Pricing of European- and American-style Asian Options using the Finite Element Method

Jesper Karlsson
Pricing of European- and American-style Asian Options using the Finite Element Method

June 2018

Supervisors
Rasmus Leijon
Cinnober Financial Technology AB
Joakim Ekspong
Department of Physics

Examiner
Martin Rosvall
Department of Physics

Jesper Karlsson
kljkarlsson@gmail.com
Master's Thesis in Engineering Physics
Umeå University
©2018 Jesper Karlsson
Abstract

An option is a contract between two parties where the holder has the option to buy or sell some underlying asset after a predefined exercise time. Options where the holder only has the right to buy or sell at the exercise time is said to be of European-style, while options that can be exercised any time before the exercise time is said to be of American-style. Asian options are options where the payoff is determined by some average value of the underlying asset, e.g., the arithmetic or the geometric average. For arithmetic Asian options, there are no closed-form pricing formulas, and one must apply numerical methods. Several methods have been proposed and tested for Asian options. For example, the Monte Carlo method is slow for European-style Asian options and not applicable for American-style Asian options. In contrast, the finite difference method have successfully been applied to price both European- and American-style Asian options. But from a financial point of view, one is also interested in different measures of sensitivity, called the Greeks, which are hard approximate with the finite difference method. For more accurate approximations of the Greeks, researchers have turned to the finite element method with promising results for European-style Asian options. However, the finite element method has never been applied to American-style Asian options, which still lack accurate approximations of the Greeks.

Here we present a study of pricing European- and American-style Asian options using the finite element method. For European-style options, we consider two different pricing PDEs. The first equation we consider is a convection-dominated problem, which we solve by applying the so-called streamline-diffusion method. The second equation comes from modelling Asian options as options on a traded account, which we solve by using the so-called $cG(1)cG(1)$ method. For American-style options, the model based on options on a traded account is not applicable. Therefore, we must consider the first convection-dominated problem. To handle American-style options, we study two different methods, a penalty method and the projected successive over-relaxation method.

For European-style Asian options, both approaches give good results, but the model based on options on a traded account show more accurate results. For American-style Asian options, the penalty method give accurate results. Meanwhile, the projected successive over-relaxation method does not converge properly for the tested parameters.

Our result is a first step towards an accurate and fast method to calculate the price and the Greeks of both European- and American-style Asian options. Because good estimations of the Greeks are crucial when hedging and trading of options, we anticipate that the ideas presented in this work can lead to new ways of trading with Asian options.

**Keywords:** Option pricing, finite element, streamline-diffusion, penalty method, projected successive over-relaxation, Asian options, American-style Asian options, Eurasian options, Amerasian options
## Contents

1 Introduction 1  
1.1 Background ................................................. 1  
1.2 Objective .................................................. 3  
1.3 Outline .................................................... 3  

2 Options 5  
2.1 The Mathematics of Option Pricing .......................... 5  
  2.1.1 Wiener Processes .................................... 5  
  2.1.2 Itô’s Lemma ......................................... 5  
  2.1.3 Self-Financing Portfolios ............................ 6  
2.2 Option Pricing ................................................ 6  
  2.2.1 Black-Scholes Equation .............................. 6  
  2.2.2 Pricing European Options ............................. 8  
2.3 American Options ............................................ 9  
  2.3.1 A Variational Inequality Problem .................... 10  
2.4 Asian Options ............................................... 10  
  2.4.1 A Pricing PDE ....................................... 11  
  2.4.2 American-Style Asian Options ...................... 14  
  2.4.3 Options on a Traded Account ....................... 14  

3 Numerical Methods 19  
3.1 The Finite Element Method ................................. 19  
  3.1.1 Weak Formulations ................................ 19  
  3.1.2 Finite Element Approximation ...................... 22  
  3.1.3 Streamline-Diffusion Method ....................... 26  
3.2 Numerical Methods for American-Style Options ............... 28  
  3.2.1 The Projected Successive Over-Relaxation Method .. 28  
  3.2.2 The Penalty Method ................................ 29  

4 Implementations 31  
4.1 The Finite Element Method ................................. 31  
  4.1.1 FEM in 1D ............................................ 31  
  4.1.2 FEM in 2D ............................................ 32  
  4.1.3 Meshing .............................................. 38  
4.2 Streamline-Diffusion method ................................ 39  
4.3 PSOR ....................................................... 39  
4.4 The Penalty Method ....................................... 40
Chapter 1

Introduction

1.1 Background

An option is a financial instrument that today is commonly traded on the financial market in various forms. It is a contract between two parties, the holder and the writer of the contract. The holder has the option to buy/sell an underlying asset, e.g. some stock or commodity, for a specified strike price at a specified exercise time. Meanwhile the writer is obliged to sell/buy. Every option is either a call option or a put option, a call option is when the holder has the right to buy and a put option is when the holder has the option to sell. The simplest option to consider is the European call option. It is specified by a strike price $K$ and the exercise time $T$. The holder then has the option to buy some underlying asset for the strike price $K$ at precisely the exercise time $T$. Another common option type that is traded is the American option. It can be described as a European option where the holder has the right to buy/sell at any time before the exercise time. Such options are said to have the early exercise feature.[1]

To be able to trade options on the financial market, we must have some method to value them. Therefore, an important aspect of options is how these should be valued. It turns out that under some assumptions of the market, one can derive pricing equations whose solutions yields well-defined prices for the different options. In 1973, Fischer Black and Myron Scholes published a paper where they had derived an equation to price European call and put options.[2] They also solved this equation to give a closed-form expression for the price. The equation, which is a parabolic partial differential equation of one spatial dimension, is today known as the Black-Scholes equation, and the closed-form solution to the European option pricing problem is called the Black-Scholes formula.[1]

European and American options are today known as examples of vanilla options. Options that are not vanilla options are called exotic options. In this work we are interested in one type of exotic options called Asian options. Asian options have a payoff that is determined from some type of average of the underlying asset in opposite to the European and American option where the payoff is determined by the value of the underlying asset at the time the option is exercised.[3] These options, whose payoff are determined from some type of average, have the advantage that they are less sensitive to market manipulations, and also they are cheaper than the European and American option.[4], [5] There are mainly two types of averages used...
for Asian options, arithmetic and geometric averages. For each type of average we also have fixed- and floating-strike Asian options. The difference between fixed- and floating-strike is how the payoff is determined.

For Asian options, one can also derive a pricing equation in a similar fashion to how Black-Scholes equation is derived. This yields a PDE with two spatial dimensions that is convection dominated. Therefore, this equation exhibit known problems that arise for convection dominated problems. For the floating-strike Asian option, this PDE can be reduced to one spatial dimension by a variable transformation.[6]

By a completely different approach developed by Jan Večeř, it is possible to derive a PDE in one spatial dimension that can be used to price both fixed- and floating-strike Asian options. For American-style Asian options, i.e. options with the early exercise feature, Večeř’s approach does not work.[7], [8] Thus, to price American-style fixed-strike Asian options one has to solve the two-dimensional PDE, and for American-style floating-strike Asian options one can solve the reduced PDE.

None of the pricing PDEs for arithmetic average Asian options can be solved analytically. Therefore one must consider numerical approaches to price these. Since their introduction, several numerical approaches have been developed. In [4] the authors price European-style Asian options using the Monte Carlo method, their method is accurate but slow. Also, one can not use Monte Carlo to price American-style Asian options [9]. In the early 90s, several approximations were also developed for Asian options of European-style, see e.g. [10]–[12]. Another approach based on the Laplace transform was developed in [13], but this method suffers from that the numerical inversion is problematic for low volatility and short expiration time. Regarding American-style Asian options, not as many methods have been developed. The earliest PDE method to price American-style Asian options that we are aware of was developed by Barraquand and Pudet in [14]. They developed a method they called forward shooting grid. A more traditional approach were developed by Zvan, Forsyth and Vetzal in [15], they developed a finite differene method (FDM) approach using a Van Leer flux limiter. More recent work, Rashidinia and Jamalzadeh [9] presents a method based on modified bicubic B-spline collocation.

However, the common problem that these numerical methods have is that they to do not give a satisfactory method to calculate the so called Greeks. The Greeks are different measure of sensitivity of the price with respect to the different parameters. From a financial point-of-view, it often more interesting to look at the Greeks than the actual price of an option. Mathematically the Greeks are partial derivatives of the price with respect to the different parameters. This means for example if one wants to use Monte Carlo, one has to run a different simulation for each Greek which is time consuming, and finite difference methods only calculates the solution at isolated nodes, which means that one must use some finite difference expression of the solution at the nodes to calculate the partial derivatives. Therefore, it is interesting to study using the finite element method (FEM) instead since from a finite element method one get a solution that is defined at every point, and by choosing the basis functions one can control the regularity of the solution. Thus, the idea is that by using FEM we can get a method that can accurately determine the Greeks. Another advantage of FEM is that compared to FDM, FEM has the advantage that is can handle more complex geometries, for instance one can consider specialized non-uniform meshes to solve the PDEs on. Also FEM handles Neumann boundary conditions better FDM. [16]

There do exist some previous work regarding using FEM to price options. Some
early work regarding option pricing using FEM can be found in [17]–[19]. In the PhD thesis [17] by Michael J. Tomas III, Tomas studies pricing of American and Barrier options using FEM. In [18], the authors studies different pricing problems using FEM and gives a general approach to achieve this. As an illustrative example they consider the following three pricing problems: convertible bonds, Asian options and two asset options. Finally, in [19], the authors look at a generalized model for the European call option and solves it using FEM. For Asian options, Zvan, Forsyth and Vetzal develops in [18] a FE method to solve the two-dimensional PDE to price Asian options, and in [5], Foufas and Larson presents a FE method solve the one-dimensional equation derived by Večeř. They also proves a posteriori error estimate and develops an adaptive FEM solver. To our knowledge, no one has explored the possibility of using FEM to price American-style Asian options and this will be the focus of this thesis.

1.2 Objective

The objective of this work is to answer whether FEM is a feasible method to use for pricing Asian options. We shall consider Asian options of both American- and European-style, i.e. both with and without the early exercise feature. To answer this question we shall price a number of variations of Asian options and compare our results with those obtained by others who also looked at this problem.

1.3 Outline

The outline of this thesis is as follows. First, in Chapter 2 we treat the theory of option pricing and look at the different pricing PDEs that exists for Asian options of both European- and American-style. Next, in Chapter 3, we look at FEM and how we apply it to the pricing PDEs of Chapter 2. We shall also treat two numerical methods to handle the early exercise feature of American-style options. Then, in Chapter 4 we look at how we implement the different methods. Finally, in Chapter 5 we presents and analyse our results, and in Chapter 6 we make some final remarks regarding this project.
Chapter 2

Options

2.1 The Mathematics of Option Pricing

The mathematical results that we shall present here will be needed later for the derivation of the pricing PDEs. All results in this section are from [1].

2.1.1 Wiener Processes

A Wiener process is a stochastic process $W(t)$ such that the following holds.

(i) $W(0) = 0$.

(ii) The process $W$ has independent increments, i.e. if $r < s < t < u$ then $W(u) - W(t)$ and $W(s) - W(r)$ are independent stochastic variables.

(iii) For $s < t$ the stochastic variable $W(t) - W(s)$ has the Gaussian distribution $N[0, \sqrt{t-s}]$.

(iv) $W$ has continuous trajectories.

2.1.2 Itô’s Lemma

Let $X(t)$ be a stochastic process with the stochastic differential given by

$$dX(t) = \mu dt + \sigma dW(t),$$

where $\mu$ and $\sigma$ are real numbers, and let $f : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}$ be a $C^{1,2}$-function. Define a new stochastic process $Z$ by $Z(t) = f(t, X(t))$, then the stochastic differential of $Z$ is given by

$$df(t, X(t)) = \left( \frac{\partial f}{\partial t} + \mu \frac{\partial f}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial x^2} \right) dt + \sigma \frac{\partial f}{\partial x} dW(t). \quad (2.1)$$

We shall also need the two dimensional version of Itô’s lemma. Let $W^1(t)$ and $W^2(t)$ be two independent Wiener processes and consider two stochastic processes $X(t)$ and $Y(t)$ with their stochastic differentials given by

$$dX(t) = \mu_1 dt + \sigma_{11} dW^1(t) + \sigma_{12} dW^2(t),$$
$$dY(t) = \mu_2 dt + \sigma_{21} dW^1(t) + \sigma_{22} dW^2(t).$$
Let \( f : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R} \) be a \( C^{1,2} \)-function, and as before, define a new stochastic process \( Z \) defined by \( Z(t) = f(t, X(t), Y(t)) \). Then, the stochastic differential of \( Z \) is given by

\[
\begin{align*}
 df(t, X(t), Y(t)) &= \left[ \frac{\partial f}{\partial t} + \mu_1 \frac{\partial f}{\partial x} + \mu_2 \frac{\partial f}{\partial y} \\
 &\quad+ \frac{1}{2} \left( \sigma_{11} \frac{\partial^2 f}{\partial x^2} + \sigma_{12} \sigma_{21} \frac{\partial^2 f}{\partial x \partial y} + \sigma_{22} \frac{\partial^2 f}{\partial y^2} \right) \right] dt \\
 &\quad+ \frac{\partial f}{\partial x} (\sigma_{11} dW_1(t) + \sigma_{12} dW_2(t)) + \frac{\partial f}{\partial y} (\sigma_{21} dW_1(t) + \sigma_{22} dW_2(t)).
\end{align*}
\] (2.2)

### 2.1.3 Self-Financing Portfolios

We shall require some results regarding portfolios, in particular self-financing portfolios. Consider a market consisting of \( N \) assets that we shall label with \( S_i(t) \), \( i = 1, \ldots, N \). Let \( h_i(t) \) denote the number of asset \( i \) held at time \( t \), and let \( h(t) \) denote the portfolio \((h_1(t), \ldots, h_N(t))\). Also, let \( V(t) \) denote the value process of the portfolio at time \( t \). Then, it holds that

\[
V(t) = \sum_{i=1}^{N} h_i(t) S_i(t).
\]

A portfolio is said to be self-financing if there is no external infusion or withdrawal of money. It can be showed that a portfolio \( h(t) \) is self-financing if, and only if,

\[
dV(t) = \sum_{i=1}^{N} h_i(t) dS_i(t).
\] (2.3)

Another concept that we need is risk-free portfolios. A portfolio is said to be risk-free if the price process is on the following form

\[
dV(t) = k(t)V(t)dt,
\]

where \( k(t) \) is some function.

### 2.2 Option Pricing

Here we shall look at how options can be priced under the Black-Scholes model. We will end this section by looking more closely at the European call and put option.

#### 2.2.1 Black-Scholes Equation

A European call option on some underlying asset \( S(t) \) is described by its strike price \( K \) and exercise time \( T \). If we assume that the holder always acts optimal, i.e. the holder will exercise the option at time \( T \) if \( K < S(T) \) and buy for profit, otherwise the holder will not exercise. Then, we can describe the payoff of the European call option at time \( T \) as \( \Phi(S(T)) = \max(S(T) - K, 0) \). The function \( \Phi \) is called a payoff function and is different for each option, e.g. for the European put option, \( \Phi(S(T)) = \max(K - S(T), 0) \).
We shall here consider how to price an option with a given payoff function. We will show that under certain assumptions on the financial market the value of an option is well-defined and can be determined from a parabolic PDE. Assume that the market consists of two assets, a risk-free asset $B$, called the bond, and a risky asset $S$, called the stock, with the dynamics given by

$$
\begin{align*}
    dB(t) &= rB(t) dt, \\
    dS(t) &= S(t) \alpha dt + S(t) \sigma dW(t),
\end{align*}
$$

(2.4)

Here $r$ is assumed to be a positive constant called the risk-free rate of interest that determines the continuous rate of return of the bond, $\alpha$ is a positive constant called the local mean rate of return, $\sigma$ is a positive constant called the volatility, and $W$ is a Wiener process. For simplicity, assume that the stock does not pay dividends. This model of the market is also known as the Black-Scholes model [1].

The main assumptions that we shall make on the market are, it is possible to buy and sell fractional amounts of the bond and the stock, there are no transaction costs, and the market is free of arbitrage opportunities. That the market is free of arbitrage opportunities, or simply arbitrage free, means that there are not two prices for the same object. The most important implication of a market without arbitrage opportunities is that a risk-free self-financing portfolio must have a price process on the following form

$$
    dV(t) = rV(t) dt.
$$

(2.5)

For a derivation see [1]. This results will be used several times, among others to derive pricing PDEs for different options.

Let $V(t, S(t))$ be the value of an option that is formed at $t = 0$ with the payoff $\Phi(S(T))$ at time $t = T$ and assume that $V$ is once differentiable in $t$ and twice differentiable in $S$. First of all, for there to be no arbitrage opportunities the value of the option at $t = T$ must equal the payoff. Next, we apply Itô’s Lemma (Equation (2.1)) on $V$, which yields that

$$
    dV = \sigma S \frac{\partial V}{\partial S} dW + \left( \alpha S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} \right) dt.
$$

(2.6)

Now, consider a portfolio with value $\Pi$ given by $\Pi = V - \Delta S$, where $-\Delta$ is the amount of the underlying asset $S$ held at each time. During each time-step $dt$ we hold $\Delta$ fixed, therefore during one time-step $dt$ it holds that

$$
    d\Pi = dV - \Delta dS.
$$

(2.7)

We see that $d\Pi$ is on the same form as in Equation (2.3), thus $\Pi$ is self-financing. Now, let $\Delta = \frac{\partial V}{\partial S}$ and insert Equation (2.6) and (2.4) into (2.7), this yields that

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

(2.8)
By equality of Equation (2.7) and (2.8), we reach the expression
\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0. \tag{2.9}
\]
Equation (2.9) is a backward parabolic PDE, which is called Black-Scholes equation named after Fischer Black and Myron Scholes who were first to publish these results in 1973 [2]. So, for an option with the payoff function \( \Phi(S(T)) \), the value of the option at \( t = 0 \) is given by \( V(0, S(0)) \) where \( V(t, S(t)) \) satisfy the boundary value problem
\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0 \quad \text{on } [0, T) \times \mathbb{R}^+, \\
V(T, S) = \Phi(S) \quad \text{on } \mathbb{R}^+.
\]

### 2.2.2 Pricing European Options

With the Black-Scholes equation, the value for a European call option with strike price \( K \) and exercise time \( T \) is given by
\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0 \quad \text{on } [0, T) \times \mathbb{R}^+, \\
V(T, S) = \max(S - K, 0) \quad \text{on } \mathbb{R}^+.
\]
For the European put option, it holds instead that \( V(T, S) = \max(K - S, 0) \). We shall now look at a couple of results regarding European options.

#### Call-Put Parity

Let \( C \) denote the price of a European call option and let \( P \) denote the price of a European put option. We can derive a relation between \( C \) and \( P \) that will be useful for us. Consider a portfolio \( \Pi \) given by
\[
\Pi = S + P - C.
\]
At the time of expiration, the payoff is given by
\[
S + \max(K - S, 0) - \max(S - K, 0) = K.
\]
Hence, the payoff is always \( K \) and is risk-free. Thus, the value of the portfolio satisfy Equation (2.5). The solution to Equation (2.5) is a simple exponential function, and since the payoff is \( K \) at the expiration time \( T \), the value of the portfolio is given by
\[
S + P - C = Ke^{-r(T-t)}. \tag{2.10}
\]
Equation (2.10) is called a call-put parity since it gives a relation between a call and a put option.

#### Black-Scholes Formula

Consider a European call option with strike price \( K \) and time \( T \) to expiration. Under the Black-Scholes model it is possible to solve Black-Scholes equation analytically.
This result is known as Black-Scholes formula and determines the price for a European call option when \( r \) and \( \sigma \) is assumed to be constants. So, according to Black-Scholes formula the price \( V(t, s) \) of a European call option is given by

\[
V(t, s) = sN[d_1(t, s)] - e^{-(r(T-t))}KN[d_2(t, s)]
\]

(2.11)

where \( N[\cdot] \) is the cumulative distribution function for standard normal distribution and

\[
d_1(t, s) = \frac{1}{\sigma \sqrt{T-t}} \left( \ln \left( \frac{s}{K} \right) + \left( r + \frac{1}{2} \sigma^2 \right) (T-t) \right),
\]

\[
d_2(t, s) = d_1(t, s) - \sigma \sqrt{T-t}.
\]

For a derivation of Black-Scholes formula see [1]. We shall make use of Black-Scholes formula later when we want to verify the numerical implementations by solving Black-Scholes equation and comparing with Black-Scholes formula.

**Boundary Conditions**

The Black-Scholes equation is defined on the semi-infinite domain \([0, \infty)\). In order solve the equation using the finite element method we must limit the domain to some finite interval \( I \) of \([0, \infty)\). Note that at \( S = 0 \) the second order term in \( S \) vanish, so the problem becomes degenerate. To avoid this we define \( I \) such that the left boundary of \( I \) is greater than 0. Therefore, to solve this problem we need to know how the solution behaves at the limits \( S \to 0 \) and \( \infty \) to apply proper boundary conditions.

First, consider a European put option. As \( S \to \infty \) it becomes less likely that the option is exercised at the exercise time \( T \). Thus, \( V \to 0 \) as \( S \to \infty \). Next, consider the limit \( S \to 0 \). If \( S = 0 \), then from Equation (2.4) \( dS \to 0 \) so \( S \) will stay at 0. Therefore, the payoff at maturity will be \( K \). Taking the risk-free rate of interest into account, the value at a time \( t \) before maturity is given by \( Ke^{-(r(T-t))} \). Thus, \( V \to Ke^{-(r(T-t))} \) as \( S \to 0 \).

We can now get the boundary conditions for a call option from the call-put parity (Equation (2.10)) and the boundary conditions for a put option. At \( S \to 0 \) we get that \( V \to 0 \), and at \( S \to \infty \) we get that \( V \to S - Ke^{-(r(T-t))} \).

### 2.3 American Options

An American option is similar to a European option except that the holder of the option has the right to exercise the option at any time before the time of expiration. This small modification makes American options much harder to price than their European counter-part [20]. There are no closed-form solutions for the American put option and therefore several numerical approaches are used when American put options are priced, see e.g. [21]. Meanwhile, as we shall see, the value of an American call option is the same as a European call option.

Consider an American put option, and let \( V \) denote the value of the option and consider what happens if \( V < \max(K-S, 0) \). Then, there is an arbitrage opportunity since one could buy the asset for \( S \) and the option for \( V \) and exercise the option which gives the profit \( K-S-V > 0 \). Hence, we must conclude that \( V \geq \max(K-S, 0) \). It
can actually be showed that for any option with payoff $\Phi$ that has the early exercise feature, it holds that $V \geq \Phi$, see [3].

For the European put option there exists domains where the price of the option is less than payoff, therefore the price of an American and European put option differs. In contrary, the price of a European call option is always larger than the payoff which can be seen from the Black-Scholes formula. Thus, for an American call option the price is the same as a European call option.

### 2.3.1 A Variational Inequality Problem

Let $\Phi(S)$ denote the payoff for an American put option. It is possible to prove that the value of an American put option satisfy the following variational inequality problem:

$$
\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV \leq 0 \quad \text{on } [0,T) \times \mathbb{R}^+,
$$

$$
V \geq \Phi \quad \text{on } [0,T) \times \mathbb{R}^+, \quad (2.13)
$$

$$
\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV\right)(V - \Phi) = 0 \quad \text{on } [0,T) \times \mathbb{R}^+, \quad (2.14)
$$

$$
V(T, S) = \Phi(S) \quad \text{on } \mathbb{R}^+. \quad (2.15)
$$

See [22] for a derivation of this result.

### 2.4 Asian Options

Asian options are any kind of option where the payoff is determined from some type of average. The two most common types of averages are arithmetic average and geometric average options. For geometric average options there are closed form solutions, and we are therefore not interested in those [23]. We shall instead focus on arithmetic average options and therefore refer to arithmetic average Asian options simply by Asian options.

There are two main types of Asian options, floating-strike and fixed-strike Asian options. The floating-strike put option has the payoff

$$
\max(A(T) - S, 0)
$$

where

$$
A(T) = \frac{1}{T} \int_0^T S(t) dt
$$

is the arithmetic average of $S$ over the period $[0,T]$. Similarly the floating-strike call option has the payoff $\max(S - A(T), 0)$.

The fixed-strike put option has the payoff

$$
\max(K - A(T), 0)
$$

where $K$ is the strike price, and $A$ is the arithmetic average as above. The corresponding call option is given by $\max(A(T) - K, 0)$. 

2.4.1 A Pricing PDE

To derive a pricing equation for Asian options we shall treat $S$ and $A$ as independent variables. This is valid since the value of $S$ does not depend on the history of $S$. Let $V(t, S, A)$ be the value of the option, treated as a function of $t, S$ and $A$. Our goal is to apply Itô’s Lemma (Equation (2.1)) on $V(t, S, A)$. For this we need the stochastic differential equation that $A$ satisfies. Let $dt$ be a small time step, then to first order it holds that

$$A(t + dt) = A(t) + dA(t) = \frac{1}{t + dt} \int_0^{t + dt} S(\tau)d\tau$$

$$= \frac{1}{t + dt} \int_0^t S(\tau)d\tau + \frac{S(t)}{t + dt}dt$$

$$= A(t) - A(t) dt + \frac{S(t)}{t}dt.$$

Thus,

$$dA = \frac{S - A}{t}dt.$$  

Applying the two-dimensional version of Itô’s Lemma (Equation (2.2)) on $V(t, S, A)$ yields that

$$dV = \sigma S \frac{\partial V}{\partial S}dW + \left( \alpha S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \frac{S - A}{t} \frac{\partial V}{\partial A} + \frac{\partial V}{\partial t} \right)dt. \tag{2.16}$$

Next, we set up a self-financing portfolio as before, $\Pi = V - \frac{\partial V}{\partial S}S$, where $\frac{\partial V}{\partial S}$ is held fixed during each time step, so

$$d\Pi = dV - \frac{\partial V}{\partial S}dS. \tag{2.17}$$

Inserting Equation (2.16) into (2.17) then yields that

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

So, $\Pi$ is risk-free and by Equation (2.5), it must hold that $d\Pi = r\Pi dt$, i.e.

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

and we conclude that $V$ satisfies the following PDE

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} + \frac{S - A}{t} \frac{\partial V}{\partial A} - rV = 0. \tag{2.18}$$

Equation (2.18) was first published by Jérôme Barraquand and Thierry Pudet in [14]. It is also possible to use the running sum $I(t) = \int_0^t S(\tau)d\tau$ as the other independent variable, and this leads to a slightly different PDE, but as noted in [15] one then requires a larger numerical domain to get accurate results.

We can directly note a couple of things regarding Equation (2.18). First, there is no diffusion term in the $A$-direction. This means that the PDE is convection dominant in that direction and can give rise to instabilities. The same is true in the $S$-direction for small $\sigma$. Also, the first order term in $A$ is singular at $t = 0$. These are numerical difficulties that needs to be dealt with when we are to solve the equation.
2.4. ASIAN OPTIONS

Similarity Reduction for floating-strike Options

Equation (2.18) has two spatial dimensions, this means that it will be more calculation heavy to solve than the Black-Scholes equation. For the fixed-strike Asian options, there is nothing we can do, but for the floating-strike it is possible to reduce the dimensionality by a change of variables.

Consider the floating-strike put option, the call option is treated the same way. The price of the floating-strike put option is given by

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} + \frac{S - A \frac{\partial V}{\partial A}}{t} - rV = 0$$ on $[0, T) \times \mathbb{R}^+ \times \mathbb{R}^+$,

$$V(T, S, A) = \max(S(T) - A(T), 0)$$ on $\mathbb{R}^+ \times \mathbb{R}^+$.

Consider the following change of variables

$$H = \frac{V}{S}, \quad R = \frac{A}{S}.$$

After some calculations one will reach the following PDE

$$\frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 R^2 \frac{\partial^2 H}{\partial R^2} - rR \frac{\partial H}{\partial R} + \frac{1 - R \frac{\partial H}{\partial R}}{t} = 0$$ on $[0, T) \times \mathbb{R}^+$, \hfill (2.19)

$$H(T, R) = \max(1 - R, 0)$$ on $\mathbb{R}^+$, \hfill (2.20)

see Appendix A.

Call-Put Parity

As for the European option, there is a relation between the call and put option for Asian options. We shall show how this can be derived later on when we come to options on a traded account since this gives a general framework where it is easier to derive the relation.

Let $C$ denote the value of the call option, and let $P$ denote the value of the put option. For the fixed-strike Asian option it holds that

$$C - P = \left(1 - e^{-r(T-t)}\right)\frac{1}{rT}S + e^{-r(T-t)}\left(\frac{t}{T}A - K\right).$$ \hfill (2.21)

For the floating-strike Asian option it holds that

$$P - C = \left(1 - e^{-r(T-t)}\right)\frac{1}{rT}S - S + e^{-r(T-t)}\frac{t}{T}A.$$

With the change of variables used for the floating-strike, this relation can be written as

$$H_P - H_C = \left(1 - e^{-r(T-t)}\right)\frac{1}{rT} - 1 + e^{-r(T-t)}\frac{t}{T}R,$$ \hfill (2.22)

where $H_P = P/S$ and $H_C = P/S$.

Boundary Conditions

Similarly as the Black-Scholes equation, to solve Equation (2.18) and (2.19) numerically we need to examine how the solution behaves in the limits, $S, A, R \to \pm\infty$. 
For Equation (2.18) we have four limits, \( S, A \to 0, \infty \). First, consider the put option. At \( S = 0 \), the equation reduces into the following PDE

\[
\frac{\partial V}{\partial t} - A \frac{\partial V}{\partial A} - rV = 0.
\]

At \( A = 0 \), the convection term in the \( A \)-direction becomes

\[
\frac{S}{t} \frac{\partial V}{\partial A}.
\]

Recall that the equation is a backward parabolic problem, therefore this is a transport term with an outward pointing velocity. This means that the boundary is a so called outflow boundary. Following [20], we do not specify any boundary conditions at the outflow.

Next, we study the limits \( A, S \to \infty \). In the numerical implementation we shall need to truncate the domain, so we introduce some positive real numbers \( S_\infty, A_\infty \) such that \( A_\infty, S_\infty \gg K, S_0 \). For the limit \( A \to \infty \), we have two cases to consider, \( S < A_\infty \) and \( S > A_\infty \). If \( S < A_\infty \), then the convection term is

\[
\frac{S - A_\infty}{t} \frac{\partial V}{\partial A}
\]

that is once again a transport term with an outward pointing velocity. Thus, we do not specify any boundary conditions. If \( S > A_\infty \), then by the same argument as for the European put option, the option will not be exercised if \( A \to \infty \). Hence, \( V = 0 \) at \( A = A_\infty \) when \( S > A_\infty \).

Finally, at \( S = S_\infty \) we follow [20] and argue that for large \( S \) the solution should approximately look like the payoff function, \( \max(K - A, 0) \), and since the payoff function does not depend on \( S \) we impose homogeneous Neumann boundary conditions \( \frac{\partial V}{\partial S} = 0 \).

The boundary conditions for the call option can now be achieved from the call-put parity. At \( A, S = 0 \), the boundary conditions becomes that same, so consider the case \( A = A_\infty \) when \( S > A_\infty \). There \( P = 0 \), so from Equation (2.21) we have that

\[
C = \left(1 - e^{-r(T-t)}\right) \frac{1}{rT} S + e^{-r(T-t)} \left(\frac{1}{T} A_\infty - K\right). \tag{2.23}
\]

At \( S = S_\infty \), we have that \( \frac{\partial P}{\partial S} = 0 \), thus by differentiating Equation (2.23) with respect to \( S \), we get that

\[
\frac{\partial C}{\partial S} = \left(1 - e^{-r(T-t)}\right) \frac{1}{rT}.
\]

The boundary conditions for Equation (2.19) are a bit easier to derive. We have two limits, \( R \to \pm \infty \). We consider the call option first. At \( R = 0 \), the convection term becomes

\[
\frac{1}{T} \frac{\partial H}{\partial R},
\]

which is a transport term with an outward pointing velocity, so we have an outflow boundary and as before we do not specify any boundary conditions. At \( R \to \infty \), the option will not be exercised, thus \( H \to 0 \).
2.4. ASIAN OPTIONS

Now, we can get the boundary conditions for the put option from the call-put parity (Equation (2.22)). At $R = 0$, we have an outflow as before, but when $R \to \infty$ the value of the call option approaches 0, so from Equation (2.22), the value of the put option in the limit $R \to \infty$ is given by

$$H = e^{-r(T-t)} \frac{t}{T} R + \left(1 - e^{-r(T-t)}\right) \frac{1}{rT} - 1.$$  

2.4.2 American-Style Asian Options

The price of an American-style Asian option can be found by solving a variational inequality problem, similar to how the American option is priced. The price $V$ for an American-style Asian option is given by the variational inequality problem

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + r S \frac{\partial V}{\partial S} + \frac{S - A \partial V}{t} - rV \leq 0$$

on $[0, T) \times \mathbb{R}^+ \times \mathbb{R}^+$,

$$V \geq \Phi$$

on $[0, T) \times \mathbb{R}^+ \times \mathbb{R}^+$,

$$(\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + r S \frac{\partial V}{\partial S} + \frac{S - A \partial V}{t} - rV)(V - \Phi) = 0$$

on $[0, T) \times \mathbb{R}^+ \times \mathbb{R}^+$,

$$V(T, S, A) = \Phi(S, A)$$

on $\mathbb{R}^+$.  

Here, $\Phi$ can be the payoff function for a fixed- or floating-strike call or put option.

For the floating-strike Asian option, we can use Equation (2.19) to price the option. Thus, for an American-style floating-strike option, it can priced by the following variational inequality problem

$$\frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 R^2 \frac{\partial^2 H}{\partial R^2} + r R \frac{\partial H}{\partial R} + \frac{1 - R \partial H}{t} \frac{\partial H}{\partial R} \leq 0$$

on $[0, T) \times \mathbb{R}^+$,

$$H \geq \Phi$$

on $[0, T) \times \mathbb{R}^+$,

$$(\frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 R^2 \frac{\partial^2 H}{\partial R^2} + r R \frac{\partial H}{\partial R} + \frac{1 - R \partial H}{t} \frac{\partial H}{\partial R})(H - \Phi) = 0$$

on $[0, T) \times \mathbb{R}^+$,

$$H(T, R) = \Phi(R)$$

on $\mathbb{R}^+$,  

where $\Phi(R) = \max(1 - R, 0)$ for a put option, and $\Phi(R) = \max(R - 1, 0)$ for a call option.

For a closer analysis and derivation of these results, see [3].

2.4.3 Options on a Traded Account

Options on a traded account, as we shall see, will give a new approach to price Asian options. It will lead to a PDE that is much simpler to solve than Equation (2.18) since it only has one spatial dimension and is not convection dominated. This approach were first considered by Jan Večeř in [7], and later he refined the approach in [8] to give one formulation that can both price fixed- and floating-strike options.

An option on a traded account is an option where the holder has the right to switch between a set of various positions of the underlying asset at any time as long as the option is active. The holder accumulates any profits or loses, and at expiration the holder receives a call option payoff with strike 0. To model an option on a traded account we let the asset $S$ evolve according to

$$dS = rSdt + \sigma SdW,$$
where \( r \) is the risk-free rate of interest, \( \sigma \) is the volatility and \( W \) is a Wiener process. We let \( q_t \) denote the number of shares held at time \( t \) of the underlying asset, and assume that \( q_t \in [\alpha_t, \beta_t] \subset \mathbb{R} \) where \( [\alpha_t, \beta_t] \) is the set of the different positions that the holder can switch between at time \( t \). A negative value of \( q_t \) corresponds to a short position. In [24], Shreve and Večeř proves that it is never optimal to hold an intermediate position, thus \( q_t \) should always equal \( \alpha_t \) or \( \beta_t \).

Next, we let \( X^q(t) \) denote the wealth at time \( t \) with the given strategy \( q_t \), we model the wealth with

\[
\begin{align*}
\frac{dX^q}{dt} &= q_t dS + r(X^q - q_t S)dt = rX^q dt + q_t(dS - rSdt), \\
X^q(0) &= X_0.
\end{align*}
\]

Here, \( X_0 \) is the initial wealth at \( t = 0 \). The payoff of this options is given by \( \max(X_T, 0) \), where \( X_T = X^q(T) \). As is showed in [24], the price of an option on a traded account satisfy the following Hamilton-Jacobi-Bellman equation

\[
0 = - rV + \frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \max_{q \in [\alpha, \beta]} \left( rX + \frac{1}{2} \sigma^2 S^2 \left( \frac{\partial^2 V}{\partial S^2} + 2q \frac{\partial^2 V}{\partial S \partial X} + q^2 \frac{\partial^2 V}{\partial X^2} \right) \right),
\]

with the terminal condition

\[
V(T, S, X) = \max(X, 0).
\]

We can reduce the dimensionality of Equation (2.25) by the following change of variables \( Z^q(t) = X^q(t)/S \). This leads to the following equation

\[
\frac{\partial u}{\partial t} + \max_{q \in [\alpha, \beta]} \frac{1}{2} (q - Z)^2 \sigma^2 \frac{\partial^2 u}{\partial Z^2} = 0
\]

with the terminal condition

\[
u(T, Z) = \max(Z, 0),
\]

see [24]. The price of the option at \( t = 0 \) is related to \( u \) by

\[
V(0, S_0, X_0) = S_0 u \left( 0, \frac{X_0}{S_0} \right),
\]

where \( S_0 \) is the initial value of the asset.

**Asian Options**

To price Asian options using options on a traded account the idea is to take a given strategy \( q_t \) that replicates the payoff of an Asian option. This is achieved by taking the strategy

\[
q_t = \frac{1}{rT} \left( 1 - e^{-r(T-t)} \right) - K_1,
\]

and we let the initial wealth be given by

\[
X_0 = \left( \frac{1}{rT} \left( 1 - e^{-rT} \right) - K_1 \right) S_0 - e^{-rT} K_2 = q_0 S_0 - e^{-rT} K_2.
\]
As we will see, if $K_1 = 0$ this will model fixed-strike Asian call options, and if $K_2 = 0$, then this will model floating-strike Asian put options. To show this, we need to determine $X_T$. Note that

$$X_T - e^{rT}X_0 = \int_0^T d(\sqrt{T-t}X) = \int_0^T e^{r(T-t)}dX - \int_0^T rXe^{r(T-t)}dt. \quad (2.29)$$

Next, we insert Equation (2.24) into (2.29), this yields that

$$X_T - e^{rT}X_0 = \int_0^T e^{r(T-t)}q_t(dS - rS dt)$$

$$= \int_0^T d\left(q_t e^{r(T-t)}S\right) - \int_0^T e^{r(T-t)}S \frac{du}{dt} dt$$

$$= q_T S_T - q_0 e^{rT}S_0 - \int_0^T e^{r(T-t)}S \frac{1}{rT} \left(-re^{-r(T-t)}\right) dt$$

$$= -K_1 S - e^{rT}X_0 - K_2 + \frac{1}{T} \int_0^T S dt.$$ 

Here, in the second step we used that

$$d\left(q_t e^{r(T-t)}S\right) = e^{r(T-t)}S \frac{du}{dt} dt = e^{r(T-t)}q_t(dS - rS dt).$$

Thus, we can conclude that

$$X_T = A - K_1 S - K_2 \quad (2.30)$$

and we see that if $K_1 = 0$ we get a fixed-strike Asian call option, and if $K_2 = 0$ we get a floating-strike Asian put option. The corresponding put/call options are derived by letting $q_t \rightarrow -q_t$ and $X_0 = q_0 S_0 + e^{-rT}K_2$.

Since our strategy $q_t$ is given, the maximum in Equation (2.26) is trivial and the value of an Asian option is given by $V(0, S_0, X_0) = S_0 u(0, Z_0)$, where the function $u$ satisfies

$$\frac{\partial u}{\partial t} + \frac{1}{2}(q - Z)^2 \sigma^2 \frac{\partial^2 u}{\partial Z^2} = 0 \quad \text{on } [0, T) \times \mathbb{R}, \quad (2.31)$$

$$u(T, Z) = \max(Z, 0) \quad \text{on } \mathbb{R},$$

and

$$Z_0 = \frac{X_0}{S_0} = q_0 - e^{-rT} \frac{K_2}{S_0}.$$ 

Compared to Equation (2.18) and (2.19), Equation (2.31) is much easier to handle numerically since it does not contain a convection term and is therefore unconditionally stable. Also, Equation (2.31) only has one spatial dimension and in a combined framework can price both fixed- and floating-strike options. Unfortunately as Večer notes in [8], this approach can not be used to price American-style Asian options since the strategy $q_t$ directly depends on the exercise time $T$.

We shall also make a remark that we can price European options using this approach. If we let $q_t \equiv 1$, and let the initial wealth be given by $X_0 = q_0 S_0 - e^{-rT}K = S_0 - e^{-rT}$, which is the same relation as before. Then, performing the same steps we did to derive Equation (2.30), we get that

$$X_T = S_T - K.$$
This is the payoff for a European call option. Thus, the value of a European call option is given by 
\[ S_0 u(0, Z_0), \]
where the function \( u \) satisfies
\[
\frac{\partial u}{\partial t} + \frac{1}{2} (1 - Z)^2 \sigma^2 \frac{\partial^2 u}{\partial Z^2} = 0 \quad \text{on } [0, T) \times \mathbb{R},
\]
and
\[ u(T, Z) = \max(Z, 0) \quad \text{on } \mathbb{R}. \]  

(2.32)

**Boundary Conditions**

As for the other pricing PDEs, we need to know how the solution behaves at the boundaries, for Equation (2.31) this is when \( Z \to \pm \infty \). As \( Z \to \infty \) it becomes more likely that \( Z \) remains larger than 0 until \( t = T \), thus \( u(t, Z) \to Z \) as \( Z \to \infty \). Similarly, if \( Z \to -\infty \) it will be more likely that \( Z \) remains smaller than 0 until \( t = T \), thus \( u(t, Z) \to 0 \) as \( Z \to -\infty \). This is also the boundary conditions that Večeř uses in [8] and as we will see it will give the correct results.

**Call-Put Parity for Asian Options**

As we noted before, the call-put parity for Asian options (Equation (2.21) and (2.22)) can more easily be derived with the framework of option on a traded account. In [24], Steven Shreve and Jan Večeř proves the following results that holds for all options on a traded account
\[
V^{[\alpha, \beta]}(t, S(t), X(t)) - V^{[-\beta, -\alpha]}(t, S(t), -X(t)) = X(t),
\]
where \( V^{[\alpha, \beta]} \) is the value of an option on a traded account with \( q_t \in [\alpha, \beta] \). With \( q_t \) given by Equation (2.27) and \( X(t) \) with initial wealth given by Equation (2.28), \( V^{[\alpha]}(t, S(t), X(t)) \) is the value of the fixed-strike call option or floating-strike put option, depending on the parameters \( K_1, K_2 \). Similarly, \( V^{[-\alpha]}(t, S(t), -X(t)) \) is the value of the fixed-strike put option or floating-strike call option. Therefore, we can get the call-put parity if have an expression for \( X(t) \). We can derive the expression for \( X(t) \) by following the steps we did to derive \( X_T \), but we integrate from 0 to \( t \) instead, where \( t \in [0, T] \). This yields that
\[
X(t) = \left(1 - e^{-r(T-t)}\right) \frac{1}{rT} S(t) - K_1 S(t) + e^{-r(T-t)} \left(\frac{t}{T} A - K_2\right).
\]

With \( K_1 = 0 \), we get Equation (2.21), and with \( K_2 = 0 \) we get Equation (2.22) after the variable transformation.
Chapter 3

Numerical Methods

3.1 The Finite Element Method

To derive the finite element methods that will be used to price the various option types introduced in Chapter 2, we shall first look at the weak formulation. Then, we shall introduce a finite subspace and derive the finite element methods. Finally, we shall look at the streamline-diffusion method that is used to prevent spurious oscillations that can appear in convection dominated problems, e.g. Equation (2.18) and (2.19).

3.1.1 Weak Formulations

All equations with one spatial dimension are on the following form

\[
\frac{\partial u}{\partial t} + \alpha(t, x) \frac{\partial^2 u}{\partial x^2} - \beta(t, x) \frac{\partial u}{\partial x} - \gamma(t, x) u = 0 \quad \text{on } J \times I, \\
u(T, x) = u_0(x) \quad \text{on } I,
\]

where \( \alpha, \beta, \gamma \) are some functions on \( J \times I \) such that \( \alpha \) is positive, \( I \) is some interval of \( \mathbb{R} \) and \( J = [0, T] \). This is a backward parabolic equation, but we can transform the equation into a forward equation by letting \( t \rightarrow T - t \). Then, we get the following equation

\[
\frac{\partial u}{\partial t} - \alpha(t, x) \frac{\partial^2 u}{\partial x^2} + \beta(t, x) \frac{\partial u}{\partial x} + \gamma(t, x) u = 0 \quad \text{on } J \times I, \\
u(0, x) = u_0(x) \quad \text{on } I,
\]

We shall use Equation (3.1) as our model problem for the one dimensional problems and derive the finite element methods for this problem. In the different pricing equations, the domain \( I \) is often infinite, but we shall restrict \( I \) to some compact interval to solve the equation numerically. Therefore, we shall assume that \( I = [a, b] \) for some \( -\infty < a < b < \infty \).

To formulate the weak form we shall need to introduce some function spaces. First, let \( V(I) \) be the following function space

\[
V(I) = \left\{ v \in L^2(I) : \frac{\partial v}{\partial x} \in L^2(I) \right\}.
\]
Also, let $L^2(J; V)$ denote the space of all square integrable functions on $J$ that takes value in $V$. Similarly, let

$$V_0(I) = \{ v \in V(I) : v(a) = v(b) = 0 \},$$

and let $L^2(J; V_0(I))$ denote the space of all square integrable functions on $J$ that takes value in $V_0(I)$.

We shall consider the case when we have Dirichlet boundary conditions at $x = a, b$, i.e. $u(t, a) = g_a(t)$ and $u(t, b) = g_b(t)$ for some functions of $g_a$ and $g_b$. Let $u \in L^2(J; V(I))$ such that $u|_{x=a} = g_a, u|_{x=b} = g_b$ and $v \in L^2(J; V_0(I))$, and multiply Equation (3.1) by $v$ and integrate over $J \times I$,

$$\int_J \int_I \left( \frac{\partial u}{\partial t} - \alpha(t, x) \frac{\partial^2 u}{\partial x^2} + \beta(t, x) \frac{\partial u}{\partial x} + \gamma(t, x)uv \right) dx dt = 0.$$

Next, we perform integration by parts on the second order term

$$\int_J \int_I \left( \frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \beta \frac{\partial u}{\partial x} v + \gamma uv \right) dx dt = 0.$$

The boundary term of the integration by parts vanish since $v$ is zero at the boundary. Hence, the weak formulation is: find $u \in L^2(J; V(I))$ such that $u|_{x=a} = g_a, u|_{x=b} = g_b,$ $u|_{t=0} = u_0(x)$ and

$$\int_J \int_I \left( \frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \beta \frac{\partial u}{\partial x} v + \gamma uv \right) dx dt = 0 \quad (3.2)$$

for all $v \in L^2(J; V_0(I))$.

We now look at the variational form for the different equations in more detail.

Black-Scholes Equation

In Black-Scholes equation (Equation (2.9)) we have that $\alpha = 1/2 \sigma^2 S^2, \beta = -r S$ and $\gamma = r$. If we choose the domain $I = [a, b]$, then for the put option the boundary conditions is $V(t, a) = Ke^{-rt}$ and $V(t, b) = 0$. Thus, the weak formulation for the European put option becomes: find $V \in L^2(J; V(I))$ such that $V(t, a) = Ke^{-rt}, V(t, b) = 0, V(0, S) = \max(K - S, 0)$, and

$$\int_J \int_I \left( \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial S} \frac{\partial V}{\partial S} + (\sigma^2 S - rS) \frac{\partial V}{\partial S} v + rv \right) dS dt = 0$$

for all $v \in L^2(J; V_0(I))$.

For the call option, we have instead that $V(t, a) = 0, V(t, b) = S - Ke^{-rt}$ and $V(0, S) = \max(S - K, 0)$.

The Floating-Strike Option Equation

For the Equation to price floating strike Asian options (Equation (2.19)), we have that $\alpha = \frac{1}{2} \sigma^2 R^2, \beta = rR - \frac{1}{2} \frac{\sigma^2 R^2}{R}$ and $\gamma = 0$. After the transformation to a forward parabolic equation, $\beta$ becomes $rR - \frac{1}{2} \frac{\sigma^2 R^2}{R}$. We let $I = [a, b]$ be the domain. For the floating strike call option, we have homogeneous Dirichlet boundary conditions
at \( R = b \). Thus, the weak formulation becomes: find \( H \in L^2(J;V(I)) \) such that
\[
V(t, b) = 0, \quad V(0, R) = \max(1 - R, 0), \quad \text{and}
\]
\[
\int J \left( \frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 R^2 \frac{\partial^2 H}{\partial R^2} + \left( \sigma^2 R + rR - \frac{1 - R}{T-t} \frac{\partial H}{\partial R} \right) \right) \, dR \, dt = 0
\]
for all \( v \in L^2(J;V(I)) \) such that \( v(t, b) = 0 \).

For the floating strike put option, we have instead that
\[
H(t, b) = e^{-\alpha t} T - t R + (1 - e^{-\alpha t}) \frac{1}{rT} - 1,
\]
and \( H(0, R) = \max(1 - R, 0) \).

**Večerš Equation**

In Večerš equation (Equation (2.31)) we only have the diffusion term that is given by \( \alpha = \frac{1}{2} \sigma^2 (q - Z)^2 \), where \( q = (1 - e^{-\alpha t}) \frac{1}{T} - K_1 \). As before, let \( I = [a, b] \) be the domain. At the boundary we have the Dirichlet boundary conditions \( u(t, a) = 0 \) and \( u(t, b) = b \). Thus, the weak formulation becomes: find \( u \in L^2(J;V(I)) \) such that \( u(t, a) = 0, u(t, b) = b, u(0, Z) = Z(0, Z, 0) \), and
\[
\int J \left( \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 (Z - q)^2 \frac{\partial^2 V}{\partial Z^2} + \sigma^2 (Z - q) \frac{\partial V}{\partial Z} \right) \, dZ \, dt = 0
\]
for all \( v \in L^2(J;V(I)) \).

**The Fixed-Strike Option Equation**

Now, we shall consider the fixed-strike pricing equation (Equation (2.18)). Let \( \Omega = [0, S_\infty] \times [0, A_\infty] \), be a subset of \( \mathbb{R}^2 \) such that \( S_\infty \geq A_\infty \). First we shall look at the fixed-strike call option, and also we transform the problem into a forward parabolic equation. Thus we have the problem
\[
\frac{\partial V}{\partial t} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rS \frac{\partial V}{\partial S} - \frac{S - A}{T-t} \frac{\partial V}{\partial A} + rV = 0 \quad \text{on } J \times \Omega, \quad (3.3)
\]
\[
V(0, S, A) = \max(A - K, 0) \quad \text{on } \Omega,
\]
\[
\frac{\partial V}{\partial S} = (1 - e^{-\alpha t}) \frac{1}{rT} \quad \text{on } J \times [0, A_\infty],
\]
\[
V(t, S, A_\infty) = (1 - e^{-\alpha t}) \frac{1}{rT} S + e^{-\alpha t} \left( \frac{T-t}{T} A_\infty - K \right) \quad \text{on } J \times [A_\infty, S_\infty].
\]

To state a weak formulation of Equation (3.3), we note that it can be written on the following form
\[
\frac{\partial V}{\partial t} - (D \nabla) \cdot \nabla V - \nu \cdot \nabla V + rV = 0, \quad (3.4)
\]
where \( \nabla = (\frac{\partial}{\partial S}, \frac{\partial}{\partial A}) \),
\[
D = \begin{pmatrix} \frac{1}{2} \sigma^2 S^2 & 0 \\ 0 & 0 \end{pmatrix},
\]
and \( \nu = \left( rS, \frac{S - A}{T-t} \right) \). Next, let
\[
v \in V_0(\Omega) = \{ v \in L^2(\Omega) : \nabla v \in L^2(\Omega), \quad v(S, A_\infty) = 0 \quad \text{for } S > A_\infty \}. \]
We multiply Equation (3.4) with $v$ and integrate over $J \times \Omega$,

$$\int_J \int_\Omega \left( \frac{\partial V}{\partial t} v - (D \nabla) \cdot (\nabla V) v - v \cdot (\nabla V) v + r V v \right) d\Omega dt = 0. \quad (3.5)$$

Next, we use that $v(D \nabla) \cdot \nabla V = \nabla \cdot (v D \nabla V) - (D \nabla V) \cdot \nabla v - v(\nabla \cdot D) \cdot \nabla V$

and the divergence theorem to write Equation (3.5) as

$$\int_J \int_\Omega \left( \frac{\partial V}{\partial t} v + (D \nabla V) \cdot \nabla v + (\nabla \cdot D - v) \cdot (\nabla V) v + r V v \right) d\Omega dt$$

$$- \int_J \int_{\partial \Omega} n \cdot (D \nabla V) v dldt = 0.$$

The boundary term is only non-zero at $S = S_\infty$ since at $A = 0, A_\infty$, $n$ and $D \nabla V$ are orthogonal, and at $S = 0$, $D$ is the zero matrix. At $S = S_\infty$, with the Neumann boundary conditions, the boundary term is

$$- \int_J \int_0^{A_\infty} (1 - e^{-rt}) \frac{1}{rT} vdAdt.$$

The weak formulation then becomes: find $V \in L^2(J; V(\Omega))$ such that

$$V(0, S, A) = \max(A - K, 0) \quad \text{on } \Omega,$$

$$V(t, S, A_\infty) = (1 - e^{-rt}) \frac{1}{rT} S + e^{-rt} \left( \frac{T - t}{T} A_\infty - K \right) \quad \text{on } J \times [A_\infty, S_\infty],$$

and

$$\int_J \int_\Omega \left( \frac{\partial V}{\partial t} v + (D \nabla V) \cdot \nabla v + (\nabla \cdot D - v) \cdot (\nabla V) v + r V v \right) d\Omega dt$$

$$= \int_J \int_0^{A_\infty} (1 - e^{-rt}) \frac{1}{rT} vdAdt \quad \text{for all } v \in V_0(\Omega).$$

For fixed-strike put option, we have homogeneous Neumann boundary conditions, hence we have no boundary term. Therefore, the weak formulation for the fixed-strike put option is: find $V \in L^2(J; V(\Omega))$ such that

$$V(0, S, A) = \max(K - A, 0) \quad \text{on } \Omega,$$

$$V(t, S, A_\infty) = 0 \quad \text{on } J \times [A_\infty, S_\infty],$$

$$\int_J \int_\Omega \left( \frac{\partial V}{\partial t} v + (D \nabla V) \cdot \nabla v + (\nabla \cdot D - v) \cdot (\nabla V) v + r V v \right) d\Omega dt = 0 \quad (3.7)$$

for all $v \in V_0(\Omega)$.

### 3.1.2 Finite Element Approximation

To derive the FEM formulation of the weak formulations from Section 3.1.1, we shall use the $cG(1)cG(1)$ method. That is, we seek our solution in the space of continuous piecewise linear functions in space and time. The acronym $cG$ stands for continuous Galerkin.
The cG(1)cG(1) method in 1D

Consider again the weak formulation of the model problem, Equation (3.2). Let \( a = x_0 < x_1 < \ldots < x_N = b \) be a division of \( I \) into \( N \) intervals with \( I_n = [x_{n-1}, x_n], \ n = 1, \ldots, N, \) and let \( 0 = t_0 < t_1 < \ldots < t_J = T \) be a division of \( J \) into \( J \) intervals with \( J_m = [t_{m-1}, t_m], \ m = 1, \ldots, M. \) We shall let \( \mathbb{P}^p(I) \) denote the set of all polynomials of degree \( p \) on \( I. \) Next, introduce the finite dimensional subspaces of \( V_h(I) \) of \( V(I) \) and \( V_{h,0}(I) \) of \( V_0(I) \) that are defined by

\[
V_h(I) = \{ v \in V(I) : \text{continuous and } v|_{I_n} \in \mathbb{P}^1(I_n), \ n = 1, \ldots, N \}
\]

and

\[
V_{h,0}(I) = \{ v \in V_0(I) : \text{continuous and } v|_{I_n} \in \mathbb{P}^1(I_n), \ n = 1, \ldots, N \}.
\]

Also, let

\[
W^1_h(I) = \{ v \in L^2(J; V_h(I)) : \text{continuous and } v|_{J_m} \in \mathbb{P}^1(J_m), \ m = 1, \ldots, M \}.
\]

Similarly, let

\[
W^0_{h,0}(I) = \{ v \in L^2(J; V_{h,0}(I)) : v|_{J_m} \in \mathbb{P}^0(J_m), \ m = 1, \ldots, M \}.
\]

The set \( W^0_{h,0}(I) \) consists of all functions that are constant on each \( J_m \) and takes values in \( V_{h,0}(I). \) Thus, elements of \( W^0_{h,0}(I) \) are not necessarily continuous at the boundary of the \( J_m \)’s.

To derive a linear system from the cG(1)cG(1) method, we seek the solution in \( W^0_{h,0}(I) \) while the test functions are in \( W^0_{h,0}(I). \) The finite element formulation is then: find \( u_h \in W^1_h(I) \) such that \( u_h(t, a) = g_u, \ u_h(t, b) = g_b, \ u_h(0, x) = u_0(x), \) and

\[
\sum_{m=1}^M \int_{J_m} \left( \frac{\partial u_h}{\partial t} v_h + \alpha \frac{\partial u_h}{\partial x} \frac{\partial v_h}{\partial x} + \beta \frac{\partial u_h}{\partial x} v_h + \gamma u_h v_h \right) dx dt = 0 \quad (3.8)
\]

for all \( v_h \in W^0_{h,0}(I). \) To make the notation easier, we introduce the bilinear forms

\[
(u, v)_I = \int_I uv dx
\]

and

\[
a_t(u, v) = \int_I \left( \alpha \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \beta \frac{\partial u}{\partial x} v + \gamma uv \right) dx,
\]

the subscript \( t \) is to note that \( a_t \) depends on \( t \) since the coefficients \( \alpha, \beta, \gamma \) could depend on \( t. \) Equation (3.8) can then be written as

\[
\sum_{m=1}^M \int_{J_m} \left( \frac{\partial u_h}{\partial t}, v_h \right)_I + a_t(u_h, v_h) dt = 0. \quad (3.9)
\]

To derive a linear system from Equation (3.9), we introduce a basis for \( V_h(I). \) The basis is the usual hat functions that can be seen in Figure 3.1. We have one basis function for each node point, hence the dimension of \( V_h(I) \) is \( N + 1. \) Let \( \varphi_i, \)
3.1. THE FINITE ELEMENT METHOD

\[ x_{k} \]

\[ x_{k-1} \]

\[ x_{k+1} \]

\[ x_{k+2} \]

Figure 3.1 – The figure shows how the hat functions look in 1D.

\[ i = 0, \ldots, N \] denote the basis functions for \( V_h(I) \). Then, for \( u_h \in W_h^0(I) \), we can on each \( J_m \) write

\[ u_h \bigg|_{J_m} = \sum_{i=0}^{N} \xi_i \left( \frac{t_m - t}{k_m} \right) \varphi_i + \eta_i \left( \frac{t - t_{m-1}}{k_m} \right) \varphi_i \quad (3.10) \]

where \( k_m = t_m - t_{m-1} \) is the length of \( J_m \), and \( \xi_i, \eta_i \) are some unknown real numbers. Similarly, each element of \( W_{h,0}^0(I) \) can on each \( J_m \) be written as a linear combination of the \( \varphi_i \) since they are constant. But, we require Equation (3.8) to hold for all \( v_h \in W_{h,0}^0(I) \) and since \( v_h \) is a linear combination of the \( \varphi_i \) on each \( J_m \), it is equivalent to that Equation (3.8) hold for each \( v_h \) on the form that is equal to \( \varphi_i \) on \( J_m \) and 0 otherwise. Thus, we can consider one term in Equation (3.8) and plug in \( v_h = \varphi_j \). This yields that

\[ \sum_{i=0}^{N} \int_{J_m} \left( \frac{\eta_i - \xi_i}{k_m} \varphi_i, \varphi_j \right) dt + a_t \left[ \xi_i \left( \frac{t_m - t}{k_m} \right) + \eta_i \left( \frac{t - t_{m-1}}{k_m} \right) \right] \varphi_i, \varphi_j = 0. \quad (3.11) \]

We can not perform the integration in time exactly since \( a_t \) depends on \( t \), instead we perform the integration in time by applying the midpoint rule. Then, we get that

\[ \sum_{i=0}^{N} \left( \frac{k_m}{2} \xi_i \varphi_i, \varphi_j \right) + \frac{k_m}{2} \left( \xi_i \varphi_i, \varphi_j \right) = 0, \quad (3.11) \]

where \( a_m \) is \( a_t \) evaluated at the midpoint \( (t_m + t_{m-1})/2 \). Equation (3.11) should hold for each \( m = 1, \ldots, M \) and each \( j = 0, \ldots, N \). If we let \( \xi^m = (\xi_0^m, \ldots, \xi_N^m), \eta^m = (\eta_0^m, \ldots, \eta_N^m) \), and introduce the matrices \( M \) and \( A^t \), where \( M_{ij} = (\varphi_i, \varphi_j)_I \) and \( A^t_{ij} = a_t(\varphi_i, \varphi_j) \), for short-hand we shall also take \( A^m_{ij} \) to mean \( A^t_{ij} \) evaluated at the midpoint of the interval \([t_{m-1}, t_m]\), then Equation (3.11) can be written as

\[ M(\eta^m - \xi^m) + \frac{k_m}{2} A^m (\xi^m + \eta^m) = 0. \quad (3.12) \]

Here we have \( 2(pN+1) \) unknowns but we only have \( pN+1 \) equations. To get \( pN+1 \) additional equations we impose the requirement that \( u_h \) should be continuous on
CHAPTER 3. NUMERICAL METHODS

Figure 3.2 – The figure shows an example of a hat function in 2D.

$J \times I$. Thus, we require that $u_h|_{J_{m-1}}(t_m) = u_h|_{J_m}(t_m)$. This yields that $\eta^{m-1} = \xi^m$ for $m = 1, \ldots, M$. For $m = 1$, $\eta^0$ is the coefficients of the projection of $u_0(x)$ into $V_h(I)$. Hence, we get the sought linear system that at step $m$ is given by

$$M(\eta^m - \eta^{m-1}) + \frac{k_m}{2} A^m (\eta^{m-1} + \eta^m) = 0,$$

or

$$\left( M + \frac{k_m}{2} A^m \right) \eta^m = \left( M - \frac{k_m}{2} A^m \right) \eta^{m-1}. \quad (3.13)$$

The Finite Element Method in 2D

To derive a finite element method for the pricing formula for fixed-strike Asian put and call options (Equation (3.6) and (3.7)), we first introduce a triangulation $K$ be a triangulation of $\Omega$ with $N_p$ number of points and $N_T$ number of triangles, and we let $0 = t_0 < t_1 < \cdots < t_M = T$ be a division of $J = [0, T]$ into $M$ intervals as before.

Let $V_h(\Omega)$ and $V_{h,0}(\Omega)$ be defined as

$$V_h(\Omega) = \{ v \in V(\Omega) : \text{continuous and } v|_K \in P^1(K), \forall K \in K \}.$$  

and

$$V_{h,0}(\Omega) = \{ v \in V_0(\Omega) : \text{continuous and } v|_K \in P^1(K), \forall K \in K \}.$$  

Next, we introduce a basis for $V_h(\Omega)$ where the basis functions are the hat functions in 2D. The hat function $i$, that we also shall denote by $\phi_i$, is defined by that it is 1 at point $p_i$ and 0 at point $p_j$ for $j \neq i$. A figure of such a hat function can be seen in Figure 3.2. Hence, the dimension of $V_h(\Omega)$ is equal to $N_p$. Now, let

$$\tilde{a}_t(u, v) = \int_{\Omega} \left[ (\nabla \cdot \mathbf{D} u) \cdot \nabla v + (\nabla \cdot \mathbf{D} - \mathbf{v}) \cdot (\nabla u) v + ruv \right] d\Omega.$$  

Hence, we can write Equation (3.6) as

$$\sum_{m=1}^M \int_{J_m} \left[ \left( \frac{\partial V}{\partial t} \right) \bigg|_\Omega \right] + \tilde{a}_t(V, v) \right] dt = \sum_{m=1}^M \int_{J_m} \left( \frac{1 - e^{-rt}}{rT} \right) [0, A_m] dt.$$
The cG(1)cG(1) method for the fixed-strike call option problem is then: find $V_h \in W_h^1(\Omega)$ such that

$$V_h(0, S, A) = P_h(\max(K - A, 0)) \quad \text{on } \Omega,$$

$$V_h(t, S, A) = P_h \left( (1 - e^{-rt}) \frac{1}{rT} S + e^{-rt} \left( \frac{T-t}{T} A - K \right) \right) \quad \text{on } \Omega \times [A_\infty, S_\infty],$$

(3.14)

and

$$\sum_{m=1}^{M} \int_{J_m} \left[ \frac{\partial V_h}{\partial t}, v_h \right]_{\Omega_t} + \tilde{a}_t(V_h, v_h) dt = \sum_{m=1}^{M} \int_{J_m} \left( \frac{1}{2} \sigma^2 S_\infty^2 \left( \frac{1 - e^{-rt}}{rT} \right), v_h \right)_{[0, A_\infty]} dt \quad (3.15)$$

for all $v_h \in W_h^0(\Omega)$. Here, $P_h : V(\Omega) \to V_h(\Omega)$ is the nodal interpolation, which is defined by that it takes a function and evaluates it at the nodes. Equation (3.15) has the same structure as Equation (3.9) except that the right-hand side is non-zero in Equation (3.15). Therefore, the resulting linear system will be similar, we just need to study the boundary term on the right-hand side. As before, let $v_h|_{J_m} = \varphi_j$ and 0 otherwise, then

$$\int_{J_m} \left( \frac{1}{2} \sigma^2 S_\infty^2 \left( \frac{1 - e^{-rt}}{rT} \right), \varphi_j \right)_{[0, A_\infty]} dt = \frac{1}{2} \sigma^2 S_\infty^2 \left( \frac{k_m}{rT} + \frac{e^{-rtm} - e^{-rtm-1}}{r^2 T} \right) (1, \varphi_j)_{[0, A_\infty]}.$$

So, if we let $\tilde{A}_j = \tilde{a}_t(\varphi_i, \varphi_j)$, $M_{ji} = (\varphi_i, \varphi_j)_{\Omega_t}$, and

$$b_{mj} = \frac{1}{2} \sigma^2 S_\infty^2 \left( \frac{k_m}{rT} + \frac{e^{-rtm} - e^{-rtm-1}}{r^2 T} \right) (1, \varphi_j)_{[0, A_\infty]},$$

then the linear system we get from Equation (3.15) is

$$\left( M + \frac{k_m}{2} \tilde{A}^m \right) \eta^m = \left( M - \frac{k_m}{2} \tilde{A}^m \right) \eta^{m-1} + b^m.$$

For the fixed-strike put option where there are no boundary term, we simply get

$$\left( M + \frac{k_m}{2} \tilde{A}^m \right) \eta^m = \left( M - \frac{k_m}{2} \tilde{A}^m \right) \eta^{m-1}.$$

### 3.1.3 Streamline-Diffusion Method

Equation (2.18) and (2.19) are both convection dominant which means that our standard finite element methods will produce spurious oscillations if the mesh size is too big [25]. So, one way to solve this is just to reduce the mesh size, but as in Equation (2.18) where we have no diffusion in the $A$-direction, we must solve this some other way. Over the years, several approaches have been developed to
solve this problem, see e.g. [26] and [27]. We shall consider the streamline-diffusion method. The idea with the streamline-diffusion method is to add a term that will introduce a numerical diffusion, thus increasing the stability. In the 2D case this is achieved by letting the test functions $v$ go over into $v + \delta(\frac{\partial v}{\partial t} + \beta \cdot \nabla v)$, where $\delta$ is some positive real number that has to be chosen. If we consider the cG(1)cG(1) method, then the test functions are constant in time, so the modification reduces to $v + \delta \beta \cdot \nabla v$. The streamline-diffusion method together with the cG(1)cG(1) method for Equation (2.18) then becomes: find $V_h \in W^1_h(\Omega)$ such that

$$V_h(0, S, A) = P_h(\max(K - A, 0)) \quad \text{on } \Omega,$$

$$V_h(t, S, A_{\infty}) = P_h\left((1 - e^{-rt})\frac{1}{rT}S + e^{-rt}\left(\frac{T}{T - t}A_{\infty} - K\right)\right) \quad \text{on } J \times [A_{\infty}, S_{\infty}],$$

$$\sum_{m=1}^{M} \int_{J_m} \left(\frac{\partial V_h}{\partial t}, v_h + \delta \beta \cdot \nabla v_h\right)_{\Omega} + \bar{a}_t(V_h, v_h + \delta \beta \cdot \nabla v_h) dt$$

$$= \sum_{m=1}^{M} \int_{J_m} \frac{1}{2} \sigma^2 S^2 \left(1 - e^{-rt}\right)(1, \varphi_j + \delta \beta \cdot \nabla \varphi_j)_{[0,A_{\infty}]} dt,$$

for all $v_h \in W^0_{h,0}(\Omega)$.\[28]

Let $A^m_{SD}$ be the matrix with $ji$:th coefficient given by

$$A^m_{SD,ji} = a_m(\phi_i, \phi_j + \delta \beta \cdot \nabla \phi_j),$$

let $M_{SD}$ be given by

$$M_{SD,ji} = (\phi_i, \phi_j + \delta \beta \cdot \nabla \phi_j)_{\Omega},$$

and let $b^m_{SD}$ be given by

$$b^m_{SD,ji} = \int_{J_m} \frac{1}{2} \sigma^2 S^2 \left(1 - e^{-rt}\right) (1, \varphi_j + \delta \beta \cdot \nabla \varphi_j)_{[0,A_{\infty}]} dt,$$

then the resulting linear system is given by

$$\left(M_{SD} + \frac{k_m^2}{2} A^m_{SD}\right) \eta^m = \left(M_{SD} - \frac{k_m^2}{2} A^m_{SD}\right) \eta^{m-1} + k_m b^m_{SD}.$$

Similarly, in the 1D case where we are interested in Equation (2.19), the SD method amounts to the substitution that the test functions $v$ go over into $v + \delta(\frac{\partial v}{\partial t} + \beta \frac{\partial v}{\partial R})$, which for the cG(1)cG(1) method reduces to $v + \delta \beta \frac{\partial v}{\partial R}$. Thus, if we define $A^m_{SD}$ by

$$A^m_{SD,ji} = a_m(\phi_i, \phi_j + \delta \beta \frac{\partial \phi_j}{\partial R})$$

and $M_{SD}$ by

$$M_{SD,ji} = (\phi_i, \phi_j + \delta \beta \frac{\partial \phi_j}{\partial R}),$$

then the corresponding linear system becomes

$$\left(M_{SD} + \frac{k_m^2}{2} A^m_{SD}\right) \eta^m = \left(M - \frac{k_m^2}{2} A^m_{SD}\right) \eta^{m-1}.$$
3.2 Numerical Methods for American-Style Options

For American-style options, we have to solve a variational inequality problem. If we discretize the problem using the cG(1)cG(1) method and proceed as before by expanding \( u_h \) as a linear combination of the hat functions and integrate over time and evaluate using the midpoint rule as before, we get the resulting system at each time step \( m \):

\[
\left( M + \frac{k_m}{2} A^m \right) \eta^m - \left( M - \frac{k_m}{2} A^m - b^m \right) \eta^{m-1} \geq 0, \quad (3.16)
\]

\[
\eta^m + \eta^{m-1} - 2 \mathcal{P} \Phi \geq 0, \quad (3.17)
\]

\[
\left( \eta^m + \eta^{m-1} - 2 \mathcal{P} \Phi \right)^T \left[ \left( M + \frac{k_m}{2} A^m \right) \eta^m \right] - \left( M - \frac{k_m}{2} A^m - b^m \right) \eta^{m-1} = 0. \quad (3.18)
\]

So, for each time-step \( m \) we have to solve this system. Note that the system given by Equation (3.16)-(3.18) is on the following form: find \( u \in \mathbb{R}^N \) such that

\[
Au - f \geq 0,
\]

\[
u - b \geq 0,
\]

\[(u - b)(Au - f) = 0,
\]

where \( b, f \in \mathbb{R}^N \), \( A \in \mathbb{R}^{N \times N} \) and \( N \) is the dimensionality of the problem. This is also called a linear complementary problem. We shall look at two different methods used to solve linear complementary problems, the projected successive over-relaxation method (PSOR) and the penalty method. We just note that there are also more advanced iterative solvers based on multigrid methods to solve linear complementary problems. We shall not consider these due to the complexity of their implementation, see e.g. [29] for a comparison of various methods.

3.2.1 The Projected Successive Over-Relaxation Method

The PSOR method is an iterative solver based on the successive over-relaxation method, and is used to solve linear complementary problems. Since it is an iterative solve, it requires a start guess. We let \( X^{(0)} \) denote the start guess, and let \( X^{(k)} \) be the \( k \)-th iteration. Then, iteration \( k + 1 \) is defined as follows: for each \( i = 1, \ldots, N \) solve for \( Y_i \) in

\[
A_{ii} Y_i + \omega \sum_{j<i} A_{ij} X_j^{(k+1)} = \omega f_i + (1 - \omega) A_{ii} X_i^{(k)} - \omega \sum_{j>i} A_{ij} X_j^{(k)}
\]

and calculate

\[
X_i^{(k+1)} = \max(Y_i, b_i),
\]

where \( 0 < \omega \leq 1 \) is a parameter that determines how fast the method will converge. The optimal value of \( \omega \) is problem dependent and can in most cases not be determined exactly, see [20] and [29] for a closer look at PSOR.
3.2.2 The Penalty Method

Another approach to solve the linear complementary problem is to use a penalty term. We shall adopt the penalty method as described in [29]. The penalty method for the linear complementary problem is defined as follows:

\[ Au = f + \frac{1}{\varepsilon} \max\{b - u, 0\}, \]

where \( \varepsilon > 0 \) is some small number to impose the constraint. This is problem is non-linear. To solve this system, we shall use a semi-smooth Newton method as described in [29]. Let \( u^{(0)} \) be the initial guess, then the \( k + 1 \)th iteration of the Newton method is given by

\[ u^{(k+1)} = u^{(k)} + d^{(k)}, \]

where \( d^{(k)} \) solves the linear system

\[ J(u^{(k)})d^{(k)} = f + \frac{1}{\varepsilon} \max\{b - u^{(k)}, 0\} - Au^{(k)}. \]

The \( ij \):th component of the matrix \( J(u^{(k)}) \) is defined by

\[ J(u^{(k)})_{ij} = A_{ij} + \begin{cases} \frac{1}{\varepsilon} & \text{if } i = j \text{ and } u_i^{(k)} < b_i, \\ 0 & \text{otherwise}. \end{cases} \]
Chapter 4
Implementations

All implementations for the numerical methods were done in Python 3, using NumPy for arrays and SciPy for linear algebra tools. All linear systems were solved using sparse matrices and we invoked the built-in function spsolve of SciPy to solve the systems. All calculations were performed on a computer equipped with an Intel® Xeon® E5-2630 2.30 GHz and 32 GB RAM.

4.1 The Finite Element Method

We can divide the finite element method in two steps. First we assemble the matrices $M$ and $A$, and the boundary vector if it is present. Then, we loop over all time steps and for each time step we solve the linear system. We shall first look at how this is done in 1D, then we shall look at the implementation in 2D.

4.1.1 FEM in 1D

To assemble the matrices in 1D we loop over all subintervals $I_n$ of $I$, and on each interval we calculate $A^{I_n}$ and $M^{I_n}$, which are the local stiffness and mass matrix respectively, defined by

$$A_{ji}^{I_n} = \int_{I_n} \left( \alpha \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \left( \frac{\partial \alpha}{\partial x} + \beta \right) \frac{\partial \varphi_i}{\partial x} \varphi_j + \gamma \varphi_i \varphi_j \right) dx,$$

and

$$M_{ji}^{I_n} = \int_{I_n} \varphi_i \varphi_j dx.$$  

Since only two hat functions are non-zero on each interval, the local matrices will only contain four non-zero entries. Thus, we take the local matrices to be precisely these $2 \times 2$ matrices. Due to the functions $\alpha$, $\beta$ and $\gamma$, we can not calculate the entries of $A^{I_n}$ exactly, instead we calculate them using Simpson's method. For $M^{I_n}$ we can calculate the entries exactly and it holds that

$$M^{I_n} = \frac{|I_n|}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},$$

31
where \(|I_n|\) is the length of \(I_n\). After the local matrices have been calculated we add them to the global ones. This is done by adding the \(ij\):th entry of the local matrix at interval \(I_n\) to the \(n-1+i\), \(n-1+j\):th entry of the corresponding global matrix.

We have one more thing to handle, and that is that the functions \(\alpha, \beta, \gamma\) are time-dependent. Therefore, the matrix \(A\) must be updated for each time step. This can be implemented in a couple of ways. In 1D the size of the matrices will not be too large and we can directly allocate the memory for all of the \(A\) matrices of the different time steps. Then, in the assembling we calculate the entries of \(A\) for all time steps at once. The assembling of the matrices in 1D is described in Algorithm 1.

**Algorithm 1 Assembly of matrices in 1D**

1: function Assembly1D\((x_n, t_n, \alpha, \beta, \gamma)\)
2: \(N \leftarrow \text{length}(x_n) - 1\)
3: \(J \leftarrow \text{length}(t_n) - 1\)
4: Allocate \((N + 1) \times (N + 1) \times J\) matrix \(A\)
5: Allocate \((N + 1) \times (N + 1)\) matrix \(M\)
6: for \(n = 1\) to \(N\) do
7: \(m \leftarrow (t_k + t_{k+1})/2\)
8: \(h \leftarrow x_n - x_{n-1}\)
9: Calculate the four entries of \(A^{I_n}\) at time \(t = m\)
10: \(A^k_{[n-1,n],[n-1,n]} \leftarrow A^k_{[n-1,n],[n-1,n]} + A^{I_n}\)
11: end for
12: \(M_{[n-1,n],[n-1,n]} \leftarrow M_{[n-1,n],[n-1,n]} + h \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\)
13: end for
14: return \(M, A\)
15: end function

Next, for the main method of the cG(1)cG(1) method we first assemble the stiffness and mass matrix, and then start to loop over all time steps, for each time step we solve the linear system given by Equation (3.13), see Algorithm 2. As can be seen in Algorithm 2, the Dirichlet boundary conditions \(u(t, a) = g_a\) and \(u(t, b) = g_b\) are implemented strongly. This means that before we solve the linear system we modify the rows corresponding to the nodes of each end point such that the solution exactly satisfy the Dirichlet boundary conditions.

### 4.1.2 FEM in 2D

In 2D, we have a triangulation of the domain. The triangulation is described by two matrices, the point matrix \(P\), and the connectivity matrix \(T\). The point matrix \(P\) is of size \(n_p \times 2\), where \(n_p\) is the number of points in the triangulation. So, row \(i\) of \(P\) is a tuple containing the coordinates for point \(i\). The connectivity matrix describes how the triangles are connected. It is of size \(n_t \times 3\), where \(n_t\) is the number of triangles. So, row \(i\) of \(T\) is three positive integers that corresponds to three indices of \(P\) that are the three points of triangle \(i\).

To assemble the stiffness and mass matrices in 2D, we loop over all triangles \(K\) of the triangulation. On each triangle, we calculate the local stiffness and mass
Algorithm 2 The cG(1)cG(1) method in 1D

1: function cG1cG1Solver1D($x_n$, $t_n$, $\alpha$, $\beta$, $\gamma$, $u_0$, $g_a$, $g_b$)
2:  $J \leftarrow \text{length}(t_n) - 1$
3:  $N \leftarrow \text{length}(x_n) - 1$
4:  $M, A \leftarrow \text{Assembly1D}(x_n, t_n, \alpha, \beta, \gamma)$
5:  $U \leftarrow u_0$
6:  for $n = 0$ to $J - 1$ do
7:     $k_n \leftarrow t_{n+1} - t_n$
8:     $M_{\text{LHS}} \leftarrow M + \frac{k_n}{2} A^n$
9:     $b \leftarrow (M - \frac{k_n}{2} A^n)U$
10:    if $g_a$ is defined then
11:        $M_{\text{LHS}}, \text{row}[0] \leftarrow 0$
12:        $M_{\text{LHS}}, 0, 0 \leftarrow 1$
13:        $b_0 \leftarrow g_a(t_{n+1})$
14:    end if
15:    if $g_b$ is defined then
16:        $M_{\text{LHS}}, \text{row}[N] \leftarrow 0$
17:        $M_{\text{LHS}}, N, N \leftarrow 1$
18:        $b_N \leftarrow g_b(t_{n+1})$
19:    end if
20:    Solve $M_{\text{LHS}} U = b$
21:  end for
22: return $U$
23: end function
matrices, \( A^K \) and \( M^K \) defined by

\[
A^K_{ji} = \int_K ((D\nabla \varphi_i) \cdot \nabla \varphi_j + (\nabla \cdot D) \cdot (\nabla \varphi_i) \varphi_j + \gamma \varphi_i \varphi_j) dx,
\]

and

\[
M^K_{ji} = \int_K \varphi_i \varphi_j dx.
\]

In 1D, the local matrices only had four non-zero elements. In 2D, there are nine since there are three hat functions that are non-zero on each triangle. Hence, we let the local matrices be these \( 3 \times 3 \) matrices. To describe how to add the local matrices to the global ones, consider a triangle that has the nodes with indices \( r, s, t \), and let \( l2g: [0, 1, 2] \rightarrow [r, s, t] \) be a map called the local to global map. We then add element \( ij \)-th of the local matrices to element \( l2g_i l2g_j \) of the global matrices.

As for the 1D case, we have to consider time dependent coefficients. For the fixed-strike Asian option, we have that

\[
\beta = \left(rS, \frac{S - A}{T - t}\right),
\]

meanwhile \( \alpha \) and \( \gamma \) does not depend on time. We could do as before and allocate and calculate the stiffness matrix for all time steps at once, but in 2D the size of the stiffness matrix is considerably larger. Therefore, when we refine the mesh it is no longer an option to allocate the memory for all time steps at once. Instead, we will split the stiffness matrix into a time-independent part \( \tilde{A} \) and a time-dependent part \( C \). The \( ji \)-th component of the time-independent part is given by

\[
\tilde{A}_{ji} = \int_\Omega rS \frac{\partial \varphi_i}{\partial A} \varphi_j dS dA,
\]

while the \( ji \)-th component of the time-dependent part is given by

\[
C_{ji} = \int_\Omega \frac{S - A}{T - t} \frac{\partial \varphi_i}{\partial A} \varphi_j dS dA.
\]

This can be factored as

\[
\frac{1}{T - t} \int_\Omega (S - A) \frac{\partial \varphi_i}{\partial A} \varphi_j dS dA,
\]

so when we assemble the matrices, the mass matrix and the time-independent part of the stiffness matrix can be assembled as usual. For the time-dependent part \( C \), we calculate only the time-independent part, and then in the main loop, we will multiply \( C \) by the factor \( 1/(T - t) \). The integrals of \( \tilde{A} \) and \( C \) must be evaluated numerically, and we apply Gaussian quadrature on triangles with three quadrature points to this end, see [25] for a definition. Meanwhile, the local mass matrix can be evaluated exactly, and it holds that

\[
M^K = \frac{|K|}{12} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},
\]

where \( |K| \) is the area of triangle \( K \) [25].
For the fixed-strike Asian call option, we also have the boundary vector $b$ that needs to be assembled which is given by

$$b^n_j = \frac{1}{2} \sigma^2 S^2 \left( \frac{k_m}{rT} + \frac{e^{-rt_m} - e^{-rt_{m-1}}}{r^2T} \right) \int_0^{A_\infty} \varphi_j dA.$$  

Note that $b$ is time-dependent, so it needs to be updated for each time-step, but as for the stiffness matrix, we can factor out the time-dependent part. Thus, we only assemble the time-independent part,

$$\int_0^{A_\infty} \varphi_j dA,$$

and then multiply with the time-dependent factor

$$\frac{1}{2} \sigma^2 S^2 \left( \frac{k_m}{rT} + \frac{e^{-rt_m} - e^{-rt_{m-1}}}{r^2T} \right)$$

in each time-step. To assemble the time-independent part, which we shall still call $b$ for convenience, we first extract all edges that lies on the boundary $S = S_\infty$, and put them into a matrix, which we shall call $E$. Hence, one row of $E$ consists of two indices that corresponds to the two points on one of the edges. Then, we loop over all edges $E_n$ in $E$ and on each edge we calculate the local component of $b^{E_n}$ given by

$$b_j^{E_n} = \int_{E_n} \varphi_j dA.$$

Note that only two hat functions are non-zero on each edge, thus only two elements of $b^{E_n}$ are non-zero. These can be evaluated exactly and the local expression for $b$ is given by

$$b^{E_n} = \frac{|E_n|}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

We add $b^{E_n}$ to the global vector $b$ the same way as we did with the local stiffness and mass matrices with a local to global map. In this case the map maps 0 to the index of the first point of the edge and 1 to the second point. The assembly of the mass- and stiffness matrix and the boundary vector $b$ is described in Algorithm 3.

The main method for the cG(1)cG(1) method in 2D is very similar to 1D, but the application of the Dirichlet boundary conditions is a bit more involved. Also, we need to extract the edge matrix $E$ for the Neumann boundary conditions.

The Dirichlet boundary conditions are applied strongly as in the 1D case. To achieve this we must extract all nodes on the Dirichlet boundary. For this we specify a boolean-valued function onDirichletBoundary that returns true if a point is on the Dirichlet boundary. The algorithm to extract all Dirichlet nodes is described in Algorithm 4. Then, for each node on the Dirichlet boundary we modify the corresponding row of the linear system such that the solution exactly satisfy the boundary condition.

To extract the edges on the Neumann boundary we also define a boolean-valued function onNeumannBoundary that returns true if a node lies on the boundary. Then, we loop over all triangles in $T$ and if two of the nodes on that triangle lies on the boundary, we append them to $E$. This is described in Algorithm 5.

Finally, the main method for 2D is described in Algorithm 6. The functions $g_D$ and $g_N$ are the Dirichlet and Neumann boundary values respectively.
Algorithm 3 Assembly of matrices and boundary vector in 2D

1: function \textsc{Assembly2D}(P, T, E, \alpha, \beta, \gamma) \\
2: \hspace{0.7cm} n_p \leftarrow \text{length}(P_{\text{column}[0]}) \quad \triangleright \text{Number of points} \\
3: \hspace{0.7cm} n_t \leftarrow \text{length}(T_{\text{column}[0]}) \quad \triangleright \text{Number of triangles} \\
4: \hspace{0.7cm} n_e \leftarrow \text{length}(E_{\text{column}[0]}) \quad \triangleright \text{Number of edges in } E \\
5: \hspace{0.7cm} \text{Allocate matrices } \tilde{A}, C, M \text{ of size } n_p \times n_p \\
6: \hspace{0.7cm} \text{Allocate vector } b \text{ of length } n_e \\
7: \hspace{1cm} \text{for } n = 0 \text{ to } n_e - 1 \text{ do} \\
8: \hspace{1.5cm} l2g \leftarrow E_n \\
9: \hspace{1.5cm} b^{E_n} \leftarrow \frac{|E_n|}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\
10: \hspace{1.5cm} \text{Add } b^{E_n}_j \text{ to } b_{l2g} \\
11: \hspace{1cm} \text{end for} \\
12: \hspace{1cm} \text{for } n = 0 \text{ to } n_t - 1 \text{ do} \\
13: \hspace{1.5cm} l2g \leftarrow \text{row}[n] \\
14: \hspace{1.5cm} \text{Calculate } \tilde{A}^{K_n} \text{ using Gaussian quadrature} \\
15: \hspace{1.5cm} \text{Calculate } C^{K_n} \text{ using Gaussian quadrature} \\
16: \hspace{1.5cm} M^{K_n} \leftarrow \frac{|K_n|}{12} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \\
17: \hspace{1.5cm} \text{Add } \tilde{A}^{K_n}_{ji} \text{ to } \tilde{A}_{l2g_j,l2g_i} \\
18: \hspace{1.5cm} \text{Add } C^{K_n}_{ji} \text{ to } C_{l2g_j,l2g_i} \\
19: \hspace{1.5cm} \text{Add } M^{K_n}_{ji} \text{ to } M_{l2g_j,l2g_i} \\
20: \hspace{1cm} \text{end for} \\
21: \hspace{0.7cm} \text{return } b, \tilde{A}, C, M \\
22: \text{end function}

Algorithm 4 Extract all nodes on the specified Dirichlet boundary.

1: function \textsc{ExtractNodes}(P, onDirichletBoundary) \\
2: \hspace{0.7cm} n_p \leftarrow \text{length}(P) \\
3: \hspace{0.7cm} \text{Allocate empty array dirichletNodes} \\
4: \hspace{1cm} \text{for } n = 0 \text{ to } n_p - 1 \text{ do} \\
5: \hspace{1.5cm} \text{if } \text{onDirichletBoundary}(P_n) \text{ then} \\
6: \hspace{1.5cm} \text{Append } n \text{ to dirichletNodes} \\
7: \hspace{1.4cm} \text{end if} \\
8: \hspace{1cm} \text{end for} \\
9: \hspace{0.7cm} \text{return } \text{dirichletNodes} \\
10: \text{end function}
Algorithm 5 Extracts all edges on the specified Neumann boundary.

1: function extractEdges(P, T, onNeumannBoundary)
2: \( n_t \leftarrow \text{length}(T) \)
3: Allocate empty matrix \( E \)
4: for \( n = 0 \) to \( n_t - 1 \) do
5: \( r, s, t \leftarrow T_{\text{row}[n]} \)
6: if onNeumannBoundary\((P_r) \) \& onNeumannBoundary\((P_s) \) then
7: Append array \([P_r, P_s]\) to \( E \)
8: end if
9: if onNeumannBoundary\((P_s) \) \& onNeumannBoundary\((P_t) \) then
10: Append array \([P_s, P_t]\) to \( E \)
11: end if
12: if onNeumannBoundary\((P_t) \) \& onNeumannBoundary\((P_r) \) then
13: Append array \([P_t, P_r]\) to \( E \)
14: end if
15: end for
16: return \( E \)
17: end function

Algorithm 6 The cG(1)cG(1) method in 2D specialized to the fixed-strike option pricing PDE

1: function cG1cG1Solver2D(P, T, \( t_n \), \( \alpha \), \( \beta \), \( \gamma \), \( u_0 \), onDirichletBoundary, \( g_D \), onNeumannBoundary, \( g_N \))
2: \( J \leftarrow \text{length}(t_n) - 1 \) \hspace{1cm} \( \triangleright \) Number of time-steps
3: \( E \leftarrow \text{extractEdges}(P, T, \text{onNeumannBoundary}) \)
4: \( b_N, A, C, M \leftarrow \text{Assembly2D}(P, T, E, \alpha, \beta, \gamma) \)
5: dirichletNodes \leftarrow \text{extractNodes}(P, \text{onDirichletBoundary})
6: \( U \leftarrow u_0 \)
7: for \( n = 0 \) to \( J - 1 \) do
8: \( k_n \leftarrow t_n - t_{n-1} \)
9: \( t_{\text{mid}} \leftarrow (t_n + t_{n-1})/2 \)
10: \( M_{\text{LHS}} \leftarrow M + \frac{k_n}{2} \left( A + \frac{1}{t_{\text{mid}}}C \right) \)
11: \( b \leftarrow k_n g_N(t_{\text{mid}}) b_N + \left( M - \frac{k_n}{2} \left( A + \frac{1}{t_{\text{mid}}}C \right) \right) U \)
12: \( M_{\text{LHS}, \text{row}[\text{dirichletNodes}]} = 0 \) \hspace{1cm} \( \triangleright \) Apply Dirichlet boundary conditions
13: \( M_{\text{LHS}, \text{dirichletNodes}, \text{dirichletNodes}} = 1 \)
14: \( b_{\text{dirichletNodes}} = g_D(\text{dirichletNodes}) \)
15: Solve \( M_{\text{LHS}} U = b \)
16: end for
17: return \( U \)
18: end function
4.1.3 Meshing

In both 1D and 2D we used a uniform mesh for the temporal dimension, meanwhile for the spatial domain we used non-uniform meshes. Since in our calculations we are only interested in one point which is the initial price \( S_0 \), for floating-strike options with the variable transformation this is \( R = 1 \), and for fixed-strike options it is the point \((S_0, S_0)\), the idea with the non-uniform meshes is to use a mesh that is refined around the relevant point. In 1D, we make take a simple approach to generate the non-uniform mesh. First, start with the point of interest, let's call it \( x \). Next, specify the interval \( I = [a, b] \) that is our domain and define \( h_{\text{min}} \) and \( h_{\text{max}} \) which is the minimum and maximum mesh sizes. Then, we add the points to the mesh that is obtained by adding \( h_{\text{min}} \) each time until we reached the point \( x + 0.05|I| \), i.e. we moved 5% of the length of \( I \). Next, we double the mesh size and add four new points. We keep doubling the mesh size until we reach \( h_{\text{max}} \), and we keep adding points until we reach the upper boundary \( b \). Then, we do the same in the other direction by adding the points obtained by subtracting \( h_{\text{min}} \) each time until we reach the point \( x - 0.05|I| \). Then, we continue as before by doubling the mesh size and adding four points.

![Figure 4.1](image)

**Figure 4.1** – The figure shows an example of a Delaunay triangulation.

In 2D, we use the built-in function Delaunay of SciPy to calculate a triangulation from a set of points. As is implied by the name, the function Delaunay uses Delaunay triangulation to create the triangulation. The Delaunay triangulation can be described by that no circumcircles of a triangle contains another point of the set. An example of such a triangulation can be seen in Figure 4.1. To generate a non-uniform mesh for our fixed-strike pricing problem, we generate two arrays of points, one for each direction, by the same method we did in 1D. Then, we create the set of points for the triangulation by taking the Cartesian product of the two arrays. An example of such a triangulation for the fixed-strike pricing problem can be seen in Figure 4.2.
4.2 Streamline-Diffusion method

The implementation of the streamline-diffusion method amounts to assemble the modified mass and stiffness matrix, and the Neumann boundary vector for the fixed-strike American-style call option. The only thing to note is the choice of the parameter $\delta$. We choose $\delta$ according to [28], so we let $\delta = \tilde{C} h$ if $h > |\alpha|$ and $\delta = 0$ else, where $\tilde{C}$ is some constant and $h$ is the mesh size. In 1D the mesh size is just the length of each interval, while in 2D the mesh size is the diameter of the triangles. We define $\delta$ locally such that in the assembly, we evaluate $h > |\alpha|$ for each interval/triangle, and if true we set $\delta = \tilde{C} h$ and if false we set $\delta = 0$. In accordance to [25], the unknown constant $\tilde{C}$ is chosen to $\frac{1}{\max_{x \in \Omega} \|\beta(t, x)\|}$ at each time step.

4.3 PSOR

The PSOR method is described in Algorithm 7. As seen in Algorithm 7, we repeat the iterations until the specified tolerance TOL is met. To use the method, we also need to specify the parameter $\omega$ and the initial guess $U$. For the initial guess, we just take the value of $U$ from the last time step, and for the first time step we take the initial value $u_0$ as our initial guess.

Regarding $\omega$, the choice of $\omega$ will determine how fast the method will converge and the optimal choice depend on the size and eigenvalues of the matrix $A$ [29]. It is not feasible to calculate the optimal value for $\omega$ for each iteration and not even possible in most cases. Therefore, we will start with an initial value of $\omega$, and we will update $\omega$ next time we call the PSOR method such that as long as the number of iterations required for the PSOR method to converge decreases, the value of $\omega$ is increased with some fixed step $\Delta \omega$. If the number of iterations start to increase, we switch sign of $\Delta \omega$ such that the value of $\omega$ is lowered. For our implementation we have set the initial $\omega = 0.7$ and the step size $\Delta \omega = 0.05$. 

Figure 4.2 – The figure shows an example of a mesh used in the pricing of fixed-strike Asian options.
Algorithm 7 The PSOR method.

1: function PSOR(U, A, f, b, ω, TOL)
2:     N ← len(U)
3:     repeat
4:         Uprevious ← U
5:         U0 ← max\left(\left(ωf0 + (1 − ω)A0,0U0 − ω\sum_{j=1}^{N-1} A_{0,j}U_j\right) / A_{0,0}, b_0\right)
6:     for i = 1 to N − 1 do
7:         Ui ← max\left(\left(ωf_i + (1 − ω)A_{i,i}U_i − ω\sum_{j=i+1}^{N-1} A_{i,j}U_j\right)
8:                 −ω\sum_{j=0}^{i-1} A_{i,j}U_j\right) / A_{i,i}, b_i\right)
9:     end for
10:     until ∥U − Uprevious∥ < TOL
11:     return U
12: end function

4.4 The Penalty Method

The implementation of the penalty method as presented in [29] is described in Algorithm 8. To use the method we need to specify an initial guess U and set the parameter ε. For the initial guess, we choose as with the PSOR method, the previous value of U from the last time step and for the first time step we take u_0 as our initial guess. According to [29] a good choice for ε is to set it equal to the tolerance TOL. In our implementation we have set ε = TOL = 10^{-4}.

Algorithm 8 The penalty method together with a semi-smooth Newton method to solve the non-linear system

1: function PENALTYMETHOD(U, A, f, b, ε, TOL)
2:     res(x) ← f + \frac{1}{2} \max(b − x, 0) − Ax
3:     Define matrix-valued function J with the ij:th component given by
4:         J_{ij}(x) ← A_{ij} + \frac{1}{2} \begin{cases} 1 & \text{if } i = j \text{ and } x_i < b_i, \\ 0 & \text{otherwise.} \end{cases}
5:     repeat \quad \triangleright \text{Newton iteration}
6:         Uprevious ← U
7:         U ← Uprevious + J(Uprevious)^{-1}\text{res}(Uprevious)
8:     until ∥U − Uprevious∥ < TOL
9:     return U
10: end function
Chapter 5

Results

The value of the options were calculated with parameters chosen such that they could be compared to earlier results obtained by others. The main reference that the results are compared to are [15], where the authors, Robert Zvan, Peter Forsyth and Kenneth Vetzal, looked at the pricing of floating- and fixed-strike Asian options, of both European- and American-style, using the finite difference method combined with a Van Leer flux limiter. In the presentation of the results Zvan will refer to the results obtained in [15]. We also compare to [14], where the authors, Jérôme Barraquand and Thierry Pudet, studies floating- and fixed-strike Asian options of both European- and American style. They apply a method they call forward shooting grid to solve the PDEs. We let B & P denote the results obtained in [14]. For the calculations of the common American put option, we compare with the results obtained by Zvan, Forsyth and Vetzal in [15] and also with results obtained by Robert Geske and Kuldeep Shastri in [30] using the binomial method. The results obtained in [30] shall be denoted by binomial.

5.1 European-Style Options

To check the validity of the implementation, the implementation of the cG(1)cG(1) method was used to calculate the price for European call options where there exists an analytical solution. Table 5.1 contains the results. In Table 5.1, column B-S contains the results obtained by solving Black-Scholes equation (Equation (2.9)) using the cG(1)cG(1) method, Večeř contains the results obtained by solving Equation (2.32), and analytic contains the analytical results from Black-Scholes formula (Equation (2.11)). The Black-Scholes equation was solved on the interval $[1/10, 1000]$ using a non-uniform mesh as described in Section 4.1.3 with $h_{\text{min}} = 2.5$ and $h_{\text{max}} = 50$ resulting in 75 grid points and a uniform mesh in time with 400 temporal points. Equation (2.32) was solved on the interval $[-1, 1]$, using a non-uniform mesh with $h_{\text{min}} = 0.05$ and $h_{\text{max}} = 0.1$ resulting in 79 grid points and a uniform mesh in time with 400 temporal points. As can be seen from the data, both solving Black-Scholes and Večeřs equation gives values close to the analytic pricing values. Večeřs equations seems to give a better estimation of the price for smaller $\sigma$ while Black-Scholes equation yields better estimation at larger $\sigma$ values, but overall it seems like the error of Večeřs equation does not change much for the various computations. That Black-Scholes equation is less exact for small $\sigma$ is expected since the diffusion
42 5.1. EUROPEAN-STYLE OPTIONS

Table 5.1 – Results for European call options with \( r = 0.1, T = 1 \) and \( S_0 = 100 \). B-S contains the results obtained from solving Black-Scholes equation using the \( cG(1)cG(1) \) method, Večeř contains the results obtained by pricing the options using options on a traded account with the \( cG(1)cG(1) \) method, and analytic refers to the analytic solution of Black and Scholes.

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( K )</th>
<th>B-S Value</th>
<th>Rel. err. (%)</th>
<th>Večeř Value</th>
<th>Rel. err. (%)</th>
<th>Analytic</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>95</td>
<td>14.295</td>
<td>0.063</td>
<td>14.303</td>
<td>0.007</td>
<td>14.304</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>10.297</td>
<td>0.107</td>
<td>10.307</td>
<td>0.010</td>
<td>10.308</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>6.872</td>
<td>0.160</td>
<td>6.882</td>
<td>0.015</td>
<td>6.883</td>
</tr>
<tr>
<td>0.2</td>
<td>95</td>
<td>16.434</td>
<td>0.030</td>
<td>16.437</td>
<td>0.012</td>
<td>16.439</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>13.265</td>
<td>0.038</td>
<td>13.267</td>
<td>0.023</td>
<td>13.270</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>10.510</td>
<td>0.048</td>
<td>10.513</td>
<td>0.019</td>
<td>10.515</td>
</tr>
<tr>
<td>0.4</td>
<td>95</td>
<td>22.783</td>
<td>0.009</td>
<td>22.780</td>
<td>0.022</td>
<td>22.785</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>20.316</td>
<td>0.010</td>
<td>20.313</td>
<td>0.025</td>
<td>20.318</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>18.074</td>
<td>0.017</td>
<td>18.071</td>
<td>0.033</td>
<td>18.077</td>
</tr>
</tbody>
</table>

term in Black-Scholes equation becomes small compared to the convection term for small \( \sigma \), therefore the problem becomes convection dominated and the solution can become unstable.

Table 5.2 contains the results for the floating-strike Asian put option. Here, 1D contains the results obtained by solving Equation (2.19) using the \( cG(1)cG(1) \) method together with the SD method on the interval \([0, 1]\) with a non-uniform spatial mesh with \( h_{\min} = 0.05 \) and \( h_{\max} = 0.1 \) resulting in 78 grid points, and a uniform mesh in time with 400 temporal points. Večeř contains the results obtained by solving Equation (2.31) using the \( cG(1)cG(1) \) method on the interval \([-1, 1]\) with a non-uniform spatial mesh with \( h_{\min} = 0.05 \) and \( h_{\max} = 0.1 \) resulting in 79 grid points, and a uniform mesh in time with 400 temporal points. The column labelled Lower contains the analytic lower bound that is known to be close to the true value obtained by Rogers and Shi in [31]. Comparing the results obtained by solving Equation (2.19) and Večeř’s equation, we can see that both methods give values close to each other, only differing by 0.01, and are close to the lower bound. Only for the case with \( \sigma = 0.2 \) and \( T = 1.0 \), there seems like Equation (2.19) gives a value slightly closer to the lower bound than Večeř’s equation, but the values are still in close proximity to the bound. If we compare with the results obtained with Zvan, Forsyth and Vetzal, their results are in close agreement with our results, and if we compare to the results obtained by Barraquand and Pudet, our results are in better agreement with the lower bound.

In Table 5.3 are the results for fixed-strike Asian call options presented. Here, 2D contains the results obtained by solving Equation (2.18) using the \( cG(1)cG(1) \) method together with the SD method on the domain \([0, 160] \times [0, 140]\) with a non-uniform mesh with \( h_{\min} = 1 \) and \( h_{\max} = 10 \) resulting in 45 grid points in the \( S \)-direction and 41 grid points in the \( A \)-direction, and a uniform mesh in time with 100 temporal points. Večeř denotes as before the results obtained by solving Equation (2.31) and this was done on the same non-uniform mesh in space and uniform mesh in time. Lower refers to the analytic lower bound obtained by Rogers and Shi in [31]. Before looking at the results, one could suspect that solving Večeř’s equation should give more accurate results than solving Equation (2.18) since Equation (2.18)
CHAPTER 5. RESULTS

Table 5.2 – Results for floating-strike Asian put options with $r = 0.1$ and $S_0 = 100$. The column 1D correspond to solving the 1D pricing PDE for floating-strike options using the cG(1)cG(1) method with the SD method and Večeř corresponds to the results obtained by solving Večeř’s equation using the cG(1)cG(1) method. Zvan and B & P contains the results obtained in [15] and [14] respectively, and lower is the lower bound for floating-strike Asian options obtained by Rogers and Shi in [31].

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$T$</th>
<th>1D</th>
<th>Večeř</th>
<th>Zvan</th>
<th>B &amp; P</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.25</td>
<td>0.629</td>
<td>0.630</td>
<td>0.636</td>
<td>0.632</td>
<td>0.628</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.665</td>
<td>0.664</td>
<td>0.668</td>
<td>0.671</td>
<td>0.666</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.599</td>
<td>0.598</td>
<td>0.598</td>
<td>0.614</td>
<td>0.598</td>
</tr>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>1.718</td>
<td>1.719</td>
<td>1.719</td>
<td>1.724</td>
<td>1.714</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>2.124</td>
<td>2.123</td>
<td>2.123</td>
<td>2.135</td>
<td>2.147</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>2.456</td>
<td>2.449</td>
<td>2.449</td>
<td>2.498</td>
<td>2.457</td>
</tr>
<tr>
<td>0.4</td>
<td>0.25</td>
<td>3.971</td>
<td>3.972</td>
<td>3.971</td>
<td>3.981</td>
<td>3.971</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>5.241</td>
<td>5.242</td>
<td>5.244</td>
<td>5.260</td>
<td>5.242</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>6.678</td>
<td>6.675</td>
<td>6.678</td>
<td>6.785</td>
<td>6.674</td>
</tr>
</tbody>
</table>

is both two dimensional and convection dominated while Večeř’s equation is one dimensional and unconditionally stable. This is also what is seen in Table 5.3, the values obtained by solving Večeř’s equation are overall closer to the lower bound than those obtained by solving Equation (2.18). Especially we see that the results from solving Equation (2.18) differs more from the bound for smaller $T$ and $\sigma$ and larger $K$ values. We therefore also suspect that we will get the largest errors when solving fixed-strike American-style Asian options for these cases. If we compare with the results obtained by Zvan, Forsyth and Vetzal, they got values in agreement with those we obtained by solving Equation (2.18), they also have larger error for small $\sigma$ and $T$, and larger $K$ values as suspected, but their method seems to handle the extreme case with $\sigma = 0.1$, $T = 0.25$ and $K = 105$ better than we did. Zvan, Forsyth and Vetzal, used a non-uniform mesh on the same domain as we used and equal number of grid points, i.e. a $S \times A$ mesh with $45 \times 41$ grid points. We do not in detail how they generated their non-uniform mesh, but they also took the Cartesian product of two sets of points for the $S$ coordinate and $A$ coordinate, thus generating a mesh similar to ours. Thus, our approach to solve Equation (2.18) and theirs seems to give comparable results, except for the extreme cases where they got more accurate results. Looking at Barraquand and Pudet’s results, theirs are more accurate for $\sigma = 0.1$, while they differ more for $\sigma = 0.4$. At $\sigma = 0.2$, Barraquand and Pudet, Zvan, Forsyth and Vetzal, and our results obtained by solving Equation (2.18) yields comparable results.

In summary, our implementation seems to be valid since both solving Black-Scholes equation and Večeř’s equation yields values close to Black-Scholes formula. For floating-strike Asian put options, both solving Equation (2.19) and Večeř’s equation yields accurate results, and are in close agreement with the results obtained by Zvan, Forsyth and Vetzal. The results from the computations of fixed-strike Asian call options shows that solving Večeř’s equation is more accurate than solving Equation (2.18), which is also expected. Solving Equation (2.18) yields results that is comparable in accuracy with both the results obtained by Zvan, Forsyth and Vetzal, and those obtained by Barraquand and Pudet.
### 5.1. EUROPEAN-STYLE OPTIONS

Table 5.3 – Results for fixed-strike Asian options with \( r = 0.1 \) and \( S_0 = 100 \). The column 2D corresponds to solving the 2D pricing PDE for fixed-strike options using the cG(1)cG(1) method with the SD method. Zvan and B & P contains the results obtained in [15] and [14] respectively, and Lower is the lower bound for fixed-strike Asian options obtained by Rogers and Shi in [31].

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( T )</th>
<th>( K )</th>
<th>2D</th>
<th>Večer</th>
<th>Zvan</th>
<th>B &amp; P</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>95</td>
<td>0.1</td>
<td>6.138</td>
<td>6.118</td>
<td>6.133</td>
<td>6.132</td>
<td>6.118</td>
</tr>
<tr>
<td>0.25</td>
<td>100</td>
<td>1.892</td>
<td>1.854</td>
<td>1.793</td>
<td>1.869</td>
<td>1.851</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>105</td>
<td>0.207</td>
<td>0.149</td>
<td>0.162</td>
<td>0.151</td>
<td>0.148</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>95</td>
<td>7.247</td>
<td>7.219</td>
<td>7.244</td>
<td>7.248</td>
<td>7.220</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>3.112</td>
<td>3.070</td>
<td>3.052</td>
<td>3.100</td>
<td>3.104</td>
<td></td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>0.755</td>
<td>0.715</td>
<td>0.726</td>
<td>0.727</td>
<td>0.714</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>95</td>
<td>5.275</td>
<td>5.255</td>
<td>5.261</td>
<td>5.279</td>
<td>5.255</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>2.319</td>
<td>2.296</td>
<td>2.314</td>
<td>2.313</td>
<td>2.294</td>
<td></td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>2.958</td>
<td>2.933</td>
<td>2.928</td>
<td>2.960</td>
<td>2.932</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>4.509</td>
<td>4.504</td>
<td>4.511</td>
<td>4.548</td>
<td>4.505</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>2.213</td>
<td>2.206</td>
<td>2.229</td>
<td>2.241</td>
<td>2.211</td>
<td></td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>0.973</td>
<td>0.949</td>
<td>0.971</td>
<td>0.966</td>
<td>0.947</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>8.094</td>
<td>8.100</td>
<td>8.123</td>
<td>8.151</td>
<td>8.102</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>100</td>
<td>7.034</td>
<td>7.041</td>
<td>7.042</td>
<td>7.079</td>
<td>7.042</td>
<td></td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>4.502</td>
<td>4.508</td>
<td>4.519</td>
<td>4.539</td>
<td>4.509</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>1.0</td>
<td>10.287</td>
<td>10.293</td>
<td>10.309</td>
<td>10.336</td>
<td>10.295</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>7.894</td>
<td>7.891</td>
<td>7.921</td>
<td>7.793</td>
<td>7.891</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>4.509</td>
<td>4.504</td>
<td>4.511</td>
<td>4.548</td>
<td>4.505</td>
<td></td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>2.213</td>
<td>2.206</td>
<td>2.229</td>
<td>2.241</td>
<td>2.211</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>10.339</td>
<td>10.343</td>
<td>10.357</td>
<td>10.425</td>
<td>10.346</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>100</td>
<td>7.562</td>
<td>7.569</td>
<td>7.574</td>
<td>7.650</td>
<td>7.572</td>
<td></td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>5.362</td>
<td>5.371</td>
<td>5.384</td>
<td>5.444</td>
<td>5.371</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1.0</td>
<td>11.121</td>
<td>11.120</td>
<td>11.115</td>
<td>11.213</td>
<td>11.121</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>8.908</td>
<td>8.910</td>
<td>8.912</td>
<td>8.989</td>
<td>8.910</td>
<td></td>
</tr>
</tbody>
</table>
5.2 American-style Options

As we did with European-style options, we first looked at the simplest case, which is American put options, to see whether our implementations work as expected. The results can be seen in Table 5.4. Penalty refers to the results obtained by using the penalty method to solve Equations (2.13)-(2.15) and PSOR refers the results obtained by using PSOR. In both cases, the problem was solved on the domain $[1/10, 400]$ with a non-uniform mesh with $h_{\text{min}} = 1$ and $h_{\text{max}} = 20$ resulting in 75 grid points and a uniform mesh in time with 400 temporal points. We can see that both PSOR and the penalty method yields results in good agreement with those obtained by Zvan, Forsyth and Vetzal, and also those obtained by Geske and Shastri using the binomial method. Thus, both methods could possibly be a valid method to price American-style fixed- and floating-strike Asian options.

Table 5.5 contains all results regarding floating-strike American-style Asian options. Each option was solved using the penalty method and PSOR, together with the SD method on the domain $[0.01, 2]$ with three different non-uniform meshes, and a uniform mesh in time with 400 temporal points. The three meshes were defined with $h_{\text{min}} = 0.05, 0.02$ and $0.01$, and $h_{\text{max}} = 0.1$ for each case, resulting in 78, 150, and 257 grid points respectively. First, looking at the results from using PSOR, we can see that the values does not seem to converge, thus using PSOR does not seems to be a valid method. Looking at the results obtained by using the penalty method instead, we see that the values converges and are in agreement with those obtained by Zvan, Forsyth and Vetzal, and those obtained by Barraquand and Pudet. We can also note that the values obtained by Zvan, Forsyth and Vetzal, and Barraquand and Pudet are not in total agreement with each other and that the values we obtained using the penalty method with a minimum mesh size of 0.01 yields values that lies in between those values. We can also see that except for the case with $\sigma = 0.1$ and $T = 0.25$, our results using the penalty method with mesh size 0.05 yields values in good agreement with Zvan, Forsyth and Vetzal, and with Barraquand and Pudet.

Finally, Table 5.6 contains the results regarding fixed-strike American-style Asian options. The options were priced using the penalty method and PSOR, together with the SD method on the domain $[0, 160] \times [0, 140]$ with three different non-uniform meshes, and a uniform mesh in time with 100 temporal points. The three different

<table>
<thead>
<tr>
<th>$T$</th>
<th>$K$</th>
<th>Penalty</th>
<th>PSOR</th>
<th>Binomial</th>
<th>Zvan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/12</td>
<td>35</td>
<td>0.07</td>
<td>0.07</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1.31</td>
<td>1.31</td>
<td>1.31</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>5.05</td>
<td>5.06</td>
<td>5.06</td>
<td>5.06</td>
</tr>
<tr>
<td>1/3</td>
<td>35</td>
<td>0.69</td>
<td>0.69</td>
<td>0.70</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>2.47</td>
<td>2.48</td>
<td>2.48</td>
<td>2.47</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>5.70</td>
<td>5.70</td>
<td>5.71</td>
<td>5.70</td>
</tr>
<tr>
<td>7/12</td>
<td>35</td>
<td>1.21</td>
<td>1.21</td>
<td>1.22</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>3.16</td>
<td>3.16</td>
<td>3.17</td>
<td>3.16</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>6.23</td>
<td>6.23</td>
<td>6.24</td>
<td>6.23</td>
</tr>
</tbody>
</table>
Table 5.5 – Results for floating-strike American-style Asian put options with $r = 0.1$ and $S_0 = 100$. Penalty refers to the results obtained by using the penalty method together with the SD method, and PSOR refers to the results obtained by using PSOR together with the SD method. Zvan refers to the results obtained in [15] and B & P refers to the results obtained in [14].

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$T$</th>
<th>Penalty, $h_{\text{min}}$</th>
<th>$\text{PSOR, } h_{\text{min}}$</th>
<th>Zvan</th>
<th>B &amp; P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.05</td>
<td>0.02</td>
<td>0.01</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td>0.25</td>
<td>0.831</td>
<td>1.243</td>
<td>1.244</td>
<td>0.892</td>
<td>1.026</td>
</tr>
<tr>
<td>0.1</td>
<td>1.604</td>
<td>1.567</td>
<td>1.568</td>
<td>1.311</td>
<td>1.236</td>
</tr>
<tr>
<td>1.0</td>
<td>1.895</td>
<td>1.891</td>
<td>1.892</td>
<td>1.613</td>
<td>1.258</td>
</tr>
<tr>
<td>0.2</td>
<td>3.818</td>
<td>3.822</td>
<td>3.822</td>
<td>3.626</td>
<td>3.198</td>
</tr>
<tr>
<td>1.0</td>
<td>4.968</td>
<td>4.970</td>
<td>4.970</td>
<td>4.815</td>
<td>4.121</td>
</tr>
<tr>
<td>0.4</td>
<td>6.177</td>
<td>6.180</td>
<td>6.180</td>
<td>5.974</td>
<td>5.146</td>
</tr>
<tr>
<td>1.0</td>
<td>11.479</td>
<td>11.484</td>
<td>11.480</td>
<td>10.926</td>
<td>11.399</td>
</tr>
</tbody>
</table>

meshes were defined with $h_{\text{min}} = 1.0, 0.5$ and 0.3 respectively, and $h_{\text{max}} = 10$ for each case, resulting in $45 \times 41, 68 \times 62$, and $95 \times 87$ grid points respectively. The results using the PSOR are not presented in Table 5.6 since the computations did not converge and for most of the cases the result was not-a-number, i.e. the method diverged. Therefore, the PSOR method does not seem to work for price fixed-strike American style Asian options. For the penalty method, we can see that for $h_{\text{min}} = 0.3$, the values are in good agreement with those obtained by Zvan and B & P. We can however note that for the case $\sigma = 0.1, T = 0.25$ and $K = 105$, only the computation with $h_{\text{min}} = 0.3$ gave good results, and for $h_{\text{min}} = 0.5$, the value is even negative, which is unreasonable since it corresponds to a price. Overall we can see that with $h_{\text{min}} = 1.0$, the results do not agree with those of Zvan, Forsyth and Vetzal, or with Barraquand and Pudet, so with our construction of the non-uniform mesh, a mesh size of $h_{\text{min}} = 0.5$ is required, and with the most extreme case, an even finer mesh is needed. In the computations done by Zvan, Forsyth and Vetzal, they used a non-uniform mesh with 45 grid points in the $S$-direction and 41 grid points in the $A$-direction. This is also the number of grid points used in the computations with $h_{\text{min}} = 1.0$. Thus, based on this comparison, Zvan, Forsyth and Vetzal’s method by using the Van Leer flux limiter give a method that converges faster and does not require as a fine mesh to give accurate results.

In conclusion, PSOR works for American put options, but for American-style Asian options PSOR does not seem to work since the computations gave results that did not converge. The penalty method worked to price all three option types we looked at. For floating-strike American-style Asian put options, the results obtained where in good agreement with those obtained by Zvan, Forsyth and Vetzal, and those obtained by Barraquand and Pudet. However, we required a minimum mesh size of 0.01 to get accurate results. For fixed-strike American-style Asian options the penalty method converges but it requires a mesh that is finer than the one used by Zvan, Forsyth and Vetzal to get accurate results. Therefore, their method seems more accurate than ours.
Table 5.6 – Results for fixed-strike American-style Asian call options with \( r = 0.1 \) and \( S_0 = 100 \). Penalty refers to the results obtained by using the penalty method together with the SD method. Zvan refers to the results obtained in [15] and B & P refers to the results obtained in [14].

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>T</th>
<th>( K )</th>
<th>Penalty, ( h_{\text{min}} )</th>
<th>Zvan</th>
<th>B &amp; P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>95</td>
<td>6.107</td>
<td>6.528</td>
<td>6.554</td>
<td>6.646</td>
</tr>
<tr>
<td>0.25</td>
<td>100</td>
<td>3.065</td>
<td>2.053</td>
<td>1.948</td>
<td>1.903</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>0.366</td>
<td>-0.048</td>
<td>0.167</td>
<td>0.161</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>8.238</td>
<td>7.560</td>
<td>7.628</td>
<td>7.687</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>100</td>
<td>2.751</td>
<td>3.041</td>
<td>3.118</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>1.780</td>
<td>0.715</td>
<td>0.717</td>
<td>0.733</td>
</tr>
<tr>
<td>1.0</td>
<td>100</td>
<td>5.507</td>
<td>5.288</td>
<td>5.390</td>
<td>5.398</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>2.919</td>
<td>2.136</td>
<td>2.285</td>
<td>2.340</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>7.785</td>
<td>7.421</td>
<td>7.453</td>
<td>7.521</td>
</tr>
<tr>
<td>0.25</td>
<td>100</td>
<td>2.968</td>
<td>3.118</td>
<td>3.200</td>
<td>3.224</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>1.093</td>
<td>1.076</td>
<td>0.977</td>
<td>1.009</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>8.703</td>
<td>8.976</td>
<td>8.876</td>
<td>8.908</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>100</td>
<td>4.850</td>
<td>4.929</td>
<td>4.879</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>2.527</td>
<td>2.109</td>
<td>2.265</td>
<td>2.337</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>11.595</td>
<td>11.200</td>
<td>11.264</td>
<td>11.295</td>
</tr>
<tr>
<td>1.0</td>
<td>100</td>
<td>7.602</td>
<td>7.424</td>
<td>7.504</td>
<td>7.548</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>4.838</td>
<td>4.722</td>
<td>4.611</td>
<td>4.742</td>
</tr>
<tr>
<td>0.25</td>
<td>100</td>
<td>6.006</td>
<td>5.798</td>
<td>5.808</td>
<td>5.846</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>11.940</td>
<td>11.922</td>
<td>11.984</td>
<td>11.997</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>100</td>
<td>8.810</td>
<td>8.436</td>
<td>8.479</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>5.745</td>
<td>5.685</td>
<td>5.769</td>
<td>5.899</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>16.240</td>
<td>15.634</td>
<td>15.741</td>
<td>15.749</td>
</tr>
<tr>
<td>1.0</td>
<td>100</td>
<td>12.948</td>
<td>12.325</td>
<td>12.417</td>
<td>12.497</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusions

From our calculations we found as expected that for European-style Asian options one get more accurate and more stable results by solving the equation based on options on a traded account than solving the PDEs based on the derivation of Black-Scholes equation. By using a non-uniform mesh we also achieved accurate results in a relative short time. Hence, we conclude that FEM provides a robust and accurate method to price European-style Asian options.

Pricing American-style Asian options using PSOR did not give accurate values, and for most cases the computations diverged, but using the penalty method did give accurate results both for floating- and fixed strike options. For the floating-strike American-style Asian options, we achieved accurate results from a relative short computation time, but for fixed-strike American-style Asian options we required a relative fine mesh to get accurate results, and therefore a long computation time. Compared to previous work, among others that of Zvan, Forsyth and Vetzal in [15] who used the finite difference method together with a Van Leer flux limiter, they achieved accurate results with a courser mesh.

Even though Zvan, Forsyth and Vetzals approach based of the finite difference method seems to be more accurate, using FEM still looks promising. This is mainly due to that it is easier from a FEM solution to get good approximations of the different measures of sensitivities, called the Greeks. But, to get an efficient FEM solver to price American-style Asian options more research is needed. For future research, the next step would be to calculate the Greeks and look at the accuracy and stability of these values. We suspect that our current approach is not enough to yield accurate values for the Greeks, but we propose a couple of possible refinement that could increase the accuracy. We have applied a simple FEM method, cG(1)cG(1). This could be improved by using more specialized basis functions, e.g. one could try using Hermitian splines in order increase the regularity of the solution. We have tried two different method to solve the linear complementary problem that arise from an American-style option pricing problem, one could investigate more advanced methods, some are described in [29]. There is also another possibility that we have yet to investigate and that is to use a more specialized mesh to the problem at hand. Possibly by using an adaptive finite element method to generate the mesh. This is explored for European-style Asian options using Večeř’s equation in [5], but it would be interesting to explore this possibilities also for American-style Asian options.
Bibliography


Appendix A

Change of Variables for Floating Strike Asian Option

Here, we shall show how to transform
\[ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} + \frac{S - A \partial V}{t - \partial A} - rV = 0 \]  
(A.1)

into
\[ \frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 R^2 \frac{\partial^2 H}{\partial R^2} - rR \frac{\partial H}{\partial R} + \frac{1}{t} - R \frac{\partial H}{\partial R} = 0 \]  
(A.2)

by the change of variables defined by
\[ H = \frac{V}{S}, \quad \text{and} \quad R = \frac{A}{S}. \]

First, using the chain rule we express the partial derivatives of \( H \) with respect to \( S \) and \( A \) in terms of \( \frac{\partial H}{\partial R} \),
\[ \frac{\partial H}{\partial S} = \frac{\partial H}{\partial R} \frac{\partial R}{\partial S} = \frac{\partial H}{\partial R} \left( \frac{A}{S} \right) = -A \frac{\partial H}{S^2} \frac{\partial R}{S} = -R \frac{\partial H}{S R}, \]
\[ \frac{\partial^2 H}{\partial S^2} = \frac{\partial}{\partial S} \left( -A \frac{\partial H}{S^2} \frac{\partial R}{S} \right) = 2A \frac{\partial H}{S^3} \frac{\partial R}{S} - A \frac{\partial R}{S^2} \frac{\partial^2 H}{S^2} \frac{\partial R}{S} = 2R \frac{\partial H}{S^2} \frac{\partial R}{S} + R^2 \frac{\partial^2 H}{S^2} \frac{\partial R}{S^2}, \]
and
\[ \frac{\partial H}{\partial A} = \frac{\partial H}{\partial R} \frac{\partial R}{\partial A} = \frac{\partial A}{S} \frac{\partial H}{\partial R} \frac{\partial R}{S} = \frac{1}{S} \frac{\partial H}{R}. \]

Next, we evaluate the first order partial derivatives of \( V \) in terms of \( H \) the partial derivatives of \( H \) with respect of \( t \) and \( R \). It holds that
\[ \frac{\partial V}{\partial t} = \frac{\partial}{\partial t} (SH) = S \frac{\partial H}{\partial t}, \]
\[ \frac{\partial V}{\partial S} = \frac{\partial}{\partial S} (SH) = H + S \frac{\partial H}{\partial S} = H - S \frac{R \partial H}{S \partial R} = H - R \frac{\partial H}{\partial R}. \]
\[
\frac{\partial V}{\partial A} = \frac{\partial}{\partial A} (SH) = S \frac{\partial H}{\partial A} = S \frac{1}{S} \frac{\partial H}{\partial R} = \frac{\partial H}{\partial R}.
\]

We also need the second derivative of \(V\) with respect to \(S\),

\[
\frac{\partial^2 V}{\partial S^2} = \frac{\partial}{\partial S} \left( H + S \frac{\partial H}{\partial S} \right) = 2 \frac{\partial H}{\partial S} + S \frac{\partial^2 H}{\partial S^2} = -2 \frac{R}{S} \frac{\partial H}{\partial R} + 2 \frac{R}{S} \frac{\partial H}{\partial R} + \frac{R^2}{S} \frac{\partial^2 H}{\partial R^2}.
\]

Inserting this expressions into Equation (A.1) yields that

\[
0 = S \frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 S^2 R^2 \frac{\partial^2 H}{\partial R^2} + rSH - rSR \frac{\partial H}{\partial R} + S - A \frac{\partial H}{\partial R} - rSH
\]

\[
= S \frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 SR^2 \frac{\partial^2 H}{\partial R^2} - rSR \frac{\partial H}{\partial R} + S \frac{1 - R}{t} \frac{\partial H}{\partial R}.
\]

Dividing with \(S\), we reach

\[
\frac{\partial H}{\partial t} + \frac{1}{2} \sigma^2 R^2 \frac{\partial^2 H}{\partial R^2} - rR \frac{\partial H}{\partial R} + \frac{1 - R}{t} \frac{\partial H}{\partial R} = 0,
\]

which is Equation (A.2).