NTNU – Trondheim
Norwegian University of
Science and Technology

Pricing Put Options using Heston's Stochastic Volatility Model

Andreas Brandsøy Våg

Master of Science in Physics and Mathematics
Submission date: July 2013
Supervisor: Espen Robstad Jakobsen, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Pricing Put Options using Heston's Stochastic Volatility Model

Andreas Brandsøy Våg

July 12, 2013

Abstract

The Heston model is a partial differential equation which is used to price options and is a further developed version of the more famous Black-Scholes equation. Heston considers stochastic volatility which results in an extra variable and a more complex equation. This paper contains numerical solutions of the Heston model for the European and American option using finite difference and element methods. First the equation will be derived followed by numerical solutions for both European and American options with a finite difference method. The equation is then expressed in weak form and solved using a finite element method. Mathematical analysis concerning stability and uniqueness of the solution will be given for the finite element solver in the European case. All solvers are implemented using Matlab.

Sammendrag

Heston modellen er en partiell differensialligning som brukes til å prise opsjoner og er en videreutviklet utgave av den mer berømte Black-Scholes ligningen. Heston betrakter stokastisk volatilitet som resulterer i en ekstra variabel og en mer komplisert ligning. Denne oppgaven inneholder numeriske løsninger av Heston modellen for europeiske og amerikanske opsjoner med endelig differanse og element metoder. Først vil ligningen bli utledet etterfulgt av en numerisk løsning for både europeiske og amerikanske opsjoner med en endelig differanse metode. Ligningen uttrykkes deretter på svak form og løses ved hjelp av en endelig element metode. Matematisk analyse angående stabilitet og entydighet av løsningen vil bli gitt for elementmetoden når vi betrakter europeiske opsjoner. Alle løsere er implementert i Matlab.

Preface

This master's thesis marks the end of five years of mathematics studies at NTNU. The work started in the beginning of February 2013, and was completed 12th of July 2013. I am deeply grateful to my supervisor Professor Espen Robstad Jakobsen for his support during my work. Especially for having the patience and taking the time to make me fully understand the many challenges along the way.

Contents

Chapter 1

Introduction

1.1 Background

Before the Black-Scholes model was published in 1973 there were no exact way of pricing options. An option is a contract that gives its owner the right to sell (or buy depending on the type) a fixed number of shares at a fixed price at a certain date. But what would be a rational value for such a contract? Black, Scholes and Merton defined this rational value as follows

- If you instead of buying an option for this rational value choose to place the same amount of money in stock and bond, you would be able to yield the same payoff using a self-financing strategy as if the option had been purchased.
- If the option were offered at any price other than this rational value, there would be an opportunity of *arbitrage*, i.e for profit without an accompanying risk of loss.

By perfectly hedging the option and thereby eliminating all risk they were able to derive partial differential equation known as the Black-Scholes equation. This led to a boom in option trading and in 1987 the world's stock markets crashed leaving the blame on the equation.

The Black-Scholes model has its limitations such as underestimation of extreme moves, constant volatility and assuming lognormal returns. There are several models taking these weaknesses into account and the Heston model is one of them. The Heston model is a stochastic volatility model among others, and is constructed under the assumption that the volatility follows a random process. In this paper we will take a closer look at the derivation of the Heston model and some numerical solvers. There are many examples of numerical solvers using the ADI method when considering finite difference methods, and also some examples of the finite element method which will be referred to throughout this paper. In this paper the first numeric solver is a straight forward finite difference method with the implicit

Euler scheme as opposed to the ADI scheme used in [8]. Also the finite element method will be implemented, but without the log transformation used in [5]. The options that will be considered are the European and American vanilla put where the European put will be further investigated with some mathematical analysis when applying the element method to the European put.

1.2 Terms and expressions

This section contains a list of expressions that are used throughout this paper which require a brief explanation

- Vanilla option - An option with no complex financial structure.
- Put option - The put option gives one the right to sell the underlying asset at a price K, while the call option gives the right to buy.
- Payoff function (P_0) - At maturity time the amount of money earned is given by $P_0 = (K - S, 0)^+ = max(K - S, 0)$ for a put option.
- European option - Can be exercised at the maturity time.
- American option - Can be exercised at any time before the maturity time and is therefore more expensive than the European option.
- Volatility - A measure for variation of price of the underlying asset.

Plan of the thesis

- **Chapter 2.1**. We derive the Heston model from two stochastic differential equations modeling stock price and volatility. The main parts are applying Itô's formula and constructing a portfolio which should yield the same pay off as a risk-free asset.

- **Chapter 2.2**. We state the American condition and formulate the problem for the American put option.

- **Chapter 2.3**. Boundary conditions are stated and explained by investigating the Heston model.

- **Chapter 3.1**. The Heston model is discretized using central difference and the implicit Euler method and the system of equations are then presented in matrix form.

- **Chapter 3.2**. The boundary conditions are transferred to the discrete case and the strategy for implementing each one of them is given in detail.

- **Chapter 3.3** We derive the truncation error for the finite difference scheme in order to obtain the order of convergence.

- **Chapter 3.4** The Brennan Schwartz algorithm is stated and its advantages when using Matlab as the programming language.

- **Chapter 3.5**. Numerical results are given for both the American and European option. Some challenges with the finite difference approach are also discussed.

- **Chapter 4.1**. The equation is expressed in matrix form followed by the standard approach for obtaining the bilinear form a, namely multiplying by a test function and use integration by parts. We then define a space V based on the bilinear form enabling us to express the weak formulation of the problem.

- **Chapter 4.2**. The discrete space V_h is introduced and we explain how to discretize the problem and express it in matrix form. We then apply the implicit Euler scheme and give a brief explanation of how to construct the mass and stiffness matrices.

- **Chapter 4.3**. Proofs of continuity and energy stability when considering the finite element method for the European option.

- **Chapter 4.4**. Numerical results for the European option when applying the element method.

- **Chapter 5.1**. Obtaining the weak formulation for the American option by introducing a new space $\mathcal{K} \subset V$ where the American condition is fulfilled.

- **Chapter 5.2-5.3**. Discretization and numerical results as in the previous chapter.
- **Chapter 6**. Comparing the methods and summarizing the results. We also discuss further work and alternative approaches for solving the Heston model.

Chapter 2

Heston's model

2.1 Deriving the model for the European option

The Heston model is a stochastic volatility model which we assume follows the diffusion

$$dS(t) = cS(t)dt + \sqrt{v(t)}S(t)dB_S \tag{2.1.1}$$

where B is Brownian motion, S is the price of the underlying asset and v is the volatility. We assume that the volatility follows an Ornstein-Uhlenbeck process (Appendix 7.2),

$$d\sqrt{v(t)} = -\beta\sqrt{v(t)}dt + \delta dB_v, \tag{2.1.2}$$

where dB_v has correlation ρ with dB_S

$$dB_v dB_s = \rho dt \tag{2.1.3}$$

(Heston, [2]). To express (2.1.2) as dv we apply Itô's lemma (Appendix 7.1) with $f(u) = u^2$ $(u = \sqrt{v(t)})$ and the result is

$$\begin{aligned} df(u) = dv(t) =& [-\beta u\frac{df}{du} + \frac{1}{2}\delta^2\frac{d^2f}{du^2}]dt + \delta\frac{df}{du}dB_v \\ \Rightarrow dv(t) =& [-\beta u \cdot 2u + \frac{1}{2}\delta^2 \cdot 2]dt + \delta \cdot 2udB_v \\ dv(t) =& [\delta^2 - 2\beta u^2]dt + 2\delta udB_v. \end{aligned}$$

This can be written as the CIR process (Appendix 7.3)

$$dv(t) = \kappa[\theta - v(t)]dt + \sigma\sqrt{v(t)}dB_v. \tag{2.1.4}$$

The parameters in (2.1.1) and (2.1.4) describes the following

- c - rate of return of the asset
- θ - long variance, the expected value of $v(t)$ tends to θ as t tends to infinity
- κ - rate at which $v(t)$ reverts to θ.
- σ - volatility of volatility, determines the variance of $v(t)$

In this case we use the no-arbitrage argument from the original Black-Scholes model, meaning that the pricing function P is obtained by constructing a hedged portfolio of assets. We consider a portfolio containing the option being priced, denote the value by

$$P_1 = P(S, v, t)$$

and a quantity b_t of another asset whose value depends on volatility, denote the value by

$$P_2 = P(S, v, t).$$

In addition to this we also include a_t shares of the underlying asset. The value of the portfolio is simply the sum of the different assets

$$c_t = P_1 + a_t S_t + b_t P_2,$$

and expressed in differential form

$$dc_t = dP_1 + da_t S_t + db_t P_2 + a_t dS_t + b_t dP_2. \tag{2.1.5}$$

The portfolio is self-financing, meaning that no money goes in or out of the system. If we sell stocks we buy the other asset and vice versa, giving $da_t S_t = -db_t P_2$ which reduces (2.1.5) to

$$dc_t = dP_1 + a_t dS_t + b_t dP_2. \tag{2.1.6}$$

We write out the expression for dP and apply Itô's lemma (Appendix 7.1)

$$dP = \frac{\partial P}{\partial t}dt + \frac{\partial P}{\partial v}dv + \frac{\partial P}{\partial S}dS + \frac{\partial^2 P}{\partial v \partial S}dvdS + \frac{1}{2}\frac{\partial^2 P}{\partial v^2}(dv)^2 + \frac{1}{2}\frac{\partial^2 P}{\partial S^2}(dS)^2. \tag{2.1.7}$$

This expression is the same for both P_1 and P_2. By inserting (2.1.1) and (2.1.4), and using the correlation (2.1.3) in the expression above, we can express it as

$$dP = \frac{\partial P}{\partial t}dt + \frac{\partial P}{\partial v}dv + \frac{\partial P}{\partial S}dS + \left(\frac{\partial^2 P}{\partial v \partial S}\rho\sigma v S + \frac{1}{2}\frac{\partial^2 P}{\partial v^2}\sigma^2 v + \frac{1}{2}\frac{\partial^2 P}{\partial S^2}S^2 v\right) dt. \tag{2.1.8}$$

The expression for dv and dS are not written out because of practical purposes. By inserting (2.1.8) into (2.1.6) we obtain

$$\begin{aligned} dc_t =& \Big\{\frac{\partial P_1}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 P_1}{\partial S^2} + \rho\sigma vS\frac{\partial^2 P_1}{\partial v\partial S} + \frac{1}{2}\sigma^2 v\frac{\partial P_1}{\partial v^2}\Big\}dt \\ &+ b_t\Big\{\frac{\partial P_2}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 P_2}{\partial S^2} + \rho\sigma vS\frac{\partial^2 P_2}{\partial v\partial S} + \frac{1}{2}\sigma^2 v\frac{\partial P_2}{\partial v^2}\Big\}dt \\ &+ \Big\{\frac{\partial P_1}{\partial S} + b_t\frac{\partial P_2}{\partial S} + a_t\Big\}dS \\ &+ \Big\{\frac{\partial P_1}{\partial v} + b_t\frac{\partial P_2}{\partial v}\Big\}dv. \end{aligned} \tag{2.1.9}$$

The whole idea behind option models in general is to create a hedged portfolio that is risk-free. In (2.1.9) we identify two terms that are associated with risk, namely the dS and dv terms. Both of these are random processes and contain a Brownian motion term. In order to make the portfolio instantaneously risk-free we must eliminate the dS and dv terms and therefore choose

$$\Big\{\frac{\partial P_1}{\partial v} + b_t\frac{\partial P_2}{\partial v}\Big\} = 0 \Longrightarrow b_t = -\frac{\frac{\partial P_1}{\partial v}}{\frac{\partial P_2}{\partial v}} \tag{2.1.10}$$

to eliminate dS terms, and

$$\Big\{\frac{\partial P_1}{\partial S} + b_t\frac{\partial P_2}{\partial S} + a_t\Big\} = 0 \Longrightarrow a_t = -\frac{\partial P_1}{\partial S} - \frac{\frac{\partial P_1}{\partial v}}{\frac{\partial P_2}{\partial v}}\frac{\partial P_2}{\partial S} \tag{2.1.11}$$

to eliminate dv terms. We now have a risk-free portfolio which should yield the same payoff as another risk-free asset, such as a bank account, called a bond. An investment of β_0 in bonds yields an amount of

$$\beta_t = \beta_0 e^{rt} \;\Rightarrow\; d\beta_t = r\beta_t dt \tag{2.1.12}$$

assuming the investment is continuously compounded (Mikosch, [3]). Applying this to our portfolio leads to

$$dc_t = r(a_tS_t + P_1 + b_tP_2)dt = rc_tdt. \tag{2.1.13}$$

Now we are able to put all of this together. The two expressions for dc_t, namely (2.1.9) and (2.1.13) are set equal to each other. Below we also insert the identities (2.1.10) and (2.1.11) and the result is

$$\begin{aligned} &\Big\{\frac{\partial P_1}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 P_1}{\partial S^2} + \rho\sigma vS\frac{\partial^2 P_1}{\partial v\partial S} + \frac{1}{2}\sigma^2 v\frac{\partial P_1}{\partial v^2}\Big\} \\ &- \frac{\frac{\partial P_1}{\partial v}}{\frac{\partial P_2}{\partial v}}\Big\{\frac{\partial P_2}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 P_2}{\partial S^2} + \rho\sigma vS\frac{\partial^2 P_2}{\partial v\partial S} + \frac{1}{2}\sigma^2 v\frac{\partial P_2}{\partial v^2}\Big\} \\ &= r\Big\{-\frac{\partial P_1}{\partial S}S - \frac{\frac{\partial P_1}{\partial v}}{\frac{\partial P_2}{\partial v}}\frac{\partial P_2}{\partial S}S + P_1 - \frac{\frac{\partial P_1}{\partial v}}{\frac{\partial P_2}{\partial v}}P_2\Big\}. \end{aligned}$$

We then move the right-hand side over to the left while keeping P_1 and P_2 separated

$$\begin{aligned}&\Big\{\frac{\partial P_1}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 P_1}{\partial S^2}+\rho\sigma vS\frac{\partial^2 P_1}{\partial v\partial S}+\frac{1}{2}\sigma^2 v\frac{\partial P_1}{\partial v^2}+rS\frac{\partial P_1}{\partial S}-rP_1\Big\}\\&-\frac{\frac{\partial P_1}{\partial v}}{\frac{\partial P_2}{\partial v}}\Big\{\frac{\partial P_2}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 P_2}{\partial S^2}+\rho\sigma vS\frac{\partial^2 P_2}{\partial v\partial S}+\frac{1}{2}\sigma^2 v\frac{\partial P_2}{\partial v^2}+rS\frac{\partial P_2}{\partial S}-rP_2\Big\}\\&=0.\end{aligned}$$

Finally we divide by $\frac{\partial P_1}{\partial v}$ and move the P_2 terms over to the right-hand side

$$\begin{aligned}&\frac{1}{\frac{\partial P_1}{\partial v}}\Big\{\frac{\partial P_1}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 P_1}{\partial S^2}+\rho\sigma vS\frac{\partial^2 P_1}{\partial v\partial S}+\frac{1}{2}\sigma^2 v\frac{\partial P_1}{\partial v^2}+rS\frac{\partial P_1}{\partial S}-rP_1\Big\}\\&=\frac{1}{\frac{\partial P_2}{\partial v}}\Big\{\frac{\partial P_2}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 P_2}{\partial S^2}+\rho\sigma vS\frac{\partial^2 P_2}{\partial v\partial S}+\frac{1}{2}\sigma^2 v\frac{\partial P_2}{\partial v^2}+rS\frac{\partial P_2}{\partial S}-rP_2\Big\}.\end{aligned}$$

In the equation above the left-hand side does not depend on T_2 and similarly the right-hand side does not depend on T_1, thus there exists a function $g(S, v, t)$ such that

$$\begin{aligned}&\frac{1}{\frac{\partial P}{\partial v}}\Big\{\frac{\partial P}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 P}{\partial S^2}+\rho\sigma vS\frac{\partial^2 P}{\partial v\partial S}+\frac{1}{2}\sigma^2 v\frac{\partial P}{\partial v^2}+rS\frac{\partial P}{\partial S}-rP\Big\}\\&=g(S,v,t).\end{aligned}\tag{2.1.14}$$

In agreement with Heston in [2], we choose $g(S, v, t) = -\kappa[\theta - v] + \lambda(S, v, t)$ where $\kappa[\theta - v]$ is the drift term from (2.1.4) and $\lambda(S, v, t)$ represents the market price of volatility risk.

$$\begin{aligned}&\frac{\partial P}{\partial t}+\frac{1}{2}vS^2\frac{\partial^2 P}{\partial S^2}+\rho\sigma vS\frac{\partial^2 P}{\partial v\partial S}+\frac{1}{2}\sigma^2 v\frac{\partial^2 P}{\partial v^2}\\&+rS\frac{\partial P}{\partial S}+\{\kappa[\theta-v]-\lambda(S,v,t)\}\frac{\partial P}{\partial v}-rP=0.\end{aligned}\tag{2.1.15}$$

The unspecified term $\lambda(S, v, t)$ must be independent of the particular asset, and from [2] the risk premium is proportional to v, hence $\lambda(S, v, t) = \lambda v$.

In order to obtain a forward parabolic equation we use the transformation $T - t \to t$, meaning that we work with time to maturity.

$$\frac{\partial P}{\partial t}-\frac{1}{2}vS^2\frac{\partial^2 P}{\partial S^2}-\rho\sigma vS\frac{\partial^2 P}{\partial v\partial S}-\frac{1}{2}\sigma^2 v\frac{\partial^2 P}{\partial v^2}-rS\frac{\partial P}{\partial S}-\{\kappa[\theta-v]-\lambda v\}\frac{\partial P}{\partial v}+rP=0.\tag{2.1.16}$$

2.2 The Heston model for American put options

We are also going to investigate the American vanilla put option in this paper. The difference between a European and American option is that the European option can only be exercised at the maturity date while the American option can be exercised at any time before the maturity date. The possibility of early exercise prevents the value of an American put to fall below the payoff function.

$$P(S, v, t) \geq \max[K - S, 0] \tag{2.2.1}$$

This is a very intuitive condition and an easy way to understand it, is to see what happens if one violates it. If the price of the option is lower than the payoff function one can buy the option, instantly exercise it and by repeating this earn an infinite amount of money. This condition divides the domain into two parts separated by the exercise boundary. The exercise boundary is the boundary between the region where (2.2.1) is active (equality) and the continuation region (inactive). When combining these relations it is possible to write the American option pricing problem as a linear complementary problem (Ikonen & Toivanen [10], and Clarke & Parrott [11]).

$$\mathcal{L}(P) \geq 0, \quad P \geq P_0, \tag{2.2.2}$$
$$\mathcal{L}(P)(P - P_0) = 0, \tag{2.2.3}$$

where $\mathcal{L}$ is the linear partial differential operator

$$\begin{aligned}\mathcal{L}(P) =& \frac{\partial P}{\partial t} - \frac{1}{2} v S^2 \frac{\partial^2 P}{\partial S^2} - \rho \sigma v S \frac{\partial^2 P}{\partial v \partial S} - \frac{1}{2}\sigma^2 v \frac{\partial^2 P}{\partial v^2} \\ & - rS\frac{\partial P}{\partial S} - \{\kappa[\theta - v] - \lambda v\}\frac{\partial P}{\partial v} + rP. \end{aligned} \tag{2.2.4}$$

Equation (2.2.3) ensures $\mathcal{L}(P) = 0$ when $P \geq P_0$ which is the region where (2.2.1) is inactive an thereby corresponds to the European case (2.1.16). There exist several methods for solving the variational inequalities above such as the projected SOR algorithm, penalization methods and front-tracking algorithms, but in this paper we will concentrate on the Brennan Schwartz algorithm (Brennan & Schwartz, [6]). This algorithm consists of solving $\mathcal{L}(P) = 0$ as in the European case, check if any values violates (2.2.1) and set those values to P_0. The choice of algorithm and its advantages will be discussed in section 3.5.

2.3 Boundary

A European put option with strike price K and maturity time T satisfies the PDE (2.1.16) subject to the following boundary conditions:

$$\begin{aligned}
&P(S,v,0) = max(K-S,0)\\
&P(0,v,t) = Ke^{-rt}\\
&\frac{\partial P(S,\infty,t)}{\partial v} = 0\\
&\frac{\partial P(\infty,v,t)}{\partial S} = 0\\
&rS\frac{\partial P}{\partial S}(S,0,t) + \kappa\theta\frac{\partial P}{\partial v}(S,0,t) - rP(S,0,t) - P_t(S,0,t) = 0
\end{aligned} \tag{2.3.1}$$

The initial condition is simply the payoff function after the time transformation $T-t \to t$. Next is the boundary at $S=0$ which corresponds to the present value of the strike price. The contract states that one can sell the underlying asset for K dollars. These K dollars will have lower value in the future and in the opposite manner of (2.1.12) we get Ke^{-rt}. The fifth equation (at $v=0$) is obtained directly from (2.1.16) by inserting $v=0$. When considering the boundary at $S=\infty$ one may take a naive approach and divide (2.1.16) by S^2 resulting in $\frac{\partial^2 P}{\partial S^2} = 0$ when $S \to \infty$. This is fulfilled by the boundary condition $\frac{\partial P}{\partial S} = 0$. Another approach is to look at P_0 which is equal to $K-S$ on the interval $[0, S=K]$ and equal to zero for $S \geq K$ and thereby clearly fulfilling $\frac{\partial P}{\partial S} = 0$ when $S \to \infty$. It is reasonable to assume that P will keep this property as the affect of the diffusion will have approximately the same impact on the nodes close to the boundary.

When considering the boundary $v \to \infty$ there are some different approaches. Winkler, Apel and Wystup use a Dirichlet type boundary condition in [5] while the O'Sullivans [4] and Ikonen & Toivanen [10] use the Neumann boundary condition stated above. The choice may depend on several aspects, but in this paper the Neumann boundary condition is used. When imposing the Dirichlet condition $P = Ke^{-rt}$ from [5], the boundary is set to Ke^{-rt}. This may be the case for large values of v_{max}, but certainly not for smaller values of v_{max}. When using the Neumann condition we should get a better result for smaller domains as this natural boundary condition do not impose a specific value on the system.

Chapter 3

Finite Differences

3.1 Discretization

In this section we will discretize (2.1.16) using a finite difference scheme. This is a straight forward approach with some minor indexing challenges when implementing the boundary conditions. For the space derivatives we use central difference and the terms from (2.1.16) is discretized below where $h = \Delta S$ and $k = \Delta v$

$$\begin{aligned}
(\frac{\partial P}{\partial S})_{i,j} &\approx \frac{P_{i+1,j} - P_{i-1,j}}{2h} \\
(\frac{\partial P}{\partial v})_{i,j} &\approx \frac{P_{i,j+1} - P_{i,j-1}}{2k} \\
(\frac{\partial^2 P}{\partial S^2})_{i,j} &\approx \frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{h^2} \\
(\frac{\partial^2 P}{\partial v^2})_{i,j} &\approx \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j+1}}{k^2} \\
(\frac{\partial^2 P}{\partial v \partial S})_{i,j} &\approx \frac{P_{i+1,j+1} + P_{i-1,j-1} - P_{i-1,j+1} - P_{i+1,j-1}}{4hk}
\end{aligned} \tag{3.1.1}$$

Applying backward difference (implicit Euler) in time and using uniform spatial discretization we get the following

$$\begin{aligned}
&- \frac{P^n_{i,j} - P^{n-1}_{i,j}}{\Delta t} + \frac{1}{2}(jk)(ih)^2 \frac{P^n_{i+1,j} - 2P^n_{i,j} + P^n_{i-1,j}}{h^2} \\
&+ \rho\sigma(jk)(ih) \frac{P^n_{i+1,j+1} - P^n_{i-1,j+1} + P^n_{i-1,j-1} - P^n_{i+1,j-1}}{4hk} \\
&+ \frac{1}{2}\sigma^2(jk) \frac{P^n_{i,j+1} - 2P^n_{i,j} + P^n_{i,j-1}}{k^2} + r(ih) \frac{P^n_{i+1,j} - P^n_{i-1,j}}{2h} \\
&+ \{\kappa[\theta - (jk)] - \lambda \cdot (jk)\} \frac{P^n_{i,j+1} - P^n_{i,j-1}}{2k} - rP^n_{i,j} = 0
\end{aligned} \tag{3.1.2}$$

with

$$P(S,v,t) = P(S_i, v_j, t_n) = P(ih, jk, n\Delta t) = P^n_{i,j} \qquad (3.1.3)$$
$$0 \leq i \leq I+1,\, 0 \leq j \leq J+1$$

We separate P^n and P^{n-1} and express (3.1.2) as

$$\begin{aligned} P^{n-1}_{i,j} =& a^n_{i,j} P^n_{i,j} + c^n_{i,j} P^n_{i-1,j} + d^n_{i,j} P^n_{i+1,j} + e^n_{i,j} P^n_{i,j-1} + f^n_{i,j} P^n_{i,j+1} \\ &+ b^n_{i,j}(P^n_{i+1,j+1} + P^n_{i-1,j-1} - P^n_{i-1,j+1} - P^n_{i+1,j-1}) \end{aligned} \qquad (3.1.4)$$

where

$$\begin{aligned} a^n_{i,j} &= 1 + \Delta t(i^2 jk + \frac{\sigma^2 j}{k} + r) \\ c^n_{i,j} &= -\frac{\Delta t}{2}(i^2 jk - ri) \\ d^n_{i,j} &= -\frac{\Delta t}{2}(i^2 jk + ri) \\ e^n_{i,j} &= -\frac{\Delta t}{2k}(\sigma^2 j - \kappa(\theta - jk) + \lambda jk) \\ f^n_{i,j} &= -\frac{\Delta t}{2k}(\sigma^2 j + \kappa(\theta - jk) - \lambda jk) \\ b^n_{i,j} &= -\frac{\rho\sigma ij}{4}\Delta t \end{aligned} \qquad (3.1.5)$$

The next step is to arrange P as a vector with natural ordering

$$P = \Big[p_{11}, p_{21}, ..., p_{I1}, p_{12}, p_{22}, ..., p_{I2}, ..., p_{IJ}\Big]^T,$$

and construct a matrix M in order to solve $MP^n = P^{n-1}$ using Matlab's backslash operator. M will be a $IJ \times IJ$ matrix given by

$$\mathrm{M} = \begin{bmatrix} A_1 & F_1 & 0 & 0 & 0 & \cdots & 0 \\ E_2 & A_2 & F_2 & 0 & 0 & \cdots & 0 \\ 0 & E_2 & A_2 & F_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & E_{J-2} & A_{J-2} & F_{J-2} & 0 \\ 0 & \cdots & \cdots & 0 & E_{J-1} & A_{J-1} & F_{J-1} \\ 0 & \cdots & \cdots & \cdots & 0 & E_J & A_J \end{bmatrix}$$

where the matrices A, F and E are $I \times I$ tridiagonal matrices given by

$$A_j = \begin{bmatrix} a_{1j} & d_{1j} & 0 & \cdots & 0 \\ c_{2j} & a_{2j} & d_{2j} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & c_{I-1j} & a_{I-1j} & d_{I-1j} \\ 0 & \cdots & 0 & c_{Ij} & a_{Ij} \end{bmatrix}$$

$$F_j = \begin{bmatrix} f_{1j} & b_{1j} & 0 & \cdots & 0 \\ -b_{2j} & f_{2j} & b_{2j} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -b_{I-1j} & f_{I-1j} & b_{I-1j} \\ 0 & \cdots & 0 & -b_{Ij} & f_{Ij} \end{bmatrix}$$

$$E_j = \begin{bmatrix} e_{1j} & -b_{1j} & 0 & \cdots & 0 \\ b_{2j} & e_{2j} & -b_{2j} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & b_{I-1j} & e_{I-1j} & -b_{I-1j} \\ 0 & \cdots & 0 & b_{Ij} & e_{Ij} \end{bmatrix}$$

Next we discretize the initial condition and boundary conditions (2.3.1). These will lead to some modifications in the matrices and the right-hand-side which will be explained step by step in the next section.

3.2 Discrete boundary

In order to solve this system we need to discretize the boundary conditions (2.3.1). The discrete boundaries are located at $i, j = 0$, $j = J+1$ and $i = I+1$, thus we have a $IJ \times IJ$ system for the internal nodes. The discrete boundary conditions are given below followed by an explanation for each of them.

$$\begin{aligned} & P^0_{i,j} = max(K - ih, 0) \\ & P^n_{0,j} = Ke^{-rn\Delta t} \\ & \frac{P^n_{i,J+1} - P^n_{i,J}}{k} = 0 \\ & \frac{P^n_{I+1,j} - P^n_{I,j}}{h} = 0 \\ & \frac{ri}{2}(P^n_{i+1,0} - P^n_{i-1,0}) + \frac{\kappa\theta}{k}(P^n_{i,1} - P^n_{i,0}) - rP^n_{i,0} - P_t(S,0,t) = 0 \end{aligned} \tag{3.2.1}$$

with two different approaches for P_t,

$$P_t(S,0,t) = \frac{P^n_{i,j} - P^{n-1}_{i,j}}{\Delta t} \text{ or } P_t(S,0,t) = \frac{P^{n+1}_{i,j} - P^n_{i,j}}{\Delta t}$$

Initial condition
Recalling the transformation $T - t \to t$ we see that the first condition is the initial condition, i.e the state of the system at $t = 0$. This is given by the payoff function which returns how much money one get by realizing the option.

Dirichlet boundary at $S = 0$ $(i = 0)$
When expressing the disretization on matrix form one always "lose" the boundary nodes. Considering the Dirichlet condition, this is handled by adding the missing terms to the right-hand-side. By investigating (3.1.2) the term

$$-c^n_{1,j}P^n_{0,j} - b^n_{1,j}(P^n_{0,j-1} - P^n_{0,j+1}) \tag{3.2.2}$$

has to be added to the right-hand side for $i = 1, \leq j \leq J$ (the first element in each sub column of P). Fortunately $P_{0,j}$ is constant for each time iteration which reduces the expression to

$$-c^n_{1,j}P^n_{0,j} \tag{3.2.3}$$

Neumann condition at $v = \infty$ $(j = J + 1)$
When considering a Neumann condition we have an expression that consists of unsolved nodes which means that we need to modify the matrix. Discretizing the boundary condition results in $P^n_{I+1,j} = P^n_{I,j}$. By investigating (3.1.2) we identify the missing nodes and rewrite the expression.

$$f^n_{i,J}P^n_{i,J+1} + b^n_{i,J}(P^n_{i+1,J+1} - P^n_{i-1,J+1}) = f^n_{i,J}P^n_{i,J} + b^n_{i,J}(P^n_{i+1,J} - P^n_{i-1,J}). \tag{3.2.4}$$

The nodes affected by this condition are the last I elements of P (the last sub column). We then add the right-hand-side of (3.2.4) to the appropriate entries in M. The corner $P_{0,J+1}$ is covered by the Dirichlet condition and by combining the two Neumann conditions we get $P_{I+1,J+1} = P_{I,J}$.

Neumann condition at $S = \infty$ $(i = I + 1)$
For the second Neumann condition we get a similar expression

$$d^n_{I,j}P^n_{I+1,j} + b^n_{I,j}(P^n_{I+1,j+1} - P^n_{I+1,j-1}) = d^n_{I,j}P^n_{I,j} + b^n_{I,j}(P^n_{I,j+1} - P^n_{I,j-1}). \tag{3.2.5}$$

which also needs to be added to M in the appropriate entries for $1 \leq j \leq J$. The corner at $P_{I+1,J+1}$ has already been included and $j = 0$ is not known before we solve the problem arising from the Robin condition.

Robin condition at $v = 0$ $(j = 0)$
The simplest way to handle this condition is to use the explicit Euler scheme. Rearranging the Robin condition with the explicit method results in

$$\frac{ri\Delta t}{2}(P^n_{i+1,0} - P^n_{i-1,0}) + P^n_{i,0}(1 - \frac{\kappa\theta\Delta t}{k} - r\Delta t) + \frac{\kappa\theta\Delta t}{k}P^n_{i,1} = P^{n+1}_{i,0} \tag{3.2.6}$$

which can be written as

$$P_0^{n+1} = Q_e P_0^n + \frac{\kappa\theta\Delta t}{k} P_1^n, \tag{3.2.7}$$

where P_0 is a vector containing the boundary nodes at $j = 0, 1 \leq i \leq I$, P_1 contains the internal nodes at $j = 1$ and Q_e is tridiag(-$\frac{ri\Delta t}{2}, 1 - r\Delta t - \frac{\kappa\theta\Delta t}{k}, \frac{ri\Delta t}{2}$). We have P_0^{n-1} from the initial condition and the boundary conditions are the Dirichlet and Neumann conditions at $i = 0$ and $i = I$ respectively. By using the explicit method we can in fact solve the boundary problem and treat it like a Dirichlet condition by adding the solved boundary to the right-hand-side of the main problem. The solving strategy will then be

1. Solve boundary
2. Add boundary as Dirichlet condition to the right-hand-side
3. Solve main problem
4. Repeat

When using the implicit Euler method the boundary condition look like

$$P_0^n = Q_i^{-1}(\frac{\kappa\theta\Delta t}{k} P_1^n + P_0^{n-1})$$

where Q_i is tridiag($\frac{ri\Delta t}{2}, 1 + r\Delta t + \frac{\kappa\theta\Delta t}{k}, -\frac{ri\Delta t}{2}$). This is a bit more complicated then the explicit case as P_1 is not known. We therefore have to add the first term on the right-hand-side to the matrix M and the second term is treated like a Dirichlet condition. The strategy for the implicit method will be the following

1. Add the first term to M (only done once)
2. Add the second term to the right-hand-side of the main problem
3. Solve the main problem
4. Update boundary
5. Repeat from point 2

The main motivation for using an implicit method is that it is unconditionally stable in norm $\|\cdot\|$ with respect to Δt while explicit methods require a certain relation between the step lengths.

3.3 Mathematical analysis

In this section we will find an expression for the truncation error. First we Taylor expand about the point (ih, jk) in the terms from (3.1.1) and then insert these approximations into (2.1.16). $P^n_{i,j}$ is denoted by $P_{i,j}$ when considering the spatial discretization. For better overview we write out $P_{i+1,j}$,

$$P_{i+1,j} \approx P_{i,j} + h\partial_S P_{i,j} + \frac{h^2}{2}\partial_S^2 P_{i,j} + \frac{h^3}{6}\partial_S^3 P_{i,j} + \frac{h^4}{24}\partial_S^4 P_{i,j}, \tag{3.3.1}$$

and $P_{i-1,j}$

$$P_{i-1,j} \approx P_{i,j} - h\partial_S P_{i,j} + \frac{h^2}{2}\partial_S^2 P_{i,j} - \frac{h^3}{6}\partial_S^3 P_{i,j} + \frac{h^4}{24}\partial_S^4 P_{i,j} \tag{3.3.2}$$

which is the same as the expression for $P_{i,j+1}$ and $P_{i,j-1}$ when substituting h and ∂_S with k and ∂_v respectively. Inserting these approximations into the expressions $\frac{\partial P}{\partial S}$ and $\frac{\partial^2 P}{\partial S^2}$ from (3.1.1) results in

$$\frac{\partial P}{\partial S} \approx \frac{P_{i+1,j} - P_{i-1,j}}{2h} \approx \frac{2h\partial_S P_{i,j} + \frac{h^3}{3}\partial_S^3 P_{i,j}}{2h} = \partial_S P_{i,j} + \frac{h^2}{6}\partial_S^3 P_{i,j} \tag{3.3.3}$$

and

$$\frac{\partial^2 P}{\partial S^2} \approx \frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{h^2} \approx \frac{h^2\partial_S^2 P_{i,j} + \frac{h^4}{12}\partial_S^4 P_{i,j}}{h^2} = \partial_S^2 P_{i,j} + \frac{h^2}{12}\partial_S^4 P_{i,j}, \tag{3.3.4}$$

which is also transferable to the derivatives of P with respect to v. The last term is the mixed term which expands to

$$\begin{aligned} P_{i+1,j+1} \approx & P_{i,j} + h\partial_S P_{i,j} + k\partial_v P_{i,j} + \frac{1}{2}[h^2\partial_S^2 P_{i,j} + 2hk\partial_S\partial_v P_{i,j} + k^2\partial_v^2 P_{i,j}] \\ & + \frac{1}{6}[h^3\partial_S^3 P_{i,j} + 3h^2k\partial_S^2\partial_k P_{i,j} + 3hk^2\partial_S\partial_k^2 P_{i,j} + k^3\partial_k^3 P_{i,j}] \\ & + \frac{1}{24}[h^4\partial_S^4 P_{i,j} + 4h^3k\partial_S^3\partial_v P_{i,j} + 6h^2k^2\partial_S^2\partial_v^2 P_{i,j} \qquad (3.3.5) \\ & + 4hk^3\partial_S\partial_v^3 P_{i,j} + k^4\partial_v^4 P_{i,j}]. \qquad (3.3.6) \end{aligned}$$

By considering $P_{i+1,j+1} - P_{i-1,j+1}$ first, we see that the only terms that will have different signs are h^{odd} resulting in

$$\begin{aligned} P_{i+1,j+1} - P_{i-1,j+1} \approx & 2h\partial_S P_{i,j} + 2hk\partial_S\partial_v P_{i,j} + \frac{1}{3}h^3\partial_S^3 P_{i,j} + hk^2\partial_S\partial_k^2 P_{i,j} \\ & + \frac{1}{3}h^3k\partial_S^3\partial_k P_{i,j} + \frac{1}{3}hk^3\partial_S\partial_k^3 P_{i,j}. \end{aligned} \tag{3.3.7}$$

$P_{i-1,j-1} - P_{i+1,j-1}$ will contain all terms with h^{odd}, but the terms with h^{odd} and k^{even} will have different signs then the previous case. Taking the sum of the two differences results in

$$
\begin{aligned}
P_{i+1,j+1} - P_{i-1,j+1} + P_{i-1,j-1} - P_{i+1,j-1} \approx & 4hk\partial_S\partial_v P_{i,j} \\
& + \frac{2}{3}h^3k\partial_S^3\partial_k P_{i,j} + \frac{2}{3}hk^3\partial_S\partial_k^3 P_{i,j},
\end{aligned}
\tag{3.3.8}
$$

which inserted into the expression in (3.1.1) leads to

$$
\begin{aligned}
\frac{\partial^2 P}{\partial S \partial v} \approx & \frac{P_{i+1,j+1} + P_{i-1,j-1} - P_{i-1,j+1} - P_{i+1,j-1}}{4hk} \\
\approx & \partial_S\partial_v P_{i,j} + \frac{1}{6}(h^2\partial_S^3\partial_k P_{i,j} + k^2\partial_S\partial_k^3 P_{i,j}).
\end{aligned}
\tag{3.3.9}
$$

We also need the approximation for the time derivative

$$
\frac{\partial P}{\partial t} \approx \frac{P^n_{i,j} - P^n_{i,j}}{\Delta t} \approx \partial_t P^n_{i,j} - \frac{\Delta t}{2}\partial_t P^n_{i,j}. \tag{3.3.10}
$$

The truncation error is obtain by taking the difference between the discrete equation and the undiscretized equation at (ih, jk) which results in

$$
\begin{aligned}
\tau^n_{i,j} = & \frac{\Delta t}{2}\partial_t P^n_{i,j} + \frac{1}{2}vS^2(\frac{h^2}{12}\partial_S^4 P^n_{i,j}) + \frac{\rho\sigma v S}{6}(h^2\partial_S^3\partial_k P^n_{i,j} + k^2\partial_S\partial_k^3 P^n_{i,j}) \\
& + \frac{1}{2}\sigma^2 v(\frac{k^2}{12}\partial_v^4 P^n_{i,j}) + rS(\frac{h^2}{6}\partial_S^3 P^n_{i,j}) + \{\kappa[\theta - v] - \lambda v\}(\frac{k^2}{6}\partial_v^3 P^n_{i,j}).
\end{aligned}
\tag{3.3.11}
$$

The information gained from the expression above is often expressed as

$$
\tau^n_{i,j} = \mathcal{O}(h^2 + k^2 + \Delta t), \tag{3.3.12}
$$

meaning that the finite difference method is of second order in both spatial dimensions and first order in time.

3.4 Brennan Schwartz

As stated earlier we apply the Brennan Schwartz algorithm for solving the pricing problem for the American option. This algorithm is easy to implement an a good fit with Matlab as it checks the solved system and forces it to be feasible instead of iteratively approaching a feasible solution. In this way we keep the advantage of using Matlab's backslash operator. The most common alternatives to Brennan Schwartz are iterative methods such as penalty methods and the projected SOR algorithm which start with an initial guess and converges to a feasible solution. These methods will result in a much slower solver when programming in Matlab. In Matlab the Brennan Schwartz algorithm can be compressed into the following code

$$
P(P < P_0) = P_0(P < P_0)
$$

where P_0 is the initial data and P is the solution vector which is updated every time iteration. $P < P_0$ returns an array of boolean variables stating if the inequality is true or not for every element, causing P to attain the values of P_0 for the elements that are below P_0. In many papers the algorithm is expressed as an LU-factorization, but this approach is slower than the backslash operator when using Matlab. In the original paper by Brennan and Schwartz [6] the algorithm is stated without LU-factorization.

3.5 Numerical results

The finite difference solver for the Heston equation has been implemented using Matlab which make the operation of inverting the matrix quite simple. In other programming languages one might have to use other methods like the conjugate gradient method in order to solve the system. The solver has been implemented by using vector and matrix operations and limiting for-loops to a minimum as Matlab is much slower with for-loops.

In this chapter we are going to investigate the different choices of explicit vs implicit boundary at $v = 0$, convergence and the difference between the European and American put. When considering convergence we will use the solver for the European put option because it is faster. Recall that the Brennan Schwartz algorithm consists of solving the equation for the European put and then forcing the American condition on the solution. The American solver will therefore obtain the same order of convergence and accuracy.

First we will take a look at the method with the boundary treated explicitly and the boundary treated implicitly. From the section 3.2 we saw that the advantage from the explicit approach is that we can break the model down into two problems. This leads to a better structure in the code and makes it easier to keep track of the indexes. The implicit approach on the other hand is more difficult to implement, but this method is unconditionally stable. On the next page we see the explicit solution and the implicit solution for the European put

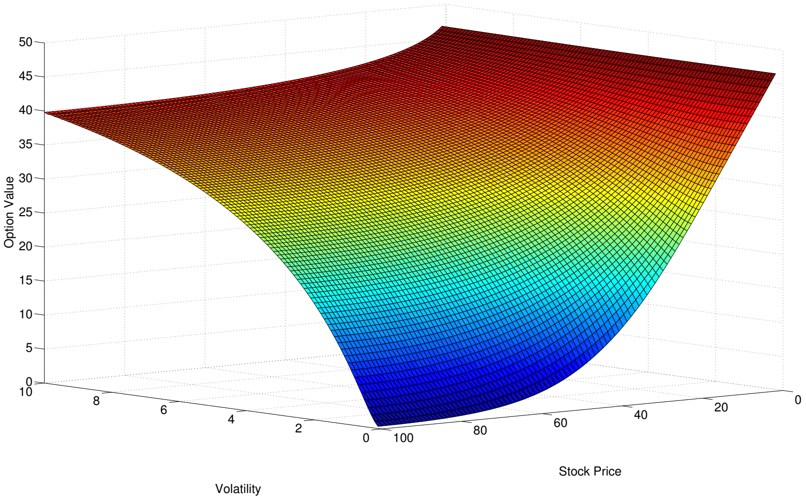

Figure 3.1: Surf-plot of the European vanilla put using the Heston model with explicit boundary. $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 0.1, \lambda = 0.1$ and $\rho = 0.5$. S, v, t range from 0 to $[100, 10, 1]$ with $[100, 100, 40]$ number of nodes.

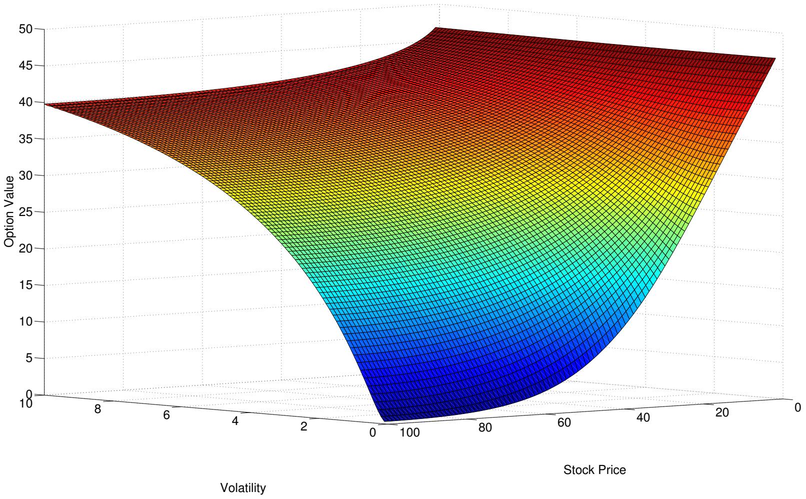

Figure 3.2: Surf-plot of the European vanilla put using the Heston model with implicit boundary. $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 0.1, \lambda = 0.1$ and $\rho = 0.5$. S, v, t range from 0 to $[100, 10, 1]$ with $[100, 100, 40]$ number of nodes.

The results are very similar which is reasonable as the only difference is the time discretization at $v = 0$. When taking the difference between these two solutions, the only difference is found close to the boundary $v = 0$ with 0.03 as the peak difference which is a small difference. For the explicit method to be stable we generally need to fulfill what is known as the CFL-condition. A well-known strategy for obtaining this condition is to perform von Neumann stability analysis, but this does not apply to problems with non-constant coefficients (Smith, [14]). It may be possible to get information on the behavior of the scheme by considering a "frozen" coefficients version, but the explicit discretization is limited to the boundary $v = 0$ which makes the analysis more complicated. Instead, the solver is run for different combinations of stepsizes in order to get an indication of the stability requirement. The table below is constructed by having $T = 1$, starting with 10 nodes in time and then increasing the number of nodes one at a time.

Δt	k	h	Inequality	Stability
0.1	0.01	0.01	Δt =5(h + k)	Unstable
0.091	0.01	0.01	Δt=4.55(h + k)	Unstable
0.083	0.01	0.01	Δt=4.17(h + k)	Unstable
0.077	0.01	0.01	Δt=3.85(h + k)	Stable

Table 3.1: Explicit solver is run for different combinations of stepsizes to record stability of the method

The table above indicates that the stability requirement is $\Delta t \leq 3.85(h + k)$. Note that this is only an observation made from the numerical results. However, by recalling that the scheme at the boundary $v = 0$ only contains first order derivatives it is reasonable to believe that the stability requirement will be on the form $\Delta t \leq C(h + k)$. The stability requirement is not very strong and since the time is measured in years while the stock price is measured in dollars it will be fulfilled in most practical cases. To be sure to have a stable method one can also easily implement a condition to ensure stability. In this paper the explicit solver has been used to produce the numerical results and the implicit method has been used to quality check them, and the results are the same. Next we will have a look at a problem with the FDM approach close to the boundary $v = v_{max}$.

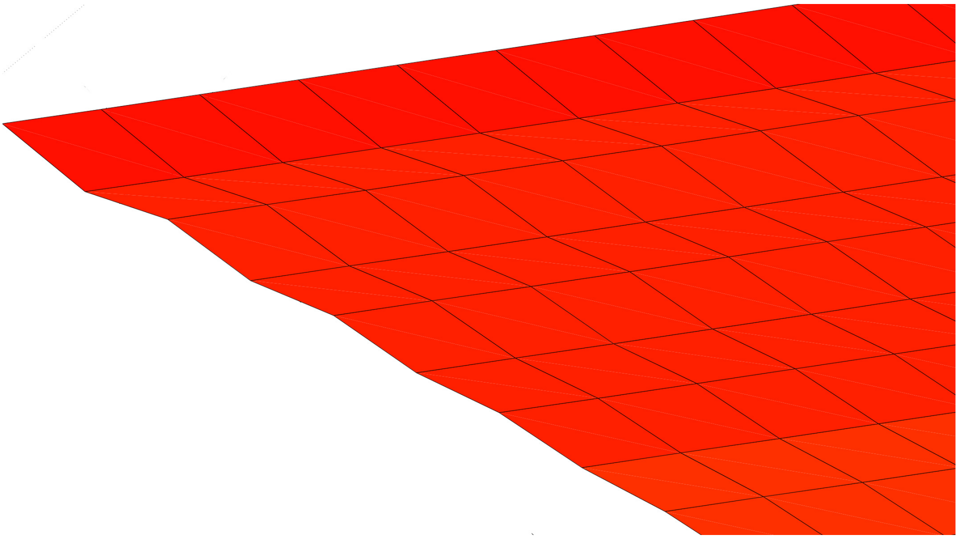

Figure 3.3: Surf-plot of the European vanilla put using the Heston model. Closeup look at $v = v_{max}$ (top of the plot) and $S = S_{max}$ (left side of the plot)

We observe that close to $v = v_{max}$ the solution is not smooth. This is a problem that is assumed to be connected with the convection term and this phenomenon is discussed by Quarteroni in [9] where he also proposes some strategies for avoiding the oscillations. One of the strategies is to use an upwinding scheme which consists of using $\frac{u_{i+1}-u_i}{h}$ or $\frac{u_i-u_{i-1}}{h}$ (depending on the equation) instead of first order central difference. The discretization of the problem when applying upwinding will be given in Appendix 7.5 including numerical results. By applying upwinding there will be a reduction of the order of convergence (Quarteroni, [9]). This cost is considered to be too high in order to eliminate the oscillations. The oscillations are only found close to the boundary $v = \infty$ and by investigating the solution in some distance from the boundary the problem can somewhat be avoided. This is done when considering the convergence of FDM by extracting a vector from the solution for a chosen v in the middle of the domain.

The convergence plot illustrating convergence in space is constructed by comparing solutions with a less refined grid to a reference solution. The reference solution is calculated using 256×256 nodes in space and 512 time iterations for $v_{max} = 15, S_{max} = 15$ and $T = 0.1$. This is then compared to results from grids ranging from 60×60 to 130×130 while keeping the number of time iterations constant. In order to compare solutions with different grids we apply interpolation using the built-in Matlab function *interp*2. This function takes the coarse grid, the fine grid and the coarse solution as input, and interpolate the coarse solution over the fine grid. Another approach would be to double/halve the grid size in order to make the nodes coincide. This leads to a very wide range for h and k which is not very fortunate as it makes it more difficult to keep the error associated with the time step neglectable. The error is computed using the L^2-norm which can be expressed as

$$\|P - P_h\|_{L^2} \approx \sqrt{\sum |P_i - P_{ih}|^2 h^2} = h\sqrt{\sum |P_i - P_{ih}|^2}, \tag{3.5.1}$$

when extracting a vector for a chosen v and having $h = k$. Below we have the loglog-plot for convergence in space.

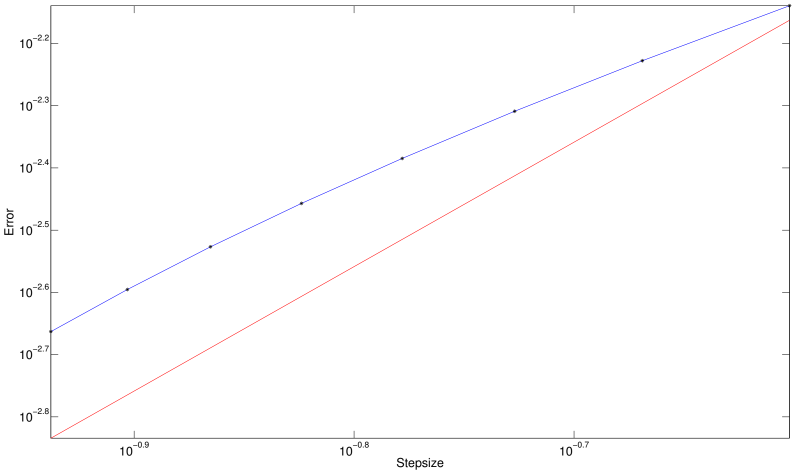

Figure 3.4: Loglog-plot for convergence in space. The red line is stepsizes plotted against stepsizes squared and the blue line is the error plotted against the stepsizes

Recalling the truncation error we should have second order convergence in both spatial dimensions. From the plot above we see that the lines are parallel to the left, but the order decreases to the right. This is most likely a consequence of using interpolation on the coarse grids. They are much more affected by oscillations which leads to an overestimate of the solution which again leads to a smaller error. We observe that the method is of second order for reasonably sized grids. When comparing the methods in Chapter 6. we will discuss the convergence further. Next we have the convergence plot for t which should be of order one.

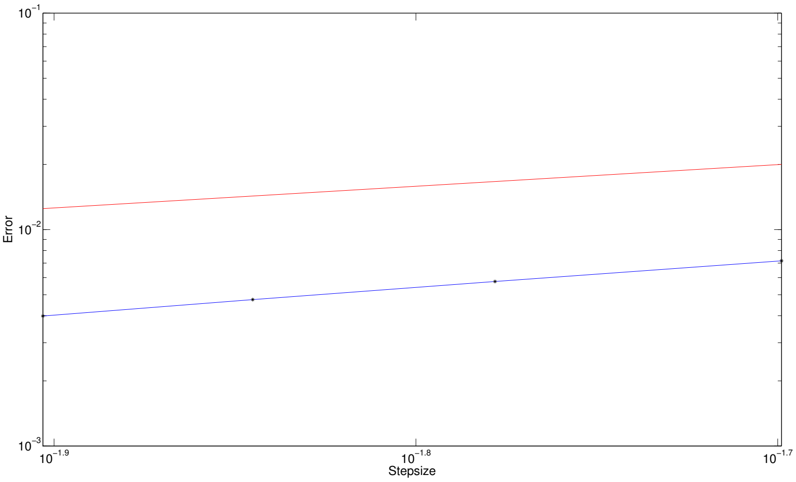

Figure 3.5: Loglog-plot for convergence in time. The red line is stepsize plotted against stepsize the blue line is the error.

The red line has slope 1 and we see that the blue line is parallel to the red, indicating first order convergence with respect to t. Comparing the solutions for different values of t is easier as the grid remains the same. This is because the nodes are located at the same position for each case. In this case the error associated with t has to be the dominating part of the error, i.e the error associated with the spatial discretization needs to be small. The solution for the American put is found on the next page.

Finally we have the solution for the American option using FDM and applying the Brennan Schwartz algorithm when solving the system of equations for each time iteration.

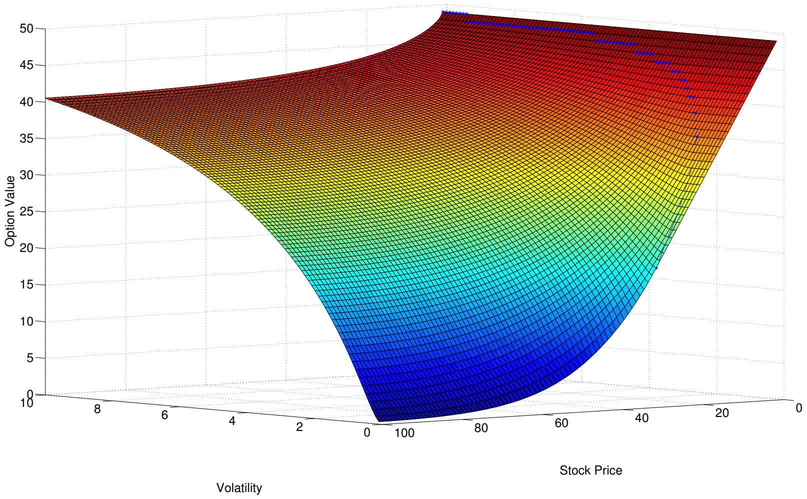

Figure 3.6: Surf-plot of the American vanilla put using the Heston model with explicit boundary. $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 0.1, \lambda = 0.1$ and $\rho = 0.5$. S, v, t range from 0 to $[100, 10, 1]$ with $[100, 100, 40]$ number of nodes. The blue line represents the exercise boundary

From this plot we see that the exercise boundary tends to $S = 0$ as v tends to infinity. However there will always be an interval where the American condition is active because the European option "sinks" with a rate e^{-rt} at $S = 0$ and this will affect the whole solution. But the American option clearly approaches the European option for $v \to \infty$ outside the exercise boundary, and as $v \to \infty$ the exercise boundary will be located at $S = 0$. More properties of the American option will be discussed when applying the element method in Chapter 5.

Chapter 4

Finite Element Method for European Options

4.1 Weak formulation

In this section we are going to formulate the variational problem in order to solve the problem using the finite element method. We start by rewriting (2.1.16) in divergence form as

$$\begin{aligned} &\frac{\partial P}{\partial t} - \frac{\partial}{\partial S}(\frac{vS^2}{2}\frac{\partial P}{\partial S}) - \frac{\partial}{\partial S}(\frac{\rho\sigma vS}{2}\frac{\partial P}{\partial v}) - \frac{\partial}{\partial v}(\frac{\rho\sigma vS}{2}\frac{\partial P}{\partial S}) - \frac{\partial}{\partial v}(\frac{\sigma^2 v}{2}\frac{\partial P}{\partial v}) \\ &-(rS - vS - \frac{\rho\sigma S}{2})\frac{\partial P}{\partial S} - (\{\kappa[\theta - v] - \lambda v\} - \frac{\rho\sigma v}{2} - \frac{\sigma^2}{2})\frac{\partial P}{\partial v} + rP = 0 \end{aligned} \tag{4.1.1}$$

which can be written as

$$P_t - \nabla \cdot A\nabla P - b \cdot \nabla P + rP = 0, \tag{4.1.2}$$

with A and b defined by

$$A = \frac{1}{2}v \begin{bmatrix} S^2 & \rho\sigma S \\ \rho\sigma S & \sigma^2 \end{bmatrix}, \tag{4.1.3}$$

$$b = \begin{bmatrix} rS - \frac{\rho\sigma S}{2} - vS \\ \kappa(\theta - v) - \lambda v - \frac{\rho\sigma v}{2} - \frac{\sigma^2}{2} \end{bmatrix} \tag{4.1.4}$$

In order to formulate this as a variational problem we multiply (4.1.2) by a test function $\phi(S, v)$ where $\phi = 0$ at $S = 0$ where the Dirichlet boundary condition is located and the solution is already known. We then integrate over the domain $\Omega = (0, S_{max}) \times (0, v_{max})$,

$$0 = \int\limits_{\Omega} P_t \phi - \int\limits_{\Omega} (\nabla \cdot A \nabla P)\phi - \int\limits_{\Omega} (b \cdot \nabla P)\phi + \int\limits_{\Omega} rP\phi. \tag{4.1.5}$$

An application of integration by parts result in

$$0 = \int\limits_{\Omega} P_t \phi + \int\limits_{\Omega} A\nabla P \cdot \nabla \phi - \int\limits_{\partial\Omega} (A\nabla P \cdot \vec{n})\phi - \int\limits_{\Omega} (b \cdot \nabla P)\phi + \int\limits_{\Omega} rP\phi. \tag{4.1.6}$$

To investigate the line integrals more closely we write out $A\nabla P$

$$A\nabla P = \frac{v}{2} \begin{bmatrix} S^2 \frac{\partial P}{\partial S} + \rho\sigma S \frac{\partial P}{\partial v} \\ \rho\sigma S \frac{\partial P}{\partial S} + \sigma^2 \frac{\partial P}{\partial v} \end{bmatrix}. \tag{4.1.7}$$

The square formed by Ω is then partitioned into four sides

$$\begin{aligned}
&\Gamma_a : S = 0,\ P(0, v, t) = Ke^{-rt},\ \vec{n}_a = [-1, 0]^T \\
&\Gamma_b : v = v_{max},\ \frac{\partial P(S, v_{max}, t)}{\partial v} = 0,\ \vec{n}_b = [0, 1]^T \\
&\Gamma_c : S = S_{max},\ \frac{\partial P(S_{max}, v, t)}{\partial S} = 0,\ \vec{n}_c = [1, 0]^T \\
&\Gamma_d : v = 0,\ \vec{n}_d = [0, -1]^T, \\
&\quad rS\frac{\partial P}{\partial S}(S, 0, t) + \kappa\theta \frac{\partial P}{\partial v}(S, 0, t) - rP(S, 0, t) - P_t(S, 0, t) = 0.
\end{aligned}$$

The line integral now divides into four terms which are calculated below. The zeros inserted in the expressions below are $S, v = 0$ and the Neumann conditions from (2.3.1).

$$\begin{aligned}
A\nabla P \cdot \vec{n}_a &= \frac{v}{2} \begin{bmatrix} 0 + 0 \\ 0 + \sigma^2 \frac{\partial P}{\partial v} \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 0 \end{bmatrix} = 0, \\
A\nabla P \cdot \vec{n}_b &= \frac{v_{max}}{2} \begin{bmatrix} S^2 \frac{\partial P}{\partial S} + 0 \\ \rho\sigma S \frac{\partial P}{\partial S} + 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{\rho\sigma v_{max}}{2} S \frac{\partial P}{\partial S}(S, v_{max}, t), \\
A\nabla P \cdot \vec{n}_c &= \frac{v}{2} \begin{bmatrix} 0 + \rho\sigma S_{max} \frac{\partial P}{\partial v} \\ 0 + \sigma^2 \frac{\partial P}{\partial v} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{\rho\sigma S_{max}}{2} v \frac{\partial P}{\partial v}(S_{max}, v, t), \\
A\nabla P \cdot \vec{n}_d &= \frac{0}{2} \begin{bmatrix} S^2 \frac{\partial P}{\partial S} + \rho\sigma S \frac{\partial P}{\partial v} \\ \rho\sigma S \frac{\partial P}{\partial S} + \sigma^2 \frac{\partial P}{\partial v} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} = 0,
\end{aligned} \tag{4.1.8}$$

The integrals at $v = 0$ and $S = 0$ disappear and we are left with the boundaries $v = v_{max}$ and $S = S_{max}$. In a "traditional" problem solved with the element method we usually have $a(u, w) = (f, w)$ were the boundary terms are added to the vector on the right-hand side. This is typically done when $\frac{\partial u}{\partial n} = g$ for some known function g, where $g\phi$ corresponds to a vector structure. In this case we can either handle the boundary terms explicitly to get a known vector on the right-hand side, or treat them implicitly and keep them on the left-hand side. In this paper the implicit approach is chosen because the implementation of the two alternatives are very similar. (4.1.6) expands to

$$\int_{\Omega} P_t\phi + \int_{\Omega} A\nabla P \cdot \nabla\phi - \int_{\Gamma_b} (A\nabla P \cdot \vec{n}_b)\phi - \int_{\Gamma_c} (A\nabla P \cdot \vec{n}_c)\phi - \int_{\Omega} (b \cdot \nabla P)\phi + \int_{\Omega} rP\phi = 0, \tag{4.1.9}$$

The bilinear form a is then defined as

$$a(u, w) = \int_{\Omega} (A\nabla u \cdot \nabla w) - \int_{\Gamma_b} (A\nabla u \cdot \vec{n}_b)w - \int_{\Gamma_c} (A\nabla v \cdot \vec{n}_c)w - \int_{\Omega} (b \cdot \nabla u)w + \int_{\Omega} rvw. \tag{4.1.10}$$

In order for the integrals in the bilinear form $a(\cdot, \cdot)$ to be well defined we introduce the norm

$$\|u\|_a^2 = a(u, u), \tag{4.1.11}$$

which is used to define the space

$$V = \{u : \|u\|_a < \infty\}. \tag{4.1.12}$$

By the definition of V the bilinear form $a(\cdot, \cdot)$ is bounded and hence all integrals in $a(\cdot, \cdot)$ is bounded. Denoting by V' the dual space of V and by $(\cdot, \cdot)$ the duality pairing between V' and V the weak formulation can be expressed as the following:

Find $P \in \mathcal{C}^0([0, T]; L^2(\Omega) \cap L^2([0, T]; V)$, such that $\frac{\partial P}{\partial t} \in L^2([0, T]; V')$, satisfying

$$P_{|t=0} = P_0 \text{ in } \Omega \text{ and for a.e } t \in (0, T) \qquad , \tag{4.1.13}$$

$$\forall u \in V, \; \left(\frac{\partial P}{\partial t}(t), u\right) + a(P(t), u) = 0. \tag{4.1.14}$$

In the next section it will be explained how this is discretized and a simple strategy for finding the expression for the basis function.

4.2 Discretization

In order to solve the problem using the element method we need to search for a solution P_h in a smaller discrete space $V_h \subset V$. Let V_h be the space of continuous piecewise linear functions on a triangulation of the domain Ω. The triangulation of Ω is constructed as a uniform finite difference grid with every square cut in half by the downward diagonal. With $(\phi_i)_{i=0,...,N}$ as the nodal basis of V_h we can write the solution P_h as a weighted sum of these, i.e $P_h = \sum_{i=1}^{N} P_i \phi_i$ with N being the total number of nodes $(I \cdot J)$. By searching for a solution in the discrete space V_h the semi-discretization of (4.1.14) is obtained.

Find $P_h \in V_h$ s.t.

$$(\frac{\partial P_h}{\partial t}, u_h) + a(P_h, u_h) = 0 \quad \forall u_h \in V_h. \tag{4.2.1}$$

Inserting the expression for P_h results in

$$(\sum_{j=1}^{N} \dot{P}_j \phi_j, \phi_i) + a(\sum_{j=1}^{N} P_j \phi_j, \phi_i) = 0 \quad , i = 1, 2, ..., N,$$

which can be written as

$$\sum_{j=1}^{N} \dot{P}_j (\phi_j, \phi_i) + \sum_{j=1}^{N} P_j a(\phi_j, \phi_i) = 0 \quad , i = 1, 2, ...N. \tag{4.2.2}$$

By defining the vector of unknowns $P = (P_0, ..., P_J)^T$ as $P_j = (P(S_0, v_j), ..., P(S_I, v_j))^T$, the mass matrix $\boldsymbol{M} = [m_{ij}] = (\phi_i, \phi_j)$ and the stiffness matrix $\boldsymbol{A} = [a_{ij}] = [a(\phi_i, \phi_j)]$, we can write (4.2.2) in matrix form as

$$\boldsymbol{M}\dot{P} + \boldsymbol{A}P = 0 \tag{4.2.3}$$

The remaining part is now to decide on a time discretization. We apply the implicit Euler scheme to (4.2.3) and the result is

$$\begin{aligned} \boldsymbol{M}(P^m - P^{m-1}) + \Delta t \boldsymbol{A} P^m &= 0 \qquad (4.2.4) \\ \Rightarrow P^m &= (\boldsymbol{M} + \Delta t \boldsymbol{A})^{-1} \boldsymbol{M} P^{m-1}. \end{aligned}$$

This is the desired formulation in order to solve the problem numerically with the element method. The next task is to create the triangulation and obtain the basis functions in order to construct the matrices.

The triangulation can be created by constructing a finite difference grid by storing x and y coordinates for every node in a matrix. This matrix is then given as input to the built-in Matlab function *delaunay* which returns the triangulation. Next we need to iterate over the elements and find the basis functions ϕ_i.

Linear basis functions with the general form $\phi_i = a_i + b_i x + c_i y$ are used with the standard definition stating $\phi_i(x_i, y_i) = 1$ and 0 otherwise. This gives nine equations for calculating nine coefficients for each element which in matrix form is

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

equivalent to

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1}$$

When programming in Matlab it is important to use matrices and vectors as often as possible instead of looping through every combination of basis functions. After finding the basis functions for each element the integral needs to be evaluated and added to the appropriate entries of the matrix. This is done by applying Gaussian quadrature (Appendix 7.4) and a local→ global mapping. Both matrices will have seven nonzero diagonals as every internal node will be part of six elements giving one unique contribution each in addition to the contribution from the node itself.

4.3 Mathematical analysis

The bilinear form $a(\cdot,\cdot)$ contains two boundary integrals that complicates the problem when considering mathematical analysis. To simplify the problem, we will assume Dirichlet type boundary conditions on all boundaries. The strategy in FEM for treating Dirichlet boundary conditions is to define the test function to be equal to zero on the boundary, causing all boundary integrals to be equal to zero. The practical consequence of such a simplification is that the error from the boundary will be greater. Recall that the two boundary integrals in $a(\cdot,\cdot)$ are evaluated at $v = v_{max}$ and $S = S_{max}$. This means that v_{max} and S_{max} has to be considerably increased for unsuitable Dirichlet conditions to minimize the error from the boundary. Also recall that the boundary condition at $v = 0$ treated explicitly is equivalent to a Dirichlet condition, hence it is mainly $v = v_{max}$ and $S = S_{max}$ that are affected by this simplification. The analysis will also be performed on the closed domain Ω. This will be further discussed when defining appropriate function spaces for the analysis. The bilinear form considered in this section is now reduced to

$$a_0(u,w) = \int\limits_{\Omega} (A\nabla u \cdot \nabla w) - \int\limits_{\Omega} (b \cdot \nabla u)w + \int\limits_{\Omega} rvw. \tag{4.3.1}$$

In order for the integrals in (4.3.1) to be well defined we define a space V_* requiring that all integrands are products of L^2 functions resulting in all integrands being in L^1 and thereby integrable. In order to obtain these terms we rewrite A in the following way

$$A = \frac{1}{2}(\rho B_1 B_1^T + (1-\rho)B_2 B_2^T)$$

where B_1 and B_2 are defined as

$$B_1 = \begin{bmatrix} \sqrt{v}S \\ \sqrt{v}\sigma \end{bmatrix} \qquad B_2 = \begin{bmatrix} \sqrt{v}S & 0 \\ 0 & \sqrt{v}\sigma \end{bmatrix}.$$

With this new representation of A we can rewrite the integrand from the first integral in (4.3.1)

$$\begin{aligned} A\nabla u \cdot A\nabla w &= (A\nabla u)^T \nabla w \\ &= \frac{1}{2}\rho(B_1 B_1^T \nabla u)^T \nabla w + \frac{1}{2}(1-\rho)(B_2 B_2^T \nabla u)^T \nabla w \\ &= \frac{1}{2}\rho(B_1^T \nabla u)^T (B_1^T \nabla w) + \frac{1}{2}(1-\rho)(B_2^T \nabla u)^T (B_2^T \nabla w). \end{aligned}$$

This formulation makes it very clear what to require as L^2 functions. For a function $u \in L^2$ we require $\sqrt{v}S\frac{\partial u}{\partial S}$ and $\sqrt{v}\frac{\partial u}{\partial v}$ in L^2. Writing out the convection terms results in the following products

$$v\frac{\partial u}{\partial v}w,\ vS\frac{\partial u}{\partial S}w,\ S\frac{\partial u}{\partial S}w,\ \frac{\partial u}{\partial v}w.$$

Expressing the above products with the factors that are already required as L^2 functions

$$(\sqrt{v}\frac{\partial u}{\partial v})(\sqrt{v}w),\ (vS\frac{\partial u}{\partial S})(\sqrt{v}w),\ (vS\frac{\partial u}{\partial S})(\frac{w}{\sqrt{v}}),\ (\sqrt{v}\frac{\partial u}{\partial v})(\frac{w}{\sqrt{v}}).$$

The above introduces two new terms to be required as L^2 functions, namely $\sqrt{v}u$ and $\frac{w}{\sqrt{v}}$. Recalling that the analysis is performed on the closed domain Ω it is possible to write $\|\sqrt{v}u\|_{L^2(\Omega)} \leq \|\sqrt{v}\|_{L^\infty(\Omega)}\|u\|_{L^2(\Omega)} = \max_\Omega(\sqrt{v})\|u\|_{L^2(\Omega)}$. This causes $\sqrt{v}u$ to be superfluous and is therefore not included in the space V_*. Defining the space

$$V_* = \{u : \sqrt{v}S\frac{\partial u}{\partial S}, \sqrt{v}\frac{\partial u}{\partial v}, \frac{u}{\sqrt{v}}, u \in L^2(\Omega)\}, \tag{4.3.2}$$

with the norm

$$\|u\|_{V*} = \left(\int_{\Omega} vS^2 \left(\frac{\partial u}{\partial S}\right)^2 + v\left(\frac{\partial u}{\partial v}\right)^2 + (1+\frac{1}{v})u^2\right)^{\frac{1}{2}}, \tag{4.3.3}$$

equivalent to

$$\|u\|_{V_*^2} = \|\sqrt{v}S\frac{\partial u}{\partial S}\|_{L^2}^2 + \|\sqrt{v}\frac{\partial u}{\partial v}\|_{L^2}^2 + \|\frac{u}{\sqrt{v}}\|_{L^2}^2 + \|u\|_{L^2}^2. \tag{4.3.4}$$

Denoting by $\mathcal{D}(\Omega)$ the space of smooth functions with compact support in Ω and by $\overline{\mathcal{D}(\Omega)}$ the closure of $\mathcal{D}(\Omega)$ we define

$$V_0 = \overline{\mathcal{D}(\Omega)}^{\|\cdot\|_{V*}}. \tag{4.3.5}$$

V_0 is a Hilbert space and has the following properties

1. V_0 is separable
2. $\mathcal{D}(\Omega) \subset V_0$ and $\mathcal{D}(\Omega)$ is dense in V_0 by definition.
3. V_0 is dense in $L^2(\Omega)$

The introduction of V_0 enables us to perform the analysis by applying $\|\cdot\|_{V*}$ in addition to all boundary integrals arising from integration by parts will be equal to zero.

There are few papers performing detailed mathematical analysis on the element method applied to the Heston model, but a similar case can be found in (Achdou & Pironneau, [7]). The differences between this case and the case in [7] is summarized below

- In [7] they simplify the expression by setting ρ equal to zero, thus eliminating the mixed term $\frac{\partial^2 P}{\partial v \partial S}$. In our case we keep ρ as it is, namely $0\ |\rho| < 1$.
- In [7] they consider $(0,\infty) \times (0,\infty)$ by introducing a scaling function. In our case we consider the closed domain Ω.
- The equation in [7] contains the term $\lambda\sqrt{v}\gamma(S,v,t)\frac{\partial u}{\partial v}$ and they make the assumption that $\gamma(S,v,t)$ is bounded. In our case we use the same term as Heston in his original paper (Heston, [2]) which is $\lambda v\frac{\partial u}{\partial v}$ resulting in $\gamma(S,v,t) = \sqrt{v}$. $\gamma(S,v,t)$ is clearly not bounded when considering $(0,\infty) \times (0,\infty)$.

In this paper we will prove continuity and energy stability. This will ensure uniqueness and stability, which will be shown at the end of this section. First we will prove continuity and then derive an inequality that is similar to Gårding's inequality in order to obtain the expression for energy stability.

Writing out the full expression for $a_0(u,w)$ (4.3.1)

$$\begin{aligned} a_0(u,w) = & \overbrace{\frac{1}{2}\int_\Omega vS^2\frac{\partial u}{\partial S}\frac{\partial w}{\partial S} + \frac{1}{2}\sigma^2\int_\Omega v\frac{\partial u}{\partial v}\frac{\partial w}{\partial v} + \frac{\rho\sigma}{2}\int_\Omega vS\left(\frac{\partial u}{\partial v}\frac{\partial w}{\partial S} + \frac{\partial u}{\partial S}\frac{\partial w}{\partial v}\right)}^{\text{Diffusion term}} \\ & \overbrace{-\int_\Omega (r - v - \frac{\rho\sigma}{2})S\frac{\partial u}{\partial S}w - \int_\Omega (\{\kappa[\theta - v] - \lambda v\} - \frac{\rho\sigma v}{2} - \frac{\sigma}{2})\frac{\partial u}{\partial v}w}^{\text{Convection term}} \end{aligned} \tag{4.3.6}$$

$$+ \quad \overbrace{r\int_\Omega uw}^{\text{Reaction term}} \quad . \tag{4.3.7}$$

Applying Cauchy-Schwartz on the expression above and using the fact that $\|\sqrt{v}u\|_{L^2(\Omega)} \leq \max_\Omega(\sqrt{v})\|u\|_{L^2(\Omega)}$ results in

$$\begin{aligned} |a_0(u,w)| \leq & \frac{1}{2}\|\sqrt{v}S\frac{\partial u}{\partial S}\|_{L^2(\Omega)}\|\sqrt{v}S\frac{\partial w}{\partial S}\|_{L^2(\Omega)} + \frac{1}{2}\|\sqrt{v}\frac{\partial u}{\partial v}\|_{L^2(\Omega)}\|\sqrt{v}\frac{\partial w}{\partial v}\|_{L^2(\Omega)} \\ & + \frac{\rho\sigma}{2}\left(\|\sqrt{v}\frac{\partial u}{\partial v}\|_{L^2(\Omega)}\|\sqrt{v}S\frac{\partial w}{\partial S}\|_{L^2(\Omega)} + \|\sqrt{v}S\frac{\partial u}{\partial S}\|_{L^2(\Omega)}\|\sqrt{v}\frac{\partial w}{\partial v}\|_{L^2(\Omega)}\right) \\ & + (r + \frac{\rho\sigma}{2})\|\sqrt{v}S\frac{\partial u}{\partial S}\|_{L^2(\Omega)}\|\frac{w}{\sqrt{v}}\|_{L^2(\Omega)} + \|\sqrt{v}S\frac{\partial u}{\partial S}\|_{L^2(\Omega)}\max_\Omega(\sqrt{v})\|w\|_{L^2(\Omega)} \\ & + (\kappa + \lambda + \frac{\rho\sigma}{2})\|\sqrt{v}\frac{\partial u}{\partial v}\|_{L^2(\Omega)}\max_\Omega(\sqrt{v})\|w\|_{L^2(\Omega)} \\ & + (\kappa\theta + \frac{\sigma}{2})\|\sqrt{v}\frac{\partial u}{\partial S}\|_{L^2(\Omega)}\|\frac{w}{\sqrt{v}}\|_{L^2(\Omega)} + r\|u\|_{L^2(\Omega)}r\|w\|_{L^2(\Omega)} \end{aligned} \tag{4.3.8}$$

All norms in the expression above are contained in $\|\cdot\|_V$, hence each product of L^2-norms is less then or equal to $\|u\|_V\|w\|_V$. Denoting C_i by the coefficient to the i'th term in the expression above it is then possible to write

$$|a_0(u,w)| \leq C\|u\|_{V_0}\|w\|_{V_0}, \tag{4.3.9}$$

where $C = \max(C_i)$ for $i = 1, ..., 8$.

Next we will derive an inequality that is similar to Gårding's inequality as a mathematical tool for the energy stability. Gårding's inequality is defined as

$$a_0(u,u) \geq C\|u\|^2_{V_0} - c\|u\|^2_{L^2(Q)} \qquad \forall u \in V_0, \tag{4.3.10}$$

while we will show

$$a_0(u,u) \geq C|u|^2_{V_0} - c\|u\|^2_{L^2(\Omega)} \qquad \forall u \in V_0. \tag{4.3.11}$$

Writing out the bilinear form

$$\begin{aligned}a_0(u,u) =&\frac{1}{2}\int_\Omega vS^2\left(\frac{\partial u}{\partial S}\right)^2+\frac{1}{2}\sigma^2\int_\Omega v\left(\frac{\partial u}{\partial v}\right)^2+\rho\sigma\int_\Omega vS\left(\frac{\partial u}{\partial v}\frac{\partial u}{\partial S}\right)\\ &-\int_\Omega(r-v-\frac{\rho\sigma}{2})S\frac{\partial u}{\partial S}u-\int_\Omega(\{\kappa[\theta-v]-\lambda v\}-\frac{\rho\sigma v}{2}-\frac{\sigma}{2})\frac{\partial u}{\partial v}u\\ &+r\int_\Omega u^2. \end{aligned}\tag{4.3.12}$$

The convection terms in the expression above can be simplified by applying integration by parts. The boundary terms are left out of the expressions because they are equal to zero,

$$\begin{aligned}-\int_\Omega(r-v-\frac{\rho\sigma}{2})S\frac{\partial u}{\partial S}u =&\int_\Omega(r-v-\frac{\rho\sigma}{2})u\frac{\partial}{\partial S}(Su)\\ =&\int_\Omega(r-v-\frac{\rho\sigma}{2})u(u+S\frac{\partial u}{\partial S})\\ \Rightarrow-\int_\Omega(r-v-\frac{\rho\sigma}{2})S\frac{\partial u}{\partial S}u =&\frac{1}{2}\int_\Omega(r-v-\frac{\rho\sigma}{2})u^2. \end{aligned}\tag{4.3.13}$$

For the sake of better overview when performing integration by parts on the second convection term we introduce $q(v)=(\{\kappa[\theta-v]-\lambda v\}-\frac{\rho\sigma v}{2}-\frac{\sigma}{2})$. The procedure is the same as above and the result is

$$\begin{aligned}-\int_\Omega q(v)\frac{\partial u}{\partial v}u =&-\int_\Omega u\frac{\partial}{\partial v}(q(v)u)\\ =&\int_\Omega u(q(v)'u+q(v)\frac{\partial u}{\partial v})\\ \Rightarrow-\int_\Omega q(v)S\frac{\partial u}{\partial S}u =&\frac{1}{2}\int_\Omega q(v)'u^2,\\ =&\frac{1}{2}(\kappa-\lambda-\frac{\rho\sigma}{2})\int_\Omega u^2. \end{aligned}\tag{4.3.14}$$

Inserting (4.3.13) and (4.3.14) into the expression for $a_0(\cdot,\cdot)$

$$\begin{aligned}a_0(u,u) =&\frac{1}{2}\int_\Omega vS^2\left(\frac{\partial u}{\partial S}\right)^2+\frac{1}{2}\sigma^2\int_\Omega v\left(\frac{\partial u}{\partial v}\right)^2+\rho\sigma\int_\Omega vS\left(\frac{\partial u}{\partial v}\frac{\partial u}{\partial S}\right)\\ &+\frac{1}{2}\int_\Omega(r-v-\frac{\rho\sigma}{2})u^2+\frac{1}{2}\int_\Omega(\kappa-\lambda-\frac{\rho\sigma}{2})u^2+r\int_\Omega u^2. \end{aligned}\tag{4.3.15}$$

The mixed term can be handled by Young's inequality

$$\begin{aligned}\left(\sqrt{v}S\frac{\partial u}{\partial S}\right)\left(\sigma\sqrt{v}\frac{\partial u}{\partial v}\right)\geq&-\left|\left(\sqrt{v}S\frac{\partial u}{\partial S}\right)\left(\sigma\sqrt{v}\frac{\partial u}{\partial v}\right)\right|\\ \geq&-\frac{1}{2}vS^2\left(\frac{\partial u}{\partial S}\right)^2-\frac{1}{2}\sigma^2v\left(\frac{\partial u}{\partial v}\right)^2\end{aligned}$$

which inserted into (4.3.15) yields

$$\begin{aligned} a_0(u,u) \geq & \frac{1}{2}(1-\rho)\int_\Omega vS^2\left(\frac{\partial u}{\partial S}\right)^2 + \frac{1}{2}(1-\rho)\sigma^2\int_\Omega v\left(\frac{\partial u}{\partial v}\right)^2 \\ & + \frac{1}{2}(3r-\kappa-\lambda-\rho\sigma)\int_\Omega u^2 - \frac{1}{2}\int_\Omega vu^2. \end{aligned} \tag{4.3.16}$$

Recall the inequality $\int_\Omega vu^2 \leq \max_\Omega(v)\int_\Omega u^2$. The above is then possible to express as

$$a_0(u,u) \geq C|u|^2_{V_0} - c\|u\|^2_{L^2(\Omega)}, \qquad \forall u \in V_0, \tag{4.3.17}$$

where $|u|^2_{V_0} = \left(\int_\Omega vS^2\left(\frac{\partial u}{\partial S}\right)^2 + v\left(\frac{\partial u}{\partial v}\right)^2 + u^2\right)$. We cannot use $\|\cdot\|_{V_0}$ in the expression above because it does not contain the term $\int_\Omega \frac{u^2}{v}$ which is a part of $\|\cdot\|_{V_0}$.

Last but not least we will investigate the energy stability of the method. Before writing out the weak formulation the following mathematical tool is needed

$$(u_t,u) = \int_\Omega u_t u = \int_\Omega \frac{\partial}{\partial t}(u^2) = \frac{\partial}{\partial t}\int_\Omega u^2 = \frac{\partial}{\partial t}\|u\|^2_{L^2(\Omega)}.$$

Writing out the weak formulation (4.1.14) and using the inequality (4.3.17) results in

$$\frac{\partial}{\partial t}\|u\|^2_{L^2(\Omega)} + C|u|^2_{V_0} - c\|u\|^2_{L^2(\Omega)} \leq (u_t,u) + a_0(u,u) = 0. \tag{4.3.18}$$

The equation above is multiplied by e^{-ct} which makes it possible to express it on the form

$$\frac{\partial}{\partial t}\left(e^{-ct}\|u(t)\|^2_{L^2(\Omega)}\right) + e^{-ct}C|u(t)|^2_{V_0} \leq 0. \tag{4.3.19}$$

Integrating from 0 to t we obtain

$$\begin{aligned} [e^{-c\tau}\|u(\tau)\|^2_{L^2(\Omega)}]^t_0 + \int_0^t e^{-ct}C|u(\tau)|^2_{V_0} \leq 0 \\ \Rightarrow e^{-ct}\|u(t)\|^2_{L^2(\Omega)} + \int_0^t e^{-ct}C|u(\tau)|^2_{V_0} \leq \|u_0\|^2_{L^2(\Omega)}. \end{aligned} \tag{4.3.20}$$

This expression shows energy stability and can also be used to prove uniqueness. By assuming that w_1 and w_2 are two solutions to the same initial value problem and inserting $w_3 = w_1 - w_2$ into the expression above we get

$$e^{-ct}\|w_3\|^2_{L^2(\Omega)} + \int_0^t e^{-ct}C|w_3(\tau)|^2_{V_0} \leq 0, \tag{4.3.21}$$

because the initial data are the same and hence $w_1 = w_2$. By inserting u_h into (4.3.20) we also have energy stability for the finite element method with the boundary conditions described in this section.

4.4 Numerical results

The finite element solver has been written in Matlab as described in section 4.2. Matrix and vector operations are used as often as possible to make the code more efficient in this case also. Although these ground rules are implemented in both FEM and FDM, there is a speed advantage in favor of FDM. This is mainly connected to the matrix construction as there is more work to be done when considering the elements. In the FDM solver we iterate over the I nodes and construct a series of vectors which make up the matrix M with a couple of insertions. With FEM on the other hand we have to iterate over all elements and the number of elements are approximately twice as many as the number of nodes. $I \cdot J$ nodes gives $(I-1)(J-1)$ squares which again gives $2(I-1)(J-1)$ triangles. Not only are there twice as many iterations, but for every iteration the basis functions are obtained by solving a 3×3 system of equations followed by Gaussian quadrature on all nine combinations. It is therefore difficult to take advantage of any buil-in Matlab functions which is done in the FDM solver.

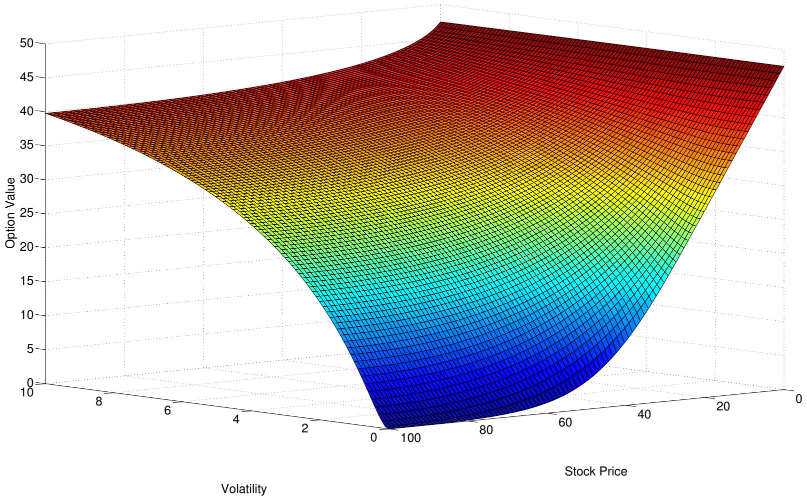

Figure 4.1: Surf-plot of the European vanilla put using the Heston model. Values used are $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 0.1, \lambda = 0.1$ and $\rho = 0.5$. S, v, t range from 0 to $[100, 10, 1]$ with $[100, 100, 40]$ number of nodes.

This figure is almost identical to the one in the FDM case, but for FEM the boundaries are plotted as well. With FEM we evaluate the boundary integrals at the boundary, which requires nodes at the boundary. FDM on the other hand is evaluated for the internal nodes with boundary conditions added to the matrix or the right hand side (artificial boundary). We can spot the same characteristics as in the FDM case. At $S = 0$ the solution is flat and sinking as a function of time with rate e^{-rt} and the partial derivatives tends to zero at the boundaries $v = v_{max}$

and $S = S_{max}$. We also have that $P(S, v_1, T) \leq P(S, v_2, T)$ for all v_1 and $v2$ so that $v_1 \leq v_2$. This is easily seen in the next plot which is viewed from the S-axis.

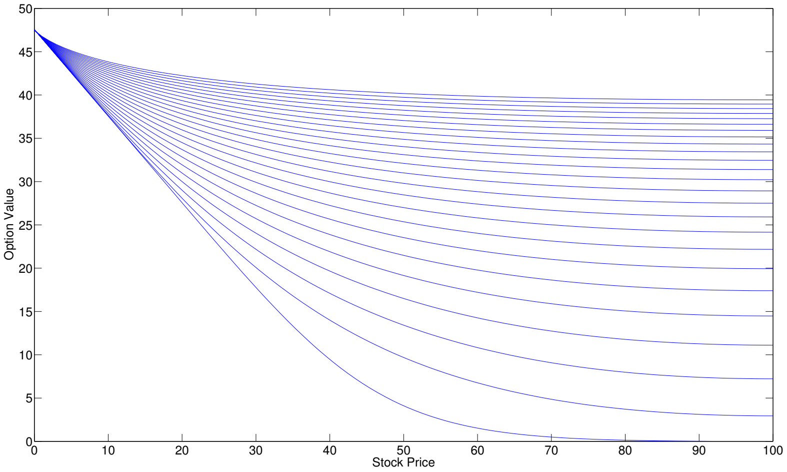

Figure 4.2: Surf-plot of the European vanilla put using the Heston model. Values used are $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 1, \lambda = 0.1$ and $\rho = 0.5$. Solution viewed from S-axis with every fourth v plotted at maturity time $T = 1$.

What this means is that the value of the option is higher for larger fluctuations of the underlying asset. This is a standard property of options because the increased difficulty of predicting the behavior of the underlying asset requires a higher price for the option because of the additional risk. Another interesting observation that can be made is that there are no oscillations at $v = v_{max}$ when using FEM which can be seen in the plot on the next page.

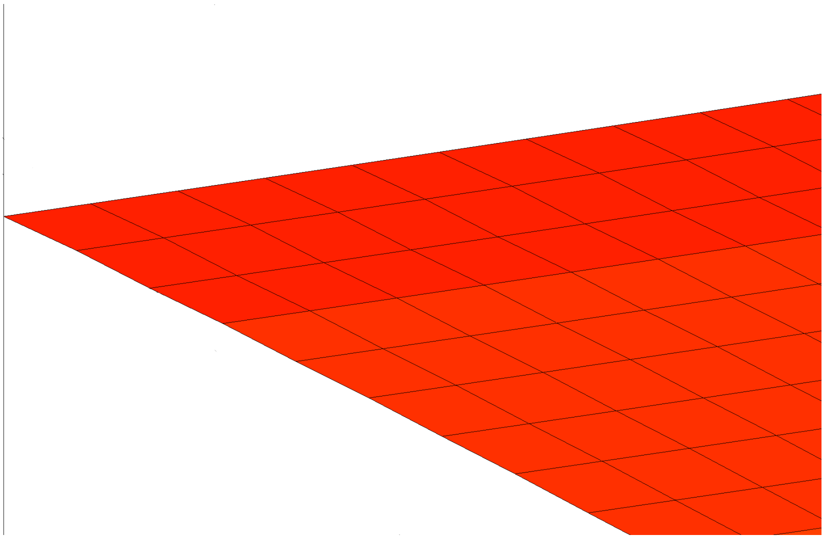

Figure 4.3: Closeup look at $v = v_{max}$ (top pf the plot) and $S = S_{max}$ (left side of the plot)

The analysis performed by Quarteroni indicates that oscillations will occur for FEM also. His analysis is not directly transferable to our case with non-constant coefficients and two spatial dimensions, and the plot above shows that there are no oscillations in the solution obtained by FEM. However, by testing FEM for different boundary conditions it produces oscillations similar to FDM when using $P = 0$ on the boundaries $S = S_{max}$ and $v = v_{max}$. In Appendix 7.5. we point out that the convection term associated with $\frac{\partial P}{\partial v}$ is the cause for oscillations close to $v = v_{max}$. We also show that the v-component of b (4.1.4) is negative for $v > 0.18$ resulting in a positive contribution from the convection term (for $v > 0.18$). This contribution increases as $v \to v_{max}$ and results in oscillations close to the boundary. However, when applying the boundary conditions in (4.1.8) we get a negative contribution at the boundary from the diffusion term, which according to the plot above is sufficient to avoid oscillations created by the convection term. Oscillations may still occur for different grids and constants, but the numerical results indicate that FEM is more resistant to oscillations than FDM:

The construction of the convergence plots are the same as in the FDM case, but with a reference solution constructed by FEM. The a priori error estimates performed by Quarteroni in [9] indicates second order convergence in space for the element method when using linear elements. It is not certain that this applies to our case with non-constant coefficients and this analysis is proposed as further work. However, we will assume second order convergence and compare the error to a square function. On the next page we have the loglog plots for space and time.

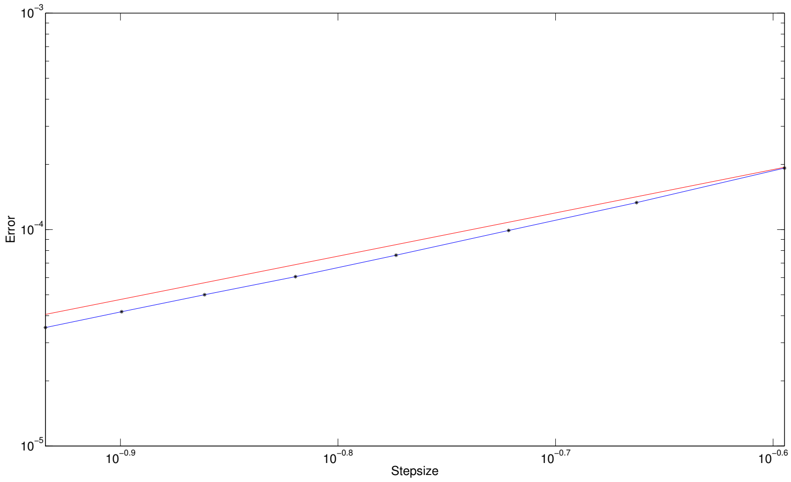

Figure 4.4: Loglog-plot for convergence in space. The red line is stepsizes plotted against stepsizes squared and the blue line is the error plotted against the stepsizes

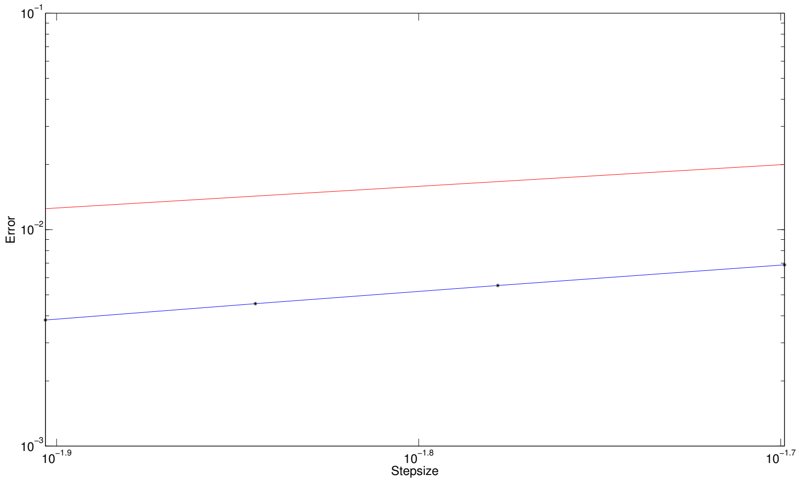

Figure 4.5: Loglog-plot for convergence in time. The red line is stepsize plotted against stepsize the blue line is the error.

The loglog-plot illustrating convergence in space shows two parallel lines with a slight exception to the right. This is not a big deviation and when comparing the methods in Chapter 6 we will calculate the order numerically and discuss the convergence further. The loglog-plot for the time convergence indicates first order convergence as in the FDM case which is as expected since the implicit Euler scheme is applied in both cases.

Chapter 5

Finite Element Method for American Options

5.1 Weak formulation

In this section we will derive the weak formulation for the American option and we start by stating the linear complementary problem described in Section 2.2.

$$\mathcal{L}(P) \geq 0, \quad P \geq P_0, \tag{5.1.1}$$

$$\mathcal{L}(P)(P - P_0) = 0, \tag{5.1.2}$$

where $\mathcal{L}$ is the linear partial differential operator

$$\mathcal{L}(P) = \frac{\partial P}{\partial t} - \frac{1}{2}vS^2\frac{\partial^2 P}{\partial S^2} - \rho\sigma vS\frac{\partial^2 P}{\partial v\partial S} - \frac{1}{2}\sigma^2 v\frac{\partial^2 P}{\partial v^2} - rS\frac{\partial P}{\partial S} - \{\kappa[\theta - v] - \lambda v\}\frac{\partial P}{\partial v} + rP$$

To obtain the weak formulation of (5.1.1) and (5.1.2), we use the space V (4.1.12) from the European case and introduce the following subset of V:

$$\mathcal{K} = \{v \in V, v \geq P_0 \text{ in } \Omega\}. \tag{5.1.3}$$

In the same manner as in the European case we can multiply (5.1.1) by a smooth nonnegative test function and integrate over the domain Ω

$$0 \leq \frac{d}{dt}\int_\Omega P(S,t)\phi(S)dS + a(P(t),\phi), \tag{5.1.4}$$

where a is the same as in (4.1.10).

Introducing $\mathcal{K}_0$ as the cone of nonnegative functions in V, the set $\mathcal{K}$ is exactly $\mathcal{K} = P_0 + \mathcal{K}_0$. With this in order we see from (5.1.4) that the weak formulation of (5.1.1) and (5.1.2) can be expressed as:

Find $P \in \mathcal{C}^0([0,T]; L^2(\Omega)) \cap L^2(0,T;\mathcal{K})$, such that $\frac{\partial P}{\partial t} \in L^2(0,T;V')$, satisfying

$$P_{|t=0} = P_0 \text{ in } \Omega \tag{5.1.5}$$

for a.e t $\in (0,T)$,

$$\forall u \in \mathcal{K}_0, \ \left(\frac{\partial P}{\partial t}(t), u\right) + a(P(t), u) \geq 0. \tag{5.1.6}$$

We have $\mathcal{K} = P_0 + \mathcal{K}_0$ which results in (5.1.6) being equivalent to

$$\forall u \in \mathcal{K}, \ \left(\frac{\partial P}{\partial t}(t), u - P_0\right) + a(P(t), u - P_0) \geq 0. \tag{5.1.7}$$

Inserting $u = P(t)$

$$\left(\frac{\partial P}{\partial t}(t), P(t) - P_0\right) + a(P(t), P(t) - P_0) = 0. \tag{5.1.8}$$

By subtracting (5.1.8) from (5.1.7) we get

$$\forall u \in \mathcal{K}, \ \left(\frac{\partial P}{\partial t}(t), u - P(t)\right) + a(P(t), u - P(t)) \geq 0. \tag{5.1.9}$$

From this the weak formulation (5.1.5) and (5.1.6) is equivalent to (5.1.5) and (5.1.9). In the next section we will discretize the weak formulation and solve the option pricing problem by applying the Brennan Schwartz algorithm.

5.2 Discretization

We use the same triangulation as in the European case and denote by $\mathcal{K}_h$ the space of continuous piecewise linear functions on the triangulation of Ω. Applying the implicit Euler scheme results in the following:

Find $(P_h^m)_{0\leq m\leq M}, P_h^m \in \mathcal{K}_h$, satisfying

$$P_h^0 = P_0$$

and for all $m, 1 \leq m \leq M$

$$\forall u \in \mathcal{K}_h,\ \left(P_h^n - P_h^{n-1}, u - P_h^n\right) + \Delta t a\left(P_h^n, u - P_h^n\right) \geq 0. \tag{5.2.1}$$

With $(\phi_i)_{i=0,\ldots,N}$ as the nodal basis of V_h we can write the solution P_h as a weighted sum of these, i.e $P_h = \sum_{i=1}^N P_i\phi_i$ with N being the total number of nodes $(I \cdot J)$. In the same manner as in the European case we define the mass matrix $\boldsymbol{M} = [m_{ij}] = (\phi_i, \phi_j)$, the stiffness matrix $\boldsymbol{A} = [a_{ij}] = [a(\phi_i, \phi_j]$, and the vectors

$$P^n = (P_h^n(S_0), ..., P_h^n(S_N))^T \text{ and } P^0 = (P_0(S_0), ..., P_0(S_N))^T.$$

Solving (5.2.1) by applying the Brennan Schwartz algorithm consists of the following:

1. Given P^{n-1} solve the problem for the European option
$$\left(\boldsymbol{M}\left(\tilde{P}^n - P^{n-1}\right) + \Delta t\boldsymbol{A}\tilde{P}^n\right) = 0. \tag{5.2.2}$$
2. Check every element of the solution
$$P^n(i) = \max(P^0(i), \tilde{P}^n(i)). \tag{5.2.3}$$
3. Repeat every time iteration.

Note that the application of the Brennan Schwartz algorithm makes the introduction of the variational inequalities somewhat superfluous. The reason why they are included is to state the American problem on the general form in order to have the possibility to perform further analysis and implement other algorithms such as SOR or front tracking for handling of the American condition.

5.3 Numerical results

The approach for solving the problem for the American option using FEM is very similar to the European case although the theory and formulation of the problem is more complex. Exactly as in the FDM case we apply the Brennan Schwartz algorithm to ensure $P \geq P_0$ which causes the American option to have a higher price than the European as we have seen before. Below we see a solution for the American option with the exercise boundary marked as a blue line.

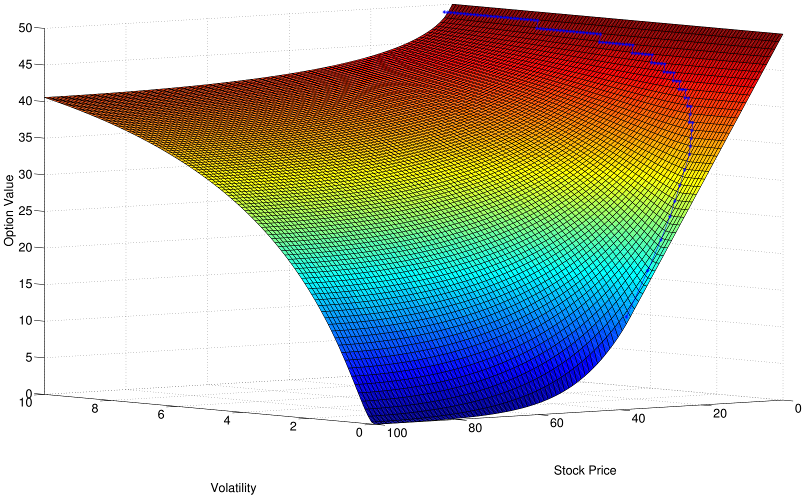

Figure 5.1: Surf-plot of the American vanilla put using the Heston model. Values used are $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 1, \lambda = 0.1$ and $\rho = 0.5$. Solution viewed from S-axis with every fourth v plotted at maturity time $T = 1$. The blue line represents the exercise boundary

This might look slightly different than Figure 3.6 in the FDM case, but this is because the boundaries are included in the plot when using FEM as described earlier in the paper. It may therefore seem like there is a larger area where $P < P_0$ is active, but this is not the case. Below we see the same solution viewed from the S-axis for every fourth value of v, this point of view will make it easier to spot the difference between the European and American option.

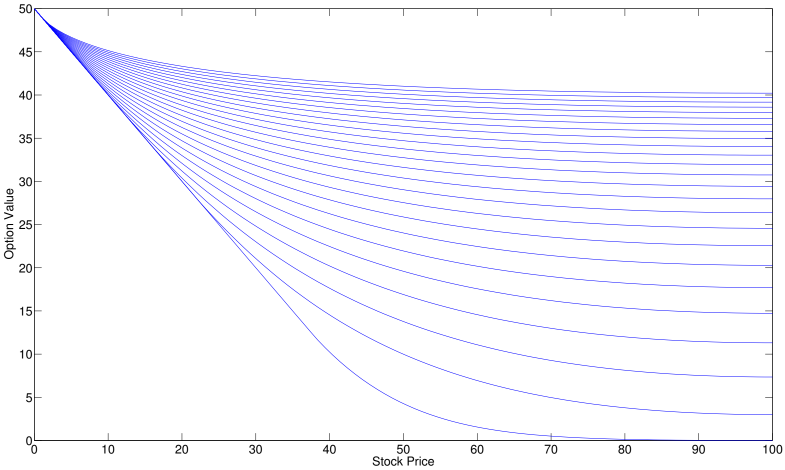

Figure 5.2: Surf-plot of the American vanilla put using the Heston model. Values used are $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 1, \lambda = 0.1$ and $\rho = 0.5$. Solution viewed from S-axis with every fourth v plotted at maturity time $T = 1$.

We see that the American option remains constant at at the boundary $S = 0$ while the European option sinks with a factor e^{-rt} and we also see that the curves for the chosen values of v intercept for larger values of S in the American case. This is because the beginning of the curves in the European case are below P_0 and are therefore shifted to the right in the American case.

As explained earlier in the paper the solver for the American case will have the same accuracy and order of convergence as in the European case. This is because it is in fact the European problem that is solved in the continuation region and when investigating the convergence we have to use nodes from this region. If the nodes are chosen from the region where the American condition is active it means that the nodes are directly affected by the Brennan Schwartz algorithm and will therefore be set to a value which does not produce a valid convergence plot.

Chapter 6

Concluding remarks

6.1 Comparing FEM to FDM

In this section we will use the reference solutions from the convergence plots to calculate the error for FEM and FDM for different grids and number of time iterations. From the truncation error in the FDM case we have that the error scales like $\mathcal{O}(\Delta t + h^2 + k^2)$ which is also assumed to apply to FEM. Denote ϵ_1 as the error obtained by using $\Delta t_1, h_1^2, k_1^2$, and ϵ_2 as the error obtained by $\Delta t_2, h_2^2, k_2^2$ where $\Delta t_2 = \frac{\Delta t_1}{4}, h_2^2 = \frac{h_1^2}{2}, k_2^2 = \frac{k_1^2}{2}$. Comparing these two erros gives

$$\epsilon_2 = \mathcal{O}(\Delta t_2 + h_2^2 + k_2^2) = \frac{1}{4}\mathcal{O}(\Delta t_1 + h_1^2 + k_1^2) = \frac{1}{4}\epsilon_1.$$

Doubling the number of nodes in space and quadrupling the time iterations should result in the error dropping to a fourth of its previous value for each sample. Because the matrix construction takes much more time in FEM we will measure two times, namely the time for the matrix construction and the time for solving the system. The order of the methods are computed by assuming an error on the form

$$\|u - u_h\|_{L^2} = Ch^\beta, \tag{6.1.1}$$

where u is the reference solution and $h = k$. By comparing the errors from two different grids we can express the order as

$$\beta = \frac{ln(\frac{\|u-u_{h_1}\|_{L^2}}{\|u-u_{h_2}\|_{L^2}})}{ln(\frac{h_1}{h_2})}. \tag{6.1.2}$$

From the table we see that the order is below 2 which was indicated by the convergence plot in the FDM case. This is with high certainty connected to the interpolation of the solutions obtained from coarse grids. We see that the order approaches 2 as the grid gets more refined and end up at 2.1 for the finest grid.

Grid	Iterations	Matrix contruction	Solving	Error	Order
20×20	4	0.05s	0.01s	0.0679	
40×40	16	0.10s	0.23s	0.0238	1.51
80×80	64	0.25s	3.67s	0.0077	1.63
160×160	256	0.58s	89s	0.0017	2.14

Table 6.1: Error drop in FDM when increasing nodes and time iterations. Reference solution is 256×256 with 512 time iterations on the domain $[0, 15] \times [0, 15]$ with $T = 0.1$.

The coarse grids are more sensitive to oscillations and even though we extract a vector away from the boundary it seems like the interpolation overestimates the solution. The lower order is not connected to the stability of the explicit method as the implicit method produces the same results. We also see that FDM is very fast when it comes to constructing the matrix.

Grid	Iterations	Matrix contruction	Solving	Error	Order
20×20	4	3.1s	0.0014s	0.0447	
40×40	16	9.0s	0.029s	0.0121	1.89
80×80	64	50.8s	1.57s	0.0028	2.11
160×160	256	650s	80s	3.95e-4	2.82

Table 6.2: Error drop in FEM when increasing nodes and time iterations. Reference solution is 256×256 with 512 time iterations on the domain $[0, 15] \times [0, 15]$ with $T = 0.1$.

FEM is not affected by oscillations and we get approximately order 2 for the solutions obtained from coarse grids. For the finest grid on the other hand we get 2.8. When checking the convergence rate for different norms, grids and domains it has been observed that the convergence varies around 2 and this may be an example of an outlying point. However, the analysis considering the order of convergence for the element method has not been done in this paper which makes it difficult to rule out other possibilities. The high convergence can also we a case of superconvergence which is described in [15].

Considering the runtime of the solvers we see that FDM constructs the matrix much faster them FEM. This is a problem connected to the use of Matlab and FEM can most definitely be optimized further, for example by constructing the matrices in C++. If we consider solving time and convergence we see that FEM performs better than FDM and by optimizing the matrix construction it may even be faster because the solving time is faster. The conclusion to be drawn from this is that FDM with explicit boundary is easily implemented and is also a fast solver when taking advantage of Matlabs built-in functions. FEM on the other hand is more difficult to implement, especially when considering the optimization of the matrix construction, but will result in a better solver. Recall that eliminating the oscillations from FDM will cost an entire order of convergence in v and will therefore be even slower when taking the quality of the solution into account.

6.2 Further work

In order to take advantage of Matlab's built-in functions it would be better to calculate the entries analytically. In that way there will be no local linear system of equations and no Gaussian quadrature to be performed for each element. The Heston model do not have constant coefficients which make the process of calculating the integrals analytically quite difficult. There are examples of the use of the transformation $x = log(S/K)$ (Winkler, Apel, Wystup ,[5]). This entirely removes S from the equation, hence simplifying the evaluation of the integrals. Even though S is removed the coefficients are still not constant because of v and we still need some form of numerical approximation for the integrals. This means that the most significant runtime enhancement would probably be obtained by using a different programming language such as C++. It would be interesting to compare FEM to FDM with an optimized method for constructing the matrices in FEM.

When it comes to the quality of the solutions we get oscillations when applying first order central difference. This is discussed in Appendix 7.5 and is fixed by applying the upwind scheme. It would be interesting to investigate this phenomenon further and find the exact requirements for when the oscillations occur and experiment with different strategies for eliminating them such as the mass-lumping technique and artificial diffusion (Quarteroni, [9]).

The analysis of the Heston model when considering the element method is quite difficult which is mainly connected to the boundary terms in the bilinear form. Performing a detailed analysis considering stability, convergence, coercivity and continuity is a good example for further work. This would be a more theoretical approach and could be combined with investigating the possibility of calculating the matrix entries for the element method analytically, and obtaining the stability requirement for the FDM method.

Chapter 7

Appendices

7.1 Itô's lemma

For an Itô drift-diffusion process

$$dX_t = \mu_t dt + \sigma_t dB_t$$

and any twice differentiable scalar function $f(t,x)$ of two real variables t and x, Itô's lemma for higher dimensions states

$$df(t,X_t) = \dot{f}(t,X_t)dt + \sum_{i=1}^{d} f_i(t,X_t)dX_t^i + \frac{1}{2}\sum_{i,j=1}^{d} f_{i,j}(t,X_t)dX_t^i dX_t^j.$$

where $X_t = (X_{t,1}, X_{t,2}, ..., X_{t,n})$ is a vector of Itô processes, $\dot{f}(t,X)$ is the partial differential w.r.t t,$f_{i,j}$ is the partial differential w.r.t the spatial dimensions, $\nabla_X^T f$ is the gradient of f w.r.t X, and $\nabla_X^2 f$ is the Hessian matrix of f w.r.t X. This immediately implies that $f(t,x)$ is itself an Itô drift-diffusion process. The last term consists of a product of Itô drift-diffusion processes $(dX_t^i dX_t^j)$, and we have that $(dt)^2 = 0$, $dtdB_t = 0$ and $dB_t^i dB_t^j =$Cov$[dB_t^i, dB_t^j]dt$.

7.2 Ornstein-Uhlenbeck process

The Ornstein-Uhlenbeck process is an example of a Gaussian process that has a bounded variance and admits a stationary probability distribution, in contrast to standard Brownian motion; the difference between the two is in their "drift" term. In standard Brownian motion the drift term is constant, but for the O-U process it is dependent on the current value of the process. If the current value is greater than the mean we have a positive drift term and opposite when it is less than the mean.

An O-U process x_t, satisfies the following stochastic differential equation:

$$dx_t = \theta(\mu - x_t)dt + \sigma dB_t$$

where $\theta > 0$, μ and $\sigma > 0$ are parameters and B_t denotes Brownian motion (Stein & Stein, [13]). In financial mathematics we typically have that μ represents the mean value supported by fundamentals, σ the degree of volatility around it caused by unexpected changes in the market, and θ the rate by which these unexpected changes dissipate and the variable reverts towards the mean.

7.3 CIR process

The CIR process is a Markov process with continuous paths defined by the following stochastic differential equation:

$$dr_t = \theta(\mu - r_t)dt + \sigma\sqrt{r_t}dB_t$$

where B_t denotes Brownian motion and θ, μ and σ are parameters (Cox, Ingersoll and Ross, [12]). The parameter θ corresponds to the speed of adjustment, μ to the mean and σ to volatility.

This process can be defined as a sum of squared Ornstein-Uhlenbeck and is widely used in finance to model short term interest rate or to model stochastic volatility in the Heston model.

7.4 Gaussian Quadrature

Integrals may be of varying complexity depending on the problem at hand, and many of them does not even have a known analytical solution. Some integrals are possible to solve analytically, but of such computational complexity that it is impractical to do so. As such, one often refers to numerical integration schemes to do the core integration. The integration scheme that will be explained here is Gaussian quadrature for the two dimensional case which is relevant for this task. In two or more dimensions the gauss quadrature takes the form

$$\int_{\hat{\Omega}} g(\boldsymbol{z})d\boldsymbol{z} \approx \sum_{q=1}^{N_q} \rho_q g(\boldsymbol{z}_q),$$

where N_q is the number of integration points, $(z)_q$ are the Gaussian quadrature points and ρ_q are the associated Gaussian weights. In two dimensions we map to barycentric coordinates $\boldsymbol{\zeta}$, $\boldsymbol{x} = \zeta_1\boldsymbol{p}_1 + \zeta_2\boldsymbol{p}_2 + \zeta_3\boldsymbol{p}_3$ where $\boldsymbol{p}_i, i = 1, 2, 3$ is the corner points of the triangle.

N_q	$(\zeta_1\zeta_2\zeta_3)$	ρ
1-point rule	(1/3,1/3,1/3)	1
3-point rule	(1/2,1/2,0)	1/3
	(1/2,0,1/2)	1/3
	(0,1/2,1/2)	1/3
4-point rule	(1/3,1/3,1/3)	-9/16
	(3/5,1/5,1/5)	25/48
	(1/5,3/5,1/5)	25/48
	(1/5,1/5,3/5)	25/48

(7.4.1)

7.5 Upwinding scheme

The analysis considering oscillations performed by Quarteroni in [9] is done by investigating the one-dimensional diffusion-transport problem

$$-\mu u'' + bu' = 0 \tag{7.5.1}$$

and the one-dimensional diffusion-reaction problem

$$-\mu u'' + \sigma u. \tag{7.5.2}$$

These problems are investigated separately and Quarteroni shows that the Péclet number defined as

$$\mathbb{P}e = \frac{|b|h}{2\mu} \qquad \text{Diffusion-Transport} \tag{7.5.3}$$

$$\mathbb{P}e = \frac{\sigma h^2}{6\mu} \qquad \text{Diffusion-Reaction,} \tag{7.5.4}$$

will cause oscillations when $\mathbb{P}e > 1$. For a more detailed description of this phenomenon the reader is directed to [9]. In our case we have a full diffusion-transport-reaction equation with non-constant coefficients which makes it uncertain if this can be applied directly. Without investigating the expression for $\mathbb{P}e$ we will implement one of the strategies proposed by Quarteroni which is upwinding. Recall that the oscillations occur for large values of v which makes it reasonable to assume that the reaction term is not causing oscillations. This term has a constant coefficient r which in most practical cases is very small and by definition less then or equal to 1. It is therefore not likely to assume that the reaction term suddenly generates oscillations for large values of v. It is more likely that the oscillations are caused by the $(\kappa[\theta - v] - \lambda v)\frac{\partial P}{\partial v}$-term. In the case where $b > 0$ Quarteroni proposes the following upwind scheme

$$\frac{\partial u}{\partial t} + ... + b\frac{P_{i,j} - P_{i,j-1}}{k} = 0. \tag{7.5.5}$$

b is $-(\kappa[\theta - v] - \lambda v)$ in our case and is positive for $v > \frac{\kappa\theta}{\kappa+\lambda}$. Inserting the constants used in the numerical results and also used by Achdou and Pironneau in [1] results in $v > \frac{1 \cdot 0.2}{1+0.1} = 0.18$. Because the oscillations only occur near $v = v_{max}$ we only apply the upwinding scheme to $\frac{\partial P}{\partial v}$ and the result is

$$(\frac{\partial P}{\partial v})_{i,j} \approx \frac{P_{i,j} - P_{i,j-1}}{k}. \tag{7.5.6}$$

This leads to the following changes in the matrix entries:

$$\begin{aligned}
a^n_{i,j} &= 1 + \Delta t(i^2 jk + \frac{\sigma^2 j}{k} + r - \kappa(\theta - jk) + \lambda jk) \\
c^n_{i,j} &= -\frac{\Delta t}{2}(i^2 jk - ri) \\
d^n_{i,j} &= -\frac{\Delta t}{2}(i^2 jk + ri) \\
e^n_{i,j} &= -\frac{\Delta t}{2k}(\sigma^2 j - 2\kappa(\theta - jk) + 2\lambda jk) \\
f^n_{i,j} &= -\frac{\Delta t}{2k}(\sigma^2 j) \\
b^n_{i,j} &= -\frac{\rho\sigma ij}{4}\Delta t.
\end{aligned}$$

The rest of the procedure is exactly as described in Chapter 3. The reduction in convergence order is due to the fact that the new Taylor expansion for the first order derivative becomes

$$\frac{P_{i,j} - P_{i,j-1}}{k} \approx \partial_v P_{i,j} - \frac{k}{2}\partial_v^2 P_{i,j}, \tag{7.5.7}$$

instead of

$$\frac{P_{i,j+1} - P_{i,j-1}}{2k} \approx= \partial_v P_{i,j} + \frac{k^2}{6}\partial_v^3 P_{i,j}. \tag{7.5.8}$$

The new expression results in the error scaling as $\mathcal{O}(\Delta t + h^2 + k)$ instead of $\mathcal{O}(\Delta t + h^2 + k^2)$. Finally we have the numerical results for the application of the upwind scheme.

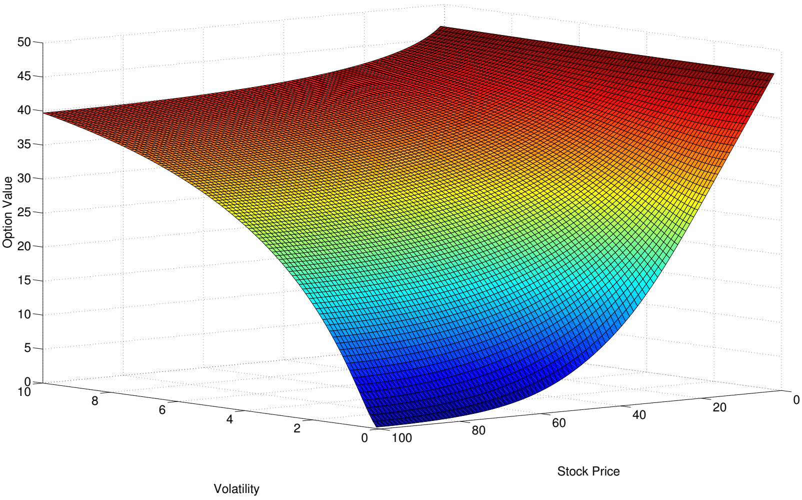

Figure 7.1: Surf-plot of the European vanilla put using the Heston model with the upwinding scheme. $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 0.1, \lambda = 0.1$ and $\rho = 0.5$. S, v, t range from 0 to $[100, 10, 1]$ with $[100, 100, 40]$ number of nodes.

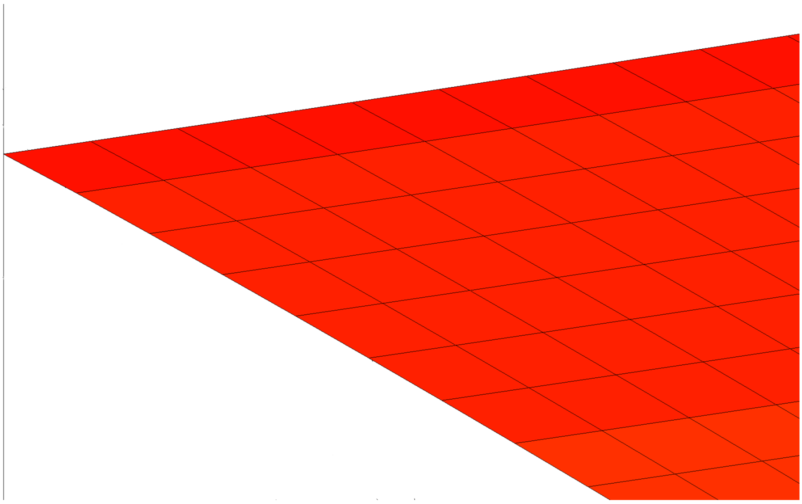

Figure 7.2: Surf-plot of the European vanilla put using the Heston model with the upwinding scheme. $K = 50, \sigma = 0.1, r = 0.05, \theta = 0.2, \kappa = 0.1, \lambda = 0.1$ and $\rho = 0.5$. S, v, t range from 0 to $[100, 10, 1]$ with $[100, 100, 40]$ number of nodes.

From the plots on the previous page we see a similar solution as in Chapter 3 but without oscillations. This confirms the hypothesis of the $\frac{\partial P}{\partial v}$-term being the cause of the oscillations in this case. Other strategies can also be applied and further analysis of this phenomenon is proposed as further work.

Bibliography

[1] Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*

[2] Steven L. Heston, *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*

[3] T. Mikosch, *Elementary Stochastic Calculus with Finance in View* (Vol. 6), World Scientific 1998

[4] C. O'Sullivan and S. O'Sullivan, *Pricing Options under Heston's Stochastic Volatility Model via Accelerated Explicit Finite Differencing Methods*

[5] G. Winkler, T. Apel and U. Wystup, *Valuation of Options in Heston's Stochastic Volatility Model Using Finite Element Methods*

[6] M. Brennan and E. Schwartz, *The valuation of American put options*, Journal of Finance, 32, (1977), pp. 449-462 other information

[7] Y. Achdou and O.Pironneau *Partial Differential Equations for Option Pricing*

[8] S. Lin *Finite Difference Schemes for Heston Model*

[9] A. Quarteroni *Numerical Models for Differential Problems*

[10] S. Ikonen and J. Toivanen, 2007b. *Efficient numerical methods for pricing American options under stochastic volatility, Numerical Methods for Partial Differential Equations,* Vol. 24, Issue 1, 104-126

[11] N. Clarke and K. Parrott,*Multigrid for American option pricing with stochastic volatility*, Applied Mathematical Finance (2010)

[12] J. Cox, J. E. Ingersoll and S. A. Ross, (1985) *" A Theory of the Term Structure of Interest Rates," Econometrica* 53, 385-408

[13] E. M. and J. C. Stein, (1991), *"Stock Price Distributions with Stochastic Volatility: An Analytic Approach," Review of Financial Studies*,4,727-752

[14] G. D. Smith (1985) *"Numerical Solution of Partial Differential Equations: Finite Difference Methods", 3rd ed.*

[15] M. F. Wheeler and J. R. Whiteman *Superconvergent Recovery of Gradients on Subdomains from Piecewise Linear Finite-Element Approximations*, Numerical Methods for Partial Differential Equations 3, 65 (1987)