}) \log(u) && \dots \text{(log is monotonic)} \\
\Rightarrow \quad \frac{\log(F_{loc} + 1)}{\log(u)} &\geqslant (-m + 2j_{min}) && \dots (u > 1 \Rightarrow \log(u) > 0) \\
\Rightarrow \quad j_{min} &\leqslant \frac{\log(F_{loc} + 1)}{2\sigma\Delta T} + \frac{m}{2} && \dots (u = e^{\sigma\Delta T}) \\
\Rightarrow \quad j_{min} &= \left\lfloor \frac{\log(F_{loc} + 1)}{2\sigma\Delta T} + \frac{m}{2} \right\rfloor && \dots (j \in [m])
\end{aligned}
$$

Where $\lfloor \cdot \rfloor$ denotes the floor function.

Similarly, in the case of local cap, we have

$$
\begin{aligned}
C_{loc} &\leqslant u^{-m+2j_{max}} - 1 \\
\Rightarrow \quad \log(C_{loc} + 1) &\leqslant (-m + 2j_{max}) \log(u) && \dots \text{(log is monotonic)} \\
\Rightarrow \quad \frac{\log(C_{loc} + 1)}{\log(u)} &\leqslant (-m + 2j_{max}) && \dots (u > 1 \Rightarrow \log(u) > 0) \\
\Rightarrow \quad j_{max} &\geqslant \frac{\log(C_{loc} + 1)}{2\sigma\Delta T} + \frac{m}{2} && \dots (u = e^{\sigma\Delta T}) \\
\Rightarrow \quad j_{max} &= \left\lceil \frac{\log(C_{loc} + 1)}{2\sigma\Delta T} + \frac{m}{2} \right\rceil && \dots (j \in [m])
\end{aligned}
$$

Where $\lceil \cdot \rceil$ denotes the ceiling function.

We represent by j_0 the number of possibilities of the return after enforcing the local floor and cap. Summarising

$$
\text{(4.8a)} \qquad\qquad j_{min} = \left\lfloor \frac{\log(F_{loc} + 1)}{2\sigma\Delta T} + \frac{m}{2} \right\rfloor
$$

$$
\text{(4.8b)} \qquad\qquad j_{max} = \left\lceil \frac{\log(C_{loc} + 1)}{2\sigma\Delta T} + \frac{m}{2} \right\rceil
$$

$$
\text{(4.8c)} \qquad\qquad j_0 = j_{max} - j_{min}
$$

$\forall j \leqslant j_{min}$, the return is F_{loc}, and $\forall j \geqslant j_{max}$, the return is C_{loc}. For the other indices, the return remains unchanged.

We shift the indices from $\{j_{min}, \dots, j_{max}\}$ to $\{0, \dots, j_0\}$ by putting $j' = j - j_{min}$. Table ?? highlights the shifted indices, and the corresponding returns and probabilities.

REMARK 4.2 (shifted indices). Note that the j's represents the indices of the possible paths that can *actually* be taken respecting the constraints of the local floor and cap, whereas j represents the indices of all possible paths without respect to any constraints. Thus, we

TABLE 4.2. Shifted returns and probabilities

Range(j)	j'	R'_j	p'_j
$j \leqslant j_{min}$	0	F_{loc}	$\sum_{k=0}^{j_{min}} p_k$
$\{j_{min}+1,\ldots,j_{max}-1\}$	$j-j_{min}$	$R_{j+j_{min}}$	$p_{j+j_{min}}$
$j \geqslant j_{max}$	j_0	C_{loc}	$\sum_{k=j_{max}}^{m} p_k$

are more interested in the shifted indices. Similarly, p'_j represents the probability of taking the path j', and R'_j represents the corresponding return.

Now we focus on how to determine the price function at time t_{N-1}. Recall that at maturity, the function $V_N(Z)$ giving the price of the cliquet option as a function of the running sum Z at maturity, is the piecewise linear function whose singular points are presented in Table ??. At the penultimate time step t_{N-1}, we must have $Z \in [(N-1)F_{loc}, (N-1)C_{loc}]$. Note that Z is the running sum till the penultimate time step. The price function at this time is given as a discounted conditional expectation of the price function at maturity given the information at penultimate time. Thus

$$(4.9) \qquad V_{N-1}(Z) = e^{-m\Delta T} \sum_{j=0}^{j_0} \left[p'_j V_N(Z + R'_j) \right],$$

where p'_j and R'_j are given in Table ??.

Since V_N is piecewise-linear and continuous, and the above is just a linear combination of such functions, the resulting function is also piecewise-linear and continuous, and thus may be completely represented by singular points. The next logical step is of course to figure out a method to compute these singular points.

From equation ??, we note that each singular point l at maturity may have $j_0 + 1$ possible returns, given by

$$(4.10) \qquad B_{l,j} = Z_N^l - R'_j \qquad j \in [j_0]$$

Then the maximum number of singular points at time t_{N-1} is $(j_0 + 1)L_N$. But not all the running sums at time t_{N-1} would belong to the interval $[(N-1)F_{loc}, (N-1)C_{loc}$. All the $B_{l,j}$ which belong to the interval become the abscissa of a singular point at time t_{N-1}. The corresponding singular price is determined by formula ??. The term $V_N(B_{l,j} + R'_k)$, required by the formula, is computed using linearity of the price function at maturity. We just need to figure out the interval l_0 such that $(B_{l,j} + R'_k) \in [Z_N^{l_0}, Z_N^{l_0+1}]$, and evaluate $V_N(B_{l,j} + R'_k)$ by linear interpolation of the extrema. Such an interpolation gives not an approximation by the exact value due to the piecewise-linear nature of the original function.

Finally, the singular points thus obtained are sorted in ascending order on the basis of the running sums. This ordered sequence of singular points $((Z_{N-1}^l, P_{N-1}^l))_l$, $l \in \{1, \ldots, L_{N-1}\}$ completely characterises the price function at time $V_{N-1}(Z)$.

At all times   The previous argument may be applied iteratively in a backward fashion at each step $N-2, N-3, \ldots, 1, 0$ to obtain the singular points at each time step. At time

0, there is only one singular point $(0, P_0^1)$, and P_0^1 provides the exact binomial price of the cliquet option. The equality of this price and the exact binomial price is proved in Proposition ??.

PROPOSITION 4.1 (The singular points method gives the binomial price). P_0^1 *coincides with the exact binomial price with* n *time steps of the cliquet option.*

PROOF. Let the running sum $Z_{i_1,\dots,i_k} = \sum_{j=1}^{k} R'_{i_j}$, $(i_1,\dots,i_k) \in [j_0]^k$, $k \in \{1,\dots,N\}$, and associate them to the prices $P_{i_1,\dots,i_k}$. Then the exact binomial price Q_n of the cliquet option is given by

$$Q_n = e^{-rT} \sum_{i_1,\dots,i_N=0}^{j_0} \left(p'_{i_1} \qquad p'_{i_N} \max\left\{ F_{glob}, \min\left\{ C_{glob}, Z_{i_1,\dots,i_N} \right\} \right\} \right),$$

where p'_{i_k} is the probability of actually taking the path i_k at the time interval k. This expression can be computed backward.

At maturity, set $P_{i_1,\dots,i_N} = \max\left\{ F_{glob}, \min\left\{ C_{glob}, Z_{i_1,\dots,i_N} \right\} \right\}$.

Along the tree the prices $P_{i_1,\dots,i_k}$, $k \in [N-1]$, are evaluated by the backward formula

$$P_{i_1,\dots,i_k} = e^{-r\frac{T}{N}} \sum_{j=0}^{j_0} p'_j P_{i_1,\dots,i_k,j}$$

Finally, we get $Q_n = e^{-r\frac{T}{N}} \sum_{j=0}^{j_0} p'_j P_j$.

We now prove the result by backward induction. At maturity, for every choice of the running sum $Z = Z_{i_1,\dots,i_N}$, the price $P_{i_1,\dots,i_N}$ coincides with the value of the option $v_N(Z)$. By construction (Equation ??), this holds true also for every running sum evaluated at step $N-1$, and iteratively, at all the nodes of the tree. Thus, the price Q_n coincides with the price P_0^1 determined by the singular points algorithm. □

REMARK 4.3 (Generalisation). The method is easily generalised to cases with (discrete) time–varying interest rates and volatilities. In other words, for each *observed* time interval $[t_{i-1}, t_i]$ denoted by $i \in \{1, 2, \dots, N\}$, we can take different values of r_i, σ_i. The technique requires modification only for the computation of the probabilities p' and the returns R' which have to be evaluated at every observation time by using the corresponding r_i and σ_i.

In fact, we may even take varying local floors and caps, denoted F_i^{loc}, C_i^{loc}. In this case, we also have to compute j_{min}, j_{max}, j_0 for each period individually.

REMARK 4.4 (Computational dependence on volatility). The time of computation of the pure binomial method for cliquet options is strongly dependent on the volatility. In fact the total number of paths needed to determine the price is m^N. Because of the presence of the global cap and floor, the number of paths can be reduced to $(j_0 + 1)N$, where $j_0 = j_{max} - j_{min}$. But j_0 is strictly dependent on σ (see Equation ??) and the difference in the computational time between large and small volatilities becomes very high (see also Table ??).

4.2.2   Approximation

The method demonstrated in Section **??** gives us the actual binomial price of the cliquet option. But as was mentioned earlier, the computational complexity of the method is theoretically the same as that of the binomial method, which is m^N. But contrary to the binomial method, the singular point method enables us to use precise, efficient and controlled approximation to accelerate the procedure significantly. Thus its efficacy become apparent when we have time and memory constraints, or we wish to substantially increase the number of intermediate time steps.

The prime idea of the approximation procedure is to remove singular points in a controlled manner in order to simplify the computations. To this end, we fix a given maximal level $h > 0$ of the error in each period. Now, we eliminate singular points in a fashion so that the function after deletion of the points $(\tilde{V}_i)$ is different from the original function by less than (h) at each point. This may be achieved as follows.

Start with the point (Z_i^1, P_i^1). Find the largest index $l > 1$ such that the distances between the straight line joining (Z_i^1, P_i^1) and (Z_i^l, P_i^l) and the points $(Z_i^2, P_i^2), (Z_i^3, P_i^3), \ldots, (Z_i^{l-1}, P_i^{l-1})$ are always less than h. Note that this, coupled with the fact that the original function is piecewise–linear, ensures that the differences in the values of the functions for any value of the running sum would be bounded over by h. Now delete the points $(Z_i^2, P_i^2), (Z_i^3, P_i^3), \ldots, (Z_i^{l-1}, P_i^{l-1})$. The point (Z_i^l, P_i^l) now becomes the second singular point, and we continue the procedure iteratively starting with this point till we cover all the points. Figure **??** elaborates on this graphically. The elimination of points S_3 to S_6 gives function V_1 (bold, blue), which has maximum error $\leqslant h$. Elimination of points S_3 to S_7 gives function V_2 (dashed, red), which has maximum error $> h$. Thus we choose V_1 as the approximation function.

Since at each N the upper bound for the error is h, if we repeat the approximation procedure at every observational time, the total error infused in the price of the option is bounded by Nh.

REMARK 4.5 (Comparison with Asian options). While using approximations for Asian options, the error bound was nh, where n was the number of time steps considered for computation (refer to Section **??** of Chapter **??**). But in the case of the cliquet options, the error depends only on the h and the number of observations – precluding any dependence on the computational parameters. This technique gives a precise approximation of the price without considering upper and lower estimates.

The following proposition proves that the approximation of the price obtained by this method does converge to the price of the cliquet option in the continuous model. The proposition and proof is directly from .

PROPOSITION 4.2 (Convergence to the continuous model). *Let us denote by Q_n^h the approximation of the price evaluated with n time steps and with maximal level of error at every monitoring date $h = h(n)$. If $h(n) \to 0$ as $n \to \infty$, then $Q_n^{h(n)}$ converges to the price of the cliquet option in the continuous model.*
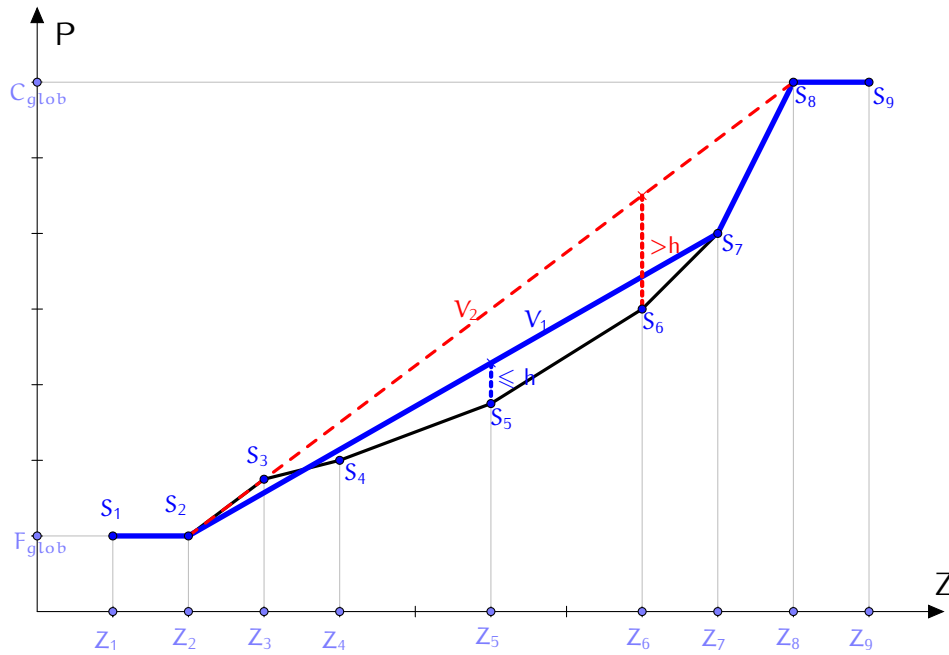
FIGURE 4.2. Approximation procedure

We skip the proof of the proposition since it is not the main focus of our exposition. The proof is mathematically delicate, and needs a general result on the weak convergence of stochastic processes. This is a generalisation of what we saw in Section **??** of Chapter **??**. The result may be found in [**kushner1984approximation**] and [**Kushner:1992:NMS:151172**], and we refer the interested reader to [**Gaudenzi2011**] for a rigorous proof.

4.3   The program

4.3.1   Algorithm

Algorithm 4.1: Pricing cliquet options using the singular points method

---

Input:

    Contract details

        time to maturity: T

        number of observations: N

        local floor and cap: F_{loc}, C_{loc}

        global floor and cap: F_{glob}, C_{glob}

    Details of the underlying asset

        initial price: s_0, volatility: σ, continuous dividend rate: q

    Market parameters – spot interest rate: r

    Computational parameters – time steps within each observed period: m

Output: The price of the option at the initial time

1 **begin**

2   Update F_{glob} and C_{glob} using Equations ??.

3   Set $\Delta T, u, p$ from the formulae in Section ??.

4   Compute the returns and probabilities using Equation ?? and Equation ??.

5   Compute j_{min}, j_{max}, j_0 (Equations ??) and shifted returns (Table ??).

      `// S`$_i$ `and S`$_+$ `denotes the current (i`th`) and next ((i + 1)`th`) list of`

          `singular points.`

6     $S_N \quad \{(NF_{loc}, F_{glob}), (F_{glob}, F_{glob}), (C_{glob}, C_{glob}), (NC_{loc}, C_{glob})\}$ `// Table ??.`

7     $S_+ \quad S_N$

8   **for** $i \; 2 \{N-1, \ldots, 0\}$ **do**

9         $S_i \quad ;, \quad L_+ \quad \text{length}(S_+)$

10   **forall the** $(l, j) \; 2 \, [L_+] \quad [j_0 + 1]$ **do**

11   Compute $B_{l,j}$ using Equation ??, replacing N by i.

12   **if** $(B_{l,j} + R'_k) \not{2} [Z_N^{l_0}, Z_N^{l_0+1}]$ **then**

13   Continue   `// to the next item in the loop`

14   Find l_0 such that $(B_{l,j} + R'_k) \; 2 \, [Z_{i+1}^{l_0}, Z_{i+1}^{l_0+1}]$.

15   Evaluate $V_{i+1}(B_{l,j} + R'_k)$ by linear interpolation of the extrema.

16   Evaluate $V_i(B_{l,j})$ by using Equation ??, replacing N by i.

17             $S_i \quad S_i \; [\; (B_{l,j}, V_{N-1}(B_{l,j}))$

18   Approximate as described in Section ??.

19         $S_+ \quad S_i$

20   **return** $(S_i)_{1,2}$ `// Singular price at time 0`

4.3.2 Implementation

The algorithm was implemented in Python 3.5.0 (2015–09–13).

mymath.py   This file contains the $\binom{n}{k}$ function.

```python
# ====================================================================== #
__author__ = "Sudip Sinha"

from math import factorial



# ---------------------------------------------------------------------- #

# @profile
def choose( n: int, r: int ):
    """Number of ways of choosing r objects among n."""

    assert (type( n ) == int
            and type( r ) == int and
            0 <= r <= n), \
        "n = {n} and r = {r} must be non-negative integers".format(
        n = n, r = r )

    if r > (n / 2):
        r = n - r
    c = 1
    for i in range( n, n - r, -1 ):
        c *= i
    return c // factorial( r )

# ====================================================================== #
```

cliquet.py   This file contains the function *cliquet_sp*, which returns the price of the cliquet option at time 0.

```python
# ====================================================================== #
__author__ = "Sudip Sinha"

from math import exp, log, sqrt, ceil, floor, isclose
from mymath import choose



# ---------------------------------------------------------------------- #
```

```python
# @profile
def cliquet_sp( r: float, q: float, sigma: float, t: float,
                f_loc: float, c_loc: float, f_glob: float, c_glob: float,
                N: int, m: int, h: float = 0. ) -> float:
    """Price of a cliquet option using the singular point method"""

    err_p = "ERROR: p_u must be a probability! " + \
            "We must have dt < ( sigma ** 2 / ( r - q ) ** 2 ). " + \
            "You may want to decrease the time interval."

    # Time period for each option
    n = m * N
    dt = t / n

    # Checks
    f_glob = max( N * f_loc, f_glob )
    c_glob = min( N * c_loc, c_glob )

    # Singular points for maturity (go to the nxt)
    now = [(N * f_loc, f_glob),
           (f_glob, f_glob),
           (c_glob, c_glob),
           (N * c_loc, c_glob)]
    if f_glob == N * f_loc:
        now = now[1:]
    if c_glob == N * c_loc:
        now = now[:-1]

    # Uniqueness of sigma
    if type( sigma ) == float:
        unique_v = True
        assert dt < (sigma ** 2 / (r - q) ** 2), err_p
    elif type( sigma ) == list and len( sigma ) == N:
        unique_v = False
        sigma_list = sigma
    else:
        raise ValueError

    # Uniqueness of r
    if type( r ) == float:
        unique_r = True
    elif type( r ) == list and len( r ) == N:
        unique_r = False
        r_list = r
    else:
```

```python
        raise ValueError

    # Unique f_loc and c_loc
    unique_fc = True

    if unique_r:
        # Effective interest rate for observable time periods
        R_N = exp( (r - q) * t / N )
        # Effective interest rate for computational time periods
        R_n = exp( (r - q) * t / n )

    if unique_v:
        u = exp( sigma * sqrt( dt ) )   # Up factor
    # d = 1. / u

    if unique_r and unique_v:
        # Risk neutral probability
        p_u = (R_n * u - 1) / (u * u - 1)
        p_d = 1. - p_u
        assert 0. <= p_u <= 1., err_p

    if unique_v and unique_r and unique_fc:
        # Possible range of return
        j_min = max( 0, floor(
            log( f_loc + 1 ) / (2 * sigma * sqrt( dt )) + m / 2 ) )
        j_max = min( m, ceil(
            log( c_loc + 1 ) / (2 * sigma * sqrt( dt )) + m / 2 ) )
        j0 = j_max - j_min

        # prb denotes p'
        prb = [0 for j in range( j0 + 1 )]
        prb[0] = 0
        for j in range( 0, j_min + 1 ):
            prb[0] += choose( m, j ) * (p_u ** j) * (p_d ** (m - j))
        prb[j0] = 0
        for j in range( j_max, m + 1 ):
            prb[j0] += choose( m, j ) * (p_u ** j) * (p_d ** (m - j))
        for j in range( j_min + 1, j_max ):
            prb[j - j_min] = choose( m, j ) * (p_u ** j) * (
            p_d ** (m - j))
        assert isclose( sum( prb ), 1. ), "Sum of PMF must be unity."

        # ret denotes R'
        ret = [0 for j in range( j0 + 1 )]
        ret[0] = f_loc
```

```
        ret[j0] = c_loc
        for j in range( j_min + 1, j_max ):
            ret[j - j_min] = u ** (-m + 2 * j) - 1

    # Come back from the nxt, one step at a time.
    for i in range( N - 1, -1, -1 ):
        nxt = now
        now = []

        # Checks
        if not unique_v or not unique_r:
            if not unique_r:
                r = r_list[i]
                # Effective interest rate for observable time periods
                R_N = exp( (r - q) * t / N )
                # Effective interest rate for computational time periods
                R_n = exp( (r - q) * t / n )
            if not unique_v:
                sigma = sigma_list[i]
                # Up factor
                u = exp( sigma * sqrt( dt ) )
                # Possible range of return
                j_min = max( 0, floor(
                    log( f_loc + 1 ) / (
                    2 * sigma * sqrt( dt )) + m / 2 ) )
                j_max = min( m, ceil(
                    log( c_loc + 1 ) / (
                    2 * sigma * sqrt( dt )) + m / 2 ) )
                j0 = j_max - j_min
            assert dt < (sigma ** 2 / (r - q) ** 2), err_p

            # Risk neutral probability
            p_u = (R_n * u - 1) / (u * u - 1)
            p_d = 1. - p_u
            assert 0. <= p_u <= 1., err_p

            # prb denotes p'
            prb = [0 for j in range( j0 + 1 )]
            prb[0] = 0
            for j in range( 0, j_min + 1 ):
                prb[0] += choose( m, j ) * (p_u ** j) * (p_d ** (m - j))
            prb[j0] = 0
            for j in range( j_max, m + 1 ):
                prb[j0] += choose( m, j ) * (p_u ** j) * (
                p_d ** (m - j))
```

```python
        for j in range( j_min + 1, j_max ):
            prb[j - j_min] = choose( m, j ) * (p_u ** j) * (
            p_d ** (m - j))
        assert isclose( sum( prb ),
                        1. ), "Sum of PMF must be unity."

        # ret denotes R'
        ret = [0 for j in range( j0 + 1 )]
        ret[0] = f_loc
        ret[j0] = c_loc
        for j in range( j_min + 1, j_max ):
            ret[j - j_min] = u ** (-m + 2 * j) - 1

    # Obtain a sorted list for B
    b_list = []
    for l in range( 0, len( nxt ) ):
        for j in range( 0, j0 + 1 ):
            b = nxt[l][0] - ret[j]
            if (i * f_loc) <= b <= (i * c_loc):
                close = False
                for rsz in b_list:
                    if isclose( rsz, b ):
                        close = True
                        break
                if not close:
                    b_list.append( b )
    b_list = sorted( b_list )

    for b in b_list:
        v = 0
        for k in range( 0, j0 + 1 ):
            # Running sum Z corresponding to B at the nxt time step.
            rszp = b + ret[k]
            if rszp <= nxt[0][0]:
                vk = nxt[0][1]
            elif rszp >= nxt[-1][0]:
                vk = nxt[-1][1]
            else:
                for idx in range( 0, len( nxt ) ):
                    if nxt[idx][0] <= rszp < nxt[idx + 1][0]:
                        vk = (nxt[idx][1] +
                                (nxt[idx + 1][1] - nxt[idx][1]) /
                                (nxt[idx + 1][0] - nxt[idx][0]) *
                                (rszp - nxt[idx][0]))
            v += prb[k] * vk
```

```python
            v /= R_N
            now.append( (b, v) )

        # Approximations
        if h > 0.:
            l = 0  # l is the starting index
            while l < len( now ) - 2:
                approx = True
                for j in range( len( now ) - 1, l + 1, -1 ):
                    slope = (now[j][1] - now[l][1]) / (
                    now[j][0] - now[l][0])
                    for k in range( l + 1, j ):
                        approx = True
                        delta = abs(
                            slope * (now[k][0] - now[l][0]) +
                            now[l][1] - now[k][1] )
                        if delta >= h:
                            approx = False
                            break
                    if approx:
                        break
                if approx and j > l + 1:
                    del now[l + 1:j]
                l += 1

    return now[0][1]


# ===================================================================== #
```

4.4   Results and conclusions

In this section, we provide some numerical results that we have obtained from our program, along with the reported values of the same from [**Gaudenzi2011**]. They compare the results with other techniques, namely Monte Carlo and the finite difference method of Windcliff et al[**Windcliff2006**]. In all the experiments, the number of time steps considered were 10, 20, 50, 100, 200, 500 and 1000. The cases considered are as follows:

(1) Constant volatility of $\sigma = 0.2$

(2) Constant volatility of $\sigma = 0.02$, in order to test the efficiency of the methods for low volatility cases

(3) Varying volatility for each observational period, as $\sigma(i) = 0.05 + 0.04i, i = 1, \ldots, 8$. (This is denoted by *Var* in the σ column of Table ??.) The contract maturity is of 2 years, with quarterly monitoring dates.

In order to obtain corroboration with the continuous value, we also report the price obtained by using Monte Carlo method with 1,000,000 of trials, along with the values in the 95% confidence interval.

The specifications of the cliquet contract for the cases with constant volatility are as follows.

$$F_{loc} = 0, C_{loc} = 0.08, F_{glob} = 0.16, C_{glob} = \infty$$

$$T = 5 \text{ years}$$

$$N = 5$$

$$r = 0.03$$

All simulations were run on a computer with the specifications given in Table ?? of Chapter ??. All times were measured to 3 significant digits in seconds using the *timeit* module of python, run from the terminal.

Table ?? highlights the results.

REMARK 4.6 (Computational dependence on volatility). The computational time of the pure binomial method for cliquet options is strongly dependent on the volatility. In fact the total number of paths needed to determine the price is mN. Because of the presence of the global cap and floor, the number of paths can be reduced to $(j_0 + 1)N$. But j_0 is strictly dependent on σ (see Equation ??), and the difference in the computational time between large and small volatilities becomes very high, as is evinced from the constant volatility cases in Table ??.

REMARK 4.7 (Computational complexity). We remarked earlier that although the computational complexity of the algorithm without approximation is the same as that of the binomial model, enabling approximations decreases the complexity to a polynomial time algorithm. In fact, the Figure ?? corroborates exactly to this. The blue dots denotes the time taken to run the algorithm for constant volatility of $\sigma = 0.2$. The green line is a quadratic function

---

[4]∞ means time taken is more than an hour.

TABLE 4.3. Results for cliquet options

σ	m	Price			Time (s)[4]	
		Bin	SP	MC	Bin	SP
	10	0.165661911	0.165661911		0.000561	0.000555
	20	0.172300056	0.172300056		0.000671	0.000675
	50	0.172501269	0.172501269	0.174106	0.00185	0.00175
0.2	100	0.172501269	0.173927464	(0.17398 –	0.00413	0.00297
	200	0.173716366	0.173716366	0.17423)	0.0165	0.00828
	500	0.173922597	0.173922671		0.0875	0.0437
	1000	0.174051949	0.174051983		2.38	0.183
	10	0.151234115	0.151234416		0.128	0.0295
	20	0.149734212	0.149734992		1.01	0.199
	50	0.150228984	0.150230185	0.150525	7.96	0.868
0.02	100	0.150386306	0.150387954	(0.150472 –	70	2.36
	200	0.150465004	0.150466828	0.150578)	600	6.09
	500	0.150508871	0.150510526		∞	24.2
	1000	0.150522368	0.150524027		∞	55
	10	0.188738321	0.188738321		0.00258	0.00253
	20	0.192927333	0.192927296		0.0153	0.0139
	50	0.192650261	0.192650204	0.226169	0.510	0.0638
Var	100	0.193452450	0.193452358	(0.225978 –	2.43	0.162
	200	0.193657595	0.193657617	0.226360)	14.6	0.364
	500	0.193742799	0.193742732		3600	1.11
	1000	0.193776040	0.193776165		∞	2.97

of the number of time steps m, whereas the red line is an exponential function of the same. We varied the parameters so that the curves fits the data. We see that the quadratic function serves as an upper bound for the running time for most data points in the plot, except for a few points, which may be treated as outliers. Thus, we may conclude that the computational complexity of the algorithm is $O(m^2)$ in this case. In fact, our experiments show that this is indeed the general trend. Thus, even though we have not found out the theoretical complexity of the algorithm, in practice the algorithm is competitive.

Note that for the Asian case, the experimental complexity is $O(n^3)$, whereas in the cliquet case, it is $O(m^2)$.

Concluding remarks   We note that, in all cases, the difference between the prices in obtained by the singular points method and the binomial method is significantly smaller than the theoretical value of N 10^{-6}. Moreover, as in the case examined $NC_{loc} < C_{glob}$ (which is equivalent to $NC_{loc} = C_{glob}$), the price functions $V_i(Z)$ are always convex and this implies that the approximation procedure will provide an upper estimate of the exact binomial value.
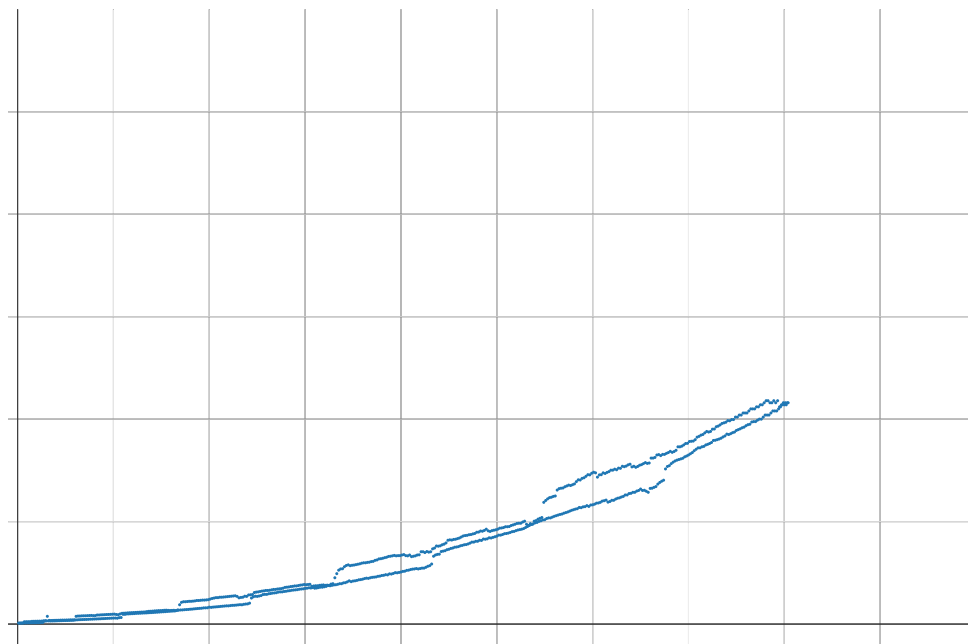
FIGURE 4.3. Timing the singular point method for cliquet option with constant volatility $\sigma = 0.2$. The quadratic curve provides a pragmatic bound for the runtimes. The exponential curve is too conservative.

Thus, the singular points method is quite capable in pricing cliquet options in a Black–Scholes framework with piecewise constant interest rates and volatilities. The implementation is quite simple, and the price obtained by the method converges to the price of continuous model. In absence of approximation, the price matches the exact binomial price. The ability to set *a priori* error bounds differentiates it from the alternative algorithms.

Numerical comparisons indicate that the method is accurate both for standard and small volatilities, and that it avoids the computational problems arising from the application of the standard binomial method. For small volatilities and large numbers of observation times in particular, the improvement with respect to the binomial standard technique is very significant. Experimental results show that the computational complexity in practice is around $O(m^2)$.

Chapter 5

Epilogue

In the thesis, our main focus was to develop the ideas behind the pricing of options, and to study two particular classes of path-dependent exotic options, namely Asian options and cliquet options. We saw that these options do not lend themselves to be priced under the Black–Scholes framework, and we have to resort to alternative ways of pricing them. They can be priced in the Cox-Ross-Rubinstein model, but the complexity is exponential with respect to the number of time steps, and is infeasible in practice. The singular points method is a new lattice based method which has the same theoretical complexity as the Cox-Ross-Rubinstein model. But it excels in allowing for approximations, and this reduces the complexity from exponential time to polynomial time of very low orders.

Albeit similar in the basic structure, the theory of how the singular points method may be applied for Asian options and cliquet options are quite different, as are the corresponding algorithms. On the one hand, in the case of Asian options, we saw that the algorithm neither did generalise for Asian options with geometric mean, nor for local volatility models. Moreover, the pre-existing algorithms (some lattice–based) were quite competitive to this method.

On the other hand, for cliquet options, the method was able to handle cases of variable rates of interest, volatility, and local caps and floors with ease. The algorithm is extremely fast in this case, and does in fact outdo most other competing algorithms in terms of simplicity, ease of implementation and low memory requirement. Furthermore, this seems to be one of the very few tree lattice–based methods available to price cliquet options.

As a further research, it might be interesting to explore on the possibility of customising the method for other exotic option types. It would also be interesting to conduct a further study on the theoretical complexity of the algorithm, especially the dependence of the number, closeness and removability of singular point with respect to initial data $(S_0, K, T, \sigma, r, F_{loc}, C_{loc}, F_{glob}, C_{glob})$ and the computational parameters (m, h).

In conclusion, the method is quite specific; its implementation depends very much on the type of option. Nevertheless, it is a significant leap in the domain of tree methods because of its inherent advantages over other methods.

List of Figures

List of Tables