Abstract

The main objective of this thesis is to derive a finite element method for a parabolic stochastic partial differential equation, driven by either an additive or a multiplicative noise. Subsequently, to implement a finite element solver in Matlab in rectangular domains in $\mathbb{R}^2$. We discretize in space by using piecewise linear finite elements and use the backward Euler method for the time discretization. The finite element solver is applied in numerical experiments in an attempt to verify theoretical results on the convergence rate of the numerical method. As we do not obtain the predicted convergence rates, we conclude that the solver must be improved. In turn, we point out possible weaknesses in the numerical procedure.

Furthermore, we explore conditions guaranteeing existence and uniqueness of a mild solution and examine the consequences of these conditions for the parameters of the model. We also develop the connection between the model parameters and the regularity of the solution. The sensitivity of the solution to the model parameters is further emphasized by numerical experiments.

We identify and describe possible extensions to this work. One involves the implementation of the finite element solver in a general domain in $\mathbb{R}^2$. This requires the estimation of the eigensystem of the Laplacian operator in the domain in question. Another extension is to perform further numerical experiments, concentrating on the model with multiplicative noise. However, the shortcomings of the numerical approach must first be addressed. This includes the assessment of the magnitude of the quadrature error. Also, the sampling error should be reduced by decreasing the run time of the program code.

Acknowledgements

I would like to use this opportunity to express my gratitude to some people who have been of help and importance to me during my studies at Chalmers University of Technology.

- First, I would like to thank my thesis supervisor, Professor Stig Larsson, for his assistance and advice during my studies in Chalmers. He suggested the topic of this thesis to me and has provided me with valuable suggestions and positive criticism throughout the project.
- Many thanks go also to Professor Ivar Gustavsson, Director of the International Master's Programme in Engineering Mathematics.
- Special thanks go to my newborn daughter, Sigurveig Einarsdóttir, for coming into this world on time and without complications. Also, for giving her mother time to complete her thesis, and of course for being so wonderful in general.
- I would also like to thank my parents for all their help. Especially for looking after their granddaughter for two weeks in April, allowing me more time to work on my thesis.
- I warmly thank the friends that I have made during my studies, who have enriched my time here in Sweden.
- Last but not least, I thank my husband Einar for his never ending love and support.

Table of Contents

1 Introduction **1**

2 Theoretical Framework **3**

2.1 Notation, definitions and other results 3

2.2 Form and meaning of the model made more precise 5

2.3 The mild solution 7

3 A Finite Element Solver **11**

3.1 An alternative version of the mathematical model 11

3.2 The finite element method 12

3.2.1 Variational formulation 12

3.2.2 Partitioning of the time and space domains 13

3.2.3 The numerical method 14

3.3 Implementation of the method 15

3.3.1 Approximating $W^n - W^{n-1}$ using vertex quadrature 15

3.3.2 Eigenvalues and eigenfunctions of the Laplacian in a domain $\mathcal{D}$ 16

3.3.3 Implementation of the solver 22

3.3.4 Implementation details 24

4 The Numerical Experiments and Expected Results **25**

4.1 The notion of convergence 26

4.1.1 Strong and weak convergence 26

4.1.2 Computation of strong and weak convergence 26

4.2 Setup of the numerical experiments 29

4.2.1 The model for the numerical experiments 29

4.2.2 An outline of the numerical experiments 29

4.3 Expected results 30

4.3.1 Expected strong convergence rates 31

4.3.2 Expected weak convergence rates 32

5 Results of Numerical Experiments **33**
5.1 Individual FEM solutions and noise terms 33
5.1.1 Individual results with $\sigma = I$ 34
5.1.2 Individual results with $\sigma = \|x\|$ 36
5.1.3 Individual results with $\sigma = u$ 37
5.2 Results of the experiments 39
5.2.1 Main numerical experiment 39
5.2.2 Experiment with $\sigma = I$ on a finer mesh 42
5.2.3 Experiment with $\sigma = \|x\|$ 43
5.3 Discussion of the results 44

6 Conclusions **45**
6.1 Possible extensions 46

Bibliography **48**

Chapter 1

Introduction

A stochastic partial differential equation (spde) is a partial differential equation containing a random term. Such equations are a basic tool for modelling systems where noise is important and arise in a variety of situations within many different fields of research. Spde's are, for example, used to model turbulence and pattern formation and to predict trends in the stock market or in the weather. They are also used for biological modelling and within the field of medicine. Although the properties of spde's have been thoroughly studied, numerical approximations of such equations have received less attention, which makes it an interesting topic for further research.

In this thesis, we study a special case of the stochastic parabolic partial differential equation, namely the stochastic heat equation,

$$du - \Delta u \, dt = \sigma \, dW, \quad \text{for } 0 < t \leq T, \quad \text{with } u(0) = u^0. \tag{1.1}$$

Here, $u(t)$ is a random process that takes its values in $L_2(\mathcal{D})$, where $\mathcal{D}$ is a bounded domain in $\mathbf{R}^d$, $d = 1, 2, 3$, with a sufficiently smooth boundary, $\Gamma = \partial\mathcal{D}$. In particular, $u^0 \in L_2(\mathcal{D})$. We have that Δ is the Laplacian operator subject to homogeneous Dirichlet boundary conditions. Furthermore, σ is an operator valued function defined on $L_2(\mathcal{D})$ and $W(t)$ is a Wiener process defined on a filtered probability space, $(\Omega, \mathcal{F}, \mathbf{P}, \{\mathcal{F}_t\}_{t>0})$, with covariance operator $Q : L_2(\mathcal{D}) \to L_2(\mathcal{D})$.

Our principal aim is to derive and implement a finite element method to solve the spde in (1.1) in a rectangular domain in $\mathbb{R}^2$. This requires us to examine the theoretical details of the model, including conditions guaranteeing existence and uniqueness of a mild solution. Another important aim is to use the finite element solver to test theoretical error estimates. Last, but not least, we use numerical experiments and the finite element solver to increase our understanding of the mathematical model, the noise term, $\sigma \, dW$, in particular. In this thesis, we build mainly on the work done by Yubin Yan and Stig Larsson at Chalmers University of Technology, which is described in Yan [10]. The extensions made here concern mainly numerical work and include the derivation and implementation of the finite element solver and the testing of error estimates put forward in Yan [11].

In essence, we achieve our main aim, i.e., to implement a finite element solver for the mathematical model in (1.1). However, we do not reach an acceptable conclusion about the convergence rate of the numerical method by using numerical experiments. Thus, we can not

verify the error estimates in Yan [11]. Even though this is a disappointment, there are many positives that we take from our work here. We have gained insight into the mathematical model, which will be valuable when it comes to planning further numerical experiments. Also, we have been able to identify some weaknesses in the numerical approach, which we believe to explain why we cannot reach a conclusion about the convergence rate of the numerical method. An extension of this thesis would be to look carefully into these weaknesses and make amends for them. Furthermore, we have pinpointed some new directions that this work could be taken into and extended, such as implementing the solver in a general domain in $\mathbb{R}^2$. Last, but not least, we have written software, the finite element solver, which will be useful in further work on this subject.

The layout of the thesis is as follows. We start, in Chapter 2, by exploring the theoretical framework within which we will be working. We specify the form of individual terms of the mathematical model, the noise term in particular, and examine conditions of existence and uniqueness of a solution to the model. In Chapter 3, we derive a finite element method for the mathematical model. Also, we discuss some issues concerning the implementation of the finite element solver, mainly focusing on the computation of the noise term of the spde. Furthermore, we discuss the implementation of the solver in a general domain, $\mathcal{D}$, in $\mathbb{R}^2$, which includes the estimation of the eigensystem of the Laplacian operator in $\mathcal{D}$. Our discussion on the numerical experiments and their results is divided into two chapters, Chapter 4 and Chapter 5. In Chapter 4, we present the setup of the numerical experiments and their expected results. Then, in Chapter 5, we display and discuss the results of the experiments. Finally, in Chapter 6, we summarize the main conclusions of the thesis and discuss some possible extensions to this work.

Chapter 2

Theoretical Framework

In the previous chapter, we introduced the mathematical model which we will be working with in this thesis. Presently, the model is in a general form which must be narrowed down in order to give a clear meaning to the model. In particular, the form of individual terms of the model must be specified and some assumptions must be made in order to establish the existence of a unique solution to the model. The principal aim of this chapter is thus to further develop the mathematical model. As a result, we obtain a model, coupled with certain restrictions on its parameters, for which we will later be able to develop a numerical method.

We begin by putting forward some definitions, notation and other results that we will need later on. This is done in Section 2.1.

In Section 2.2, we explain the form and clarify the meaning of individual terms in the spde (1.1). Here, the noise term of the spde, $\sigma\, dW$, is of particular importance since it is essential to know what type of noise we are working with and how its properties are affected by the parameters of the model.

In Chapter 3, we use the variational form of the spde to develop a numerical method for the model. When it comes to nailing down the concept of a solution, however, the mild solution to the spde is more convenient. In Section 2.3, we present the mild solution to the spde and establish sufficient conditions for its existence and uniqueness. These conditions will, in turn, result in restrictions on the parameters of the model. Furthermore, we examine the regularity of the mild solution and the relationship between the regularity of the solution and the regularity of the noise term of the spde.

2.1 Notation, definitions and other results

Below, we present definitions and other results that are needed in the remainder of the chapter. We also introduce notation that is used throughout the thesis.

Notation and definitions of some spaces

For convenience, we let $H = L_2(\mathcal{D})$, with inner product defined by $(u, v) = \int_{\mathcal{D}} u\, v\, dx$ and corresponding norm $\|\cdot\| = (\cdot, \cdot)^{1/2}$. Furthermore, we let $\mathcal{A}$ be the operator $-\Delta$ with Dirichlet

boundary conditions, i.e., $\mathcal{A} = -\Delta$ with domain $D(\mathcal{A}) = H_0^1 \cap H^2$, where the spaces H^2 and H_0^1 are according to Definition 2.1.1 below.

Definition 2.1.1. *We define $H^k = H^k(\mathcal{D})$ to be the space of all functions whose weak partial derivatives of order $\leq k$ belong to L_2, i.e.,*

$$H^k = H^k(\mathcal{D}) = \{v \in L_2 : D^\alpha v \in L_2,\ |\alpha| \leq k\},$$

where $\alpha = (\alpha_1, ..., \alpha_d)$ is a vector of non-negative integers and $D^\alpha v = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} ... \partial x_d^{\alpha_d}}$, with $|\alpha| = \sum_{i=1}^d \alpha_i$. Furthermore, we define $H_0^1 = H_0^1(\mathcal{D})$ as

$$H_0^1 = H_0^1(\mathcal{D}) = \{v \in H^1 : v = 0 \text{ on } \Gamma\}.$$

The space H^k has the inner product $(v, w)_k = \sum_{|\alpha| \leq k} \int_{\mathcal{D}} D^\alpha v D^\alpha w \, dx$ and the corresponding norm $\|v\|_k = (v, w)_k^{1/2}$.

We note that when $k = 0$, we have the space $H = L_2(\mathcal{D})$.

Definition 2.1.2. *We define the space $\dot{H}^s = \dot{H}^s(\mathcal{D}) = D(\mathcal{A}^{s/2})$, with norm $|v|_s = \|\mathcal{A}^{s/2} v\|$, for any s in $\mathbb{R}$.*

We have from Parseval's relation that $|v|_s^2 = \|\mathcal{A}^{s/2} v\|^2 = \sum_{j=1}^\infty \lambda_j^s \hat{v}_j^2$, where λ_j are eigenvalues of $\mathcal{A}$ and $\hat{v}_j = (v, \varphi_j)$, with φ_j an orthonormal basis of corresponding eigenfunctions.

Definition 2.1.3. *The triple $(\Omega, \mathcal{F}, \mathbf{P})$, where Ω is a set, $\mathcal{F} \subset \mathcal{P}(\Omega)$ is a σ-algebra and $\mathbf{P} : \mathcal{F} \to [0, 1]$ is a probability measure, is called a probability space. A filtered probability space, $(\Omega, \mathcal{F}, \mathbf{P}, \{\mathcal{F}_{t \geq 0}\})$, is a probability space equipped with an increasing family of σ-algebras, $\{\mathcal{F}_{t \geq 0}\}$, called a filtration.*

Definition 2.1.4. *For any Hilbert space, H, we define*

$$L_2(\Omega; H) = \left\{ v : \mathbf{E}\|v\|_H^2 = \int_\Omega \|v(\omega)\|_H^2 \, d\mathbf{P}(\omega) < \infty \right\}$$

with norm $\|v\|_{L_2(\Omega;H)} = \left(\mathbf{E}\|v\|_H^2\right)^{1/2}$.

Hilbert-Schmidt operators and spaces

In our work, we follow Da Prato and Zabczyk [1] and let σ be a Hilbert-Schmidt operator from $Q^{1/2}(H)$ to H. The space of all such operators is denoted by $\mathrm{HS}(Q^{1/2}(H), H)$. A Hilbert-Schmidt operator is defined as follows.

Definition 2.1.5. *We say that $\psi(s)$ is a Hilbert-Schmidt operator on H, or $\psi(s) \in HS(H, H)$, if*

$$\|\psi(s)\|_{HS} = \left(\sum_{j=1}^\infty \|\psi(s)\varphi_j\|^2 \right)^{1/2} < \infty,$$

where $\{\varphi_j\}$ is an arbitrary orthonormal basis in H.

Thus, for ψ to be a Hilbert-Schmidt operator from $Q^{1/2}(H)$ to H, we must have,

$$\|\psi(s)Q^{1/2}\|_{\mathrm{HS}} = \left(\sum_{j=1}^{\infty} \|\psi(s)Q^{1/2}\varphi_j\|^2\right)^{1/2} < \infty.$$

Itô isometry and Brownian motion

We now define an important property of stochastic processes, the socalled Itô isometry

Theorem 2.1.6. *If $\psi \in HS(Q^{1/2}(H), H)$, then the stochastic integral $\int_0^t \psi(s)dW(s)$ is well defined and we have the Itô isometry*

$$\mathbf{E}\left\|\int_0^t \psi(s)\, dW(s)\right\|^2 = \int_0^t \|\mathbf{E}\psi(s)Q^{1/2}\|_{HS}^2 ds,$$

where $\mathbf{E}$ *stands for expectation.*

We will need to formalize the concept of Brownian motion.

Definition 2.1.7. *A random variable B(t) that depends continuously on $t \in [0,T]$, is called a standard Brownian motion over $[0,T]$ if it satisfies the following three conditions:*

1. $\mathbf{P}(B(0) = 0) = 1$, *where* $\mathbf{P}$ *stands for probability.*
2. *For $0 \le s < t < T$ we have $B(t)-B(s) \sim \sqrt{(t-s)}N(0,1)$, where N(0,1) is the standard normal distribution.*
3. *For $0 \le s < t < u < v < T$ the increments $B(t)-B(s)$ and $B(v)-B(u)$ are independent.*

Other results

In the proof of the regularity of the mild solution, we need some results that are related to $E(t) = e^{-\mathcal{A}t}$ being an analytic semigroup on H. These results are presented as a lemma and the proof can be found in Thomée [9].

Lemma 2.1.8. *For any μ, $\nu \in \mathbb{R}$ and $l \ge 0$, there exists $C > 0$ such that*

(i) $|D_t^l E(t)v|_\nu \le Ct^{-(\nu-\mu)/2-l}|v|_\mu$, *for $t > 0$, $2l + \nu \ge \mu$,*

(ii) $\int_0^t s^\mu |D_t^l E(s)v|_\nu^2\, ds \le C|v|^2_{2l+\nu-\mu-1}$, *for $t \ge 0$, $\mu \ge 0$.*

2.2 Form and meaning of the model made more precise

In this section, we seek to make the form and meaning of the terms u, σ, and W in the mathematical model more precise.

The solution to the spde, u

We start with the term u. We have that u is the solution to the spde (1.1). It is a H-valued random process, which takes the form

I. $u : [0,T] \times \Omega \to H$, or $(t,\omega) \mapsto u(t,\omega)$ ($u(t)$ for short), with $u(t) : \mathcal{D} \to \mathbb{R}$.

The Wiener process, W

W is a Wiener process on a filtered probability space, $(\Omega, \mathcal{F}, \mathbf{P}, \{\mathcal{F}\}_{t\geq 0})$, with covariance operator $Q : H \to H$. It takes the form

II. $W : [0,T] \times \Omega \to V$, or $(t,\omega) \mapsto W(t,\omega)$ ($W(t)$ for short), with $W(t) : \mathcal{D} \to \mathbb{R}$. The space V is determined by the properties of the Wiener process.

In this thesis, we follow Da Prato and Zabczyk [1] and Yan [10] and write the Wiener process in terms of its Fourier series,

$$W(t) = \sum_{j=1}^{\infty} \gamma_j^{1/2} e_j \beta_j(t). \tag{2.1}$$

Here, $\beta_j(t)$ is a sequence of independent identically distributed Brownian motions and $\{\gamma_j, e_j\}$ is the eigensystem for Q. The operator Q is self-adjoint, positive definite, bounded and linear. Moreover, Q is defined such that it has the same eigenfunctions as $\mathcal{A}$.* The relationship between the eigenvalues of Q and $\mathcal{A}$ is given by the equation

$$\gamma_j = \lambda_j^{-\alpha}, \tag{2.2}$$

where $\alpha \in \mathbb{R}$.

We concentrate on the following two types of Wiener processes:

1. $W(t)$ is a so-called nuclear Wiener process, which follows from the assumption that the covariance operator Q is of trace class, i.e., $\mathrm{Tr}(Q) < \infty$. Here, $V = H$ (see II), since we have

$$\mathbf{E}\|W(t)\|^2 = \mathbf{E}\Big\| \sum_{j=1}^{\infty} \gamma_j^{1/2} \beta_j(t) e_j \Big\|^2 = \sum_{j=1}^{\infty} \gamma_j \mathbf{E}\beta_j(t)^2 = t \sum_{j=1}^{\infty} \gamma_j = t\,\mathrm{Tr}(Q) < \infty.$$

2. $W(t)$ has identity covariance, i.e., Q is the identity operator. Thus $\mathrm{Tr}(Q) = \infty$. Here, $W(t)$ does not take its values in H, i.e., $V \neq H$. The noise term $\sigma\, dW$ is white in space and time.

In case 2, where $Q = I$, the eigenvalues of Q are all 1 and from (2.2) we see that we must have $\alpha = 0$. In case 1, we get

$$\mathrm{Tr}(Q) = \sum_{j=1}^{\infty} \gamma_j = \sum_{j=1}^{\infty} \lambda_j^{-\alpha} \leq \{\lambda_j \leq C j^{2/d}, \text{ see (3.17).}\} \leq C \sum_{j=1}^{\infty} j^{-2\alpha/d}, \tag{2.3}$$

which gives us $\mathrm{Tr}(Q) < \infty$ if $\alpha > d/2$.

*Here, $\mathcal{A} = -\Delta$.

The operator σ

The noise term in the spde is $\sigma\, dW$, where σ is a Hilbert-Schmidt operator from $Q^{1/2}(H)$ to H, not necessarily linear. The form of σ determines the type of noise of the spde. If σ is in the form

III. $\sigma : [0,T]\times H \to \mathrm{HS}\big(Q^{1/2}(H), H\big)$, or $(t,f) \mapsto \sigma(t;f)$ ($\sigma(t)$ for short), with $\sigma(t) : \mathcal{D} \to \mathbb{R}$,

the spde has an *additive* noise term. However, we say that the noise term, $\sigma(u(t))\, dW(t)$, is *multiplicative* if σ is in the following form:

IV. $\sigma : [0,T] \times \Omega \times H \to \mathrm{HS}\big(Q^{1/2}(H), H\big)$, or $(t,\omega,f) \mapsto \sigma(t,\omega;f)$. In particular, we have $u(t) \in H$, so we consider $(t,\omega,u(t)) \mapsto \sigma(t,\omega;u(t))$ ($\sigma(u(t))$ for short), with $\sigma(u(t)) : \mathcal{D} \to \mathbb{R}$.

In Section 2.3, we impose some conditions on the operator σ, which will guarantee the existence and uniqueness of a mild solution to the spde.

2.3 The mild solution

Here, we present the mild solution to the spde in (1.1), together with conditions that guarantee its existence and uniqueness. These conditions result in restrictions on the parameters of the model, which we will investigate further. Also, we examine the regularity of the mild solution and the relationship between the regularity of the solution and the regularity of the noise term of the spde.

Sufficient conditions for existence and uniqueness of a mild solution

Following Da Prato and Zabczyk [1], we assume that σ fulfills the following Lipschitz and growth conditions. There exists a constant C such that,

(i) $\|\big(\sigma(f) - \sigma(g)\big)Q^{1/2}\|_{\mathrm{HS}} \le C\|f-g\|, \quad$ for $f, g \in H$.

(ii) $\|\sigma(f)Q^{1/2}\|_{\mathrm{HS}} \le C\|f(t)\|, \quad$ for $f \in H$.

Then we have that there exists a unique mild solution, $u \in \big([0,T]; L_2(\Omega, H)\big)$, to (1.1):

$$u(t) = E(t)u_0 + \int_0^t E(t-s)\sigma(u(s))\, dW(s), \tag{2.4}$$

where $E(t) = e^{-t\mathcal{A}}$ is the analytic semigroup on H generated by $-\mathcal{A}$. We have, moreover, that

$$\sup_{t\in[0,T]} \mathbf{E}\|u(t)\|^2 \le C(1 + \mathbf{E}\|u_0\|^2).$$

We note, that in the case that $\mathrm{Tr}(Q) = \infty$, the identity mapping $\sigma = I$ does not fulfill (ii), i.e.,

$$\|\sigma Q^{1/2}\|_{\mathrm{HS}} = \left(\sum_{j=1}^{\infty} \|Q^{1/2}\varphi_j\|^2\right)^{1/2} \le \left(\sum_{j=1}^{\infty} \|Q^{1/2}\|^2\right)^{1/2} = \infty.$$

Therefore, we introduce a modified version of condition (ii),

(ii') $\|\mathcal{A}^{(\beta-1)/2}\sigma(f)Q^{1/2}\|_{\text{HS}} \leq C\|f\|, \quad$ for some $\beta \in [0,\infty)$, $f \in H$.

Remark 2.3.1. *We can see that the case (ii) corresponds to $\beta = 1$ in (ii'). Also, that if $\sigma = I$, (ii') reduces to $\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{HS} \leq C$.*

Consequences for the parameters of the Wiener process

The above conditions (i), (ii) and (ii') are used in proofs of the existence of a mild solution and proofs of error estimates and the regularity of the mild solution. Here, however, our main concern is how these conditions affect the parameters of the model. As of yet, we have not been able to fully explore the consequences of the conditions. However, we have derived restrictions on the parameters α and β, imposed by (ii'), in the case where $\sigma = I$.

We let $\sigma = I$. Then, the condition (ii') reduces to $\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{\text{HS}} \leq C$. We get

$$\begin{aligned}
\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{\text{HS}} &= \sum_{j=1}^{\infty} \|\mathcal{A}^{(\beta-1)/2}Q^{1/2}e_j\|^2 \\
&= \{\mathcal{A}e_j = \lambda_j e_j \text{ and } Qe_j = \gamma_j e_j\} \\
&= \sum_{j=1}^{\infty} \|\lambda_j^{(\beta-1)/2}\gamma_j^{1/2}e_j\|^2 = \sum_{j=1}^{\infty} \lambda_j^{\beta-1}\gamma_j\|e_j\|^2 \\
&= \{\gamma_j = \lambda_j^{-\alpha}\} \\
&= \sum_{j=1}^{\infty} \lambda_j^{\beta-1-\alpha}\|e_j\|^2 = \sum_{j=1}^{\infty} \lambda_j^{\beta-1-\alpha} \\
&\leq \{\lambda_l \leq Cl^{2/d}, \text{ see (3.17)}\} \leq C\sum_{j=1}^{\infty} j^{\frac{2(\beta-1-\alpha)}{d}}.
\end{aligned} \tag{2.5}$$

For this to be finite, we need $\frac{2(\beta-1-\alpha)}{d} < -1$. This means that in order for (ii') to be valid, β must be such that $\beta < \alpha - \frac{d}{2} + 1$.

As previously mentioned, we concentrate on spde's with nuclear Wiener processes and Wiener processes with identity covariance. When $W(t)$ is nuclear, we have shown in (2.3) that we must have $\alpha > d/2$. Thus, when $W(t)$ is nuclear and $\sigma = I$, we have that for (ii') to be valid, we must have $\beta \in [0, \alpha + 1 - d/2)$, with $\alpha > d/2$.

For space-time white noise, i.e., when $Q = I$, we have that $\alpha = 0$ and thus for (ii') to be valid we must have that $\beta \in [0, 1 - d/2)$. For $\beta \in [0,\infty)$ this is only possible when $d = 1$, which gives $\beta \in [0, 1/2)$.

When $\sigma \neq I$, the condition (ii') is not trivial to evaluate, in general. The computation above breaks down since we are not able to simplify the term $\mathcal{A}^{(\beta-1)/2}\sigma(u(t))e_j$. This is a problem, moreover, when it comes to determining whether a particular operator σ fulfills the condition. Here, we will have to be content by saying that the conditions (i) and (ii') are quite strong and that σ must be a very well behaved operator in order to fulfill them.

Regularity of the mild solution

Theorem 2.1 in Yan [11] expresses the regularity of the mild solution. Below, we reproduce this theorem, together with its proof.

Theorem 2.3.2. *Assume that σ satisfies (i) and (ii'). Let $u(t)$ be the mild solution (2.4) of (1.1). Then we have, for $u_0 \in L_2(\Omega, \dot{H}^\beta)$,*

$$\|u(t)\|_{L_2(\Omega;\dot{H}^\beta)} \leq C \left(\|u_0\|_{L_2(\Omega;\dot{H}^\beta)} + \sup_{0\leq s\leq t} \|u(s)\|_{L_2(\Omega;H)} \right). \tag{2.6}$$

Proof. For any $\beta \geq 0$, using the stability of $E(t)$, see Lemma 2.1.8, and Itô isometry, we get

$$\begin{aligned}
\mathbf{E}|u(t)|_\beta^2 &= \mathbf{E} \left| E(t)u_0 + \int_0^t E(t-s)\sigma(u(s))\, dW(s) \right|_\beta^2 \\
&\leq 2\mathbf{E}\, |E(t)u_0|_\beta^2 + 2\mathbf{E} \left| \int_0^t E(t-s)\sigma(u(s))\, dW(s) \right|_\beta^2 \\
&= \{\text{using the definition of the } |\cdot|_s \text{ norm}\} \\
&= 2\mathbf{E}\, |E(t)u_0|_\beta^2 + 2\mathbf{E} \left\| \int_0^t \mathcal{A}^{\beta/2} E(t-s)\sigma(u(s))\, dW(s) \right\|^2 \\
&\leq \{\text{using Lemma 2.1.8 (ii) with } l=0 \text{ and } \mu=\nu \text{ and Itô isometry}\} \\
&\leq 2\mathbf{E}\, |u_0|_\beta^2 + 2\mathbf{E} \int_0^t \|\mathcal{A}^{\beta/2} E(t-s)\sigma(u(s))Q^{1/2}\|_{\mathrm{HS}}^2\, ds \\
&= 2\mathbf{E}\, |u_0|_\beta^2 + 2\mathbf{E} \int_0^t \|\mathcal{A}^{1/2} E(t-s)\mathcal{A}^{(\beta-1)/2}\sigma(u(s))Q^{1/2}\|_{\mathrm{HS}}^2\, ds.
\end{aligned}$$

Further, we have

$$\begin{aligned}
&\mathbf{E} \int_0^t \|\mathcal{A}^{\beta/2} E(t-s)\sigma(u(s))Q^{1/2}\|_{\mathrm{HS}}^2\, ds \leq \mathbf{E} \int_0^t \|\mathcal{A}^{1/2} E(t-s)\|^2 \|\mathcal{A}^{(\beta-1)/2}\sigma(u(s))Q^{1/2}\|_{\mathrm{HS}}^2\, ds \\
&\leq \{\text{using (ii') }\} \leq C \left(\int_0^t \|\mathcal{A}^{1/2} E(t-s)\|^2\, ds \right) \sup_{0\leq s\leq t} \mathbf{E}\|u(s)\|^2 \\
&\leq \{\text{using Lemma 2.1.8 (ii) with } l=0,\ \mu=0,\ \nu=1 \text{ and } v=1\} \leq C \sup_{0\leq s\leq t} \mathbf{E}\|u(s)\|^2.
\end{aligned}$$

We thus have

$$\mathbf{E}|u(t)|_\beta^2 \leq C \left(\mathbf{E}|u_0|_\beta^2 + \sup_{0\leq s\leq t} \mathbf{E}\|u(s)\|^2 \right).$$

Noting that

$$\left(\sup_{0\leq s\leq t} \mathbf{E}\|u(s)\|^2 \right)^{1/2} \leq \sup_{0\leq s\leq t} \left(\mathbf{E}\|u(s)\|^2 \right)^{1/2} = \sup_{0\leq s\leq t} \|u(s)\|_{L_2(\Omega;H)}$$

we get the inequality (2.6). □

The regularity estimate in (2.6) indicates that if the initial value, u_0, is in $L_2(\Omega, \dot{H}^\beta)$, the mild solution is also in $L_2(\Omega, \dot{H}^\beta)$. Thus, the number of possible derivatives of u is at least β. However, we are more interested in how the regularity of the mild solution is connected to the regularity of the noise term of the spde.

It is relatively simple to show that $W(t)$ (and thus also $dW(t)$) is in $L_2(\Omega, \dot{H}^{\beta-1})$. We have, using Parseval's relation and (2.1), (2.2) and (3.17),

$$\begin{aligned}
\mathbf{E}|W(t)|_{\beta-1}^2 &= \mathbf{E}\|\mathcal{A}^{(\beta-1)/2}W(t)\|^2 = \mathbf{E}\sum_{l=1}^{\infty}\big(\mathcal{A}^{(\beta-1)/2}W(t), e_l\big)^2 \\
&= \mathbf{E}\sum_{l=1}^{\infty}\big(\mathcal{A}^{(\beta-1)/2}\sum_{j=1}^{\infty}\gamma_j^{1/2}\beta_j(t)e_j, e_l\big)^2 = \mathbf{E}\sum_{l=1}^{\infty}\gamma_l\beta_l(t)^2\big(\mathcal{A}^{(\beta-1)/2}e_l, e_l\big)^2 \\
&= \mathbf{E}\sum_{l=1}^{\infty}\lambda_l^{\beta-1}\gamma_l\beta_l(t)^2 = \mathbf{E}\sum_{l=1}^{\infty}\lambda_l^{\beta-1-\alpha}\beta_l(t)^2 \\
&= \sum_{l=1}^{\infty}\lambda_l^{\beta-1-\alpha}\mathbf{E}\beta_l(t)^2 = t\sum_{l=1}^{\infty}\lambda_l^{\beta-1-\alpha} \leq t\sum_{l=1}^{\infty}l^{\frac{2(\beta-1-\alpha)}{d}} < \infty,
\end{aligned}$$

for $\beta \in [0, \alpha + 1 - d/2)$. When $\sigma = I$, we thus have that the noise term, σdW, is in $L_2(\Omega, \dot{H}^{\beta-1})$. However, we are not able to draw any such conclusions when $\sigma \neq I$.

When $\sigma = I$, we thus have that the relationship between the regularity of the mild solution and the noise term of the spde is that the solution has one more derivative than the noise term, i.e., the solution is smoother than the noise term.

In terms of the parameters α and β, we have that α determines the upper limit of the interval to which β is restricted. Although we do not have any equation that precisely describes the relationship between α and β, the results displayed in Section 5.1 show that dW becomes more regular as α increases. This indicates that a higher value of α means a more regular noise term and thus a more regular solution to the spde, i.e., a higher value of α results in a higher value of β.

When $\sigma \neq I$, we must rely on the information provided by numerical experiments.

Remark 2.3.3. *$dW(t) \in L_2(\Omega, \dot{H}^{\beta-1})$ with $\beta \in [0, \alpha + 1 - d/2)$ implies, in particular, that dW is in $L_2(\Omega, \dot{H}^{-1})$. For $\sigma = I$, we thus have that the noise term, $\sigma\, dW$, is in $L_2(\Omega, \dot{H}^{-1})$. This we need for the finite element method in order to be able to compute the inner product of $\sigma\, dW$ and χ, where $\chi \in S_h \subset H_0^1$. As before, we cannot make such conclusions about σ in a general form.*

Chapter 3

A Finite Element Solver

Here, we present a finite element method for our mathematical model and discuss issues concerning the implementation of the method. The chapter is computationally oriented in the sense that we derive and implement a numerical method without paying much attention to existence and uniqueness of the solution to the spde, which was dealt with in Chapter 2. The result is a general finite element solver for the spde. Before the solver is used in practice, however, the validity of different versions of the model must be verified.

We begin the chapter by presenting an alternative version of the mathematical model in (1.1). The model in Section 3.1 is equivalent to the model in (1.1), except that we add a source term, $f\,dt$, to the right hand side of the equation. However, the model presented here is more convenient when developing the finite element method, while the former version is better suited for further theoretical work.

In Section 3.2, we derive the variational form of the spde and then present a finite element method for the spde in a convex or polygonal domain in $\mathbb{R}^2$. We employ the backward Euler method for the time stepping.

In the final section of the chapter, we discuss issues in connection with the implementation of the finite element solver, mainly focusing on the computation of the noise term of the spde. We go through the approximation of the noise term by quadrature and outline the construction of the finite element solver. Here, we find it beneficial in terms of efficiency to distinguish between cases when the spde has additive or multiplicative noise terms. Since we represent the Wiener process in the spde in terms of its Fourier series, we need to compute the eigensystem of the Laplacian operator on $\mathcal{D}$. Thus, we devote a part of Section 3.3 to the discussion on the computation or estimation of these eigenvalues and eigenfunctions in different spatial domains. Finally, we go through some technical details concerning the implementation of the solver, such as the computation language and speed of execution.

3.1 An alternative version of the mathematical model

Below, we will derive a finite element solver for the spde in (1.1) with the added source term $f\,dt$, where $f(t) \in L_2(\mathcal{D})$. This model can formally be written as

$$\frac{du}{dt} - \Delta u = f + \sigma \dot{W}, \quad \text{for } 0 < t \leq T, \quad \text{with } u = u^0 \text{ if } t = 0, \tag{3.1}$$

with $\dot{W} = \frac{dW}{dt}$. Since Wiener processes are not differentiable with respect to time, $\dot{W}$ does not exist as a function of t in the usual sense. Equation (3.1) must thus be interpreted in its integral form.

We find it convenient to consider the terms u, σ and W to be in different forms to what was described in I-IV, i.e., all to be the following real valued functions,

I'. $u : [0,T] \times \Omega \times \mathcal{D} \to \mathbb{R}$, or $(t,\omega,x) \mapsto u(t,\omega,x)$ ($u(t,x)$ for short).

II'. $W : [0,T] \times \Omega \times \mathcal{D} \to \mathbb{R}$, or $(t,\omega,x) \mapsto W(t,\omega,x)$ ($W(t,x)$ for short).

III'. **Additive noise :** $\sigma : [0,T] \times \mathcal{D} \times H \to \mathbb{R}$, or $(t,x,f) \mapsto \sigma(t,x;f)$ ($\sigma(t,x)$ for short).

IV'. **Multiplicative noise:** $\sigma : [0,T] \times \Omega \times \mathcal{D} \times H \to \mathbb{R}$, or $(t,\omega,x,f) \mapsto \sigma(t,\omega,x;f)$. In particular, $u(t) \in H$ so we have $(t,\omega,x,u(t)) \mapsto \sigma(t,\omega,x;u(t))$ ($\sigma(u(t,x))$ for short).

In addition, the source term, f, is in the following form:

V'. $f : [0,T] \times \mathcal{D} \to \mathbb{R}$, or $(t,x) \mapsto f(t,x)$.

We find that, applying I'-V', the model in equation (3.1) becomes

$$\frac{\partial u(t,x)}{\partial t} - \Delta u(t,x) = f(t,x) + \sigma \dot{W}(t,x), \quad \text{for } 0 < t \leq T, \quad \text{with } u(0,x) = u^0, \tag{3.2}$$

where $\sigma = \sigma(t,x)$ or $\sigma = \sigma(t,x,u(t,x))$. As before, we write the Wiener process, W, in terms of its Fourier series,

$$W(t,x) = \sum_{j=1}^{\infty} \gamma_j^{1/2} e_j(x) \beta_j(t), \tag{3.3}$$

where $\gamma_j > 0$, e_j and β_j are as described in Section 2.2. This is the model that we use to derive the finite element method.

3.2 The finite element method

We will now present the finite element method for the mathematical model in (3.2). We start by deriving the variational formulation, then the partition of the time and space domains and finally we apply the backward Euler method to derive the numerical method.

3.2.1 Variational formulation

In order to derive the variational formulation of (3.2), we multiply the equation with a test function $v \in H_0^1 = \{v : (\|v\|_{L_2}^2 + \|\nabla v\|_{L_2}^2)^{\frac{1}{2}} < \infty,\ v = 0 \text{ on } \Gamma\}$ and integrate over $\mathcal{D} \times I$, $I = [0,T]$, to obtain

$$\begin{aligned} &\int_I \int_{\mathcal{D}} \frac{\partial}{\partial t} u(t,x) v(x)\, dx\, dt + \int_I \int_{\mathcal{D}} \nabla u(t,x) \nabla v(x)\, dx\, dt \\ &= \int_I \int_{\mathcal{D}} f(t,x) v(x)\, dx\, dt + \int_I \int_{\mathcal{D}} v(x) \sigma(t,x,u(t,x)) \dot{W}(t,x)\, dx\, dt, \quad \forall v \in H_0^1. \end{aligned} \tag{3.4}$$

If we let $W(t,x)$ be as in equation (2.1) above, the stochastic integral in (3.4) is defined as

$$\sum_{j=1}^{\infty} \gamma_j^{1/2} \int_I \int_{\mathcal{D}} v(x)\sigma(t,x,u(t,x))e_j(x)\,dx\,d\beta_j(t).$$

Applying the above definition, the variational formulation of the problem will be as follows. We seek $u \in H_0^1 \times I$ such that

$$\begin{aligned}\int_I \int_{\mathcal{D}} \frac{\partial}{\partial t} u(t,x)v(x)\,dx\,dt + \int_I \int_{\mathcal{D}} \nabla u(t,x)\nabla v(x)\,dx\,dt = \int_I \int_{\mathcal{D}} f(t,x)v(x)\,dx\,dt \\ + \sum_{j=1}^{\infty} \gamma_j^{1/2} \int_I \int_{\mathcal{D}} v(x)\sigma(t,x,u(t,x))e_j(x)\,dx\,d\beta_j(t), \quad \forall v \in H_0^1.\end{aligned} \tag{3.5}$$

3.2.2 Partitioning of the time and space domains

The discretization of the spde is based on a partitioning of the time and space domains. Here, we restrict the spatial domain to be a convex or polygonal domain in $\mathbb{R}^2$.

The time partition is obtained by dividing the time interval $I = [0,T]$ into N subintervals

$$0 = t_0 < t_1 < \cdots < t_N = T.$$

We denote $I_n = (t_{n-1}, t_n)$ and let $k_n = t_n - t_{n-1}$ and $k = \max_{0\le n\le N} k_n$.

When partitioning the spatial domain, we start by assuming that $\mathcal{D}$ is a polygonal domain in $\mathbb{R}^2$ and let $\mathcal{T}_h = \{K\}$ be a subdivision of $\mathcal{D}$ into closed triangles, which are such that a vertex of a triangle cannot lie on the edge of another triangle. $\mathcal{T}_h$ is then called a triangulation of $\mathcal{D}$ and we have $\bar{\mathcal{D}} = \bigcup_{K\in\mathcal{T}_h}$. Furthermore, the interior vertices of the triangulation are denoted as $\{P_j\}_{j=1}^{M_h}$, where M_h is the total number of vertices. The size of a triangle $K \in \mathcal{T}_h$ is measured by the length of its largest side (diameter of K), denoted h_K, and we let $h = \max_{K\in\mathcal{T}_h} h_K$.

We note that if the boundary Γ of $\mathcal{D}$ is not polygonal, a triangulation of the above type will not fit $\mathcal{D}$ exactly. However, if $\mathcal{D}$ is convex it is possible to triangulate the domain such that the union $\mathcal{D}_h$ of the triangles still approximates $\mathcal{D}$, i.e. such that the set $\mathcal{D}\backslash\mathcal{D}_h$ of points in $\mathcal{D}$ not covered by the triangulation has a width of order $O(h^2)$. Nothing is lost by this approximation when the finite element space consists of piecewise linear functions, but for piecewise polynomials of a higher degree the situation is more problematic. (See [7]).

We now define the finite element space S_h, consisting of continuous, piecewise linear functions on $\mathcal{T}_h$, vanishing on Γ,

$$S_h = \{v \in \mathcal{C}(\bar{\mathcal{D}}) : v \text{ is linear on each } K \in \mathcal{T}_h,\ v = 0 \text{ on } \Gamma\}.$$

In particular, we have that $S_h \subset H_0^1$. The requirement that a function $v \in S_h$ is linear on each triangle clearly implies that v is uniquely determined by its function values on the vertices of the triangulation. The set of tent functions $\{\Phi_i\}_{i=1}^{M_h} \subset S_h$,

$$\Phi_i(P_j) = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j, \end{cases} \tag{3.6}$$

forms a basis for S_h.

A finite element approximation u_h of the solution u to the spde in (3.2) is thus a function on $S_h \times I$.

3.2.3 The numerical method

We discretize in time by using the backward Euler, or implicit Euler, method. This is obtained by letting $u^n \in H_0^1$ be the approximation of $u(t,\cdot)$ at time $t = t_n$, and the time derivative $\frac{\partial}{\partial t}u(t,x)$ is approximated by $k_n^{-1}(u^n - u^{n-1})$. The time discretized problem is thus to find $u^n := u(t_n,\cdot) \in H_0^1$, such that

$$\int_{\mathcal{D}} u^n(x)v(x)\,dx + k_n \int_{\mathcal{D}} \nabla u^n(x)\nabla v(x)\,dx = \int_{\mathcal{D}} u^{n-1}(x)v(x)\,dx + k_n \int_{\mathcal{D}} f^n(x)v(x)\,dx$$
$$+ \sum_{j=1}^{J} \gamma_j^{1/2}(\beta_j(t_n) - \beta_j(t_{n-1})) \int_{\mathcal{D}} v(x)\sigma(t_{n-1}, x, u^{n-1})e_j(x)\,dx, \quad \forall v \in H_0^1,$$

where $f^n \in H$ is an approximation of $f(t,\cdot)$ at time $t = t_n$. Note, that in obtaining the time discretized problem we have truncated the sum in equation (3.5) to J terms. In Yan [11], it is proved that when $\sigma = I$, it is sufficient to take $J = M_h$. However, no proof is provided for other values of σ. In our numerical experiments we let $J = M_h$.

Further, discretizing in space, we seek the approximation in the finite element space S_h instead of in H_0^1. The fully discrete method is then to find $U^n := U(t_n,\cdot) \in S_h$ such that

$$\begin{aligned}
&\int_{\mathcal{D}} U^n(x)\chi(x)\,dx + k_n \int_{\mathcal{D}} \nabla U^n(x)\nabla\chi(x)\,dx \\
&\quad = \int_{\mathcal{D}} U^{n-1}(x)\chi(x)\,dx + k_n \int_{\mathcal{D}} f^n(x)\chi(x)\,dx \\
&\qquad + \sum_{j=1}^{J} \gamma_j^{1/2}(\beta_j(t_n) - \beta_j(t_{n-1})) \int_{\mathcal{D}} \chi(x)\sigma(t_{n-1}, x, U^{n-1})e_j(x)\,dx, \quad \forall \chi \in S_h.
\end{aligned} \tag{3.7}$$

Finally, we want to write equation (3.7) in matrix form. Since $U^n \in S_h$ we can write U^n in terms of the basis $\{\Phi_k\}_{k=1}^{M_h}$ in (3.6) as

$$U^n = \sum_{i=1}^{M_h} \xi_i^n \Phi_i(x), \tag{3.8}$$

where $\xi_i^n = \xi^n(P_i)$. Inserting (3.8) into (3.7) and taking $\chi = \Phi_k, k = 1,\ldots,M_h$, our problem can be stated as follows: Find the coefficients ξ_i^n, such that

$$\begin{aligned}
&\sum_{i=1}^{M_h} \xi_i^n \int_{\mathcal{D}} \Phi_i(x)\Phi_k(x)\,dx + k_n \sum_{i=1}^{M_h} \xi_i^n \int_{\mathcal{D}} \nabla\Phi_i(x)\nabla\Phi_k(x)\,dx \\
&\quad = \sum_{i=1}^{M_h} \xi_i^{n-1} \int_{\mathcal{D}} \Phi_i(x)\Phi_k(x)\,dx + k_n \int_{\mathcal{D}} f^n(x)\Phi_k(x)\,dx \\
&\qquad + \sum_{j=1}^{J} \gamma_j^{1/2}(\beta_j^n - \beta_j^{n-1}) \int_{\mathcal{D}} \Phi_k(x)\sigma\left(t_{n-1}, x, \sum_{i=1}^{M_h} \xi_i^{n-1}\Phi_i(x)\right) e_j(x)\,dx,
\end{aligned} \tag{3.9}$$

for $k = 1, \ldots, M_h$. We let φ_j denote the nodal values of the initial approximation u_h^0, $\varphi_j(0) = \varphi_j$, $j = 1, \ldots, M_h$. In matrix notation, equation (3.9) becomes

$$B\xi^n + k_n A\xi^n = B\xi^{n-1} + k_n F + W^n - W^{n-1}, \qquad \text{for } n \geq 1, \qquad \text{and } \xi^0 = \varphi, \tag{3.10}$$

where $B = (b_{ik})$ is the mass matrix with elements $b_{ik} = \int_{\mathcal{D}} \Phi_i \Phi_k \, dx$, $A = (a_{ik})$ is the stiffness matrix with elements $a_{ik} = \int_{\mathcal{D}} \nabla\Phi_i \nabla\Phi_k \, dx$, $F = (f_k)$ is the load vector with elements $f_k = \int_{\mathcal{D}} f^n \Phi_k \, dx$, ξ^n is the vector of unknowns ξ_i^n and the vector $W^n - W^{n-1} = (w_k)$ contains the elements

$$w_k = \sum_{j=1}^{J} \gamma_j^{1/2} (\beta_j^n - \beta_j^{n-1}) \int_{\mathcal{D}} \Phi_k(x) \sigma \left(t_{n-1}, x, \sum_{i=1}^{M_h} \xi_i^{n-1} \Phi_i(x) \right) e_j(x) \, dx. \tag{3.11}$$

3.3 Implementation of the method

In order to implement a solver for the spde in equation (3.2), the vectors and matrices in equation (3.10) must be computed or approximated. Since a large part of this implementation is standard in finite element work, such as the computation of the stiffness and mass matrices, we do not provide details here.

It is necessary, however, to cover the implementation of the vector $W^n - W^{n-1}$ more thoroughly. This includes a derivation of how the vector is approximated by using quadrature (Subsection 3.3.1). Also, we provide an algorithm which describes the assembly of the vector.

In Subsection 3.3.2, we discuss the computation of the eigenvalues and eigenvectors of the Laplacian operator on $\mathcal{D}$. Our numerical work in Chapters 4 and 5 is performed in rectangular domains in $\mathbb{R}^2$, for which we have analytical formulas for the eigensystem of the Laplacian operator. However, for most other domains, some estimation procedure must be used. Although we do not cover the estimation in detail, we outline a finite element approach for the estimation of the eigensystem of the Laplacian in general domains and highlight some of the problems that materialize in connection with it.

In Subsection 3.3.3, we outline the construction of the spde solver, and in Subsection 3.3.4 we present some technical details concerning our implementation of the solver.

3.3.1 Approximating $W^n - W^{n-1}$ using vertex quadrature

In order to compute a finite element solution to the spde, we must find an approximation to the vector $W^n - W^{n-1}$. The elements of the vector, w_k are given in (3.11). For convenience, we define

$$\mathcal{I}_k := \int_{\mathcal{D}} \Phi_k(x) \sigma \left(t_{n-1}, x, \sum_{i=1}^{M_h} \xi_i^{n-1} \Phi_i(x) \right) e_j(x) \, dx.$$

To approximate the integral $\mathcal{I}_k$, we use the so-called vertex quadrature, i.e., integrating the function g over a triangle K, the quadrature formula is

$$\int_K g \, dx \approx \frac{|K|}{3} \sum_{l=1}^{3} g(x_K^l), \tag{3.12}$$

where $|K|$ is the area of K and x_K^l are the corner nodes of the triangle. Applying the above quadrature formula, we get

$$\begin{aligned}\mathcal{I}_k &= \sum_{K\in\mathcal{T}_h}\left[\int_K \Phi_k(x)\sigma\left(t_{n-1},x,\sum_{i=1}^{M_h}\xi_i^{n-1}\Phi_i(x)\right)e_j(x)\,dx\right]\\ &\approx \sum_{K\in\mathcal{T}_h}\left[\frac{1}{3}\sum_{l=1}^{3}\sigma\left(t_{n-1},x_K^l,\sum_{i=1}^{M_h}\xi_i^{n-1}\Phi_i(x_K^l)\right)e_j(x_K^l)\int_K\Phi_k(x)\,dx\right].\end{aligned}$$

From the way the functions $\{\Phi_i\}$ are constructed, we know that $\Phi_k(x)=0$ on a triangle K, unless the point P_k is a corner point of the triangle, i.e., $P_k = x_K^l$ with $l\in\{1,2,3\}$. Thus, if P_k is not a corner point of the triangle, we get $\int_K \Phi_k(x)\,dx = 0$ and the triangle makes no contribution to $\mathcal{I}_k$. We now define a subset $\mathcal{T}_h^k$ of $\mathcal{T}_h$ such that if P_k is a corner node of K, then $K\in\mathcal{T}_h^k$. This gives us

$$\mathcal{I}_k = \sum_{K\in\mathcal{T}_h^k}\frac{|K|}{9}\sum_{l=1}^{3}\sigma\left(t_{n-1},x_K^l,\sum_{i=1}^{M_h}\xi_i^{n-1}\Phi_i(x_K^l)\right)e_j(x_K^l).$$

Noting that we also have that $\Phi_i(P_k)=0$ if $i\neq k$ and $\Phi_i(P_k)=1$ if $i=k$ gives us

$$\mathcal{I}_k = \sum_{K\in\mathcal{T}_h^k}\frac{|K|}{9}\sum_{l=1}^{3}\sigma(t_{n-1},x_K^l,\xi_{K,l}^{n-1})e_j(x_K^l),$$

where $\xi_{K,l}^{n-1}$ with $l\in\{1,2,3\}$, stands for ξ^{n-1} in the corner points of the triangle K. This leaves us with

$$w_k = \sum_{K\in\mathcal{T}_h^k}\frac{|K|}{9}\sum_{l=1}^{3}\left(\sigma(t_{n-1},x_K^l,\xi_{K,l}^{n-1})\sum_{j=1}^{J}\gamma_j^{1/2}(\beta^n-\beta^{n-1})e_j(x_K^l,)\right). \tag{3.13}$$

Algorithm for the assembly of $W^n - W^{n-1}$

We note that the elements w in equation (3.13) depend on ξ^{n-1}, i.e., on the finite element solution at the previous time step. Therefore, the vector $W^n - W^{n-1}$ must be computed in each time step when the noise term σ is multiplicative, i.e., $\sigma=\sigma(x,t,u)$. However, when the noise term is additive, i.e., $\sigma=\sigma(x,t)$, w is independent of ξ^{n-1}, which allows for a one-time assembly of the vector. This significantly reduces the run time of the solver, as we will see in Section 3.3.4.

In Algorithm 1 below, we outline the assembly of the vector $W^n - W^{n-1}$ when the noise term is multiplicative. Note that instead of focusing on each element in the vector at a time like in equation (3.13), we look at the contribution of each triangle to the vector elements.

3.3.2 Eigenvalues and eigenfunctions of the Laplacian in a domain $\mathcal{D}$

As we have briefly mentioned, since the Wiener process in (3.2) is written in terms of its Fourier series, it becomes necessary to compute the eigenvalues and eigenfunctions of the Laplacian operator Δ on $\mathcal{D}$. The reason is that the quantities γ_j and $e_j(x)$ are needed

Algorithm 1 – Assembly of W^n - W^{n-1}

Input parameters: time point, t^{n-1}; fem solution at t^{n-1}, u^{n-1}; eigenvalues, $gamma_j$; eigenfunctions of Q, e_j, Brownian increments at time t^n; mesh data.

Output: The assembled vector, W.

INITIALIZATION
begin
    initialize matrix for data storage, W
    obtain information on number of triangles and node points in mesh
end

LOOP OVER TRIANGLES IN MESH
for *each triangle* **do**
    find corner points of triangle, $[c_1\ c_2\ c_3]$
    calculate triangle area, triSize
    find solution at previous time point in corner coordinates, $[u_1\ u_2\ u_3]$
    evaluate *sigma* in corner coordinates at time t_{n-1} and u_{n-1}, $[s_1\ s_2\ s_3]$
    find eigenfunctions at corner nodes, $[e_1\ e_2\ e_3]$

    compute contribution of the triangle to $W^n - W^{n-1}$:

$$tmp_1 = s_1 * sum(gamma^{0.5} * e_1 * dB)$$
$$tmp_2 = s_2 * sum(gamma^{0.5} * e_2 * dB)$$
$$tmp_3 = s_3 * sum(gamma^{0.5} * e_3 * dB)$$

$$W(c_1) = W(c_1) + triSize/9 * (tmp_1 + tmp_2 + tmp_3)$$
$$W(c_2) = W(c_2) + triSize/9 * (tmp_1 + tmp_2 + tmp_3)$$
$$W(c_3) = W(c_3) + triSize/9 * (tmp_1 + tmp_2 + tmp_3)$$

end

in order to compute the quadrature in (3.13), where $\{\gamma_j, e_j(x)\}$ is the eigensystem for the covariance operator of the Wiener process, and $\gamma_j = \lambda_j^{-\alpha}$, $\alpha \in \mathbb{R}$, where $\{\lambda_j, e_j(x)\}$ is the eigensystem for the Laplacian. As a consequence, the eigensystem of the Laplacian on $\mathcal{D}$ must be known in advance, or must be possible to estimate.

Rectangular domains in $\mathbb{R}^2$

All the numerical experiments in this thesis are done in rectangular domains in $\mathbb{R}^2$, $\mathcal{D} = (a, b) \times (c, d)$, on which we have the following analytical formulas for the eigenvalues and eigenfunctions of the Laplacian,

$$\lambda_{ml} = \left(\frac{m^2}{(b-a)^2} + \frac{l^2}{(d-c)^2} \right) \pi^2, \tag{3.14}$$

and

$$e_{ml}(x, y) = \frac{2}{\sqrt{(b-a)(d-c)}} \sin\left(\frac{m\pi x}{b-a} \right) \sin\left(\frac{l\pi y}{d-c} \right), \tag{3.15}$$

respectively. Since the eigenvalues and eigenfunctions in (3.13) are given by a single index, while the equations above use a double index, the correspondence between the equations must be clarified. We have that λ_j are computed by using equation (3.14), but that the eigenvalues are ordered by increasing size, including multiple eigenvalues. The eigenfunction, $e_j(x)$, is computed using the indices m and l corresponding to j.

It can be shown that

$$\frac{4\pi j}{(b-a)(d-c)} \tag{3.16}$$

is an adequate approximation to λ_j for large j. Also, since every domain in $\mathbb{R}^2$ contains a square and is contained in one, we have $cj \leq \lambda_j \leq Cj$. The corresponding inequality in $\mathbb{R}^d$ is

$$cj^{2/d} \leq \lambda_j \leq Cj^{2/d}, \tag{3.17}$$

see, e.g., Larsson and Thomée [7]. Although we use the analytical formulas (3.14) and (3.15) in our calculations, the above approximations of the eigenvalues can be useful in theoretical work.

The graph on the left hand side in Figure 3.1 displays the 1200 smallest eigenvalues of the Laplacian on the unit square, precisely calculated and using the approximation in (3.16). The plot on the right hand side shows the logarithm of the relative difference between the precise and approximate values. We can see that as the eigenvalues grow in size, the relative difference decreases.

Figure 3.2 shows a few eigenfunctions of the Laplacian on the unit square, calculated using (3.15). We note that the oscillations increase with the index j.

General bounded domains in $\mathbb{R}^2$

It is clear that the use of precise eigenvalues and eigenfunctions of the Laplacian greatly restricts the choice of spatial domains that the spde can be solved in. Therefore, it is desirable to develop a procedure to estimate the eigensystem of the Laplacian in a general bounded

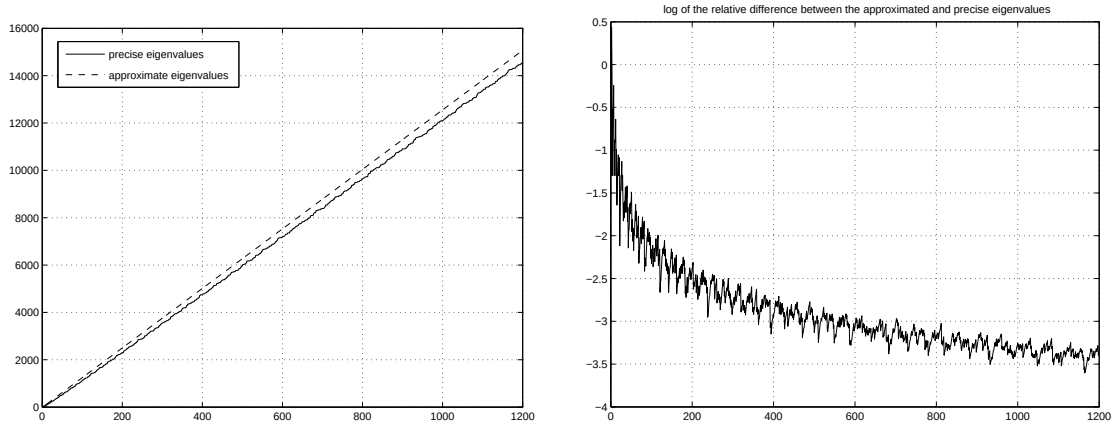

Figure 3.1: Left: 1200 precise and approximate eigenvalues of the Laplacian operator on the unit square. Right: log of the relative difference between the precise eigenvalues and the approximate eigenvalues of the Laplacian.

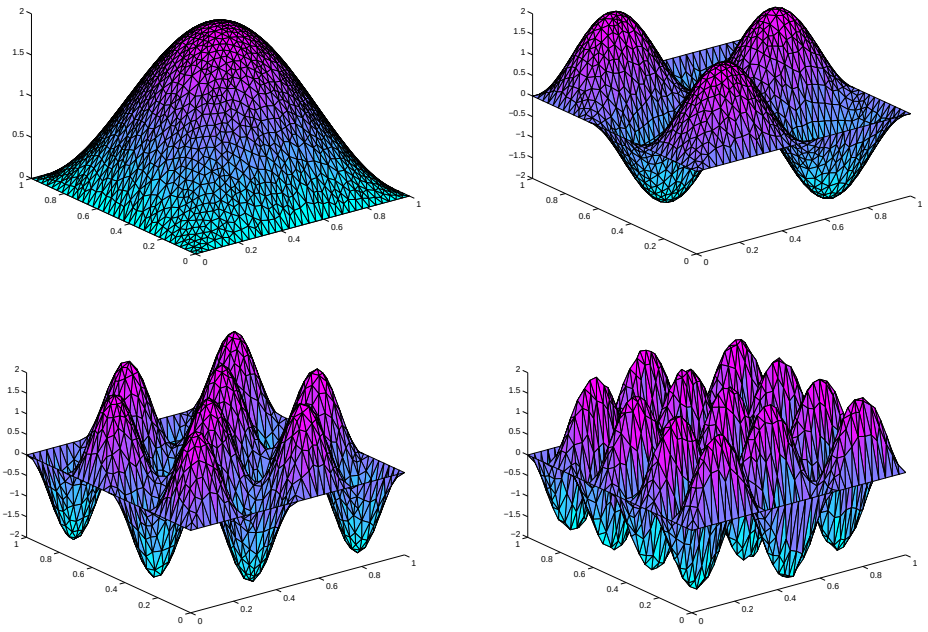

Figure 3.2: Eigenfunctions, $e_j(x)$, of the Laplacian operator on the unit square. Top left: $e_4(x)$, top right: $e_{15}(x)$, bottom left: $e_{31}(x)$, bottom right: $e_{67}(x)$.

domain in $\mathbb{R}^2$. Although we do not go into much detail here, we outline an estimation procedure based on the finite element method. It requires us to solve the eigenvalue problem,

$$-\Delta u = \lambda u \qquad \text{in } \mathcal{D}, \qquad \text{with } u = 0 \text{ on } \Gamma. \tag{3.18}$$

We apply the finite element method to solve the above problem, using the spatial discretization described in Section 3.2.2. The discrete problem corresponding to (3.18) is then as follows. We seek $u_h \in S_h$ and $\lambda_h \in \mathbb{R}$ such that

$$\int_{\mathcal{D}} \nabla u_h \nabla \chi \, dx = \lambda_h \int_{\mathcal{D}} u_h \chi \, dx \qquad \forall \chi \in S_h. \tag{3.19}$$

Equation (3.19) can be written in matrix form as the following generalized eigenvalue problem,

$$AU = \lambda_h BU, \tag{3.20}$$

where the matrix A is the stiffness matrix and B the mass matrix. Since Δ is self-adjoint, the eigenvalue problem has positive eigenvalues $\{\lambda_{n,h}\}_{n=1}^{M_h}$ and orthonormal eigenfunctions $\{e_{n,h}\}_{n=1}^{M_h}$. Also, the eigenfunctions are the stationary points, u_n, of the Rayleigh quotient

$$\mathcal{R}(v) := \frac{\int_{\mathcal{D}} \nabla v \cdot \nabla v \, dx}{\int_{\mathcal{D}} v^2 \, dx},$$

with corresponding eigenvalues $\lambda_n = \mathcal{R}(u_n)$. This allows for the derivation of the following upper bound for the discretization error of the computed eigenvalues,

$$0 \leq \lambda_{n,h} - \lambda_n \leq C \lambda_n^k h^{2(k-1)}, \tag{3.21}$$

where $k-1$ is the polynomial degree of the finite element shape functions. (See [3]).

We make three important observations from the a priori error estimate in equation (3.21). Firstly, the accuracy of the estimated eigenvalues improves as we use finite elements of a higher degree. However, this improved accuracy comes at the cost of more storage space and memory requirements.

Secondly, we obtain better approximations of the eigenvalues for finer triangulation meshes, i.e., the error decreases as the mesh parameter h gets smaller. This behavior is shown in Figure 3.3, which shows 469 precise eigenvalues as well as the same number of approximated eigenvalues computed on 4 different mesh sizes. The approximation is clearly least accurate for the coarsest mesh, with $h = 2^{-4}$, which contains exactly 469 mesh points. However, for the finest mesh, $h = 2^{-7}$, the approximation is quite good.

Thirdly, for a given triangulation, the accuracy of higher eigenvalues deteriorates at the rate λ_n^k, or at the rate λ_n fore linear finite elements. This, can be observed from Figure 3.4, which displays the logarithm of the relative discretization error. For the accurate computation of a large number of eigenvalues, we thus need a much finer discretization mesh, compared with the case when only a few eigenvalues are needed.

As the above figures indicate, it is important to evaluate the quality of the computed eigenvalues before they are used in practice. This can be done by solving the eigenvalue problem on a hierarchy of refined discretizations until the predicted asymptotic behavior,

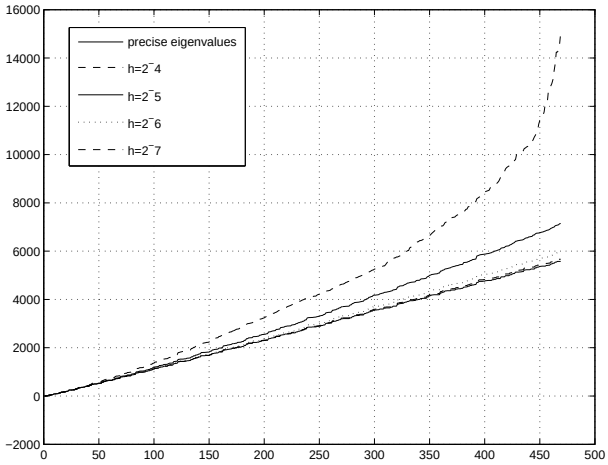

Figure 3.3: A comparison between 469 precise and approximate eigenvalues of the Laplacian. The approximations are computed on 4 different mesh sizes, using the finite element method of degree one.

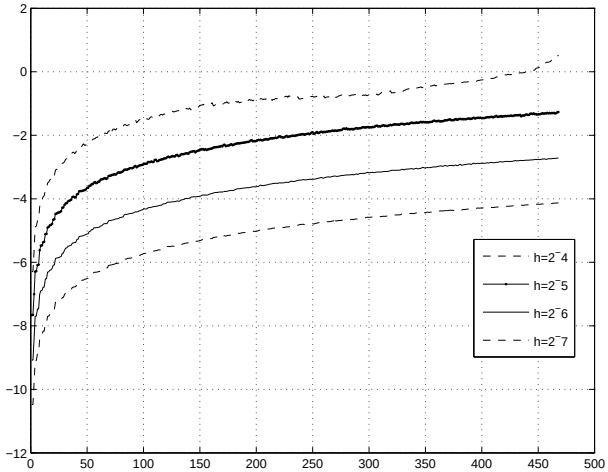

Figure 3.4: Logarithm of the relative discretization error of the eigenvalues, computed by using 4 different mesh sizes.

according to the a priori error estimate (3.21), is detected. We do not go into details about this procedure here, but refer to Heuveline [3] and references therein for further information. Moreover, Heuveline [3] describes a computation process for the eigenvalue estimation of self-adjoint elliptic operators that is built on the Lanczos and multigrid methods. This procedure allows for an efficient computation of a very large number of eigenvalues, and allows, e.g., the consideration of complex geometries.

As a final point, it is interesting to contemplate how the above method would work for our problem. That is, could we apply this method to generate approximate eigenvalues and eigenvectors that we could use to solve the spde in (3.2)? Our results indicate that the method could be used when we want to solve the spde on a coarse mesh, e.g., with $h = 2^{-4}$, at least when $\mathcal{D}$ is the unit square. However, we may run into trouble when considering finer meshes, since we would then require more eigenvalues. For example, in order to solve the spde on the unit square on a uniform mesh with $h = 2^{-6}$, we need at least $M_h = 7132$ eigenvalues of the Laplacian. The solution could be to use a finite element method of a higher degree, but this is problem dependent and may also depend on the geometry of the domain.

3.3.3 Implementation of the solver

When all the calculation routines for the matrices in equation (3.10) have been implemented, the spde solver can be constructed. Before we present an algorithm for the solver, it is worth keeping the following points in mind, which concern our method of implementation.

1. The solver allows us to compute the finite element solution for several different partitions of the time interval at the same time. However, the length of the time step of a particular partition must be an integer multiple of the basic time step.

2. The parameter α controls the regularity of the Wiener process and thus of the solution. As we saw in Chapter 2, $\alpha \in (0, \infty)$ guarantees existence and uniqueness of a solution for the spde in $\mathbb{R}^2$ with $\sigma = I$.

3. The random number generation for the Brownian increments is performed in the solver. Also the computation of the eigenvalues of Q and eigenvectors at the mesh points. These are then sent as inputs into the function that assembles the vector $W^n - W^{n-1}$.

4. Note that this algorithm does not restrict us to rectangular domains in $\mathbb{R}^2$. That restriction is implemented in the function that computes the eigenvalues of the Laplacian.

5. The user input parameter NOISE can take on the values 'add' or 'mult'. If we have NOISE='mult' we need to assemble the $W^n - W^{n-1}$ vector for each time point and for each of the different partitions of the time interval. The reason is that the vector depends on the finite element solution at the previous time point. However, if NOISE='add', we only need to assemble the vector once for each of the different time partitions.

An outline for the implementation of the finite element solver is given in Algorithm 2 below.

Algorithm 2 – The solver:

Input parameters: alfa, initial time point – t_0, final time point – T, approximation of fem solution at t_0 – u_0, data on different time steps, mesh data, NOISE (additive or multiplicative).
Output: The finite element solution at time T for each of the different time steps.

```
Look for errors in input data

Assemble stiffness- and mass matrices – call the user defined function Assemble

INITIALIZATION
begin
    | initialize matrices for data storage
    | find how many different time steps we are looking at
    | find the number of time intervals for each time step
    | compute the length of the basic time step
end

RANDOM NUMBER GENERATION
begin
    | create J paths of Brownian increments for the basic timestep – dB
    | compute the corresponding Brownian paths - B
end

compute eigenvalues of the Laplacian on the space mesh specified by the
mesh data – call the user defined function Eigen

compute eigenvalues of Q (covariance operator of the Wiener process)

compute eigenfunctions in the mesh points

LOOP OVER DIFFERENT TIME STEPS
for each timestep do
    | compute the length of the current time step
    | compute the Brownian increments for this particular time step - dB_step
    | if noise is additive do
    |       Assemble the W-vector – call the function Assemble_W_add
    | end
    |       TIME LOOP
    |       for each timestep do
    |           | if noise is additive do
    |           |       Assemble the W-vector – call Assemble_W_mult
    |           | end
    |           | compute the fem solution at the current time point
    |       end
end
```

3.3.4 Implementation details

The code used to implement the finite element solver is solely written in Matlab. The main advantage of Matlab is ease of use; it provides a good working environment which makes for fast code development time. A significant drawback of using Matlab is the execution time of the code. Since the computation process involves several iterations, especially in the case of the multiplicative solver, it would be desirable to employ another computer language, such as C++ or Fortran. However, an implementation in Matlab is often a useful first step in the development process. The solution can then be transported into other working environments if greater efficiency is required.

In the table below we show the execution times in seconds for the finite element solver, using space- and time meshes of different size. On the left, the solver is run on the spde with a multiplicative noise term and on the right, the spde has an additive noise term. The simulations are run on a dual Opteron machine. See

`http://www.math.chalmers.se/~kf99crch/Brummelisa/`

for further information on software and hardware.

Time	Space - **h**			
k	2^{-6}	2^{-5}	2^{-4}	2^{-3}
2^{-11}	24549.02	1560.73	195.44	34.65
2^{-10}	12218.13	757.81	93.64	16.32
2^{-9}	5983.64	376.24	45.49	7.89
2^{-8}	3069.28	187.75	22.59	3.86
2^{-7}	1602.91	94.46	11.28	1.93
2^{-6}	774.95	47.83	5.79	0.99
2^{-5}	414.07	24.56	2.97	0.51
2^{-4}	207.31	13.02	1.57	0.27
2^{-3}	117.52	7.19	0.87	0.15
2^{-2}	66.34	4.37	0.53	0.10
2^{-1}	42.72	2.91	0.35	0.05

Time	Space - **h**			
k	2^{-6}	2^{-5}	2^{-4}	2^{-3}
2^{-11}	2593.30	179.68	27.83	5.80
2^{-10}	1173.68	77.38	10.52	1.82
2^{-9}	741.30	37.04	4.50	0.57
2^{-8}	373.38	19.67	2.04	0.25
2^{-7}	204.66	11.61	0.92	0.15
2^{-6}	121.63	6.01	0.61	0.11
2^{-5}	85.20	3.79	0.47	0.07
2^{-4}	59.42	3.25	0.41	-
2^{-3}	42.77	2.94	0.36	-
2^{-2}	39.08	2.77	0.36	-
2^{-1}	8.25	2.65	0.35	-

Table 3.1: Run times of the finite element solver (in seconds) for different sizes of time and space meshes. Left: spde with a multiplicative noise term. Right: spde with an additive noise term.

As is clear from the table, the run times differ greatly depending on whether the spde has a multiplicative or an additive noise term. In the latter case, the running times are quite acceptable and it will be possible to carry out simulations for all the time- and space meshes above. However, when the noise term is multiplicative, the running time of a single simulation for the smallest time and space meshes with $h = 2^{-6}$ and $k = 2^{-11}$ is almost 7 hours. Keeping in mind that very many simulations are needed to obtain a result, the execution time of the code will most certainly be a restricting factor for the numerical simulations.

Chapter 4

The Numerical Experiments and Expected Results

In the previous two chapters, we drew up the theoretical framework of the thesis and derived a method to compute a numerical solution to the spde in (1.1). Here, we put theory into practice and present some numerical experiments. A major purpose of numerical experiments in general is to gain insight into mathematical theory, to visualize the behavior of the mathematical model and to discover how its solution is affected by changes to the model parameters. Moreover, numerical tests can be used to verify mathematical theorems, e.g., on error bounds. When theoretical estimates do not exist, numerical experiments can give an idea about the magnitude of the error.

The main purpose of the numerical experiments presented in this thesis is to examine the convergence behavior of the numerical method, i.e., how the convergence rate depends on the parameters of the mathematical model. In particular, we examine how the regularity of the Wiener process affects the convergence of the numerical solution and how the convergence depends on the form of the operator σ.

Another important purpose of the numerical experiments is to verify theoretical error estimates. Here, we compare our results on strong convergence of the numerical method to error bounds presented in Theorems 1.1 and 1.2 in Yan [11]. We also examine the weak convergence rate of the method.

We split the discussion on the numerical experiments into two separate chapters. In this chapter, we present a description of the setup of the experiments and their expected results, while in Chapter 5 we put forward the actual results of our experiments and compare them to the expected results. In Chapter 5, we also examine how individual solutions to the model are dependent on the model parameters.

The layout of this chapter is as follows. We begin, in Section 4.1, by a discussion on convergence, where we define the concepts of weak and strong convergence. Also, we put forward a procedure for the computation of the convergence rate. In Section 4.2, we specify the mathematical model and provide a description of the experiments. Finally, in Section 4.3 we present the expected results of the experiments, according to the available theoretical error estimates.

4.1 The notion of convergence

In the numerical experiments below, we examine the convergence of the numerical method. In other words, the convergence of the finite element solution to the true solution as the time step, $k = \Delta t$, gets smaller. We begin by presenting two convergence concepts, strong- and weak convergence, in Subsection 4.1.1. Then, in Subsection 4.1.2, we describe the computational procedure for strong and weak convergence that is used in the numerical experiments. Here, we also discuss some error sources in connection with the computation.

4.1.1 Strong and weak convergence

In some of the experiments below, we want to examine whether the finite element solution converges to the true solution as the time step, k, gets smaller. In order to make the notion of convergence precise, we must decide how to measure the difference between two random variables, X_1^n and X_2^n, at time $t = t^n$. Using $\mathbf{E}|X_1^n - X_2^n|$, where $\mathbf{E}$ stands for expected value, leads to the concept of **strong convergence**. A method is said to have strong convergence of order δ if there exists a constant C such that

$$e_{strong}^k := \mathbf{E}|X_1^n - X_2^n| \leq Ck^\delta, \tag{4.1}$$

for a fixed time, $t = t_n \in [0, T]$ with k sufficiently small. In our numerical experiments we concentrate on the error at the endpoint of the time interval, $t = T$. If we assume approximate equality in equation (4.1) and take logarithms, we get

$$\log e_{strong}^k \approx \log C + \delta \log k. \tag{4.2}$$

Plotting the errors on a log-log scale against k should give us a line with slope δ, where δ can be found by a least squares fit.

The strong rate of convergence measures the rate of how the mean of the errors decreases as $k \to 0$. Another, and less demanding, alternative is to measure the decay of the error of the means. This results in the concept of **weak convergence**. A method is said to have weak convergence of order δ if

$$e_{weak}^k := |\mathbf{E}p(X_1^n) - \mathbf{E}p(X_2^n)| \leq Ck^\delta, \tag{4.3}$$

for a fixed time, $t = t_n \in [0, T]$, with k sufficiently small and for all p in a large class of functions, which must fulfill some smoothness and growth conditions.

4.1.2 Computation of strong and weak convergence

We use the same basic procedure for the computation of strong and weak convergence. We fix a small space step, h, and a sequence of different time steps, k_i, where $i = 1, \ldots, N$. We carry out M simulations, denoted by ω_j, with $j = 1, \ldots, M$, and compute the finite element solution for each simulation and for each of the N time steps. For a given time step, k_i, the finite element solution at time t_n for simulation j is denoted $U^n(\omega_j)$. Analogously, $u^n(\omega_j)$ is the true solution at time t_n for simulation j.

We note that since the true solution to the spde is a random process, the true solution is not explicitly known and must be approximated. Here, we take the "true" solution to be the finite element solution computed on a very fine space and time grid.

Computation of strong convergence

We want to compute strong convergence in the H-norm, where $H = L_2$. That is, explore how the norm of the error,

$$\|U^n - u^n\|_{L_2(\Omega;L_2)} = \left(\mathbf{E}\|U^n - u^n\|^2\right)^{1/2},$$

depends on the length of the time step, k. Here, U^n is the finite element approximation at time t_n and u^n is the true solution to the spde at time t_n. In other words, we examine if and how the finite element approximation approaches the true solution as the length of the time step decreases.

In order to find $(\mathbf{E}\|U^n - u^n\|^2)^{1/2}$ we take the average over all the M simulations for a given time step k_i, i.e.,

$$S(k_i) := \left(\frac{1}{M}\sum_{j=1}^{M}\|U^n(\omega_j) - u^n(\omega_j)\|^2\right)^{1/2} \approx \left(\mathbf{E}\|U^n - u^n\|^2\right)^{1/2}.$$

We compute $S(k_i)$ for all the time steps k_i, where $i = 1, \ldots, N$.

Let us now demonstrate how to compute $\|U^n(\omega_j) - u^n(\omega_j)\|$ for a given simulation, ω_j, and a given time step, k_i. Just as when we were computing the vector $W^n - W^{n-1}$ in Section 3.3, we use vertex quadrature. We get

$$\begin{aligned}\|U^n - u^n\|^2 &= \int_{\mathcal{D}}(U^n - u^n)^2 dx \\ &= \sum_{K\in\mathcal{T}_h}\int_K (U^n - u^n)^2 dx \\ &\approx \sum_{K\in\mathcal{T}_h}\frac{|K|}{3}\sum_{l=1}^{3}\left(U^n(x_K^l) - u^n(x_K^l)\right)^2.\end{aligned} \tag{4.4}$$

The contribution of each triangle in the space mesh to the squared L_2-norm of the error is thus the average of the squared difference between the true solution and the finite element approximation in the corner points of the triangle, multiplied by the area of the triangle in question.

Computation of weak convergence

In order to assess the rate of weak convergence of the numerical method, we examine how the estimate

$$|\mathbf{E}p(U^n) - \mathbf{E}p(u^n)|,$$

where U^n and u^n are the finite element approximation and the true solution to the spde at time t_n, respectively, depends on the length of the time step, k.

Below, we examine four different functions, p. The first two, p_1 and p_2, are in the form $H \to \mathbb{R}$, with $p_1(v) = \|v\|_H$ and $p_2(v) = \|v\|_H^2$, for all $v \in H$. Here, we have $H = L_2(\mathcal{D})$. We thus want to evaluate $\left|\mathbf{E}\|U^n\|^l - \mathbf{E}\|u^n\|^l\right|$, where $l = 1, 2$ depending on whether we are

evaluating p_1 or p_2. In order to find $\mathbf{E}\|U^n\|^l$, we compute the L_2 norm of the finite element solution for all the M simulations and take the average, i.e.,

$$\mathbf{E}\|U^n\|^l \approx \left(\frac{1}{M} \sum_{j=1}^{M} \|U^n(\omega_j)\|^l \right).$$

The other two functions, p_3 and p_4, are in the form $H \to \mathbb{R}$, with $p_3(v) = v(x_0)$ and $p_4(v) = v(x_0)^2$, for all $v \in H$ and $x_0 \in \mathcal{D}$. As before, we have that $H = L_2(\mathcal{D})$ and we want to evaluate $\left|\mathbf{E}U^n(x_0)^l - \mathbf{E}u^n(x_0)^l\right|$, where $l = 1, 2$ depending on whether we are evaluating p_3 or p_4. In order to find $\mathbf{E}U^n(x_0)^l$, we take the average over all the M simulations of the finite element solutions at the point x_0, i.e.,

$$\mathbf{E}U^n(x_0)^l \approx \left(\frac{1}{M} \sum_{j=1}^{M} U_j^n(x_0)^l \right),$$

where $U_j^n(x_0)$ is the finite element solution for the j-th simulation, ω_j, at the point x_0 and at time $t = t_n$.

Computational errors

There are a number of errors involved in the above computation of strong convergence, including the following:

- **Sampling error:** arising from the approximation of expected value by a sampled average.
- **Random number bias:** errors in random number generators.
- **Rounding error:** errors in rounding off floating point numbers.
- **Quadrature error:** error of approximating integrals by quadrature formulas.

Out of the first three above, sampling error is typically the most significant. Therefore, care must be taken to perform sufficiently many simulations so that the predicted rate of convergence is observable. The sampling error decays like $1/\sqrt{M}$, where M is the number of sampling points used, see, e.g., Higham [4].

We do not know the full importance of quadrature error. For a smooth function, $f \in H^s$, the quadrature error is approximately

$$h^s \|f\|_{H^s}, \quad \text{with } s \leq 2, \tag{4.5}$$

for the vertex quadrature. Since dW is not a smooth function, we expect the quadrature error to be larger in our case. In fact, we have $dW \in H^{\beta-1}$ which would, as a worst case scenario, give us a quadrature error of magnitude $h^{\beta-1}\|dW\|_{H^{\beta-1}}$. However, the smoothing effect of the numerical method must be taken into account and thus we expect the error to be somewhat smaller. In any case, we suspect that quadrature error is an important error source,

which can greatly affect our results and consequently the convergence rate of the numerical method.

A suggested extension to the work done here is to derive an estimate of the quadrature error and to find and implement ways to compensate for it. We note, that using a more precise quadrature method should not have any effect here, since it will only increase s in (4.5). This has no effect, since s is an upper bound that does not come into play in the stochastic case.

Since the run time of the solver is a major limiting factor in our numerical experiments, we must seek to obtain a balance between the number of simulations, M, and the mesh size, h. However, it should be clear that the error sources discussed above can prevent us from observing the expected convergence rate of the numerical method.

4.2 Setup of the numerical experiments

The main purpose of the numerical experiments is to examine the convergence rate of the numerical method. In particular, the relationship between the rate of convergence and the regularity of the Wiener process, W, which in turn depends on the parameters α and β, as discussed in Chapter 2. It is also of special interest to compare the numerical results to the theoretical error bounds, proved in Yan [11], for different forms of the operator σ.

Below, we present the mathematical model that we use for the numerical experiments and describe the experiments that we performe.

4.2.1 The model for the numerical experiments

The numerical experiments are performed on the model in equation (3.2), with the function f in the form

$$f(x,t) = 2(8\pi^2 - 1)e^{-t}\sin(2\pi x_1)\sin(2\pi x_2),$$

where $x = (x_1, x_2) \in \mathcal{D}$, $\mathcal{D}$ is the unit square in $\mathbb{R}^2$, and $t \in [0,1]$. Furthermore, we have $u(x,0) = 2\sin(2\pi x_1)\sin(2\pi x_2)$ and $u(0,t) = 0$.

The true solution to the deterministic version of the model is given by

$$u(x,t) = 2e^{-t}\sin(2\pi x_1)\sin(2\pi x_2),$$

and is displayed in Figure 4.1 at time $t = 1$.

4.2.2 An outline of the numerical experiments

The main aim of the numerical experiments is to examine the rate of strong and weak convergence of the numerical method. In particular, how the rate of convergence is related to the parameters of the noise term, α, β, and σ.

Although the finite element solver is built to handle all forms of the operator σ, our experiments concentrate mainly on the case when $\sigma = I$. Partly, the reason is that most of the theoretical conclusions in Chapter 2 apply only when $\sigma = I$. The main reason is, however, that we are not able to establish the convergence rate of the numerical method in this case.

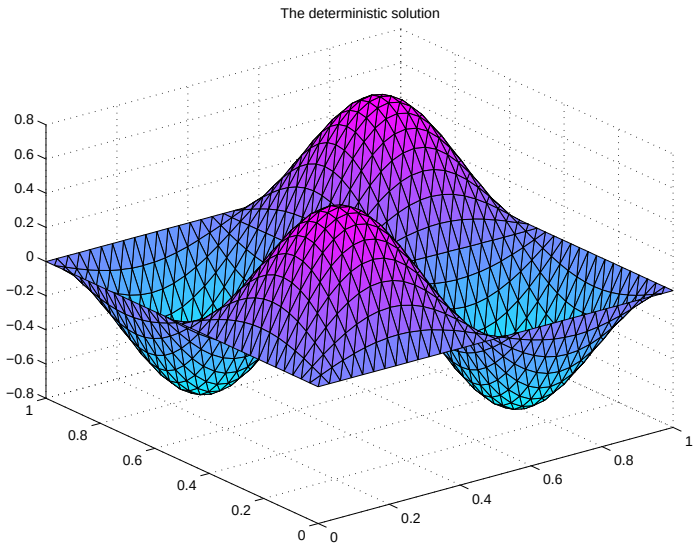

Figure 4.1: The true solution to the deterministic version of the model at time $t = 1$

Since this is the simplest version of the model, we believe it to be more important to try to discover why we do not obtain the predicted convergence rates than to perform simulations for other versions of the operator σ.

In our main numerical experiment, we compute the rate of strong and weak convergence of the numerical method for the following six different values of α: $\alpha = 0$, $\alpha = 0.5$, $\alpha = 1$, $\alpha = 1.5$, $\alpha = 2$, and $\alpha = 4$. Due to the lengthy run time of the finite element solver, displayed in Table 3.1, we let the space step be $h = 2^{-5}$. We examine the time steps, $k_i = 2^{-i}$, with $i = 1, \ldots, 8$. The "true" solution is the finite element solution computed on the same space grid as before, but with the time step, $k = 2^{-11}$. We perform $N = 1000$ simulations.

In an attempt to assess if and how our results are affected by the mesh size, we run an experiment on a finer mesh with $h = 2^{-6}$, for $\alpha = 1$. Here, we also perform 1000 simulations.

Finally, we perform an experiment with $\sigma = \|x\|$, for $\alpha = 0$ and $\alpha = 1$, with $N = 1000$, $h = 2^{-5}$, and k_i as above. Here, we concentrate on computing the rate of strong convergence for the method.

4.3 Expected results

When predicting the rate of strong convergence in the H-norm, we use the following error estimates, which are in accordance to Theorems 1.1 and 1.2 in Yan [11].

Theorem 4.3.1. *Let U^n and u^n be solutions of (3.7) and (1.1), respectively. Assume that σ and Q satisfy (i) and (ii'). Assume that $u_0 \in L_2(\Omega; \dot{H}^\beta)$. Then there exists a constant $C = C(T)$ such that, for $t_n \in [0, T]$ and $0 \leq \gamma \leq \beta$,*

$$\|U^n - u^n\|_{L_2(\Omega;H)} \leq C(k^{\gamma/2} + h^\beta) \left(\|u_0\|_{L_2(\Omega;\dot{H}^\beta)} + \sup_{0 \leq s \leq t} \|u(s)\|_{L_2(\Omega;H)} \right). \tag{4.6}$$

When $\sigma = I$, the following error estimation applies,

Theorem 4.3.2. *Let U^n and u^n be solutions of (3.7) and (1.1), respectively. Assume that $\sigma(\cdot) = I$. Further assume that $u_0 \in L_2(\Omega; \dot{H}^\beta)$, $\beta \geq 0$. If $\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{HS} < \infty$ for some $\beta \geq 0$, then we have*

$$\|U^n - u^n\|_{L_2(\Omega;H)} \leq C(k^{\beta/2} + h^\beta)\left(\|u_0\|_{L_2(\Omega;\dot{H}^\beta)} + \|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{HS}\right). \tag{4.7}$$

We do not have any theoretical results to build on for the weak convergence rate. Our predicted convergence rate will in this case build on an educated guess.

4.3.1 Expected strong convergence rates

When $\sigma = I$, Theorem 4.3.2 should apply. This implies that the order of strong convergence of our method should be around $O(k^{\beta/2} + h^\beta)$. If h is sufficiently small, such that the error estimates are dominated by k, the predicted rate of convergence should be $O(k^{\beta/2})$. This gives us

$$\frac{S(k_i)}{S(k_{i+1})} \approx \left(\frac{k_i}{k_{i+1}}\right)^{\beta/2} = 2^{\beta/2},$$

or if we want to compute β, we have

$$\beta = \frac{2}{\log 2}\log\left(\frac{S(k_i)}{S(k_{i+1})}\right). \tag{4.8}$$

An alternative way of computing the convergence rate in the case when the convergence rate is dominated by k, is by using equation (4.2), i.e., plotting the errors, $S(k)$, against k on a log-log scale and estimating the slope of the line using least squares.

In order for the condition $\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{\text{HS}} < \infty$ to apply, β must be in the interval $[0, \alpha + 1 - d/2)$, which reduces to $\beta \in [0, \alpha)$ when $d = 2$. We note, however, that the strong convergence rate of the Euler method for the deterministic model is $O(k)$, which is equivalent to $\beta = 2$. Since the convergence rate of the Euler method for the stochastic model cannot be higher than the convergence rate for the deterministic model, this limits β to the interval $[0, \min(2, \alpha))$. Therefore, the error estimate in Theorem 4.3.2 must be valid for all values of β in the interval $[0, \min(2, \alpha))$, $\beta = \min(2, \alpha)$ in particular. This means that the convergence rate of our method should be close to $O(k^{\min(1,\alpha/2)})$.

As we have stated, we examine strong convergence of the method for six different values of α. The expected convergence rates are as follows:

- $\alpha = \mathbf{0}$: Here, we have time-space white noise with $Tr(Q) = \infty$. For $\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{\text{HS}} < \infty$ to apply, we must have $\beta \in \big[0, \min(2, 1 - d/2)\big)$, which is clearly only valid when $d = 1$. Thus, we expect neither of the error estimates in Theorems 4.3.1 or 4.3.2 to apply.

  We are, however, interested in examining whether the theory is consistent with our numerical experiments in the case where $d = 2$, which is right on the boundary of giving some convergence. We must also note that the condition $\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{\text{HS}} < \infty$ is a *sufficient* condition for existence and uniqueness of a mild solution and is used in the proofs of the error estimates, while a weaker condition might in fact be enough in order to obtain some convergence.

- $\alpha = \mathbf{0.5}$: When $\alpha = 0.5$, the Wiener process is not of trace class.* Nevertheless, we expect the error estimate in Theorem to apply, since we have $\|\mathcal{A}^{(\beta-1)/2}Q^{1/2}\|_{\mathrm{HS}} < \infty$ for $\beta \in [0, 0.5)$. This means that the expected convergence rate of the numerical method is $O(k^{0.25})$.
- $\alpha = \mathbf{1}$: For $\alpha = 1$, the Wiener process is on the verge of being of trace class, when $d = 2$. Here, condition (ii') applies for $\beta \in [0, 1)$ and we thus expect a convergence rate of $O(k^{0.5})$.
- $\alpha = \mathbf{1.5}$: When $\alpha = 1.5$, condition (ii') applies for $\beta \in [0, 1.5)$ and the expected convergence rate is $O(k^{0.75})$.
- $\alpha = \mathbf{2}$ and $\alpha = \mathbf{4}$: In this case, the upper limit of $\beta < 2$ applies and we get an expected convergence rate of $O(k^{1})$.

As we discussed in chapter 2, we have not been able to confirm whether the conditions (i) and (ii') apply when σ is not in the form $\sigma = I$. However, in our experiment with $\sigma = \|x\|$, σ is a reasonably well behaved function and we would therefore expect the error estimate in Theorem 4.3.1 to apply. The expected convergence rate of the numerical method will then be $O(k^{\min(1,\alpha/2)})$, for sufficiently small h such as the error estimates are dominated by k.

4.3.2 Expected weak convergence rates

As we stated above, we do not have any theoretical results regarding the weak convergence rate of the numerical method. If we want to form an idea in advance on the expected convergence rate, we must thus make an educated guess. We guess that the weak convergence rate of the method is $O(k^{\min(2,\alpha)} + h^{2\beta})$, or $O(k^{\min(2,\alpha)})$, for sufficiently small h such as the error estimates are dominated by k. This guess is built on intuition and prior experience of stochastic ordinary differential equations.

*As seen in Section 2.2, $\alpha > 1$ is required for the Wiener process to be of trace class if $d = 2$.

Chapter 5

Results of Numerical Experiments

In this chapter, we present and discuss the results of the numerical experiments described in Chapter 4. It is fair to say that this is one of the most interesting parts of the thesis, since we will discover whether the numerical method conforms to the theoretical convergence estimates. That is, whether the setup that is described in this thesis works in practise.

Although the main purpose of this chapter is to present the results of the numerical experiments, it is of interest to examine some individual solutions to the spde. This provides additional insight into the mathematical model, since it highlights the relationship between the parameters of the noise term and the finite element solution. Thus, we start the chapter by displaying and discussing some individual finite element solutions, along with the corresponding noise processes.

In Section 5.2, we then put forward the results of our numerical experiments described in Chapter 4.

Finally, in Section 5.3, we summarize and discuss the main conclusions of the numerical experiments.

5.1 Individual FEM solutions and noise terms

Here, we present some individual finite element solutions to the mathematical model in (1.1) for different values of α. Also, we display the corresponding instances of the noise processes at a specified point in time. Note, that we observe all the noise processes at the same time point, $t = 30 * dt$, where $dt = 2^{-7}$. It is important to keep in mind that the magnitude of the noise is highly dependent on the size of dt. All the individual solution and noise processes displayed in this section make use of the same random numbers. The random numbers are generated in Matlab, using the command `randn`.*

The discussion below is divided into three parts, depending on the nature of the parameter σ.

*In order to reproduce the finite element solutions, the random number generator should be given a starting value of 100.

5.1.1 Individual results with $\sigma = I$

Noise terms

Figure 5.1 displays the noise of the spde at a single point in time for four different values of the parameter α. It is clear from the Figure that the the noise becomes more regular as the

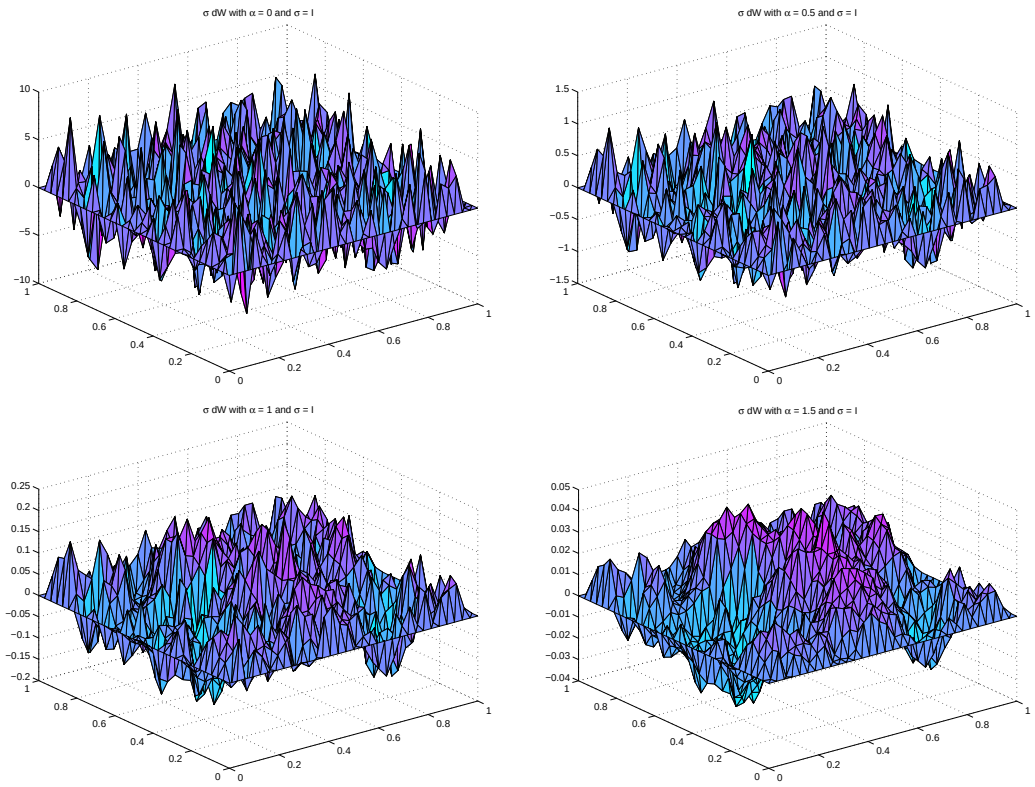

Figure 5.1: The process $\sigma\, dW(t)$, with $\sigma = I$, for four different values of α at a single point in time.

value of α increases. While the values of the process range between 8.4 and -9.2 when $\alpha = 0$, it ranges between 0.21 and -0.18 when $\alpha = 1$. We expect the finite element solution to show a similar trend.

Finite element solutions

In Figure 5.2, we display some individual finite element solutions for the mathematical model. Three of the solutions are stochastic with different values of α and $\sigma = I$, while one is the solution of the deterministic model. The figure shows clearly how the variable α influences the regularity of the finite element solution. When $\alpha = 0$, the noise is white in space and time, and the solution to the model is quite irregular. However, as α is increased, the solution becomes more regular and almost indistinguishable from the deterministic solution. Figure 5.3 displays the difference between the stochastic solution with $\alpha = 1$ and the solution to the deterministic model.

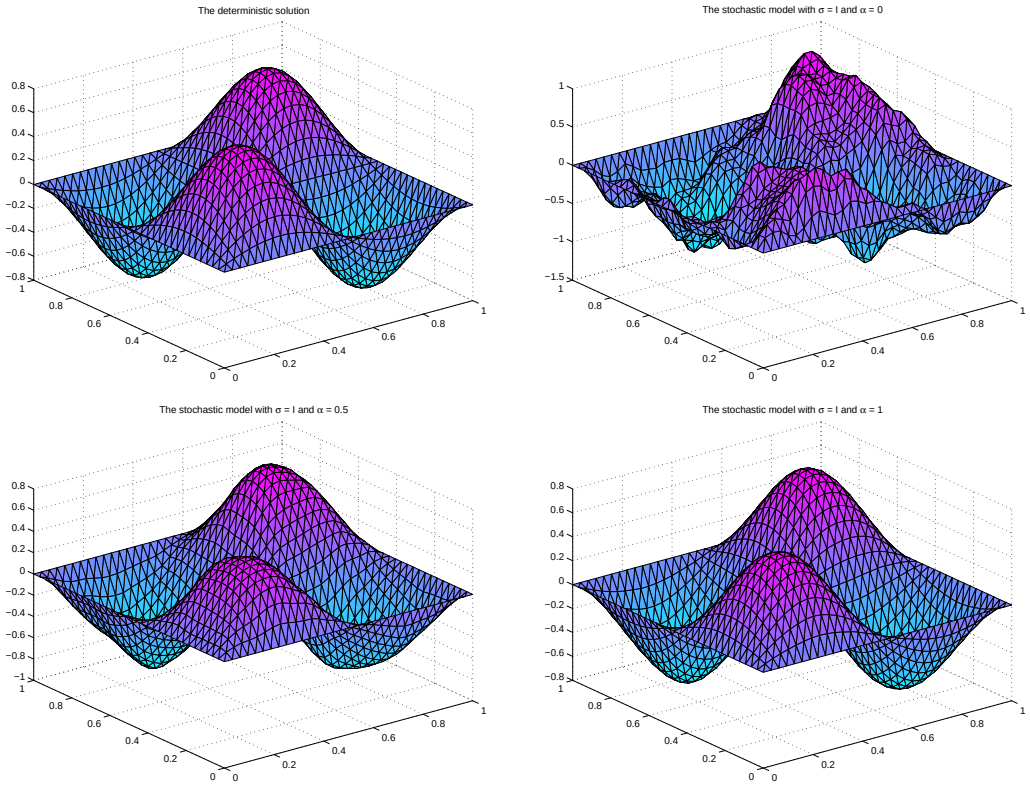

Figure 5.2: Examples of finite element solutions to the mathematical model. Upper left: The deterministic model. Upper right: The stochastic model with $\sigma = I$ and $\alpha = 0$. Lower left: The stochastic model with $\sigma = I$ and $\alpha = 0.5$. Lower right: The stochastic model with $\sigma = I$ and $\alpha = 1$.

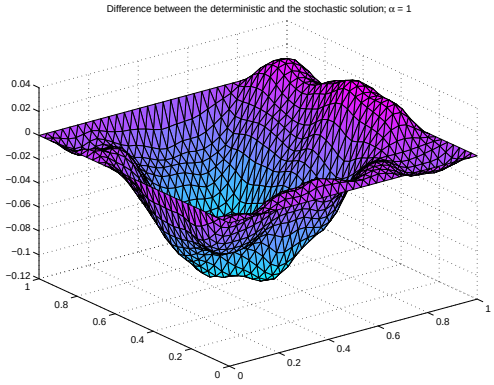

Figure 5.3: Difference between the deterministic and the stochastic solution, where $\alpha = 1$ and $\sigma = I$.

In Section 2.3, we proved that when $\sigma = I$, the parameter β is restricted to the interval $[0, \alpha)$. This guarantees the existence and uniqueness of a mild solution for the spde in (1.1). Furthermore, we showed that β determines the regularity of the Wiener process and the solution, where the solution is smoother than the Wiener process. More specifically, the solution has β derivatives, while the Wiener process has $\beta - 1$ derivatives.

We were, however, not able to prove any more direct relationship between the parameter α and the regularity of either the solution or the Wiener process. The results above support our claims in Section 2.3 that a higher value of α means a more regular noise term and solution, i.e., results in a higher value of β.

5.1.2 Individual results with $\sigma = \|x\|$

Noise terms

In Figure 5.4, we display an example of an additive noise term of the spde, where we let $\sigma = \|x\|$, for $\alpha = 0$ and $\alpha = 1$. The difference from the case shown previously is that we multiply the Wiener process dW with a dampening factor, $\sigma = \|x\|$, where $x = (x_1, x_2)$, $x \in \mathcal{D}$. The dampening effect of σ is most prominent for small values of x_1 and x_2. This

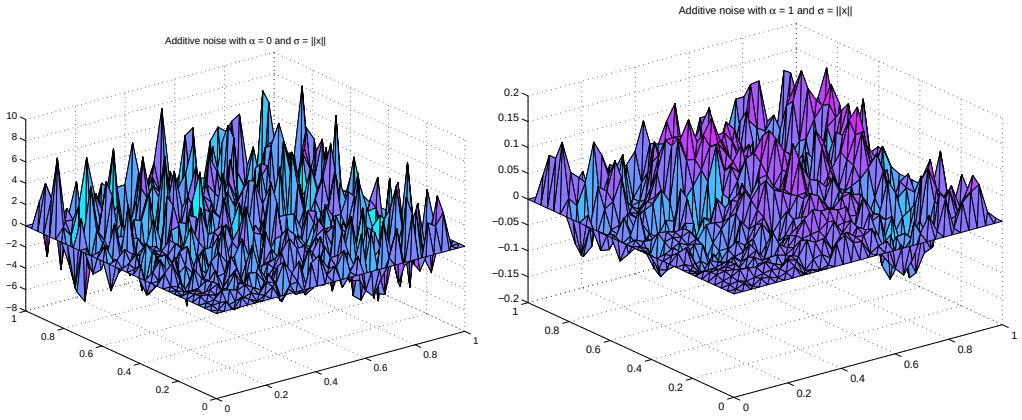

Figure 5.4: Additive noise terms, with $\sigma = \|x\|$ for $\alpha = 0$ and $\alpha = 1$.

is clear from Figure 5.5, which shows the values of the noise term along the diagonal, where $x_1 = x_2$.

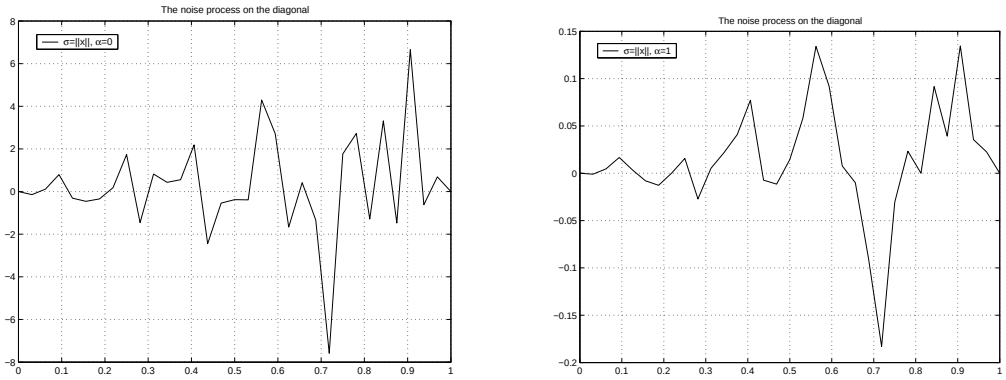

Figure 5.5: Additive noise terms, with $\sigma = \|x\|$ for $\alpha = 0$ and $\alpha = 1$, where $x = (x_1, x_2) \in \mathcal{D}$ such that $x_1 = x_2$.

Finite element solutions

In Figure 5.6, we present individual finite element solutions for two values of α, $\alpha = 0$ and $\alpha = 1$. The solutions remind us of the results in Figure 5.2. As could be expected, the main

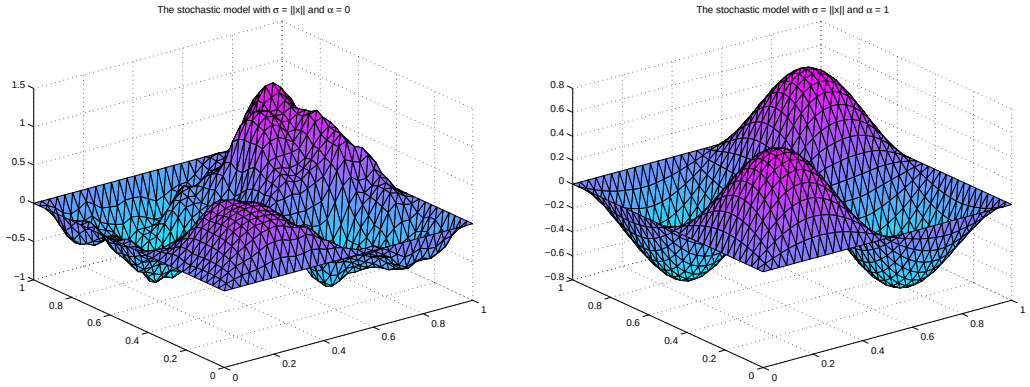

Figure 5.6: Examples of finite element solutions to the mathematical model at time $t = T$ with $\sigma = I$ and $\alpha = 0$ (plot on the left) and $\alpha = 1$ (plot on the right).

difference is that the solution here is somewhat dampened, especially for small values of x. Again we observe the relationship between α and β, i.e., that the regularity of the solution increases with higher values of α.

5.1.3 Individual results with $\sigma = u$

Noise terms

Finally, we consider the model in (1.1) driven by multiplicative noise, $u\,dW$, where u is the solution to the model. In Figure 5.7, we show the noise process at the 30^{th} time interval, where $\alpha = 0$ and $\alpha = 1$. It is not possible to generalize about the nature of multiplicative

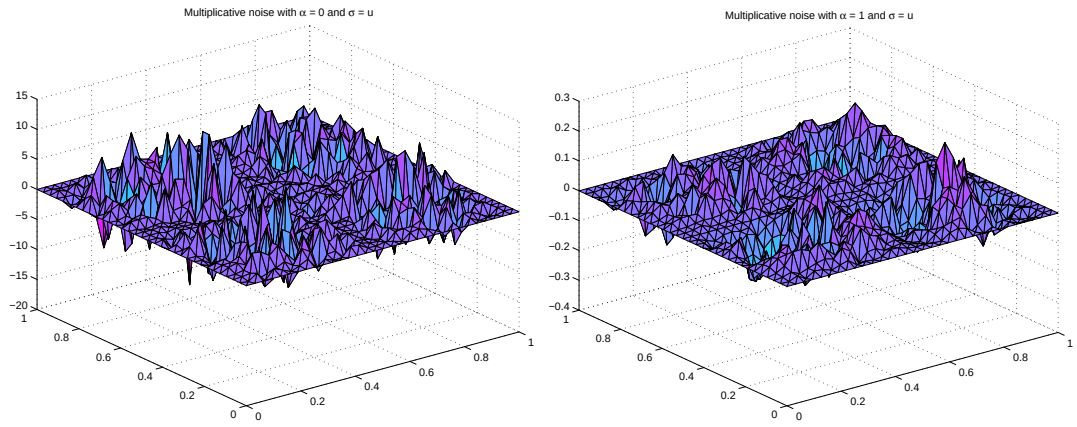

Figure 5.7: Multiplicative noise terms, with $\sigma = u$ for $\alpha = 0$ and $\alpha = 1$.

noise by looking at a single example. However, we will make a few observations based on the figure above. When we compare Figures 5.1 and 5.7, we note that the noise process in Figure 5.7 seems more regular. There appears to be some covariance between the values in adjacent node points of the multiplicative noise process. This is even more apparent when $\alpha = 1$. The noise ranges between 10.9 and -15.6 when $\alpha = 0$ and 0.23 and -0.28 when $\alpha = 1$, which is a larger range than when $\sigma = I$. Furthermore, we notice the similarity in appearance of the noise process and the finite element solution. That is, the spikes of the noise process are largest where the absolute value of the solution is greatest.

Finite element solutions

Below, in Figures 5.8 and 5.9, we observe the finite element solution at two different time points, at the 30^{th} interval and at the end of the time interval, respectively.[†]

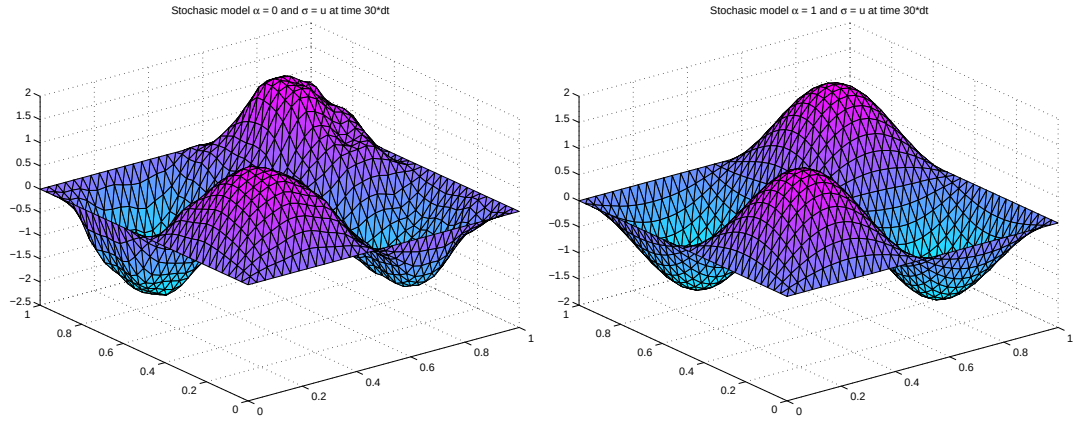

Figure 5.8: Examples of finite element solutions to the mathematical model at time $t = 30 * dt$ with $\sigma = I$ and $\alpha = 0$ (plot on the left) and $\alpha = 1$ (plot on the right).

[†]The noise terms in Figure 5.7 generate the solutions in the 30^{th} time interval, displayed in Figure 5.8.

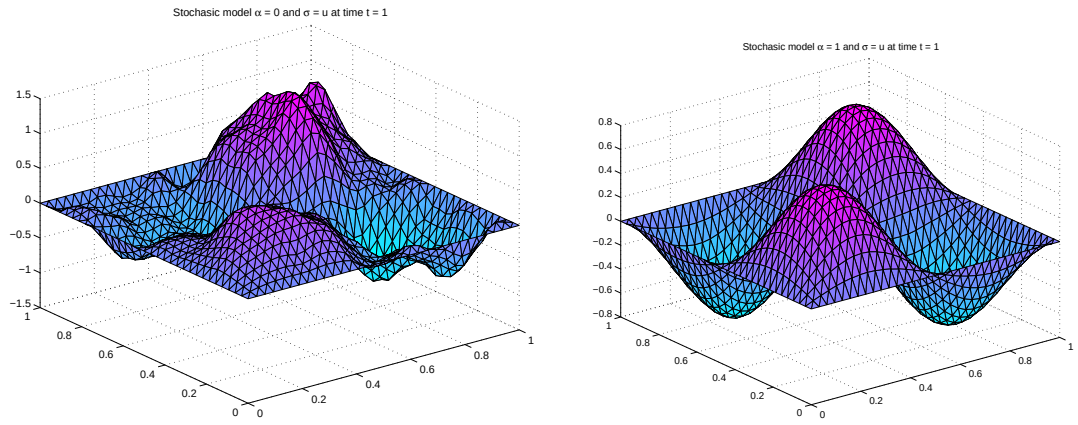

Figure 5.9: Examples of finite element solutions to the mathematical model at time $t = 1$ with $\sigma = I$ and $\alpha = 0$ (plot on the left) and $\alpha = 1$ (plot on the right).

We note, that the solution becomes more irregular when we reach the end of the time interval but the range of the solution process decreases. As before, the regularity of the solution increases with a higher value of α.

5.2 Results of the experiments

Below, we present the results of the numerical experiments that were outlined in Section 4.2.

5.2.1 Main numerical experiment

In this experiment, we run 1000 simulations on a grid with mesh size $h = 2^{-5}$ and time steps $k_i = 2^{-i}$, where $i = 1, \ldots, 8$. We compute finite element solutions for each simulation for six different values of α, with $\sigma = I$.

Strong convergence rate found by a least squares fit

If the error due to spatial discretization is sufficiently small, such that the convergence rate is dominated by the time step, k, we can use the procedure described in Subsection 4.1.1 to compute the convergence rate. The convergence rate, δ, in (4.2) is found by a least squares fit. In Figure 5.10, we plot δ for all six values of α. We also plot the expected convergence rate, together with the residuals of the ordinary least squares (ols) method. In Figure 5.11, we plot the errors, computed according to (4.4), on a log-log scale against k for four values of α, which should give us a line with slope δ. We also plot a reference line with slope α.

Although the convergence rates are reasonably close to the expected convergence rate, the residuals from the ols analysis are quite large, which indicates that the fit is not very good. A possible reason is that the initial assumption that the convergence rate is dominated by the time step is not fully valid. We note, that the spatial step in this experiment is not that small.

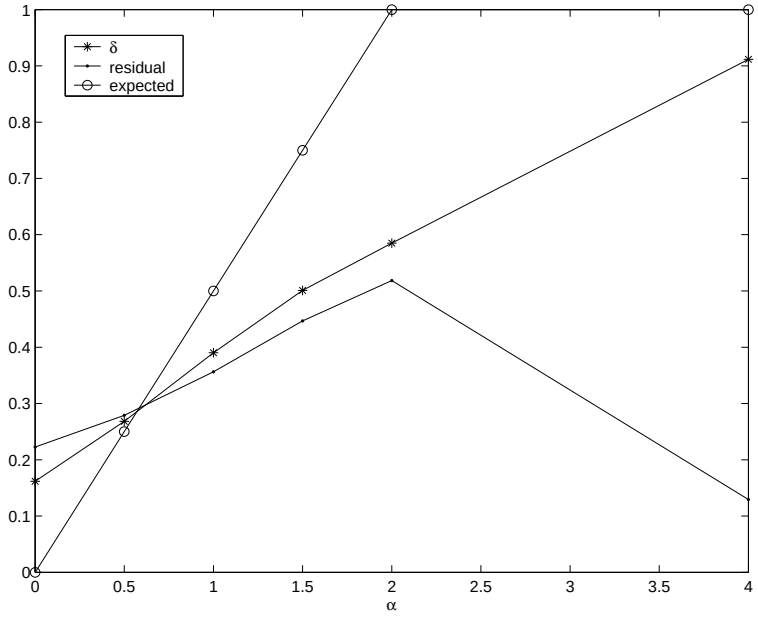

Figure 5.10: The convergence rate, δ, of the numerical method, computed by ols, together with the residual from the ols computation, and the expected convergence rate.

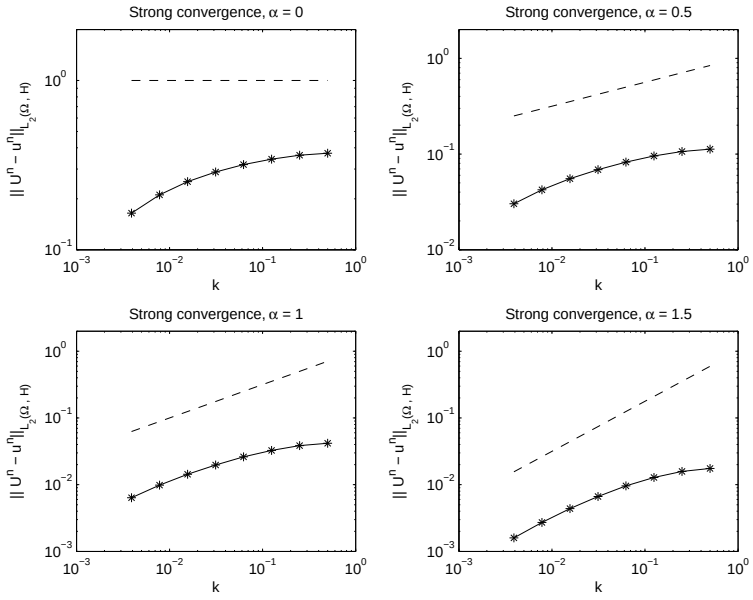

Figure 5.11: Logarithm of the errors plotted against the log of the time step, k, for four values of α. The dashed line shows the expected convergence rate.

Strong convergence rate found by computing error ratios

In Table 5.1, we display the computed error ratios $S(k_i)/S(k_{i+1})$, as introduced in Section 4.3.1 and in Table 5.2, we show the corresponding values of β.

		$S(k_i)/S(k_{i+1})$					
i	k_i	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 4$
1	2^{-1}	1.0286	1.0576	1.0855	1.1071	1.1295	2.0277
2	2^{-2}	1.0545	1.1136	1.1792	1.2331	1.2728	1.8308
3	2^{-3}	1.0774	1.1570	1.2528	1.3384	1.4025	1.7690
4	2^{-4}	1.1045	1.1994	1.3223	1.4444	1.5430	1.8297
5	2^{-5}	1.1411	1.2431	1.3771	1.5185	1.6379	1.8917
6	2^{-6}	1.1975	1.3084	1.4518	1.6064	1.7388	1.9646
7	2^{-7}	1.2818	1.3986	1.5451	1.7032	1.8378	2.0369

Table 5.1: Strong convergence - error ratios for 6 different values of α.

		β					
i	k_i	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 4$
1	2^{-1}	0.0813	0.1615	0.2368	0.2934	0.3513	2.0397
2	2^{-2}	0.1530	0.3106	0.4756	0.6045	0.6961	1.7450
3	2^{-3}	0.2150	0.4209	0.6502	0.8409	0.9759	1.6459
4	2^{-4}	0.2869	0.5246	0.8060	1.0610	1.2515	1.7432
5	2^{-5}	0.3807	0.6279	0.9233	1.2053	1.4237	1.8393
6	2^{-6}	0.5199	0.7757	1.0757	1.3676	1.5962	1.9485
7	2^{-7}	0.7164	0.9680	1.2555	1.5365	1.7560	2.0527

Table 5.2: Strong convergence - β-s corresponding to error ratios in Table 5.1.

It is difficult to reach any conclusion about the convergence rate from the tables above. Admittedly, the expected value of β, $\beta = \alpha$, is always somewhere between the highest and lowest value in Table 5.2.[‡] However, we cannot conclude that these results either confirm or disprove the theoretical error estimates.

When we compute β in Table 5.2, we assume that the error estimates are dominated by k. Our previous results indicate that this is overly simplified, since our computation mesh is not very fine. If we instead assume that the predicted rate of convergence is according to $O(k^{\beta/2} + h^{\beta})$, the error ratio becomes

$$\frac{S(k_i)}{S(k_{i+1})} \approx \frac{k_i^{\beta/2} + h^{\beta}}{k_{i+1}^{\beta/2} + h^{\beta}}, \tag{5.1}$$

and we get the values of β in Table 5.3. Computing β according to (5.1) gives us higher convergence rate than before. Still, we cannot comment much on the validity of the error estimate in Theorem 4.3.2.

[‡]Except in the special case when $\alpha = 0$, where we do not have an expected convergence rate of the method.

		β					
i	k_i	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 4$
1	2^{-1}	0.1357	0.2409	0.3270	0.3873	0.4458	2.0447
2	2^{-2}	0.2359	0.4158	0.5808	0.7021	0.7863	1.763
3	2^{-3}	0.3197	0.5443	0.7662	0.9420	1.0650	1.6834
4	2^{-4}	0.4164	0.6708	0.9406	1.1746	1.3480	1.8004
5	2^{-5}	0.5439	0.8066	1.0920	1.3525	1.5516	1.9314
6	2^{-6}	0.7337	1.0053	1.2993	1.5725	1.7822	2.1033
7	2^{-7}	1.0143	1.2867	1.5777	1.8494	2.0561	2.3308

Table 5.3: Strong convergence - β-s corresponding to error ratios according to (5.1).

Weak convergence rate found by a least squares fit

Below, we present the weak convergence rates found by a least squares fit for the functions p_1, p_2, p_3 and p_4[§], see Subsection 4.1.2. Since we do not have any theoretical estimates about

		$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 4$
p_1	δ	0.2481	0.3551	0.3794	0.4989	1.3271	1.0547
	residual	0.3192	0.4223	0.8308	2.4156	0.3581	0.0615
p_2	δ	0.2321	0.3570	0.3832	0.5001	1.3423	1.0550
	residual	0.3108	0.4315	0.8160	2.0749	0.3693	0.0622
p_3	δ	0.5892	0.6531	0.5888	0.6241	0.6988	0.9920
	residual	0.9081	1.5041	0.9731	0.8088	0.6947	0.1818
p_4	δ	0.2773	0.4339	0.5600	0.7681	0.8381	0.9940
	residual	0.2784	0.3678	0.5190	2.4514	1.7345	0.1810

Table 5.4: Weak convergence rate estimated by ols, together with the ols residuals for the functions p_1 to p_4.

the weak convergence rate, we guess that it is of order $O(k^\beta)$, if the time step dominates the convergence. However, we cannot reach any conclusion based on the results above, since the fit of the ols method is quite bad. More theoretical groundwork must be done before we can expect to obtain reasonable numerical results on the weak convergence rate of the numerical method. Thus, we do not proceed any further with the weak convergence rate of the method.

5.2.2 Experiment with $\sigma = I$ on a finer mesh

The purpose of this experiments is to try to assess whether using a finer mesh will improve the results. Thus, we run 1000 simulations on a mesh with $h = 2^6$, instead of $h = 2^5$ as before. Due to the long run times of the solver, we only perform the experiment for $\alpha = 1$. In Table 5.5, we display the convergence rate estimated by ols. The convergence rate that we obtain with the ols method does not differ much from our previous results on a coarser mesh. In Table 5.6, we display the values of β, computed according to (4.8). We note that

[§]For p_3 and p_4 we have that $x_0 = (0.719, 0.531)$.

	$\alpha = 1$
δ	0.3850
residual	0.3395

Table 5.5: Convergence rate estimated by ols, together with the ols residuals.

		β
i	k_i	$\alpha = 1$
1	2^{-1}	0.2578
2	2^{-2}	0.44098
3	2^{-3}	0.65973
4	2^{-4}	0.81602
5	2^{-5}	0.91968
6	2^{-6}	1.0389
7	2^{-7}	1.1953

Table 5.6: Strong convergence - β-s according to (4.8)

the values of β lie in a closer range than before (see Table 5.2). This could be an indication that using a finer space mesh will improve the results.

5.2.3 Experiment with $\sigma = \|x\|$

Here, we perform 1000 simulations for two values of α, $\alpha = 0$ and $\alpha = 1$. We use a grid with mesh size $h = 2^{-5}$ and time steps $k_i = 2^{-i}$, where $i = 1, \ldots, 8$. In Table 5.7, we present the convergence rate of the numerical method, estimated by ols. Just as in the other experiments,

	$\alpha = 0$	$\alpha = 1$
δ	0.1580	0.3793
residual	0.2239	0.3627

Table 5.7: Convergence rate estimated by ols, together with the ols residuals.

the high residuals indicate that the assumption that the time step dominates the convergence process is inaccurate.

The β-s in Table 5.8 are very similar to the β-s in the main experiment. This indicates that taking $\sigma = \|x\|$ instead of $\sigma = I$ does not greatly affect the convergence rate of the numerical method. Again, we cannot conclude anything about whether or not the convergence rate is in accordance to the theoretical results.

		β	
i	k_i	$\alpha = 0$	$\alpha = 1$
1	2^{-1}	0.0740	0.2191
2	2^{-2}	0.1418	0.4459
3	2^{-3}	0.2061	0.6202
4	2^{-4}	0.2799	0.7794
5	2^{-5}	0.3750	0.9066
6	2^{-6}	0.5135	1.0610
7	2^{-7}	0.7088	1.2471

Table 5.8: Strong convergence - β-s according to (4.8)

5.3 Discussion of the results

The main conclusion that we can draw from the experiments in Section 5.2 is that we can neither reject nor confirm the error estimates in Theorems 4.3.1 and 4.3.2. This motivates us to seek to improve the numerical procedure.

Our results on the strong convergence rate of the numerical method were inconclusive. Although the convergence rate from the ols method were not very far from the predicted rates, the residuals were large, which indicates a bad fit of the method. Furthermore, the β-s that we obtained from (4.8) were on a wide range.

The two methods that we used to compute the convergence rate are based on the assumption that the time step, k, determines the convergence process. The results indicate that this is not the case here. This could be expected, since the experiments are not performed on a very fine space mesh. The experiment that was run on a finer space mesh added weight to the claim that we will obtain better estimates of the convergence rate when we use finer spatial discretization, since we obtained β-s in a closer range than before.

It would be informative to perform further numerical experiments, in order to gain more knowledge about the reason why we are not able to reach a conclusion about the convergence rates. We would, in particular, like to try an even finer spatial mesh and also to perform more simulations to diminish the sampling error.

Chapter 6

Conclusions

In essence, we have achieved our principal aim, i.e., to implement a finite element solver for the stochastic heat equation in (1.1). The solver can provide finite element approximations to stochastic partial differential equations with either additive or multiplicative noise terms, and will without doubt be useful in further work.

There remains, however, some uncertainty regarding the ability of the solver to give accurate solutions to the mathematical model. This is due to our inability to reach a conclusion about the convergence rate of the numerical method. In other words, our numerical experiments aimed to verify theoretical error estimates do not produce satisfactory results.

In deriving and implementing the finite element solver, we apply implementation methods that are normally used for deterministic partial differential equations (pde's). An important conclusion of this thesis is that these methods do not seem to be adequate in the case of the stochastic model.

Even though this is a disappointing conclusion, this work is probably been a necessary step in the development process for the finite element solver. Possessing the knowledge that the straight-forward implementation method does not work in its current form, we can now concentrate on identifying where the method lets us down and attempt to develop and implement amendments.

Although a thorough investigation is out of the scope of this thesis, we attempt to point out possible weaknesses in the numerical procedure. We identify error sources of which we believe sampling error and quadrature error to be the most significant, even though we do not know the full extent of the quadrature error.

While there are some unanswered questions about the quality of our solver, we can still take many positives from our work. One is that we have taken the first step in the exploration of the theoretical framework surrounding numerical approximation of parabolic spde's. Even though this is a vast subject, which cannot be thoroughly investigated in a master's thesis, we have made some progress and achieved a better understanding of the mathematical model in (1.1) in return.

Another positive is that numerical experiments with the finite element solver have given us a better insight into the mathematical model. By generating instances of the noise term and the corresponding solution of the spde, we have seen how the noise is affected by parameter changes and how these effects spill over to the solution. An increased understanding of the model is advantageous, in particular when organizing further numerical experiments.

Last, but not least, we have identified some possible extensions to this work, which we will discuss in more detail in Section 6.1.

It should be noted that the fact that our numerical experiments did not yield the expected results added some unexpected complications and complexity to the project. Consequently, the task of writing this thesis became more difficult and demanding, since it is clearly much easier to interpret results which are in accordance to the theoretical theorems.

When the numerical results failed to comply with the expected results, a careful examination of the computer code and the whole procedure followed. We were able to point out possible weaknesses of the numerical approach, which we must leave for others to look closer into.

6.1 Possible extensions

We have identified several possible extensions to the work that is described in this thesis. The most interesting, in our opinion, are the following:

1. **Implementation of the solver in a general domain in $\mathbb{R}^2$.** This involves the computation of the eigensystem of the Laplacian operator in the domain in question. We recap this estimation procedure in Subsection 3.3.2, but expect the actual implementation to be worthy of another master's thesis.

2. Using **wavelet multiresolution** to approximate the noise term of the spde. This method is promising, particularly since it is applicable in general domains. However, we have not examined this method in any detail.

3. **Implementation of a solver in cubic domains in $\mathbb{R}^3$.** Although this is an interesting project, our problems regarding run times and memory requirements should be much more severe than in two dimensions. Also, before going to three dimensions we must first sort out our problems with the finite element solver in two dimensions.

4. **Derive the magnitude of the quadrature error.** As stated in Subsection 4.1.2, quadrature error is possibly a very important source of error in our computations. It might be a major factor in why we do not observe the expected convergence rate of the numerical method. Therefore, it is important to derive a theoretical error bound for the quadrature error. If necessary, the next step will then to implement some measures to correct for this error.

5. Address the issue of **sampling error** in the computations. As discussed in Subsection 4.1.2, the sampling error declines at the rate $1/\sqrt{M}$, where M is the number of simulations. To decrease this error, we must therefore perform more simulations. This means, in turn, that we must either run our computations on a coarser space mesh or decrease the run time of the solver. However, the former option is not clearly not feasible in general. We decrease the run time of the solver either by improving the existing Matlab code or moving the implementation to another computer language. Although there may be some room for improvement in the Matlab code, we expect that implementing the solver in C++ or Fortran will be more effective and should result in significant improvements.

6. It would be interesting to perform **further numerical experiments with the finite element solver**. In particular on the spde with a multiplicative noise term.

Although the extensions that we have described above are very interesting, most of them require us to be confident about the finite element solver in two dimensions. Thus, our priority should clearly be to identify and address the weaknesses of the current finite element solver. Items 4 and 5 in the list above might be the first steps in this procedure.

Bibliography

[1] G. Da Prato and J. Zabczyk. *Stochastic Equations in Infinite Dimensions*. Cambridge University Press, 1992.

[2] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Studentlitteratur, 1996.

[3] V. Heuveline. On the computation of a very large number of eigenvalues for selfadjoint elliptic operators by means of multigrid methods. *Journal of Computational Physics*, 184/1:321–337, 2003.

[4] D.J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43:526–546, 2001.

[5] F.C. Klebaner. *Introduction to Stochastic Calculus with Applications*. Imperial College Press, 1998.

[6] S. Larsson. Numerical methods for stochastic ODEs. Lecture notes. Department of Mathematical Sciences, Chalmers University, Göteborg, 2004.

[7] S. Larsson and V. Thomée. *Partial Differential Equations with Numerical Methods*. Springer, 2003.

[8] T. Müller-Gronbach and K. Ritter. Lower bounds and non-uniform time discretization for approximation of stochastic heat equations. Preprint, 2005.

[9] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*. Springer, 1997.

[10] Y. Yan. Semidiscrete Galerkin approximation for linear stochastic parabolic partial differential equation driven by an additive noise. *BIT*, 44:829–847, 2004.

[11] Y. Yan. Galerkin finite element methods for stochastic parabolic partial differential equations. *SIAM J. Numer. Anal.*, 43:1363–1384 (electronic), 2005. ISSN 0036-1429.