Nonlinear Black Scholes Modelling
– FDM vs FEM

d-fine

Christ Church College
University of Oxford

A thesis submitted in partial fulfilment of the requirements for the MSc in

Mathematical Finance

31st of March, 2004
Contents

1 Introduction 1

2 Uncertain Parameter Model 3

3 Discretisation in Time and Addressing the Non-Linearity 7

4 Bounding the Asset-Space 10

5 Formulation of the semi-discretised Problem 13

6 Finite Difference Method 15

6.1 Remark on the Case $d > 1$ 17

7 Finite Element Method 19

7.1 Introduction to Sobolev Spaces 20

7.2 Deriving a Weak Formulation 26

7.2.1 Reduced Regularity Assumptions 29

7.3 Defining the Finite Dimensional Space 30

7.3.1 $P_1$-Elements 31

7.3.2 $P_2$-Elements 32

7.3.3 One-Dimensional Quadratic Macro-Elements 34

7.3.4 Two-Dimensional Bicubic Hermite Elements 35

7.4 Remarks on the Implementation 36

8 Numerical Examples 38

8.1 Double Barrier Call Option with Fixed Parameters 38
8.2 Double Barrier Call Option with Uncertain Volatility .................. 40
8.3 Digital Call Option .................................................. 42
8.4 Exchange Option with Fixed Parameters ............................... 44
8.5 Exchange Option with Barrier and Uncertain Parameters .......... 47
8.6 Worst-of-Two Call Option with Barrier ................................. 51

9 Closing Remarks ........................... 55

A Program Code ............................... 57
A.1 Main Program for the One-Dimensional Problem .................... 57
A.2 Functions for the Finite Difference Scheme ......................... 59
A.3 Functions for the Finite Element Method (Quadratic Macro-Elements) ... 61
A.4 Problem Depending Functions: Double Barrier with Uncertain Volatility .. 64

B Analytic Solutions for Numerical Examples .......................... 66
B.1 European Double Barrier Call Option ............................... 66
B.2 European Digital Call Option ................................. 67
B.3 European Exchange Option ........................................ 67

References ........................................ 68
List of Figures

4.1 Comparison of different cut-off boundary conditions .................................. 12

7.1 Example of a regular and a non-regular triangulation ................................. 24

7.2 Two basis functions for one-dimensional $P_1$-elements ............................ 32

7.3 Nodes for $P_2$-elements ................................................................................ 33

7.4 Three basis functions for one-dimensional $P_2$-elements .......................... 33

7.5 Four basis functions for one-dimensional quadratic macro-elements .......... 35

7.6 Basis functions of the bicubic Hermite element ........................................... 36

8.1 Double Barrier Option with Fixed Parameters ........................................... 39

8.2 Double Barrier Option with Uncertain Volatility ......................................... 41

8.3 Digital Call Option with Fixed and with Uncertain Volatility ................. 43

8.4 Analytic Solution of the Exchange Option with Fixed Parameters ............. 45

8.5 Exchange Option with Fixed Parameters, solved using Finite Differences ... 46

8.6 Exchange Option with Fixed Parameters, solved using Finite Elements ...... 48

8.7 Exchange Option with Barriers and Uncertain Parameters, Option Value and First Derivatives ................................................................. 49

8.8 Exchange Option with Barriers and Uncertain Parameters, Second Derivatives ................................................................. 50

8.9 Worst-of-Two Call Option with Barrier and Uncertain Parameters, Option Value and First Derivatives ................................................................. 52

8.10 Worst-of-Two Call Option with Barrier and Uncertain Parameters, Second Derivatives ................................................................. 53
Chapter 1

Introduction

In the classical Black-Scholes model, the financial parameters, like the volatilities and correlations, are assumed to be known. These are very strong assumptions that are unrealistic in the real world. Hence, a common extension of the Black-Scholes model assumes one or more of the parameters to be random. This leads to partial differential equations that include the financial parameters as unknowns.

Another approach is the uncertain parameter model, where the financial parameters are assumed to lie between some known upper and lower bound. Solving the problem for the worst case scenario leads to a non-linear generalisation of the classical Black-Scholes equation. Several works have addressed this type of problem. One asset problems with an uncertain volatility are discussed in [ALP95, AB99, Wil98, WO98, Top01a, PFV03]. The works [Wil98, Top01a, Top01b] analyse two assets problems where the correlation is only known to lie within a given interval. However, in these cases, the volatility of both asset is always assumed to be known. Finally, in [Wil98, WO98], problems with an uncertain risk-free interest rate are addressed. In all cases, the resulting problem is a non-linear generalisation of the classical Black-Scholes equation. It reduces to the standard form when the upper and lower bound for the parameter are chosen to be equal. Although several of these works state that a generalisation of these different types of problems to a problem where more than one parameter is uncertain is 'straight forward', there seems to be no work addressing this generalisation, neither analytically nor numerically. Hence, in this work, we turn our attention to the pricing equation of financial instruments depending on several assets whose volatilities as well as their correlations are uncertain. It is a partial differential equation of order two of the form

\[
\frac{\partial V}{\partial t} = rV - \bar{\mu} \cdot \nabla_S V - \sum_{j=1}^{d} \sum_{k=1}^{d} A_{jk} \partial^2_{S_j S_k} V,
\]

(1.1)
where the functions $A_{jk}$ depend non-linearly on $\Gamma$, the second derivative of $V$ with respect to $\bar{S}$. It is found that this formulation is a non-trivial generalisation of the problems addressed in the works mentioned above and, at least when solving numerically, needs some special attention.

To solve the problem numerically, in this work, we focus on two methods for the asset space discretisation. One, the finite difference method (FDM), is widely used for all types of pricing equations, [Izv98, AB99, Hul99]. The other method, the finite element method (FEM), is a common tool used for example in mechanical engineering problems, [Bra97, BS94], for almost 30 years but has only recently been chosen for solving partial differential equations arising in the financial world. Some of those works are [FVZ99, JS98, Top03, KKK, KN00]. In this thesis, the FEM is discussed and compared to the FDM. Special attention is payed to the FEM for the non-linear uncertain parameter model.

The remainder of this work starts with the derivation of the uncertain parameter model in chapter 2. In chapter 3, the resulting partial differential equation is discretised in time. Here, we also address the non-linearity of the problem and how this feature will be handled numerically. We briefly address the problem of cutting off the unbounded asset space for computational purposes in chapter 4. Out of convenience to the reader, in chapter 5, we restate the semi-discrete problem resulting from the steps in chapters 3 and 4.

Chapter 6 then discusses the discretisation in the asset space using the finite difference method. As the FDM is a common numerical tool in mathematical finance, details will be omitted. The asset space discretisation using finite elements is described in detail in chapter 7. We will show that the choice of finite elements depends on whether we consider the classical model with fixed parameters or the non-linear equation resulting from the uncertain parameter model.

The FDM and the FEM are compared for several numerical examples in chapter 8. They support the theoretical results from chapter 7. The work closes in chapter 9 with a conclusion and suggestions for future work. In the appendix A, we provide the Matlab code of the FDM and the FEM used for the one dimensional examples in chapter 8.
Chapter 2

Uncertain Parameter Model

To derive the partial differential equation for the uncertain parameter model, we start in the same way as for the derivation of the classical Black-Scholes equation, as can be found in the standard literature, [WDH93, Hul99]. Assume, we have $d$ assets that follow the usual stochastic differential equation (SDE)

$$dS_j = \mu_j S_j \, dt + \sigma_j S_j \, dX_j, \quad (j = 1, \ldots, d),$$

where the $dX_j$ are correlated as

$$E(dX_j \, dX_k) = \rho_{jk} \, dt, \quad (j, k = 1, \ldots, d).$$

However, for the volatilities $\sigma_j$ and the correlations $\rho_{jk}$, only an upper and lower bound are known:

$$0 < \sigma_j^- \leq \sigma_j \leq \sigma_j^+, \quad \rho_{jk}^- \leq \rho_{jk} \leq \rho_{jk}^+, \quad \text{where} \quad |\rho_{jk}^\pm| \leq 1 \text{ and } \rho_{jj}^\pm = 1.$$

Then, using Itô’s lemma, the option price $V(S_1, \ldots, S_d, t)$ satisfies the SDE

$$dV = \partial_t V \, dt + \sum_{j=1}^d \partial_{S_j} V \, dS_j + \frac{1}{2} \sum_{j=1}^d \sum_{k=1}^d \rho_{jk} \sigma_j \sigma_k S_j S_k \partial_{S_j} \partial_{S_k} V \, dt.$$

Setting up a hedged portfolio consisting of a long position in the option $V$ and short positions in all underlyings leads to

$$\Pi = V(S_1, \ldots, S_d, t) - \sum_{j=1}^d \Delta_j S_j.$$

Then,

$$d\Pi = \partial_t V \, dt + \sum_{j=1}^d \left( \partial_{S_j} V \, dS_j - \Delta_j \, dS_j \right) + \frac{1}{2} \sum_{j=1}^d \sum_{k=1}^d \rho_{jk} \sigma_j \sigma_k S_j S_k \partial_{S_j} \partial_{S_k} V \, dt.$$
Choosing $\Delta_j = \partial S_j V$ yields an instantaneous risk-free portfolio, i.e.

$$d\Pi = \partial_t V dt + \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \rho_{jk} \sigma_j \sigma_k S_j S_k \partial S_j \partial S_k V dt.$$ 

In contrast to the regular derivation of the Black-Scholes equation, here $\sigma_j$ and $\rho_{jk}$ are uncertain. So, knowing $V$ does not imply knowing $\Pi$. Instead, in every infinitesimal time step, we assume those values for the parameters that lead to a minimal growth in the portfolio. This yields the following worst-case-scenario model

$$\min_{\sigma_j \leq \sigma_j \leq \sigma_j^+ , \rho_{jk} \leq \rho_{jk} \leq \rho_{jk}^+ } (d\Pi) = r \Pi dt. \quad (2.1)$$

If $V^-$ denotes the worst-case solution, we then have

$$\partial_t V^- + \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \rho_{jk} \sigma_j \sigma_k S_j S_k \partial S_j \partial S_k V^- = r V^- - r \sum_{j=1}^{d} S_j \partial S_j V^-.$$ 

If in (2.1), we took the maximum instead of the minimum we would get a best-case-scenario model. Since, financially, it only makes sense to look at the worst case, in the following we will restrict ourselves to the worst-case scenario and write $V$ for $V^-$. 

The next task is to find the minimising $\sigma_j$ and $\rho_{jk}$, i.e. to minimise the function

$$f(x_1, \ldots , x_d, y_{11}, \ldots, y_{dd}) = \sum_{j=1}^{d} \sum_{k=1}^{d} y_{jk} x_j x_k S_j S_k \partial S_j \partial S_k V$$

on a $(d + d^2)$-dimensional (closed and bounded) rectangle. Note that $x_j > 0$, $y_{jj} = 1$, and $y_{jk} = y_{kj}$. Either, the minimiser lies truly inside the rectangle — in which case we have a local minimum with $\nabla f = 0$ — or on the boundary of the $(d + d^2)$-dimensional rectangle (but not necessarily at a corner!)

Taking the partial derivative of $f$ with respect to $y_{jk}$, the necessary condition for a local minimum yields

$$0 = x_j x_k S_j S_k \partial S_j \partial S_k V.$$

Since $x_j > 0$, the necessary condition for a local minimum reduces to $S_j S_k \partial S_j \partial S_k V = 0$. But, in that case $y_{jk}$ is arbitrary. On the other hand, if $S_j S_k \partial S_j \partial S_k V \neq 0$, we cannot have a local minimiser. Since $S_j$ and $S_k$ are prices of the underlying assets, $S_j, S_k \geq 0$ and we thus have

$$\rho_{jk} = \rho_{jk}(\partial S_j \partial S_k V) = \begin{cases} \rho_{jk}^{+} & \text{if } \partial S_j \partial S_k V > 0, \\ \rho_{jk}^{+} & \text{else} \end{cases}$$
as the minimiser of \( f \) with respect to \( y_{jk} \) (independent of whether \( S_j S_k \partial S_j \partial S_k V = 0 \) or not). Note that the minimising \( \rho_{jk} \) are independent of each other and of \( x_j \) and \( x_k \).

Having found \( \rho_{jk} \) we now consider the reduced problem of finding minimising \( x_1, \ldots, x_n \) for

\[
f(x_1, \ldots, x_d) = \sum_{j=1}^{d} \sum_{k=1}^{d} x_j x_k \rho_{jk}(\partial S_j \partial S_k V) S_j S_k \partial S_j \partial S_k V
\]

on the \( d \)-dimensional rectangle \([\sigma_{-1}^1, \sigma_{+1}^1] \times \cdots \times [\sigma_{-d}^d, \sigma_{+d}^d] =: Q\) which lies inside the first quadrant. Defining a \( d \)-by-\( d \) matrix \( M \) with \( M_{jk} = \rho_{jk}(\partial S_j \partial S_k V) S_j S_k \partial S_j \partial S_k V \) and writing the \( x_j \) as a \( d \)-dimensional (row) vector \( \vec{x} \) we have

\[
(\sigma_1, \ldots, \sigma_d) = \arg\min_{\vec{x} \in Q} \vec{x} \cdot M \cdot \vec{x}^T
\]

where the \( d \)-dimensional rectangle \( Q \) lies inside the first quadrant. Note that the matrix \( M \) is symmetric but not necessarily positive or negative definite. So, there is no general simplification for this minimisation problem.

Hence, we derived the non-linear problem of finding \( V = V(S_1, \ldots, S_d, t) \) such that

\[
\partial_t V = r V - r \sum_{j=1}^{d} S_j \partial S_j V - \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \rho_{jk}(\Gamma_{jk}[V]) \sigma_j(M[V]) \sigma_k(M[V]) S_j S_k \partial S_j \partial S_k V
\]

(2.2a)

\[
\rho_{jk}(x) = \begin{cases} \rho_{jk}^- & \text{if } x > 0, \\ \rho_{jk}^+ & \text{else} \end{cases}
\]

(2.2b)

\[
(\sigma_1, \ldots, \sigma_d)(A) = \arg\min_{\vec{x} \in Q} \vec{x} \cdot A \cdot \vec{x}^T
\]

(2.2c)

\[
M_{jk}[W] = \rho_{jk}(\Gamma_{jk}[W]) S_j S_k \Gamma_{jk}[W], \quad Q = [\sigma_{-1}^1, \sigma_{+1}^1] \times \cdots \times [\sigma_{-d}^d, \sigma_{+d}^d].
\]

(2.2d)

Here, \( \Gamma_{jk}[W] \) denotes the operator acting on \( W \) that returns the second derivative of \( W \) with respect to \( S_j \) and \( S_k \), that is \( \Gamma_{jk}[W] = \partial S_j \partial S_k W \). Again, it is recalled that \( \rho_{jj}^- = \rho_{jj}^+ = 1 \) and that \( \rho_{jk}^\pm = \rho_{kj}^\pm \).

*Remark.* The parameters \( \rho_{jk} \) and \( \sigma_j \) depend on \( V \) in a non-linear fashion. Hence, equation (2.2a) is a non-linear PDE.

*Remark.* For \( d = 1 \), we have \( \rho_{jk} = \rho_{11} = 1 \) and \( \sigma_1 = \arg\min_{\sigma_{-1} \leq x \leq \sigma_{+1}} x^2 S^2 \partial_{SS}^2 V \). Thus, problem (2.2) reduces to

\[
\partial_t V = r V - r S \partial S V - \frac{1}{2} \sigma^2 S^2 \partial_{SS}^2 V
\]

(2.3a)

\[
\sigma = \begin{cases} \sigma^- & \text{if } \partial_{SS}^2 V > 0 \\ \sigma^+ & \text{else} \end{cases}
\]

(2.3b)

This problem is found in the literature, for example in [ALP95, AB99, Wil98, WO98, Top01a, PFV03].
Remark. The condition (2.2c) is a non-trivial generalisation of (2.3b). In particular, the $\sigma_j$ cannot be found independently from each other. Instead, when solving (2.2) numerically, the condition (2.2c) must be applied by solving the $d$-dimensional minimisation problem locally, for example at each grid point for the finite difference method.

Remark. For the special case that $d = 2$ and $\sigma_j^- = \sigma_j^+$ for $j = 1, 2$, problem (2.2) reduces to
\[
\partial_t V = r V - r S_1 \partial_{S_1} V - r S_2 \partial_{S_2} V - \frac{1}{2} \sigma_1^2 S_1^2 \partial_{S_1}^2 V - \frac{1}{2} \sigma_2^2 S_2^2 \partial_{S_2}^2 V - \rho(\Gamma_{12}[V]) \sigma_1 \sigma_2 S_1 S_2 V
\]
\[
\rho(x) = \begin{cases} 
\rho^- & \text{if } x > 0, \\
\rho^+ & \text{else} 
\end{cases}
\]
This problem is discussed for example in [Wil98, Top01a, Top01b]. The more general version of the case $d = 2$ is as follows
\[
\partial_t V = r V - r S_1 \partial_{S_1} V - r S_2 \partial_{S_2} V - \frac{1}{2} \sigma_1(M)^2 S_1^2 \partial_{S_1}^2 V - \frac{1}{2} \sigma_2(M)^2 S_2^2 \partial_{S_2}^2 V - \rho(\Gamma_{12}) \sigma_1 \sigma_2(M) S_1 S_2 V
\]
\[
(\sigma_1, \sigma_2)(M) = \arg\min_{\sigma_1 \in [\sigma_1^-, \sigma_1^+], \sigma_2 \in [\sigma_2^-, \sigma_2^+]} \bar{x} M \bar{x}^T,
\]
\[
M = \begin{pmatrix}
S_1^2 \Gamma_{11} & \rho(\Gamma_{12}) S_1 S_2 \Gamma_{12} \\
\rho(\Gamma_{12}) S_1 S_2 \Gamma_{12} & S_2^2 \Gamma_{22}
\end{pmatrix}
\]
and $\rho(\Gamma_{12})$ defined as above.
Chapter 3

Discretisation in Time and Addressing the Non-Linearity

In this chapter, we discretise the problem (2.2) in time. This is done using finite differences. Further, we address the non-linearity and discuss how one can deal with it when solving the problem numerically.

For the time discretisation, first, the time interval of interest must be fixed. We consider problems where the final condition at some finite time of expiry $T > 0$ is given as a pay-off function. The solution of the problem is searched for at time $t = 0$. The relevant interval $[0, T]$ is partitioned into $L$ subintervals, $[t_{\ell-1}, t_\ell]$, with $0 = t_0 < t_1 < \cdots < t_L = T$, and we define $\delta t_\ell := t_{\ell+1} - t_\ell$. For simplification purposes, we introduce the operator $\mathcal{L}$ as

$$
\mathcal{L}[V] := r V - r \sum_{j=1}^d S_j \partial_j V - \frac{1}{2} \sum_{j=1}^d \sum_{k=1}^d \rho_{jk} (\Gamma_{jk}[V]) \sigma_j (M[V]) \sigma_k (M[V]) S_j S_k \partial_j \partial_k V
$$

where we used $\partial_j$ as a short notation for $\partial_{S_j}$. The general mid-point scheme for the problem (2.2) is then defined as

$$
\frac{V^{\ell+1}_\ell - V^\ell_\ell}{\delta t_\ell} = (1 - \theta) \mathcal{L}[V^{\ell+1}] + \theta \mathcal{L}[V^\ell] \tag{3.1}
$$

where $\theta \in [0, 1]$ is a parameter for the mid-point scheme and $V^\ell$ denotes the approximation of $V$ at time $t_\ell$, i.e. $V^\ell(\bar{S}) \approx V(S_1, \ldots, S_d, t_\ell)$. Note that we solve backwards in time, i.e. $V^{\ell+1}_\ell$ is considered to be known and $V^\ell$ is the function to be solved for. Hence, the case $\theta = 0$ can be identified as the explicit Euler scheme, $\theta = 1$ as the implicit Euler scheme, and $\theta = 1/2$ as the Crank-Nicholson scheme. For the very first time-step, $V^{L+1} = V^L$ approximates the pay-off function. Hence, equation (3.1) requires the approximation of the pay-off to be twice differentiable, unless $\theta = 1$. In most real-world problems, the pay-off
is at most continuous. Also, in [ZVF00], Zvan et al. report that, when not using the fully implicit Euler scheme but, for example, the Crank-Nicholson scheme, errors occur in the numerical solution unless the time step size is chosen very small. This is an effect also widely known in mechanics. Hence, we will restrict ourselves to the implicit Euler scheme in time, that is $\theta = 1$.

In (3.1), the operator $L$ depends non-linearly on its argument. A simple approach to approximate the non-linear problem is to replace $L$ by $L_{V^{\ell+1}}$ where $L_W$ is defined as

$$L_W[V] := r V - r \sum_{j=1}^{d} S_j \partial_j V - \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \rho_{jk} (\Gamma_{jk}[W]) \sigma_j (M[W]) \sigma_k (M[W]) S_j S_k \partial_j \partial_k V. \quad (3.2)$$

As $V^{\ell+1}$ is known, $L_{V^{\ell+1}}$ is a linear operator on $V^\ell$.

Another approach is to introduce the following iteration scheme.

**Algorithm 1.**

1. Set $V^{\ell,0} := V^{\ell+1}$.

2. Given $V^{\ell,\kappa - 1}$, find $V^{\ell,\kappa}$ such that $V^{\ell+1} = V^{\ell,\kappa} + \delta t_{\ell} L_{V^{\ell,\kappa-1}}[V^{\ell,\kappa}]$.

3. If

   - $\|V^{\ell,\kappa - 1} - V^{\ell,\kappa}\|$ is below some given tolerance or
   - $\|\delta t_{\ell} L_{V^{\ell,\kappa}}[V^{\ell,\kappa}] - V^{\ell+1} + V^{\ell,\kappa}\|$ is below some given tolerance or
   - $\kappa$ is equal to some given maximal number of iteration steps,

   set $V^\ell = V^{\ell,\kappa}$ and continue with the next time step. Else, increase $\kappa$ by one and continue with step 2.

Again, the operator $L_{V^{\ell,\kappa-1}}$ is linear when acting on $V^{\ell,\kappa}$. Note that in the second stopping criterion we use the operator $L_{V^{\ell,\kappa}}$, i.e. with $\sigma_j$ and $\rho_{jk}$ evaluated using the new solution $V^{\ell,\kappa}$. By that, we test if the new solution (approximately) satisfies the non-linear equation.

**Remark.** The first approach is a special case of the second approach with the maximal number of iteration chosen to be one.

**Remark.** As part of the stopping criteria for the second approach, the norm of the change in the solution of two successive iteration steps and the norm of the error in the non-linear equation are compared to some tolerance. There are numerous ways to choose these
norms. The choice may depend on the discretisation method (for the asset space) and might influence the convergence analysis of this linearisation method. For the examples given in chapter 8, we always chose the $\ell^2$-norm of the solution vector in the discretised asset-space, i.e. the square root of the sum of the squares of all vector entries.

In the second approach, in each iteration step, the approximation of the non-linear operator $L$ is corrected using the latest solution to the linearised problem. A convergence analysis of this scheme is omitted here. In fact, convergence is not always guaranteed as some examples in chapter 8 showed.

One approach to handle the non-linearity that can be found in the literature is the Newton method, [FV01]. A critical point when applying this method is finding the derivative of the operator $L$. In general, the non-linear term involving $\rho_{jk}$ and $\sigma_j$ is not differentiable at all points. Hence, for the Newton method, one has to give a reasonable substitution for the non-defined derivatives. One way is to regularise the functions for $\rho_{jk}$ and $\sigma_j$. Another suggestion is to use the one sided derivative. Forsyth and Vetzal analysed both alternatives in [FV01] for the special case (2.3) and concluded to prefer the latter method. However, a generalisation of the Newton method to the $d$-dimensional version of the non-linear Black-Scholes equation is non-trivial: the derivative of the function $(\sigma_1, \ldots, \sigma_d)(A) = \arg\min_{\vec{x} \in Q} \vec{x} \cdot A \cdot \vec{x}^T$ with respect to $A$ cannot be given analytically. A numerical approximation must be used instead.

For this work, we chose the approach described in algorithm 1.
Chapter 4

Bounding the Asset-Space

In general, the asset-space under consideration, call it $\Omega$, for which problem 1 from the previous chapter must be solved is unbounded. Typically, we have $\vec{S} \in [0, \infty)^d = [0, \infty) \times \cdots \times [0, \infty)$. Cases when $\Omega$ is bounded, at least in some dimension, are for example the presence of an up-&-out barrier. But even these must not necessarily apply to all asset variables and / or at all times. If $\Omega$ is unbounded, for discretisation in the asset-space, we first have to introduce some artificial cut-off boundary. Further, we have to define a boundary condition at that cut-off boundary. Both, the choice of the cut-off boundary and the choice of the new boundary condition depend on the specific problem that is to be solved.

The introduction of the cut-off boundary and its cut-off condition rely on the fact that the solution to the given problem becomes asymptotic for large values of $\vec{S}$. Hence the boundary should be chosen far away from the area of interest and far away from particularities like an option strike, etc.

Different boundary conditions for the cut-off boundary are suggested and investigated in the literature. In [Top03], a boundary condition of the form $\partial^2_{S_j S_j} V(S_j^{\text{max}}, t) = 0$ is used. By this condition, one assumes a linear behavior of the solution for large $S_j$. In case of, for example, power options, this boundary condition is not applicable. Kangro and Nicolaides suggest in [KN00] to use a non-homogenic Dirichlet boundary condition with the value of pay-off on the cut-off boundary for all times as the value on that boundary, i.e. $V(S_j^{\text{max}}, t) = \text{Payoff}(S_j^{\text{max}})$ for all $t \in [0, T]$. Jackson, S"uili, and Howison use in their example in [JSH98] a slight variation of this approach: they also choose a Dirichlet boundary condition but discount the constant part of the pay-off to include the time-effect. For example, for a vanilla call option with strike $E << S^{\text{max}}$ at expiry $T$, instead of setting $V(S_j^{\text{max}}, t) = S_j^{\text{max}} - E$, as Kangro and Nicolaides suggest, Jackson et al. choose $V(S_j^{\text{max}}, t) = S_j^{\text{max}} - e^{-r(T-t)} E$ where $r$ denotes the risk free rate. Finally, Topper includes in [Top01b] as a possible cut-
off-boundary condition one of Neumann type, that is, fixing the derivative orthogonal to the boundary. To set the value on the Neumann-type cut-off boundary, one can again use the pay-off function. Setting \( \frac{\partial}{\partial \vec{n}} V(\vec{S}^{\text{max}}, t) = \frac{\partial}{\partial \vec{n}} \text{Payoff}(\vec{S}^{\text{max}}) \), where \( \frac{\partial}{\partial \vec{n}} \) denotes the derivative normal to the boundary, omits the question whether to discount the constant part of the pay-off or not.

Figure 4.1 compares the first three methods described above for a European vanilla call option and for a European digital call option that pays one monetary unit at expiry if the asset is above the strike and nothing otherwise. The figures also include the analytic solution and the numerical solution when no boundary condition at all is applied at the cut-off boundary. For the discretisation in the asset space, the finite difference method described in the following chapter is used.

Both figures clearly show that when no boundary condition is applied explicitly (lightly dotted curve), the solution is restricted to a homogeneous Neumann boundary condition at the cut-off boundary, also known as the natural boundary condition. This is a well-known fact. Sometimes, a boundary with no explicit conditions applied to it is also referred to as a reflection boundary.

For the two examples chosen, setting the second derivative to zero (boldly dotted curve) and applying a Dirichlet boundary according to [JSH98] (dashed curve) both approximate the true solution (continuous curve) equally well up to the cut-off boundary. However, applying a Dirichlet boundary with values as suggested in [KN00] (dash-dotted curve) clearly lead to errors near the cut-off boundary, especially for the digital call option where the value of the option at the cut-off boundary remains small compared to the time effect.

Based on this small survey, we restrict ourselves to problems defined on an (if necessary artificially) bounded asset-space \( \Omega = (S_1^{\min}, S_1^{\max}) \times \cdots \times (S_d^{\min}, S_d^{\max}) \) with Dirichlet boundary conditions on all sides. The value on an artificial Dirichlet boundary is chosen in the spirit of [JSH98]. For the numerical examples in chapter 8, comments on this choice are made where necessary.
Figure 4.1: Comparison of different cut-off boundary conditions
Summarising the results from the previous two chapters, we here reformulate the resulting problem for the semi-discretised and linearised uncertain parameter model.

**Problem 1.** Given the asset-space $\Omega = (S_1^{\text{min}}, S_1^{\text{max}}) \times \cdots \times (S_d^{\text{min}}, S_d^{\text{max}})$, the pay-off function $P: \Omega \rightarrow \mathbb{R}$, some function $V_D: \overline{\Omega} \times [0, T] \rightarrow \mathbb{R}$ that defines the Dirichlet condition on $\partial \Omega$, and some partition $0 = t_0 < \cdots < t_\ell < \cdots < t_L = T$ of the interval $[0, T]$, for $\ell = L, L-1, \ldots, 0$, find $V^\ell: \overline{\Omega} \rightarrow \mathbb{R}$ using the following algorithm.

1. Set $V^L = P$ on $\Omega$ and $\ell = L - 1$.
2. Set $V^{\ell, 0} = V^{\ell+1}$ on $\overline{\Omega}$ and $\kappa = 1$.
3. Find $V_0: \overline{\Omega} \rightarrow \mathbb{R}$ such that

   $$V_0 = 0 \quad \text{on } \partial \Omega \quad (5.1a)$$

   $$\left(\text{id} + \delta t_\ell \mathcal{L}_{V^{\ell, \kappa-1}}\right)[V_0] = V^{\ell+1} - \left(\text{id} + \delta t_\ell \mathcal{L}_{V^{\ell, \kappa-1}}\right)[V_D(\cdot, t_\ell)] \quad \text{in } \Omega. \quad (5.1b)$$

   The operator $\text{id}$ is the identity operator, i.e. $\text{id}[V] = V$ and $\mathcal{L}_W[V]$ is defined as in (3.2).

4. Set $V^{\ell, \kappa} = V_0 + V_D$ on $\overline{\Omega}$.
5. If $\|V^{\ell, \kappa-1} - V^{\ell, \kappa}\|$ and $\left\|\left(\text{id} + \delta t_\ell \mathcal{L}_{V^{\ell, \kappa}}\right)[V^{\ell, \kappa}] - V^{\ell+1}\right\|$ are above some given tolerance and $\kappa$ is less than some given maximal number of iteration steps, increase $\kappa$ by one and continue with step 3.
6. Set $V^{\ell} = V^{\ell, \kappa}$. 

Chapter 5

Formulation of the semi-discretised Problem
7. If $\ell$ is greater than zero decrease $\ell$ by one and continue with step 2.

8. Return $(V^L, V^{L-1}, \ldots, V^0)$.

Note that we formulated the algorithm such that (5.1) became a problem with homogeneous Dirichlet conditions.
Chapter 6

Finite Difference Method

This chapter is dedicated to the (asset-) space discretisation of one iteration step of the semi-discretised problem as given in (5.1) using a finite difference method. As finite difference methods are standard numerical procedures for solving PDEs in general and for the Black-Scholes equation in particular, we will only highlight the main steps. Details for the classical Black-Scholes equation may be found in the standard text books, like [Hul99, WDH93].

Recall the operator $L_W$ as defined in (3.2),

$$\mathcal{L}_W[V](\vec{S}) := r V(\vec{S}) - r \sum_{j=1}^{d} S_j \partial_j V(\vec{S}) - \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} \rho_{jk} (\partial_j \partial_k W(\vec{S})) \sigma_j (M[W(\vec{S})]) \sigma_k (M[W(\vec{S})]) S_j S_k \partial_j \partial_k V(\vec{S}),$$

(6.1)

Both, $V$ and $W$, are scalar valued functions on $\Omega := (S_{\text{min}}^1, S_{\text{max}}^1) \times \cdots \times (S_{\text{min}}^d, S_{\text{max}}^d)$.

For the discretisation, we define a grid on $\Omega$ by choosing $P_j + 1$ points in the closed interval $[S_j^{\text{min}}, S_j^{\text{max}}]$, $j = 1, \ldots, d$, namely $S_j^{\text{min}} = S_j^0 < \cdots < S_j^{p_j} < \cdots < S_j^{P_j} = S_j^{\text{max}}$. To simplify notation, we will restrict ourselves to equally spaced partitionings of the intervals. Then, we can define a mesh-size $h_j := (S_j^{\text{max}} - S_j^{\text{min}})/P_j$ and it follows that $S_j^p = S_j^{\text{min}} + p h_j$.

Remark. It is widely known that in general, evenly spaced meshes are not the best choice, not only for the Black-Scholes equation. Many times, in some areas the solution is very smooth and in others large derivatives or oscillations occur. In the smooth areas, a large mesh size is sufficient whereas in areas with large derivatives etc., closer grid points are desired to gain a higher resolution of these features. If these areas are known ahead, it might be possible to choose the grid adequately. This is for example suggested in [Hul99, Izv98]. If the areas are
not known ahead or move in time, an adaptive mesh technique might be useful. Numerous techniques are used in the numerical community. Writing down a finite difference scheme using non-uniform meshes is more a notational than a mathematical challenge. The focus of this work is not on non-uniform meshes or adaptive mesh refinement.

Now, at each grid point \( \vec{S}_p := (s_{\text{min}}^1, \ldots, s_{\text{min}}^d) + (p_1 h_1, \ldots, p_d h_d) \in \Omega \) the value \( L^W[V](\vec{S}_p) \) is approximated by some \( L^\vec{p}W[V] \). This approximation of \( L^W[V](\vec{S}_p) \) is best described when applying the following rules and finite differences to the operator \( L^W[V](\vec{S}_p) \) defined in (6.1).

1. Substitute the factors \( s_j \) and \( s_k \) by \( s_{\text{min}}^j + p_j h_j \) and \( s_{\text{min}}^k + p_k h_k \), respectively.

2. Substitute \( V(\vec{S}_p) \) by \( V(\vec{p}) := V(\vec{s}_{\text{min}} + p_1 h_1, \ldots, \vec{s}_{\text{min}} + p_d h_d) \).

3. Approximate \( \partial_j V(\vec{S}_p) \) using a central difference:

\[
\partial_j V(\vec{S}_p) \approx \frac{V(p_1, \ldots, p_j + 1, \ldots, p_d) - V(p_1, \ldots, p_j - 1, \ldots, p_d)}{2h_j}. \tag{6.2}
\]

4. For \( j \neq k \), approximate \( \partial_j \partial_k V(\vec{S}_p) \) and \( \partial_j \partial_k W(\vec{S}_p) \) using a central difference:

\[
\partial_j \partial_k V(\vec{S}_p) \approx \frac{V(p_1, \ldots, p_j + 1, \ldots, p_k + 1, \ldots, p_d) - \left(V(p_1, \ldots, p_j + 1, \ldots, p_k - 1, \ldots, p_d) + V(p_1, \ldots, p_j - 1, \ldots, p_k + 1, \ldots, p_d)\right)}{4h_j h_k}. \tag{6.3}
\]

For notational purposes (and without loss of generality), we assumed \( j < k \).

5. Approximate \( \partial_j^2 V(\vec{S}_p) \) and \( \partial_j^2 W(\vec{S}_p) \) using a central difference:

\[
\partial_j^2 V(\vec{S}_p) \approx \frac{V(p_1, \ldots, p_j + 1, \ldots, p_d) - 2V(p_1, \ldots, p_j, \ldots, p_d) + V(p_1, \ldots, p_j - 1, \ldots, p_d)}{h_j^2}. \tag{6.4}
\]

Note that we discretised \( L_W \) only for the inner grid points. On the boundary of \( \Omega \), the function \( V_0 \) is forced to zero in (5.1).

Still, for some purposes it is necessary to numerically approximate the derivatives on the boundary of \( \Omega \). For example, financially, the first and second derivative of \( V \) with respect to the assets, often named delta and gamma, respectively, are of particular interest and their numerical approximation might be desired. Hence, for completeness, we give one-sided finite difference approximations of the first and second derivative that can be used along the border. As the finite differences used for the inner grid points are of order two, we chose one-sided differences of that order, too.
The one-sided finite differences used if \( p_j = 0 \) for only one \( j \) are

\[
\partial_j V(\bar{S}_p) \approx \frac{-3V_{(...,0,...)} + 4V_{(...,1,...)} - V_{(...,2,...)}}{2h_j}, \tag{6.5}
\]

\[
\partial_j \partial_k V(\bar{S}_p) \approx \frac{(-3V_{(...,0,...,p_k+1,...)} + 4V_{(...,1,...,p_k+1,...)} - V_{(...,2,...,p_k+1,...)} + 3V_{(...,0,...,p_k-1,...)} - 4V_{(...,1,...,p_k-1,...)} + V_{(...,2,...,p_k-1,...)})}{(4h_j h_k)}, \tag{6.6}
\]

\[
\partial_j^2 V(\bar{S}_p) \approx \frac{2V_{(...,0,...)} - 5V_{(...,1,...)} + 4V_{(...,2,...)} - V_{(...,3,...)}}{h_j^2}. \tag{6.7}
\]

The one-sided finite difference of the mixed second derivative used if \( p_j = p_k = 0 \) is

\[
\partial_j \partial_k V(\bar{S}_p) \approx (5V_{(...,0,...,0,...)} - 4V_{(...,1,...,0,...)} - 4V_{(...,0,...,1,...)} - V_{(...,2,...,0,...)} - 3V_{(...,2,...,2,...)} + 4V_{(...,2,...,1,...)} + 4V_{(...,1,...,2,...)})/(4h_j h_k). \tag{6.8}
\]

The necessary changes for the one-sided finite differences used if \( p_j = p_j + 1 \) etc. are obvious.

### 6.1 Remark on the Case \( d > 1 \)

Recall that for the case \( d = 2 \), the operator \( \mathcal{L}_W \) simplifies as follows (cf. (2.5)).

\[
\mathcal{L}_W[V](\bar{S}) = rV - rS_1 \partial_1 V - rS_2 \partial_2 V - \frac{1}{2}(\sigma_W)^2 S_1^2 \partial_1^2 V - \frac{1}{2}(\sigma_W)^2 S_2^2 \partial_2^2 V - \rho_W(\sigma_W)1(\sigma_W)2S_1S_2\partial_1\partial_2 V
\]

with

\[
\tilde{\sigma}_W = \arg\min_{\bar{x} \in Q} \bar{x} \begin{pmatrix} S_1^2 \partial_1^2 W & \rho_W S_1 S_2 \partial_1 \partial_2 W \\ \rho_W S_1 S_2 \partial_1 \partial_2 W & S_2^2 \partial_2^2 W \end{pmatrix} \bar{x}^T, \quad Q = [\sigma_1^-, \sigma_1^+] \times [\sigma_2^-, \sigma_2^+],
\]

and

\[
\rho_W = \begin{cases} 
\rho^- & \text{if } \partial_1 \partial_2 W > 0, \\
\rho^+ & \text{else} 
\end{cases}
\]

The discretised versions of the \( W \)-dependent parameters, \( \rho_{W}^{p,q} \) and \( \tilde{\sigma}_{W}^{p,q} \), are defined as

\[
\rho_{W}^{p,q} = \begin{cases} 
\rho^- & \text{if } (\Gamma_{W}^{p,q})_{12} > 0, \\
\rho^+ & \text{else} 
\end{cases},
\]

and

\[
\tilde{\sigma}_{W}^{p,q} = \arg\min_{\bar{x} \in Q} \bar{x} \begin{pmatrix} (p h_p)^2 (\Gamma_{W}^{p,q})_{11} & \rho_{W}^{p,q} p h_p q h_q (\Gamma_{W}^{p,q})_{12} \\ (p h_p)^2 (\Gamma_{W}^{p,q})_{11} & (q h_q)^2 (\Gamma_{W}^{p,q})_{22} \end{pmatrix} \bar{x}^T
\]

17
where $\Gamma^{p,q}_W$ denotes the discrete second derivative of $W$ at $\vec{S}^{p,q} = (S_{1,\min}, S_{2,\min}) + (p h_p, q h_q)$.

To determine $\vec{\sigma}^{p,q}_W$, at each grid point, $\vec{S}^{p,q}$, we have to solve a minimisation problem of the form

$$(y_1, y_2) = \arg\min_{\vec{x} \in Q} (x_1, x_2) \begin{pmatrix} a & b \\ b & c \end{pmatrix} (x_1, x_2)^T$$

where $Q$ is a rectangle lying completely inside the first quadrant. It is easy to see that a minimiser may always be found on the boundary of $Q$. So, we look for the minimiser on each of the four sides (this can be done analytically by solving a problem of the form $y = \arg\min_{x \in [l,u]} (\alpha x^2 + \beta x)$) and choose the total minimiser from these four solutions. This method can be generalised for higher dimensional problems by some recursive algorithm that solves the $d$-dimensional minimisation problem by solving $(d - 1)$-dimensional minimisation problems on all sides of $Q$. 

18
Chapter 7

Finite Element Method

This chapter is devoted to the finite element discretisation. It is not as common as the finite difference method and by far not so widely used for numerically solving the Black-Scholes equation. Therefore, we will provide a much more detailed overview of the derivation of the scheme. As a thorough introduction to the theory of finite elements we suggest [BS94, Bra97].

The three main steps of the finite element method are

1. Finding a suitable weak formulation, also known as a variational form, of the differential equation. This includes deciding on the function spaces in which the variational equation is viewed. These function spaces are Sobolev spaces.

2. Subdividing the area \( \Omega \) using a regular triangulation. The term "triangulation" is quite common in the literature even when the geometric shapes used for the subdivision of \( \Omega \) are not triangles but for example quadrilaterals.

3. Approximating the infinite dimensional function spaces by finite dimensional (sub-) spaces, for example by using piecewise polynomials with respect to the chosen triangulation. The variational equation from the first step together with the finite dimensional spaces defined in this step then give a discrete problem. Initial and boundary conditions of the problem are projected onto the finite dimensional space in the discrete formulation of the problem.

The first section of this chapter will briefly summarise the main definitions and results from the theory of Sobolev spaces. Details and proofs can be found in mathematical text books like [BS94] and [Bra97]. In fact, most definitions and results given here are taken from [BS94]. This section will also provide a formal definition of a regular triangulation.
With the mathematical setting provided in the first section, the second section then focuses on finding a weak formulation of the linearised uncertain parameter model as given in (5.1). A subsection discusses a second weak form one might be tempted to use.

The third section of this chapter is devoted to finding the finite dimensional spaces. These finite dimensional spaces are characterised by the type of finite elements used in the discretisation. For conform finite elements, the finite dimensional space is chosen as a subspace of the function space chosen in the first step. Several choices are presented and discussed. They will be compared in chapter 8 for some numerical examples.

We close this chapter with some explanatory remarks on the implementation of the finite element method.

### 7.1 Introduction to Sobolev Spaces

We start off with some basic concepts to agree on a notation. In the following, $\Omega$ will always denote a Lebesgue-measurable (open or closed) subset of $\mathbb{R}^d$ with non-empty interior, which we will also call a "domain".

**Definition 1.** For an open domain $\Omega \subset \mathbb{R}^d$, let $C^k(\Omega; \mathbb{R}^n)$, $0 \leq k < \infty$ denote the set of all functions $f: \Omega \to \mathbb{R}^n$ that are continuously differentiable up to the order $k$. Further, $C^\infty(\Omega; \mathbb{R}^n)$ denotes the set of all arbitrarily often differentiable functions. If the context allows, we will also write $f \in C^k(\Omega)$ or say "$f$ is $C^k$" etc. Further, $C^k_0(\Omega; \mathbb{R}^n)$, $0 \leq k \leq \infty$, denotes the subset of $C^k(\Omega; \mathbb{R}^n)$ with the additional condition that any $f \in C^k_0(\Omega; \mathbb{R}^n)$ vanishes on the boundary of $\Omega$. The set $C^\infty_0(\Omega; \mathbb{R}^n)$ is also denoted by $\mathcal{D}(\Omega)$ and often called the set of test functions on $\Omega$.

**Definition 2 (Lebesgue norm, Lebesgue space).** For $1 \leq p < \infty$, the Lebesgue norm of a Lebesgue-measurable function $f$ is defined as

$$\|f\|_{L^p(\Omega)} := \left(\int_{\Omega} |f(x)|^p \, dx\right)^{1/p}$$

and for $p = \infty$, its definition is

$$\|f\|_{L^\infty(\Omega)} := \text{ess sup}\{ |f(x)| : x \in \Omega \}.$$

For $1 \leq p \leq \infty$, the Lebesgue spaces are then defined as

$$L^p(\Omega) := \{ f : \|f\|_{L^p(\Omega)} < \infty \}.$$
As usual, all functions $f$ and $g$ satisfying $\|f - g\|_{L^p(\Omega)} = 0$ are identified with each other and we construct equivalence classes of functions with respect to that identification.

**Definition 3 (Locally integrable functions).** The set of locally integrable functions on $\Omega$ is denoted by

$$L^1_{\text{loc}}(\Omega) := \{ f : f \in L^1(K) \text{ for all compact } K \subset \text{int } \Omega \}$$

where $\text{int } \Omega$ denotes the interior of $\Omega$.

**Definition 4 (Multi-index).** For a vector $\alpha \in \mathbb{N}_0^d$ used as a multi-index, its length is defined as

$$|\alpha| := \sum_{j=1}^d \alpha_j$$

and for some $\phi \in C^\infty$ we will write $D^\alpha \phi$ or $(\frac{\partial}{\partial x_1})^{\alpha_1} \cdots (\frac{\partial}{\partial x_d})^{\alpha_d} \phi$. The order of the derivative is $|\alpha|$.

**Definition 5 (Weak derivative).** A function $f \in L^1_{\text{loc}}(\Omega)$ is said to have a weak derivative, $D^\alpha_w f$, if there exists a function $g \in L^1_{\text{loc}}(\Omega)$ such that for all $\phi \in D(\Omega) = C_0^\infty(\Omega)$ we have

$$\int_{\Omega} g(x) \phi(x) \, dx = (-1)^{|\alpha|} \int_{\Omega} f(x) D^\alpha \phi(x) \, dx.$$ 

In this case we define $D^\alpha_w f := g$.

For example, the first (weak) derivative of $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto |x|$ is

$$D^1_w f(x) = \begin{cases} -1 & \text{if } x < 0, \\ 1 & \text{if } x > 0. \end{cases}$$

The value of $D^1_w f(0)$ is arbitrary. However, $f$ does not have a second (or higher order) weak derivative.

Note that for $f \in C^{[\alpha]}(\Omega)$ we have $D^\alpha_w f = D^\alpha f$, i.e. the weak derivative is identical to the classical derivative. Hence, we later will drop the subscript $w$, remarking that the derivative might be understood in the weak sense, when necessary.

The Sobolev norms and Sobolev spaces are generalisations of the Lebesgue norms and spaces including (weak) derivatives:

**Definition 6 (Sobolev norm, Sobolev semi-norm, Sobolev space).** Let $k$ be a non-negative integer, and let $f \in L^1_{\text{loc}}(\Omega)$ with the property that the weak derivatives $D^\alpha_w f$ exist for all $|\alpha| \leq k$. Then, for $1 \leq p < \infty$ define the Sobolev norm

$$\|f\|_{W^k_p(\Omega)} := \left( \sum_{|\alpha| \leq k} \|D^\alpha_w f\|_{L^p(\Omega)}^p \right)^{1/p}$$
and for \( p = \infty \)

\[
\|f\|_{W^k_p(\Omega)} := \max_{|\alpha| \leq k} \|D^\alpha f\|_{L^\infty(\Omega)}.
\]

The Sobolev semi-norms are defined as

\[
1 \leq p < \infty : \quad |f|_{W^k_p(\Omega)} := \left( \sum_{|\alpha| = k} \|D^\alpha f\|^p_{L^p(\Omega)} \right)^{1/p}
\]

and

\[
|f|_{W^k_{\infty}(\Omega)} := \max_{|\alpha| = k} \|D^\alpha f\|_{L^\infty(\Omega)}.
\]

Further, for \( 1 \leq p \leq \infty \), the Sobolev spaces are defined as

\[
W^k_p(\Omega) := \{ f \in L^1_{\text{loc}}(\Omega) : \|f\|_{W^k_p(\Omega)} < \infty \}.
\]

The space \( W^k_p(\Omega) \) can also be viewed as the completion of \( C^\infty(\Omega) \) with respect to the corresponding Sobolev-norm:

**Theorem 1.** For any open domain \( \Omega \) the space \( C^\infty(\Omega) \cap W^k_p(\Omega) \) lies dense in \( W^k_p(\Omega) \).

This theorem was originally proven in [MS64].

**Definition 7 (Lipschitz domain, Lipschitz boundary).** An open domain \( \Omega \) is called a Lipschitz domain and its boundary a Lipschitz boundary if it is connected and if for each point on its boundary, \( x \in \partial \Omega := \overline{\Omega} \setminus \Omega \), there exists a coordinate transformation \( \Phi: \mathbb{R}^d \to \mathbb{R}^d \), a \( \delta > 0 \), and a Lipschitz-continuous function \( \eta: [-\delta, +\delta]^{d-1} \to \mathbb{R} \) such that

\[
\Omega \cap B(x, \delta) = \{ \Phi(y_1, \ldots, y_d) \in B(x, \delta) : \eta(y_1, \ldots, y_{d-1}) > y_d \}
\]

\[
\partial \Omega \cap B(x, \delta) = \{ \Phi(y_1, \ldots, y_d) \in B(x, \delta) : \eta(y_1, \ldots, y_{d-1}) = y_d \}
\]

\[
B(x, \delta) \setminus \overline{\Omega} = \{ \Phi(y_1, \ldots, y_d) \in B(x, \delta) : \eta(y_1, \ldots, y_{d-1}) < y_d \}
\]

where \( B(x, \delta) \) denotes the open ball of radius \( \delta \), centered at \( x \). A closed set is called a (closed) Lipschitz domain if it is the closure of an (open) Lipschitz domain.

Loosely speaking, a domain is a Lipschitz domain, if locally its boundary can be represented as the graph of a Lipschitz function and if the domain lies locally on one side of the boundary.

The following two results imply a relationship between the Sobolev spaces and the spaces \( C^m(\Omega) \).
**Theorem 2 (Sobolev’s Inequality).** Let \( \Omega \subset \mathbb{R}^d \) be a Lipschitz domain, let \( k \) be a positive integer, and let \( 1 \leq p < \infty \). Further, let \( d, k, \) and \( p \) fulfil the condition

\[
    k \geq d \quad \text{when} \quad p = 1, \\
    k > d/p \quad \text{when} \quad p > 1.
\]

Then there is a constant \( C \) such that for all \( u \in W^k_p(\Omega) \)

\[
    \|u\|_{L^\infty(\Omega)} \leq C \|u\|_{W^k_p(\Omega)}.
\]

Moreover, there is a continuous function in the \( L^\infty(\Omega) \) equivalence class of \( u \). Hence, for \( d, k, \) and \( p \) as specified above, we can say that \( W^k_p(\Omega) \subset C(\Omega) \).

A proof may be found, for example, in [BS94].

Applying Sobolev’s inequality to the derivatives of \( u \) leads to the following corollary.

**Corollary 1.** Let \( \Omega \subset \mathbb{R}^d \) be a Lipschitz domain, let \( k \) and \( m \) be a positive integers satisfying \( m < k \), and let \( 1 \leq p < \infty \). Further, let \( d, k, n, \) and \( p \) fulfil the condition

\[
    k - m \geq d \quad \text{when} \quad p = 1, \\
    k - m > d/p \quad \text{when} \quad p > 1.
\]

Then, there is a constant \( C \) such that for all \( u \in W^k_p(\Omega) \)

\[
    \|u\|_{W^m_\infty(\Omega)} \leq C \|u\|_{W^k_p(\Omega)}.
\]

Moreover, there is a \( C^m \) function in the \( L^p(\Omega) \) equivalence class of \( u \). Hence, for \( d, k, m, \) and \( p \) as specified above, we can say that \( W^k_p(\Omega) \subset C^m(\Omega) \).

We will have a closer look at the special case \( p = 2 \) as this will lead to the relevant function spaces for the weak formulation. Historically, these Sobolev spaces are often denoted by \( H^k(\Omega) \), that is \( H^k(\Omega) = W^k_2(\Omega) \). The bilinear form

\[
    (u, v)_k := \sum_{|\alpha| \leq k} \int_{\Omega} D^\alpha_u u D^\alpha v \, dx
\]

defines an inner product on \( H^k(\Omega) \). Note that \( \|u\|_{H^k(\Omega)} := \|u\|_{W^k_2(\Omega)} = \sqrt{(u, v)_k} \). In fact, the normed linear space \((H^k(\Omega), \|\cdot\|_{H^k(\Omega)})\) is complete and thus \((H^k(\Omega), (\cdot, \cdot)_k)\) is a Hilbert space. (In fact, the letter \( H \) was chosen in dedication to Hilbert.)
Finally, we draw some particular results from Sobolev’s inequality and its corollary for the special case $p = 2$:

For $d = 1$ we have $H^{m+1} \subset C^m$,
for $d = 2, 3$ we have $H^{m+2} \subset C^m$.

However, for piecewise polynomials, that will serve as functions in our discrete subspace, we have a much stronger result. For this, we first need the formal definition of a regular triangulation.

**Definition 8 (Regular triangulation).** The finite collection $\mathcal{T} = \{T_1, \ldots, T_M\}$ of open polygons $T_j$ is called a triangulation of the domain $\Omega \subset \mathbb{R}^d$ if $\bigcup T_j = \Omega$ and if $T_j \cap T_k = \emptyset$ whenever $j \neq k$. The triangulation is called regular if the following two conditions are satisfied.

1. If $T_j \cap T_k$ consists of a single point $X$ then $X$ is a vertex to both, $T_j$ and $T_k$.
2. If $\overline{T_j} \cap \overline{T_k}$ consists of more than one point then $T_j \cap T_k$ is a side for both polygons, $T_j$ and $T_k$.

In particular, a regular triangulation must not have so called hanging nodes. Figure 7.1 shows an example of a regular triangulation (left) and one that is not regular due to a hanging node. For $d = 1$, the set $\mathcal{T}$ consists of non-overlapping intervals. For $d = 2$, the $T_j$ typically are triangles or quadrilaterals. For $d = 3$, $\mathcal{T}$ typically consists of tetrahedrons and / or rectangular parallelepipeds.

![Figure 7.1: Example of a regular and a non-regular triangulation](image)
We can now write down a proposition we will use when choosing conform finite elements to a given Sobolev space. This result and its proof are taken from [Bra97, Kapital II, Satz 5.2].

**Proposition 1.** Let $\Omega$ be an open and bounded domain with a regular triangulation $T$. Further, let $v: \overline{\Omega} \to \mathbb{R}$ be a function with the property that for each $T_j \in T$ we have $f|_{T_j} \in C^\infty(\text{int}(T_j))$. The space of all such functions will be denoted as $C^\infty(T)$. Then, any such function belongs to $H^k(\Omega)$, $k \geq 1$, if and only if $f \in C^{k-1}(\Omega)$. In other words,

$$\Omega \text{ bounded, } T \text{ a regular triangulation of } \Omega, \text{ and } k \geq 1 \implies H^k(\Omega) \cap C^\infty(T) = C^{k-1}(\Omega) \cap C^\infty(T).$$

**Proof.** It is sufficient to look at $k = 1$ as for $k > 1$ the proposition can be derived by looking at derivatives up to order $k - 1$. Further, to simplify notation, we restrict ourselves to the case $d = 2$.

(1) Let $v \in C(\Omega) \cap C^\infty(T)$. We show that $v \in H^1(\Omega)$. For $j = 1, 2$, define $w_j: \Omega \to \mathbb{R}$ as $w_j(x) = \partial_j v(x)$ in the interior of each $T \in T$ and arbitrary in the remaining null-set. Then for $\phi \in \mathcal{D}(\Omega) = C^\infty_0(\Omega)$ we have

$$\int_\Omega \phi w_j \, dx \, dy = \sum_{T \in T} \int_T \phi \partial_j v \, dx \, dy,$$

where $\partial T$ denotes the boundary of $T$ and $n_j$ is the $j$-th component of the outer normal vector to $T$. Since $v$ is continuous by assumption, the integrals across the inner sides of the $T \in T$ cancel each other. Further, $\phi$ vanishes on the boundary of $\Omega$. Hence, $\int_\Omega \phi w_j \, dx \, dy = -\int_\Omega \partial_j \phi v \, dx \, dy$, that is, $w_j$ is the weak derivative of $v$ and thus $v \in H^1(\Omega)$.

(2) Let $v \in H^1(\Omega) \cap C^\infty(T)$. We show that $v \in C(\Omega)$. As $v$ is chosen to be continuous in the interior of each $T \in T$, all we have to show is that $v$ is continuous across the inner sides of the $T$. (Note that, as $T$ is assumed to be regular, each side of some $T$ falls either on the boundary of $\Omega$ or the side is shared completely with some other $T \in T$.) In the following we focus on $v$ in the neighborhood of an arbitrary inner side and assume a coordinate system such that this side falls on the $y$-axis. Then the side contains the interval $[y_1 - \delta, y_2 + \delta]$ with $y_1 < y_2$ and $\delta > 0$ and we define the function

$$\Psi(x) := \int_{y_1}^{y_2} v(x, y) \, dy.$$
If \( v \in C^\infty(\Omega) \cap H^1(\Omega) \) we have
\[
|\Psi(x_1) - \Psi(x_2)|^2 = |\int_{x_1}^{x_2} \int_{y_1}^{y_2} \partial_1 v \, dx \, dy|^2 \\
\leq \left| \int_{x_1}^{x_2} \int_{y_1}^{y_2} 1 \, dx \, dy \right| \cdot |v|_{W^1_2(\Omega)}^2 \\
\leq |x_1 - x_2| \cdot |y_1 - y_2| \cdot |v|_{W^1_2(\Omega)}^2
\]
where we applied the fundamental theorem of calculus, Schwarz’ inequality, and the trivial estimate \(|ab| \leq |a|, |b|\), in that order. As \( C^\infty(\Omega) \cap H^1(\Omega) \) lies dense in \( H^1(\Omega) \), this inequality holds for our initially chosen \( v \), too. Hence \( \Psi \) is continuous in \( x \), and in particular at \( x = 0 \), as \( |v|^2_{W^1_2(\Omega)} \) is bounded for \( v \in H^1(\Omega) \). Since \( y_1 < y_2 \) where chosen arbitrary, it follows that \( v \) is continuous across the whole inner side.

7.2 Deriving a Weak Formulation

Given the mathematical setting provided in the previous section, we can now derive a weak formulation of one iteration-step of the linearised and time-discretised uncertain parameter model as given in (5.1).

Introducing the functions \( A: \Omega \to \mathbb{R}^{d \times d}, \vec{b}: \Omega \to \mathbb{R}^d \) and \( c: \Omega \to \mathbb{R} \) defined via
\[
A_{jk}(\vec{S}) = \frac{1}{2} \delta_{lr} \rho_{jk}(\Gamma_{jk}[W]) \sigma_j(M[W]) \sigma_k(M[W]) S_j S_k, \\
b_j(\vec{S}) = \delta_{lr} S_j, \\
c(\vec{S}) = 1 + \delta_{lr},
\]
our problem in one iteration step is of the following form.

\textbf{Problem 2 (Strong form).} Given \( f: \Omega \to \mathbb{R} \), find \( u: \Omega \to \mathbb{R} \) such that
\[
- \sum_{j=1}^{d} \sum_{k=1}^{d} A_{jk} \partial_j \partial_k u - \sum_{j=1}^{d} b_j \partial_j u + cu = f \quad \text{in } \Omega. \tag{7.1}
\]

To be able to formulate (7.1), \( u \) has to be twice differentiable.

To derive a weak formulation, we define a bilinear form \( a(\cdot, \cdot): U \times V \to \mathbb{R} \) and a linear functional \( F: V \to \mathbb{R} \) as
\[
a(u, v) := - \sum_{j=1}^{d} \sum_{k=1}^{d} \int_{\Omega} A_{jk} \partial_j \partial_k u v \, d\vec{S} - \sum_{j=1}^{d} \int_{\Omega} b_j \partial_j uv \, d\vec{S} + \int_{\Omega} cuv \, d\vec{S}, \tag{7.2a}

F(v) := \int_{\Omega} fv \, d\vec{S}, \tag{7.2b}
\]
where \((U, (\cdot, \cdot)_U)\) and \((V, (\cdot, \cdot)_V)\) are Hilbert spaces on \(\Omega\). It is obvious how the equation
\[
a(u, v) = F(v) \quad \forall v \in V
\]
relates to (7.1): if \(\hat{u}\) satisfies (7.1) and \(\hat{u} \in U\), then \(\hat{u}\) satisfies (7.3) (independent of \(V\)); if \(\hat{u}\) satisfies (7.3) and \(\hat{u} \in C(\Omega) \cap C^2(\Omega)\) then \(\hat{u}\) satisfies (7.1) if \(C^\infty(\Omega) \subset V\).

The choice of the spaces \(U\) and \(V\) affect the existence and uniqueness of the solution to (7.3). One commonly used theorem to analyse existence and uniqueness when \(U = V\) is the Lax-Milgram lemma.

**Theorem 3 (Lax-Milgram lemma).** Given a Hilbert space \((V, (\cdot, \cdot))\), a continuous linear functional \(F \in V'\) (\(V'\) denotes the dual space to \(V\)), and a continuous bilinear form \(a(\cdot, \cdot)\) on \(V\) with the additional property
\[
a(\cdot, \cdot) \text{ is coercive, i.e. } \exists c > 0: \forall v \in V: a(v, v) \geq c \|v\|_V^2,
\]
there exists a unique \(u \in V\) such that
\[
a(u, v) = F(v) \quad \forall v \in V.
\]

In the more general case where \(U\) and \(V\) may be different, we can define a linear operator \(L: U \to V'\) via \(L(u)(v) := a(u, v)\) and (7.3) can be read as \(L(u) = f\) in \(V'\). Then the following theorem taken from [Bra97, Kapitel III, Satz 3.6] characterises existence and uniqueness of a solution to (7.3).

**Theorem 4.** Let \(U\) and \(V\) be Hilbert spaces and \(a(\cdot, \cdot)\) be a bilinear form on \(U \times V\). The linear operator \(L: U \to V'\) defined via \(L(u)(v) := a(u, v)\) is an isomorphism, that is, a bijective linear mapping such that \(L\) and \(L^{-1}\) are continuous, if and only if the bilinear form \(a(\cdot, \cdot)\) satisfies the following three conditions.

1. \(a(\cdot, \cdot)\) is continuous.
2. There exists some \(c > 0\) such that \(\inf_{u \in U} \sup_{v \in V} \frac{a(u, v)}{\|u\|_U \|v\|_V} \geq c\).
3. For each \(v \in V\), \(v \neq 0\), there is some \(u \in U\) such that \(a(u, v) \neq 0\).

The second condition is often referred to as the inf-sup-condition.

To formulate our weak problem, we chose \(U = V = H^2_0(\Omega) := \{v \in H^2(\Omega): v = 0\text{ on } \partial \Omega\}\). The choice of \(U\) is obvious: it uses the largest Sobolev space \(W^{k}_2(\Omega)\) such that
\(a(\cdot, \cdot)\) is well defined and adds the homogeneous Dirichlet condition. The derivatives of \(u\) are to be understood in the weak sense. The test space \(V\) was then chosen to match the solution space \(U\) so that \(U\) and \(V\) can be discretised in the same manner. Note that \(H^2_0(\Omega)\) is a subspace of \(H^2(\Omega)\) and again a Hilbert space.

We do not address the question whether this choice assures existence and / or uniqueness. It is easy to see that \(a(\cdot, \cdot)\) is continuous as required for both theorems. The main task is to show that \(a(\cdot, \cdot)\) is coercive or satisfies the inf-sup-condition. In fact, it is not clear to us, whether this is true for arbitrary bounded \(\Omega\) and / or parameters in the assumed limits. However, the results in chapter 8 will justify the choice of \(U\) and \(V\) for the given examples.

We can now state the weak form of problem 2.

**Problem 3 (Weak form).** Given \(a: H^2(\Omega) \times H^2(\Omega) \rightarrow \mathbb{R}\) and \(F: H^2(\Omega) \rightarrow \mathbb{R}\) as defined in (7.2), find \(u \in H^2_0(\Omega)\) such that

\[
a(u, v) = F(v) \quad \forall v \in H^2_0(\Omega).
\]

(7.4)

For (7.4) to be well-defined it suffices to have \(A_{jk}, b_j, c \in L^\infty(\Omega)\) and \(f \in L^2(\Omega)\).

**Remark.** Note that for the problems arising from one iteration step in the time-discretised and linearised uncertain parameter model as given in (5.1), \(a(\cdot, \cdot)\) and \(F\) specify to

\[
a(V^{\ell, \kappa}, W) = \int_{\Omega} (V^{\ell, \kappa} W + \delta t \mathcal{L}_{V^{\ell, \kappa}} W) d\vec{S},
\]

\[
F(W) = \int_{\Omega} V^{\ell+1} W d\vec{S} - a(V_D, W).
\]
7.2.1 Reduced Regularity Assumptions

We assume for a moment that $\sigma_j^- = \sigma_j^+$ and $\rho_{jk}^- = \rho_{jk}^+$ for all $j, k = 1, \ldots, d$, that is, all parameters are fixed, and take another look at our bilinear form $a(u, v)$:

$$
a(u, v) = -\sum_{j=1}^d \sum_{k=1}^d \int_{\Omega} A_{jk} \partial_j u \partial_k v \, d\bar{S} - \sum_{j=1}^d \int_{\Omega} b_j \partial_j u v \, d\bar{S} + \int_{\Omega} cuv \, d\bar{S}
= -\sum_{j=1}^d \sum_{k=1}^d \int_{\Omega} A_{jk} \partial_j u \partial_k v \, d\bar{S} - \sum_{j=1}^d \int_{\Omega} (b_j - \sum_{k=1}^d \partial_k A_{jk}) \partial_j u v \, d\bar{S} + \int_{\Omega} cuv \, d\bar{S}
= \sum_{j=1}^d \sum_{k=1}^d \left[ \int_{\Omega} A_{jk} \partial_j u \partial_k v \, d\bar{S} + \int_{\partial\Omega} (A_{jk} \partial_j u) n v \, dS \right] - \sum_{j=1}^d \int_{\Omega} (b_j - \sum_{k=1}^d \partial_k A_{jk}) \partial_j u v \, d\bar{S} + \int_{\Omega} cuv \, d\bar{S}

$$

assuming $v = 0$ on $\partial\Omega$. As $\rho_{jk}$ and $\sigma_j$ are constant here, for the term $\partial_k A_{jk}$ in the second sum we have

$$
\partial_k A_{jk} = \frac{1}{2} \delta t \rho_{jk} \sigma_j \sigma_k S_j \quad \text{if } j \neq k,
\quad \text{and} \quad \partial_k A_{kk} = \delta t \sigma_k^2 S_k.
$$

Hence, if we define

$$
\hat{b}_j(S) = \delta t (r - \sigma_j^2) S_j - \sum_{k \neq j} \frac{1}{2} \delta t \rho_{jk} \sigma_j \sigma_k S_j
$$

and a second bilinear operator $\hat{a}(\cdot, \cdot)$ as

$$
\hat{a}(u, v) = \sum_{j=1}^d \sum_{k=1}^d \int_{\Omega} A_{jk} \partial_j u \partial_k v \, d\bar{S} - \sum_{j=1}^d \int_{\Omega} \hat{b}_j \partial_j u v \, d\bar{S} + \int_{\Omega} cuv \, d\bar{S} \quad (7.5)
$$

then $a(u, v) = \hat{a}(u, v)$ for all $(u, v) \in H^2_0(\Omega) \times H^2_0(\Omega)$. However, $\hat{a}(\cdot, \cdot)$ is also well defined on the larger space $H^1_0(\Omega) \times H^1_0(\Omega)$.

Hence, when assuming fixed parameters, a second weak form of problem 2 is the following problem.

**Problem 4 (Second weak form).** Given $\hat{a}: H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$ as defined in (7.5) and $F: H^1(\Omega) \to \mathbb{R}$ as defined in (7.2b), find $u \in H^1_0(\Omega)$ such that

$$
\hat{a}(u, v) = F(v) \quad \forall v \in H^1_0(\Omega). \quad (7.6)
$$
Note the lower regularity assumption on the solution $u$. Also, as we will see in section 7.3, it is much easier to construct a finite dimensional sub-space to $H^1_0(\Omega)$ than to $H^2_0(\Omega)$. Hence, one might be tempted to use problem 4 as a start off and formulate a weak form for the uncertain parameter model by replacing the fixed $\rho_{jk}$ and $\sigma_j$ in the definition of $A_{jk}$ and $\hat{b}_j$ by the appropriate functions $\rho_{jk}(\Gamma[W])$ and $\sigma_j(M[W])$. However, this ignores the fact that in the definition of $\hat{b}_j$ we used derivatives of $A_{jk} = \frac{1}{2} \delta t \rho_{jk} \sigma_j \sigma_k S_j S_k$. These derivatives are not defined, not even in a weak sense, for the uncertain parameter model, as the functions $\rho_{jk}$ and $\sigma_j$ contain jumps. In chapter 8 we will show some examples of what happens, if this is ignored.

### 7.3 Defining the Finite Dimensional Space

In this section we address the task of discretising a weak problem of the form:

Given a bilinear form $a: H^m(\Omega) \times H^m(\Omega) \to \mathbb{R}$ and a linear form $F: H^m(\Omega) \to \mathbb{R}$, find $u \in H^m_0(\Omega)$ such that

$$a(u, v) = F(v) \quad \forall v \in H^m_0(\Omega).$$

After discussing the general steps common to any problem of this form, we then will turn our attention to the two weak problems specified in the previous section.

First, we choose a regular triangulation $T$ of the domain $\Omega$, that is, a finite collection of non-overlapping subsets $T \subset \Omega$ with the properties given in definition 8. Next, we construct a function space $S^m(T)$ that consists of functions that are polynomial on each $T \in T$. Further, we require that $S^m(T) \subset H^m(\Omega)$. The functions in $S^m(T)$ are called finite elements. (For a formal definition of finite elements we refer to standard text books, [BS94].) The latter condition yields conform finite elements. For non-conform finite element methods, $S^m(T)$ does not have to be a subspace of $H^m(\Omega)$. Convergence analysis for the two methods differ substantially and further requirements are necessary for non-conforming finite elements to guarantee convergence. We do not address any convergences analysis issues, here, but refer to standard text books like [BS94, Bra97] for general techniques.

As the functions in $S^m(T)$ are arbitrarily often differentiable on each $T \in T$ (they are polynomials on each $T$), that is $S^m(T) \subset C^\infty(T)$, proposition 1 yields that the additional condition $S^m(T) \subset H^m(\Omega)$ is equivalent to additionally requiring $S^m(T) \subset C^{m-1}(\Omega)$. Hence,

$$S^m(T) \text{ consists of } T\text{-piecewise polynomials that are globally } C^{m-1}. $$
This latter condition is what makes it difficult to construct finite element spaces for \( m > 1 \). Conversely, these difficulties are the reason for the interest in non-conform finite elements where the condition \( S^m(T) \subset C^{m-1}(\Omega) \) may be violated.

The class of polynomials is generally identical on each element. Their degree depends on the derivatives involved in \( a(\cdot, \cdot) \). For example, for a problem that contains derivatives up to order two, piecewise quadratic functions should be used, at least. Using higher order polynomials might increase the approximation quality but also increases the degrees of freedom on each element and results in a less sparse matrix for the final linear system. Instead, to improve approximation, often, a finer triangulation is chosen.

Having chosen \( S^m(T) \), the discretised problem then is:

Find \( u_S \in S^m_0(T) := \{ u \in S^m(T) : u = 0 \text{ on } \partial \Omega \} \) such that

\[
a(u_S, v_S) = F(v_S) \quad \forall v_S \in S^m_0(T).
\]

As \( S^m_0(T) \) is finite dimensional, there is a basis \( \{ \eta_1, \ldots, \eta_M \} \) that spans \( S^m_0(T) \) and we can write \( u_S = \sum_{j=1}^M u_j \eta_j \). Further, instead of testing with all \( u_S \in S^m_0(T) \) we can simply test with all \( \eta_j \).

This then leads to the problem of finding \( (u_1, \ldots, u_M) \in \mathbb{R}^M \) such that

\[
A \bar{u}^T = \bar{f}^T
\]

where \( A_{jk} := a(\eta_k, \eta_j) \) and \( \bar{f}_j := F(\eta_j) \).

The basis of \( S^m_0(T) \) should be such that \( A \) is as sparse as possible. This is achieved by choosing \( \eta_j \) that vanish on most of the \( T \in T \). In the following subsections, we present different finite elements for the two problems 3 and 4. There, we also specify how the basis is chosen for those finite element spaces. Many more types of elements can be found in [BS94, Bra97].

### 7.3.1 \( P_1 \)-Elements

For problem 4, the discrete space must consist of globally continuous functions. As \( \hat{a}(u, v) \) contains derivatives of \( u \) and \( v \) up to order one, the chosen space \( S^1(T) \) should contain piecewise polynomials of at least first order. We assume that all elements in \( T \) are simplices in \( \Omega \), that is, intervals for \( d = 1 \), triangles for \( d = 2 \), tetrahedrons for \( d = 3 \), etc. Then, the simplest choice for the discrete space is

\[
S^1(T) = P_1(T) := \{ f \in C(\Omega) : \forall T \in T \ f|_T \text{ is affine} \}.
\]
The functions in this space are often referred to as \( P_1 \)-elements or first order Lagrange elements. To select a basis for \( P_1(\mathcal{T}) \), let \( \mathcal{N} = \{N_1, \ldots, N_M\} \) denote the collection of all corners of the elements in \( \mathcal{T} \). We then define piecewise affine, globally continuous functions \( \{\eta_1, \ldots, \eta_M\} \) via \( \eta_j(N_k) = \delta_{jk} \), where \( \delta_{jk} \) denotes the Kronecker symbol, that is \( \delta_{jk} = 1 \) if \( j = k \) and zero otherwise. Then, \( \{\eta_1, \ldots, \eta_M\} \) form a basis for \( P_1(\mathcal{T}) \), provided all elements of \( \mathcal{T} \) are simplices. Also, from the definition of \( \hat{a}(\cdot, \cdot) \), it is easy to see that \( a(\eta_j, \eta_k) = 0 \) unless the corresponding nodes \( N_j \) and \( N_k \) are corners of the same element \( T \in \mathcal{T} \). For \( \Omega \subset \mathbb{R}^1 \), Figure 7.2 shows the two basis functions that do not vanish on the element \( T_j \).

If on a reference element \( \hat{T} = (-1,1) \), we define the two functions

\[
\varphi_1(x) = \frac{(1 - x)}{2} \\
\varphi_2(x) = \frac{(1 + x)}{2}
\]

and further define the transformation from \( \hat{T} \) to \( T_j \) via

\[
\Phi_{T_j}(x) = \frac{(1 - x)N_k + (1 + x)N_{k+1}}{2}
\]

then \( \eta_k|_{T_j} = \varphi_1 \circ \Phi_{T_j} \) and \( \eta_{k+1}|_{T_j} = \varphi_2 \circ \Phi_{T_j} \).

### 7.3.2 \( P_2 \)-Elements

Again, we assume that \( \mathcal{T} \) consists only of simplices. Second order Lagrange elements, or \( P_2 \)-elements, are quadratic polynomials on each \( T \in \mathcal{T} \) and globally \( C^0 \). Hence, they can be used as conform elements for \( H^1(\Omega) \). Obviously, they have a non-vanishing second derivative on each \( T \in \mathcal{T} \). To define a basis, we construct the set \( \mathcal{N} = \{N_1, \ldots, N_M\} \) as consisting of all corners of all elements in \( \mathcal{T} \) as well as the midpoints of all edges, that is, of the lines connecting any two corners of an element \( T \in \mathcal{T} \). Figure 7.3 depicts these nodes for an element \( T \) for the cases \( d = 1, d = 2, \) and \( d = 3 \). We then define piecewise quadratic,
globally continuous functions \( \{ \eta_1, \ldots, \eta_M \} \) via \( \eta_j(N_k) = \delta_{jk} \). They form a basis of

\[
\mathcal{P}_2(T) = \{ f \in C(\Omega) : \forall T \in \mathcal{T} \ f|_T \text{ is a polynomial of maximal order two} \}.
\]

For \( \Omega \subset \mathbb{R}^1 \), Figure 7.4 shows the three basis functions that do not vanish on the element \( T_j = (N_{3j-2}, N_{3j}) \).

Here, on the reference element \( \hat{T} = (-1, 1) \), we define the three functions

\[
\begin{align*}
\varphi_1(x) &= x(x - 1)/2, \\
\varphi_2(x) &= x(x + 1)/2, \\
\varphi_3(x) &= 1 - x^2.
\end{align*}
\]

Then, using the transformation (7.9) yields \( \eta_{3j-2}|_{T_j} = \varphi_1 \circ \Phi_{T_j} \), \( \eta_{3j}|_{T_j} = \varphi_2 \circ \Phi_{T_j} \), and \( \eta_{3j-1}|_{T_j} = \varphi_3 \circ \Phi_{T_j} \).
7.3.3 One-Dimensional Quadratic Macro-Elements

To discretise problem 3 using conform finite elements, we need a finite dimensional subspace of $H^2(\Omega)$. In particular, $S^2(T)$ must be a subspace of $C^1(\Omega)$, that is, the functions and their first derivative must be continuous across the boundaries of the $T \in T$. At the same time, the polynomial degree on each $T \in T$ should be as low as possible and for the basis we want a set of functions that vanish on most of the $T \in T$. To construct such finite elements becomes increasingly difficult in higher dimensions. Here, we present one solution for the case $d = 1$: finite elements, that are piecewise quadratic and globally $C^1$.

Let $T = \{T_1, \ldots, T_M\}$ denote the triangulation of $\Omega = (S_{\min}, S_{\max}) \subset \mathbb{R}$ and define $N = \{N_1, \ldots, N_{M+1}\}$ as the set of the corners of all $T \in T$. We assume that $T$ and $N$ are ordered such that $T_j = (N_j, N_{j+1})$. On our reference element $\hat{T} = (-1, 1)$, we introduce the four piecewise quadratic functions

$$\varphi_1(x) = \begin{cases} (-x^2 - 2x + 1)/2 & x \in (-1,0) \\ (x^2 - 2x + 1)/2 & x \in [0,1) \end{cases}$$

$$\varphi_2(x) = \begin{cases} (x^2 + 2x + 1)/2 & x \in (-1,0) \\ (-x^2 + 2x + 1)/2 & x \in [0,1) \end{cases}$$

$$\varphi_3(x) = \begin{cases} (-3x^2 - 2x + 1)/4 & x \in (-1,0) \\ (x^2 - 2x + 1)/4 & x \in [0,1) \end{cases}$$

$$\varphi_4(x) = \begin{cases} (x^2 + 2x + 1)/4 & x \in (-1,0) \\ (-3x^2 + 2x + 1)/4 & x \in [0,1) \end{cases}$$

Note that $\varphi_j$ and $\varphi'_j$ are continuous at $x = 0$ for $j = 1, \ldots, 4$. Further, we have

$$\varphi_1(-1) = 1 \quad \varphi_1(1) = 0 \quad \varphi'_1(-1) = 0 \quad \varphi'_1(1) = 0$$

$$\varphi_2(-1) = 0 \quad \varphi_2(1) = 1 \quad \varphi'_2(-1) = 0 \quad \varphi'_2(1) = 0$$

$$\varphi_3(-1) = 0 \quad \varphi_3(1) = 0 \quad \varphi'_3(-1) = 1 \quad \varphi'_3(1) = 0$$

$$\varphi_4(-1) = 0 \quad \varphi_4(1) = 0 \quad \varphi'_4(-1) = 0 \quad \varphi'_4(1) = 1$$

Using the transformation (7.9), we then define

$$\eta_{4j-3}|T_j = \varphi_1 \circ \Phi_{T_j}, \quad \eta_{4j-2}|T_j = \varphi_2 \circ \Phi_{T_j}, \quad \eta_{4j-1}|T_j = \varphi_3 \circ \Phi_{T_j}, \quad \eta_{4j}|T_j = \varphi_4 \circ \Phi_{T_j}.$$ 

The factor $\frac{N_{j+1} - N_j}{2}$ in the definition of $\eta_{4j-1}$ and $\eta_{4j}$ ensures that $\eta'_{4j-1}(N_j) = 1$ and $\eta'_{4j}(N_{j+1}) = 1$. 

34
Using \( \{\\eta_1, \ldots, \eta_M\} \) as a basis defines \( S^2(T) \) as a space consisting of piecewise quadratic polynomials that are globally \( C^1 \). As each \( T \in T \) is made up of two similar geometric objects, the functions of \( S^2(T) \) are called macro-elements. Figure 7.5 shows the four basis functions that do not vanish on the element \( T_j = (N_j, N_{j+1}) \).

![Figure 7.5: Four basis functions for one-dimensional quadratic macro-elements](image)

7.3.4 Two-Dimensional Bicubic Hermite Elements

We now turn to problem 3 in two dimensions. In chapter 4, we restricted \( \Omega \subset \mathbb{R}^2 \) to be rectangular. We therefore can construct a triangulation \( T \) consisting only of rectangles. A rectangular finite element that is globally \( C^1 \) is the bicubic Hermite element. On each \( T \in T \), the functions are of the form \( f(x, y) = \sum_{j,k=0}^{3} \alpha_{jk} x^j y^k \). As we have 16 degrees of freedom on each \( T \in T \), we require 16 conditions that define the basis functions with respect to that \( T \). For this, we fix the functional value at each of the four corners (4 conditions), the gradient at the four corners (8 conditions), and the mixed second derivative at each corner (4 conditions). Formally, if \( T \in T \) has the four corners \( N_1, \ldots, N_4 \), we define 16 functionals \( Z_j: C^2(\hat{T}) \rightarrow \mathbb{R}, j = 1, \ldots, 16 \), such that for \( f \in C^2(\hat{T}) \)

\[
Z_1(f) = f(N_1), \quad Z_2(f) = f(N_2), \quad Z_3(f) = f(N_3), \quad Z_4(f) = f(N_4),
\]

\[
Z_5(f) = \partial_x f(N_1), \quad Z_6(f) = \partial_x f(N_2), \quad Z_7(f) = \partial_x f(N_3), \quad Z_8(f) = \partial_x f(N_4),
\]

\[
Z_9(f) = \partial_y f(N_1), \quad Z_{10}(f) = \partial_y f(N_2), \quad Z_{11}(f) = \partial_y f(N_3), \quad Z_{12}(f) = \partial_y f(N_4),
\]

\[
Z_{13}(f) = \partial_x \partial_y f(N_1), \quad Z_{14}(f) = \partial_x \partial_y f(N_2), \quad Z_{15}(f) = \partial_x \partial_y f(N_3), \quad Z_{16}(f) = \partial_x \partial_y f(N_4).
\]

For the 16 basis functions that do not vanish on \( T \), we then require \( Z_j(\eta_k) = \delta_{jk}, j, k = 1 \ldots, 16 \). Figure 7.6 shows the 16 basis functions on the reference element \( \hat{T} = [0,1]^2 \).

![Figure 7.6: 16 basis functions](image)

If we want to allow a more irregular shaped \( \Omega \) or aim for locally refined triangulations, we have to use triangular elements in \( T \). An example for a triangular finite element that
is globally $C^1$ is the Clough-Tocher-element. It is a macro-element with twelve degrees of freedom and described for example in [BS94, Bra97].

### 7.4 Remarks on the Implementation

To implement the finite element method, the main task is to calculate the matrix $A$ and the vector $\vec{f}$ of equation 7.8. For this, we have to evaluate $a(\eta_k, \eta_j)$ and $F(\eta_j)$, where $\eta_j$ and $\eta_k$ are two basis functions resulting from the choice of the finite element space. For
problem 3, \(a(\cdot, \cdot)\) and \(F(\cdot)\) are defined in (7.2). Then, for example for \(a(\eta_k, \eta_j)\), we get

\[
a(\eta_k, \eta_j) = -\sum_{p=1}^{d} \sum_{q=1}^{d} \int_{\Omega} A_{pq} \partial_p \partial_q \eta_k \eta_j \, d\tilde{S} - \sum_{p=1}^{d} \int_{\Omega} b_p \partial_p \eta_k \eta_j \, d\tilde{S} + \int_{\Omega} c \eta_k \eta_j \, d\tilde{S}
\]

\[
= \sum_{T \in T} \left[ -\sum_{p=1}^{d} \sum_{q=1}^{d} \int_{T} A_{pq} \partial_p \partial_q \eta_k \eta_j \, d\tilde{S} - \sum_{p=1}^{d} \int_{T} b_p \partial_p \eta_k \eta_j \, d\tilde{S} + \int_{T} c \eta_k \eta_j \, d\tilde{S} \right].
\]

The integrals on each element \(T \in T\) can be evaluated using the description of the basis functions on some reference element, as presented in the previous section, and the coordinate transformation \(\Phi_T\). The resulting integrals on the reference element can then be evaluated either analytically or using a quadrature formula, depending on the coefficients \(A_{jk}, b_j,\) and \(c\).

For the uncertain parameter model, the parameter functions \(A_{jk}\) contain the two functions \(\rho_{jk}(\cdot)\) and \(\sigma_j(\cdot)\). They depend on the second derivative of the numerical solution from the previous iteration step. Even if the elements in \(S^m(T)\) are piecewise quadratic and hence their second derivative on each \(T \in T\) is constant, the definition of the operator \(M\), which acts on \(\Gamma\) and whose values serves as the argument of \(\sigma_j(\cdot)\) (cf. (2.2)) shows that \(A_{jk}\) in general is not constant on \(T\). To simplify the implementation, we chose to approximate \(\sigma_j(\cdot)\) by \(\sigma_j(M_T(\cdot))\) where \(M_T\) denotes the operator that returns the piecewise integral mean of its argument, that is

\[
M_T(f) = \frac{1}{|T|} \int_{T} f(\tilde{S}) \, d\tilde{S} \quad \text{on each } T \in T.
\]
Chapter 8

Numerical Examples

The numerical solutions presented in this chapter are obtained using self-implemented Mat-
lab programs. The code is given in the appendix A for two schemes for the one dimensional
case.

8.1 Double Barrier Call Option with Fixed Parameters

The first example is a european call option expiring at time $T$ with a down & out as well
as an up & out barrier $B_{\text{min}}$ and $B_{\text{max}}$, respectively. The strike $E$ lies between the two
barriers. The volatility $\sigma$ is chosen to be fixed. As the problem has an analytic solution
which is given in the Appendix, B.1, it serves as a validator of our numerical schemes. The
specific parameter values chosen are given in table 8.1. Figure 8.1 compares the analytic

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>down &amp; out barrier $B_{\text{min}}$</td>
<td>10</td>
</tr>
<tr>
<td>up &amp; out barrier $B_{\text{max}}$</td>
<td>100</td>
</tr>
<tr>
<td>expiry $T$</td>
<td>1</td>
</tr>
<tr>
<td>strike $E$</td>
<td>60</td>
</tr>
<tr>
<td>risk-free rate $r$</td>
<td>5%</td>
</tr>
<tr>
<td>volatility $\sigma$</td>
<td>30%</td>
</tr>
</tbody>
</table>

Table 8.1: Parameters for the Double Barrier Option with Fixed Parameters

solution (solid curve without markers) to the numerical solution of the finite difference
method (solid curve with triangular markers) as well as the finite element method using $P_1$-
elements (dash-dotted curve with squared markers), using $P_2$-elements (dashed curve with
circular markers), and quadratic macro-elements (dotted curve with star-markers). The top
row shows the option value $V(S)$, the middle row displays the option delta, $\Delta(S)$, and the
last row gives the option gamma, $\Gamma(S)$. The left column displays the numerical solution
Figure 8.1: Double Barrier Option with Fixed Parameters
on a coarse mesh, showing how the different schemes approximate the solution. The right column gives the solutions on a fine mesh to present the quality of the methods. As $P_1$-elements are piecewise affine, their second derivative vanishes. To give an approximation of gamma (the second derivative), the piecewise constant first derivative was mapped to a piecewise affine function by mapping to each element node the weighted mean of the first derivative on the adjacent elements and connecting the points. The resulting piecewise affine function is again a piecewise differentiable function. As sub figure 8.1(f) shows, the procedure leads to a reasonable piecewise constant approximation of the true gamma.

8.2 Double Barrier Call Option with Uncertain Volatility

For the next example, we again use the double barrier call option from section 8.1 but this time the volatility is only known to lie in a given interval, $\sigma^- \leq \sigma \leq \sigma^+$. We look for the worst case solution. Table 8.2 gives the financial parameters for this example. The finite element method using $P_1$-elements and using $P_2$-elements has been used as falsely "suggested" in subsection 7.2.1 by simply replacing $\sigma$ by the function

$$
\sigma = \begin{cases} 
\sigma^- & \text{if } \partial^2 SSV_{L,K}^{-1} > 0, \\
\sigma^+ & \text{else.}
\end{cases}
$$

Figure 8.2 shows the numerical solutions for the finite difference scheme (solid curve with triangles) and for the finite element scheme using $P_1$-elements (dash-dotted curve with squares), using $P_2$-elements (dashed curve with circles), and quadratic macro-elements (dotted curve with stars). Again, the first row shows the option value, the second the delta, and the third row the gamma of the option. The three columns show the numerical solutions with decreasing time-step-size. Next to the numerical solutions, the analytical solution of the double barrier call option with $\sigma = \sigma^-$ and $\sigma = \sigma^+$ is displayed using a dash-dotted and a dashed curve, respectively.

<table>
<thead>
<tr>
<th>down &amp; out barrier $B^{\min}$</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>up &amp; out barrier $B^{\max}$</td>
<td>100</td>
</tr>
<tr>
<td>expiry $T$</td>
<td>1</td>
</tr>
<tr>
<td>strike $E$</td>
<td>60</td>
</tr>
<tr>
<td>risk-free rate $r$</td>
<td>5%</td>
</tr>
<tr>
<td>minimal volatility $\sigma^-$</td>
<td>20%</td>
</tr>
<tr>
<td>maximal volatility $\sigma^+$</td>
<td>40%</td>
</tr>
</tbody>
</table>

Table 8.2: Parameters for the Double Barrier Call Option with Uncertain Volatility
Figure 8.2: Double Barrier Option with Uncertain Volatility
Figure 8.2 clearly shows how using the improper elements leads to false results, even for a very small time-step-size. On the other hand, the solution for the finite difference scheme practically coincides with the solution for the finite element method using quadratic macro-elements that are globally $C^1$. Furthermore, for the two false methods, the iteration that approximated the non-linearity never converged and was always stopped by the maximal number of iterations. For the finite difference scheme and the macro-elements, the iteration-scheme converged in each time step in less than ten iteration steps, even for the largest time-step size. The tolerance for the relative difference from one iteration step to the next as well as the tolerance for the relative error in the updated linear system were both chosen to be 0.001.

8.3 Digital Call Option

As a final numerical example in one dimension, we consider a european digital call option. We compare the numerical solution for both cases, using a fixed volatility as well as uncertain volatility. The chosen financial parameters are given in table 8.3. For a fixed volatility, the problem has an analytic solution which is given in the appendix, B.2. To solve the fixed parameter problem numerically, we again used all four possible schemes presented in this work, namely the finite difference method and the finite element method using $P_1$-elements, $P_2$-elements, and quadratic macro-elements. For the uncertain volatility version, we only used the mathematically rational scheme, that is, the finite difference method and the finite element method with quadratic macro-elements. Figure 8.3 displays the solutions for both schemes. The left column shows the option value (top), the delta (middle), and the gamma (bottom) for the problem with fixed volatility. Next to the numerical solutions the exact solution is printed. The right column gives the numerical solutions for the uncertain volatility problem and the two analytic solutions using both extremal volatility values as fixed volatility. The same marking is used as in the previous two examples, that is a solid curve with triangles for the finite difference scheme, a dash-dotted curve with squares for the finite element scheme using $P_1$-elements, a dashed curve with circles for using $P_2$-elements, a

<table>
<thead>
<tr>
<th></th>
<th>fixed</th>
<th>uncertain</th>
</tr>
</thead>
<tbody>
<tr>
<td>expiry $T$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>strike $E$</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>risk-free rate $r$</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>volatility $\sigma$</td>
<td>30%</td>
<td>20% – 40%</td>
</tr>
</tbody>
</table>

Table 8.3: Parameters for the Digital Call Option
Figure 8.3: Digital Call Option with Fixed and with Uncertain Volatility
dotted curve with stars for quadratic macro-elements, a dash-dotted curve with no markers for the analytic solution using $\sigma = \sigma^-$, and a dashed curve with no markers for the case $\sigma = \sigma^+$. All schemes applied give similar reasonable solutions. However, the implementational effort for the $P_2$-elements and even more for the quadratic macro-elements is much higher compared to that for the $P_1$-elements. Also, $P_2$-elements and quadratic macro-elements lead to more non-zero matrix entries. Hence, taking a second look at the numerical solutions for the fixed volatility case clearly suggests the use of $P_1$-elements (or finite differences) over $P_2$-elements or quadratic macro-elements. The solutions are of comparable quality for these simple elements — provided that the volatility is fixed.

8.4 Exchange Option with Fixed Parameters

With the fourth example, we turn to problems with two underlying assets. We start again with an example with fixed parameters that has an analytic solution. For that, we choose a european exchange option with the pay-off

$$ P(S_1, S_2) = \max(S_2 - S_1, 0). $$

Its analytic solution is given in the appendix, in B.3. We set the financial parameters as given in table 8.4. Figure 8.4 shows the analytic solution. More precisely, sub figure 8.4(a) gives

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>expiry $T$</td>
<td>1</td>
</tr>
<tr>
<td>risk-free rate $r$</td>
<td>5%</td>
</tr>
<tr>
<td>volatility of first asset, $\sigma_1$</td>
<td>30%</td>
</tr>
<tr>
<td>volatility of second asset, $\sigma_2$</td>
<td>40%</td>
</tr>
<tr>
<td>correlation $\rho$</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 8.4: Parameters for the Exchange Option with Fixed Parameters

the option value, sub figures 8.4(b) and 8.4(c) the gradient in the $S_1$- and the $S_2$-direction, respectively, and sub figures 8.4(d)–8.4(f) the second derivative in the $S_1$-direction, the second derivative in the $S_2$-direction, and the mixed second derivative, in turn. Figure 8.5 shows the same graphs for the numerical solution using finite differences. The grid displayed coincides with the grid used for the asset space discretisation. Note that the pay-off has a jump in the first derivatives along the line $S_1 = S_2$. In particular, this means that this jump will always cross an artificial cut-off boundary. It is not possible to formulate good Dirichlet type boundary conditions in the vicinity of that line without knowing the analytic
Figure 8.4: Analytic Solution of the Exchange Option with Fixed Parameters
Figure 8.5: Exchange Option with Fixed Parameters, solved using Finite Differences
solution. We used the pay-off function as the source for the Dirichlet condition at the cut-off boundary, i.e. \( V_D(S_1, S_2, t) = \max(S_2 - S_1, 0) \) on \( \partial \Omega \). This leads to singularities in the numerical derivatives near the boundary where \( S_1 \approx S_2 \). The cut-off boundary must thus be chosen large enough so that the influence on the region of interest is small. A different cut-off boundary condition might be preferred for this example.

Finally, figure 8.6 displays the solution from the finite element method using bicubic Hermite elements. Here, the grid shows the piecewise bicubic elements used to solve the problem numerically. As the number of degrees of freedom is 16 on each element, we chose a coarser mesh for the finite element method compared to the grid for the finite difference method.

Aside from the error where the line \( S_1 = S_2 \) crosses the cut-off boundary, both numerical schemes approximate the analytic solution equally well.

### 8.5 Exchange Option with Barrier and Uncertain Parameters

As the first example for an option with two underlyings and uncertain parameters, we again choose the european exchange option. However, to increase the variation in the second derivative inside the domain, we set knock-out barriers on all four sides of the domain. More precisely, we use the financial parameters given in table 8.5. Using barriers on all sides of \( \Omega \) also simplifies the boundary conditions to homogeneous Dirichlet-type conditions. Figures 8.7 and 8.8 display the numerical solution using finite differences and finite bicubic Hermite elements. Again, the grids in the graph coincide with the asset space discretisation used for the two schemes.

| down & out barrier for \( S_1 \), \( B_1^{\text{min}} \) | 40 |
| up & out barrier for \( S_1 \), \( B_1^{\text{max}} \) | 120 |
| down & out barrier for \( S_2 \), \( B_2^{\text{min}} \) | 40 |
| up & out barrier for \( S_2 \), \( B_2^{\text{max}} \) | 120 |
| expiry \( T \) | 1 |
| risk-free rate \( r \) | 5% |
| volatility of first asset, \( \sigma_1 \) | 25% – 35% |
| volatility of second asset, \( \sigma_2 \) | 30% – 50% |
| correlation \( \rho \) | -10% – +30% |

Table 8.5: Parameters for the Exchange Option with Barriers and Uncertain Parameters
Figure 8.6: Exchange Option with Fixed Parameters, solved using Finite Elements
Figure 8.7: Exchange Option with Barriers and Uncertain Parameters, Option Value and First Derivatives
Figure 8.8: Exchange Option with Barriers and Uncertain Parameters, Second Derivatives
Here, to simplify comparison, we placed the solutions of the two schemes next to each other. The left column contains $V$, $\Delta_1$, $\Delta_2$, $\Gamma_{11}$, $\Gamma_{22}$, and $\Gamma_{12}$ (in turn) from the finite difference scheme, the right column those resulting from the finite element scheme using bicubic elements. Except on the very outer border of $\Omega$ in the second derivatives, both solution sets are practically identical.

In both cases, we chose 100 time-steps for the implicit Euler time-discretisation. We stopped the iteration approximating the non-linearity when

- the relative change in the solution from one iteration step to the next (measured in the $\ell^2$-norm) became smaller than 0.001;
- the error in the updated linear equation system relative to the size of the solution (measured in the $\ell^2$-norm) became smaller than 0.001;
- the number of iterations within one time-step reached the maximal number of 10 iteration steps.

For the finite difference scheme, the iteration converged within one to three iteration steps for all time-steps. Contrasting this, the iteration in the finite element scheme was always stopped by the maximal number of iterations. Although not displayed here, we remark, that better convergence was observed when the time-step size was reduced. Also, the remaining error in the updated linear system was much larger close to the expiry of the option. Near expiry, the second derivative of the numerical solution becomes almost singular.

### 8.6 Worst-of-Two Call Option with Barrier

Our last example is based on the two-dimensional example in [ZVF00]. It is a european worst-of-two call option with up & out as well as down & out barriers for both underlyings. Its pay-off is

$$P(S_1, S_2) = \max(\min(S_1, S_2) - E, 0).$$

We modified the problem in [ZVF00] in the way that we changed $\sigma_1$, $\sigma_2$, and $\rho$ to be uncertain. Table 8.6 displays the financial parameters. In figure 8.9 we compare the option value and the first derivatives as calculated by the finite difference method and by the finite element scheme using bicubic Hermite elements. Figure 8.10 compares the numerical second derivatives for these two methods. As in the previous example, the left column displays the solutions from the finite difference method, the right those from the finite element method.
Figure 8.9: Worst-of-Two Call Option with Barrier and Uncertain Parameters, Option Value and First Derivatives
Figure 8.10: Worst-of-Two Call Option with Barrier and Uncertain Parameters, Second Derivatives
Table 8.6: Parameters for the Worst-of-Two Call Option with Barriers and Uncertain Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>down &amp; out barrier for $S_1$, $B_1^\text{min}$</td>
<td>90</td>
</tr>
<tr>
<td>up &amp; out barrier for $S_1$, $B_1^\text{max}$</td>
<td>120</td>
</tr>
<tr>
<td>down &amp; out barrier for $S_2$, $B_2^\text{min}$</td>
<td>90</td>
</tr>
<tr>
<td>up &amp; out barrier for $S_2$, $B_2^\text{max}$</td>
<td>120</td>
</tr>
<tr>
<td>expiry $T$</td>
<td>0.25</td>
</tr>
<tr>
<td>strike $E$</td>
<td>100</td>
</tr>
<tr>
<td>risk-free rate $r$</td>
<td>5%</td>
</tr>
<tr>
<td>volatility of first asset, $\sigma_1$</td>
<td>30% - 50%</td>
</tr>
<tr>
<td>volatility of second asset, $\sigma_2$</td>
<td>10% - 30%</td>
</tr>
<tr>
<td>correlation $\rho$</td>
<td>-60% - -40%</td>
</tr>
</tbody>
</table>

Again, the solutions of the two methods are very much comparable. Only in the second derivatives the solutions of the two schemes differ notably. In particular, in this example, the discontinuity of the second derivatives $\partial^2 S_1$ and $\partial^2 S_2$ become prominent in those areas where the finite difference scheme shows spurious oscillations.

For the time discretisation, we used 400 time steps. As stopping criteria for the iteration scheme, we again used a tolerance of 0.001 for both, the relative difference in the solution from one iteration step to the next and the relative error in the updated linear equation, as well as a maximal number of ten iterations per time step. The finite difference schemes only did not converge within ten iterations for the first eight time steps. After that, convergence was reached for each time step. Again, the finite element scheme showed a worse convergence behaviour in the iteration scheme. Only for the last 40 time steps convergence was reached with less than ten iteration steps. Most likely this is due to the fact that the parameters are non-constant on each element. As stated in section 7.4, we chose a very simple approximation using piecewise integral means. This, combined with the use of a coarse finite element grid leads to a rather rough approximation of the true parameter values. A finer mesh and / or a better approximation of the non-constant parameter value on each element very likely improve the convergence for the finite element method. However, we stress again that, regardless of the bad convergence of the iteration scheme when using finite elements, the resulting numerical solution is comparable to the finite difference solution.
Chapter 9

Closing Remarks

In this work, we derived a non-linear extension of the Black-Scholes equation resulting from an uncertain parameter model. It allows at the same time uncertain volatility and uncertain correlations for arbitrarily many underlyings. We suggested a simple iteration scheme to approximate the non-linearity. It was seen that the scheme does not necessarily converge, especially for larger time steps close to maturity, but still, it lead to reasonable results in all sensible examples presented.

The main focus of this work was on the finite element method for the non-linear Black-Scholes model derived and its comparison to the finite difference method. We showed that the finite elements used to numerically approximate the non-linear problem must be of higher order than those for the constant parameter model. In particular, we need elements that have a continuous first derivative. These elements lead to matrices that have more non-zero entries than the matrix resulting from the finite difference scheme. Hence, one has to judge whether to use finite differences or finite elements based on several aspects:

- The finite difference method is well established, very easy to implement, especially in one dimension, and there are numerous convergence results and error estimates. Even adaptive local mesh refinement is not a difficulty for the finite difference method in one dimension.

- The use of fine elements might become relevant if error estimates and / or convergence results are needed not point wise but over whole integrals. The finite difference method only approximates the solution in a point wise sense.

- There are problems where the parameters are not smooth enough to provide the necessary prerequisites for formulating the common stability and convergences results of the finite difference method. As was seen in section 7.2, the requirements on the
parameters (including the pay-off function and the boundary condition) to formulate the weak problem that serves as the start-off for the finite element discretisation, were much weaker. In such a case the use of the finite element method might be necessary to ensure a converging solution.

In two and more dimensions, there are two additional reasons to favour the finite element method over the finite difference method. Firstly, if the asset space domain is irregular due to irregular barriers, for example, discretising the domain using triangular or even curved elements is much simpler than approximating the domain with a finite difference grid. Secondly, (adaptive) local mesh refinement is straightforward and very common when using triangular finite elements. Numerous approaches have been made in mechanics. Locally refined meshes for the finite difference method are restricted to special cases, only.

Finally, when using finite element techniques, there are several alternative ways to deal with unbounded domains, besides introducing artificial boundary conditions. One approach is the coupling of finite elements with boundary elements. Another method uses so-called infinite elements. Both methods have the advantage of not assuming particular values on the boundary but applying the partial differential equation in the unbounded region. For both, numerous works have been published, mainly for problems arising in mechanical engineering.

This work bases its arguments for the choice of the specific elements used to solve the non-linear problem on common results in the theory of finite elements. Further, it provides numerical evidence of the right choice. However, a rigorous analysis is left out and is one of the next steps to take.

A possible variation of the method presented here is regarding the uncertain parameters as unknowns, coupled to the option $V$ as given in (2.2), and applying mixed finite element methods. We do not go into details on this method but refer to works on, for example, the Navier-Stokes equation where the use of mixed elements is common.
Appendix A

Program Code

The code for the finite element method is written in Matlab in the spirit of the series [ACF99, ACFK02, CK02, BCH04] and we refer to these for details on the implementation rules. The code for the finite difference scheme is structured to match that of the finite element program to reach a common main program that is given in the first section. The following two sections contain the scheme-depending functions. The last section gives the problem-depending functions for the double barrier option.

We stress that the code is written under academic aspects rather than reaching optimal performance. There is much potential for improving the run-time of the programs.

A.1 Main Program for the One-Dimensional Problem

The main program is written as to be common for both schemes. The functions called from the main program differ for the various schemes.

```matlab
function [grid,V] = UPM1D(K,theta,Smin,Smax,N,MaxCnt,change_tol,eqn_tol)
% Numerically solves the 1D Uncertain Parameter Model
% INPUT
% K   number of time steps
% theta parameter for general mid-point scheme in time
% Smin lower boundary for asset space
% Smax upper boundary for asset space
% N   number of asset space grid points
% MaxCnt maximal number of iterations in one time-step
% change_tol  tolerance for stopping criterion "no more changes"
% eqn_tol     tolerance for stopping criterion "convergence"
```
% OUTPUT

% grid asset space grid parameters as structure-type variable
% V solution array in time and space

% initialise financial parameters as global variables
init_financial_parameters;
global sigma_min sigma_max

% create time-space-grid and dimensionalise V
[dt,grid,V] = create_grid(K,Smin,Smax,N);

% get operator to calculate Gamma from V
GammaOperator = second_derivative(grid);

% get final condition
V(:,K+1) = final_condition(grid);

% get matrix for identity operator
id = IDmatrix(grid);

% step backwards in time using generalized Euler-scheme
for m = K:-1:1
    % iterate to solve non-linearity
    for cnt = 0:MaxCnt
        if cnt == 0
            % set solution from previous time-step as initial value for iteration
            V(:,m) = V(:,m+1);
            relchange = realmax;
        else
            % apply Dirichlet condition
            Vnext = zeros(size(V(:,m)));
            Vnext(grid.DirichletNodes) = dirichlet_condition(grid,(m-1)*dt);
            b = b - A*Vnext;

            % calculate new value for V
            Vnext(grid.FreeNodes) = A(grid.FreeNodes,grid.FreeNodes) \ b(grid.FreeNodes);
            relchange = norm(V(:,m) - Vnext) / max([norm(V(:,m)),norm(Vnext),1]);
            V(:,m) = Vnext;
        end
    end
    % stop if relative change is below tolerance
    if relchange < change_tol break, end

    % estimate uncertain parameters from current V
    Gamma = GammaOperator * V(:,m);
    sigma = sigma_min*(Gamma>0) + sigma_max*(Gamma<=0);

    % get discrete Black-Scholes operator and set up linear system
    L = BSMatrix(grid,sigma);
    A = id + dt*theta*L;
    b = (id - dt*(1-theta)*L)*V(:,m+1);
% stop if current solution solves updated linear system
ErrInEqn = norm(A(grid.FreeNodes,:)*V(:,m) - b(grid.FreeNodes)) / max([norm(V(:,m)),1]);
if ErrInEqn < eqn_tol break, end

% display warning if maximum number of iterations is reached
if cnt == MaxCnt
disp(sprintf('Maximum number of iterations reached for t = %0.5g !',(m-1)*dt))
disp(sprintf('Last rel. change in iteration: %0.5g',relchange))
disp(sprintf('Last rel. error in the equation: %0.5g',ErrInEqn))
end
end

% display warning if maximum number of iterations is reached
if cnt == MaxCnt
disp(sprintf('Maximum number of iterations reached for t = %0.5g !',(m-1)*dt))
disp(sprintf('Last rel. change in iteration: %0.5g',relchange))
disp(sprintf('Last rel. error in the equation: %0.5g',ErrInEqn))
end
end

A.2 Functions for the Finite Difference Scheme

function [dt,grid,V] = create_grid(K,Smin,Smax,N)
% creates the grid for the 1D finite difference method

global T

dt = T/K;

% asset-space grid
grid.dS = (Smax-Smin)/N;
grid.S = (0:N)'*grid.dS + Smin;

% define free nodes and nodes for dirichlet boundary
grid.FreeNodes = 2:N;
grid.DirichletNodes = [1; N+1];

% dimensionalise V
V = zeros(N+1,K+1);

function gamma = second_derivative(grid)
% calculates gamma as matrix such that gamma*v yields the numerical second
% derivative of V, using 1D finite differences of order two

% get dimension of linear system
N = length(grid.S);

% set gamma operator on inner grid points
p = 2:(N-1);
gamma = (sparse(p,p+1,1,N,N) + sparse(p,p,-2,N,N) + sparse(p,p-1,1,N,N)) / (grid.dS^2);

% gamma operator at Smin
% gamma operator at Smax
gamma(end,(end-3):end) = [-1 4 -5 2] / (grid.dS^2);

function V = final_condition(grid)
% set final condition for 1D finite difference method
V = Payoff(grid.S);

function id = IDmatrix(grid)
% calculate identity operator for 1D finite difference method
id = sparse(1:length(grid.S),1:length(grid.S),1);

function L = BSmatrix(grid,sigma)
% calculates discrete Black-Scholes operator for 1D finite difference method
global r
N = size(grid.S,1);
I = r*sparse(1:N,1:N,1,N,N);
F = sparse(2:N-1,3:N,d,N,N) - sparse(2:N-1,1:N-2,d,N,N);
D = sigma(2:end-1).^2/2 .* grid.S(2:end-1).^2 / (grid.dS^2);
D = sparse(2:N-1,3:N,d,N,N) - 2*sparse(2:N-1,2:N-1,d,N,N) + sparse(2:N-1,1:N-2,d,N,N);
L = I - F - D;

function VAtDirichletNodes = dirichlet_condition(grid,t)
% calculate V at the dirichlet nodes
VAtDirichletNodes = V_D(grid.S(grid.DirichletNodes),t);

% # BSMatrix.m
% Calculates discrete Black-Scholes operator for 1D finite difference method.
% Inputs:
% grid - a structure containing grid parameters.
% sigma - volatility parameter.
% Outputs:
% L - the discrete Black-Scholes operator.

gamma(1,1:4) = [2 -5 4 -1] / (grid.dS^2);
% gamma operator at Smax
gamma(end,(end-3):end) = [-1 4 -5 2] / (grid.dS^2);
A.3 Functions for the Finite Element Method (Quadratic Macro-Elements)

```
function [dt,grid,V] = create_grid(K,Smin,Smax,N)
% creates grid for the 1D finite element method using Macro-elements

global T

% time-step size
dt = T/K;

% asset-space grid
grid.dS = (Smax-Smin)/N;
grid.coordinates(1:2:2*N+1,1) = [0:N]'*grid.dS + Smin;
grid.coordinates(2:2:2*N+2,1) = [0:N]'*grid.dS + Smin;

% define free nodes and nodes for dirichlet boundary
grid.DirichletNodes = [1; 2*N+1];
grid.FreeNodes = setdiff(1:2*(N+1),grid.DirichletNodes);

% dimensionalise V
V = zeros(2*(N+1),K+1);
```

```
function gamma = second_derivative(grid)
% calculates gamma as matrix such that gamma*v yields the numerical second
% derivative of V on each sub-element

% initialise matrix
gamma = sparse(2*size(grid.elements,1),size(grid.coordinates,1));

% assemble matrix
for j = 1:size(grid.elements,1)
a = grid.coordinates(grid.elements(j,1));
b = grid.coordinates(grid.elements(j,2));
DinvPhi_T = 2 / (b-a);
D2phi = macro_D2phi([-1 1],(b-a)/2);
GammaOperator([2*j-1,2*j],grid.elements(j,:)) = D2phi' * DinvPhi_T^2;
end
```

```
function V = final_condition(grid)
% set final condition for 1D FEM using Macro-elements

N = size(grid.coordinates,1);
V = zeros(N,1);
V(1:2:N-1) = Payoff(grid.coordinates(1:2:N-1));
V(2:2:N) = PayoffDeriv(grid.coordinates(1:2:N-1),grid.dS);
```
function id = IDmatrix(grid)
% calculate identity operator for 1D FEM using Macro-elements

% identity operator on the reference element
id_ref = [92 28 27 -13; 28 92 13 -27; 27 13 10 -6; -13 -27 -6 10]/120;

% assemble global identity operator
id = sparse(size(grid.coordinates,1),size(grid.coordinates,1));
for j = 1:size(grid.elements,1)
    DPhi_T = (grid.coordinates(grid.elements(j,2)) ... 
    - grid.coordinates(grid.elements(j,1))) / 2;
    index = grid.elements(j,:);
    id(index,index) = id(index,index) + DPhi_T * id_ref;
end

function L = BSmatrix(grid,sigma)
% calculates the discrete Black-Scholes operator for 1D FEM using macro-elements

% Assembly of the matrix L
L = sparse(size(grid.coordinates,1),size(grid.coordinates,1));
for j = 1:size(grid.elements,1)
    index = grid.elements(j,:);
    L(index,index) = L(index,index) ... 
    + local_BSmatrix(grid.coordinates(grid.elements(j,1:2)), ... 
    sigma([2*j-1, 2*j]));
end

function BS = local_BSmatrix(vertices, sigma)
% computes local matrix of discrete Black-Scholes operator for a Macro-element

% define Gauss-Quadrature points and weights for two sub-elements
x = [-sqrt(3/5) 0 sqrt(3/5)];
w = [5/9 8/9 5/9];
x = [(x-1)/2, (x+1)/2];
w = [w, w]/2;

% evaluate necessary functions at Gauss points
phi = macro_phi(x,(b-a)/2);
Dphi = macro_Dphi(x,(b-a)/2);
D2phi = macro_D2phi(x,(b-a)/2);
Phi_T = (a*(1-x)+b*(1+x))/2;
DPhi_T = (b-a) / 2;
DinvPhi_T = 2 / (b-a);
sigma = sigma(1)*(x<0) + sigma(2)*(x>=0);

% local mass matrix
Mass = r*DPhi_T * [92 28 27 -13; 28 92 13 -27; 27 13 10 -6; -13 -27 -6 10]/120;

% flux and diffusion term, evaluated using quadrature on each sub-element
for j = 1:4
    for k = 1:4
        Flux(j,k) = r * (phi(j,:) .* Dphi(k,:) .* Phi_T)*w';
        Diff(j,k) = DinvPhi_T * (phi(j,:) .* D2phi(k,:) .* sigma.^2/2 .* Phi_T.^2)*w';
    end
end

% combine terms to discrete local BS-operator
BS = Mass - Flux - Diff;

% calculate V at the dirichlet nodes
VAtDirichletNodes = V_D(grid.coordinates(grid.DirichletNodes),t);

function VAtDirichletNodes = dirichlet_condition(grid,t)
% calculate V at the dirichlet nodes
VAtDirichletNodes = V_D(grid.coordinates(grid.DirichletNodes),t);

function phi = macro_phi(x,h);
% evaluates the four basis functions
phi = zeros(4,length(x));
phi(1,:) = (-1/2*x.^2 .* (x<0) + 1/2*x.^2 .* (x>=0)) -x +1/2;
phi(2,:) = ( 1/2*x.^2 .* (x<0) - 1/2*x.^2 .* (x>=0)) +x +1/2;
phi(3,:) = h*((-3/4*x.^2 .* (x<0) + 1/4*x.^2 .* (x>=0)) -x/2 +1/4);
phi(4,:) = h*((-1/4*x.^2 .* (x<0) + 3/4*x.^2 .* (x>=0)) -x/2 -1/4);

function Dphi = macro_Dphi(x,h);
% evaluates the first derivative of the four basis functions
Dphi = zeros(4,length(x));
Dphi(1,:) = (- x .* (x<0) + x .* (x>=0)) -1;
Dphi(2,:) = ( x .* (x<0) - x .* (x>=0)) +1;
Dphi(3,:) = h*((-3/2*x .* (x<0) + 1/2*x .* (x>=0)) -1/2);
Dphi(4,:) = h*((-1/2*x .* (x<0) + 3/2*x .* (x>=0)) -1/2);

function D2phi = macro_D2phi(x,h);
% evaluates the second derivative of the four basis functions
% on the reference element [-1,1]

D2phi = zeros(4,length(x));
D2phi(1,:) = ( - (x<0) + (x>=0));
D2phi(2,:) = ( (x<0) - (x>=0));
D2phi(3,:) = h*(-3/2* (x<0) + 1/2* (x>=0));
D2phi(4,:) = h*(-1/2* (x<0) + 3/2* (x>=0));

A.4 Problem Depending Functions: Double Barrier with Uncertain Volatility

function init_financial_parameters
% initialise financial parameters for Double Barrier Call Option
% with uncertain volatility

global r sigma_min sigma_max T E

r = 0.05; % risk-free rate
sigma_min = 0.2; % minimal volatility
sigma_max = 0.4; % maximal volatility
T = 1; % expiry
E = 60; % strike

function P = Payoff(S)
% Payoff of Double Barrier Call option with strike E

global E

P = max(S-E,0);

function dP = PayoffDeriv(S,dS)
% Derivative of Payoff of Double Barrier Call option with strike E
% it is assumed that Smin < E < Smax and that the barriers coincide
% with the asset space boundaries

global E

dP = zeros(length(S),1);
dP(find(S==E)) = 1/2;
dP(find(S>E)) = 1;

function VAtDirichletNodes = V_D(S,t)
% returns the dirichlet condition for the double barrier call option
3
4 VATDirichletNodes = zeros(length(S),1);
________________________ End of VD.m __________________________
Appendix B

Analytic Solutions for Numerical Examples

B.1 European Double Barrier Call Option

The analytic solution for the european double barrier call option can be derived using a fourier ansatz. It yields

\[
V(S, t) = B_{\text{min}}^{1-\alpha} S^\alpha e^{-(1-\alpha)^2 \tau} \sum_{n=1}^{\infty} a_n \sin(\lambda_n x(S)) \exp\left(-\lambda_n^2 \tau\right)
\]

\[
V_S(S, t) = B_{\text{min}}^{1-\alpha} S^{\alpha-1} e^{-(1-\alpha)^2 \tau} \sum_{n=1}^{\infty} a_n \left[ \alpha \sin(\lambda_n x(S)) + \lambda_n \cos(\lambda_n x(S)) \right] \exp(-\lambda_n^2 \tau)
\]

\[
V_{SS}(S, t) = B_{\text{min}}^{1-\alpha} S^{\alpha-2} e^{-(1-\alpha)^2 \tau} \sum_{n=1}^{\infty} a_n \left[ ((\alpha-1) \alpha - \lambda_n^2) \sin(\lambda_n x(S)) + (1 + \alpha) \lambda_n \cos(\lambda_n x(S)) \right] \exp(-\lambda_n^2 \tau)
\]

where we used the abbreviations \( \alpha = (1 - 2 \frac{r}{\sigma^2}) / 2 \), \( \tau = \frac{\sigma^2}{2} (T - t) \), \( x(S) = \log \frac{S}{B_{\text{min}}} \), \( \tilde{x} = \log \frac{B_{\text{max}}}{B_{\text{min}}} \), \( k = \log \frac{E}{B_{\text{min}}} \), \( \lambda_n = \frac{n \pi}{\tilde{x}} \), and

\[
a_n = 2 \frac{(-1)^{n+1} n \pi e^{(1-\alpha) \tilde{x}} + e^{(1-\alpha) k} \left( n \pi \cos(n \pi \frac{\tilde{k}}{2}) - (1-\alpha) \tilde{x} \sin(n \pi \frac{\tilde{k}}{2}) \right)}{(1-\alpha)^2 \tilde{x}^2 + (n \pi)^2}
\]

\[-2 \frac{(-1)^{n+1} n \pi e^{k-\alpha \tilde{x}} + e^{(1-\alpha) k} \left( n \pi \cos(n \pi \frac{\tilde{k}}{2}) + \alpha \tilde{x} \sin(n \pi \frac{\tilde{k}}{2}) \right)}{\alpha^2 \tilde{x}^2 + (n \pi)^2}\]
B.2 European Digital Call Option

The analytic solution of the european digital call option is

\[ V(S,t) = e^{-r(T-t)} N(d_2(S)) \]
\[ V_S(S,t) = e^{-r(T-t)} N'(d_2(S)) d_2'(S) \]
\[ V_{SS}(S,t) = e^{-r(T-t)} \left[ N''(d_2(S)) (d_2(S))^2 + N'(d_2(S)) d_2'(S) \right] \]

where \( N \) denotes the normal cumulative, that is

\[ N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\xi^2/2} d\xi \]
\[ N'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \]
\[ N''(x) = \frac{-x}{\sqrt{2\pi}} e^{-x^2/2} \]

and we used the function \( d_2(S) \), defined as

\[ d_2(S) = \frac{\log \frac{S}{E} + (r - \sigma^2/2)(T-t)}{\sigma \sqrt{T-t}} \]

so that

\[ d_2'(S) = \frac{1}{S \sigma \sqrt{T-t}}, \quad \text{and} \quad d_2''(S) = -\frac{1}{S^2 \sigma \sqrt{T-t}}. \]

B.3 European Exchange Option

The analytic solution of the european exchange option with the pay-off \( \max(S_2 - S_1, 0) \) can be found using the ansatz \( V(S_1, S_2, t) = S_1 W(S_2/S_1, t) \). We get the result

\[ V(S_1, S_2, t) = S_2 N(d_1) - S_1 N(d_2) \]
\[ V_{S_1}(S_1, S_2, t) = -N(d_2) - \frac{1}{\sigma \sqrt{T-t}} (S_2/S_1 N'(d_1) - N'(d_2)) \]
\[ V_{S_2}(S_1, S_2, t) = N(d_1) - \frac{1}{\sigma \sqrt{T-t}} (N'(d_1) - S_2/S_1 N'(d_2)) \]
\[ V_{S_1S_1}(S_1, S_2, t) = \frac{S_2/S_1 N'(d_1) + N'(d_2)}{S_1 \sigma \sqrt{T-t}} + \frac{S_2/S_1 N''(d_1) - N''(d_2)}{S_1 \sigma^2 (T-t)} \]
\[ V_{S_2S_2}(S_1, S_2, t) = \frac{S_1/S_2 N'(d_2) + N'(d_1)}{S_2 \sigma \sqrt{T-t}} + \frac{S_1/S_2 N''(d_2) - N''(d_1)}{S_2 \sigma^2 (T-t)} \]
\[ V_{S_1S_2}(S_1, S_2, t) = -\frac{N'(d_1)/S_1 + N'(d_2)/S_2}{\sigma \sqrt{T-t}} - \frac{N''(d_1)/S_1 - N''(d_2)/S_2}{\sigma^2 (T-t)} \]
where $N(x)$ is the normal cumulative defined in section B.2 and we use the abbreviations

\[ d_1 = \frac{\log \frac{S_k}{S_1} + \sigma^2 / 2(T - t)}{\sigma \sqrt{T - t}}, \]

\[ d_2 = \frac{\log \frac{S_k}{S_1} - \sigma^2 / 2(T - t)}{\sigma \sqrt{T - t}}. \]
References


