Thesis on applications of the
Alternating Direction Implicit method

Bjørn Fjelland Applied Economics and Finance
Supervisor: Professor Bjarne A. Jensen Department of Finance

Copenhagen Business School – January – 2012
## Contents

1 Abstract 4

2 Motivation 5

3 Methods 6
   3.1 Black – Scholes Equation 6
      3.1.1 Brownian motion 6
      3.1.2 Itô’s lemma 7
      3.1.3 Black – Scholes equation 8
   3.2 Partial Differential Equations 11
      3.2.1 Boundary conditions 12
   3.3 Numerical estimation 14
      3.3.1 Discretization 14
      3.3.2 Implicit, Explicit and Crank Nicolson Scheme 16
      3.3.3 Stability, Accuracy and Convergence 17
      3.3.4 Transformations 22
      3.3.5 Tridiagonal Matrix Algorithm (TDMA) 24

4 ADI 25
   4.1 Scheme 25

5 Applications of ADI 26
   5.1 Asian Options 26
      5.1.1 PDE 27
      5.1.2 ADI Schema 28
      5.1.3 Stability constraints 30
      5.1.4 Convergence 33
   5.2 Heston Model 36
      5.2.1 PDE 36
      5.2.2 ADI Schema 37
      5.2.3 Stability 39
      5.2.4 Convergence 42

6 Conclusion 45
7 Matlab Code

7.1 Asian ADI .............................................. 46
7.1.1 Asian.m ............................................. 46
7.1.2 IexplicitAsian.m ................................. 48
7.1.3 IimplicitAsian.m ................................. 50

7.2 Heston ADI .............................................. 51
7.2.1 Heston.m ............................................. 51
7.2.2 IexplicitHeston.m ............................... 54
7.2.3 IimplicitHeston.m ............................... 55
7.2.4 mixedderivativeexplicit.m ....................... 57
1 Abstract

In this thesis I show that finite difference methods are a very good alternative to the much used Monte Carlo simulations for financial problems considering three dimensional PDEs. The ADI scheme produces stable results within one standard deviation of the Monte Carlo price faster than the Monte Carlo simulation. I show two applications of the ADI scheme, one on Asian options where the price depend on the arithmetic average of the underlying, and the Heston model, an option pricing model with stochastic volatility. For the Heston model I also show how to implement a mixed derivative term when the correlation between the two underlying processes differs from zero, and how such terms can impose problems to the stability.

In chapter three I spend time on explaining the Black – Scholes PDE as it is the basis of both the Asian PDE and the Heston PDE, and show finite difference on this two dimensional equation. In chapter four I show the general setup for finite differenced in three dimensions, the ADI scheme. In chapter five I apply the ADI setup on the Asian PDE and than the Heston PDE. I go through stability conditions for both applications and show numerical results on how it converges. I also compare the numerical results against Monte Carlo simulations. Last, in chapter seven the matlab code for both applications are printed.

All numerical results are computed on a quad-core 3.4GHz computer with 8GB RAM, and all code are run in matlab.
2 Motivation

The fall 2009 i chose the elective Mathematical Finance 1, a course thought by professor Bjarne Astrup Jensen. Prior to this the interest for the mathematical applications of economy had been triggered by the mandatory courses International Finance and Capital Market Theory which up til then had been my most quantitative subjects. Without much mathematical background Mathematical Finance showed to be a demanding but also very interesting course, and professor Jensen an inspiring teacher. It was so interesting that I continued with the courses Mathematical Finance 2 and 3. The third course included numerical pricing of instruments, and the logic behind finite difference caught my interest. After discussing the possibility of writing a thesis on something related to finite difference professor Jensen sent me to a PhD course on finite difference in derivatives pricing thought by Jesper Andreasen from Danske Bank. Professor Jensen came with the idea that I could implement the ADI method as we had been treating 2D problems in the mathematical finance courses, and agreed to be my supervisor. After a talk on which instruments to consider I landed on Asian options and options with stochastic volatility.

One week after the course held by Andreasen I started as product controller in Nordea Markets in Norway. Writing the thesis after work made it take longer time than I had hoped for. I have not used professor Jensen as much as I should and have not been good at communicating the progress of the thesis. Working in Nordea have at the same time given valuable input on how numerical pricing are used in real life situations, and I have gotten to use a lot of what I have learned in my daily work. I will definitely continue to explore numerical pricing methods, and hope that the future will provide me with tasks where the knowledge can be applied and further developed.

I want to express my gratitude to professor Jensen for being as patient as he has been with me.
3 Methods

3.1 Black – Scholes Equation

The Black – Scholes equation is a well known equation in mathematical finance. It was first shown in the the paper *The Pricing of Options and Corporate Liabilities* from 1973. Together with Robert Merton, Fischer Black and Myron Scholes revolutionized option pricing theory, and both of the PDEs that ADI will be applied on in this paper originates from the Black – Scholes equation. Merton and Scholes received the 1997 Nobel Prize in Economics for their work.

3.1.1 Brownian motion

The pure Brownian motion is a continuos stochastic process. If we let $B_t$ describe a Brownian motion then:

- $B_0 = 0$
- $(B_{t+s} - B_t) \in \mathcal{N}[0, s]$
- $B_{t+s_1} - B_t$ is independent from $B_t - B_{t-s_2}$ for all $s_1, s_2 > 0$
- $B_t$ is continuos

The pure Brownian motion is used to build more realistic economic models. If we want to model a continuos stochastic process with a drift term, and let the variance grow with a constant over time, we get what we call an arithmetic Brownian motion.

$$dX_t = \mu dt + \sigma dB_t$$

If we instead of modeling the absolute difference in $X_t$ want to model it’s relative development we get the geometrical Brownian motion.

$$\frac{dX_t}{X_t} = \mu dt + \sigma dB_t$$

Stock prices are expected to follow a geometric Brownian motion.
3.1.2 Itô’s lemma

Let $X$ follow the process

$$dX_t = \mu dt + \sigma dB_t, \quad dB_t = \text{pure Brownian motion}$$

and we have a function $f(X_t, t)$ which is twice differential in $X_t$. If we expand this function in a Taylor series we get

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial X_t} dX_t + \frac{1}{2} \frac{\partial^2 f}{\partial X_t^2} dX_t^2 + O(dt^{3/2})$$

where $O(dt^{3/2})$ is infinitely many terms containing $dt$ to the power of two or higher, and $dX_t$ to the power of three or higher. We will later argue that these terms can be seen as $dt$ to the power of $\frac{3}{2}$ or higher. If we now substitute for $dX_t$ in our Taylor expansion, we get

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial X_t} (\mu dt + \sigma dB_t) + \frac{1}{2} \frac{\partial^2 f}{\partial X_t^2} (\mu dt + \sigma dB_t)^2 + O(dt^{3/2})$$

We now consider the third term on the right hand side.

$$\frac{1}{2} \frac{\partial^2 f}{\partial X_t^2} (\mu dt + \sigma dB_t)^2 = \frac{1}{2} \frac{\partial^2 f}{\partial X_t^2} (\mu^2 dt^2 + \sigma^2 dB_t^2 + 2\mu dt dB_t)$$

If recall that $dB_t = \epsilon \sqrt{dt}$ where $\epsilon \in N[0, 1]$ we see that $dB_t^2 = dt \text{var}(\epsilon)$, and the variance of a standard normal variable is of course one. Since $dB_t^2$ equals $dt$ in the power of one it can not be neglected. The $dB_t dt$ term for the same reason equals $\epsilon dt^{3/2}$. If we use the same procedure on the $dX_t$ terms in $O(dt^{3/2})$ we get that all terms in $O(dt^{3/2})$ contains $dt$ to the power of $\frac{3}{2}$ or higher. As $dt$ moves to zero, higher powers vanishes much faster and becomes neglectable in the limit.

If we now neglect all terms with $dt$ to the power of $\frac{3}{2}$ or higher our Taylor series becomes

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial X_t} dX_t + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial X_t^2} dt$$ \hspace{1cm} (3.1)

This relationship is known as Itô’s lemma. We can also let $\mu$ and $\sigma$ be functions of both time and/or $X$. Then the process of $dX$ would be considered
as
\[ dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dB_t \]
and for the special case of a geometrical Brownian motion with \( \mu(X_t, t) = X_t\mu \) and \( \sigma(X_t, t) = X_t\sigma \), Itô’s lemma would state that
\[ df = \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial X}dX + \frac{1}{2}X_t^2\sigma^2 \frac{\partial^2 f}{\partial X^2}dt \]

### 3.1.3 Black – Scholes equation

To derive the Black – Scholes PDE we can construct a portfolio of underlying asset, \( S_t \), derivative, \( C(S_t, t) \), and by combining \( C(S_t, t) \) with \( -\frac{\partial C(S_t, t)}{\partial S_t}S_t \) eliminate the risk. This portfolio should yield the risk free interest rate, due to it’s lack of risk exposure. Since not all underlying risks can be bought and sold\(^1\) we will instead use two derivatives on the same risk source to show the Black – Scholes PDE.

Consider an underlying stochastic variable, \( \theta_t \), which is used as risk source for some assets, \( f_1 \) and \( f_2 \). The dynamics are described by
\[ d\theta_t = \mu_\theta(\theta_t, t)dt + \sigma_\theta(\theta_t, t)dB_t \tag{3.2} \]
\[ df_i(\theta_t, t) = \mu_i(\theta_t, t)dt + \sigma_i(\theta_t, t)dB_t \quad , \quad i = 1, 2 \tag{3.3} \]
In the further I will skip the parentheses to simplify notation. We now combine these two derivatives in a portfolio which eliminates the risk.

\[ \sigma_2df_1 - \sigma_1df_2 = \sigma_2\mu_1dt - \sigma_1\mu_2dt + \sigma_2\sigma_1dB_t - \sigma_1\sigma_2dB_t \]
\[ \text{eliminates risk} \]

Since this portfolio does not have any risk it must clearly yield the risk free rate of return.

\[ (\sigma_2\mu_1 - \sigma_1\mu_2)dt = r(\sigma_2f_1 - \sigma_1f_2)dt \]

Dividing both sides on \( dt \) and rearranging the equation gives us
\[ \sigma_2(\mu_1 - rf_1) = \sigma_1(\mu_2 - rf_2) \]
\[ \frac{(\mu_1 - rf_1)}{\sigma_1} = \frac{(\mu_2 - rf_2)}{\sigma_2} \equiv \lambda \tag{3.4} \]

\(^1\)e.g. weather derivatives
This equation tells us that excess return on the derivative is proportional to the amount of risk undertaken, and lambda, or Sharpe ratio, is the amount of which the expected excess return increases per risk. This lambda is linked to the specific risk source, in this case $dB$, and is the key figure in risk free pricing, the subject of the next subsection. If $\mu_i$ and/or $\sigma_i$ are functions of $\theta_t$ and/or $t$, then so must the Sharpe ratio, $\lambda(\theta_t, t)$. We could have found the Sharpe ratio for the $d\theta$ term by representing the two derivative processes as

$$df_i = \mu_{d\theta,i}dt + \sigma_{d\theta,i}d\theta_t \quad , \quad i = 1, 2$$  \hspace{1cm} (3.5)

and gotten the sharpe ratio for the $d\theta$-risk as

$$\frac{\mu_{d\theta,1} - rf_1}{\sigma_{d\theta,1}} = \frac{\mu_{d\theta,2} - rf_2}{\sigma_{d\theta,2}} \equiv \lambda_{d\theta}$$  \hspace{1cm} (3.6)

If we apply Itô’s lemma on $f_i$ we get

$$df_i = \frac{\partial f_i}{\partial t}dt + \frac{\partial f_i}{\partial \theta_t}d\theta_t + \frac{1}{2}\sigma^2_{\theta} \frac{\partial^2 f_i}{\partial \theta^2_t} dt$$

And by inserting $d\theta_t$ in the equation we get

$$df_i = \frac{\partial f_i}{\partial t}dt + \frac{\partial f_i}{\partial \theta_t}(\mu_\theta dt + \sigma_\theta dB_t) + \frac{1}{2}\sigma^2_{\theta} \frac{\partial^2 f_i}{\partial \theta^2_t} dt$$

When comparing this with the original equation for $df_i$ we can find $\mu_i$ and $\sigma_i$.

$$df_i = \mu_i dt + \sigma_i dB_t$$  \hspace{1cm} (3.7)

\[ \downarrow \]

$$\mu_i = \frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial \theta_t} \mu_\theta + \frac{1}{2}\sigma^2_{\theta} \frac{\partial^2 f_i}{\partial \theta^2_t}$$  \hspace{1cm} (3.8)

$$\sigma_i = \frac{\partial f_i}{\partial \theta_t} \sigma_\theta$$  \hspace{1cm} (3.9)

There is a logical intuition to this. The derivative, $f_i(\theta_t, t)$ must in addition to it’s individual movement over time move together with $\theta_t$ according to their relationship. Since the risk comes from $\theta_t$, and only that, the last term in $\mu_i$ describes the convexity drift due to the variation. The variation in $f_i(\theta_t, t)$ comes only from $\theta_t$ so that $\sigma_i$ must be $\theta_t$’s risk multiplied with $\theta_t$’s
influence on \( f_i(\theta, t) \).

If we now recall the relationship between excess return and risk

\[
\frac{(\mu_i - rf_i)}{\sigma_i} = \lambda
\]

\[
\mu_i - rf_i = \lambda \sigma_i
\]

and insert the expressions for \( \mu_i \) and \( \sigma_i \) and sorting it we get the partial differential equation, PDE,

\[
\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial \theta_t} \mu_\theta + \frac{1}{2} \sigma_\theta^2 \frac{\partial^2 f_i}{\partial \theta_t^2} - rf_i = \lambda \frac{\partial f_i}{\partial \theta_t} \sigma_\theta
\]

\[
\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial \theta_t} (\mu_\theta - \lambda \sigma_\theta) + \frac{1}{2} \sigma_\theta^2 \frac{\partial^2 f_i}{\partial \theta_t^2} - rf_i = 0
\]

If the derivative, \( f_i \), is the underlying asset, \( \theta \), and we insert it’s partial derivatives into the PDE, it becomes clear that \( (\mu_\theta - \lambda \sigma_\theta) = r \theta_t \). This also follows from (3.4). As a result of this we can change the term \( (\mu_\theta - \lambda \sigma_\theta) \) with \( r \theta_t \) in the equation.

\[
f_i = \theta \quad , \quad \frac{\partial f_i}{\partial t} = 0 \quad , \quad \frac{\partial f_i}{\partial \theta_t} = 1 \quad , \quad \frac{\partial^2 f_i}{\partial \theta_t^2} = 0
\]

\[
\downarrow
\]

\[
0 + 1(\mu_\theta - \lambda \sigma_\theta) + \frac{1}{2} \sigma_\theta^2 0 - rf_i = 0
\]

\[
\downarrow
\]

\[
\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial \theta_t} r \theta_t + \frac{1}{2} \sigma_\theta^2 \frac{\partial^2 f_i}{\partial \theta_t^2} - rf_i = 0
\]

The PDE with full notation

\[
\frac{\partial f_i(\theta_t, t)}{\partial t} + \frac{\partial f_i(\theta_t, t)}{\partial \theta_t} r \theta_t + \frac{1}{2} \sigma_\theta(\theta_t, t) \frac{\partial^2 f_i(\theta_t, t)}{\partial \theta_t^2} - rf_i(\theta_t, t) = 0 \quad (3.10)
\]

This is the Black–Scholes partial differential equation. If \( \theta \) follows a geometric Brownian motion, which stock prices are considered to do, the term
\( \sigma_0(\theta_t, t) \) will equal \( \theta^2 \sigma^2 \) and the PDE will take it's famous form

\[
\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial \theta_t} \theta_t + \frac{1}{2} \sigma^2 \theta^2 \frac{\partial^2 f_i}{\partial \theta^2_t} - rf_i = 0
\]

If considering an interest rate derivative, bond, \( P \), an increase in interest rate is considered negative for the price. We then model the processes for the interest rate and the bond price as

\[
dr = \mu_r(r)dt + \sigma_r(r)dB_t
\]

\[
dP(r, t, T) = \mu_P(r, t, T)dt - \sigma_P(r, t, T)dB_t
\]

where \( t \) is time, \( T \) is maturity and \( r \) is the interest rate. When doing the same procedure to derive the Black – Scholes PDE we get only some minor differences. \( \lambda \) is no longer constant, but is varying over \( t \) and \( r \), and \( \mu_r \) is not reduced with \( \lambda \) times amount of risk, it is added on. The PDE for the bonds then becomes

\[
\frac{\partial P}{\partial t} + \frac{\partial P}{\partial r}(\mu_r(r) + \lambda(r, t)\sigma_r(r)) + \frac{1}{2} \sigma_r(r)^2 \frac{\partial^2 P}{\partial r^2} - rP = 0
\]

where \( P \) is the simplified notation of \( P(r, t, T) \).

### 3.2 Partial Differential Equations

A differential equation is an equation containing a function and at least one of its derivatives. A partial differential equation is an equation involving a function of more than one variable and its derivatives. The Black – Scholes equation is a partial differential equation. To sort what kind of PDE we are dealing with we arrange them by orders. The order of the PDE is the highest order of derivative that is included, and if every term only include either constants, the function or at most one of the partial derivatives it is called linear. If there are no constant term the PDE is called homogeneous. We can by this see that the Black – Scholes PDE is a linear, homogeneous PDE of the second order.
3.2.1 Boundary conditions

When solving a PDE analytically and finding the function, \( f(\theta_t, t) \), some more information than the equation in itself is needed. We call this *boundary conditions*, and will typically be a curve in the \((\theta_t, t)\) plane where \( f \) is known. If this known \( f \) is in \( t = 0 \) we call it an *initial boundary problem* where \( f(\theta_0, 0) \) is the *initial boundary condition*. For the Black – Scholes PDE on an european type derivative we know the derivative value in \( f(\theta_T, T) \) and by stating that \( \tau \equiv T - t \) and \( g(\theta_\tau, \tau) \equiv f(\theta_{T-\tau}, T - \tau) \) we see that solving for \( g(\theta_\tau, \tau) \) is an initial boundary problem. The solved problem will give us the function \( g(\theta_T, T) \) which from our transform must equal \( f(\theta_0, 0) \). \( \theta_0 \) is the underlying assets price today, which is known, and tells us which value in \( f(\theta_0, 0) \) we want.

When pricing an american style derivative, which may be exercised for all \( t, 0 \leq t \leq T \), we no longer know when the derivative is going to be exercised and undertake the known functional form, \( f(\theta_T, T) \). We are no longer considering an initial value problem, but a *free boundary problem*. For these problems there are seldom analytic solutions.\(^2\) For american type call options the level of where the option would be exercised becomes infinitely high when there is no dividend, it will not be exercised premature, and the price will equal the analytic price on an european type, cet. par. With a positive dividend the option might be exercised premature. The only difference between American and European options is the boundary condition, so the Black – Scholes PDE still describes the price development, and since the American type does not have an initial boundary condition it needs to be priced numerically. When treating numerical solutions of the PDE we must put even more constraints to the system. In addition to the obvious discretization of the domain, \( \Omega \), where \( f(\theta_t, t) \) can move, one must seal of the domain by restraining the maximum and minimum values of which \( \theta_t \) can undertake at any \( t \). Combined with the boundary condition in \( f(\theta_T, T) \) and the ending of the domain in \( t = 0 \) we have closed the whole domain.

Geometric Brownian motions, which many underlying assets are sup-

\(^2\)Except for perpetuities which makes the Black – Scholes equation an *Ordinary Differential Equation* where the free boundary is constant over \( t \).

\[
\frac{\partial f}{\partial \theta_t} r \theta_t + \frac{1}{2} \sigma^2 \theta_t^2 \frac{\partial^2 f}{\partial \theta_t^2} - rf_t = 0
\]
posed to follow, can not undertake negative values, and the lower boundary are by the process automatically set to $f(0, t)$. Other processes might undertake values from $-\infty$, but might have a natural maximum boundary. As a part of the system we have to decide on how we handle the derivatives terms in the boundaries $f(0, t)$ and $f(max(\theta_t), t)$. There are two common ways to do this. Either one set $\frac{\partial f}{\partial \theta}$ and $\frac{\partial^2 f}{\partial \theta^2}$ to be zero, a Dirichlet condition, we use a one sided approximation of $\frac{\partial f}{\partial \theta}$, or state that $\frac{\partial f}{\partial \theta}$ is constant in the boundary, a Neuman boundary condition. The Neuman condition gives a higher accuracy, but as we will see later it might give unstable solutions. Setting proper boundary conditions are together with doing a good discretization and applying a good numerical method the main issues in numerical pricing.
Figure 1: The grids over $\theta$ and $\tau$, the inversed time, for finite difference estimation of the PDE when $\theta \in [0,\infty)$ to the left, and $\theta \in (-\infty,\infty)$ to the right.

3.3 Numerical estimation

3.3.1 Discretization

When discretizing the domain, $\Omega$, where $\theta_t$ can move, one first have to seal it of. Some derivatives have natural bounds to the domain, but we might not have an analytic solution to where these bounds are. Other derivatives might not have underlaying risk sources with natural sealings up, nor down, and need some constructed limits on how far the underlaying can move. The main rule is that the broader the domain, the better the estimation, but there are methods to restrain the domain without having large reductions in the quality of our numerical estimate [5].

When the domain is sealed we discretize it by constructing a grid on $N$ steps in the $t$-direction and $K$ steps in the $\theta$-direction. Every step in the $t$-direction has the length of $\Delta t$ and every step in the $\theta$-direction has the length of $\Delta x$. See figure (1)

$$N\Delta t = T$$
$$\theta_{\text{min}} + K\Delta x = \theta_{\text{max}}$$

There are, as discussed later in the section about convergence, a close connection between the quality of the solution, estimate of $f(\theta_0, 0)$, and the size of $\Delta x$ relative to $\Delta t$. As the size of $\rho$ where $\rho \equiv \frac{\Delta x}{\Delta t}$ decreases, the quality of the numerical solution improves.

When choosing where to place the discrete grid in the $\theta$-direction one have to consider both the initial boundary condition, and the choice of one-
or two sided approximation. As shown in the example to the right in figure (2) when placing the discretization so that the strike is located between two points the discrete values will underestimate the real values to the right of the point, but overestimate the real value to the left. The over- and underestimation will net off each other. This is also the case for all except one point when choosing a discretization where one of the points fall at the strike. However, it is obvious that for this point the right hand side still underestimates the real value, but there are no net off to the left. We end up with an under estimation that could have been avoided if discretization were placed as the example to the right in figure (2). When handling part wise linear boundary conditions we should therefore place the strike between two points. If considering at the money options, derivatives where the strike equals the spot price, using a discretization like the one left in figure (2) will let the desired $\theta_0$ match a node, so preventing errors in interpolation between two points can be a good reason for choosing this discretization to the left anyway. When considering a transformed PDE, with the boundary condition transformed accordingly, the transformed boundary condition will typically not be part wise linear but should be examined, and the grid placed, to avoid over- or underestimation of the discretized boundary condition.

It seems reasonable when discretizing the domain, $\Omega$, for a stock option to choose a discretization over the stock price, $\theta$, from zero to some great amount, covering a sufficient part of the probability distribution for $\theta_T$. Even when $\theta_t$ is downward bounded by zero, like in this example, it is
not rational to expand $\Omega$ all the way to $\theta = 0$. The highest accuracy will be obtained if the cutoff from the up- and downside of the probability distribution are equally great. When including the total downside distribution one uses computational power that would be more efficiently used on the upper probability distribution.

### 3.3.2 Implicit, Explicit and Crank Nicolson Scheme

When numerically solving a PDE like the Black Scholes equation one starts of by approximating the derivatives terms.

$$\frac{\partial v_i}{\partial t} + \frac{\partial v_i}{\partial S} r S_t + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 v_i}{\partial S^2} - r v_i = 0$$  
\[ (3.11) \]

\[ \downarrow \]

$$\frac{V_{i+1}^{t+1} - V_i^t}{\Delta t} + \frac{V_{i+1}^{t+1} - V_{i-1}^{t+1}}{2\Delta S} r S_t + \frac{1}{2} \sigma^2 S^2 \frac{V_{i+1}^{t+1} - 2V_i^{t+1} + V_{i-1}^{t+1}}{\Delta S^2} = r V_i$$  
\[ (3.12) \]

where $V$ is the discretized $v$, and we have chosen a one sided approximation of $\frac{\partial v_i}{\partial t}$, and two sided approximations of both $\frac{\partial v_i}{\partial S}$ and $\frac{\partial^2 v_i}{\partial S^2}$. We could of course choose both central, two sided approximations, or one sided approximations in either direction. Later we will see that the choice of approximation will affect the stability of the system. Except for the $\frac{\partial v_i}{\partial t}$ term none are yet defined in the time direction. The $V$s can be of any linear combination of $t$ and $(t + 1)$. The most common is to choose them of either $t$ or $(t + 1)$, and if set to $(t + 1)$ we have an explicit system.

$$\frac{V_{i+1}^{t+1} - V_i^t}{\Delta t} + \frac{V_{i+1}^{t+1} - V_{i-1}^{t+1}}{2\Delta S} r S_t + \frac{1}{2} \sigma^2 S^2 \frac{V_{i+1}^{t+1} - 2V_i^{t+1} + V_{i-1}^{t+1}}{\Delta S^2} = r V_i$$  
\[ (3.13) \]

It is called an explicit system as we can explicitly calculate the array of $V$’s when we know $V_{i+1}^{t+1}$. Recall that in the Black – Scholes equation it is the end time, not the start time, that has a known boundary. if we on the other hand choose all derivatives approximations of the time $t$, we would have an implicit system.

$$\frac{V_{i+1}^{t+1} - V_i^t}{\Delta t} + \frac{V_{i+1}^{t+1} - V_{i-1}^{t+1}}{2\Delta S} r S_t + \frac{1}{2} \sigma^2 S^2 \frac{V_{i+1}^{t+1} - 2V_i^{t+1} + V_{i-1}^{t+1}}{\Delta S^2} = r V_i$$  
\[ (3.14) \]
This system has to be implicitly calculated, and even if this might impose computational difficulties one could avoid by choosing an explicit approximation, we will later see that implicit calculations of the Black–Scholes equation are unconditionally stable. The most used method, both due to it always being stable, and that it got higher accuracy than both the explicit and the implicit scheme, is the Crank–Nicolson scheme. If we move the right hand side over to the left on both the explicit and the implicit scheme and set them as equal we get a scheme that is both implicit and explicit.

\[
\frac{V_{i+1}^{t+1} - V_i^t}{\Delta t} + \frac{V_{i+1}^{t+1} - V_{i-1}^{t+1}}{2\Delta S} r S_t + \frac{1}{2} \sigma^2 S S S \frac{V_{i+1}^{t+1} - 2V_{i}^{t+1} + V_{i-1}^{t+1}}{\Delta S^2} - r V_i^{t+1} \tag{3.15}
\]

\[
\frac{V_{i+1}^{t+1} - V_i^t}{\Delta t} + \frac{V_{i+1}^t - V_{i-1}^t}{2\Delta S} r S_t + \frac{1}{2} \sigma^2 S S S \frac{V_{i+1}^t - 2V_{i}^t + V_{i-1}^t}{\Delta S^2} - r V_i^t \tag{3.16}
\]

This scheme, Crank–Nicolson, is the one used in the Alternating Direction Implicit method.

### 3.3.3 Stability, Accuracy and Convergence

#### Stability

The schemes, Implicit, Explicit and Crank Nicolson, have different stability properties. When describing the process as

\[V^t = BV^{t+1}\]

for the system to be stable an arbitrary norm of \(B\) can be no larger than one, or the solution would grow exponentially and not converge.

\[\|B^k\| \leq 1 \quad \forall k \quad n > 0\]

or, if written in terms of eigenvalues, where \(\Lambda_i\) is the eigenvalues of \(B\).

\[|\Lambda_i| \leq 1\]
knowing that expressions for $B$ yields (from the discretization chapter) and describing $M$'s eigenvalues as $\lambda_i$

**Implicit:** 
$$B = \left( \frac{1}{1 + M \Delta t} \right) \quad \Rightarrow \quad \Lambda_i = \left( \frac{1}{1 + \lambda_i \Delta t} \right)$$

**Explicit:** 
$$B = \left( \frac{1 - M \Delta t}{1} \right) \quad \Rightarrow \quad \Lambda_i = \left( \frac{1 - \lambda_i \Delta t}{1} \right)$$

**Crank Nicolson:** 
$$B = \left( \frac{1 - M \Delta t}{1 + \frac{M \Delta t}{2}} \right) \quad \Rightarrow \quad \Lambda_i = \left( \frac{1 - \lambda_i \Delta t}{1 + \frac{\lambda_i \Delta t}{2}} \right)$$

We see that for both the implicit and the Crank Nicolson scheme it holds to ensure $0 \leq \lambda_i$ for all $i$, while for the explicit scheme $0 \leq \lambda_i \Delta t \leq 2$. By applying the **Gershgorin circle theorem** which bounds the domain of possible eigenvalues we will show that $\lambda_i$ is in fact positive for all economically rationale situations. The Gershgorin circle theorem states that all eigenvalues for a square matrix with entries $m_{ij}$ must lay in the union of circles with centrum in $m_{ii}$ for all $i$ and with radius as $R_i$ where

$$R_i = \sum_{i=1}^{n} |m_{ji}|$$

We know that for $2 \leq i \leq (n - 1)$ will give $R_i$ as $|a_i| + |c_i|$, or more precise

$$R_i = |a_i| + |c_i| = \begin{cases} \sigma_{j,n}^{2} \left( \frac{\Delta x}{\Delta t} \right)^2 & \text{if } \frac{\mu_{j,n}}{\Delta x} < \frac{\sigma_{j,n}^{2}}{2(\Delta x)^2} \\ \frac{\mu_{j,n}}{\Delta x} & \text{if } \frac{\mu_{j,n}}{\Delta x} > \frac{\sigma_{j,n}^{2}}{2(\Delta x)^2} \end{cases}$$

For the first, which from $(\Delta x)^2 \ll \Delta x$ holds for all normal problems we see that the possible values of the eigenvalues, $\lambda_i$, is given as

$$\lambda_i \in \left( b_{ii} \pm \frac{\sigma_{j,n}^{2}}{(\Delta x)^2} \right)$$

with the lower bound as

$$\min (\lambda_i) = b_i - \frac{\sigma_{j,n}^{2}}{(\Delta x)^2} = r_{j,n}$$
Since one of the assumptions of our models is a positive interest in all states, \( r_{j,n} > 0 \), we only need to ensure boundary conditions with

\[
\begin{align*}
  m_{11} - \sum_{i=2}^{n} |m_{1i}| &> 0 \quad (3.17) \\
  m_{nn} - \sum_{i=1}^{n-1} |m_{ni}| &> 0 \quad (3.18)
\end{align*}
\]

to have a stable numerical estimation using the Implicit or Crank Nicolson scheme. We see that for a very small \( \Delta x \) the maximal possible eigenvalue of the matrix, \( M \), becomes very large, and a sufficiently small \( \Delta t \) is required to fulfill the demanded inequality \( 0 \leq \lambda_i \Delta t \leq 2 \) for the Explicit scheme.

\[
\max (\lambda_i) \Delta t = \left( r_{j,n} + 2 \frac{\sigma_{j,n}^2}{(\Delta x)^2} \right) \Delta t \leq 2
\]

Even though the Crank Nicolson scheme is unconditionally stable both its internal explicit and implicit schemes have to be stable for one to be certain of the stability of its higher order derivatives. [4] show that with non smooth functions as boundary conditions Crank Nicolson produce non stable solutions for the higher order derivatives when \( \frac{\Delta x}{\Delta t} \) becomes sufficiently large. Non smooth boundary conditions are normally the case in derivatives pricing as they often have at least one option element.

**Accuracy**

The accuracy of the different schemes vary with the accuracy of the approximations of the partial derivatives. Both the Implicit and the Explicit scheme use one sided approximations of \( \frac{\partial f}{\partial t} \), and all schemes might use either one sided or two sided approximations of \( \frac{\partial f}{\partial \theta} \). By expanding the series in a Taylor
expansion we can obtain the order of accuracy of the one sided schemes.

\[
\frac{V(x + \Delta x, t) - V(x, t)}{\Delta x} = \frac{1}{\Delta x} \left( \frac{\Delta x}{1} \frac{\partial V}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 V}{\partial x^2} + \ldots \right) = \frac{\partial V(x, t)}{\partial x} + O(\Delta x)
\]

\[
\frac{V(x + \Delta t) - V(x, t)}{\Delta t} = \frac{1}{\Delta t} \left( \frac{\Delta t}{1} \frac{\partial V}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 V}{\partial t^2} + \ldots \right) = \frac{\partial V(x, t)}{\partial t} + O(\Delta t)
\]

And for the two sided approximation of \(\frac{\partial f}{\partial \theta}\).

\[
\frac{V(x + \Delta x, t) - V(x - \Delta x, t)}{2\Delta x} = \frac{1}{2\Delta x} \left( \frac{\Delta x}{1} \frac{\partial V}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 V}{\partial x^2} + \frac{(\Delta x)^3}{6} \frac{\partial^3 V}{\partial x^3} + \ldots \right) = \frac{\partial V(x, t)}{\partial x} + O\left((\Delta x)^2\right)
\]

When considering the time derivative in the Crank Nicolson scheme we recall that the approximation yields \(\frac{1}{2} \left( V(x, t) + V(x, t + \Delta t) \right) \), hence we got a two sided approximation of \(\frac{\partial f}{\partial t}\) around \(V(x, t + \frac{1}{2}\Delta t)\)

\[
\frac{V(x, t + \Delta t) - V(x, t)}{\Delta t} = \frac{1}{\Delta t} \left( \frac{\Delta t}{2} \frac{\partial V}{\partial t} + \frac{(\Delta t)^2}{8} \frac{\partial^2 V}{\partial t^2} + \frac{(\Delta t)^3}{48} \frac{\partial^3 V}{\partial t^3} + \ldots \right) = \frac{\partial V(x, t + \frac{1}{2}\Delta t)}{\partial t} + O\left((\Delta t)^2\right)
\]

We see that the while the Implicit and Explicit scheme have first order accuracy in \(\Delta t\), Crank Nicolson has second order accuracy in both.

**Convergence**

Even if both the Implicit and the Crank Nicolson scheme are theoretically unconditionally stable, \[4\] show that we might suffer problems with convergence, either for our estimate of \(f(\theta t, t)\) or one or more of it’s derivatives. Our boundary condition in \(f(\theta T, T)\) is often only piecewise smooth. Consider an european call, \(C(S_t, t)\) with \(C(S_T, T) = \max(S_T \geq K, 0)\), where the payout is described in figure 3. We see that the second derivative is infinitely high.
in $S_T = K$, zero before and after, so it can not be expressed as an ordinary function. It is a Dirac delta function, $\delta(x)$, with the properties

$$\delta(x) = \begin{cases} \infty & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}, \quad \int_{-\infty}^{\infty} \delta(x)dx = 1$$

As this is not an ordinary function it can not be expressed pointwise as we normally express functions, it has to be expressed multiplied with another function in an integral. Since its own integral is one, and its only value is in $x = 0$ the multiplication with another function in a integral must yield

$$\int_{-\infty}^{\infty} \delta(x)f(x)dx = f(0) \quad (3.19)$$

When computing the option price we are not making a numerical estimate for the function $C(S_T, T)$, we are estimating $C(S_t, t)$ which is smooth for all $t < T$. Problems occur when the time grid becomes very fine, and we are to produce numerical estimates very close to $t = T$ in $S_t = K$. Here the real value of the first derivative is extreme, but as our estimate does not take the discontinuity of $C(S_T, T)$ into account, our estimate depends on the fineness of our grid. When numerically estimating option prices using the Black – Scholes PDE we are estimating both the first and second derivative with regards to $S_t$, so the problem is present for all options with only piecewise smooth initial boundary conditions.

For non continuous options, like digital options, the first derivative in $S$ - direction would be the Dirac delta function. In $t = T$ the digital call options

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{The payout on an european call option where the second order derivative with regards to $S_T$ is the Dirac delta function.}
\end{figure}
second derivative with regards to $S_T$ must be the same as the first derivative of the Dirac delta function, $\delta(x)$, with regards to $x$. This function is zero at every time except from $x = 0$ where it is undefined. This is an additional problem for our numerical estimation, and we might find that we need to refine our methods to make the system converge. By choosing $\Delta x$ so that both the explicit and the implicit scheme is stable, would remedy this.

3.3.4 Transformations

When solving the black–Scholes PDE it is a common approach to transform the PDE to the Heat equation (3.20), a well explored PDE from physics. There is a known analytic solution to this equation, but even when solving the Black–Scholes PDE numerical it is an often used transformation. The choice of transformation can effect our final result.

$$\frac{\partial v}{\partial t} = c \frac{\partial^2 v}{\partial x^2}$$ (3.20)

The normal approach to the transform is to define a new variable, $\hat{\theta} \equiv \ln \left( \frac{\theta}{\pi} \right)$, to make the equation dimensionless. In addition we define a new variable, $\hat{f}(\hat{\theta}, t) \equiv f(\theta, t)$. We recall that the Black–Scholes equation is stated as

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial \theta} r \theta + \frac{1}{2} \sigma^2 \theta^2 \frac{\partial^2 f}{\partial \theta^2} - rf = 0$$ (3.21)

and see that we can rewrite the derivatives as

$$\frac{\partial \hat{f}}{\partial \theta} = \frac{\partial \hat{f}}{\partial \theta} \frac{\partial \hat{\theta}}{\partial \theta} = \frac{1}{\theta} \frac{\partial \hat{f}}{\partial \theta}$$ (3.22)

$$\frac{\partial^2 \hat{f}}{\partial \theta^2} = \frac{1}{\theta^2} \left( \frac{\partial^2 \hat{f}}{\partial \theta^2} - \frac{\partial \hat{f}}{\partial \theta} \right)$$ (3.23)

then we can rewrite the Black–Scholes PDE as (3.24) where the equation no longer is dependent on the level of $\theta$.

$$\frac{\partial \hat{f}}{\partial t} + \frac{\partial \hat{f}}{\partial \theta} \left( r - \frac{1}{2} \sigma^2 \right) + \frac{1}{2} \sigma^2 \frac{\partial^2 \hat{f}}{\partial \theta^2} - r \hat{f} = 0$$ (3.24)
We see that there still is a discount factor from $r\hat{f}$, and by defining $x(\hat{\theta}, t) \equiv e^{r(T-t)} f(\hat{\theta}, t)$ and substituting it into (3.24) we get the equation

$$\frac{\partial x}{\partial t} + \frac{\partial x}{\partial \hat{\theta}} \left( r - \frac{1}{2} \sigma^2 \right) + \frac{1}{2} \sigma^2 \frac{\partial^2 x}{\partial \hat{\theta}^2} = 0$$

(3.25)

By scaling we want to get common coefficients for the $\hat{\theta}$-derivatives. Then we need to define the new variables

$$z \equiv \hat{\theta} r - \frac{1}{2} \sigma^2 \frac{1}{2} \sigma^2$$

(3.26)

$$\hat{t} \equiv (T-t) r - \frac{1}{2} \sigma^2 \frac{1}{2} \sigma^2$$

(3.27)

Which when substituted into (3.25) is giving the equation

$$\frac{\partial x}{\partial \hat{t}} = \frac{\partial x}{\partial z} + \frac{\partial^2 x}{\partial z^2}$$

(3.28)

By redefining and substituting $\hat{x}(\hat{z}, \hat{t}) \equiv \hat{x}(z + \hat{t}, \hat{t}) \equiv x(z, \hat{t})$ the equation becomes the previously mentioned heat equation

$$\frac{\partial \hat{x}}{\partial \hat{t}} = \frac{\partial^2 \hat{x}}{\partial z^2}$$

(3.29)

There is no longer a boundary condition in $\hat{x} = 0$, because $\hat{x}$ can, from the very first transformation, $\hat{\theta} = \ln \left( \frac{\theta}{K} \right)$ undertake any value in $(-\infty, \infty)$. If choosing to solve the derivative prices from this PDE rather than the Black–Scholes it is important to remember that an uniform grid over this transformed equation does not give an uniform grid over $\theta$.

$$\ln \left( \frac{\theta}{K} \right) \quad \theta$$

<table>
<thead>
<tr>
<th></th>
<th>1004.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>369.5</td>
</tr>
<tr>
<td>2</td>
<td>135.9</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>0</td>
<td>18.4</td>
</tr>
<tr>
<td>-1</td>
<td>6.8</td>
</tr>
<tr>
<td>-2</td>
<td>2.5</td>
</tr>
</tbody>
</table>

(3.30)
3.3.5 Tridiagonal Matrix Algorithm (TDMA)

The Tridiagonal Matrix Algorithm was developed by Lleuvellyn Thomas and is a way of solving implicit three diagonal system equations. Consider the system below where the $v^n$s are known and $v^{n+1}$s are unknown.

\[
\begin{bmatrix}
    b_I & c_I & 0 & 0 \\
    a_i & b_i & c_i & 0 \\
    0 & \cdots & \cdots & \cdots \\
    0 & 0 & a_0 & b_0
\end{bmatrix}
\times
\begin{bmatrix}
    v_{I_i}^{n+1} \\
    v_{i}^{n+1} \\
    \vdots \\
    v_{0}^{n+1}
\end{bmatrix}
=
\begin{bmatrix}
    v_I^n \\
    v_i^n \\
    \vdots \\
    v_0^n
\end{bmatrix}
\quad (3.31)
\]

Since all rows sum up to zero we are free to both scaling them individually and subtracting them from each other. By scaling the first row with $a_{I-1}$ and subtract it from row two scaled by $b_I$, we get the new row for equation two as

\[
b_I \left( a_{I-1}v_{I-1}^{n+1} + b_{I-1}v_{I-2}^{n+1} + c_{I-1}v_{I-1}^{n+1} \right) - a_{I-1} \left( b_I v_{I-1}^{n+1} + c_I v_{I-1}^{n+1} \right) = b_I v_{I-1}^n + a_{I-1} v_{I-1}^n
\]

As we see the $v_{I-1}^{n+1}$ term disappears, and by performing the same operation with row three and the modified row two we can remove the term $v_{I-2}^{n+1}$ from the third equation. Doing this procedure on all $I$ rows will make (3.31) into a two-diagonal system.

\[
\begin{bmatrix}
    b_I & \hat{c}_I & 0 & 0 \\
    0 & \hat{b}_i & \hat{c}_i & 0 \\
    0 & \cdots & \cdots & \cdots \\
    0 & 0 & 0 & \hat{b}_0
\end{bmatrix}
\times
\begin{bmatrix}
    \hat{v}_{I_i}^{n+1} \\
    \hat{v}_{i}^{n+1} \\
    \vdots \\
    \hat{v}_{0}^{n+1}
\end{bmatrix}
=
\begin{bmatrix}
    \hat{v}_I^n \\
    \hat{v}_i^n \\
    \vdots \\
    \hat{v}_0^n
\end{bmatrix}
\quad (3.32)
\]

where

\[
\hat{b}_i = 1 \quad \hat{c}_i = \frac{c_i}{b_i - \hat{c}_{i+1} a_i} \quad \hat{v}_i^n = \frac{v_i^n}{b_i - \hat{v}_{i+1} a_i}
\]

This system can easily be solved without inverting and multiplying the two-diagonal matrix to the right hand side. By starting off with solving for the one unknown in the last row, finding $v_0^{n+1}$ and then having only one unknown in the next row, and so forth, the whole array can be computed in $I$ steps.
4 ADI

4.1 Scheme

Finite difference methods can be extended to handle derivative terms in more dimensions. If we consider a PDE with differential terms in $t$, $I$ and $S$ direction, discretized by 1000 points each, every iteration will have to perform calculations connected to $1000^2$ grid points. As one would like to minimize the number of calculations without losing accuracy, convergence and stability, a method called Alternating Direction Implicit method (ADI) is invented.

If we call the unknown function $V$ and introduce the multiplicators $D_S$ and $D_I$ we can write the equation as

$$0 = \frac{\partial V}{\partial t} + D_S V + D_I V$$ (4.1)

If we discretize $V(S,I,t)$, note it as $v_{st}$ and use a one sided approximation of $\frac{\partial V}{\partial t}$ we can write the equation as

$$0 = \left[ v_{st} + \Delta t \right] - \left[ v_{st} \right] + D_S [v] + D_I [v]$$ (4.2)

where $[v]$ is the matrix of $v_{st}$'s and $\bar{D}_S$ and $\bar{D}_I$ are matrixes that suits the discretization of $D_S$ and $D_I$. The matrixes $[v]$ can be of any arbitrary linear combination of the discretized times, $t - \Delta t$, $t + \frac{1}{2} \Delta t$ and $t$. If we choose one of $[v]$ to be evenly split between $t + \Delta t$ and $t$, the other in $t + \frac{1}{2} \Delta t$, and add $\left[ v_{st+\frac{1}{2}\Delta t} \right] - \left[ v_{st+\frac{1}{2}\Delta t} \right] = 0$ in the nominator of the time derivative approximation we get

$$0 = \frac{\left[ v_{st+\Delta t} \right] + \left[ v_{st+\frac{1}{2}\Delta t} \right] - \left[ v_{st+\frac{1}{2}\Delta t} \right] - \left[ v_{st} \right]}{\Delta t} + \frac{1}{2} \left[ \bar{D}_I \left[ v_{st+\frac{1}{2}\Delta t} \right] + \bar{D}_S \left[ v_{st+\Delta t} \right] \right] + \frac{1}{2} \left[ \bar{D}_I \left[ v_{st+\frac{1}{2}\Delta t} \right] + \bar{D}_S \left[ v_{st} \right] \right]$$ (4.3)

$$0 = \frac{\left[ v_{st+\Delta t} \right] - \left[ v_{st+\frac{1}{2}\Delta t} \right]}{\Delta t} + \frac{1}{2} \left[ \bar{D}_I \left[ v_{st+\frac{1}{2}\Delta t} \right] + \bar{D}_S \left[ v_{st+\Delta t} \right] \right] + \frac{1}{2} \left[ \bar{D}_I \left[ v_{st+\frac{1}{2}\Delta t} \right] + \bar{D}_S \left[ v_{st} \right] \right]$$ (4.4)

$$0 = \frac{\left[ v_{st+\frac{1}{2}\Delta t} \right] - \left[ v_{st-\Delta t} \right]}{\Delta t} + \frac{1}{2} \left[ \bar{D}_I \left[ v_{st+\frac{1}{2}\Delta t} \right] + \bar{D}_S \left[ v_{st} \right] \right]$$ (4.5)

Setting both (4.4) and (4.5) to zero will clearly satisfy the original equation.
Sorting them on the different \( t \)'s will lead to the traditional ADI scheme.

\[
\begin{align*}
\left( 1 + \frac{1}{2} \Delta t D_S \right) [v^{t+\Delta t}] &= \left( 1 - \frac{1}{2} \Delta t D_I \right) [v^{t+\frac{1}{2}\Delta t}] \quad (4.6) \\
\left( 1 + \frac{1}{2} \Delta t D_I \right) [v^{t+\frac{1}{2}\Delta t}] &= \left( 1 - \frac{1}{2} \Delta t D_S \right) [v^t] \quad (4.7)
\end{align*}
\]

When solving the PDE one starts with \([v^t]\), computes \([v^{t+\frac{1}{2}\Delta t}]\) which is used to compute \([v^{t+\Delta t}]\). The scheme has a nice symmetry. In (4.6) the scheme is implicit in \( I \)-direction and explicit in \( S \)-direction. In (4.7) it is exactly the opposite way around. The scheme is implicit in \( S \)-direction and explicit in \( I \)-direction. As of this both steps are Crank–Nicolson schemes, alternating the direction in \( I \) and \( S \), as the name of the method implies.

## 5 Applications of ADI

### 5.1 Asian Options

Amongst exotic derivatives we find the Asian type options. Common for them are the sensitivity to the underlying arithmetic average price. There are no analytic solution to Asian option problems, and there are mainly two types, the average strike options, and the fixed strike options, and for both of the types there are puts and calls. The average strike options will have a payout that depends on both the underlying price and the average, depending on whether it is a put or a call. The fixed strike options will not rely on the final price, only the arithmetic average of the underlying, and the strike will be fixed. Besides the boundary condition in \( T \), everything is equal, so constructing a scheme for one would hold for the other. We can write the boundary conditions as

Fixed strike call = \([\bar{S} - K]^+\)  
Fixed strike put = \([K - \bar{S}]^+\) 

Fixed strike call = \([S_T - \bar{S}]^+\)  
Fixed strike call = \([\bar{S} - S_T]^+\)
5.1.1 PDE

Unlike the options that are only influenced by the stochastic process of the underlying in future times and time itself, asian options contain a term with information about the arithmetic average of the historical prices. For such derivatives the two arbitrary functions $df_i(\theta_t, t)$ in (3.3) would be stated as

$$df_i(\theta_t, t) = \mu_i(\theta_t, t)dt + \sigma_i(\theta_t, t)dB_t + (\theta_t - I_t) t^{-1} \frac{\partial f_i}{\partial I_t} dt, \quad i = 1, 2 \quad (5.1)$$

where $(\theta_t - I_t) t^{-1}$ is the current contribution to the average and $\frac{\partial f_i}{\partial I_t}$ is the sensitivity to the average of $\theta$. If we apply Itô’s lemma on equation (5.1) and insert $d\theta_t$ as in (2.2) we get

$$df_i = \frac{\partial f_i}{\partial t} dt + \frac{\partial f_i}{\partial \theta_t} d\theta_t + \frac{1}{2} \sigma_i^2 \frac{\partial^2 f_i}{\partial \theta_t^2} dt + (\theta_t - I_t) t^{-1} \frac{\partial f_i}{\partial I_t} dt \quad (5.2)$$

Following the steps in (3.7) to (3.10) leads to the final PDE

$$\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial \theta_t} r_\theta_t + \frac{1}{2} \sigma_i^2 \frac{\partial^2 f_i}{\partial \theta_t^2} + (\theta_t - I_t) t^{-1} \frac{\partial f_i}{\partial I_t} = 0 \quad (5.3)$$

and if the underlying is a stock and we call the instrument $V$, the equation will take the form of the Asian derivatives PDE:

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial S_t} r S_t + \frac{1}{2} \sigma^2 S_t^2 \frac{\partial^2 V}{\partial S_t^2} + (I_t - I_t) \frac{\partial V}{\partial I_t} - rV = 0 \quad (5.4)$$

where

$$I(t) = \frac{1}{t} \int_0^t S_\tau d\tau \quad (5.5)$$

The PDE could alternatively be written as

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial S_t} r S_t + \frac{1}{2} \sigma^2 S_t^2 \frac{\partial^2 V}{\partial S_t^2} + S_t \frac{\partial V}{\partial I_t} - rV = 0 \quad (5.6)$$

where

$$I(t) = \int_0^t S_\tau d\tau \quad (5.7)$$

but then our start point in $t_0$ would be at the boundary of our domain for any contract, $I_0 = 0$, and the error from the boundary condition could be great. For our solution I have therefor chosen to consider the PDE in (5.4)
5.1.2 ADI Schema

By introducing the multiplicators $D_S$ and $D_I$ we can write equation (5.4) as

$$0 = \frac{\partial V}{\partial t} + D_S V + D_I V$$

(5.8)

where

$$D_S = \frac{1}{2} \sigma^2 S^2 \frac{\partial^2}{\partial S^2} + rS \frac{\partial}{\partial S}$$

(5.9)

$$D_I = (S-I) \frac{\partial}{\partial I} - r$$

(5.10)

When we discretize $V(S,I,t)$, according to (4.2), note it as $v_{st}$, we can follow the steps in (4.3) to (4.7), and end with the ADI scheme

$$\left(1 + \frac{1}{2} \Delta t D_S\right) [v^{t+\Delta t}] = \left(1 - \frac{1}{2} \Delta t D_I\right) [v^{t+\frac{1}{2} \Delta t}]$$

(5.11)

$$\left(1 + \frac{1}{2} \Delta t D_I\right) [v^{t+\frac{1}{2} \Delta t}] = \left(1 - \frac{1}{2} \Delta t D_S\right) [v^{t}]$$

(5.12)

When considering the PDE (5.4) the term $\frac{1}{t} (S-I) \frac{\partial V}{\partial t}$ gives some computational challenges. Since the coefficient in front of $\frac{\partial V}{\partial t}$ is dependent on both the level of $S$ and $I$, it is not possible to construct a tridiagonal matrix as we can for the derivative terms with regards to $S$. As of this there are no matrix that can be inverted and multiplied to the right hand side for the implicit term, or simply multiplied to the $v$ array for the explicit term. The coefficient has in addition a $t^{-1}$ term that forces us to recalculate for every iteration. There are probably several ways to work around this problem and I have chosen to split the columns in $v$ so that every new array contains only $v$’s at one level of $S$. Every new array has one tridiagonal matrix associated with it, and by using the Thomas Algorithm on the implicit term and normal matrix calculations on the explicit term we can in a fast manner compute all arrays and merge them into $v$ again.

If we insert (5.9) and (5.10) in equation equation (5.11) and (5.12) we get

$$\left(1 + \frac{1}{2} \Delta t \left(\frac{1}{2} \sigma^2 S^2 \frac{\partial^2}{\partial S^2} + rS \frac{\partial}{\partial S}\right)\right) [v^{t+\Delta t}] = \left(1 - \frac{1}{2} \Delta t \left(\frac{1}{t} (S-I) \frac{\partial}{\partial I} - r\right)\right) [v^{t+\frac{1}{2} \Delta t}]$$

(5.13)
and
\[
\left( 1 + \frac{1}{2} \Delta t \left( \frac{1}{t} (S - I) \frac{\partial}{\partial t} - r \right) \right) [v^{t + \frac{1}{2}\Delta t}] = \left( 1 - \frac{1}{2} \Delta t \left( \frac{1}{2} \sigma^2 S^2 \frac{\partial^2}{\partial S^2} + r S \frac{\partial}{\partial S} \right) \right) [v^t]
\]
and if we discretize the differential terms we get
\[
\left( v^{t + \frac{1}{2}\Delta t}_S + \frac{1}{2} \left( \Delta t \frac{1}{2} \sigma^2 S^2 \frac{v^{t + \frac{1}{2}\Delta t}_S + v^{t - \frac{1}{2}\Delta t}_S}{\Delta S^2} - 2 v^{t + \frac{1}{2}\Delta t}_S + v^{t - \frac{1}{2}\Delta t}_S + r S \frac{v^{t + \frac{1}{2}\Delta t}_S - v^{t - \frac{1}{2}\Delta t}_S}{2 \Delta S} \right) \right)
\]
\[
= \left( v^{t + \frac{1}{2}\Delta t}_I - \frac{1}{2} \Delta t \left( S \frac{v^{t + \frac{1}{2}\Delta t}_I - v^{t - \frac{1}{2}\Delta t}_I}{2 \Delta I} - r v^{t + \frac{1}{2}\Delta t}_I \right) \right)
\]
and
\[
\left( v^{t + \frac{1}{2}\Delta t}_I + \frac{1}{2} \Delta t \left( S \frac{v^{t + \frac{1}{2}\Delta t}_I - v^{t - \frac{1}{2}\Delta t}_I}{2 \Delta I} - r v^{t + \frac{1}{2}\Delta t}_I \right) \right)
\]
\[
= \left( v^{t}_S - \frac{1}{2} \left( \Delta t \frac{1}{2} \sigma^2 S^2 \frac{v^{t + \frac{1}{2}\Delta t}_S + v^{t - \frac{1}{2}\Delta t}_S}{\Delta S^2} + 2 v^{t}_S + v^{t - \frac{1}{2}\Delta t}_S + r S \frac{v^{t + \frac{1}{2}\Delta t}_S - v^{t - \frac{1}{2}\Delta t}_S}{2 \Delta S} \right) \right)
\]
To make it usable and to construct our tridiagonal systems we sort the two equations on the vs.

\[
a_i V^{n+1}_{i+1} + b_i V^{n+1}_{i} + c_i V^{n+1}_{i-1} = \alpha_{ij} V^{n+\frac{1}{2}}_{ij} + \beta_{ij} V^{n-\frac{1}{2}}_{ij} + \gamma_{ij} V^{n-\frac{1}{2}}_{ij-1}
\]

\[
d_i V^{n+\frac{1}{2}}_{i+1} + e_i V^{n+\frac{1}{2}}_{i} + f_i V^{n+\frac{1}{2}}_{i-1} = \kappa_i V^{n}_{i+1} + \lambda_i V^{n}_{i} + \mu_i V^{n}_{i-1}
\]
where the coefficients are given as

\[
a_i = \frac{\sigma^2 S^2}{2 \Delta S^2} + \frac{r S}{2 \Delta S}, \quad \alpha_{ij} = -d_{ij}
\]

\[
b_i = \frac{2}{\Delta I} - \frac{\sigma^2 S^2}{\Delta S^2}, \quad \beta_{ij} = -e_i + \frac{4}{\Delta I}
\]

\[
c_i = \frac{\sigma^2 S^2}{\Delta S^2} - \frac{r S}{\Delta S}, \quad \gamma_{ij} = -f_{ij}
\]

\[
d_{ij} = \frac{S_i - I_i}{\Delta I}, \quad \kappa_i = -a_i
\]

\[
e_{ij} = \frac{2}{\Delta I} - r, \quad \lambda_i = -b + \frac{4}{\Delta I}
\]

\[
f_{ij} = -\frac{S_i - I_i}{\Delta I}, \quad \mu_i = -c_i
\]

This scheme has central differencing in both the I and the S direction, and by setting Neumann boundary conditions on all four sides, aiming for the highest degree of convergence, we get the coefficients for the discretized boundaries as below. The relationships between the implicit and the explicit
coefficients, e.g. $\alpha = -d_{ij}$, still hold.

\[
\begin{align*}
    a_0 &= \frac{r S_0}{\Delta S} \\
    b_0 &= \frac{2}{\Delta t} \frac{S_0}{\Delta S} - \frac{r S_0}{\Delta S} \\
    c_0 &= 0 \\
    a_I &= 0 \\
    b_I &= \frac{2}{\Delta t} + \frac{r S_I}{\Delta S} \\
    c_I &= -\frac{S_I}{\Delta S}
\end{align*}
\]

(5.14)

\[
\begin{align*}
    d_{i0} &= \frac{S_i - I_0}{t \Delta I} \\
    e_{i0} &= \frac{2}{\Delta t} \frac{S_i - I_0}{\Delta S} - r \\
    f_{i0} &= 0
\end{align*}
\]

When displaying the schema as full matrices it becomes clear why the $I$ direction can not be treated like the $S$ direction:

\[
\begin{bmatrix}
    b_I & c_I & 0 & 0 \\
    a_i & b_i & c_i & 0 \\
    0 & \ldots & \ldots & \ldots \\
    0 & 0 & a_0 & b_0
\end{bmatrix} \times \begin{bmatrix} V_{n+1}^{ij} \end{bmatrix} = \begin{bmatrix}
    e_{IJ} & f_{IJ} & 0 & 0 \\
    d_{Ij} & e_{Ij} & f_{Ij} & 0 \\
    0 & \ldots & \ldots & \ldots \\
    0 & 0 & d_{j0} & e_{j0}
\end{bmatrix} \begin{bmatrix} V_{n}^{ij} \end{bmatrix}^T
\]

(5.15)

and

\[
\begin{bmatrix}
    \beta_{IJ} & \gamma_{IJ} & 0 & 0 \\
    \alpha_{IJ} & \beta_{IJ} & \gamma_{IJ} & 0 \\
    0 & \ldots & \ldots & \ldots \\
    0 & 0 & \alpha_{I0} & \beta_{I0}
\end{bmatrix} \times \begin{bmatrix} V_{n+1}^{ij} \end{bmatrix} = \begin{bmatrix}
    \beta_{0J} & \gamma_{0J} & 0 & 0 \\
    \alpha_{0J} & \beta_{0j} & \gamma_{0j} & 0 \\
    0 & \ldots & \ldots & \ldots \\
    0 & 0 & \alpha_{00} & \beta_{00}
\end{bmatrix} \begin{bmatrix} V_{n0}^{ij} \end{bmatrix}^T
\]

(5.16)

### 5.1.3 Stability constraints

We know that for the ordinary Black – Scholes PDE the Crank Nicolson scheme is unconditionally stable regardless of the choice of $\Delta S$ and $\Delta t$. 

30
Ensuring the stability is crucial as we want to ensure that an increase in iterations, or a finer discretization in the \( I \) and \( S \) direction, leads to more accurate results. When writing the two equations on the multiplicator form, and applying the Gerschgorin circle theorem we find that the scheme is not unconditionally stable for the central difference ADI scheme.

\[
\begin{align*}
(1 + \frac{1}{2} \Delta t \bar{D}_S) [v^{t+\Delta t}] &= (1 - \frac{1}{2} \Delta t \bar{D}_I) [v^{t+\frac{1}{2} \Delta t}]
\end{align*}
\]

\[
\begin{align*}
(1 + \frac{1}{2} \Delta t \bar{D}_I) [v^{t+\frac{1}{2} \Delta t}] &= (1 - \frac{1}{2} \Delta t \bar{D}_S) [v^t]
\end{align*}
\]

\[
\begin{align*}
\frac{(1 + \frac{1}{2} \Delta t \bar{D}_S)}{(1 - \frac{1}{2} \Delta t \bar{D}_I)} [v^{t+\Delta t}] &= [v^{t+\frac{1}{2} \Delta t}]
\end{align*}
\]

\[
\begin{align*}
\frac{(1 + \frac{1}{2} \Delta t \bar{D}_I)}{(1 - \frac{1}{2} \Delta t \bar{D}_S)} [v^{t+\frac{1}{2} \Delta t}] &= [v^t]
\end{align*}
\]

The first fraction is the same as in the ordinary Black–Scholes PDE, so we know it is unconditionally stable. For the whole system to be stable any arbitrary norm of both fractions would have to be zero or negative, implying that any arbitrary norm, or eigenvalue, of \( \bar{D}_S \) and \( \bar{D}_I \) should be zero or negative. If we apply the Gerschgorin circle theorem on \( \bar{D}_S \) and \( \bar{D}_I \) we find the eigenvalues, \( \lambda_S^i \) and \( \lambda_I^i \), upper limits.

**Inner scheme**

\[
\begin{align*}
\max (\lambda_S^i) &= -\frac{\sigma^2 S_i^2}{\Delta S^2} + \| \frac{\sigma^2 S_i^2}{2 \Delta S^2} + \frac{r S_i}{2 \Delta S} + \| \frac{\sigma^2 S_i^2}{2 \Delta S^2} - \frac{r S_i}{2 \Delta S} \\
\max (\lambda_I^i) &= -r + \left| \frac{S_i - I_j}{t \Delta I} \right| + \left| \frac{S_i - I_j}{t \Delta I} \right|
\end{align*}
\]

**Upper \( S \) and \( I \) boundaries**

\[
\begin{align*}
\max (\lambda_S^i) &= \frac{r S_i}{\Delta S} + \left| \frac{r S_i}{\Delta S} \right|
\end{align*}
\]

\[
\begin{align*}
\max (\lambda_I^i) &= -\frac{S_i - I_j}{t \Delta I} - r + \left| \frac{S_i - I_j}{t \Delta I} \right| = (-r \quad | \quad S_i > I_j)
\end{align*}
\]
Lower $S$ and $I$ boundaries

\[
\max (\lambda^S_i) = \frac{r S_i}{\Delta S} \quad \text{and} \quad \left| \frac{r S_i}{\Delta S} \right| = 0
\]

\[
\max (\lambda^I_i) = \frac{S_i - I_j}{t \Delta I} - r + \left| -\frac{S_i - I_j}{t \Delta I} \right| = (-r \mid S_i < I_j)
\]

We see that unless we choose a very large $\Delta I$ we cannot ensure a stable solution for the inner scheme in the $I$ direction and the upper boundary for $S$. However, choosing a Dirichlet boundary condition on the upper $S$ bound and a one sided differencing like the one in the upper $I$ boundary, downwind, when $S_i > I_j$ and a one sided differencing like the one in the lower $I$ boundary, upwind, when $S_i < I_j$ would give a scheme unconditionally stable. One will lose some accuracy both compared to the central differencing in $I$ and in the upper boundary in $S$, but the scheme will have a guaranteed stability. The boundaries in $I$ would have to be changed so that they use a Neumann condition in the upper bound when $S_i < I_j$ and the lower bound when $S_i > I_j$. For all other boundaries in $I$ the Dirichlet condition can remain unchanged.

The unconditionally stable coefficients are then given as

\[
(d_{ij} | S_i \geq I_j) = \frac{S_i - I_j}{t \Delta I} \\
(e_{ij} | S_i \geq I_j) = \frac{2}{\Delta t} - \frac{S_i - I_j}{t \Delta I} - r \\
(f_{ij} | S_i \geq I_j) = 0 \\
(d_{ij} | S_i < I_j) = 0 \\
(e_{ij} | S_i < I_j) = \frac{2}{\Delta t} + \frac{S_i - I_j}{t \Delta I} - r \\
(f_{ij} | S_i < I_j) = -\frac{S_i - I_j}{t \Delta I}
\]

(5.17)
With the boundaries

\[
\begin{align*}
(d_0 | S_i \geq I_j) &= \frac{S_i - I_0}{\Delta I} \quad & (d_0 | S_i < I_j) &= 0 \\
(e_0 | S_i \geq I_j) &= \frac{2}{\Delta I} - \frac{S_i - I_0}{\Delta I} - r \quad & (e_0 | S_i < I_j) &= \frac{2}{\Delta I} \\
(f_0 | S_i \geq I_j) &= 0 \quad & (f_0 | S_i < I_j) &= 0 \\
(d_{iJ} | S_i \geq I_j) &= 0 \quad & (d_{iJ} | S_i < I_j) &= 0 \\
(e_{iJ} | S_i \geq I_j) &= \frac{2}{\Delta I} \quad & (e_{iJ} | S_i < I_j) &= \frac{2}{\Delta I} + \frac{S_i - I_j}{\Delta I} - r \\
(f_{iJ} | S_i \geq I_j) &= 0 \quad & (f_{iJ} | S_i < I_j) &= -\frac{S_i - I_j}{\Delta I} \\
\end{align*}
\]

\[
a_0 = \frac{r S_0}{\Delta S} \quad & a_I = 0 \\
b_0 = \frac{2}{\Delta I} - \frac{r S_0}{\Delta S} \quad & b_I = \frac{2}{\Delta I} \\
c_0 = 0 \quad & c_I = 0 \\
\]

(5.18)

### 5.1.4 Convergence

To test how well our scheme performs, and to see how fast it converges against a credible solution we test it against Monte Carlo simulations. Below you will find a table for different delta sizes and the computational time the calculations took. The Option is a one year call with \( \sigma = 0.2 \), \( K = 100 \), \( S = 100 \) and \( r = 15 \). 10k Monte Carlo paths of 100 observations pr day gave a price of 8.42 with a standard error of 0.10. The computational time was 11s on the same computer as the one used for running the ADI schemes.

<table>
<thead>
<tr>
<th>Price</th>
<th>( \Delta t ) ↓ (( \Delta S / \Delta I ))</th>
<th>20/20</th>
<th>2/2</th>
<th>0.2/0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>8.3856</td>
<td>8.4675</td>
<td>8.4117</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>16.4512</td>
<td>8.4729</td>
<td>8.4182</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>97.7853</td>
<td>8.4730</td>
<td>8.4148</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta t ) ↓ (( \Delta S / \Delta I ))</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.01</td>
</tr>
<tr>
<td>0.001</td>
</tr>
</tbody>
</table>

33
Price ($\Delta t$ fixed at 0.001)

\[
\begin{array}{ccc}
\Delta S \downarrow (\Delta I) & \rightarrow & 20 & 2 & 0.2 \\
20 & 97.7853 & 7.4858 & 7.4396 \\
2 & 21.9348 & 8.4730 & 8.4133 \\
0.2 & 21.3325 & 8.4785 & 8.4182 \\
\end{array}
\]

Computational time

\[
\begin{array}{ccc}
\Delta S \downarrow (\Delta I) & \rightarrow & 20 & 2 & 0.2 \\
20 & 0.18s & 0.47s & 70.42s \\
2 & 5.24s & 16.74s & 175.25s \\
0.2 & 103.57s & 228.71s & 975.75s \\
\end{array}
\]

We can see from the first table that the price converges very fast with a finer grid in the $\Delta S$ and $\Delta I$ direction. When choosing $\Delta S$ and $\Delta I$ as 2 and $\Delta t$ as 0.1 one gets a result within 0.5 standard errors from the Monte Carlo solution computed on under 0.01 seconds. This is about thousand times faster than the Monte Carlo, and with a much greater accuracy. From the second table we see that the scheme gains more from a finer grid in the $\Delta i$ direction than in the $\Delta S$ direction. This is rational as a variation in $S$ results in a smaller variation in $I$. Figure (4) shows the whole calculated surface for the asian option PDE. As $I \Rightarrow S$ when $t \Rightarrow 0$ the solution will be located in the diagonal between $[I_{\text{min}}, S_{\text{min}}]$ and $[I_{\text{max}}, S_{\text{max}}]$. The time is expanded to four years and the remaining parameters are the same as in the above tables. Figure (5) shows the comparison between the analytic Black – Scholes solution and the solution to the scheme when the boundary condition is set to $\max[S_T - K, 0]$. Using this boundary condition the price is no longer sensitive to the average price, $I$, and the price equals the Black – Scholes analytic price. As we see the max difference is located at the strike, with a maximum of -0.01.
Figure 4: Asian fixed strike call price with when $\sigma = 0.2$, $K = 100$, $r = 0.15$ and $T = 4y$.

Figure 5: Difference between Black–Scholes analytic price and the numeric scheme for a call price with when $\sigma = 0.2$, $K = 100$, $r = 0.05$ and $T = 1y$ and the boundary condition in $T$ equals $max[S_T - K, 0]$.
5.2 Heston Model

The Heston model has a different purpose than the Asian options. While the Asian options are contracts of different types, the Heston model tries to give a more correct price on ordinary options. In a normal Black–Scholes world one assumes the volatility between $t = 0$ and $t = T$ to be constant. This is of course not correct. The Heston model tries to incorporate the volatility of the volatility in the option prices by assuming that it has its own stochastic process. Since it is pricing plain vanilla options, the boundary condition in $T$ will look the same as for the ordinary Black–Scholes equation.

5.2.1 PDE

In the Heston model the process of the derivatives in (3.3) will still be stated as

$$df_i(\theta, t) = \mu_i(\theta, t) dt + \sigma_i(\theta, t) dB_t, \quad i = 1, 2$$  \hspace{1cm} (5.19)

The process of the underlying, $\theta_t$, are stated as

$$d\theta_t = r\theta_t dt + \sigma_t \theta_t dB_{1,t}$$  \hspace{1cm} (5.20)

where the risk adjusted volatility follow the process

$$d\sigma^2_t = \kappa(\eta - \sigma^2_t) dt + \sigma_t \sigma_t dB_{2,t}, \quad E[dB_{1,t}dB_{2,t}] = \rho dt$$  \hspace{1cm} (5.21)

If we apply Itô’s lemma on (5.19) we get the equation

$$df_i(\theta, v, t) = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \theta} + \frac{\partial f}{\partial v} dv + \frac{1}{2} v \theta^2 \frac{\partial^2 f}{\partial \theta^2} dt + \frac{1}{2} \sigma^2 v \frac{\partial^2 f}{\partial v^2} dt + \sigma v \theta \frac{\partial^2 f}{\partial v \partial \theta} dt$$  \hspace{1cm} (5.22)

Following the steps in (3.7) to (3.10) and changing $\theta$ with $S$ and $f$ with $V$ leads to the final Heston PDE

$$\frac{\partial V}{\partial t} + \frac{1}{2} v S^2 \frac{\partial^2 V}{\partial S^2} + \sigma v p S \frac{\partial^2 V}{\partial v \partial S} + \frac{1}{2} \sigma^2 v \frac{\partial^2 V}{\partial v^2} + r S \frac{\partial V}{\partial S} + \kappa(\eta - \sigma^2_t) \frac{\partial V}{\partial v} = r V$$  \hspace{1cm} (5.23)
5.2.2 ADI Schema

When using the multiplicators $D_vS$, $D_S$ and $D_v$ we can write equation (5.4) as

$$0 = \frac{\partial V}{\partial t} + D_vS V + D_S V + D_v V$$  \hspace{1cm} (5.24)

where

$$D_vS = \rho v \sigma S \frac{\partial}{\partial S} \frac{\partial V}{\partial v}$$  \hspace{1cm} (5.25)
$$D_S = \frac{1}{2} v S^2 \frac{\partial^2}{\partial S^2} + r S \frac{\partial}{\partial S}$$  \hspace{1cm} (5.26)
$$D_v = \frac{1}{2} \sigma^2 v \frac{\partial^2}{\partial v^2} + \kappa (\eta - v) \frac{\partial}{\partial v} - r$$  \hspace{1cm} (5.27)

Here we see that compared to the Asian option PDE we have a mixed derivatives term. The approximation of this term will be

$$\frac{V_{i+1,j+1} - V_{i+1,j-1} - V_{i-1,j+1} + V_{i-1,j-1}}{4 \Delta v \Delta S}$$

We split (5.24) in different times according to our preferred ADI scheme. Due to the complexity of the mixed derivatives term we want to keep it on the explicit side. We do this by stating it in the time $(t + \Delta t)$ only.

$$0 = \frac{\left[ v^{t+\Delta t} + v^{t+\Delta t} \right] - \left[ v^{t+\Delta t} + v^{t+\Delta t} \right] - \left[ v^t \right]}{\Delta t}$$
$$+ \frac{\left[ v^{t+\Delta t} - v^{t+\Delta t} \right]}{\Delta t} + \frac{\left[ v^{t+\Delta t} - v^{t+\Delta t} \right]}{\Delta t}$$
$$0 = \frac{\left[ v^{t+\Delta t} \right]}{\Delta t} + \frac{\left[ v^{t+\Delta t} \right]}{\Delta t} + \frac{\left[ v^{t+\Delta t} \right]}{\Delta t} + \frac{\left[ v^{t+\Delta t} \right]}{\Delta t}$$

Just like we did with the Asian ADI scheme, we are setting both (5.28) and (5.29) to zero, and we end with our Heston ADI scheme. Unlike the Asian it is not symmetrical as we have chosen to keep the $\bar{D}_vS$ term on the explicit side only. It is no problem to implement, but due to this term the scheme is
proximations on the differential terms we get When we insert (5.25) to (5.27) into (5.30) and (5.31) and use Taylor approximations on the differential terms we get

\[ \left(1 + \frac{1}{2} \Delta t \bar{D}_v + \Delta t \bar{D}_v S \right) [v^{t+\Delta t}] = \left(1 - \frac{1}{2} \Delta t \bar{D}_S \right) [v^{t+\frac{1}{2} \Delta t}] \]  
(5.30)

\[ \left(1 + \frac{1}{2} \Delta t \bar{D}_S \right) [v^{t+\frac{1}{2} \Delta t}] = \left(1 - \frac{1}{2} \Delta t \bar{D}_v \right) [v^t] \]  
(5.31)

When we insert (5.25) to (5.27) into (5.30) and (5.31) and use Taylor approximations on the differential terms we get

\[ \left( v_{v}^{t+\Delta t} + \frac{1}{2} \left( \Delta t \frac{\sigma v_{v}^{t+\Delta t} - v_{v}^{t+\Delta t} + r_{v} v_{v}^{t+\Delta t}}{\Delta v^2} + \kappa (\eta - v) \frac{\bar{v}_{v}^{t+\Delta t} - v_{v}^{t+\Delta t}}{2 \Delta v} - r_{v} v_{v}^{t+\Delta t} \right) \right) \]

\[ + \left( \rho v \sigma \bar{S} \frac{v_{S}^{t+\Delta t} - v_{S}^{t+\Delta t} + \bar{v}_{S}^{t+\Delta t} - \bar{v}_{S}^{t+\Delta t} + v_{S}^{t+\Delta t} - v_{S}^{t+\Delta t}}{4 \Delta S \Delta v} \right) \]

\[ = \left( v_{S}^{t+\frac{1}{2} \Delta t} - \frac{1}{2} \left( \Delta t \frac{v_{S}^{t+\frac{1}{2} \Delta t} + v_{S}^{t+\frac{1}{2} \Delta t} - v_{S}^{t+\frac{1}{2} \Delta t}}{2 \Delta S} - r_{v} v_{v}^{t+\frac{1}{2} \Delta t} \right) \right) \]

\[ \text{and} \]

\[ \left( v_{S}^{t+\frac{1}{2} \Delta t} - \frac{1}{2} \left( \Delta t \frac{v_{S}^{t+\frac{1}{2} \Delta t} + v_{S}^{t+\frac{1}{2} \Delta t} - v_{S}^{t+\frac{1}{2} \Delta t}}{2 \Delta S} - r_{v} v_{v}^{t+\frac{1}{2} \Delta t} \right) \right) \]

\[ = \left( v_{v}^{t} + \frac{1}{2} \left( \Delta t \frac{v_{v}^{t} + v_{v}^{t} - v_{v}^{t}}{2 \Delta v} + \kappa (\eta - v) \frac{v_{v}^{t} + v_{v}^{t} - v_{v}^{t}}{2 \Delta v} - r_{v} v_{v}^{t} \right) \right) \]

When we sort on the $v$ the two equations become

\[ h_i \left( V_{i+1,j+1}^{n+1} - V_{i+1,j-1}^{n+1} - V_{i-1,j+1}^{n+1} + V_{i-1,j-1}^{n+1} \right) \]

\[ + a_{ij} V_{i,j}^{n+1} + b_{ij} V_{i+1,j}^{n+1} + c_{i} V_{i-1,j}^{n+1} = \alpha_{ij} V_{i,j+1}^{n+1} + \beta_{ij} V_{i+1,j}^{n+1} + \gamma_{ij} V_{i,j-1}^{n+1} \]

\[ d_{ij} V_{i+1,j}^{n+1} + e_{ij} V_{i,j}^{n+1} + f_{ij} V_{i-1,j}^{n+1} = \kappa_{ij} V_{i+1,j}^{n+1} + \lambda_{ij} V_{i,j-1}^{n+1} + \mu_{ij} V_{i-1,j}^{n+1} \]

38
where the coefficients are given as

\[
\begin{align*}
h_{ij} &= \frac{\rho v_j \sigma S_i}{4 \Delta S \Delta v} \\
a_j &= \frac{\sigma^2 v_j}{2 \Delta v} + \frac{\kappa (\eta - v_j)}{2 \Delta v} \\
b_j &= \frac{\eta - \sigma^2 v_j - r}{2 \Delta v} \\
c_j &= \frac{\sigma^2 v_j}{2 \Delta v} - \frac{\kappa (\eta - v_j)}{2 \Delta v} \\
d_{ij} &= \frac{v_i \sigma^2 S_i}{4 \Delta S \Delta v} + \frac{r S_i}{2 \Delta S} \\
e_{ij} &= \frac{2}{\Delta v} - \frac{v_i \sigma^2 S_i}{\Delta S^2} \\
f_{ij} &= \frac{v_i \sigma^2 S_i}{2 \Delta S^2} - \frac{r S_i}{2 \Delta S} \\
\alpha_{ij} &= -d_{ij} \\
\beta_{ij} &= -e + \frac{4}{\Delta t} \\
\gamma_{ij} &= -f_{ij} \\
\kappa_j &= -a_i \\
\lambda_j &= -b + \frac{4}{\Delta t} \\
\mu_j &= -c_i \\
\end{align*}
\]

(5.32)

We see that \(h_{ij}, d_{ij}, e_{ij}\) and \(f_{ij}\), and their implicit opposites, are dependent on both the level of \(S\) and the level of \(v\). This is the same as we saw in the Asian PDE, and our solution were to split the \(V\) matrix into columns, one for each level of \(v\). Then there are a tridiagonal matrix for each of those vectors, and by using the TDMA algorithm it can easily be solved implicitly. For the explicit side this will impose no problem at all.

To ensure the highest order of accuracy Neumann boundary conditions are used. They are given as

\[
\begin{align*}
a_J &= 0 \\
b_J &= \frac{\eta - \sigma^2 v_j - r}{2 \Delta v} \\
c_J &= -\frac{\kappa (\eta - v_j)}{\Delta v} \\
d_{ij} &= 0 \\
e_{ij} &= \frac{2}{\Delta v} - \frac{r S_i}{\Delta S} \\
f_{ij} &= 0 \\
a_0 &= \frac{\kappa (\eta - v_j)}{\Delta v} \\
b_0 &= \frac{2}{\Delta v} - \frac{\kappa (\eta - v_j)}{\Delta v} - r \\
c_0 &= 0 \\
d_{0j} &= \frac{r S_i}{\Delta S} \\
e_{0j} &= \frac{2}{\Delta v} - \frac{r S_i}{\Delta S} \\
f_{0j} &= 0 \\
h_{ij} &= 0 \\
h_{i0} &= 0 \\
h_{0j} &= 0 \\
h_{i0} &= 0
\end{align*}
\]

(5.33)

5.2.3 Stability

As we saw for the Asian PDE, our central differencing gave stable results in the \(S\) direction, while nonstable results in the \(I\) direction, and we had to modify our scheme to guarantee stable output. For our Heston scheme
we have one extra multiplicator. This multiplicator, the mixed derivatives term, will impose stability issues.

\[
\left(1 + \frac{1}{2} \Delta t \tilde{D}_v + \Delta t \tilde{D}_v S\right) \left[ v^{t+\Delta t} \right] = \left(1 - \frac{1}{2} \Delta t \tilde{D}_S \right) \left[ v^{t+\Delta t} \right]
\]

\[
\left(1 + \frac{1}{2} \Delta t \tilde{D}_v \right) \left[ v^{t+\Delta t} \right] = \left(1 - \frac{1}{2} \Delta t \tilde{D}_v \right) \left[ v^{t} \right]
\]

\[
\frac{(1 + \frac{1}{2} \Delta t \tilde{D}_v + \Delta t \tilde{D}_v S)}{(1 - \frac{1}{2} \Delta t \tilde{D}_S)} \left[ v^{t+\Delta t} \right] = \left[ v^{t+\frac{1}{2} \Delta t} \right]
\]

\[
\frac{(1 + \frac{1}{2} \Delta t \tilde{D}_S)}{(1 - \frac{1}{2} \Delta t \tilde{D}_v)} \left[ v^{t+\Delta t} \right] = \left[ v^{t} \right]
\]

\[
\frac{(1 + \frac{1}{2} \Delta t \tilde{D}_v + \Delta t \tilde{D}_v S) \cdot (1 + \frac{1}{2} \Delta t \tilde{D}_S)}{(1 - \frac{1}{2} \Delta t \tilde{D}_v) \cdot (1 - \frac{1}{2} \Delta t \tilde{D}_S)} \left[ v^{t+\Delta t} \right] = \left[ v^{t} \right]
\]

We see that if the correlation, \( \rho \), is zero the term \( \Delta t \tilde{D}_v S \) will disappear, and we could prove stability by ensuring that \( \tilde{D}_v \) has only negative eigenvalues.

The first fraction will be the unconditionally stable Black – Scholes scheme. Again, by applying the Gerschgorin theorem we can examine the stability of these schemes. We start off by examining the eigenvalues of \( \tilde{D}_S \) and \( \tilde{D}_v \)

**Inner scheme**

\[
\text{max} (\lambda_v^i) = -\frac{\sigma^2 v_j}{\Delta v^2} - r + \frac{\sigma^2 v_j}{2 \Delta v^2} + \frac{\kappa (\eta - v_j)}{2 \Delta v} + \frac{\sigma^2 v_j}{2 \Delta v^2} - \frac{\kappa (\eta - v_j)}{2 \Delta v}
\]

\[
\text{max} (\lambda_S^i) = -\frac{v_j S_i^2}{\Delta S^2} + \frac{v_j S_i^2}{2 \Delta S^2} + \frac{r S_i}{2 \Delta S} + \frac{v_j S_i^2}{2 \Delta S^2} - \frac{r S_i}{2 \Delta S}
\]

**Upper S and v boundaries**

\[
\text{max} (\lambda_v^u) = \frac{\kappa (\eta - v_j)}{\Delta v} - r + \frac{\kappa (\eta - v_j)}{\Delta v}
\]

\[
\text{max} (\lambda_S^u) = \frac{r S_i}{\Delta S} + \frac{r S_i}{\Delta S} - r
\]
Lower $S$ and $v$ boundaries

\[
\max(\lambda^v_i) = -\frac{\kappa (\eta - v_j)}{\Delta v} - r + \left| \frac{\kappa (\eta - v_j)}{\Delta v} \right|
\]

\[
\max(\lambda^S_i) = -\frac{r S_i}{\Delta S} + \frac{r S_i}{\Delta S}
\]

It becomes clear that the inner scheme without $\bar{D}vS$ is stable. But at the upper boundaries for $S$, and as $\kappa(\eta - v_j)\Delta v^{-1}$ can be both positive and negative both boundaries for $v$ has to be considered again. As we see that for all relevant numbers $\max(\lambda^v_i)$ is the negative rate, and the upper boundary for the eigenvalues of $\bar{D}vS$ should be $r$. We can not represent multiplicator with a tridiagonal matrix, but if we split it in two it can be represented by two tridiagonal matrices.

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
\rho v_j \sigma S_i & 0 & 0 & 0 & 0 \\
0 & \rho v_j \sigma S_i & 0 & 0 & 0 \\
0 & 0 & \rho v_j \sigma S_i & 0 & 0 \\
0 & 0 & 0 & \rho v_j \sigma S_i & 0
\end{bmatrix} \times V_{ij+1} +
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
-\rho v_j \sigma S_i & 0 & 0 & 0 & 0 \\
0 & -\rho v_j \sigma S_i & 0 & 0 & 0 \\
0 & 0 & -\rho v_j \sigma S_i & 0 & 0 \\
0 & 0 & 0 & -\rho v_j \sigma S_i & 0
\end{bmatrix} \times V_{ij-1}
\]

Just by looking at these two matrices it becomes clear that the center of the Gerschgorin discs will be zero, and the radius will be the sum of the four coefficients scaled with the relationship between $v_{ij+1}$, $v_{ij}$, and $v_{ij-1}$. This radius might very well be larger than $r$. For the stable inner scheme without correlation we can construct boundary coefficients to ensure that the whole scheme is stable.

\[
\begin{align*}
(a_J|v_J \geq \eta) &= 0 & (a_J|v_J < \eta) &= 0 \\
(b_J|v_J \geq \eta) &= \frac{2\Delta t}{\Delta v} + \frac{\kappa(\eta - v_j)}{\Delta v} - r & (b_J|v_J < \eta) &= \frac{\kappa(\eta - v_j)}{\Delta v} \\
(c_J|v_J \geq \eta) &= -\frac{\kappa(\eta - v)}{\Delta v} & (c_J|v_J < \eta) &= 0 \\
(a_0|v_0 \geq \eta) &= 0 & (a_0|v_0 < \eta) &= \frac{\kappa(\eta - v_j)}{\Delta v} \\
(b_0|v_0 \geq \eta) &= \frac{2\Delta t}{\Delta v} & (b_0|v_0 < \eta) &= \frac{2\Delta t}{\Delta v} - \frac{\kappa(\eta - v)}{\Delta v} - r \\
(c_0|v_0 \geq \eta) &= 0 & (c_0|v_0 < \eta) &= 0 \\
(d_I) &= 0 & d_0 &= \frac{r S_j}{\Delta S} \\
(e_I) &= 1 & e_0 &= \frac{2\Delta t}{\Delta v} - \frac{r S_i}{\Delta S} \\
(f_I) &= 0 & f_0 &= 0
\end{align*}
\]

(5.34)
5.2.4 Convergence

As for the Asian options we test against a Monte Carlo estimation to see how fast our scheme converges against a credible solution. In below table the price of an call option priced with the parameters $S = 100$, $r = 0.05$, $v = 0.2$, $\sigma = 0.3$, $\eta = 0.2$, $\kappa = 1$, $T = 1y$, $\rho = 0.5$ and $K = 100$. As comparison we have a Monte Carlo simulation with one million paths with 10.000 steps. The result of the Monte Carlo simulation was a price of 19.72 with a standard error of 0.04. The Monte Carlo runtime on the same computer as the ADI schemes were 31:12 minutes.

<table>
<thead>
<tr>
<th>Price</th>
<th>$\Delta t \downarrow (\Delta S/\Delta v)$</th>
<th>20/0.04</th>
<th>2/0.004</th>
<th>0.2/0.0004</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>19.2772</td>
<td>17.6423</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>19.2819</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>19.2828</td>
<td>19.7192</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t \downarrow (\Delta S/\Delta v)$</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.01</td>
</tr>
<tr>
<td>0.001</td>
</tr>
</tbody>
</table>

One can see that when having large large grid sizes, with few nodes, it will be better to have few and larger time steps. The interesting result is that there seem to be an optimal ratio between the grid size in $v$ and $S$ direction and the time steps. If we consider 19.72 to be the true price, as it was the result of both the Monte Carlo and the fines ADI discretization, both the combination of $\Delta S$ and $\Delta v$ as 2 pared with a $\Delta t$ of 0.001 and pared with a $\Delta t$ of 0.01 gives a result within two standard errors. The computational time is however 70 times higher for the finest grid.

As shown previously the mixed derivatives term prevents the scheme from being unconditionally stable. When $\Delta S\Delta v$ becomes relatively large compared to $\Delta t$ the term grows exponentially and fast, and as we see in the above tables the scheme does not manage to produce a solution for these settings. Having a fine discretization in the $t$ - direction is therefor key to
keep the scheme stable. Below, in figure (6), the whole surface is shown. Unlike the Asian scheme the solution does not have to be located on the diagonal. In figure (7) the difference between the analytic solution to the Black – Scholes model is displayed. When setting \( \kappa \) and \( \sigma \) to zero the volatility is no longer stochastic and the PDE follows the traditional Black – Scholes PDE. As we see, due to our boundary conditions the large deviation comes in \( S_{\text{max}} \) pared with \( v_{\text{max}} \). It also shows that the difference decays fast and is very small on the lower diagonal.

Price (\( \Delta t \) fixed at 0.0001)

\[
\Delta S \downarrow (\Delta v) \rightarrow 0.04 \quad 0.004 \quad 0.0004 \\
20 \quad 19.2829 \quad 19.2814 \quad 19.2813 \\
2 \quad 19.7212 \quad 19.7193 \quad 19.7192 \\
0.2 \quad 19.7254 \quad N/A \quad N/A
\]

Computational time

\[
\Delta S \downarrow (\Delta v) \rightarrow 0.04 \quad 0.004 \quad 0.0004 \\
20 \quad 1.37s \quad 50.86s \quad 961.81s \\
2 \quad 4.09s \quad 171.31 \quad 2414.40s \\
0.2 \quad 148.93s \quad N/A \quad N/A
\]
Figure 6: Call option price with $T = 1y$, $\sigma = 0.3$, $\kappa = 1$, $\eta = 0.2$, $\rho = 0.5$, $r = 0.05$. $v_0 = 0.2$

Figure 7: Difference in price between the analytic solution of Black–Scholes and the numeric Heston scheme for a call option price with $T = 1y$, $\sigma = 0$, $\kappa = 0$, $\rho = 0$, $r = 0.05$. $v_0 = 0.2$
6 Conclusion

In this thesis I have shown that the finite difference scheme, ADI, computes accurate results faster than the frequently used Monte Carlo estimation. I have shown two applications of the ADI on two PDEs. The first PDE describes Asian options, options where the payout rely on the arithmetic average on the underlying price. The second equation is the Heston PDE which is an extension of the Black – Scholes PDE where the volatility is given its own stochastic process. I also show how to implement the mixed derivative term that follows from the correlation of the volatility’s- and the underlying stock’s stochastic process, and demonstrate how it leads to stability problems when the discretization in the t - direction becomes large related to the discretization in the vol and underlying stock - direction. For Asian options the ADI scheme produced prices within one standard deviation from the Monte Carlo simulated prices in less than one thousand of the time of the Monte Carlo simulation. For the Heston model ADI computed prices within one standard deviation in less than one hundred of the Monte Carlo runtime. When choosing a discretization that fits the given problem and considering stability issues when defining the boundary conditions the ADI scheme produces stable results.
7 Matlab Code

7.1 Asian ADI

7.1.1 Asian.m

function [aa] = Asian(years, vol, rate, strike, discS, discI, type)
  t_0 = cputime;
  %Set the discretization parameters
  Idisc = discI;
  Sdisc = discS;
  Smax = strike*2;
  Imax = strike*2;
  %Calculate the deltas
  dt = 0.01;
  dS = Smax / (Sdisc-1);
  dI = Imax / (Idisc-1);
  %Calculate I_array and S_array
  S = zeros(discS,1);
  I = zeros(discI,1);
  for i = 1:Sdisc
    S(1 + Sdisc - i) = (i-1)*dS;
  end
  for i = 1:Idisc
    I(1 + Idisc - i) = (i-1)*dI;
  end
  %boundary:
  a = zeros(discS,discI);
  if type == 1 %type 1 = fixed strike call
    for i = 1:discS
      for j = 1:discI
        if I(j) > strike
          a(i,j) = I(j) - strike;
        else
          a(i,j) = 0;
        end
      end
    end
  elseif type == 2 %type 2 = fixed strike put
    for i = 1:discS
      for j = 1:discI
        if I(j) < strike
          a(i,j) = I(j) - strike;
        else
          a(i,j) = 0;
        end
      end
    end
  end
  end
  t_0 = cputime - t_0;
  disp(t_0);
end
a(i,j) = strike - I(j);
else
a(i,j) = 0;
end
end
elseif type == 3 %type 3 = floating strike call
for i = 1:discS
    for j = 1:discI
        if S(i) > I(j)
            a(i,j) = S(i) - I(j);
        else
            a(i,j) = 0;
        end
    end
end
elseif type == 4 %type 4 = floating strike put
for i = 1:discS
    for j = 1:discI
        if I(j) > S(i)
            a(i,j) = I(j) - S(i);
        else
            a(i,j) = 0;
        end
    end
end
end %if

%Define the tridiagonal matrices for the S-direction
tridiagS_exp = zeros(Sdisc,Sdisc);
tridiagS_imp = zeros(Sdisc,Sdisc);

%Create the tridiagonal matrices for the S-direction (explicit)
for i = 1:Sdisc
    if i == 1
        %tridiagS_exp(i,i) = 2/dt;
        tridiagS_exp(i,i) = (2/(dt) - rate + rate * S(i) / (dS));
        tridiagS_exp(i,i+1) = (-rate * S(i) / (dS));
    elseif i == Sdisc
        %tridiagS_exp(i,i) = 2/dt;
        tridiagS_exp(i,i-1) = (rate * S(i) / (dS));
        tridiagS_exp(i,i) = (2/(dt) - rate - rate * S(i) / (dS));
    else
        %tridiagS_exp(i,i) = 2/dt;
        tridiagS_exp(i,i-1) = (rate * S(i) / (dS));
        tridiagS_exp(i,i) = (2/(dt) - rate - rate * S(i) / (dS));
    end
end
tridiagS_exp(i,i-1) = (0.5*vol^2*S(i)^2/(dS^2) + rate * S(i) / (2*dS));
tridiagS_exp(i,i) = (2 /(dt) - vol^2*S(i)^2/(dS^2) - rate);
tridiagS_exp(i,i+1) = (0.5*vol^2*S(i)^2/(dS^2) - rate * S(i) / (2*dS));
end %i
end %i

%Create the tridiagonal matrices for the S-direction (implicit)
for i = 1:Sdisc
    if i == 1
        %tridiagS_imp(i,i) = 2/dt;
        tridiagS_imp(i,i) = (2 /(dt) + rate - rate * S(i) / (dS));
        tridiagS_imp(i,i+1) = (+ rate * S(i) / (dS));
    elseif i == Sdisc
        %tridiagS_imp(i,i) = 2/dt;
        tridiagS_imp(i,i-1) = (- rate * S(i) / (dS));
        tridiagS_imp(i,i) = (2 /(dt) + rate + rate * S(i) / (dS));
    else
        tridiagS_imp(i,i-1) = (-0.5*vol^2*S(i)^2/(dS^2) - rate * S(i) / (2*dS));
        tridiagS_imp(i,i) = (2 /(dt) + vol^2*S(i)^2/(dS^2) + rate);
        tridiagS_imp(i,i+1) = (-0.5*vol^2*S(i)^2/(dS^2) + rate * S(i) / (2*dS));
    end %if
end %i

temp = a';
titers = years / dt;
for i = 1:titers
    temp = (tridiagS_exp * temp)';
    temp = IimplicitAsian(temp, Idisc, Sdisc, years - (i-0.5)*dt, S, I, dI, dt);
    temp = IexplicitAsian(temp, Idisc, Sdisc, years - (i-0.5)*dt, S, I, dI, dt)';
    temp = tridiagS_imp \ temp;
end
surf(I, S, temp);
timer = cputime - t_0
out = temp((discS-1)/2 + 1,(discI-1)/2 + 1)
aa = temp;
end %function

7.1.2 IexplicitAsian.m

function utdata = IexplicitAsian(V, discI, discS, t, S, I, dI, dt)
Vspl = zeros(discI, discS);
a = zeros(discI);
b = zeros(discI);
\[ c = \text{zeros}(\text{discI}); \]

%Create the tridiagonal matrices for the I-direction (explicit)
for \( i = 1: \text{discS} \)
  for \( j = 1: \text{discI} \)
    if \( j == 1 \)
      if \( S(i) < I(j) \)
        \( b(j) = 2 / (dt) + t^{-1} * (S(i) - I(j)) / (dI); \)
        \( c(j) = -t^{-1} * (S(i) - I(j)) / (dI); \)
      \else
        \( b(j) = 2 / dt; \)
      \end
    \else
      if \( S(i) < I(j) \)
        \( b(j) = 2 / dt; \)
      \else % One sided differencing
        \( a(j) = t^{-1} * (S(i) - I(j)) / (dI); \)
        \( b(j) = 2 / (dt) - t^{-1} * (S(i) - I(j)) / (dI); \)
      \end
    \else
      %Upwind/Downwind
      if \( S(i) < I(j) \)
        \( a(j) = 0; \)
        \( b(j) = 2 / (dt) + t^{-1} * (S(i) - I(j)) / (dI); \)
        \( c(j) = -t^{-1} * (S(i) - I(j)) / (dI); \)
      \else
        \( a(j) = t^{-1} * (S(i) - I(j)) / (dI); \)
        \( b(j) = 2 / (dt) - t^{-1} * (S(i) - I(j)) / (dI); \)
        \( c(j) = 0; \)
      \end
    \end
  \end
  for \( k = 1: \text{discI} \)
    if \( k == 1 \)
      \( Vspl(k,i) = V(k,i) * b(k) + V(k+1,i) * c(k); \)
    \else
      if \( k == \text{discI} \)
        \( Vspl(k,i) = V(k,i) * b(k) + V(k-1,i) * a(k); \)
      \else
        \( Vspl(k,i) = V(k,i) * b(k) + V(k+1,i) * c(k) + V(k-1,i) * a(k); \)
      \end
  \end
\]
Vspl(k,i) = V(k,i)*b(k) + V(k-1,i)*a(k) + V(k+1,i)*c(k);
end %if
end %i
end %i
utdata = Vspl;
end %function

7.1.3 IimplicitAsian.m

function utdata = IimplicitAsian(V, discI, discS, t, S, I, dI, dt)
Vout = zeros(discI, discS);
a = zeros(discI);
b = zeros(discS);
c = zeros(discS);
x = zeros(discI);
%Create the tridiagonal matrices for the I-direction (implicit)
for i = 1:discS
for j = 1:discI
if j == 1
if S(i) < I(j) %One sided differencing
b(j) = 2 / (dt) - t^-1 *(S(i)-I(j))/(dI);
c(j) = t^-1 *(S(i)-I(j))/(dI);
else
b(j) = 2/dt;
end
elseif j == discI
if S(i) <I(j)
b(j) = 2/dt;
else % One sided differencing
a(j) = -t^-1 *(S(i) - I(j))/(dI);
b(j) = 2 / (dt) + t^-1 *(S(i) - I(j))/(dI);
end
else
%Upwind/Downwind
if S(i) <I(j)
a(j) = 0;
b(j) = 2 / (dt) - t^-1 *(S(i) - I(j))/(dI);
c(j) = t^-1 *(S(i) - I(j))/(dI);
else
a(j) = -t^-1 *(S(i) - I(j))/(dI);
end
end
end
end %function
\[ b(j) = \frac{2}{(dt)} + t^{-1} \cdot (S(i) - I(j))/(dI); \]
\[ c(j) = 0; \]
end

% Central differencing:
% \[ a(j) = -t^{-1} \cdot (S(i) - S(j))/(2 \cdot dI); \]
% \[ b(j) = \frac{2}{(dt)}; \]
% \[ c(j) = t^{-1} \cdot (S(i) - S(j))/(2 \cdot dI); \]
end % if
end % j

for k = 1:discI
  x(k) = V(k,i);
end % k

c(1) = c(1) / b(1);
x(1) = x(1) / b(1);
for l = 2:discI-1
  temp = b(l) - a(l) \cdot c(l-1);
  c(l) = c(l) / temp;
  x(l) = (x(l) - a(l) \cdot x(l-1))/temp;
end % l
x(discI) = (x(discI) - a(discI) \cdot x(discI-1))/(b(discI) - a(discI) \cdot c(discI-1));
x(discI) = x(discI);
for l = discI-1:-1:1
  x(l) = x(l) - c(l) \cdot x(l + 1);
end % l
for k = 1:discI
  Vout(k,i) = x(k);
end % k
end % i
end % function

7.2 Heston ADI

7.2.1 Heston.m

function \[ [aa] = Heston(years, sigma, kappa, eta, rho, rate, strike, spot, vol_now, disc, type) \]
function \[ [aa, ab, ac] = Heston(years, sigma, kappa, eta, rho, rate, strike, spot, vol_now, discS, discv, type) \]
t_0 = cputime;
% Set the discretization parameters
Smax = spot*2;
\[ v_{max} = vol\_now*2; \]
%Calculate the deltas
dt = 0.01;
dS = Smax / (discS-1)
dv = vmax / (discv-1)
%Calculate I_array and S_array
S = zeros(discS,1);
v = zeros(discv,1);
for i = 1:discS
    S(1 + discS - i) = (i-1)*dS;
end
for i = 1:discv
    v(1 + discv - i) = (i-1)*dv;
end
%boundary:
a = zeros(discv,discS);
if type == 1 %type1 = fixed strike call
    for i = 1:discv
        for j = 1:discS
            if S(j) > strike
                a(i,j) = S(j)- strike;
            else
                a(i,j) = 0;
            end
        end
    end
elseif type == 2 %type 2 = fixed strike put
    for i = 1:discv
        for j = 1:discS
            if S(j) < strike
                a(i,j) = strike - S(j);
            else
                a(i,j) = 0;
            end
        end
    end
end
%Define the tridiagonal matrices for the v-direction
tridiagv_exp = zeros(discv,discv);
tridiagv_imp = zeros(discv,discv);
%Create the tridiagonal matrices for the v-direction (explicit)
for i = 1:discv
    for j = 1:discS
        if S(j) > strike
            a(i,j) = S(j)- strike;
        else
            a(i,j) = 0;
        end
    end
end
%if
%Define the tridiagonal matrices for the v-direction
tridiagv_exp = zeros(discv,discv);
tridiagv_imp = zeros(discv,discv);
%Create the tridiagonal matrices for the v-direction (explicit)
for i = 1:discv

52
if i == 1
    if v(i) > eta
        tridiagv_exp(i,i) = (2/(dt) - rate + kappa*(eta - v(i)) / (dv)); %Dirichlet
        tridiagv_exp(i,i+1) = -kappa * (eta - v(i))/(dv);
    else
        tridiagv_exp(i,i) = 2/(dt); % Neumann
    end
elseif i == discv
    if v(i) > eta
        tridiagv_exp(i,i) = 2/(dt); % Neumann
    else
        tridiagv_exp(i,i-1) = kappa * (eta - v(i))/ (dv); %Dirichlet
        tridiagv_exp(i,i) = (2 /(dt) - rate - kappa * (eta - v(i))/ (dv));
    end
else
    tridiagv_exp(i,i-1) = (0.5*sigma^2*v(i)/(dv^2) + kappa*(eta - v(i)) / (2*dv));
    tridiagv_exp(i,i) = (2 /(dt) - sigma^2*v(i)/(dv^2) - rate);
    tridiagv_exp(i,i+1) = (0.5*sigma^2*v(i)/(dv^2) - kappa*(eta - v(i)) / (2*dv));
end %if
end %i

%Create the tridiagonal matrices for the v-direction (implicit)
for i = 1:discv
    if i == 1
        if v(i) > eta
            tridiagv_imp(i,i) = (2 /(dt) + rate - kappa*(eta - v(i)) / (dv)); %Dirichlet
            tridiagv_imp(i,i+1) = ( + kappa*(eta - v(i)) / (dv));
        else
            tridiagv_imp(i,i) = 2/(dt); %Neumann
        end
    elseif i == discv
        if v(i) > eta
            tridiagv_imp(i,i) = 2/(dt); %Neumann
        else
            tridiagv_imp(i,i-1) = ( - kappa*(eta - v(i)) / (dv)); %Dirichlet
            tridiagv_imp(i,i) = (2 /(dt) + rate + kappa*(eta - v(i)) / (dv));
        end
    else
        tridiagv_imp(i,i-1) = -(0.5*sigma^2*v(i)/(dv^2) + kappa*(eta - v(i)) / (2*dv));
        tridiagv_imp(i,i) = (2 /(dt) + sigma^2*v(i)/(dv^2) + rate);
        tridiagv_imp(i,i+1) = -(0.5*sigma^2*v(i)/(dv^2) - kappa*(eta - v(i)) / (2*dv));
    end %if
temp = a;
titers = years / dt;
%figure;
%h1 = surf(a);
for i = 1:titers;
    temp2 = mixedderivexplicit2(rho, sigma, S, v, temp, discS, discv, dS, dv, dt)';
    temp = (tridiagv_exp * temp)';
    temp = temp + temp2;
    temp = IimplicitHeston(temp, discS, discv, S, v, dS, dt, rate);
    temp = IexplicitHeston(temp, discS, discv, S, v, dS, dt, rate)';
    temp = (tridiagv_imp \ temp);
    %temp = upperIbound(temp, disc);
end
surface(S, v, temp);
timer = cputime - t_0
out = temp((discv-1)/2 + 1,(discS-1)/2 + 1)
aa = temp;
ab = S;
ac = v;
end %function

function [ utdata ] = upperIbound(V, disc)
for i = 1:disc
    V(1,i) = 3*V(2,i) - 3*V(3,i) + V(4,i);
end %i
for i = 1:disc
    V(1,i) = 3*V(2,i) - 3*V(3,i) + V(4,i);
end %i
utdata = V;
end

7.2.2 IexplicitHeston.m

function utdata = IexplicitHeston(V, discS, discv, S, v, dS, dt, rate)
Vspl = zeros(discS, discv);
a = zeros(discS,1);
b = zeros(discS,1);
c = zeros(discS,1);
%Create the tridiagonal matrices for the S-direction (explicit)
for i = 1:discv
  for j = 1:discS
    if j == 1
      \( b(j) = \frac{2}{dt}; \)
      \( b(j) = \frac{2}{dt} + \frac{\text{rate}\cdot S(j)}{dS}; \)
      \( c(j) = -\frac{\text{rate}\cdot S(j)}{dS}; \)
    elseif j == discS
      \( b(j) = \frac{2}{dt}; \)
      \( a(j) = \frac{\text{rate}\cdot S(j)}{dS}; \)
      \( b(j) = \frac{2}{dt} - \frac{\text{rate}\cdot S(j)}{dS}; \)
    else
      % Central differencing:
      \( a(j) = \frac{v(i)\cdot S(j)^2}{2\cdot dS^2} + \frac{\text{rate}\cdot S(j)}{2\cdot dS}; \)
      \( b(j) = \frac{2}{dt} - \frac{v(i)\cdot S(j)^2}{dS^2}; \)
      \( c(j) = \frac{v(i)\cdot S(j)^2}{2\cdot dS^2} - \frac{\text{rate}\cdot S(j)}{2\cdot dS}; \)
    end %if
  end %j
  for k = 1:discS
    if k == 1
      \( V_{spl}(k,i) = V(k,i)\cdot b(k) + V(k+1,i)\cdot c(k); \)
    elseif k == discS
      \( V_{spl}(k,i) = V(k,i)\cdot b(k) + V(k-1,i)\cdot a(k); \)
    else
      \( V_{spl}(k,i) = V(k,i)\cdot b(k) + V(k-1,i)\cdot a(k) + V(k+1,i)\cdot c(k); \)
    end %if
  end %i
end %i
utdata = V_{spl};
end %function
x = zeros(discS,1);

% Create the tridiagonal matrices for the S-direction (implicit)
for i = 1:discv
    for j = 1:discS
        if j == 1
            b(j) = 2/(dt);
            c(j) = rate*S(j)/(dS);
        elseif j == discS
            b(j) = 2/dt;
            a(j) = -rate*S(j)/(dS);
            b(j) = 2/(dt) + rate*S(j)/(dS);
        else
            % Central differencing:
            a(j) = -v(i)*S(j)^2/(2*dS^2) - rate*S(j)/(2*dS);
            b(j) = 2/(dt) + v(i)*S(j)^2/(dS^2);
            c(j) = -v(i)*S(j)^2/(2*dS^2) + rate*S(j)/(2*dS);
        end
    end
    for k = 1:discS
        x(k) = V(k,i);
    end
    c(1) = c(1)/b(1);
    x(1) = x(1)/b(1);
    for l = 2:discS-1
        temp = b(l) - a(l)*c(l-1);
        c(l) = c(l)/temp;
        x(l) = (x(l) - a(l)*x(l-1))/temp;
    end
    x(discS) = (x(discS) - a(discS)*x(discS-1))/(b(discS) - a(discS)*c(discS-1));
    x(discS) = x(discS);
    for l = discS-1:-1:1
        x(l) = x(l) - c(l)*x(l+1);
    end
    for k = 1:discS
        Vout(k,i) = x(k);
    end
end
end

utdata = Vout;
7.2.4 mixedderivexplicit.m

function utdata = mixedderivexplicit2(rho, sigma, S, v, temp, discS, discv, dS, dv, dt)
temp2 = zeros(discv,discS);
for i = 2:discv-1
    for j = 2:discS-1
        temp2(i,j) = rho*sigma*S(j)*v(i)*(temp(i-1, j-1) - temp(i-1, j+1) ...
            - temp(i+1, j-1) + temp(i+1, j+1))/(4*dS*dv);
    end %j
end %i
utdata = 2*temp2;
end
References


