

Numerical solution of SPDEs on a class of two-dimensional domains and of random differential equations using PCE with exponential time differencing

von der Fakultät 1 - MINT - Mathematik, Informatik, Physik, Elektro- und Informationstechnik der Brandenburgischen Technischen Universität Cottbus–Senftenberg genehmigte Dissertation zur Erlangung des akademischen Grades eines Dr. rer. nat

vorgelegt von

Julian Clausnitzer

geboren am 22.12.1995 in Hannover

Vorsitzender: Prof. Dr. Michael Oevermann

Gutachter: Prof. Dr. Andreas Kleefeld

Gutachter: Prof. Dr. Michael Breuß

Gutachterin: Prof. Dr. Annika Lang

Tag der mündlichen Prüfung: 04.12.2024

Abstract

The first part of this thesis deals with the numerical solution of semilinear parabolic stochastic partial differential equations on general two-dimensional \mathcal{C}^2 -bounded domains. The existing exponential Euler time stepping scheme is used on two-dimensional domains, using a spectral approximation in space. Since the base functions are not given analytically, they are numerically approximated using a boundary element method combined with a contour integral method to solve nonlinear eigenvalue problems. An error analysis is given, and numerical experiments conclude the investigation of the method.

The second part of the thesis compares the performance of non-intrusive and intrusive polynomial chaos expansion methods based on exponential time differencing schemes for a range of random ordinary and partial differential equations. It is shown in comprehensive numerical experiments that the two approaches are competitive for a range of different equations, but intrusive polynomial chaos becomes less efficient for polynomial nonlinearities of degree greater than two and breaks down for a reaction-diffusion equation exhibiting complex pattern formation behavior.

Zusammenfassung

Im ersten Teil der vorliegenden Arbeit wird die numerische Lösung semilinearer parabolischer stochastischer Differentialgleichungen auf zweidimensionalen \mathcal{C}^2 -berandeten Gebieten untersucht. Das exponential Euler-Verfahren aus der Literatur wird auf zweidimensionale Gebiete angewandt, wobei eine spektrale Approximation im Raum benutzt wird. Da die Basisfunktionen nicht analytisch gegeben sind, müssen sie numerisch approximiert werden, wofür eine Randelementmethode in Kombination mit einer Konturintegralmethode zur Lösung nichtlinearer Eigenwertprobleme benutzt wird. Nach der Durchführung einer Fehleranalyse wird die Untersuchung dieser Methode mit numerischen Experimenten abgeschlossen. Der zweite Teil der Arbeit vergleicht die Leistungsfähigkeit nicht-intrusiver und intrusiver polynomieller Chaos-Methoden unter Benutzung exponentieller Integratoren für eine Reihe zufälliger ordinärer und partieller Differentialgleichungen. Es wird in umfangreichen numerischen Experimenten demonstriert, dass die zwei Methoden für die Lösung verschiedener Gleichungen vergleichbar performant sind. Die intrusive polynomielle Chaos-Methode ist jedoch weniger effizient für polynomielle Nichtlinearitäten vom Grad größer als zwei und liefert keine korrekten Ergebnisse für Reaktions-Diffusionsgleichungen, deren Lösungen komplexe Musterbildung aufweisen.

Acknowledgements

First and foremost, I express my utmost gratitude to my supervisor Prof. Andreas Kleefeld for his guidance and encouragement throughout all stages of my PhD project. I consider myself quite lucky to have a supervisor who has always been available for questions, discussions and honest feedback.

Especially during the first half of my PhD project, as a result of the limited social contacts during the pandemic it was very difficult to get in touch with people. Having just moved to a new place, getting used to the daily struggles of a doctoral researcher surely was a challenge. I am all the more grateful to my family for providing a safe haven to fall back to, and for giving their unwavering support. I also want to give a shout-out to the DocTeam of Forschungszentrum Jülich, who have worked very hard to make life for all doctoral researchers brighter, fairer and more cheerful, both during and after the pandemic.

I have thoroughly enjoyed working in my office largely thanks to my awesome officemates Mehdi and Thomas. Our ping pong breaks and tea breaks have always been very welcome occasions for procrastination. I also thank the Mathematics and Education department of the Jülich Supercomputing Centre for the financial support and for providing an overall very enjoyable working environment.

I also want to thank my friends Mehdi and Josef for all the good times outside the office, and my friend Karl for an amazing trip to Oxford. A big thank you also goes to my neighbors Jörg, Zien and Daniel who made our ‘FZJ house’ a very cozy and familiar place to live in.

Finally, I want to thank some trusted and wonderful ‘old’ friends from Göttingen and Hanover: Alina, Anna, Eric, Fred, Gabriel, Henning, Jun, Markus, Ramona and Valentin.

Contents

List of Figures	1
List of Tables	5
List of Algorithms	7
1 Scope of the thesis	9
2 Stochastic partial differential equations - Mathematical preliminaries	11
2.1 Wiener processes and stochastic analysis on Hilbert spaces	12
2.1.1 Normal distributions on Hilbert spaces	12
2.1.2 Hilbert space valued Wiener processes	13
2.1.3 The stochastic integral on Hilbert spaces	14
2.2 Analytic semigroups and parabolic SPDEs	17
2.2.1 \mathcal{C}_0 -semigroups	19
2.2.2 Analytic semigroups	21
2.2.3 Existence and uniqueness of mild solutions	23
2.3 Numerical solution of parabolic SPDEs	28
2.3.1 Different schemes in the literature	28
2.3.2 The exponential Euler scheme	30
2.3.2.1 Approximation in space.	32
2.3.2.2 Approximation in time.	32
2.4 Numerically approximating Dirichlet eigenvalues and eigenfunctions: The boundary integral method and Beyn's contour integral algorithm	33
3 Solving semilinear parabolic SPDEs on two-dimensional domains with a \mathcal{C}^2 boundary	39
3.1 The exponential Euler scheme for two-dimensional domains	39
3.2 Preprocessing: Computation of the orthonormal basis	40
3.2.1 Convergence and accuracy of base functions.	41

3.2.2	Computation of the orthonormal basis.	42
3.3	Error analysis	47
3.4	Numerical experiments	56
4	Polynomial chaos and exponential time differencing - Mathematical preliminaries	61
4.1	The polynomial chaos expansion	62
4.1.1	Using shifted Legendre polynomials	66
4.2	Numerical treatment of PDEs with uncertainties using PCE	68
4.2.1	Non-intrusive PCE	68
4.2.1.1	The classical Monte Carlo method	70
4.2.1.2	The quasi-Monte Carlo method	70
4.2.1.3	Gaussian quadrature	71
4.2.2	Intrusive PCE	72
4.3	Numerical schemes for deterministic equations	73
4.3.1	Explicit Euler scheme	74
4.3.2	ETD and ETD-RDP-IF	75
4.3.3	ETDRK4	79
5	Solving random differential equations using polynomial chaos with exponential time differencing	83
5.1	Intrusive PCE schemes	83
5.1.1	Explicit Euler	87
5.1.2	ETD-RDP and ETD-RDP-IF	87
5.1.3	ETDRK4	89
5.2	Numerical experiments.	91
5.2.1	One-dimensional random equation with linear term	92
5.2.2	One-dimensional random equation with quadratic term	93
5.2.3	One-dimensional random equation with cubic term	97
5.2.4	Two-dimensional random equation with linear term	97
5.2.5	Performance plots	100
5.2.6	Random Gray-Scott model	100
6	Summary and Outlook	111
6.1	SPDEs on two-dimensional domains	111
6.2	Polynomial chaos with exponential time differencing	112

List of Figures

2.2.1	For a given $\omega \in \mathbb{R}$, $\theta \in (0, \pi)$ and $\varepsilon > 0$, the curves $\gamma_{\varepsilon, \theta}^+$, $\gamma_{\varepsilon, \theta}^-$ and $\gamma_{\varepsilon, \theta}^0$ and the sector $S_{\omega, \theta}$ (left-hand side). In this work, it is assumed that $\theta \in (\frac{\pi}{2}, \pi)$ and $\rho(A) \subset S_{\omega, \theta}$ or, equivalently, $\sigma(A) \subset \mathbb{C} \setminus S_{\omega, \theta}$ (right-hand side).	19
2.2.2	Diagram from [36, p. 108] showing the relations between a sectorial operator A with domain $D(A)$, its resolvent $R(\lambda, A)$ and its generated analytic semigroup $(S(t))_{t \geq 0}$	22
3.2.1	In the first step, the eigenvalues are still unknown, so the real axis up to a certain point is covered with contours $\hat{\gamma}_i$ which cover the real axis up to the point where the desired number N of eigenvalues has been found. For the subsequent, more precise computation, contours γ_i , $i = 1, \dots, N$ around the found eigenvalues are chosen.	42
3.2.2	Convergence results for two eigenfunctions of the Peanut shape (on the left, the first one, on the right, the 56th one), in red, the error is measured with respect to reference eigenvalues and eigenfunctions computed for $n_f^{\text{ref}} = 600$ boundary elements, in green, the reference data is computed for $n_f^{\text{ref}} = 1200$ boundary elements.	44
3.2.3	This plot shows, for different numbers of boundary elements n_f , absolute errors for the 300th eigenvalue and its corresponding eigenfunction in green (left axis) as well as CPU time for their computation in seconds (right axis). It can be seen that the CPU time for the eigenvalues behaves like $\mathcal{O}(n_f^2)$, while the time for the eigenfunctions is $\mathcal{O}(n_f)$. The computing times are very similar for other eigenvalues and eigenfunctions.	45
3.2.4	The total computation time on our machine for an orthonormal basis consisting of N eigenvalues and eigenfunctions. Scenarios 1 to 3 correspond, from top to bottom, to the three pairs of error tolerances shown in Figure 3.2.5.	45

3.2.5	The number of boundary elements to approximate wavenumbers and their respective eigenfunctions to stay below certain error tolerances $\varepsilon^{(\lambda)}$ and $\varepsilon^{(\eta)}$. For each base function, we computed a reference solution with $n_f^{\text{ref}} = 600$ boundary elements and compared the same function with less boundary elements to it, with $n_f \in \{50, 75, 100, 150, 200, 300, 400, 500\}$ boundary elements used. The plots on the left show raw data, with the least number of required boundary elements. The plots on the right show linearly interpolated numbers of needed boundary elements based off the data from the plots on the left. We can observe the linear relationship between wavenumber and needed boundary elements n_f , and that the needed n_f increases as the error tolerances decrease (in red are the values which exceed the error tolerance for every tested n_f). The blue reference slopes are functions $f(x) = mx + b$, for $m \in \{7, 8.5, 10\}$, $b \in \{80, 65, 70\}$, from top to bottom.	46
3.4.1	Convergence tests for the Peanut shape. The first plot shows the convergence with respect to M for the linear function $f_1(x) = x$, the second plot shows the convergence with respect to N for $f_1(x) = x$ as well as $f_2(x) = \frac{1}{1+x^2}$. The dashed lines in blue are reference lines for an order of convergence equal to one.	58
3.4.2	The approximated solution of the SPDE (2.3.1) on the Peanut shape for $N = 400$, $T = 0.1$, $M = 100$ and the nonlinear functions (3.4.1) for $p \in \{-0.4, -0.2, 0.2, 0.4\}$ from top-left to bottom-right. It appears that the process is ‘dimmed down’ from using nonlinear functions with a peak of higher absolute value.	59
4.2.1	Three plots showing 100 points sampled in the unit square according to classical Monte Carlo (left), quasi-Monte Carlo with a Sobol sequence (middle) and Gauss-Legendre quadrature with different weights (right). . .	72
5.1.1	Relative runtimes R_N for $N = 0, \dots, 9$, where $R_0 := 1$. Each plot shows six curves for six different equations investigated in Section 5.2: Equation (5.2.1) with linear, quadratic or cubic F , with $D = 0$ or $D = 1$. Each data point is an average over the runtimes of ten identical iPCE simulations with $T = 0.1$ and $M = 10$ time steps.	91

5.2.1	Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.2) with linear term with diffusion constant $D = 0$. The exact mean is given by (5.2.3).	94
5.2.2	Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.2) with linear term with diffusion constant $D = 1$. The exact mean is given by (5.2.3).	95
5.2.3	Plots showing the relative time-dependent error for the same setup as in Figure 5.2.1, but for the variance computed according to equation (4.2.4) with $q = 10$ coefficient functions using formula (4.2.3). The large errors for MC and QMC are due to the variance being close to zero, so that the sampling error introduced in (4.2.3) plays a much bigger role (see (5.2.5)). The exact variance is given by (5.2.4).	96
5.2.4	Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.8) with quadratic term with diffusion constant $D = 0$. . .	98
5.2.5	Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.8) with quadratic term with diffusion constant $D = 1$. . .	99
5.2.6	Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.9) with cubic term with diffusion constant $D = 0$	101
5.2.7	Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.9) with cubic term with diffusion constant $D = 1$	102
5.2.8	Plots showing the error for the different equations at time $T = 2$ (except for equation 5.2.8 with the quadratic term and $D = 0$, where $T = 0.4$). The shorthands EEi, EEn in the legend stand for EE iPCE and EE niPCE, respectively, and analogously with RDP for ETD-RDP and RK4 for ETDRK4. For $D = 1$, only ETD-RDP and ETDRK4 are shown, since EE requires higher M in order to be stable.	103

5.2.9	Time-dependent L^2 errors for equation (5.2.10), using the iPCE algorithms 5.1, 5.3 and 5.4. We used the same parameters and numbers of time steps as in the 1D case from Figure 5.2.1 and Table 5.2.2. The exact expected value is given by (5.2.11). The spatial resolution in the ETD-RDP case was picked in this simulation as $p = 512$. For lower p , the error caused by the finite difference discretization dominates and the curves for different N are identical.	105
5.2.10	Time-dependent error plot for a random 1D Gray-Scott system. None of the iPCE schemes work in this case: Initially, the solution is relatively accurate, but as pattern formation starts to occur, iPCE breaks down. For a visual example in two dimensions, see Figure 5.2.12.	105
5.2.11	Deterministic simulations of Gray-Scott patterns for different schemes and resolutions for the same initial condition at $T = 5000$ (using Algorithms 4.1, 4.3, and 4.5). It becomes apparent that after a long-term integration, Gray-Scott patterns are extremely sensitive to the spatial resolution (and number of time steps), and for the correct solution, a high spatial resolution is needed, along with a scheme such as ETDRK4 which performs a spectral approximation in space. The numbers of time steps used are $M = 100000$ for EE and $M = 10000$ for ETD-RDP-IF and ETDRK4.	106
5.2.12	Simulations for $\mathbb{E}[u(x, T, \cdot)]$ of the random Gray-Scott system (5.2.12) with $F = 0.04$ and $k(\omega) \sim \mathcal{U}[0.058, 0.062]$. A niPCE simulation with ETDRK4 (left-hand side) and GQ gives an impression of superimposed patterns for different k . Over time, the four off-center bumps of the initial condition (5.2.13) expand and connect, forming intricate patterns as observed on the left-hand side. The EE iPCE simulation fails to correctly propagate the patterns, and the four rings from the initial condition stop expanding. For the ETDRK4 iPCE simulation, the edges of the rings propagate too fast, and no pattern formation is observed.	107
5.2.13	A diagram showing all the different MATLAB programs in the repository and how they are connected. The error plots in Figures 5.2.1 to 5.2.7, as well as Figure 5.2.9 are generated by <code>PCE_time_errorplot.m</code> for iPCE and by <code>Nonintrusive_PCE.m</code> for niPCE. The program <code>Performance_Plot.m</code> generates the plots in Figure 5.2.8. The programs in the repository contain more detailed descriptions.	107

List of Tables

4.1.1 Continuous probability distributions associated with families of orthogonal polynomials, table from [102, Table 4.1], with $\alpha > -1$, $\beta > -1$. Note that Legendre and Jacobi polynomials can be linearly rescaled so that they are supported on an arbitrary closed interval $[a, b] \subset \mathbb{R}$ (see also Section 4.1.1).	64
4.1.2 Weight function, support, recurrence coefficients occurring in (4.1.5) and the term occurring in the Rodriguez formula (4.1.6) for different families of orthogonal polynomials	65
5.1.1 Number \tilde{N} of summands in (5.1.14) for low N	86
5.2.1 Runtimes for the created plots, all times in seconds, for iPCE with $N = 5$ and for niPCE with 50 realizations	108
5.2.2 Numbers of step sizes M and of samples (for niPCE) for each simulation shown in Figures 5.2.1 to 5.2.7. ‘ETDRK4 ref.’ refers to the reference solution used for that simulation. For the linear equation with $D = 0$, the exact solution is known.	109

List of Algorithms

2.1	Exponential Euler scheme	33
2.2	Beyn contour integral algorithm	37
2.3	Beyn contour integral algorithm (continued)	38
4.1	Deterministic finite differences explicit Euler scheme	75
4.2	Deterministic finite differences ETD-RDP scheme	78
4.3	Deterministic finite differences ETD-RDP-IF scheme	79
4.4	Deterministic ETDRK4 time stepping scheme	82
4.5	Deterministic discretized Spectral ETDRK4 scheme	82
5.1	iPCE Explicit Euler scheme	87
5.2	iPCE ETD-RDP scheme	87
5.3	iPCE ETD-RDP-IF scheme	89
5.4	iPCE ETDRK4 scheme	90

Chapter 1

Scope of the thesis

This thesis will be concerned with two different kinds of partial differential equations (PDEs) which involve randomness: Stochastic partial differential equations (SPDEs) in the first part and initial value problems with random coefficients in the second part.

The first part of the thesis comprises Chapters 2 and 3. Parts of the chapters make up the paper [21] which has been accepted for publication. In this part, we deal with a stochastic PDE (SPDE) which contains a noise term defined as a Hilbert space-valued Wiener process. The novelty in this work is the numerical solution of an SPDE on a two-dimensional domain with a general \mathcal{C}^2 boundary. Most works in the literature on SPDEs assume simple spatial domains such as one-dimensional intervals or two-dimensional rectangles. In other theoretical works, a general domain in a Euclidean space is assumed with Dirichlet boundary conditions (see for example [12, 31, 32]). However, there are few works actually numerically solving an SPDE on a spatial domain which is more complicated than an interval or a rectangle using a spectral approach. The numerical solution of SPDEs on complicated domains was pointed out in [86] as a topic for future research. Two- or higherdimensional spatial domains have barely been investigated because they require more strict assumptions on the smoothness of the Wiener process to guarantee existence and uniqueness of a solution. Also, they combine difficulties arising in one spatial dimension, such as irregularity of the solutions and high or infinite dimension of the noise, with problems related to the curse of dimensionality, i.e. massively increased computational effort for higher dimensions. While recent works (see [22, 69]) deal with solving stochastic wave and heat equations on a two-dimensional sphere, we focus in this work on general two-dimensional planar domains which have a \mathcal{C}^2 boundary.

The necessary mathematical preliminaries on SPDEs are provided in Chapter 2. In Chapter 3, we demonstrate how to apply a spectral approach for SPDEs to \mathcal{C}^2 -bounded domains by numerically approximating the necessary base functions using a boundary element method. We show how the accuracy of the approximated base functions depends on the number

of used boundary elements. The numerical approximation of the base functions poses an additional error source for the used time stepping scheme. An error bound is also derived, and it is shown that it depends critically on the accuracy of the base function, reiterating the need for a highly accurate set of numerically approximated base functions.

The second part of the thesis consists of Chapters 4 and 5. The main results of the chapters have been submitted for publication, see also the preprint [20]. This part is concerned with the use of exponential time differencing schemes for non-intrusive and intrusive polynomial chaos methods. Exponential time differencing (ETD) schemes have turned out to be highly efficient and stable for solving a wide range of initial value problems such as reaction-diffusion problems – for an overview, see [54]. We investigate in this work to what extent these benefits also apply to random initial value problems, and a reaction-diffusion system with a random parameter. Both non-intrusive PCE (niPCE) and intrusive PCE (iPCE) have been applied to a variety of problems in fluid dynamics [79], sensitivity analysis [93], modeling of transport in porous media [43], reacting flow simulations [87] and many others. Intrusive polynomial chaos requires a significant modification of the underlying program code, since the PCE is inserted into the governing equation. Non-intrusive, sampling-based polynomial chaos methods rely on using deterministic solvers to repeatedly solve the deterministic problem resulting from sampling the input random variables. Given the program code of solvers for deterministic problems, niPCE is straightforward to implement. Its performance has been compared favorably to Monte Carlo methods [93]. Direct comparisons between an iPCE and niPCE can be found in [83, 92], indicating that intrusive methods can compete with non-intrusive methods as long as no strong nonlinearities and no long-term integration is involved. These problems with intrusive methods, as well as difficulties dealing with sharp dependencies in the stochastic parameter space, have also been noted in [34, p. 1] and [65, pp. 1], as well as in [42, 70, 100].

The necessary mathematical preliminaries on PCE are given in Chapter 4. In Chapter 5, we develop and implement niPCE and iPCE schemes based on three different schemes: A simple explicit Euler scheme and two ETD schemes. We compare the performance of iPCE and niPCE, for which naive Monte Carlo sampling, Quasi-Monte Carlo sampling and Gaussian quadrature is used. It is seen that niPCE and iPCE yield similar errors for random ordinary and partial differential equations involving a random linear or quadratic function. For a cubic random nonlinear function, however, iPCE becomes unstable, and for a random reaction-diffusion system only niPCE is viable. This reiterates the aforementioned problems of iPCE dealing with sharp dependencies in the stochastic parameter space and with complex behavior in random dynamical systems.

Chapter 2

Stochastic partial differential equations - Mathematical preliminaries

This chapter provides the theoretical framework for the first work of this thesis in Chapter 3, where we solve stochastic partial differential equations of the form

$$dU(t) = [AU(t) + F(U(t))]dt + dW^Q(t) \quad (2.0.1)$$

(see also (2.2.7)). In the course of this chapter, we explain the components of this equation part by part, we then define what a mild solution of the equation is and we then show existence and uniqueness of a mild solution.

The structure of this chapter is as follows. Firstly, we will recall in Section 2.1 normal distributions, Wiener processes and stochastic integrals in a Hilbert space setting. Gaussian measures and Q -Wiener processes W^Q are introduced in Sections 2.1.1 and 2.1.2 and the Hilbert space-valued stochastic integral is recalled in Section 2.1.3. The linear operator A in (2.0.1) is required to be a generator of a \mathcal{C}_0 -semigroup as well as an analytic semigroup. We give an introduction to semigroup theory and mild solutions in Section 2.2 and recall the concepts of \mathcal{C}_0 -semigroups (Section 2.2.1) and analytic semigroups (Section 2.2.2). It is then explained in Section 2.2.3 what a mild solution of equation (2.0.1) is and the existence and uniqueness of a mild solution is discussed under suitable assumptions on the operator A and the nonlinearity F . We then give an overview in Section 2.3 on how to numerically approximate solutions for the SPDE. An overview of the literature is given in 2.3.1 and in 2.3.2 we recall the exponential Euler scheme which we are going to use. Finally, in Section 2.4 it is explained how to numerically compute Dirichlet eigenvalues and eigenfunctions for a given domain. This is a necessary prerequisite for applying the exponential Euler scheme to a general two-dimensional domain, which we will do in Chapter 3.

2.1 Wiener processes and stochastic analysis on Hilbert spaces

It is well-known that the distribution of a real-valued, normally distributed random variable X is uniquely determined by its mean $\mu \in \mathbb{R}$ and its variance $\sigma^2 > 0$. In this case we write $X \sim \mathcal{N}(\mu, \sigma^2)$. Generalizing this to a vector-valued setting, the distribution of a normally distributed, \mathbb{R}^d -valued random variable is uniquely determined by its mean $\mu \in \mathbb{R}^d$ and its covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$. If we want to generalize normally distributed random variables to take values in a Hilbert space H , we would therefore expect their distribution to be determined by a mean $\mu \in H$ and a covariance operator $Q : H \rightarrow H$. This is exactly the case and will be made precise in Section 2.1.1.

2.1.1 Normal distributions on Hilbert spaces

First, we define Gaussian measures on a separable Hilbert space H , which will lead us to a generalization of the real-valued normal distribution. We denote by $L(H)$ the space of linear, bounded operators on H .

Definition 2.1.1 [72, 2.1.1]. Let H be a Hilbert space and $\mathcal{B}(H)$ the Borel σ -algebra on H . A probability measure μ on $(H, \mathcal{B}(H))$ is called Gaussian if for all $u \in H$, the linear mapping

$$\begin{aligned} v' : H &\rightarrow \mathbb{R} \\ u &\mapsto \langle u, v \rangle_H \end{aligned}$$

has a Gaussian law: For all $v \in H$, there are $m = m(v)$ and $\sigma = \sigma(v) \geq 0$ such that, if $\sigma > 0$,

$$(\mu \circ (v')^{-1})(A) = \mu(v' \in A) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_A e^{-\frac{(x-m)^2}{2\sigma^2}} dx$$

for all $A \in \mathcal{B}(\mathbb{R})$.

Theorem 2.1.2 [72, Theorem 2.1.2] *Let $\{e_n\}_{n \in \mathbb{N}}$ be an orthonormal basis of H and let $L(H)$ be the space of linear, bounded operators on H . A measure on $(H, \mathcal{B}(H))$ is Gaussian if and only if*

$$\hat{\mu}(u) := \int_H e^{i\langle u, v \rangle_H} \mu(dv) = e^{i\langle m, u \rangle_H - \frac{1}{2}\langle Qu, u \rangle_H}, \quad u \in H,$$

where $m \in H$ and $Q \in L(H)$ is nonnegative, self-adjoint, and has finite trace

$$\mathrm{tr}(Q) = \sum_{k=1}^{\infty} \langle Qe_k, e_k \rangle_H. \quad (2.1.1)$$

Remark 2.1.3 The trace (2.1.1) is independent of the chosen orthonormal basis $\{e_n\}_{n \in \mathbb{N}}$. Note that since Q is a nonnegative and self-adjoint operator, the condition $\mathrm{tr}(Q) < \infty$ is equivalent to Q being a trace class operator. An operator Q is a trace class operator if $\mathrm{tr}(|Q|) < \infty$, where $|Q|$ is the unique bounded positive operator on H such that $|Q| \circ |Q| = Q^*Q$.

We denote the measure μ in Theorem 2.1.2 by $\mathcal{N}(m, Q)$ where m is called the mean and Q is called the covariance operator. The following proposition shows how the normal distribution on a Hilbert space and its mean and covariance relate to their finite-dimensional counterparts.

Proposition 2.1.4 [72, Proposition 2.1.4] Let X be an H -valued Gaussian random variable on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, i.e. there exist $m \in H$ and $Q \in L(H)$ nonnegative, self-adjoint and with finite trace such that $\mathbb{P} \circ X^{-1} = \mathcal{N}(m, Q)$. Then $\langle X, u \rangle_H$ is normally distributed for all $u \in H$ and the following hold:

- $\mathbb{E}(\langle X, u \rangle_H) = \langle m, u \rangle_H$
- $\mathbb{E}(\langle X - m, u \rangle_H \cdot \langle X - m, v \rangle_H) = \langle Qu, v \rangle_H$

for all $u, v \in H$.

We now use the normal distribution on a Hilbert space H to define the H -valued Wiener process.

2.1.2 Hilbert space valued Wiener processes

Definition 2.1.5 [72, Definition 2.1.9]. Let H be a Hilbert space and let $Q : H \rightarrow H$ be an operator with finite trace. A H -valued stochastic process $W^Q(t), t \in [0, T]$, on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is called a (standard) Q -Wiener process if

- $W^Q(0) = 0$,
- W^Q has \mathbb{P} -a.s. (almost surely) continuous trajectories
- W^Q has independent increments
- The increments are normally distributed:

$$\mathbb{P} \circ \left(W^Q(t) - W^Q(s) \right)^{-1} = \mathcal{N}(0, (t-s)Q)$$

for all $0 \leq s < t \leq T$.

We will make use of a series representation of the Q -Wiener process given in the following proposition.

Proposition 2.1.6 [72, Proposition 2.1.10] *Let $\{e_n\}_{n \in \mathbb{N}}$, be an orthonormal basis of U consisting of eigenvectors of Q with corresponding eigenvalues λ_k , $k \in \mathbb{N}$. Then a U -valued stochastic process $(W^Q(t))_{t \in [0, T]}$ is a Q -Wiener process if and only if*

$$W^Q(t) = \sum_{k=1}^{\infty} \sqrt{\lambda_k} \beta_k(t) e_k, \quad t \in [0, T] \quad (2.1.2)$$

with equality in distribution, where β_k , $k \in \mathbb{N}$, are independent real-valued Brownian motions on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The series converges in $L^2(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{C}([0, T], U))$ (where $\mathcal{C}([0, T], U)$ is equipped with the sup norm) and thus always has a \mathbb{P} -a.s. continuous version. In particular, for every self-adjoint, nonnegative $Q \in L(U)$ with finite trace there exists a Q -Wiener process on U .

The series (2.1.2) will be crucial in the numerical treatment of SPDEs, for which the Q -Wiener process is simulated by the truncated series (2.1.2).

2.1.3 The stochastic integral on Hilbert spaces

For real-valued Wiener processes, a definition of the stochastic (Itô) integral can for example be found in [82, Chapter III.]. Since we work exclusively with stochastic processes taking values in a Banach space E , we will need the stochastic integral with respect to a Banach space-valued Q -Wiener process introduced in Section 2.1.2. We will define the stochastic integral in this section, leaning on [72, Section 2.3].

Let U and H be separable Hilbert spaces, $Q \in L(U)$ a symmetric, nonnegative covariance operator with finite trace, and $(W^Q(t))_{t \in [0, T]}$ a Q -Wiener process with respect to a normal filtration $(\mathcal{F}_t)_{t \in [0, T]}$. The construction of the stochastic integral proceeds in four steps:

1. A class \mathcal{E} of $L(U, H)$ -valued elementary processes $(\Phi(t))_{t \in [0, T]}$ is considered and a mapping

$$\begin{aligned} \text{Int} : \mathcal{E} &\rightarrow \mathcal{M}_T^2(H) \\ \Phi &\mapsto \int_0^t \Phi(s) \, dW^Q(s), \quad t \in [0, T] \end{aligned}$$

is defined, where $\mathcal{M}_T^2(H)$ denotes the Banach space of all H -valued, continuous, square-integrable martingales $(M(t))_{t \in [0, T]}$.

2. It is shown that there exists a certain norm on \mathcal{E} such that the mapping Int is an isometry. The mapping Int can then be uniquely extended to the (sequential) completion $\overline{\mathcal{E}}$, and the extension is also an isometry.
3. An explicit representation of $\overline{\mathcal{E}}$ is given.
4. The definition of the stochastic integral is extended by a process called localization.

Step 1: The definition of elementary processes and the mapping Int .

Definition 2.1.7 [72, Definition 2.3.1]. An $L(U, H)$ -valued stochastic process $(\Phi(t))_{t \in [0, T]}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with normal filtration $(\mathcal{F}_t)_{t \in [0, T]}$ is said to be elementary if there exist $0 = t_0 < \dots < t_k = T$, $k \in \mathbb{N}$, such that

$$\Phi(t) = \sum_{m=0}^{k-1} \Phi_m \mathbb{1}_{(t_m, t_{m+1}]}(t), \quad t \in [0, T],$$

where $\Phi_m : \Omega \rightarrow L(U, H)$ is \mathcal{F}_{t_m} -measurable w.r.t. the Borel σ -algebra on $L(U, H)$, and Φ_m only takes a finite number of values in $L(U, H)$, $0 \leq m \leq k-1$.

Now we define

$$\text{Int}(\Phi)(t) := \int_0^t \Phi(s) dW^Q(s) := \sum_{m=0}^{k-1} \Phi_m (W^Q(t_{m+1} \wedge t) - W^Q(t_m \wedge t)), \quad t \in [0, T]$$

for all $\Phi \in \mathcal{E}$. Then it follows (see [72, Proposition 2.3.2]) that for $\Phi \in \mathcal{E}$ the stochastic integral $\int_0^t \Phi(s) dW^Q(s)$, $t \in [0, T]$, is a square integrable martingale.

Step 2: Verifying that Int is an isometry.

Definition 2.1.8. Let $\{e_k\}_{k \in \mathbb{N}}$ be an orthonormal basis of U . An operator $A \in L(U, H)$ is called Hilbert–Schmidt if

$$\sum_{k \in \mathbb{N}} \langle Ae_k, Ae_k \rangle_H < \infty.$$

For an orthonormal base $\{e_k\}_{k \in \mathbb{N}}$, we can define the Hilbert–Schmidt norm

$$\|A\|_{L_2} := \left(\sum_{k \in \mathbb{N}} \|Ae_k\|^2 \right)^{1/2}, \quad (2.1.3)$$

which is independent of the choice of the orthonormal base $\{e_k\}_{k \in \mathbb{N}}$, and we denote

$$L_2(U, H) := \{A \in L(U, H) \mid \|A\|_{L_2} < \infty\}.$$

(Note the difference to L^2 spaces of square-integrable functions, which we denote with a superscript.) It is a fact (see [72, Proposition 2.3.4]) that for $Q \in L(U)$ nonnegative and symmetric, there is exactly one nonnegative and symmetric operator $Q^{1/2} \in L(U)$ such that $Q^{1/2} \circ Q^{1/2} = Q$. If Q has finite trace, then in addition for any $A \in L(U, H)$, $L \circ Q^{1/2} \in L_2(U, H)$.

With that, we can define a norm on \mathcal{E} with the following Proposition:

Proposition 2.1.9 [72, Proposition 2.3.5] *Let $\Phi = \sum_{m=0}^{k-1} \Phi_m \mathbb{1}_{(t_m, t_{m+1}]}$ be an elementary $L(U, H)$ -valued process. Then we have the Itô isometry for Q -Wiener processes:*

$$\left\| \int_0^\cdot \Phi(s) dW^Q(s) \right\|_{\mathcal{M}_T^2}^2 = \mathbb{E} \left(\int_0^T \|\Phi(s) \circ Q^{1/2}\|_{L_2}^2 ds \right) =: \|\Phi\|_T^2.$$

As defined above, $\|\cdot\|_T$ is only a seminorm, but becomes a norm after we consider \mathcal{E}' to be the space of equivalence classes in \mathcal{E} with respect to $\|\cdot\|_T$. We will in the following keep the notation unchanged, i.e. we denote $\mathcal{E} := \mathcal{E}'$.

Step 3: Providing an explicit representation of $\overline{\mathcal{E}}$. We introduce the subspace $U_0 := Q^{1/2}(U)$ with the inner product

$$\langle u_0, v_0 \rangle_0 := \langle Q^{-1/2}u_0, Q^{-1/2}v_0 \rangle_U,$$

$u_0, v_0 \in U_0$, where in case Q is not one-to-one, $Q^{-1/2}$ is the pseudo inverse as defined in [72, Definition C.0.1]. Then, $(U_0, \langle \cdot, \cdot \rangle_0)$ is again a separable Hilbert space. We denote $L_2(U_0, H) =: L_2^0$ with the norm $\|A\|_{L_2^0} = \|A \circ Q^{1/2}\|_{L_2}$ for $A \in L_2^0$.

It can then be shown (see [72, p. 33]) that an explicit representation of $\overline{\mathcal{E}}$ is given by

$$\mathcal{N}_W^2(0, T; H) := \{\Phi : [0, T] \times \Omega \rightarrow L_2^0 \mid \Phi \text{ is predictable and } \|\Phi\|_T < \infty\}$$

Step 4: Generalization of the stochastic integral. The stochastic integral is extended to the linear space

$$\mathcal{N}_W(0, T; H) := \left\{ \Phi : [0, T] \times \Omega \rightarrow L_2^0 \mid \Phi \text{ is predictable with } \mathbb{P} \left(\int_0^T \|\Phi(s)\|_{L_2^0}^2 ds \right) = 1 \right\},$$

which is called the class of stochastically integrable processes on $[0, T]$. From this point onward, all processes in stochastic integrals will be of this nature. For details on this extension process called localization, we refer to [72, pp. 36-37].

2.2 Analytic semigroups and parabolic SPDEs

When dealing with SPDEs, there are different kinds of solutions in the literature: Strong (or classical) solutions, weak solutions and mild solutions. Strong solutions are, in many cases, nonexistent, while weak solutions require a variational setting and are more frequently employed for elliptic and hyperbolic equations (see e.g. [72, 75]). The kind of solution we want to focus on here are mild solutions, which are used frequently to treat parabolic SPDEs. For a rigorous understanding of mild solutions, we will have to introduce concepts from semigroup theory.

The semigroup method is a powerful tool for solving evolution equations, and has been developed and widely used to show the existence of solutions on a range of differential equations on Banach spaces (for an overview, see e.g. [103, 36]). Likewise, for parabolic SPDEs the semigroup method has been of vital importance (see e.g. [19, 26, 27, 39, 41, 95]). Since classical solutions are often not available, using the notions of \mathcal{C}_0 -semigroups and analytical semigroups it is possible to define the weaker notion of mild solutions, which we recall in this chapter and which we use later in Chapter 3.

To give a motivation on why semigroups are a useful tool, consider that the initial value problem [36, Problem 1.1]

$$\begin{cases} \frac{d}{dt}S(t) = aS(t), & t \geq 0, \\ S(0) = 1 \end{cases}$$

for a constant $a \in \mathbb{C}$ is uniquely solved by the function $S(t) := e^{ta}$. At the same time, this family of exponential functions is unique up to the constant a in its property that $S(t+s) = S(t)S(s)$ and $S(0) = 1$ for all $s, t \geq 0$.

Likewise, the differential equation [36, Proposition 2.8]

$$\begin{cases} \frac{d}{dt}S(t) = AS(t), & t \geq 0, \\ S(0) = I_n \end{cases}$$

for $S(t) \in \mathbb{C}^{n \times n}$, the identity matrix $I_n \in \mathbb{C}^{n \times n}$ and a matrix $A \in \mathbb{C}^{n \times n}$ is (among all differentiable functions) uniquely solved by $S(t) := e^{tA}$, i.e. the matrix exponential of tA , and again these functions are unique in their property of satisfying the functional equation $S(t+s) = S(t)S(s)$, $S(0) = I$ for $s, t \geq 0$.

In the setting of a Banach space E with a linear operator $A \in L(E)$, there exists an associated one-parameter semigroup $(S(t))_{t \geq 0}$ which is generated by A (for details, see [36, Chapter

I.3]) and satisfies $S(t+s) = S(t)S(s)$ for $s, t \geq 0$. Then, the E -valued initial value problem

$$\begin{cases} \frac{d}{dt}u(t) = Au(t), & t \geq 0, \\ u(0) = x \end{cases}$$

is uniquely solved by $u(t) = S(t)x$ [36, p. 145].

These Banach space-valued one-parameter semigroups can, if they are \mathcal{C}_0 -semigroups (see Section 2.2.1) also be defined for unbounded operators A defined on a dense subset $D(A) \subset E$ (see [36, p. 47 ff.]). The operator A will, in our case, be the Laplacian Δ on $D(A) = H_0^1(D) \cap H^2(D) \subset E = L^2(D)$ with Dirichlet boundary conditions for a domain $D \subset \mathbb{R}^2$ with a \mathcal{C}^2 boundary. We will see that the collection $(S(t))_{t \geq 0}$ generated by Δ can be formalized as an analytic semigroup (see Section 2.2.2).

In the case where A is bounded, the family of operators $(S(t))_{t \geq 0}$ making up a semigroup can be given as the series expansion of the exponential function [36, p. 52]. Juxtaposing the exponential series for real numbers x and for a bounded operator A , it is

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}, \quad S(t) = \sum_{n=0}^{\infty} \frac{t^n A^n}{n!}, \quad t \geq 0.$$

In the general case where A is unbounded, for example for the Laplace operator $A = \Delta$, this series need not converge, so a different way to generalize the exponential function has to be found. Another approach is to make use of Cauchy's integral formula. Let f_1 be a scalar holomorphic function and let f_2 be a operator-valued holomorphic function on $U \subset \mathbb{C}$. Let γ be a closed contour in U and let A be an operator such that $\sigma(A)$ is contained in the interior of γ . There is a generalization of Cauchy's integral formula to the operator-valued function f_2 (see [33, p. 568]). Juxtaposing the traditional Cauchy's formula and the operator-valued Cauchy's formula, we have for some point a in the interior of γ that

$$f_1(a) = \frac{1}{2\pi i} \int_{\gamma} \frac{f_1(z)}{z-a} dz, \quad f_2(A) = \frac{1}{2\pi i} \int_{\gamma} f_2(\lambda) R(\lambda; A) d\lambda \quad (2.2.1)$$

with the resolvent operator $R(\lambda; A) = (\lambda I - A)^{-1}$ (for details and well-definedness, see Section 2.2.2). Using $f_2(x) = e^{tx}$ later gives rise to the definition of an analytic semigroup (see (2.2.3)).

In the following, for a Banach space E , integrals of E -valued functions will denote the so-called Bochner integral, which is a generalization of the Lebesgue integral to E -valued functions. For an introduction, see [36, Appendix C].

2.2.1 \mathcal{C}_0 -semigroups

As explained above, initial value problems (scalar as well as Banach space-valued) can be solved by a family of functions or operators $(S(t))_{t \geq 0}$ satisfying the property $S(t+s) = S(t)S(s)$, $S(0) = I$ known as the semigroup property. The idea being that families $(S(t))_{t \geq 0}$ might solve many more different evolution equations, this gives rise to the following definition of a \mathcal{C}_0 -semigroup.

Definition 2.2.1 [26, p. 406–407]. A \mathcal{C}_0 -semigroup is a family $(S(t))_{t \geq 0}$ of bounded operators on a Banach or Hilbert space H satisfying

1. $S(0) = I$, the identity operator.
2. $S(t+s) = S(t)S(s)$ for all $s, t \geq 0$.
3. $S(\cdot)u$ is continuous in $[0, \infty)$ for all $u \in H$.

The infinitesimal generator A of $S(\cdot)$ is defined as

$$Au = \lim_{h \searrow 0} \frac{S(h)u - u}{h}$$

for all $u \in H$ where this limit exists.

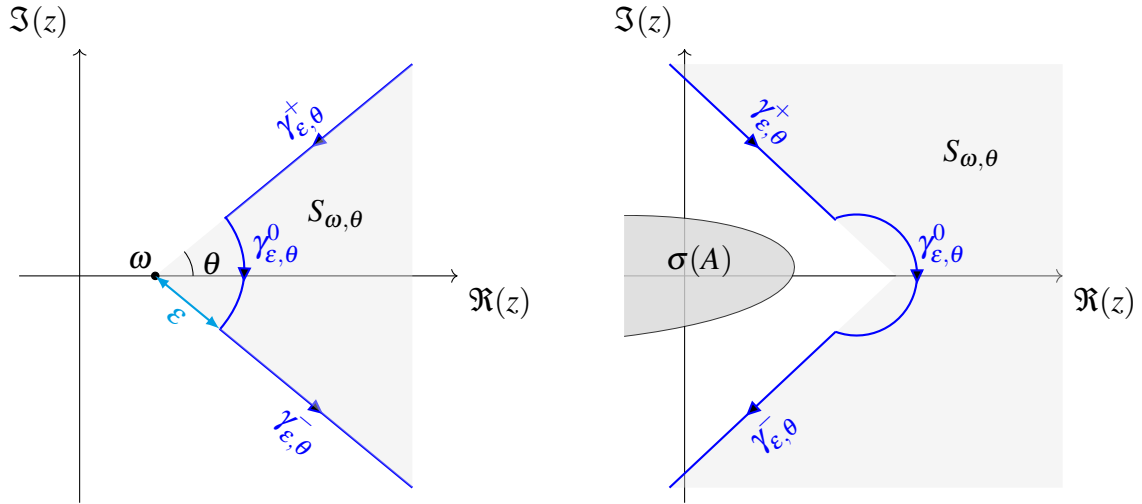


Figure 2.2.1: For a given $\omega \in \mathbb{R}$, $\theta \in (0, \pi)$ and $\varepsilon > 0$, the curves $\gamma_{\varepsilon, \theta}^+$, $\gamma_{\varepsilon, \theta}^-$ and $\gamma_{\varepsilon, \theta}^0$ and the sector $S_{\omega, \theta}$ (left-hand side). In this work, it is assumed that $\theta \in (\frac{\pi}{2}, \pi)$ and $\rho(A) \subset S_{\omega, \theta}$ or, equivalently, $\sigma(A) \subset \mathbb{C} \setminus S_{\omega, \theta}$ (right-hand side).

Definition 2.2.2 . The resolvent set $\rho(L)$ of the operator L is the set of complex numbers λ for which $\lambda I - L$ is bijective. For $\lambda \in \rho(L)$, the operator

$$R(\lambda, L) = (\lambda I - L)^{-1}$$

is called the resolvent operator of L . The complement of $\rho(L)$ in \mathbb{C} is called the spectrum $\sigma(L)$ of L .

For an $\omega \in \mathbb{R}$ and $\theta \in (0, \pi)$, we denote by $S_{\omega, \theta}$ the sector in \mathbb{C} (see also Figure 2.2.1)

$$S_{\omega, \theta} = \{\lambda \in \mathbb{C} \setminus \{\omega\} \mid |\arg(\lambda - \omega)| \leq \theta\}.$$

Assumption 2.2.3 1. *There exist $\omega \in \mathbb{R}$ and $\theta_0 \in (\frac{\pi}{2}, \pi)$ such that $\rho(A) \supset S_{\omega, \theta_0}$.*

2. *There exists $M > 0$ such that*

$$\|R(\lambda, A)\| \leq \frac{M}{|\lambda - \omega|} \quad \forall \lambda \in S_{\omega, \theta_0}.$$

An operator $A \in L(H)$ satisfying Assumption 2.2.3 is called sectorial operator.

The first part of Assumption 2.2.3 is illustrated in Figure 2.2.1, along with the definitions of $S_{\omega, \theta}$, $\gamma_{\varepsilon, \theta}^{\pm}$ and $\gamma_{\varepsilon, \theta}^0$.

The following theorem by Hille and Yosida gives a characterization of \mathcal{C}_0 -semigroups in terms of so-called Yosida approximations, which will be needed in Section 2.2.3 on the existence of solutions of the treated SPDEs.

Theorem 2.2.4 (Hille-Yosida) [26, Theorem A.4] *Let $A : D(A) \subset E \rightarrow E$ be a linear closed operator on the Banach space E . The following statements are equivalent:*

(i) *A is the infinitesimal generator of a \mathcal{C}_0 -semigroup $S(\cdot)$ such that*

$$\|S(t)\| \leq M e^{\omega t} \quad \text{for all } t \geq 0.$$

(ii) *$D(A)$ is dense in E , $\rho(A) \supset (\omega, +\infty)$ and it is*

$$\|R^k(\lambda, A)\| \leq \frac{M}{(\lambda - \omega)^k} \quad \text{for all } k \in \mathbb{N}.$$

Moreover, if either (i) or (ii) holds, then

$$R(\lambda, A)x = \int_0^\infty e^{-\lambda t} S(t)x \, dt \quad \text{for all } x \in D(A), \lambda > \omega.$$

Also, for all $x \in E$, $S(t)x = \lim_{n \rightarrow \infty} e^{tA_n}x$ and

$$\|e^{tA_n}\| \leq Me^{\frac{\omega nt}{n-\omega}}. \quad (2.2.2)$$

where $A_n = nAR(n, A)$. The operators A_n , $n > \omega$, are called the Yosida approximations of A .

2.2.2 Analytic semigroups

Suppose that $A \in L(E)$ is sectorial, i.e. Assumption 2.2.3 holds. In the beginning of Section 2.2.1, we remarked that the formal series expansion of the exponential function need not converge when applied to unbounded operators. Another possible way is to make use of the contour integral expression (2.2.1), which is done to define analytic semigroups.

For $\varepsilon > 0$ and $\theta \in (\frac{\pi}{2}, \theta_0)$, we denote by $\gamma_{\varepsilon, \theta}$ the path (see also Figure 2.2.1)

$$\begin{aligned} \gamma_{\varepsilon, \theta} &= \gamma_{\varepsilon, \theta}^+ \cup \gamma_{\varepsilon, \theta}^- \cup \gamma_{\varepsilon, \theta}^0, \\ \gamma_{\varepsilon, \theta}^\pm &= \{z \in \mathbb{C} \mid z = \omega + re^{\pm i\theta}, r \geq \varepsilon\}, \\ \gamma_{\varepsilon, \theta}^0 &= \{z \in \mathbb{C} \mid z = \omega + \varepsilon e^{i\tilde{\theta}}, \tilde{\theta} \in (-\theta, \theta)\}. \end{aligned}$$

Then, we define a semigroup $S(\cdot)$ of bounded linear operators in E by setting $S(0) = I$ and

$$S(t) = \frac{1}{2\pi i} \int_{\gamma_{\varepsilon, \theta}} e^{\lambda t} R(\lambda, A) d\lambda, \quad t > 0. \quad (2.2.3)$$

This integral converges absolutely and uniformly in $L(E)$, so the operators $S(t)$ are well-defined (for details, see [36, pp. 97-98]). Then the following theorem gives rise to the definition of an analytic semigroup:

Theorem 2.2.5 [26, Theorem A.9] (i) *The mapping*

$$\begin{aligned} S &: (0, \infty) \rightarrow L(E) \\ t &\mapsto S(t) \end{aligned}$$

with $S(t)$ given in (2.2.3) is analytic. Moreover, for any $x \in E$, $t > 0$ and $n \in \mathbb{N}$, $S(t)x \in D(A^n)$ and

$$S^n(t)x = A^n S(t)x.$$

(ii) *We have $S(t+s) = S(t)S(s)$ for all $s, t \geq 0$.*

(iii) $S(\cdot)x$ is continuous at 0 if and only if $x \in \overline{D(A)}$.

(iv) There exist $M, N > 0$ such that for all $t \geq 0$

$$\begin{aligned} \|S(t)\| &\leq Me^{\omega t}, \\ \|AS(t)\| &\leq e^{\omega t} \left(\frac{N}{t} + \omega M \right) \end{aligned}$$

(v) $S(\cdot)$ can be extended to an analytic $L(E)$ -valued function in $S_{0, \theta_0 - \frac{\pi}{2}}$.

The semigroup defined by (2.2.3) and having these properties is called analytic semigroup.

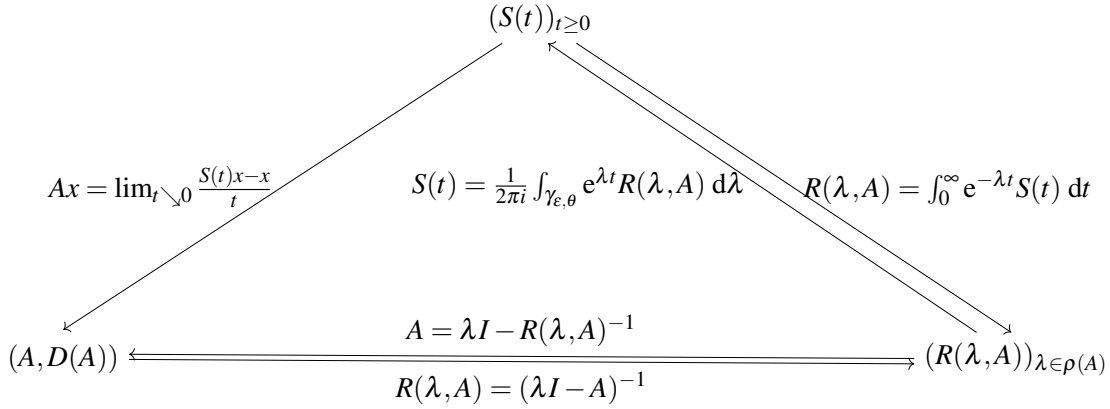


Figure 2.2.2: Diagram from [36, p. 108] showing the relations between a sectorial operator A with domain $D(A)$, its resolvent $R(\lambda, A)$ and its generated analytic semigroup $(S(t))_{t \geq 0}$.

In Section 2.2.3 and Chapter 3, we will use the Dirichlet Laplacian Δ on \mathcal{C}^2 -bounded domains. It is therefore important to mention that Δ generates a \mathcal{C}_0 -semigroup and an analytic semigroup, which we explain in the following remarks.

Remark 2.2.6 (The Dirichlet Laplacian on a \mathcal{C}^2 -bounded domain generates an analytic semigroup) *It is shown in [74, Theorem 2.5.1] that if $D \subset \mathbb{R}^d$ has a \mathcal{C}^2 boundary, then the Laplace operator on $D(\Delta) = H_0^1(D) \cap H^2(D)$ is sectorial in $L^2(D)$ and that $D(\Delta)$ is dense in $L^2(\Omega)$. Hence, the Laplace operator generates an analytic semigroup $(S(t))_{t \geq 0}$.*

Remark 2.2.7 (Dirichlet Laplacian on a \mathcal{C}^2 -bounded domain generates a \mathcal{C}_0 -semigroup) *An analytic semigroup is a \mathcal{C}_0 -semigroup if and only if the domain $D(A)$ is dense in the Banach space E (see e.g. [90]). Given that $\mathcal{C}_0^\infty(\Omega) \subset H_0^1(\Omega) \cap H^2(\Omega) \subset L^2(\Omega)$ and $\mathcal{C}_0^\infty(\Omega)$ is dense in $L^2(\Omega)$, so is $D(\Delta)$ and therefore Δ generates a \mathcal{C}_0 -semigroup in $L^2(\Omega)$ as well.*

2.2.3 Existence and uniqueness of mild solutions

In the introduction of this chapter, we noted that exponential functions can be generalized to a semigroup setting in Banach spaces to solve Banach space-valued initial value problems. We recall the initial value problem

$$\begin{cases} \frac{d}{dt}u(t) = Au(t), & t > 0, \\ u(0) = x \end{cases} \quad (2.2.4)$$

for u an E -valued function, $A \in L(E)$ and $x \in E$ the initial value. If A generates a \mathcal{C}_0 -semigroup $(S(t))_{t \geq 0}$, then $u(t) = S(t)x$ is the unique classical solution for this initial value problem [36, Proposition II.6.2]. In general, classical solutions might not exist and especially in the case where the solution is not differentiable, it is useful to introduce a different, more general concept of a solution called the mild solution.

Definition 2.2.8 [36, Definition II.6.3]. A continuous function $u : \mathbb{R}_+ \rightarrow E$ is called a mild solution of (2.2.4) if $\int_0^t u(s) \, ds \in D(A)$ for all $t \geq 0$ and

$$u(t) = A \int_0^t u(s) \, ds + x.$$

Similarly, we consider the inhomogeneous abstract initial value problem

$$\begin{cases} \frac{d}{dt}u(t) = Au(t) + F(t), & t > 0, \\ u(0) = x. \end{cases} \quad (2.2.5)$$

If a function $u : \mathbb{R}_+ \rightarrow E$ is continuously differentiable with $u(t) \in D(A)$ and satisfies (2.2.5), it is called a classical solution.

Definition 2.2.9 [36, Definition VI.7.2]. Let $(A, D(A))$ be the generator of a \mathcal{C}_0 -semigroup $(S(t))_{t \geq 0}$ on E and let $x \in E$ and $f \in L^1(\mathbb{R}_+, E)$. Then the function u defined by

$$u(t) := S(t)x + \int_0^t S(t-s)f(s) \, ds \quad (2.2.6)$$

is called the mild solution of (2.2.5).

Every classical solution is also a mild solution. We now turn our attention to stochastic evolution equations with additive noise. For a Hilbert space H , consider the nonlinear

stochastic differential equation

$$\begin{cases} dU(t) = [AU(t) + F(U(t))] dt + dW(t). \\ U(0) = u_0 \in E \end{cases} \quad (2.2.7)$$

for linear operators $A : D(A) \subset H \rightarrow H$, F an H -valued stochastic process and ξ an \mathcal{F}_0 -measurable H -valued random variable. It is assumed again that A generates a \mathcal{C}_0 -semigroup.

Definition 2.2.10 . An H -valued stochastic process $(U(t))_{t \in [0, T]}$ is called a mild solution of (2.2.7) if

$$\mathbb{P} \left(\int_0^T |U(s)| ds < \infty \right) = 1$$

and for $t \in [0, T]$,

$$U(t) = S(t)\xi + \int_0^t S(t-s)F(s) ds + \int_0^t S(t-s) dW(s). \quad (2.2.8)$$

For this equation, U , A and F are defined in a Banach space $E \subset H$ which is continuously and densely embedded in H as a Borel subset. We denote by A_E the restriction of A to E and introduce the following assumption:

Assumption 2.2.11 [26, Hypothesis 7.3] *Either*

- (i) A_E generates a \mathcal{C}_0 -semigroup $S(\cdot)$ on E or
- (ii) A generates an analytic semigroup $S_E(\cdot)$ on E .

Moreover, the stochastic convolution W_A has an E -continuous version.

Before we specify assumptions on the nonlinear function F in Assumption 2.2.13, we need to introduce subdifferentials.

Definition 2.2.12 [26, Section D.1]. Suppose that E is a Banach space and $x, y \in E$. The mapping

$$\mathbb{R} \setminus \{0\} \rightarrow \mathbb{R}, \quad h \mapsto \phi(h) = \|x + hy\|$$

is convex and nondecreasing and, for any $y \in E$, has the limits

$$D_+ \|x\| \cdot y := \lim_{h \rightarrow 0^+} \frac{\|x + hy\| - \|x\|}{h}, \quad D_- \|x\| \cdot y := \lim_{h \rightarrow 0^-} \frac{\|x + hy\| - \|x\|}{h}.$$

The subdifferential $\partial\|x\|$ of $\|x\|$ is defined as:

$$\partial\|x\| = \{x^* \in E^* \mid D_-\|x\| \cdot y \leq \langle y, x^* \rangle_{E \times E^*} \leq D_+\|x\| \cdot y \text{ for all } y \in E\}$$

where E^* is the dual of E and $\langle y, x^* \rangle_{E \times E^*} := x^*(y)$ the dual pairing. The set $\partial\|x\| \subset E^*$ is nonempty, closed and convex and can be shown to be given by

$$\partial\|x\| = \{x^* \in E^* \mid \langle x, x^* \rangle = \|x\|, \|x^*\| = 1\}.$$

Now, the assumptions for Theorem 2.2.14 on existence and uniqueness of mild solutions can be stated.

Assumption 2.2.13 [26, Hypothesis 7.4] (i) $D(F) \supset E$, $F(E) \subset E$ and the restriction $F_E := F|_E$ is locally Lipschitz continuous and bounded on bounded subsets of E .

(ii) There is an increasing function $a : \mathbb{R}_+^1 \rightarrow \mathbb{R}_+^1$ such that

$$\langle A_{E,n}x + F(x+y), x^* \rangle_{E \times E^*} \leq a(\|y\|_E)(1 + \|x\|_E) \quad (2.2.9)$$

for all $x, y \in E$, $x^* \in \partial\|x\|_E$, $n \in \mathbb{N}$, and $A_{E,n}$ denote the Yosida approximations of A_E , $\langle \cdot, \cdot \rangle_{E \times E^*}$ is the duality form on $E \times E^*$ and $\partial\|x\|_E$ is the subdifferential of the E -norm $\|\cdot\|_E$ at $x \in E$.

We now state the existence and uniqueness theorem for mild solutions.

Theorem 2.2.14 [26, Theorem 7.7] Suppose that A generates a \mathcal{C}_0 -semigroup and Assumption 2.2.13 holds.

(i) If Assumption 2.2.11 (i) holds, then equation (2.2.7) has a unique mild solution in $C([0, \infty); E)$.

(ii) If Assumption 2.2.11 (ii) holds, then equation (2.2.7) has a unique mild solution in $C([0, \infty); E) \cap L_{\text{loc}}^\infty(0, \infty; E)$.

In order to show that (2.2.7) has a solution by using Theorem 2.2.14, we follow [26, Example 7.8], and show that Assumption 2.2.11 (i) as well as Assumption 2.2.13 both hold.

We start with Assumption 2.2.11 (i): Let the operator A be the Dirichlet Laplacian on a domain $D \subset \mathbb{R}^d$, i.e. $Ay = \Delta y$ for $y \in D(A) = H^2(D) \cap H_0^1(D)$, and let $E = \mathcal{C}(\overline{D})$ equipped with the supremum norm $\|\cdot\|_E$. As we noted in Section 2.2.2, A is the infinitesimal generator of an analytic semigroup $S(\cdot)$ which is also of class \mathcal{C}_0 .

For the restriction A_E , we note that the Laplacian is understood in the weak sense, so that

$$\int_D (\Delta u(\hat{x})) \phi(\hat{x}) \, d\hat{x} = \int_D u(\hat{x}) \Delta \phi(\hat{x}) \, d\hat{x}$$

for any test function $\phi \in \mathcal{C}_0^\infty(D)$ provided that $u \in L_{\text{loc}}^1(D)$, i.e. u is integrable on any compact subset of D (see e.g. [38, §5.2]). Since $\mathcal{C}(D) \subset L_{\text{loc}}^1(D)$ (on a compact subset, any continuous function must be bounded and therefore integrable), the domain of the restricted Laplacian can be given as

$$D(A_E) = \{y \in \mathcal{C}(\overline{D}) \mid \Delta y \in \mathcal{C}(\overline{D}) \text{ and } y = 0 \text{ on } \partial D\}, \quad A_E y = \Delta y, \quad y \in D(A_E).$$

In Remark 2.2.6, we noted that the Dirichlet Laplacian generates an analytic semigroup on $L^2(D)$. It is easily checked that by restricting to $E = \mathcal{C}(D)$, it also generates an analytic semigroup on E , so Assumption 2.2.11 (i) is satisfied (after restricting to a subspace, it can easily be verified that the conditions in Assumption 2.2.3 for a sectorial operator still hold). We noted in Remark 2.2.7 that the semigroup $S(\cdot)$ is of class \mathcal{C}_0 if and only if $D(A_E)$ is dense in E . However, since $\overline{D(A_E)} \subset \{y \in \mathcal{C}(\overline{D}) \mid y = 0 \text{ on } \partial D\}$ (the limit y of any convergent sequence $(y_n)_{n \in \mathbb{N}}$ in $D(A_E)$ must satisfy $y = 0$ on ∂D since $y_n = 0$ on ∂D for all $n \in \mathbb{N}$), $D(A_E)$ cannot be dense in E and Assumption 2.2.11 (ii) is not satisfied. We therefore rely on point (i) in Theorem 2.2.14.

We now continue with Assumption 2.2.13: We first note that $\|S(t)\|_E \leq 1$ for $t \geq 0$ (see [26, inequality (A.53)]). By (2.2.2) with $M = 1$ and $\omega = 0$, it follows that $\|e^{tA_{E,n}}\|_E \leq 1$ and (see [26, Example D.8]) that $\langle A_{E,n}x, x^* \rangle_{E \times E^*} \leq 0$ for all $x \in E$, $x^* \in \partial\|x\|$. Hence, in order to prove that inequality (2.2.9) holds, it is sufficient to show

$$\langle F(x+y), x^* \rangle_{E \times E^*} \leq a(\|y\|_E)(1 + \|x\|_E), \quad \text{for all } x, y \in E, \, x^* \in \partial\|x\|. \quad (2.2.10)$$

In Proposition 2.2.18, which shows under which assumption (2.2.10) holds and equation (2.2.7) has a unique mild solution, we will make use of a representation of the dual space $(\mathcal{C}(\overline{D}))^*$ which uses Radon measures, the definition of which we recall now.

Definition 2.2.15 . Let D be a Hausdorff topological space and let \mathcal{A} be a σ -algebra on D (for example, the Borel σ -algebra). A measure μ on (D, \mathcal{A}) is called inner regular if for any open set $U \in \mathcal{A}$,

$$\mu(U) = \sup_{\substack{K \subset U \\ K \text{ compact}}} \mu(K).$$

A measure μ is called locally finite if every point in E has a neighborhood U for which $\mu(U)$ is finite. The measure μ is called Radon measure if it is inner regular and locally finite.

We also need a special case of the following theorem (see e.g. [88, Theorem 2.14]) which identifies functionals on $\mathcal{C}(D)$ with Radon measures:

Theorem 2.2.16 (Riesz-Markov-Kakutani representation theorem.) *Let D be a compact Hausdorff space and let ψ be a positive linear functional on $\mathcal{C}(D)$. Then there exists a Borel σ -algebra \mathcal{A} on E and a unique Radon measure μ with the identification*

$$\psi(x) = \int_D x(\xi) d\mu(\xi), \quad x \in \mathcal{C}(E), \xi \in D. \quad (2.2.11)$$

Conversely, every Radon measure defines a linear functional on $\mathcal{C}(E)$ via the identification (2.2.11), so the dual space $(\mathcal{C}(D))^$ can be identified with $\mathcal{M}(D)$, the space of Radon measures on D .*

For the subdifferential $\partial\|x\| \subset E^*$, we have the following, more specific characterization given in [26, p. 441]:

Remark 2.2.17 *Let $E = \overline{\mathcal{D}}$ and x be a nonzero element of E . Set*

$$M_x = \{\xi \in D \mid |x(\xi)| = \|x\|_E\}$$

and let $\mathcal{B}(\overline{D})$ be the Borel σ -algebra on D . Then, in the sense of the identification (2.2.11), $\mu \in \partial\|x\|$ if and only if

- (i) *The Radon measure μ on \overline{D} satisfies $\|\mu\| := |\mu|(\overline{D}) = 1$,*
- (ii) *The support of μ is included in M_x ,*
- (iii) *$\int_A \text{sgn}(x(\hat{x})) \mu(\hat{x}) \geq 0, \quad \forall A \in \mathcal{B}(\overline{D})$*

where sgn denotes the sign function.

The nonlinear operator F in many SPDEs in the literature can be thought of as concatenation operators with an associated real-valued function; for example, one might want the nonlinear operator F to map every function $x(\hat{x}) \in \mathcal{C}(\overline{D})$ to its squared function $(x(\hat{x}))^2$. This kind of operator is called Nemytskii operator.

Proposition 2.2.18 *Let $F = F_\phi$ be a Nemytskii operator defined as*

$$F_\phi(x)(\hat{x}) = \phi(x(\hat{x})), \quad x \in E, \hat{x} \in D,$$

with an associated function $\phi \in \mathcal{C}^1(\mathbb{R})$. If there exists an increasing function $a : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ such that

$$\phi(\xi + \eta) \operatorname{sgn}(\xi) \leq a(|\eta|)(1 + |\xi|) \quad \text{for all } \xi, \eta \in \mathbb{R}. \quad (2.2.12)$$

Then inequality (2.2.10) and therefore Assumption 2.2.13 (ii) holds.

Proof. Following Remark 2.2.17, the element $x^* \in \partial \|x\|$ can be identified with a Radon measure μ for which $\mu(\overline{D}) = 1$ holds, using the identification (2.2.11). Using this identification, we have

$$\begin{aligned} \langle F_\phi(x+y), x^* \rangle_{E \times E^*} &= x^*(\phi \circ (x+y)) \\ &= \int_D \phi(x(\hat{x}) + y(\hat{x})) \, d\mu(\hat{x}) \\ &\leq \int_D a(|y(\hat{x})|)(1 + |x(\hat{x})|) \, d\mu(\hat{x}) \\ &\leq a(\|y\|_E)(1 + \|x\|_E), \end{aligned}$$

where in the last step it was used that $\mu(\overline{D}) = 1$ and hence, with the supremum norm $\|\cdot\|_E$,

$$\int_D a(|y(\hat{x})|) \, d\mu(\hat{x}) \leq \int_D a(\|y\|) \, d\mu(\hat{x}) = a(\|y\|).$$

The other term involving x is treated analogously. This proves that inequality (2.2.10) holds. \square

2.3 Numerical solution of parabolic SPDEs

2.3.1 Different schemes in the literature

The numerical solution of SPDEs has attracted much attention, and many different methods for temporal as well as spatial discretization have been developed (for a comprehensive overview, see e.g. [86, Section 1.2]). However, due to the irregular behavior of SPDEs, strong regularity conditions have to be imposed on either the noise or the spatial domain. Some publications focused on a one-dimensional spatial domain and potentially space-time white noise [2, 3, 11, 14, 28, 47, 60, 59, 64, 98], while others have worked with trace-class or smooth noise and a potentially two-dimensional domain [19, p. 43][49, 52, 57, 58, 73]. We intend to use the latter approach by considering additive noise that has sufficiently quickly

decaying modes, but in exchange being able to consider a much larger class of general two-dimensional domains $D \subset \mathbb{R}^2$ with a \mathcal{C}^2 boundary. This broadening of the class of spatial domains has been pointed out in [86, p. 260] as a subject of future research. Also, while there has been some theoretical treatment of SPDEs on domains with a \mathcal{C}^2 boundary of dimension two or more with homogeneous Dirichlet boundary conditions [12, 31, 32], to our knowledge no solutions of SPDEs have been numerically approximated in such domains. Spectral methods have been widely used for the solution of semilinear parabolic SPDEs [11, 49, 60, 89] and are straightforward to implement in cases where the base functions are known explicitly. If this is not the case, the spectrum of the linear differential operator as well as its eigenfunctions have to be numerically approximated. This can be achieved using boundary element methods (BEM) resulting in a nonlinear eigenvalue problem. The advantage of this method is that any bounded domain with a \mathcal{C}^2 boundary can be considered, and the eigenfunction can be computed pointwise at any desired point in the interior of the domain, making it possible to store the functions on a uniform grid. Another possibility to compute the base functions are finite element methods (FEM), and for the Dirichlet Laplacian this has been done in [94]. The difference to BEM is that instead of domains with a \mathcal{C}^2 boundary, FEM can deal with polygons with a Lipschitz boundary in \mathbb{R}^2 . Since FEM requires the triangulation of the domain, if the functions are to be stored on a uniform grid, an additional error would be introduced by evaluating the functions on the triangularization at the required grid points. In this work, we focus on BEM and domains with \mathcal{C}^2 boundaries. We specify in Section 2.3.2 a precise mathematical framework, including the conditions placed on each component of the equations which can be solved. We also discuss in Section 2.3.2 the discretization in space and time of the mild solutions which are given as a Galerkin projection in space and the exponential Euler scheme in time as described by Jentzen and Kloeden [59]. The boundary integral equation method is recalled in Section 2.4, including the necessary discretization steps turning the Helmholtz equation into a nonlinear eigenvalue problem. The algorithm developed by Beyn [9], which allows for the solution of nonlinear eigenvalue problems, is also recalled in Section 2.4, also sketching the discretization steps necessary for the implementation. We will make some remarks on assembling the first elements of an orthonormal basis and it is shown how many boundary elements are needed to compute the eigenfunctions to a given accuracy. In Section 3.3, we prove a result on the strong error of the spectral Galerkin-exponential Euler scheme with eigenvalues and eigenfunctions which are approximated to a given accuracy. In Section 3.4, we demonstrate the convergence on an asymmetric shape and present a few solutions for different nonlinear functions. The MATLAB program code, including the data and programs to generate the plots shown in the figures, is available at GitHub <https://github.com/JulianSPDE/2D-SPDE>.

2.3.2 The exponential Euler scheme

The exponential Euler scheme was developed by Jentzen and Kloeden in [59]. It is a spectral scheme which allows for the solution of general H -valued SPDEs for a Hilbert space H under certain conditions. In the following, we will specify those conditions and derive the numerical scheme.

Let $D \subset \mathbb{R}^2$ be a bounded domain whose boundary is of class \mathcal{C}^2 and can be described by a parametrization γ such that $\|\gamma'(t)\|_2 > 0$, $t \in [0, 2\pi)$, $\gamma(0) = \gamma(2\pi)$ and γ is nonintersecting, i.e. $\gamma(s) \neq \gamma(t)$ for distinct $s, t \in [0, 2\pi)$. Let $T > 0$ be fixed. We consider SPDEs of the form

$$dU(t) = [AU(t) + F(U(t))]dt + dW^Q(t), \quad U(0) = u_0, \quad t \in [0, T] \quad (2.3.1)$$

where $(U(t))_{t \in [0, T]}$ is a stochastic process taking values in a Hilbert space H , $u_0 \in H$, $A : D(A) \subset H \rightarrow H$ is a linear sectorial operator, i.e. it generates an analytic semigroup $(e^{At})_{t \in [0, T]}$, and $F : H \rightarrow H$ is a nonlinear operator. The process $(W^Q(t))_{t \in [0, T]}$ is a cylindrical Q -Wiener process taking values in H , where $Q : H \rightarrow H$ is a trace-class operator. We will consider the Laplacian $A = \Delta : D(\Delta) \subset H \rightarrow H$ with homogeneous Dirichlet boundary conditions on the domain D , so that we have $H = L^2(D)$ and $D(A) = H_0^1(D) \cap H^2(D)$. It is also assumed that Δ and the covariance operator Q commute, so that they have a complete set consisting of the same orthonormal eigenfunctions e_i , $i \in \mathbb{N}$.

The SPDE (2.3.1) can be rigorously understood as an integral equation involving a stochastic dW^Q integral [26, Chapter 4]. It is interpreted in the mild sense [26, p. 161], meaning that an H -valued process $(U(t))_{t \in [0, T]}$ is a solution of (2.3.1) if

$$U(t) = e^{At}u_0 + \int_0^t e^{A(t-s)}F(U(s))ds + \int_0^t e^{A(t-s)}dW^Q(s). \quad (2.3.2)$$

We now give some details on the properties of each component of the SPDE (2.3.1).

The linear sectorial operator $A = \Delta$.

It is known that for Dirichlet, Neumann and mixed boundary conditions, the Laplacian has a complete set of orthonormal smooth eigenfunctions e_i , $i \in \mathbb{N}$. Also, the eigenvalues $-\lambda_i$ are all negative and real having exactly one accumulation point at $-\infty$ (see e.g. [8, Appendix W]; we use a notation where $\lambda_i > 0$, $i \in \mathbb{N}$). Hence, it is possible to give a spectral representation

$$\Delta f(x) = \sum_{i=1}^{\infty} -\lambda_i \langle f, e_i \rangle_H e_i(x)$$

for $f \in D(\Delta)$. We assume Dirichlet boundary conditions, so we call the λ_i Dirichlet eigenvalues and the e_i Dirichlet eigenfunctions.

The nonlinear operator F .

In our setting, the operator F on H is a Nemytskii operator, i.e. there is an associated function $u_F : \mathbb{R} \rightarrow \mathbb{R}$ such that the operator F can be defined as $[F(g)](x) = u_F(g(x))$ for $g \in H$. For F , we adopt the assumption from [59, Assumption 2.4] and require that F be two times continuously Fréchet differentiable, and that its derivatives satisfy

$$\|F'(u) - F'(v)\|_H \leq \tilde{L}\|u - v\|_H, \quad (2.3.3)$$

$$\|(-\Delta)^{-r}F'(u)(-\Delta)^r w\|_H \leq \tilde{L}\|w\|_H \quad (2.3.4)$$

for all $u, v \in H$ and $w \in D((-\Delta)^r)$ for $r \in \{0, 1/2, 1\}$ and

$$\|\Delta^{-1}F''(u)(v, w)\|_H \leq \tilde{L}\|(-\Delta)^{-1/2}v\|_H \cdot \|(-\Delta)^{-1/2}w\|_H \quad (2.3.5)$$

for all $u, v, w \in H$ and $\tilde{L} > 0$ a constant.

Furthermore, to ensure existence and uniqueness of a mild solution, we require that the associated function u_F satisfies the inequality (2.2.12) discussed in Section 2.2.3.

The Wiener process W^Q .

The process $W^Q = (W^Q(t))_{t \in [0, T]}$ is a Q -cylindrical Wiener process [26, §4.1.2][75, §3.2.35], i.e. a collection

$$\{W_u^Q(t), u \in H, t \in [0, T]\}$$

of zero-mean Gaussian random variables that satisfy

$$\mathbb{E}(W_u^Q(t)W_v^Q(s)) = \langle Qu, v \rangle_H \cdot \min(t, s)$$

for $s, t \in [0, T]$, $u, v \in H$. Then, for an orthonormal basis $\{e_n^Q\}_{n \in \mathbb{N}}$ of eigenfunctions of Q with corresponding eigenvalues $\{q_k\}_{k \in \mathbb{N}}$, the process W^Q has the series expansion in H (see Proposition 2.1.6)

$$W^Q(t) = \sum_{n=1}^{\infty} \sqrt{q_n} \beta_n(t) e_n^Q, \quad (2.3.6)$$

where $\beta_i, i \in \mathbb{N}$, are independent real-valued standard Brownian motions. We assume that Q and the operator Δ both have a joint eigenbasis consisting of the same functions. The stochastic integral with respect to the cylindrical Q -Wiener process is defined in a similar way to the classical Itô integral (for details, see [72, 2.5]).

2.3.2.1 Approximation in space.

For the spectral approximation in space, we use a projection

$$P_N : H \rightarrow H_N := \text{span}(e_1, \dots, e_N)$$

$$f = \sum_{i=1}^{\infty} \langle f, e_i \rangle_H e_i \mapsto \sum_{i=1}^N \langle f, e_i \rangle_H e_i$$

with the inner product

$$\langle f, e_i \rangle_H = \int_D f(x) \cdot e_i(x) \, dx$$

and denote $U^N := P_N U$, $F_N := P_N F$, $A_N := P_N A$, $u_0^N := P_N u_0$ and

$$W_N^Q := P_N W^Q = \sum_{i=1}^N \sqrt{q_i} e_i \beta_i(t).$$

The projected SPDE

$$dU^N(t) = (A_N U^N(t) + F_N(U^N(t)))dt + dW_N^Q(t)$$

has the mild solution satisfying for $t \in [0, T]$

$$U^N(t) = e^{A_N t} u_0^N + \int_0^t e^{A_N(t-s)} F_N(U^N(s)) \, ds + \int_0^t e^{A_N(t-s)} dW_N^Q(s). \quad (2.3.7)$$

2.3.2.2 Approximation in time.

We introduce the equidistant time steps $t_i = h \cdot i$, $i = 0, \dots, M$, with $h = T/M$, and denote $u_k^{N,M} = U^N(t_k)$. Equation (2.3.7) can be rewritten in one time step from t_k to t_{k+1} as

$$u_{k+1}^{N,M} = e^{A_N h} u_k^{N,M} + \int_{t_k}^{t_{k+1}} e^{A_N(t_{k+1}-s)} F_N(u^{N,M}(s)) \, ds + \int_{t_k}^{t_{k+1}} e^{A_N(t_{k+1}-s)} dW_N^Q(s).$$

We now make the approximation $F_N(u^{N,M}(s)) \approx F_N(u^{N,M}(t_k))$. Let \widetilde{U}^N be the process after this approximation and let $v_k^{N,M} := \widetilde{U}^N(t_k)$. We denote the inner products by

$$v_{k,j}^{N,M} := \langle v_k^{N,M}, e_j \rangle_H, \quad F_N^j(v_k^{N,M}) := \langle F_N(v_k^{N,M}), e_j \rangle_H.$$

Taking the inner product with e_j on both sides and applying the above approximation, we obtain for the approximate Fourier-Galerkin coefficients

$$\begin{aligned}
v_{k+1,j}^{N,M} &= \langle e^{A_N h} v_k^{N,M}, e_j \rangle_H + \langle \int_{t_k}^{t_{k+1}} e^{A_N(t_{k+1}-s)} F_N(v_k^{N,M}) \, ds, e_j \rangle_H \\
&\quad + \langle \int_{t_k}^{t_{k+1}} e^{A_N(t_{k+1}-s)} \, dW_N^Q(s), e_j \rangle_H \\
&= e^{-\lambda_j h} v_{k,j}^{N,M} + F_N^j(v_k^{N,M}) \cdot \int_{t_k}^{t_{k+1}} e^{-\lambda_j(t_{k+1}-s)} \, ds \\
&\quad + \int_{t_k}^{t_{k+1}} e^{-\lambda_j(t_{k+1}-s)} \sqrt{q_j} \, d\beta_j(s),
\end{aligned}$$

using linearity of the inner product and the fact that $A_N h e_i = -\lambda_i h e_i$ and hence $e^{A_N h} e_i = e^{-\lambda_i h} e_i$ for $h > 0$. The $d\beta_j(s)$ integral is normally distributed with mean zero and its variance can be computed using the Itô isometry Proposition 2.1.9 (which holds not only for elementary processes, but for general stochastically integrable processes). With this, we finally obtain the exponential Euler scheme shown in Algorithm 2.1 and first developed in [59, Section 3]. The random variables R_k^j are independently and identically distributed with a standard normal distribution.

Algorithm 2.1: Exponential Euler scheme

$$\begin{aligned}
v_{k+1,j}^{N,M} &= e^{-\lambda_j h} v_{k,j}^{N,M} + \frac{1 - e^{-\lambda_j h}}{\lambda_j} F_N^j(v_k^{N,M}) + \left(\frac{q_j}{2\lambda_j} (1 - e^{-2\lambda_j h}) \right)^{1/2} R_k^j, \\
j &= 1, \dots, N, \quad k = 0, \dots, M-1.
\end{aligned}$$

2.4 Numerically approximating Dirichlet eigenvalues and eigenfunctions: The boundary integral method and Beyn's contour integral algorithm

Let $D \subset \mathbb{R}^2$ be a bounded domain which has a \mathcal{C}^2 boundary. In this section, we explain how the eigenvalue problem for the Helmholtz equation

$$-\Delta u + \kappa^2 u = 0 \text{ in } D, \quad u = 0 \text{ on } \partial D, \quad (2.4.1)$$

with $\kappa \in \mathbb{R}_{>0}$ and homogeneous Dirichlet boundary conditions can be solved by rewriting it in terms of boundary integral layer operators, discretizing the resulting boundary integral equation and solving the resulting nonlinear eigenvalue problem with Beyn's contour integral algorithm.

Let $\Phi_\kappa(x, y)$, for $x, y \in \mathbb{R}^2$ and $x \neq y$ be the fundamental solution

$$\Phi_\kappa(x, y) = iH_0^{(1)}(\kappa\|x - y\|)/4$$

of the Helmholtz operator $\Delta + \kappa^2 I$ in two dimensions, where $H_0^{(1)}$ denotes the first-kind Hankel function of order zero (see [23, p. 66]) and let $\kappa \in \mathbb{C} \setminus \{0\}$ be fixed. For a point y on the boundary ∂D , we denote the exterior normal by $\nu(y)$ and we use the double layer potential

$$\text{DL}_\kappa[\psi](x) = \int_{\partial D} \psi(y) \cdot \partial_{\nu(y)} \Phi_\kappa(x, y) \, ds(y), \quad x \in \mathbb{R}^2 \setminus \partial D, \quad (2.4.2)$$

where $\psi \in C(\partial D)$ is the unknown density function. We also define

$$\text{D}_\kappa[\psi](x) = \int_{\partial D} \psi(y) \cdot \partial_{\nu(y)} \Phi_\kappa(x, y) \, ds(y), \quad x \in \partial D.$$

For $x \in \partial D$, it is known [23, Theorem 3.1] that the so-called jump relations

$$\begin{aligned} \lim_{h \rightarrow 0} \int_{\Gamma} \psi(y) \partial_{\nu(y)} \Phi_\kappa(x \pm h\nu(x), y) \, ds(y) = \\ \int_{\Gamma} \psi(y) \partial_{\nu(y)} \Phi_\kappa(x, y) \, ds(y) \pm \frac{1}{2} \psi(x) \end{aligned} \quad (2.4.3)$$

hold. Since the fundamental solution Φ_κ satisfies the Helmholtz equation (2.4.1), so does the double-layer potential (2.4.2). Conversely, every solution of (2.4.1) can be written as a double-layer potential [23, p. 39]. We use the double layer ansatz

$$u(x) = \text{DL}_\kappa[\psi](x), \quad x \in D \quad (2.4.4)$$

which, after letting x tend towards the boundary ∂D , with the jump relation (2.4.3) becomes

$$-\frac{1}{2} \psi(x) + \text{D}_\kappa[\psi](x) = 0, \quad x \in \partial D. \quad (2.4.5)$$

Writing $M(\kappa) := -\frac{1}{2}I + \text{D}_\kappa$, (2.4.5) becomes $M(\kappa)\psi = 0$, which is a nonlinear eigenvalue problem. First, the boundary is subdivided into n_f curved elements called $\Delta_1, \dots, \Delta_{n_f}$, and

(2.4.5) becomes

$$-\frac{1}{2}\psi(x) + \sum_{j=1}^{n_f} \int_{\Delta_j} \psi(y) \partial_{\mathbf{v}(y)} \Phi_{\kappa}(x, y) \, \mathrm{d}s(y) = 0, \quad x \in \partial D. \quad (2.4.6)$$

Let $m_j : [0, 1] \rightarrow \Delta_j$, $j = 1, \dots, n_f$, be bijective, continuous maps. With a change of variables, (2.4.6) becomes

$$-\frac{1}{2}\psi(x) + \sum_{j=1}^{n_f} \int_0^1 \psi(m_j(s)) \partial_{\mathbf{v}(m_j(s))} \Phi_{\kappa}(x, m_j(s)) \|\partial_s m_j(s)\| \, \mathrm{d}s(s) = 0, \quad x \in \partial D. \quad (2.4.7)$$

We now denote by $L_i(s)$ the quadratic Lagrange polynomials and by $\widehat{L}_i(s)$ the generalized Lagrange basis functions, $i = 1, 2, 3$:

$$\begin{aligned} L_1(s) &= (1-s)(1-2s), & L_2(s) &= 4s(1-s), & L_3(s) &= s(2s-1), \\ \widehat{L}_1(s) &= \frac{1-s-\alpha}{1-2\alpha} \frac{1-2s}{1-2\alpha}, & \widehat{L}_2(s) &= 4 \frac{s-\alpha}{1-2\alpha} \frac{1-s-\alpha}{1-2\alpha}, & \widehat{L}_3(s) &= \frac{s\alpha}{1-2\alpha} \frac{2s-1}{1-2\alpha} \end{aligned}$$

and we fix a parameter

$$\alpha \in (0, 1/2). \quad (2.4.8)$$

Each $m_j(s)$ is approximated by a quadratic interpolation polynomial

$$m_j(s) \approx \tilde{m}_j(s) = \sum_{i=1}^3 v_i^{(j)} L_i(s),$$

where $v_i^{(j)}$, $i = 1, 3$ are the end points of the j -th boundary element and $v_2^{(j)}$ is its midpoint. We define the collocation nodes $\tilde{v}_{j,k} := \tilde{m}_j(q_k)$ for $j = 1, \dots, n_f$, $k = 1, 2, 3$ and $q_1 = \alpha$, $q_2 = 1/2$, $q_3 = 1 - \alpha$. The unknown function $\psi(\tilde{m}_j(s))$ is approximated on each piece Δ_j by a quadratic interpolation polynomial

$$\sum_{k=1}^3 \psi(\tilde{m}_j(q_k)) \widehat{L}_k(s) = \sum_{k=1}^3 \psi(\tilde{v}_{j,k}) \widehat{L}_k(s).$$

Equation (2.4.7) then becomes

$$-\frac{1}{2}\psi(x) + \sum_{j=1}^{n_f} \sum_{k=1}^3 \int_0^1 \partial_{\mathbf{v}(\tilde{m}_j(s))} \Phi_{\kappa}(x, \tilde{m}_j(s)) \|\partial_s \tilde{m}_j(s)\| \widehat{L}_k(s) \, \mathrm{d}s(s) \psi(\tilde{v}_{j,k}) = r(x),$$

with a residue $r(x)$ which results from the various approximations. For the boundary element collocation method, we require $r(\tilde{v}_{i,\ell}) = 0$ [7, p. 50] and obtain the linear system of size $3n_f \times 3n_f$

$$-\frac{1}{2}\psi(\tilde{v}_{i,\ell}) + \sum_{j=1}^{n_f} \sum_{k=1}^3 a_{i,j,k,\ell} \psi(\tilde{v}_{j,k}) = 0 \quad (2.4.9)$$

with $i = 1, \dots, n_f$, $\ell = 1, 2, 3$ and with the integrals

$$a_{i,j,k,\ell} = \int_0^1 \partial_{v(\tilde{m}_j(s))} \Phi_{\kappa}(\tilde{v}_{i,\ell}, \tilde{m}_j(s)) \|\partial_s \tilde{m}_j(s)\| \hat{L}_k(s) \, ds$$

which are approximated using a Gauß-Kronrod quadrature. We write (2.4.9) abstractly as the nonlinear eigenvalue problem $M(\kappa)\phi = 0$, where

$$\phi = (\psi(v_{1,1}), \psi(v_{1,2}), \psi(v_{1,3}), \psi(v_{2,1}), \dots, \psi(v_{n_f,3}))^\top. \quad (2.4.10)$$

This leaves us with a nonlinear eigenvalue problem that needs to be solved, which is done using Beyn's contour integral algorithm [9, p. 3849], which makes use of Keldysh's theorem and is based on complex contour integrals of the resolvent $M^{-1}(\kappa)$, whose poles are the Dirichlet eigenvalues.

Theorem 2.4.1 (Keldysh's Theorem. [66]) *Let $D \subset \mathbb{C}$ be a domain, and let $v^* = \overline{v}^\top$ denote the conjugate transpose of a vector or matrix. Suppose that $C \subset D$ is a compact subset containing only simple eigenvalues λ_n , $n = 1, \dots, n(\gamma)$ with eigenvectors v_n , w_n satisfying*

$$M(\lambda_n)v_n = 0, \quad w_n^* M(\lambda_n) = 0, \quad w_n^* M'(\lambda_n)v_n = 1.$$

Then there exists some neighborhood U of C in D and a holomorphic function $R : U \rightarrow \mathbb{C}^{3n_f \times 3n_f}$ such that

$$M(z)^{-1} = \sum_{n=1}^{n(\gamma)} \frac{1}{z - \lambda_n} v_n w_n^* + R(z), \quad z \in U \setminus \{\lambda_1, \dots, \lambda_{n(\gamma)}\}. \quad (2.4.11)$$

Let $\sigma(M)$ denote the spectrum of M and let $\gamma \subset D$ be a contour which satisfies $\gamma \cap \sigma(M) = \emptyset$ and containing $n(\gamma)$ eigenvalues $\lambda_1, \dots, \lambda_{n(\gamma)}$. Let $f : D \rightarrow \mathbb{C}$ be holomorphic. Then, applying the residue theorem to (2.4.11) we have

$$\frac{1}{2\pi i} \int_{\gamma} f(z) M(z)^{-1} \, dz = \sum_{n=1}^{n(\gamma)} f(\lambda_n) v_n w_n^*.$$

Algorithm 2.2: Beyn contour integral algorithm

For a specified contour $\gamma(t) = \mu + R\cos(t) + iR\sin(t)$, $\mu \in \mathbb{R}$, $t \in [0, 2\pi)$, $R \in \mathbb{R}_{>0}$ and the number $n(\gamma)$ of eigenvalues enclosed in γ (with multiplicity), we pick a random matrix $\hat{V} \in \mathbb{C}^{3n_f \times \ell}$ with $3n_f \gg \ell \geq n(\gamma)$. We compute the contour integrals

$$\begin{aligned} A_0 &= \frac{1}{2\pi i} \int_{\gamma} M^{-1}(\kappa) \hat{V} \, ds(\kappa) = \frac{1}{2\pi i} \int_0^{2\pi} M^{-1}(\gamma(t)) \hat{V} \gamma'(t) \, ds(t) \\ &= \sum_{n=1}^{n(\gamma)} v_n w_n^* \hat{V} = V W^* \hat{V}, \\ A_1 &= \frac{1}{2\pi i} \int_{\gamma} \kappa M^{-1}(\kappa) \hat{V} \, ds(\kappa) = \frac{1}{2\pi i} \int_0^{2\pi} \gamma(t) M^{-1}(\gamma(t)) \hat{V} \gamma'(t) \, ds(t) \\ &= \sum_{n=1}^{n(\gamma)} \lambda_n v_n w_n^* \hat{V} = V \Lambda W^* \hat{V}, \end{aligned}$$

Algorithm 2.3: Beyn contour integral algorithm (continued)

with $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{n(\gamma)})$. These integrals are approximated using the trapezoidal rule:

$$A_{0,K} = \frac{1}{iK} \sum_{j=0}^{K-1} M^{-1}(\gamma(t_j)) \hat{V} \gamma'(t_j), \quad A_{1,K} = \frac{1}{iK} \sum_{j=0}^{K-1} \gamma(t_j) M^{-1}(\gamma(t_j)) \hat{V} \gamma'(t_j), \quad (2.4.12)$$

with $t_j = \frac{2\pi j}{K}$, $j = 0, \dots, K$. Next, we compute a singular value decomposition $A_{0,K} = V \Sigma W^*$ with $V \in \mathbb{C}^{3n_f \times \ell}$, $\Sigma \in \mathbb{C}^{\ell \times \ell}$ and $W \in \mathbb{C}^{\ell \times \ell}$. For $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_\ell)$ and a given tolerance $\varepsilon_{\text{sing}}$, we find $n(\gamma)$ such that

$$\sigma_1 \geq \dots \geq \sigma_{n(\gamma)} > \varepsilon_{\text{sing}} > \sigma_{n(\gamma)+1} \geq \dots \geq \sigma_\ell. \quad (2.4.13)$$

The singular values $\sigma_{n(\gamma)+1} \geq \dots \geq \sigma_\ell$ which are below the tolerance are likely to be different from zero only due to rounding errors and are discarded. With the matrices $V_0 := (V_{ij})_{1 \leq i \leq m, 1 \leq j \leq n(\gamma)}$, $\Sigma_0 := (\Sigma_{ij})_{1 \leq i \leq n(\gamma), 1 \leq j \leq n(\gamma)}$ as well as $W_0 := (W_{ij})_{1 \leq i \leq \ell, 1 \leq j \leq n(\gamma)}$ there is a regular matrix $S \in \mathbb{C}^{n(\gamma) \times n(\gamma)}$ such that $V = V_0 S$, and after rewriting some matrices, we have that

$$S \Lambda S^{-1} = V_0^* A_1 W_0 \Sigma_0^{-1}.$$

The matrix on the right-hand side therefore has the same eigenvalues and eigenvectors as Λ . We compute the $n(\gamma)$ eigenvalues and eigenvectors s_i of the matrix $V_0^* A_1 W_0 \Sigma_0^{-1} \in \mathbb{C}^{n(\gamma) \times n(\gamma)}$. The i -th eigenfunction u_i is then approximated by inserting $\phi = V_0 s_i$ (see (2.4.10)) into

$$u_i(x) = \text{DL}_\kappa[\psi](x) \approx \sum_{j=1}^{n_f} \sum_{k=1}^3 \hat{a}_{j,k}(x) \cdot \psi(\tilde{v}_{j,k}),$$

where

$$\hat{a}_{j,k}(x) = \int_0^1 \partial_{v(\tilde{m}_j(s))} \Phi_\kappa(x, \tilde{m}_j(s)) \|\partial_s \tilde{m}_j(s)\| \hat{L}_k(s) \, ds(s),$$

which again are approximated using a Gauß-Kronrod quadrature.

Chapter 3

Solving semilinear parabolic SPDEs on two-dimensional domains with a \mathcal{C}^2 boundary

This chapter deals with solving SPDEs of the type shown in (2.2.7) on general bounded two-dimensional domains with a \mathcal{C}^2 boundary and with Dirichlet boundary conditions. For this, the exponential Euler scheme recalled in Section 2.3.2 as well as Beyn's algorithm to numerically approximate Dirichlet eigenvalues and eigenfunctions, recalled in Section 2.4, are used. Most of this chapter can also be found in the paper [21].

3.1 The exponential Euler scheme for two-dimensional domains

In the literature on SPDEs, at least one of the two constraints have to be made: Either a one-dimensional spatial domain is considered, where white noise as well as trace-class noise can be considered, or the covariance operator Q cannot be allowed to be $Q = I$, i.e. space-time white noise cannot be considered. This is due to the fact that a solution of an SPDE on a multidimensional domain (i.e. $D \subset \mathbb{R}^n$, $n \geq 2$) with space-time white noise (i.e. W^Q is a cylindrical Wiener process with $Q = I$) does not exist even for $F = 0$, since vital regularity properties of the stochastic convolution

$$W_A^Q(t) = \int_0^t e^{A(t-s)} dW^Q(s), \quad t \in [0, T],$$

in (2.3.2) are violated. See [26, Example 5.7] [97, p. 329] for discussions of these failures in spatial dimensions of two and higher. The deciding factor is the asymptotic behavior of Dirichlet eigenvalues, which has been described by Weyl's law [48, 4.1]: If $0 < \lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots$ are the Dirichlet eigenvalues of the negative Laplacian for an arbitrary domain $D \subset \mathbb{R}^d$, then

$$\lambda_n \propto \frac{4\pi^2}{(\omega_d \mu_d(D))^{2/d}} n^{2/d}, \quad n \rightarrow \infty, \quad (3.1.1)$$

where $\mu_d(D)$ is the d -dimensional Lebesgue measure of D and ω_d is the volume of the d -dimensional unit ball. The regularity condition [59, Assumption 2.2] for the exponential Euler scheme which we will use requires

$$\sum_{n=1}^{\infty} (\lambda_n)^{2\gamma-1} q_n < \infty \quad (3.1.2)$$

for some $\gamma \in (0, 1)$, which for space-time white noise, i.e. $q_n = 1$ for all $n \in \mathbb{N}$, is not fulfilled in two dimensions. Instead, we assume that $q_n = \mathcal{O}(n^{-\alpha})$ for some $\alpha > 0$, in which case (3.1.2) holds if $\gamma < \alpha/2$. In our numerical experiments, we will use $q_n = n^{-2}$ so that γ can be arbitrarily close to 1.

3.2 Preprocessing: Computation of the orthonormal basis

The analytical expressions of the Dirichlet eigenfunctions are known for some shapes. In the following, we will consider a shape which we call the Peanut shape (see Figure 3.2.2), whose Dirichlet eigenvalues and eigenfunctions are not known analytically and whose parametrization for $t \in [0, 2\pi)$ is given by

$$\begin{pmatrix} 0.06 \cdot ((\cos(t) + 2) \cdot (\cos(t + 0.6) + 2) \cdot (0.1 \cdot \cos(3t) + 2)) - 0.1 \\ 0.06 \cdot (\sin(t) + 2) \cdot (\sin(t - 0.5) + 2) \cdot (0.4 \cdot \cos(2t) + 2) \cdot (0.1 \cdot \sin(4t) + 1)) - 0.06 \end{pmatrix}.$$

For the boundary element collocation method as described in Section 2.4, we choose $\alpha = (1 - \sqrt{3/5})/2$ (see (2.4.8)). Note that while any $\alpha \in (0, \frac{1}{2})$ ensures at least cubic convergence, empirically it was seen [67, Section 4.6] that the choice $\alpha = (1 - \sqrt{3/5})/2$ yielded superconvergence for domains with a \mathcal{C}^2 boundary. Whenever the Beyn integral algorithm is used to compute its Dirichlet eigenvalues and eigenfunctions, we use $\varepsilon_{\text{sing}} = 10^{-4}$ as singular value tolerance (see (2.4.13)) and $K = 24$ for the periodic trapezoidal rule (see (2.4.12)).

3.2.1 Convergence and accuracy of base functions.

In previous work, the Beyn integral algorithm was used to compute Neumann eigenvalues, and a convergence with rates up to three for a fixed wavenumber could be observed [67]. However, the eigenvalues and eigenfunctions become less accurate for higher wavenumbers, so that more boundary elements are needed for higher wavenumbers. Previous publications from high-frequency scattering theory using boundary element collocation methods to solve the Helmholtz equation have pointed out that this effort rises linearly with the wavenumber [16, 18], which we could confirm for the case of the Dirichlet eigenfunctions and eigenvalues (see Figure 3.2.5). For the Peanut shape, we have conducted a convergence analysis for the first 200 eigenfunctions. For each wavenumber, we computed a reference function for 600 boundary elements which we compare to functions with lower accuracy (computed with 500, 400, 300, 200, 150, 100, 75 and 50 boundary elements, respectively). For speeding up the computations, these functions were stored only on an equidistant 81×81 grid inside $[0, 1]^2$, while the final functions making up the ONB are stored on a 301×301 grid to obtain results with better resolution (for the error analysis in Section 3.3, we will denote the grid resolution as R and will use $R = 301$). The pointwise storage on the equidistant grid means that the inner products, i.e. the integrals in the scheme (2.1) can be approximated numerically. As it is easy to implement and has a convergence order of four [37, p. 21 f.] (and hence, a convergence order of two for two-dimensional functions), we use the composite two-dimensional Simpson rule and set the numerically integrated function to be zero outside the boundary, so that the Simpson rule can be applied on the unit square. The application of the Simpson rule is possible since eigenfunctions of the Laplacian are always smooth [48, §2.(i)]. Suppose that we use $n_f^{(i)}$ boundary elements for eigenfunction number i , $i = 1, \dots, N$. We fix error tolerances

$$\varepsilon^{(\lambda)} := \max_{i=1, \dots, N} |\tilde{\lambda}_i^{(n_f^{(i)})} - \tilde{\lambda}_i^{(600)}|, \quad \varepsilon^{(\eta)} := \max_{i=1, \dots, N} \|\tilde{e}_i^{(n_f^{(i)})} - \tilde{e}_i^{(600)}\|_H,$$

where the tilde denotes approximation and the superscript denotes that the approximation is done with $n_f^{(i)}$ boundary elements. Depending on which $\varepsilon^{(\lambda)}$ and $\varepsilon^{(\eta)}$ we choose, different wavenumber dependent amounts of boundary elements are needed (see Figure 3.2.5). We will later observe (see Example 1) that for the error bound derived in Theorem 3.3.4 it is crucial to reduce $\varepsilon^{(\lambda)}$ and $\varepsilon^{(\eta)}$ as much as possible.

3.2.2 Computation of the orthonormal basis.

After precomputing the necessary numbers of boundary elements for each function (see Section 3.2.1) and the spectrum of $-\Delta$ on D is known up to a few digits, the eigenvalues and eigenfunctions can be computed by applying the Beyn algorithm to each eigenvalue by enclosing it by a contour which does not enclose any other eigenvalue (see Figure 3.2.1). The Peanut shape analyzed here is asymmetric and does not appear to have any multiple Dirichlet eigenvalues. If multiple eigenvalues occur, it can happen that the different eigenfunctions belonging to one eigenvalue are not returned by the algorithm as orthogonal functions, but as linear combinations of them. This would necessitate an additional orthogonalization routine applied to those eigenfunctions, which would pose another error source.

Based on the convergence results for the Peanut shape, our base functions for the Peanut have been computed such that the error to a reference function with $n_f^{\text{ref}} = 600$ boundary elements stays below the tolerances $\varepsilon^{(\lambda)} = 2 \cdot 10^{-4}$ and $\varepsilon^{(\eta)} = 5 \cdot 10^{-6}$. In addition, we have

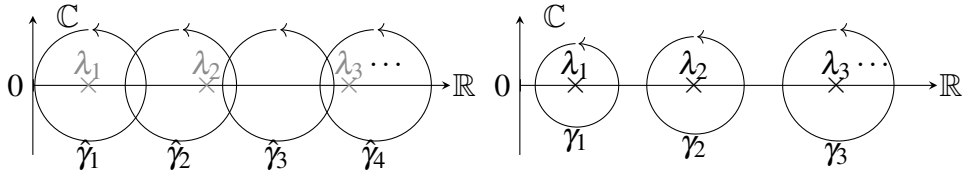


Figure 3.2.1: In the first step, the eigenvalues are still unknown, so the real axis up to a certain point is covered with contours $\hat{\gamma}_i$ which cover the real axis up to the point where the desired number N of eigenvalues has been found. For the subsequent, more precise computation, contours γ_i , $i = 1, \dots, N$ around the found eigenvalues are chosen.

computed some errors to even more accurate wavenumbers and eigenfunctions computed with $n_f^{\text{ref}} = 1,200$ boundary elements. A comparison of the reference errors is shown in Figure 3.2.2 for two wavenumbers. It can be seen that there is a slight discrepancy for the wavenumber error at $n_f = 500$, but the reference error for the eigenfunctions is almost the same for both reference functions, so it can be assumed that the reference functions are sufficiently accurate.

On the computational cost, we note that for an orthonormal basis of a certain number N of eigenvalues and eigenfunctions the computational time with our implementation behaved asymptotically as $\mathcal{O}(N^2)$. In more detail, it is shown in Figure 3.2.3 that, with growing number n_f of used boundary elements, the computation times for an eigenvalue rises as $\mathcal{O}(n_f^2)$, and for an eigenfunction as $\mathcal{O}(n_f)$ (although it is more costly for low n_f). The computation time is very similar for different eigenvalues and the corresponding eigenfunctions, and neither increases nor decreases as the eigenvalue increases. Figure 3.2.4 shows an estimate

of the total computational cost to compute an orthonormal basis of N eigenvalues and eigenfunctions, i.e. the sum

$$\sum_{i=1}^N \text{ev}(n_f(\lambda_i)) + \text{ef}(n_f(\lambda_i)), \quad (3.2.1)$$

where $n_f(\lambda)$ is the needed number of boundary elements for a given set of error tolerances, and $\text{ev}(n_f)$ and $\text{ef}(n_f)$ are the computation times for eigenvalue and eigenfunction using n_f boundary elements. Based on empirical tests such as shown in Figure 3.2.3, we assumed that $\text{ev}(n_f) = 0.15 \cdot n_f^2$ and $\text{ef}(n_f) = 50 \cdot n_f$. Since $\lambda_i \sim i$, $n_f(i) \sim \sqrt{i}$ (see Figure 3.2.5), $\text{ev}(i) \sim i^2$ and $\text{ef}(i) \sim i$, the sum (3.2.1) contains a term behaving as i and one behaving as \sqrt{i} , so that the whole sum is made up of two terms (one each for eigenvalues and eigenfunctions) behaving as $\mathcal{O}(N^2) + \mathcal{O}(N^{3/2})$.

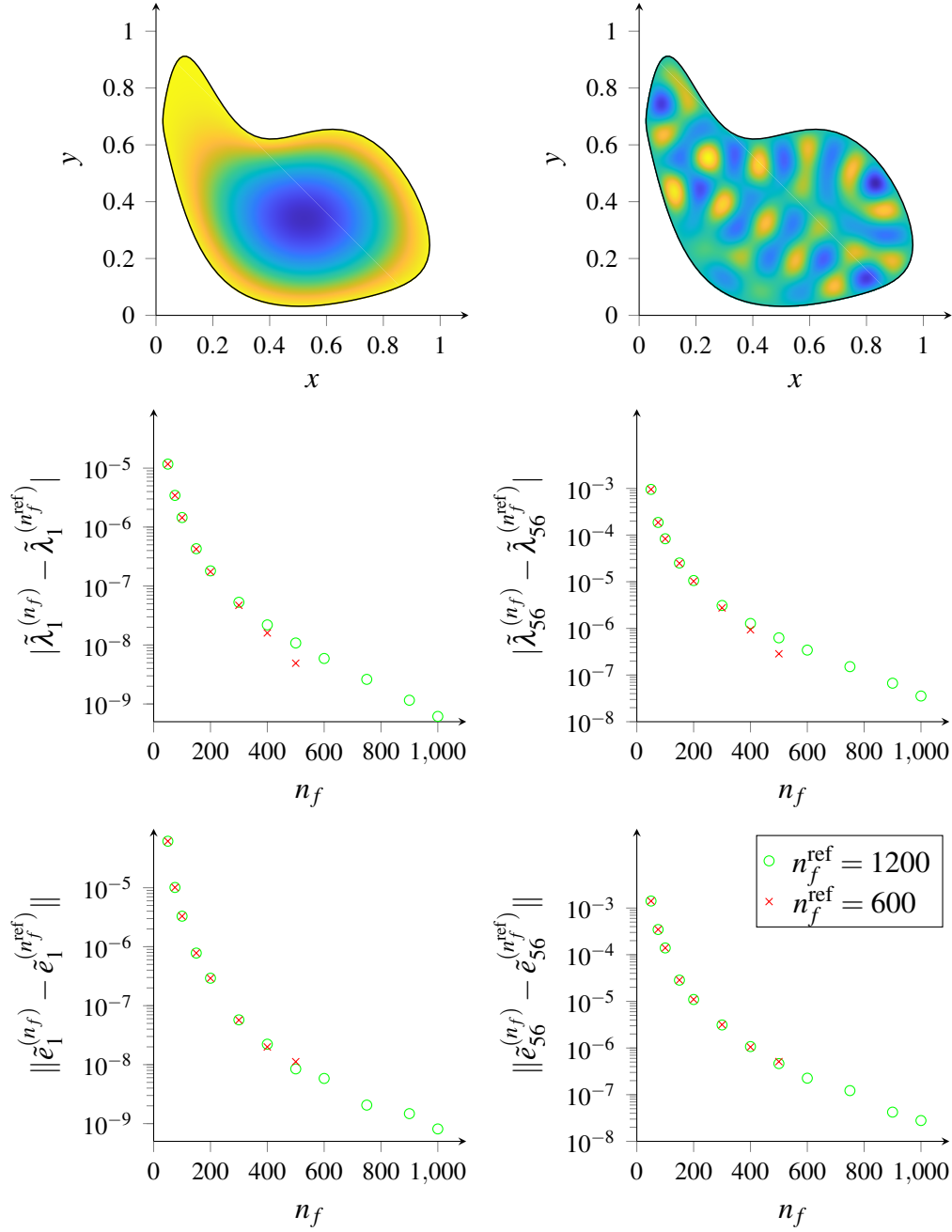


Figure 3.2.2: Convergence results for two eigenfunctions of the Peanut shape (on the left, the first one, on the right, the 56th one), in red, the error is measured with respect to reference eigenvalues and eigenfunctions computed for $n_f^{\text{ref}} = 600$ boundary elements, in green, the reference data is computed for $n_f^{\text{ref}} = 1200$ boundary elements.

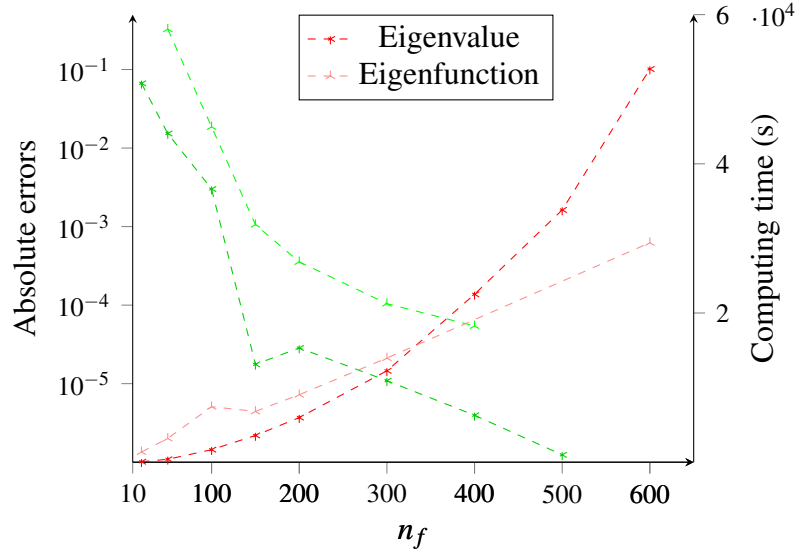


Figure 3.2.3: This plot shows, for different numbers of boundary elements n_f , absolute errors for the 300th eigenvalue and its corresponding eigenfunction in green (left axis) as well as CPU time for their computation in seconds (right axis). It can be seen that the CPU time for the eigenvalues behaves like $\mathcal{O}(n_f^2)$, while the time for the eigenfunctions is $\mathcal{O}(n_f)$. The computing times are very similar for other eigenvalues and eigenfunctions.

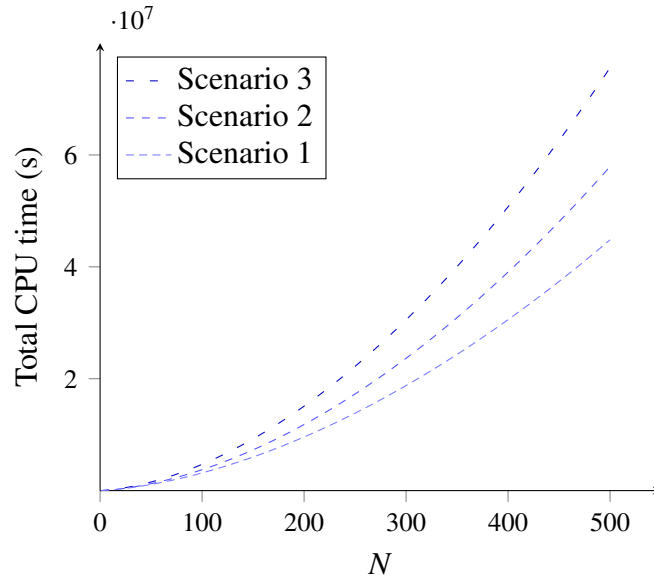


Figure 3.2.4: The total computation time on our machine for an orthonormal basis consisting of N eigenvalues and eigenfunctions. Scenarios 1 to 3 correspond, from top to bottom, to the three pairs of error tolerances shown in Figure 3.2.5.

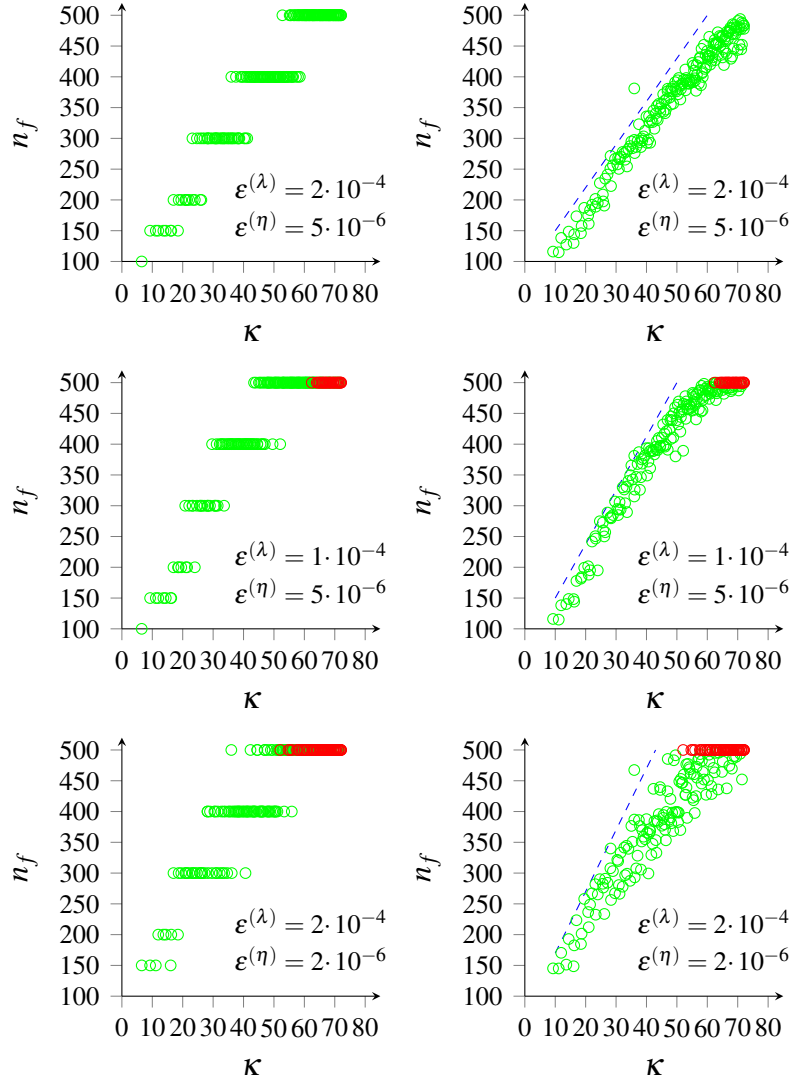


Figure 3.2.5: The number of boundary elements to approximate wavenumbers and their respective eigenfunctions to stay below certain error tolerances $\varepsilon^{(\lambda)}$ and $\varepsilon^{(\eta)}$. For each base function, we computed a reference solution with $n_f^{\text{ref}} = 600$ boundary elements and compared the same function with less boundary elements to it, with $n_f \in \{50, 75, 100, 150, 200, 300, 400, 500\}$ boundary elements used. The plots on the left show raw data, with the least number of required boundary elements. The plots on the right show linearly interpolated numbers of needed boundary elements based off the data from the plots on the left. We can observe the linear relationship between wavenumber and needed boundary elements n_f , and that the needed n_f increases as the error tolerances decrease (in red are the values which exceed the error tolerance for every tested n_f). The blue reference slopes are functions $f(x) = mx + b$, for $m \in \{7, 8.5, 10\}$, $b \in \{80, 65, 70\}$, from top to bottom.

3.3 Error analysis

We present an error analysis for the strong error in the Galerkin exponential Euler scheme from [59] with approximated eigenfunctions and eigenvalues. We denote $\langle \cdot, \cdot \rangle := \langle \cdot, \cdot \rangle_H$ and $\| \cdot \| := \| \cdot \|_H$, and the exact solution at time t by $U(t)$.

Its approximation by the spectral Galerkin exponential Euler scheme is denoted by $V_k^{(1)}$. The approximation of $V_k^{(1)}$ by computing the inner products as numerical integrals over D is denoted by $V_k^{(2)}$. The approximation of $V_k^{(2)}$ using approximate eigenvalues is denoted by $V_k^{(3)}$. Finally, the approximation of $V_k^{(3)}$ using approximate eigenfunctions is denoted by $V_k^{(4)}$.

Throughout the section, let $L = \|f'\|_\infty$ be the optimal Lipschitz constant of the Nemytskii function f of the nonlinearity in the SPDE (i.e. the infimum of all Lipschitz constants, yielding the best estimate). Also, we denote with R the resolution of the spatial grid along each spatial dimension, used for computing the numerical integrals whose error is dealt with in Lemma 3.3.1.

We denote

$$V_{k,j}^{(i)} := \langle V_k^{(i)}, e_j \rangle, \quad i = 1, 2, 3, 4, \quad \varepsilon_{k,j}^{(i)} := \mathbb{E} |V_{k,j}^{(i-1)} - V_{k,j}^{(i)}|, \quad i = 2, 3, 4,$$

with $k = 0, \dots, M$ (time step) and $j = 1, \dots, N$ (eigenfunction number). First, error estimates for $\|V_k^{(i)} - V_k^{(i+1)}\|$ are given for $i = 1, 2, 3$ in Lemmas 3.3.1, 3.3.2 and 3.3.3 which are then combined to give a general result on the error $\|U(t_k) - V_k^{(4)}\|$ in Theorem 3.3.4. Note that $V_k^{(4)}$ is the output of the computer programs, so the error estimate in Theorem 3.3.4 contains Fourier-Galerkin coefficients of $V_k^{(4)}$ which can then either be inserted for a run-specific bound, or be bounded by an empirically found constant which is generally found to be an upper bound (which we have done in Example 1).

Lemma 3.3.1 *We have*

$$\mathbb{E} \|V_k^{(1)} - V_k^{(2)}\| \leq \sum_{j=1}^N \varepsilon_{k,j}^{(2)} \quad (3.3.1)$$

and, for a constant $C > 0$,

$$\varepsilon_{k+1,j}^{(2)} \leq e^{-\lambda_j h} \varepsilon_{k,j}^{(2)} + \frac{1 - e^{-\lambda_j h}}{\lambda_j} \cdot \left[L \cdot \sum_{i=1}^N \varepsilon_{k,i}^{(2)} + C \cdot R^{-2} \right]. \quad (3.3.2)$$

Proof. We compute, distinguishing $f_N^j(V_k^{(2)})$ computed as an integral and $\widetilde{f_N^j(V_k^{(2)})}$ computed as a numerical integral,

$$\begin{aligned}\mathbb{E}\|V_k^{(1)} - V_k^{(2)}\| &= \mathbb{E}\left\|\sum_{j=1}^N V_{k,j}^{(1)} \cdot e_j - \sum_{j=1}^N V_{k,j}^{(2)} \cdot e_j\right\| = \mathbb{E}\left\|\sum_{j=1}^N (V_{k,j}^{(1)} - V_{k,j}^{(2)}) \cdot e_j\right\| \\ &\leq \sum_{j=1}^N \mathbb{E}|V_{k,j}^{(1)} - V_{k,j}^{(2)}| = \sum_{j=1}^N \varepsilon_{k,j}^{(2)}\end{aligned}\tag{3.3.3}$$

and

$$\begin{aligned}\varepsilon_{k+1,j}^{(2)} &= \mathbb{E}\left|e^{-\lambda_j h} V_{k,j}^{(1)} + \frac{1 - e^{-\lambda_j h}}{\lambda_j} f_N^j(V_k^{(1)}) + \left(\frac{q_j}{2\lambda_j}(1 - e^{-2\lambda_j h})\right)^{1/2} R_k^j\right. \\ &\quad \left.- \left[e^{-\lambda_j h} V_{k,j}^{(2)} + \frac{1 - e^{-\lambda_j h}}{\lambda_j} \widetilde{f_N^j(V_k^{(2)})} + \left(\frac{q_j}{2\lambda_j}(1 - e^{-2\lambda_j h})\right)^{1/2} R_k^j\right]\right| \\ &\leq e^{-\lambda_j h} \varepsilon_{k,j}^{(2)} + \frac{1 - e^{-\lambda_j h}}{\lambda_j} \mathbb{E}|f_N^j(V_k^{(1)}) - \widetilde{f_N^j(V_k^{(2)})}|. \\ \mathbb{E}|f_N^j(V_k^{(1)}) - \widetilde{f_N^j(V_k^{(2)})}| &\leq \mathbb{E}|f_N^j(V_k^{(1)}) - f_N^j(V_k^{(2)})| + \mathbb{E}|f_N^j(V_k^{(2)}) - \widetilde{f_N^j(V_k^{(2)})}| \\ &\leq \mathbb{E} \int_D \left[f_N(V_k^{(1)}(x)) - f_N(V_k^{(1)}(x) + \eta_k(x))\right] e_j(x) dx + C \cdot R^{-2} \\ &\leq \mathbb{E} \left[\int_D C[L \cdot \eta_k(x)]^2 dx\right]^{1/2} + C \cdot R^{-2} = L \cdot \|\eta_k\| + C \cdot R^{-2} \\ &\leq L \cdot \sum_{i=1}^N \varepsilon_{k,i}^{(2)} + C \cdot R^{-2},\end{aligned}$$

where the Hölder inequality and the Lipschitz continuity of f_N were used. In total, we have

$$\varepsilon_{k+1,j}^{(2)} \leq e^{-\lambda_j h} \varepsilon_{k,j}^{(2)} + \frac{1 - e^{-\lambda_j h}}{\lambda_j} \cdot \left[L \cdot \sum_{i=1}^N \varepsilon_{k,i}^{(2)} + C \cdot R^{-2}\right]. \quad \square$$

Lemma 3.3.2 *We have*

$$\mathbb{E}\|V_k^{(2)} - V_k^{(3)}\| \leq \sum_{j=1}^N \varepsilon_{k,j}^{(3)}\tag{3.3.4}$$

and

$$\begin{aligned}
\epsilon_{k+1,j}^{(3)} &\leq e^{-\lambda_j h} \epsilon_{k,j}^{(3)} + e^{-\lambda_j h} |1 - e^{-\epsilon_j^{(\lambda)} h}| \cdot (\epsilon_{k,j}^{(4)} + \mathbb{E}|V_{k,j}^{(4)}|) \\
&\quad + L \cdot \frac{1 - e^{-\lambda_j h}}{\lambda_j} \cdot \sum_{j=1}^N \epsilon_{k,j}^{(3)} \\
&\quad + \frac{1}{\lambda_j \tilde{\lambda}_j} \left[\epsilon^{(\lambda)} + |e^{-\tilde{\lambda}_j h} \lambda_j - e^{-\lambda_j h} \tilde{\lambda}_j| \right] \cdot \left[L \cdot \sum_{j=1}^N \epsilon_{k,j}^{(4)} + \mathbb{E}|f_N^j(V_k^{(4)})| \right] \\
&\quad + \left| \sqrt{\frac{q_j}{2\lambda_j}} (1 - e^{-2\lambda_j h}) - \sqrt{\frac{q_j}{2(\lambda_j + \epsilon_j^{(\lambda)})}} (1 - e^{-2(\lambda_j + \epsilon_j^{(\lambda)})h}) \right|.
\end{aligned} \tag{3.3.5}$$

Proof. Suppose that we have approximate eigenvalues $\tilde{\lambda}_j = \lambda_j + \epsilon_j^{(\lambda)}$, $j = 1, \dots, N$. Then by the same steps as in (3.3.3), the resulting error of the whole function is

$$\mathbb{E}\|V_k^{(2)} - V_k^{(3)}\| \leq \sum_{j=1}^N \epsilon_{k,j}^{(3)}.$$

The coefficient errors after the next time step are given by

$$\begin{aligned}
\epsilon_{k+1,j}^{(3)} &= \mathbb{E} \left| e^{-\lambda_j h} V_{k,j}^{(2)} + \frac{1 - e^{-\lambda_j h}}{\lambda_j} f_N^j(V_k^{(2)}) + \left(\frac{q_j}{2\lambda_j} (1 - e^{-\lambda_j h}) \right)^{1/2} R_k^j \right. \\
&\quad \left. - \left[e^{-\tilde{\lambda}_j h} V_{k,j}^{(2)} + \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} f_N^j(V_k^{(3)}) + \left(\frac{q_j}{2\tilde{\lambda}_j} (1 - e^{-2\tilde{\lambda}_j h}) \right)^{1/2} R_k^j \right] \right| \\
&\leq \mathbb{E} |e^{-\lambda_j h} V_{k,j}^{(2)} - e^{-\tilde{\lambda}_j h} V_{k,j}^{(3)}| \\
&\quad + \mathbb{E} \left| \frac{1 - e^{-\lambda_j h}}{\lambda_j} f_N^j(V_k^{(2)}) - \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} f_N^j(V_k^{(3)}) \right| \\
&\quad + \mathbb{E} \left| \left[\left(\frac{q_j}{2\lambda_j} (1 - e^{-2\lambda_j h}) \right)^{1/2} - \left(\frac{q_j}{2\tilde{\lambda}_j} (1 - e^{-2\tilde{\lambda}_j h}) \right)^{1/2} \right] R_k^j \right|.
\end{aligned}$$

Going through this term by term, we have that

$$\begin{aligned}
&\mathbb{E} |e^{-\lambda_j h} V_{k,j}^{(2)} - e^{-\tilde{\lambda}_j h} V_{k,j}^{(3)}| \\
&\leq \mathbb{E} |e^{-\lambda_j h} V_{k,j}^{(2)} - e^{-\lambda_j h} V_{k,j}^{(3)}| + \mathbb{E} |e^{-\lambda_j h} V_{k,j}^{(3)} - e^{-\tilde{\lambda}_j h} V_{k,j}^{(3)}| \\
&= e^{-\lambda_j h} \mathbb{E} |V_{k,j}^{(2)} - V_{k,j}^{(3)}| + |e^{-\lambda_j h} - e^{-(\lambda_j + \epsilon_j^{(\lambda)})h}| \cdot \mathbb{E} |V_{k,j}^{(3)}|
\end{aligned}$$

$$\begin{aligned}
&\leq e^{-\lambda_j h} \varepsilon_{k,j}^{(3)} + e^{-\lambda_j h} |1 - e^{-\varepsilon_j^{(\lambda)} h}| \cdot \mathbb{E}|V_{k,j}^{(3)}| \\
&\leq e^{-\lambda_j h} \varepsilon_{k,j}^{(3)} + e^{-\lambda_j h} |1 - e^{-\varepsilon_j^{(\lambda)} h}| \cdot (\mathbb{E}|V_{k,j}^{(3)} - V_{k,j}^{(4)}| + \mathbb{E}|V_{k,j}^{(4)}|) \\
&\leq e^{-\lambda_j h} \varepsilon_{k,j}^{(3)} + e^{-\lambda_j h} |1 - e^{-\varepsilon_j^{(\lambda)} h}| \cdot (\varepsilon_{k,j}^{(4)} + \mathbb{E}|V_{k,j}^{(4)}|), \tag{3.3.6}
\end{aligned}$$

where we have made the last estimate since unlike $\mathbb{E}[V_k^{(3)}]$, $\mathbb{E}[V_k^{(4)}]$ can be approximated from the numerical implementation. Next, we have

$$\begin{aligned}
&\mathbb{E} \left| \frac{1 - e^{-\lambda_j h}}{\lambda_j} f_N^j(V_k^{(2)}) - \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} f_N^j(V_k^{(3)}) \right| \\
&\leq \mathbb{E} \left| \frac{1 - e^{-\lambda_j h}}{\lambda_j} f_N^j(V_k^{(2)}) - \frac{1 - e^{-\lambda_j h}}{\lambda_j} f_N^j(V_k^{(3)}) \right| \\
&\quad + \mathbb{E} \left| \frac{1 - e^{-\lambda_j h}}{\lambda_j} f_N^j(V_k^{(3)}) - \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} f_N^j(V_k^{(3)}) \right| \\
&= \frac{1 - e^{-\lambda_j h}}{\lambda_j} \mathbb{E}|f_N^j(V_k^{(2)}) - f_N^j(V_k^{(3)})| + \left| \frac{1 - e^{-\lambda_j h}}{\lambda_j} - \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \right| \cdot \mathbb{E}|f_N^j(V_k^{(3)})| \\
&\leq L \cdot \frac{1 - e^{-\lambda_j h}}{\lambda_j} \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(3)} \\
&\quad + \frac{1}{\lambda_j \tilde{\lambda}_j} \left[\varepsilon^{(\lambda)} + |e^{-\tilde{\lambda}_j h} \lambda_j - e^{-\lambda_j h} \tilde{\lambda}_j| \right] \cdot \left[\mathbb{E}|f_N^j(V_k^{(3)}) - f_N^j(V_k^{(4)})| + \mathbb{E}|f_N^j(V_k^{(4)})| \right] \tag{3.3.7}
\end{aligned}$$

$$\begin{aligned}
&\leq L \cdot \frac{1 - e^{-\lambda_j h}}{\lambda_j} \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(3)} \tag{3.3.8} \\
&\quad + \frac{1}{\lambda_j \tilde{\lambda}_j} \left[\varepsilon^{(\lambda)} + |e^{-\tilde{\lambda}_j h} \lambda_j - e^{-\lambda_j h} \tilde{\lambda}_j| \right] \cdot \left[L \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \mathbb{E}|f_N^j(V_k^{(4)})| \right].
\end{aligned}$$

Finally, we have (using that $\mathbb{E}|X| \leq [\mathbb{E}|X|^2]^{1/2}$)

$$\begin{aligned}
&\mathbb{E} \left| \left[\left(\frac{q_j}{2\lambda_j} (1 - e^{-2\lambda_j h}) \right)^{1/2} - \left(\frac{q_j}{2\tilde{\lambda}_j} (1 - e^{-2\tilde{\lambda}_j h}) \right)^{1/2} \right] R_k^j \right| \\
&= \left| \sqrt{\frac{q_j}{2\lambda_j} (1 - e^{-2\lambda_j h})} - \sqrt{\frac{q_j}{2(\lambda_j + \varepsilon_j^{(\lambda)} h)} (1 - e^{-2(\lambda_j + \varepsilon_j^{(\lambda)} h) h})} \right| \mathbb{E}|(R_k^j)^2|^{1/2}, \tag{3.3.9}
\end{aligned}$$

and $\mathbb{E}|(R_k^j)^2| = \text{Var}(R_k^j) = 1$. Summing (3.3.6), (3.3.8) and (3.3.9), we obtain (3.3.5). \square

Lemma 3.3.3 *We have*

$$\mathbb{E}\|V_k^{(3)} - V_k^{(4)}\| \leq \sum_{j=1}^N \left[\varepsilon_{k,j}^{(4)} + \varepsilon^{(\eta)} \mathbb{E}|V_{k,j}^{(4)}| \right] \quad (3.3.10)$$

and

$$\begin{aligned} \varepsilon_{k+1,j}^{(4)} &\leq e^{-\tilde{\lambda}_j h} \varepsilon_{k,j}^{(4)} + e^{-\tilde{\lambda}_j h} \varepsilon_j^{(\eta)} \mathbb{E}\|V_k^{(4)}\| \\ &\quad + \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \left[L \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \varepsilon_j^{(\eta)} \cdot \mathbb{E}\|f_N(V_k^{(4)})\| \right]. \end{aligned} \quad (3.3.11)$$

Proof. Suppose that we have approximate eigenfunctions

$$\tilde{e}_j = e_j + \eta_j, \quad \|\eta_j\| \leq \varepsilon_j^{(\eta)}.$$

Then, the Fourier-Galerkin expansion of $V_k^{(4)}$ also involves the approximate eigenfunctions, i.e. $V_k^{(4)} = \sum_{j=1}^N (e_j + \eta_j) V_{k,j}^{(4)}$, and the error for the function is

$$\begin{aligned} &\mathbb{E}\|V_k^{(3)} - V_k^{(4)}\| \\ &= \mathbb{E} \left\| \sum_{j=1}^N e_j V_{k,j}^{(3)} - \sum_{j=1}^N (e_j + \eta_j) V_{k,j}^{(4)} \right\| = \mathbb{E} \left\| \sum_{j=1}^N e_j (V_{k,j}^{(3)} - V_{k,j}^{(4)}) - \sum_{j=1}^N \eta_j V_{k,j}^{(4)} \right\| \\ &\leq \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \sum_{j=1}^N \varepsilon^{(\eta)} \cdot \mathbb{E}|V_{k,j}^{(4)}| = \sum_{j=1}^N \left[\varepsilon_{k,j}^{(4)} + \varepsilon^{(\eta)} \mathbb{E}|V_{k,j}^{(4)}| \right]. \end{aligned}$$

The error for the next coefficient errors is (with the wide tilde denoting inner products with $e_j + \eta_j$ instead of with e_j)

$$\begin{aligned} \varepsilon_{k+1,j}^{(4)} &= \mathbb{E} \left| e^{-\tilde{\lambda}_j h} V_{k,j}^{(3)} + \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} f_N^j(V_k^{(3)}) + \left(\frac{q_j}{2\tilde{\lambda}_j} (1 - e^{-2\tilde{\lambda}_j h}) \right)^{1/2} R_k^j \right. \\ &\quad \left. - \left[e^{-\tilde{\lambda}_j h} \widetilde{V_{k,j}^{(4)}} + \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \widetilde{f_N^j(V_k^{(4)})} + \left(\frac{q_j}{2\tilde{\lambda}_j} (1 - e^{-2\tilde{\lambda}_j h}) \right)^{1/2} R_k^j \right] \right| \\ &= \mathbb{E} \left| e^{-\tilde{\lambda}_j h} (V_{k,j}^{(3)} - \widetilde{V_{k,j}^{(4)}}) + \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} [f_N^j(V_k^{(3)}) - \widetilde{f_N^j(V_k^{(4)})}] \right|. \end{aligned}$$

Dealing with these terms separately, we have

$$\begin{aligned}
& e^{-\tilde{\lambda}_j h} \mathbb{E} |V_{k,j}^{(3)} - V_{k,j}^{(4)}| + e^{-\tilde{\lambda}_j h} \mathbb{E} |V_{k,j}^{(4)} - \widetilde{V_{k,j}^{(4)}}| \\
& \leq e^{-\tilde{\lambda}_j h} \varepsilon_{k,j}^{(4)} + e^{-\tilde{\lambda}_j h} \mathbb{E} |\langle V_k^{(4)}, e_j \rangle - \langle V_k^{(4)}, e_j + \eta_j \rangle| \\
& = e^{-\tilde{\lambda}_j h} \varepsilon_{k,j}^{(4)} + e^{-\tilde{\lambda}_j h} \mathbb{E} |\langle V_k^{(4)}, -\eta_j \rangle| \\
& \leq e^{-\tilde{\lambda}_j h} \varepsilon_{k,j}^{(4)} + e^{-\tilde{\lambda}_j h} \varepsilon_j^{(\eta)} \mathbb{E} \|V_k^{(4)}\|
\end{aligned} \tag{3.3.12}$$

and

$$\begin{aligned}
& \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \left[\mathbb{E} |f_N^j(V_k^{(3)}) - f_N^j(V_k^{(4)})| + \mathbb{E} |f_N^j(V_k^{(4)}) - \widetilde{f_N^j(V_k^{(4)})}| \right] \\
& \leq \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \left[\mathbb{E} |f_N^j(V_k^{(3)}) - f_N^j(V_k^{(4)})| + \mathbb{E} |\langle f_N(V_k^{(4)}), e_j \rangle - \langle f_N(V_k^{(4)}), e_j + \eta_j \rangle| \right] \\
& \leq \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \left[\mathbb{E} |f_N^j(V_k^{(3)}) - f_N^j(V_k^{(4)})| + \mathbb{E} |\langle f_N(V_k^{(4)}), -\eta_j \rangle| \right] \\
& \leq \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \left[L \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \varepsilon_j^{(\eta)} \cdot \mathbb{E} \|f_N(V_k^{(4)})\| \right].
\end{aligned} \tag{3.3.13}$$

Summing (3.3.12) and (3.3.13), we obtain (3.3.11). \square

Theorem 3.3.4 Suppose that we have fixed parameters T, N eigenfunctions, M time steps, a grid resolution R , Dirichlet eigenvalues $0 < \lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n$, eigenvalue error tolerance $\varepsilon^{(\lambda)}$, and eigenfunction error tolerance $\varepsilon^{(\eta)}$. We denote the Fourier coefficients $V_{k,j}^{(4)}$ of the approximated solution $V_k^{(4)}$ and the Fourier coefficients $f_N^j(V_k^{(4)})$ of $f_N(U_{t_k})$ in the matrices $\bar{V} = (V_{k,j}^{(4)})$, $f_N(\bar{V}) = (f_N^j(V_k^{(4)}))$, $k = 1, \dots, M$, $j = 1, \dots, N$. Then, for any $\varepsilon \in (0, 1)$, we have a bound for the strong error of the complete scheme at time T

$$\begin{aligned}
& \mathbb{E} \|U(T) - V_M^{(4)}\| \leq \mathcal{E}^s(T, N, M, R, L, \lambda_1, \lambda_N, \varepsilon^{(\lambda)}, \varepsilon^{(\eta)}, \bar{V}, f_N(\bar{V})) = \\
& \mathcal{E}_M^{(1)}(T, N, M, \lambda_N) + \mathcal{E}_M^{(234)}(T, N, M, R, L, \lambda_1, \lambda_N, \varepsilon^{(\lambda)}, \varepsilon^{(\eta)}, \bar{V}, f_N(\bar{V})),
\end{aligned} \tag{3.3.14}$$

where, for a constant $C_T > 0$,

$$\mathcal{E}_M^{(1)}(T, N, M, \lambda_N) \leq C_T \left(\lambda_N^{\varepsilon-1} + \frac{\log(M)}{M} \right) \tag{3.3.15}$$

and

$$\mathcal{E}_M^{(234)}(T, N, M, R, L, \lambda_1, \varepsilon^{(\lambda)}, \varepsilon^{(\eta)}, \bar{V}, f_N(\bar{V}))$$

$$\begin{aligned}
&\leq a_2^M \cdot \varepsilon_0^{(2)} + b_2 \cdot \frac{1 - a_2^M}{1 - a_2} + a_2^M \cdot \varepsilon_0^{(3)} + b_3 \cdot \frac{1 - a_2^M}{1 - a_2} + \tilde{a}_2^M \cdot \varepsilon_0^{(4)} + b_4 \cdot \frac{1 - \tilde{a}_2^M}{1 - \tilde{a}_2} + b_5 \\
&\rightarrow \frac{b_2}{1 - a_2} + \frac{b_3}{1 - a_2} + \frac{b_4}{1 - \tilde{a}_2} + b_5 \text{ as } M \rightarrow \infty, \text{ if } a_2 < 1 \text{ (since } \tilde{a}_2 \leq a_2 \text{)}.
\end{aligned} \tag{3.3.16}$$

where, for some constant $C > 0$,

$$\begin{aligned}
a_2 &= e^{-\lambda_1 h} + L \cdot N \cdot \frac{1 - e^{-\lambda_1 h}}{\lambda_1}, \quad \tilde{a}_2 = e^{-\tilde{\lambda}_1 h} + L \cdot N \cdot \frac{1 - e^{-\tilde{\lambda}_1 h}}{\tilde{\lambda}_1}, \\
b_2 &= \frac{1 - e^{-\lambda_1 h}}{\lambda_1} \cdot N \cdot C \cdot R^{-2}, \\
b_3 &= e^{-\lambda_1 h} |1 - e^{-\varepsilon^{(\lambda)} h}| \cdot \sup_{k=0, \dots, M} \left[\sum_{j=1}^N (\varepsilon_{k,j}^{(4)} + \mathbb{E}|V_{k,j}^{(4)}|) \right] \\
&\quad + \frac{1}{\lambda_1 \tilde{\lambda}_1} \left[\varepsilon^{(\lambda)} + |e^{-\tilde{\lambda}_1 h} \lambda_1 - e^{-\lambda_1 h} \tilde{\lambda}_1| \right] \\
&\quad \cdot \sup_{k=0, \dots, M} \left[N \cdot L \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \sum_{j=1}^N \mathbb{E}|f_N^j(V_k^{(4)})| \right] \\
&\quad + N \cdot \left| \sqrt{\frac{1}{2\lambda_1} (1 - e^{-2\lambda_1 h})} - \sqrt{\frac{1}{2(\lambda_1 + \varepsilon^{(\lambda)})} (1 - e^{-2(\lambda_1 + \varepsilon^{(\lambda)}) h})} \right| \\
b_4 &= N \cdot \sup_{j=1, \dots, N} \left\{ (1 + \mathbb{E}|V_{M,j}^{(4)}|) \right. \\
&\quad \cdot \left[e^{-\tilde{\lambda}_1 h} \cdot \varepsilon^{(\eta)} \cdot \sup_{k=0, \dots, M} \mathbb{E}\|V_k^{(4)}\| + \varepsilon^{(\eta)} \cdot \frac{1 - e^{-\tilde{\lambda}_1 h}}{\tilde{\lambda}_1} \cdot \sup_{k=0, \dots, M} \mathbb{E}\|f_N(V_k^{(4)})\| \right] \Big\}, \\
b_5 &= \varepsilon^{(\eta)} \cdot \sum_{j=1}^N \mathbb{E}|V_{M,j}^{(4)}|.
\end{aligned}$$

Remark 3.3.5 For the case of the simpler error bound with $M \rightarrow \infty$, the condition $a_2 < 1$ is equivalent to $L \cdot N < \lambda_1$. This turns out to be a rather strict condition (see Section 3.4). The error bound (3.3.14) requires very strict conditions on the parameters in order to be usable, we address this in Example 1. This strictness stems from some crude estimates in the proofs to simplify expressions, for example the step of replacing all λ_n by λ_1 in the proof of Theorem 3.3.4.

Proof. The error of the scheme can be split up as

$$\mathbb{E}(\|U(t_k) - V_k^{(4)}\|)$$

$$\begin{aligned}
&\leq \mathbb{E}(\|U(t_k) - V_k^{(1)}\|) + \mathbb{E}(\|V_k^{(1)} - V_k^{(2)}\|) + \mathbb{E}(\|V_k^{(2)} - V_k^{(3)}\|) + \mathbb{E}(\|V_k^{(3)} - V_k^{(4)}\|) \\
&=: \mathbb{E}(\|U(t_k) - V_k^{(1)}\|) + \varepsilon_k^{(2)} + \varepsilon_k^{(3)} + \varepsilon_k^{(4)} \leq \mathcal{E}_k^{(1)} + \mathcal{E}_k^{(2)} + \mathcal{E}_k^{(3)} + \mathcal{E}_k^{(4)}, \quad k = 1, \dots, M,
\end{aligned}$$

where the error bound $\mathcal{E}_M^{(1)}$ in (3.3.15) is from [59]. The other terms relate to the second error term above as

$$\mathcal{E}_k^{(234)} = \mathcal{E}_k^{(2)} + \mathcal{E}_k^{(3)} + \mathcal{E}_k^{(4)}, \quad k = 1, \dots, M.$$

The error bounds $\mathcal{E}_k^{(i)}$, $i = 2, 3, 4$, are established in Lemmas 3.3.1, 3.3.2 and 3.3.3 as

$$\varepsilon_k^{(2)} \leq \sum_{j=1}^N \varepsilon_{k,j}^{(2)} =: \mathcal{E}_k^{(2)}, \quad \varepsilon_k^{(3)} \leq \sum_{j=1}^N \varepsilon_{k,j}^{(3)} =: \mathcal{E}_k^{(3)}, \quad \varepsilon_k^{(4)} \leq \sum_{j=1}^N (1 + |V_{k,j}^{(4)}|) \varepsilon_{k,j}^{(4)} =: \mathcal{E}_k^{(4)}.$$

Furthermore, the inequalities (3.3.2), (3.3.5) and (3.3.11) establish relations between $\varepsilon_{k,j}^{(i)}$ and $\varepsilon_{k+1,j}^{(i)}$ for $k = 0, \dots, M-1$, $j = 1, \dots, N$ and $i = 2, 3, 4$. First, we consider in each time step the sum of all errors (summing over index j) and their bounds given by

$$\begin{aligned}
\sum_{j=1}^N \varepsilon_{k+1,j}^{(2)} &\leq \sum_{j=1}^N e^{-\lambda_j h} \varepsilon_{k,j}^{(2)} + \sum_{j=1}^N \frac{1 - e^{-\lambda_j h}}{\lambda_j} \left[L \cdot \sum_{i=1}^N \varepsilon_{k,i}^{(2)} + D \cdot R^{-2} \right], \\
\sum_{j=1}^N \varepsilon_{k+1,j}^{(3)} &\leq \sum_{j=1}^N e^{-\lambda_j h} \varepsilon_{k,j}^{(3)} + \sum_{j=1}^N e^{-\lambda_j h} |1 - e^{-\varepsilon^{(\lambda)} h}| \cdot (\varepsilon_{k,j}^{(4)} + \mathbb{E}|V_{k,j}^{(4)}|) \\
&\quad + \sum_{j=1}^N L \cdot \frac{1 - e^{-\lambda_j h}}{\lambda_j} \cdot \sum_{i=1}^N \varepsilon_{k,i}^{(3)} \\
&\quad + \sum_{j=1}^N \frac{1}{\lambda_j \tilde{\lambda}_j} \left[\varepsilon^{(\lambda)} + |e^{-\tilde{\lambda}_j h} \lambda_j - e^{-\lambda_j h} \tilde{\lambda}_j| \right] \cdot \left[L \cdot \sum_{i=1}^N \varepsilon_{k,i}^{(4)} + \mathbb{E}|f_N^j(V_k^{(4)})| \right] \\
&\quad + \sum_{j=1}^N \left| \sqrt{\frac{q_j}{2\lambda_j}} (1 - e^{-2\lambda_j h}) - \sqrt{\frac{q_j}{2(\lambda_j + \varepsilon^{(\lambda)})}} (1 - e^{-2(\lambda_j + \varepsilon^{(\lambda)}) h}) \right|, \\
\sum_{j=1}^N \varepsilon_{k+1,j}^{(4)} &\leq \sum_{j=1}^N e^{-\tilde{\lambda}_j h} \varepsilon_{k,j}^{(4)} + \sum_{j=1}^N e^{-\tilde{\lambda}_j h} \varepsilon^{(\eta)} \mathbb{E}\|V_k^{(4)}\| \\
&\quad + \sum_{j=1}^N \frac{1 - e^{-\tilde{\lambda}_j h}}{\tilde{\lambda}_j} \left[L \cdot \sum_{i=1}^N \varepsilon_{k,i}^{(4)} + \varepsilon^{(\eta)} \cdot \mathbb{E}\|f_N(V_k^{(4)})\| \right].
\end{aligned}$$

Since the terms in the above expressions involving λ_j like $e^{-\lambda_j h}$, $\frac{1-e^{-\lambda_j h}}{\lambda_j}$, $\frac{1}{\lambda_j \tilde{\lambda}_j} \left[\varepsilon^{(\lambda)} + |e^{-\tilde{\lambda}_j h} \lambda_j - e^{-\lambda_j h} \tilde{\lambda}_j| \right]$ and

$$\left| \sqrt{\frac{q_j}{2\lambda_j}} (1 - e^{-2\lambda_j h}) - \sqrt{\frac{q_j}{2(\lambda_j + \varepsilon^{(\lambda)})}} (1 - e^{-2(\lambda_j + \varepsilon^{(\lambda)})h}) \right|$$

are monotone decreasing in λ_j , we can bound all the above terms from above by replacing all λ_j by λ_1 . We obtain

$$\begin{aligned} \sum_{j=1}^N \varepsilon_{k+1,j}^{(2)} &\leq e^{-\lambda_1 h} \sum_{j=1}^N \varepsilon_{k,j}^{(2)} + N \cdot L \cdot \frac{1 - e^{-\lambda_1 h}}{\lambda_1} \sum_{j=1}^N \varepsilon_{k,j}^{(2)} + \frac{1 - e^{-\lambda_1 h}}{\lambda_1} \cdot N \cdot D \cdot R^{-2} \\ &= \left[e^{-\lambda_1 h} + N \cdot L \cdot \frac{1 - e^{-\lambda_1 h}}{\lambda_1} \right] \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(2)} + \frac{1 - e^{-\lambda_1 h}}{\lambda_1} \cdot N \cdot D \cdot R^{-2}, \end{aligned} \quad (3.3.17)$$

$$\begin{aligned} \sum_{j=1}^N \varepsilon_{k+1,j}^{(3)} &\leq e^{-\lambda_1 h} \sum_{j=1}^N \varepsilon_{k,j}^{(3)} + e^{-\lambda_1 h} |1 - e^{-\varepsilon^{(\lambda)} h}| \sum_{j=1}^N (\varepsilon_{k,j}^{(4)} + \mathbb{E}|V_{k,j}^{(4)}|) \\ &\quad + N \cdot L \cdot \frac{1 - e^{-\lambda_1 h}}{\lambda_1} \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(3)} \\ &\quad + \frac{1}{\lambda_1 \tilde{\lambda}_1} \end{aligned} \quad (3.3.18)$$

$$\begin{aligned} &\cdot \left[\varepsilon^{(\lambda)} + |e^{-\tilde{\lambda}_1 h} \lambda_1 - e^{-\lambda_1 h} \tilde{\lambda}_1| \right] \cdot \left[N \cdot L \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \sum_{j=1}^N \mathbb{E}|f_N^j(V_k^{(4)})| \right] \\ &\quad + N \cdot \left| \sqrt{\frac{1}{2\lambda_1}} (1 - e^{-2\lambda_1 h}) - \sqrt{\frac{1}{2(\lambda_1 + \varepsilon^{(\lambda)})}} (1 - e^{-2(\lambda_1 + \varepsilon^{(\lambda)})h}) \right| \\ &\leq \left[e^{-\lambda_1 h} + N \cdot L \cdot \frac{1 - e^{-\lambda_1 h}}{\lambda_1} \right] \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(3)} \\ &\quad + e^{-\lambda_1 h} |1 - e^{-\varepsilon^{(\lambda)} h}| \cdot \sup_{k=0, \dots, M} \left[\sum_{j=1}^N (\varepsilon_{k,j}^{(4)} + \mathbb{E}|V_{k,j}^{(4)}|) \right] \\ &\quad + \frac{1}{\lambda_1 \tilde{\lambda}_1} \left[\varepsilon^{(\lambda)} + |e^{-\tilde{\lambda}_1 h} \lambda_1 - e^{-\lambda_1 h} \tilde{\lambda}_1| \right] \\ &\quad \cdot \sup_{k=0, \dots, M} \left[N \cdot L \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \sum_{j=1}^N \mathbb{E}|f_N^j(V_k^{(4)})| \right] \\ &\quad + N \cdot \left| \sqrt{\frac{1}{2\lambda_1}} (1 - e^{-2\lambda_1 h}) - \sqrt{\frac{1}{2(\lambda_1 + \varepsilon^{(\lambda)})}} (1 - e^{-2(\lambda_1 + \varepsilon^{(\lambda)})h}) \right|, \end{aligned} \quad (3.3.19)$$

$$\begin{aligned}
\sum_{j=1}^N \varepsilon_{k+1,j}^{(4)} &\leq e^{-\tilde{\lambda}_1 h} \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + e^{-\tilde{\lambda}_1 h} \cdot N \cdot \varepsilon^{(\eta)} \cdot \mathbb{E} \|V_k^{(4)}\| \\
&\quad + N \cdot \frac{1 - e^{-\tilde{\lambda}_1 h}}{\tilde{\lambda}_1} \left[L \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} + \varepsilon^{(\eta)} \cdot \mathbb{E} \|f_N(V_k^{(4)})\| \right] \\
&\leq \left[e^{-\tilde{\lambda}_1 h} + N \cdot L \cdot \frac{1 - e^{-\tilde{\lambda}_1 h}}{\tilde{\lambda}_1} \right] \cdot \sum_{j=1}^N \varepsilon_{k,j}^{(4)} \\
&\quad + e^{-\tilde{\lambda}_1 h} \cdot N \cdot \varepsilon^{(\eta)} \cdot \sup_{k=0,\dots,M} \mathbb{E} \|V_k^{(4)}\| \\
&\quad + N \cdot \varepsilon^{(\eta)} \cdot \frac{1 - e^{-\tilde{\lambda}_1 h}}{\tilde{\lambda}_1} \cdot \sup_{k=0,\dots,M} \mathbb{E} \|f_N(V_k^{(4)})\|.
\end{aligned} \tag{3.3.20}$$

$$\begin{aligned}
&\quad + e^{-\tilde{\lambda}_1 h} \cdot N \cdot \varepsilon^{(\eta)} \cdot \sup_{k=0,\dots,M} \mathbb{E} \|V_k^{(4)}\| \\
&\quad + N \cdot \varepsilon^{(\eta)} \cdot \frac{1 - e^{-\tilde{\lambda}_1 h}}{\tilde{\lambda}_1} \cdot \sup_{k=0,\dots,M} \mathbb{E} \|f_N(V_k^{(4)})\|.
\end{aligned} \tag{3.3.21}$$

We can now see that the error sum at t_{k+1} can be bounded by an expression which is the image of the affine map

$$\psi_{a,b} : \mathbb{R} \rightarrow \mathbb{R}, \quad \psi_{a,b}(x) = ax + b$$

for $a > 0$, $b \in \mathbb{R}$. For a given start value, the behavior of the iteration $\psi_{a,b}^M(x_0)$, $M > 0$, depends on whether $a < 1$, $a = 1$ or $a > 1$. We have the following cases:

$$\psi_{a,b}^M(x_0) = \begin{cases} a^M x_0 + b \sum_{i=1}^{M-1} a^i = a^M x_0 + b \cdot \frac{1 - a^M}{1 - a} \xrightarrow{M \rightarrow \infty} \frac{b}{1 - a} & a < 1, \\ x_0 + Mb, & a = 1, \\ a^M \left(x_0 + \frac{b}{a - 1} \right) - \frac{b}{a - 1}, & a > 1. \end{cases} \tag{3.3.22}$$

For our three error sums, the constant a is given by a_2 and \tilde{a}_2 , and the constant b is given by b_2 , b_3 and b_4 , respectively. Applying the behavior of the iterated affine map $\psi_{a,b}$ as shown in (3.3.22) to (3.3.23) to the maps (3.3.17), (3.3.19) and (3.3.20) with the constants a_2 , \tilde{a}_2 , b_2 , b_3 and b_4 , as well as the bounds (3.3.1), (3.3.4) and (3.3.10) shown in the three lemmas, we obtain the terms shown in (3.3.16). \square

3.4 Numerical experiments

For the case where the function f associated to the operator F in equation 2.3.1 is linear and given by $f(x) = x$, we can compute an exact solution by taking the function inside the linear operator A , so that we consider the eigenvalues of the operator $A + I$ which are given by $-\lambda_j + 1$, $j \in \mathbb{N}$. For the linear function $f_1(x) = x$, we present convergence plots in Figure

3.4.1 and compare the approximated solution to the exact solution. For the nonlinear function $f_2(x) = \frac{1}{1+x^2}$, we additionally compare the approximate solution to a reference solution with $N = 400$ and $M = 100$ and show the convergence with respect to N in the second plot. Since the constant for the noise term scales nonlinearly with the step size (see the final term in (2.1)), we cannot compute a reference solution to compare the other approximations to, as we cannot sensibly merge the finer random increments into more rough ones. For the convergence plots with $f = f_1$, the error was averaged over 100 independent realizations, while for $f = f_2$, 10 independent realizations were computed. We can see in all cases that the convergence with respect to M and N is of order one. We note at this point that we were not able to verify that the nonlinear function f_2 indeed satisfies the conditions (2.3.3) to (2.3.5), but convergence to a reference solution is observed nevertheless. We also remark that if strictly nonlinear f is used, the computational effort is substantially higher since in every time step N two-dimensional numerical integrals have to be computed.

We now shed some light on how the error bound from Theorem 3.3.4 behaves here: We first note that for the Peanut shape it is $\lambda_1 \approx 6.5155$, so in order for the condition $a_2 < 1$ to be fulfilled we would need $L \cdot N < 6.5155$. This is a very strict condition as either N or L needs to be quite small; for instance, if $N = 100$, only nonlinearities with $L < 0.065155$ are covered by the case $M \rightarrow \infty$ in Theorem 3.3.4. But as this is not a necessary condition to use Theorem 3.3.4, we can still give bounds also if $a_2 > 1$.

Example 1 We give a concrete example for an error bound resulting from Theorem 3.3.4. Suppose $T = 0.1$, $N = 100$, $M = 50$, $R = 301$, $L = 0.01$, $\lambda_1 = 6.5155$, $|V_{k,j}^{(4)}| \leq 1$ for all $k = 0, \dots, M$, $j = 1, \dots, N$ (we have observed in practice that this has always been the case). We also assume $C = 1$ (as Simpson's rule is very accurate for smooth functions) and $\varepsilon_{k,j}^{(4)} \leq 1$ for all k, j (which is a very loose assumption, but does not matter much as the corresponding terms in b_3 are small anyway). We assume error tolerances $\varepsilon^{(\lambda)} = 2 \cdot 10^{-4}$ and $\varepsilon^{(\eta)} = 5 \cdot 10^{-6}$. We also assume $\varepsilon_0^{(2)} = \varepsilon_0^{(3)} = \varepsilon_0^{(4)} = 10^{-4}$ (as the initial condition is smooth, its approximation can be assumed to be accurate). With the above assumptions on the parameters, we obtain $a_2 \approx 0.98904$ (with \tilde{a}_2 , $b_2 \approx 2.19316 \cdot 10^{-6}$, $b_3 \approx 1.96257 \cdot 10^{-4}$, $b_4 \approx 1.83156 \cdot 10^{-4}$ and $b_5 = 5 \cdot 10^{-5}$ and Theorem 3.3.4 yields a total error bound of $\mathcal{E}_M^{(234)} \leq 0.2234$. While this is clearly not a satisfactory bound especially compared to $\mathcal{E}_M^{(1)}$, we note that this bound depends critically on the error tolerances $\varepsilon^{(\lambda)}$ and $\varepsilon^{(\eta)}$. For example, for $\varepsilon^{(\lambda)} = 2 \cdot 10^{-5}$ and $\varepsilon^{(\eta)} = 5 \cdot 10^{-7}$, leaving all other parameters unchanged, we obtain $\mathcal{E}_M^{(234)} \leq 0.02257$. This shows that the error bound is usable in practice, but only with an as yet infeasible effort of computation (see Figure 3.2.4).

In the plots in Figure 3.4.2, we give a visual impression of a solution of an SPDE (2.3.1) on the Peanut shape by showing realizations of the approximated solution again for noise with

$q_n = n^{-2}$, $A = \Delta$, $N = 400$, $M = 100$ on the asymmetric Peanut shape introduced in Section 3.2. For each simulation, the initial condition was picked to be the bump function

$$B_{x_1, x_2, y_1, y_2}(x, y) = \begin{cases} \exp\left(-\frac{1}{1-r^2}\right), & r^2 = \left(x - \frac{x_1+x_2}{2}\right)^2 + \left(y - \frac{y_1+y_2}{2}\right)^2 < 1, \\ 0, & r^2 \geq 1 \end{cases}$$

supported on an ellipse within the rectangle $[x_1, x_2] \times [y_1, y_2]$, where we picked $[x_1, x_2] = [0.4, 0.6]$, $[y_1, y_2] = [0.3, 0.5]$. The random input, i.e. the realizations of the Wiener increments was also the same for each simulation. Only the nonlinear function was changed for the different simulations: It was chosen to be a function

$$b_p(x) = \exp(-10 \cdot (x - p)^2), \quad (3.4.1)$$

with one maximum at $p \in \{-0.4, -0.2, 0.2, 0.4\}$.

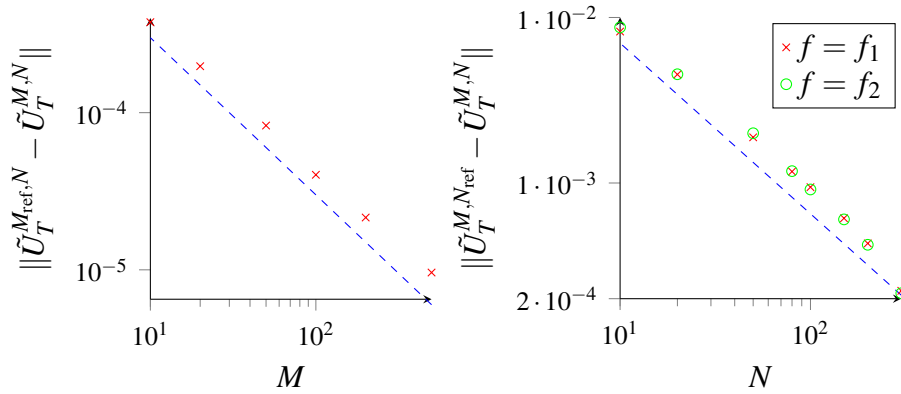


Figure 3.4.1: Convergence tests for the Peanut shape. The first plot shows the convergence with respect to M for the linear function $f_1(x) = x$, the second plot shows the convergence with respect to N for $f_1(x) = x$ as well as $f_2(x) = \frac{1}{1+x^2}$. The dashed lines in blue are reference lines for an order of convergence equal to one.

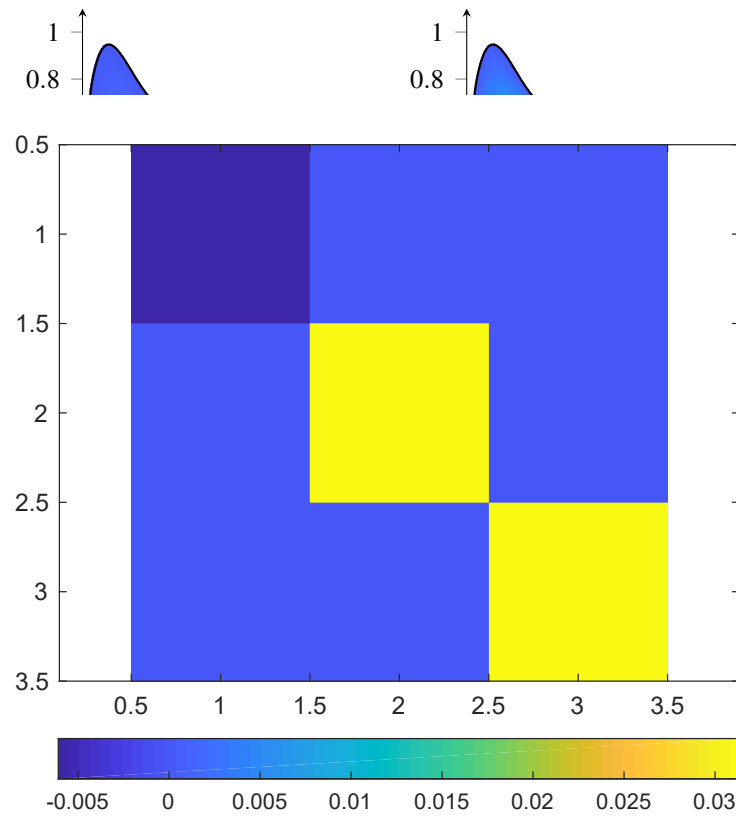


Figure 3.4.2: The approximated solution of the SPDE (2.3.1) on the Peanut shape for $N = 400$, $T = 0.1$, $M = 100$ and the nonlinear functions (3.4.1) for $p \in \{-0.4, -0.2, 0.2, 0.4\}$ from top-left to bottom-right. It appears that the process is ‘dimmed down’ from using nonlinear functions with a peak of higher absolute value.

Chapter 4

Polynomial chaos and exponential time differencing - Mathematical preliminaries

The second part of the thesis will be about PDEs with random inputs, rather than SPDEs which are driven by noise. More precisely, for a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and $\omega \in \Omega$, we will not deal with SPDEs of the form

$$dU(t, \omega) = [AU(t, \omega) + F(U(t, \omega))] + dW(t, \omega), \quad (4.0.1)$$

but rather with random PDEs of the form

$$\frac{\partial u(t, \omega)}{\partial t} = Au(t, \omega) + F(\omega, u(t, \omega)), \quad (4.0.2)$$

so that $F(\omega)$ is a (potentially nonlinear) function which has a random input. Hence, the solution U of (4.0.1) as well as the solution u of (4.0.2) are stochastic processes, but the randomness is introduced via the Wiener process W in (4.0.1) (see Chapter 2) and via a random function F in (4.0.2), usually by introducing a random coefficient. For instance, the Gray-Scott system

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = D_u \Delta u(x,t) - u(x,t)v(x,t)^2 + \tilde{F}(1 - u(x,t)), \\ \frac{\partial v(x,t)}{\partial t} = D_v \Delta v(x,t) + u(x,t)v(x,t)^2 - (\tilde{F} + k)v(x,t), \end{cases} \quad (4.0.3)$$

with $x \in (-1, 1)^d$, $t \in (0, T]$ and with some initial conditions $u(x, 0) = u_{\text{init}}$, $v(x, 0) = v_{\text{init}}$ constants $\tilde{F}, k, D_u, D_v > 0$ considered in Section 5.2.6 takes the form in (4.0.2) by making

the constants $\tilde{F}, k > 0$ random variables $\tilde{F}(\omega), k(\omega)$:

$$\begin{cases} \frac{\partial u(x,t,\omega)}{\partial t} = D_u \Delta u(x,t,\omega) - u(x,t,\omega)v(x,t,\omega)^2 + \tilde{F}(\omega)(1 - u(x,t,\omega)), \\ \frac{\partial v(x,t,\omega)}{\partial t} = D_v \Delta v(x,t,\omega) + u(x,t,\omega)v(x,t,\omega)^2 - (\tilde{F}(\omega) + k(\omega))v(x,t,\omega). \end{cases}$$

The method used in this work to deal with the random PDEs is the polynomial chaos method, which makes use of the polynomial chaos expansion (PCE). We give an introduction to non-intrusive and intrusive PCE (niPCE and iPCE) and give an overview of the literature on how PCE has been used for quantifying uncertainties in random PDEs like (4.0.2). Apart from the random Gray-Scott model (4.0.3) with periodic boundary conditions, we will mostly deal with random differential equations of the form

$$\frac{\partial u(x,t,\omega)}{\partial t} = D \Delta u(x,t,\omega) + F(\omega, u(x,t,\omega)), \quad u(x,0,\omega) = u_{\text{init}}(x), \quad (4.0.4)$$

where $x \in (-1,1)^d$, $d \in \{1,2\}$, $t \in (0,T]$ for some $T > 0$, $D = 0$ (so that (4.0.4) is an ODE) or $D = 1$ (so that (4.0.4) is a PDE) and $F(\omega, u(x,t,\omega)) = K(\omega)u^\kappa(x,t,\omega)$ where K is a uniformly distributed constant and $\kappa = 1$ (Sections 5.2.1 and 5.2.4), $\kappa = 2$ (Section 5.2.2) or $\kappa = 3$ (Section 5.2.3). We assume periodic boundary conditions for every differential equation considered in Chapters 4 and 5.

4.1 The polynomial chaos expansion

The term *chaos* in polynomial chaos was first used by Norbert Wiener in 1938 to denote a kind of random process. The general definition given in [101, §2] is that a chaos is a real- or vector-valued function $F(S; \alpha)$, where S is taken from a sufficiently rich class of subsets of \mathbb{R}^n , and $\alpha \in [0,1]$ is a representative point of an underlying probability space (see also [77, p. 480]). The modern term *chaos* in the context of chaos theory, which has nothing to do with polynomial chaos, came into use only in 1977 in a paper by mathematicians Tien-Yien Li and James A. Yorke [71]. From that point onwards it became associated with the high sensitivity of certain nonlinear dynamical systems to changes in their initial condition.

Wiener's original work described a stochastic process, i.e. a collection of random variables depending on a certain kind of random input. The PCE for a general second-order random process $u(\omega)$ for a Gaussian random vector $\xi(\omega) = (\xi_1(\omega), \xi_2(\omega), \dots, \xi_n(\omega))$ of possibly

infinite dimension n is given by (see [102, (3.1)])

$$u = a_0 H_0 + \sum_{i_1=1}^{\infty} a_{i_1} H_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1, i_2} H_2(\xi_{i_1}, \xi_{i_2}) \quad (4.1.1)$$

$$+ \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1, i_2, i_3} H_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \dots \quad (4.1.2)$$

$$+ \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \dots \sum_{i_n=1}^{i_{n-1}} a_{i_1, i_2, \dots, i_n} H_n(\xi_{i_1}, \xi_{i_2}, \dots, \xi_{i_n}) + \dots, \quad (4.1.3)$$

where the $a_{i_1}, a_{i_1, i_2}, \dots, a_{i_1, \dots, i_n}$ are real coefficients and the $H_n(\xi_{i_1}, \dots, \xi_{i_n})$ are n -dimensional Hermite polynomials. We provide more details on these polynomials below.

In the early 1990s, Roger Georges Ghanem and Pol Dimitrios Spanos used the PCE as an alternative to Monte Carlo simulations and Karhunen-Loève expansions [44, 45], from where research on PCE in numerical mathematics rapidly gained momentum.

While the original PCE is a spectral expansion of Gaussian random variables or random fields in terms of Hermite polynomials, expansions for other probability distributions are also possible. Dongbin Xiu and George Em Karniadakis showed in 2002 that this is the case for a class of probability distributions associated with families of orthogonal polynomials belonging to the Askey scheme of polynomials [6, 102]. Examples for continuous distributions and their respective polynomials are the normal distributions with Hermite polynomials, the uniform distribution with Legendre polynomials, the Gamma distribution with Jacobi polynomials and the Beta distribution with Laguerre polynomials. An overview of the different polynomial families and their corresponding distributions is given in Table 4.1.1. Each family $\{P_n(\zeta)\}_{n \in \mathbb{N}_0}$, $\zeta \in S \subset \mathbb{R}$ of orthogonal polynomials can be obtained by applying Gram-Schmidt orthogonalization to the set of monomials $\{\zeta^n\}_{n \in \mathbb{N}_0}$ with respect to the inner product

$$\langle f, g \rangle_w = \int_S f(\zeta) g(\zeta) w(\zeta) d\zeta. \quad (4.1.4)$$

on the probability space $L^2(S, \mu_w)$, where $w(x)$ is a weight function which is supported on a subset $S \subset \mathbb{R}$ and \mathcal{A}_1 is the σ -algebra generated by a random variable ξ with the given probability distribution (Gaussian, uniform, etc.). As a result, the orthogonality property

$$\int_S P_m(\zeta) P_n(\zeta) w(\zeta) d\zeta = h_n^2 \delta_{mn}, \quad m, n \in \mathbb{N}_0,$$

Distribution ξ	Polynomial P_n	Weight function $w(\zeta)$	Support S
Gaussian	Hermite H_n	$e^{-\zeta^2}$	$(-\infty, \infty)$
Uniform	Legendre L_n	$\frac{1}{2}$	$[-1, 1]$
Gamma	Jacobi $J_n^{(\alpha, \beta)}$	$(1 - \zeta)^\alpha (1 + \zeta)^\beta$	$[-1, 1]$
Beta	Laguerre $La_n^{(\alpha)}$	$e^{-\zeta} \zeta^\alpha$	$[0, \infty)$

Table 4.1.1: Continuous probability distributions associated with families of orthogonal polynomials, table from [102, Table 4.1], with $\alpha > -1$, $\beta > -1$. Note that Legendre and Jacobi polynomials can be linearly rescaled so that they are supported on an arbitrary closed interval $[a, b] \subset \mathbb{R}$ (see also Section 4.1.1).

holds, where the h_n are nonzero constants. Also, $\{P_n(\zeta)\}_{n \in \mathbb{N}_0}$ is a complete orthogonal system in $L^2(S, \mu_w)$, which is a generalization of the Cameron-Martin theorem [17, Theorem 1]. Each family of orthogonal polynomials satisfies a recurrence relation

$$P_{n+1}(\zeta) = (A_n \zeta + B_n) P_n(\zeta) - C_n P_{n-1}(\zeta), \quad n \geq 1, \quad (4.1.5)$$

which, together with $P_{-1}(\zeta) := 0$ and $P_0(\zeta) = 1$ uniquely determines all polynomials $P_n(\zeta)$, $n \geq 1$. Furthermore, $\{P_n(\zeta)\}_{n \in \mathbb{N}_0}$ satisfies the generalized Rodriguez formula

$$P_n(\zeta) = \frac{1}{w(\zeta)} \frac{d^n}{d\zeta^n} [w(\zeta) s^n(\zeta)], \quad (4.1.6)$$

where $s(\zeta)$ is a polynomial of at most second degree which, together with a polynomial $\tau(\zeta)$ of at most first degree, satisfies the differential equation

$$s(\zeta) y'' + \tau(\zeta) y' + \lambda y = 0; \quad \lambda = \lambda_n = -n\tau' - \frac{1}{2}n(n-1)s''.$$

For $d \in \mathbb{N}$, we define the multiindex set

$$\mathcal{J}_d := \left\{ \mathbf{n} = (n_1, n_2, \dots), \quad n_1, n_2, \dots \geq 0, \quad |\mathbf{n}| := \sum_{i=1}^{\infty} n_i = d \right\}.$$

For a dimension $S^d \subset \mathbb{R}^d$ and the d -fold product $\mu_w^d := \mu_w \otimes \dots \otimes \mu_w$, the multidimensional base for $L^2(S^d, \mu_w^d)$ is, for $\mathbf{x} = (x_1, \dots, x_d) \in S^d$, given by

$$\{P_{\mathbf{n}}^{(d)}(\mathbf{x})\}_{\mathbf{n} \in \mathcal{J}_d} = \left\{ \prod_{i=1}^d P_{n_i}(x_i) \right\}_{\mathbf{n} \in \mathcal{J}_d}. \quad (4.1.7)$$

Distribution ξ	Gaussian	Uniform
Polynomial P_n	Hermite	Legendre
Weight function $w(x)$	e^{-x^2}	1
Support S	\mathbb{R}	$[-1, 1]$
Recurrence coefficients A_n, B_n, C_n	$2, 0, 2(n+1)$	$\frac{2n+1}{n+1}, 0, \frac{n}{n+1}$
Rodriguez formula polynomial $s(x)$	$(-1)^n$	$\frac{(x^2-1)^n}{2^n n!}$

Table 4.1.2: Weight function, support, recurrence coefficients occurring in (4.1.5) and the term occurring in the Rodriguez formula (4.1.6) for different families of orthogonal polynomials

Note that the space (S^d, μ_w^d) is a probability space whose σ -algebra is generated by d i.i.d. random variables that have the given probability distribution ξ (normal, uniform, etc.). A random process $u(\omega)$ can be understood as an element of a space $L^2(\Omega, \mathcal{A}, \mu_w)$ where $(\Omega, \mathcal{A}, \mu_w)$ is a probability space whose σ -algebra \mathcal{A} is generated by a countable number of i.i.d. random variables $\{\xi_i\}_{i \in \mathbb{N}_0}$ with density w . It is also a result of Cameron and Martin (see e.g. [75, Theorem 5.1.12]) that

$$L^2(\Omega, \mathcal{A}, \mu_w) = \bigotimes_{i=0}^{\infty} L^2(S^d, \mu_w^d). \quad (4.1.8)$$

Let $\xi^{(d)} := (\xi_1, \dots, \xi_d)$. Combining (4.1.7) and (4.1.8), we obtain that as in the PCE (4.1.1), the gPCE for a general random process $u(\omega)$ for a multidimensional random input $\xi(\omega) = (\xi_1(\omega), \xi_2(\omega), \dots, \xi_n(\omega))$ of possibly infinite dimension n is

$$u(\xi) = a_0 P_0 + \sum_{\mathbf{n}_1 \in \mathcal{J}_1} a_{\mathbf{n}_1} P_{\mathbf{n}_1}^{(1)}(\xi^{(1)}) + \sum_{\mathbf{n}_2 \in \mathcal{J}_2} a_{\mathbf{n}_2} P_{\mathbf{n}_2}^{(2)}(\xi^{(2)}) + \sum_{\mathbf{n}_3 \in \mathcal{J}_3} a_{\mathbf{n}_3} P_{\mathbf{n}_3}^{(3)}(\xi^{(3)}) + \dots \quad (4.1.9)$$

$$= a_0 P_0 + \sum_{i_1=1}^n a_{i_1} P_1(\xi_{i_1}) + \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} a_{i_1, i_2} P_2(\xi_{i_1}, \xi_{i_2}) \quad (4.1.10)$$

$$+ \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1, i_2, i_3} P_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \dots \quad (4.1.11)$$

For convenience of notation, we will later use a series expansion in which the multi-indices in (4.1.9) are replaced by a term-based index, with a one-to-one correspondence between $a_{\mathbf{n}_i}$ and u_j and between $P_{\mathbf{n}_i}^{(i)}(\xi^{(i)})$ and $P_j(\xi)$, so that

$$u(\xi) = \sum_{j=0}^{\infty} u_j P_j(\xi). \quad (4.1.12)$$

In this thesis, we will be mostly concerned with cases where the random input is one-dimensional, i.e. $\xi = \xi$, and the gPCE simplifies to

$$u(\xi) = \sum_{j=0}^{\infty} u_j P_j(\xi). \quad (4.1.13)$$

4.1.1 Using shifted Legendre polynomials

In this section, we provide a few details on how Legendre polynomials can be used for a general uniformly distributed random variable $\xi \sim \mathcal{U}[a, b]$ for $a < b$, $a, b \in \mathbb{R}$ and how to adjust the inner product (4.1.4) accordingly, as we will later in this work use Legendre polynomials supported on different intervals $[a, b] \subset \mathbb{R}$.

We denote the canonical Legendre polynomials as L_n , $n \in \mathbb{N}_0$. Define the affine transformation

$$\phi^{(a,b)} : [a, b] \rightarrow [-1, 1], \phi^{(a,b)}(\zeta) = \frac{2}{b-a}\zeta + \frac{a+b}{a-b}, \quad (4.1.14)$$

$$(\phi^{(a,b)})^{-1} : [-1, 1] \rightarrow [a, b], (\phi^{(a,b)})^{-1}(\zeta) = \frac{b-a}{2}\zeta - \left[\frac{b-a}{2} - b \right]. \quad (4.1.15)$$

Using the canonical Legendre polynomials

$$L_n(\zeta) = \frac{1}{2^n} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \binom{n}{k} \binom{2n-2k}{n} \zeta^{n-2k}, \quad n \in \mathbb{N}_0,$$

defined on $[-1, 1]$, we define the Legendre polynomials $L_n^{(a,b)}$ defined on the interval $[a, b]$ by $L_n^{(a,b)}(\zeta) := L_n(\phi^{(a,b)}(\zeta))$. The polynomials $\{L_n^{(a,b)}\}_{n \in \mathbb{N}}$ then form an orthogonal basis of $L^2(\Omega, [a, b], \mu)$ of square-integrable random variables $X : (\Omega, \mathcal{A}, \mu) \rightarrow [a, b]$ where μ is the uniform measure with $\mu([a, b]) = 1$.

It is known that there is the orthogonality relation

$$\int_{-1}^1 L_m(\zeta) L_n(\zeta) d\zeta = \frac{2}{2n+1} \delta_{nm} \quad (4.1.16)$$

between the canonical Legendre polynomials. We are using the inner product (with the density function $\frac{1}{2} \mathbb{1}_{[-1, 1]}$ for a random variable $\xi \sim \mathcal{U}[-1, 1]$ in $L^2(\Omega, [-1, 1], \mu)$

$$\langle L_n(\xi), L_m(\xi) \rangle = \mathbb{E}[L_n(\xi) L_m(\xi)] = \frac{1}{2} \int_{-1}^1 L_n(\zeta) L_m(\zeta) d\zeta. \quad (4.1.17)$$

In order to obtain an orthonormal basis, we can use the normalized canonical Legendre polynomials defined as (using (4.1.16))

$$\hat{L}_n = \sqrt{\frac{2n+1}{2}} L_n, \quad n \in \mathbb{N}_0.$$

We then apply the affine transformations (4.1.14) to create normalized Legendre polynomials $\hat{L}_i^{(a,b)}$ defined on the interval $[a, b]$. The polynomials defined on $[a, b]$ will still be normalized because

$$\begin{aligned} & \frac{1}{b-a} \int_a^b \hat{L}_n^{(a,b)}(\zeta) \hat{L}_n^{(a,b)}(\zeta) d\zeta \\ &= \frac{1}{b-a} \int_a^b \hat{L}_n(\phi^{(a,b)}(\zeta)) \hat{L}_n(\phi^{(a,b)}(\zeta)) d\zeta \\ &= \frac{1}{b-a} \int_{-1}^1 \hat{L}_n(\zeta) \hat{L}_n(\zeta) \frac{1}{(\phi^{(a,b)})'} d\zeta \\ &= \frac{1}{2} \int_{-1}^1 \hat{L}_n(\zeta) \hat{L}_n(\zeta) d\zeta = 1 \end{aligned}$$

since $(\phi^{(a,b)})' = \frac{2}{b-a}$, and using (4.1.16). We will henceforth always denote the normalized Legendre polynomials by L_n and $L_n^{(a,b)}$.

We will have to compute integrals of the form

$$\int_a^b L_n^{(a,b)}(\zeta) L_m^{(a,b)}(\zeta) \cdot g(\zeta) d\zeta \quad (4.1.18)$$

later, but it will be easier to stick with the canonical Legendre polynomials since we have an addition theorem for them (see (5.1.2)). Therefore, we will express (4.1.18) in terms of an integral involving L_n and L_m . Let $g(\zeta) = \frac{1}{b-a} \mathbb{1}_{[a,b]}(\zeta) \cdot \zeta$ be the density function multiplied with ζ which is needed in the integral. It is

$$\int_a^b L_n^{(a,b)}(\zeta) L_m^{(a,b)}(\zeta) \cdot g(\zeta) d\zeta \quad (4.1.19)$$

$$= \int_a^b L_n(\phi^{(a,b)}(\zeta)) L_m(\phi^{(a,b)}(\zeta)) \cdot g(\zeta) d\zeta \quad (4.1.20)$$

$$= \int_{\phi^{(a,b)}(a)}^{\phi^{(a,b)}(b)} L_n(\zeta) L_m(\zeta) \cdot g((\phi^{(a,b)})^{-1}(\zeta)) \cdot \frac{b-a}{2} d\zeta \quad (4.1.21)$$

$$= \int_{-1}^1 L_n(\zeta) L_m(\zeta) \cdot \left[g\left(\frac{b-a}{2}\zeta - \left(\frac{b-a}{2} - b\right)\right) \right] \cdot \frac{b-a}{2} d\zeta \quad (4.1.22)$$

$$= \left(\frac{b-a}{2}\right) \int_{-1}^1 L_n(\zeta) L_m(\zeta) \cdot g\left(\frac{b-a}{2}\zeta - \left(\frac{b-a}{2} - b\right)\right) d\zeta \quad (4.1.23)$$

$$= \left(\frac{b-a}{2} \right) \int_{-1}^1 L_n(\zeta) L_m(\zeta) \cdot \frac{1}{b-a} \cdot \left(\frac{b-a}{2} \zeta - \left(\frac{b-a}{2} - b \right) \right) d\zeta \quad (4.1.24)$$

$$= \frac{1}{2} \int_{-1}^1 L_n(\zeta) L_m(\zeta) \cdot \left(\frac{b-a}{2} \zeta - \left(\frac{b-a}{2} - b \right) \right) d\zeta \quad (4.1.25)$$

where in the second step integration by substitution and the fact that $(\phi^{(a,b)}(\zeta))' = \frac{2}{b-a}$ was used (the inverse of this term appears in the next integral) and in the next to last step (4.1.29) was used. We have thus expressed the integral involving $L_n^{(a,b)}$ in terms of one involving canonical L_n .

We note additionally that if we use a uniform distribution on $[a, b]$, we have the density $g(\zeta) = \frac{1}{b-a} \mathbb{1}_{[a,b]}(\zeta)$ and we have

$$\frac{b-a}{2} \zeta - \left(\frac{b-a}{2} - b \right) \in [a, b] \quad (4.1.26)$$

$$\Leftrightarrow \frac{b-a}{2} \zeta \in \left[-\frac{b-a}{2}, \frac{b-a}{2} \right] \quad (4.1.27)$$

$$\Leftrightarrow \zeta \in [-1, 1], \quad (4.1.28)$$

so that

$$\mathbb{1}_{[a,b]} \left(\frac{b-a}{2} \zeta - \left(\frac{b-a}{2} - b \right) \right) = \mathbb{1}_{[-1,1]}(\zeta). \quad (4.1.29)$$

4.2 Numerical treatment of PDEs with uncertainties using PCE

In the following, we recall the two main methods of using the PCE (4.1.9). For the rest of the thesis, we will assume that all orthogonal polynomials in the family $\{P_n\}_{n \in \mathbb{N}}$ are normed, i.e. $\langle P_n, P_n \rangle = 1$, $n \in \mathbb{N}_0$. We will also occasionally use the shorthand $\langle P_n \rangle := \langle P_n, P_n \rangle$.

4.2.1 Non-intrusive PCE

An overview of niPCE approaches can be found in [35]. In niPCE, one treats the simulation of the solution of a PDE as a black box and the computation of properties of the random solution process is based on sampling the random input in a certain way. Concretely, the computation of moments of the solution like the mean or variance involves the computation of inner products, i.e. the numerical approximation of integrals, which can be done using different sampling methods. To this end, we will consider the classical Monte Carlo (MC)

method, a quasi Monte Carlo method (QMC) and Gaussian quadrature.

We use the expansion (4.1.13) and truncate it to $N + 1$ terms:

$$u(\xi) \approx \sum_{j=0}^N u_j P_j(\xi).$$

The PCE coefficients u_j can be retrieved by performing a spectral projection: Using the inner product

$$\langle P_i(\xi), P_j(\xi) \rangle_w = \int_S P_i(\zeta) P_j(\zeta) w(\zeta) d\zeta = \mathbb{E}[P_i(\xi) P_j(\xi)] \quad (4.2.1)$$

it follows

$$u_j = \langle R, P_j \rangle_w = \int_S R(\zeta) P_j(\zeta) w(\zeta) d\zeta, \quad (4.2.2)$$

where w is the weight function corresponding to the underlying probability distribution of the gPCE (see Section 4.1). The idea of niPCE is to compute the coefficient functions u_i in (4.2.2) as numerical integrals

$$u_i(x, t) = \int_a^b u(x, t, \xi) P_i(\xi) w(\xi) d\xi \approx \sum_{j=1}^q w_j u(x, t, \xi_j) P_i(\xi_j) w_j, \quad (4.2.3)$$

(with $\langle P_i^2 \rangle = 1$, see above) where the integral in (4.2.3) is a Banach space-valued integral (for example a Bochner integral, see [36, Appendix C]) and the points ξ_j and weights w_j , $j = 1, \dots, q$ are chosen according to the used method. In this work, we use three different methods: A naive Monte Carlo (MC) method (see Section 4.2.1.1), a quasi-Monte Carlo (QMC) method (see Section 4.2.1.2) and Gaussian quadrature (see Section 4.2.1.3). An illustration of these three methods with the different quadrature points x_j and weights w_i is shown in Figure 4.2.1.

We will mainly be interested in computing the expected value $\mathbb{E}[u(x, t, \xi)]$ and the variance $\text{Var}[u(x, t, \xi)]$ of the solution. It is

$$\mathbb{E}[u(x, t, \xi)] = u_0(x, t), \quad \text{Var}[u(x, t, \xi)] = \sum_{i=1}^{\infty} |u_i(x, t)|^2 \quad (4.2.4)$$

due to $\mathbb{E}[P_0(\xi)] = 1$, $\mathbb{E}[P_i(\xi)] = 0$ for $i > 0$ (P_0 is equal to one, so $\mathbb{E}[P_i(\xi)] = \mathbb{E}[P_0(\xi) P_i(\xi)] = 0$ for $i > 0$ due to orthogonality with the inner product (4.2.1)). The variance formula follows

due to

$$\begin{aligned}
\text{Var}[u(x, t, \xi)] &= \mathbb{E} \left[\left(\sum_{i=0}^{\infty} u_i P_i(\xi) \right)^2 \right] - \left[\mathbb{E} \left(\sum_{i=0}^{\infty} u_i P_i(\xi) \right) \right]^2 \\
&= \mathbb{E} \left[\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} u_i u_j P_i(\xi) P_j(\xi) \right] - u_0^2 \\
&= \mathbb{E} \left[\sum_{i=0}^{\infty} u_i^2 P_i(\xi)^2 \right] - u_0^2 = \sum_{i=1}^{\infty} |u_i(x, t)|^2.
\end{aligned}$$

We also note that the periodic boundary conditions for u are transferred directly to the PCE base functions: Inserting the periodic boundary condition $u(-1, t) = u(1, t)$, $t \in [0, T]$, into (4.1.12) and then multiplying with $P_j(\xi)$ and taking $\mathbb{E}[\cdot]$ yields

$$\sum_{i=0}^{\infty} u_i(-1, t) P_i(\xi) = \sum_{i=0}^{\infty} u_i(1, t) P_i(\xi) \Rightarrow u_j(-1, t) = u_j(1, t), \quad j \in \mathbb{N}_0, t \in [0, T] \quad (4.2.5)$$

due to the orthonormality of $\{P_i\}_{i \in \mathbb{N}_0}$. In higher spatial dimensions, the above argument can easily be replicated by applying (4.2.5) to any pair of points x_1, x_2 on the boundary for which $u(x_1, t) = u(x_2, t)$ holds.

4.2.1.1 The classical Monte Carlo method

A classical way to compute the integral (4.2.2) is the Monte Carlo method: The random vector ξ is sampled independently M_{MC} times according to its probability distribution, yielding random realizations $(\xi_1, \dots, \xi_{M_{\text{MC}}})$. The central limit theorem guarantees that this approximation is convergent with an error of $\mathcal{O}(M_{\text{MC}}^{-1/2})$ independent of the dimension of the random vector ξ . While P_j and w are easily evaluated at ξ_i , the computation of $R(\xi_i)$ is more expensive, since an ODE or PDE might have to be solved. Therefore, it is beneficial to use methods which obtain a good accuracy with as few samples ξ_i as possible.

4.2.1.2 The quasi-Monte Carlo method

One drawback of classical MC is that the randomly sampled points can cluster in certain areas, while leaving gaps in other areas, so that the parameter space is unevenly covered, as can be seen on the left-hand side in Figure 4.2.1. Quasi-Monte Carlo methods use so-called low-discrepancy sequences, which aim to cover the space more evenly. Examples for such sequences are Sobol sequences [91] (see the second plot in Figure 4.2.1) or Halton sequences [51].

Given a sequence $x_1, \dots, x_{\hat{M}}$ of \hat{M} points in the s -dimensional unit cube I^s , the Koksma-Hlawka inequality [53, Satz 1] states that

$$\left| \int_{I^s} f(x) \, dx - \frac{1}{\hat{M}} \sum_{i=1}^{\hat{M}} f(x_i) \right| \leq V(f) D_{\hat{M}}^*,$$

where $V(f)$ is the variation of f in the sense of Hardy and Krause (see e.g. [81, p. 967]). With the counting function $A(E; \hat{M}) := \sum_{i=1}^{\hat{M}} \mathbb{1}_E(x_i)$ and the s -dimensional Lebesgue measure $|E|$ of a set E ,

$$D_{\hat{M}}^* = \sup_{E \in \mathcal{A}} \left| \frac{A(E; \hat{M})}{\hat{M}} - |E| \right|, \quad \mathcal{A} := \{[0, t_1) \times \dots \times [0, t_s) \mid t_1, \dots, t_s \in (0, 1)\}$$

is the discrepancy of the set $\{x_1, \dots, x_{\hat{M}}\} \subset I^s$. It has been shown [50] that for any dimension $s \geq 1$, there exists an infinite sequence of points in I^s such that (even for D_M ; it is $D_M^* \leq D_M$)

$$D_M^* = \mathcal{O}(M^{-1}(\log M)^s),$$

which can be shown to hold for Sobol sequences and Halton sequences [81, p. 980]. A drawback compared to classical MC is that sampling the low-discrepancy sequences can become infeasible if the dimension d of ξ is very high.

4.2.1.3 Gaussian quadrature

As opposed to the MC and QMC methods shown above, numerical integration with Gaussian quadrature is done by taking a weighted sum of function values at specified points, rather than an average where every function value has the same weight. Gauss quadrature aims to approximate integrals of the form

$$\int_S f(x) w(x) \, dx \approx \sum_{i=1}^q w_i f(x_i),$$

where $S \subset \mathbb{R}$ is the support of f and the weight function w , w_i are called the weights and x_i are called the quadrature nodes or abscissas for $i = 1, \dots, q$, $q \in \mathbb{N}$. The weights and nodes depend on the choice of the weight function, which can be chosen as a weight function associated with a family of orthogonal polynomials $\{P_n\}_{n \in \mathbb{N}_0}$ (see Section 4.1). The quadrature nodes x_1, \dots, x_n are then the roots of the polynomial P_n and the weights are given by the formula

$$w_i = \frac{a_n}{a_{n-1}} \frac{\langle P_{n-1}, P_{n-1} \rangle_w}{P_n'(x_i) P_{n-1}(x_i)},$$

where a_i is the coefficient of x^i in P_n . For later numerical computations, we will use an efficient and numerically stable way to obtain the Gauss quadrature abscissas and weights called the Golub-Welsch algorithm, devised by Gene H. Welsch and John H. Golub [46] and explained in [85, p. 188]. Gaussian quadrature rules are known to exhibit spectral convergence, meaning that the error of a Gauss quadrature rule on $I \subset \mathbb{R}$ using $q \in \mathbb{N}$ quadrature points for a function $f \in \mathcal{C}^r(I)$, $r \in \mathbb{N}$, behaves as $\mathcal{O}(q^{-r})$.

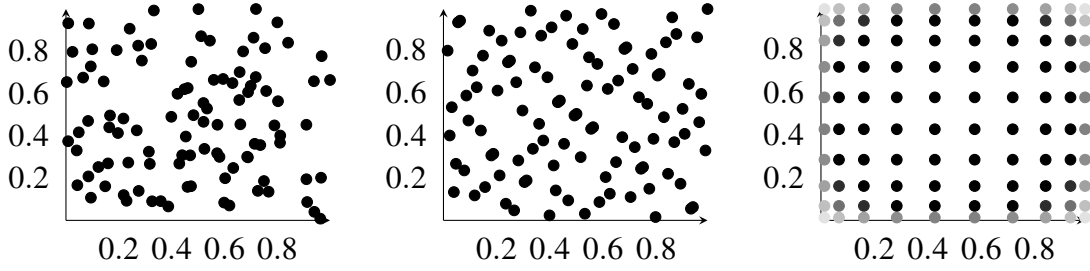


Figure 4.2.1: Three plots showing 100 points sampled in the unit square according to classical Monte Carlo (left), quasi-Monte Carlo with a Sobol sequence (middle) and Gauss-Legendre quadrature with different weights (right).

4.2.2 Intrusive PCE

In Section 4.2.1, we explained that in niPCE methods, obtaining samples of the random quantity of interest R – for example by solving a PDE – is treated as a black box and the PCE is applied afterwards. Therefore, the main effort lies in sampling R in a ‘good’ way.

Suppose that we aim to solve a given ODE or PDE

$$\frac{\partial u(t, x, \boldsymbol{\xi}(\omega))}{\partial t} = Au(t, x, \boldsymbol{\xi}(\omega)) + F(\boldsymbol{\xi}(\omega))(u(t, x, \boldsymbol{\xi}(\omega))),$$

depending on a random quantity $\boldsymbol{\xi}$. In iPCE methods, the gPCE (4.1.12) with time- and space-dependent coefficients $u_j(x, t)$, $j \geq 0$, is inserted into (4.0.2) right away, so that we obtain

$$\sum_{j=0}^{\infty} \frac{\partial}{\partial t} u_j(x, t) P_j(\boldsymbol{\xi}) = \sum_{j=0}^{\infty} Au_j(x, t) P_j(\boldsymbol{\xi}) + F(\boldsymbol{\xi}) \left[\sum_{j=0}^{\infty} u_j(x, t) P_j(\boldsymbol{\xi}) \right].$$

At this point a Galerkin projection for each $\boldsymbol{\eta} \in \mathbb{N}_0$ is performed which involves multiplying by $P_{\boldsymbol{\eta}}(\boldsymbol{\xi})$ and taking the expected value \mathbb{E}_w in the underlying probability space $L^2(\Omega, \mathcal{A}, \mu_w)$

equipped with $\langle f, g \rangle_w = \mathbb{E}_w[fg] := \int_S f(x)g(x)w(x) dx$. We obtain

$$\begin{aligned} & \mathbb{E} \left[\sum_{j=0}^{\infty} \frac{\partial}{\partial t} u_j(x, t) P_j(\xi) P_{\eta}(\xi) \right] \\ &= \mathbb{E} \left[\sum_{j=0}^{\infty} A u_j(x, t) P_j(\xi) P_{\eta}(\xi) \right] + \mathbb{E} \left[F(\xi) \left[\sum_{j=0}^{\infty} u_j(x, t) P_j(\xi) \right] P_{\eta}(\xi) \right]. \end{aligned}$$

We still assume that the polynomials are normed, i.e. $\langle P_m, P_n \rangle_w = \delta_{mn}$, $m, n \in \mathbb{N}_0$, so it follows that

$$\frac{\partial}{\partial t} u_{\eta}(x, t) = A u_{\eta}(x, t) + \mathbb{E} \left[F(\xi) \left[\sum_{j=0}^{\infty} u_j(x, t) P_j(\xi) \right] P_{\eta}(\xi) \right], \quad \eta = 0, 1, 2, \dots \quad (4.2.6)$$

Now, the series in the system of equations (4.2.6) is truncated to $N + 1$ terms and only the equations for $\eta = 0, \dots, N$ are considered, so we obtain

$$\frac{\partial}{\partial t} u_{\eta}(x, t) \approx A u_{\eta}(x, t) + \mathbb{E} \left[F(\xi) \left[\sum_{j=0}^N u_j(x, t) P_j(\xi) \right] P_{\eta}(\xi) \right], \quad \eta = 0, 1, 2, \dots, N. \quad (4.2.7)$$

The system of equations is then discretized, and a potential solver for a PDE system can be used after it is modified appropriately to deal with the PCE system (4.2.7). The appearance and further development heavily depends on the chosen discretization in space and time, and on the nonlinear function F .

As this requires the modification of existing solvers, the implementation of iPCE is more complicated and also potentially limited as it might interfere with certain characteristics of the deterministic system which are exploited in solvers (we will discuss one such case in Remark 5.1.1 where a random diffusion coefficient cannot be handled easily by a spectral iPCE method).

4.3 Numerical schemes for deterministic equations

We will now recall the deterministic numerical schemes for solving the deterministic initial value problem

$$\begin{cases} \frac{\partial u(x, t)}{\partial t} = D \Delta u(x, t) + F(u(x, t)), \\ u(x, 0) = u_{\text{init}}(x) \end{cases} \quad (4.3.1)$$

with $x \in (-1, 1)^d$, $t \in (0, T]$, $T > 0$, $D \geq 0$, a potentially nonlinear function F and periodic boundary conditions. These schemes can be used directly for niPCE (see Section 4.2.1) or modified to be used in iPCE (see Section 5.1). We start with the well-known explicit Euler scheme in Section 4.3.1, for example as introduced in [55, pp. 240], secondly recall the ETD-RDP and ETD-RDP-IF schemes in Section 4.3.2 presented in [5], and finally the ETDRK4 scheme presented in [62]. We will, for the rest of the thesis, denote by k the step size in time and by h the step size of the spatial grid.

4.3.1 Explicit Euler scheme

The explicit Euler scheme or forward Euler method is one of the oldest numerical procedures for solving initial value problems. For the explicit Euler scheme and the ETD-RDP-IF scheme covered in Section 4.3.2, we will use a second-order central finite difference discretization in space. We denote by p the spatial resolution in each dimension, so that in one dimension we have the discretized function $\mathbf{u}(t) = (u(x_0, t), u(x_1, t), \dots, u(x_p, t))^T$ where $x_i = -1 + 2i/p$, $i = 0, \dots, p$, $t \in [0, T]$. At one given point t_n in time, the second-order central finite difference approximation is given by

$$u''(x_i, t_n) \approx \frac{1}{k^2} [u(x_{i-1}, t_n) - 2u(x_i, t_n) + u(x_{i+1}, t_n)].$$

Applied to the discretized function \mathbf{u} , the second-order finite difference approximation of the Laplacian with periodic boundary conditions is given by the matrix

$$A_p = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & & 1 \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{p \times p}.$$

Also, the second-order central finite difference discretization of the two-dimensional Laplacian with periodic boundary conditions is given by $A = A_1 + A_2$ with $A_1 := I_{N+1} \otimes A_p$ and $A_2 := A_p \otimes I_{N+1}$ (see, for example, [5, p. 3]). Denoting $\mathbf{u}^n = (u(x_0, t_n), u(x_1, t_n), \dots, u(x_p, t_n))^T$ for $n = 0, \dots, M$, $t_n = n \cdot k$, the explicit Euler scheme for solving (4.0.4) relies on taking the Taylor expansion

$$\mathbf{u}^{n+1} = k \cdot \frac{\partial \mathbf{u}^n}{\partial t} + \frac{1}{2} k^2 \frac{\partial^2 \mathbf{u}^n}{\partial t^2} + \mathcal{O}(k^3)$$

and after truncating the expansion after the linear term and substituting $\frac{\partial \mathbf{u}^n}{\partial t} = D\mathbf{A}\mathbf{u}^n + F(\mathbf{u}^n)$ from (4.3.1), we obtain the explicit Euler scheme (for the 2D case, A_p must be replaced with A)

Algorithm 4.1: Deterministic finite differences explicit Euler scheme

$$\mathbf{u}^{n+1} = \mathbf{u}^n + k(DA_p\mathbf{u}^n + F(\mathbf{u}^n)), \quad n = 0, \dots, M-1.$$

4.3.2 ETD and ETD-RDP-IF

The ETD-RDP-IF scheme presented in [5] is a second-order exponential time-differencing (ETD) scheme which makes use of the approximation of the matrix exponentials by rational functions having real distinct poles (RDP). Additionally, if the spatial dimension is greater than one, the scheme uses dimensional splitting with an integrating factor (IF) approach, making it the ETD-RDP-IF scheme.

Applying the finite difference discretization to (4.3.1), using the matrix A from above, or using $A = A_p$ in one spatial dimension, yields the system

$$\frac{\partial \mathbf{u}}{\partial t} + A\mathbf{u} = F(\mathbf{u}), \quad \mathbf{u}^0 = \mathbf{u}_{\text{init}}. \quad (4.3.2)$$

In the multidimensional case, an integrating factor approach is used to solve (4.3.2). Considering the two-dimensional case and $A = A_1 + A_2$ (using notation from above), we introduce the time-dependent function $\mathbf{v}(t) = e^{A_1 t} \mathbf{u}$ and note for the time derivative \mathbf{v}_t that

$$\mathbf{v}_t = e^{A_1 t} \mathbf{u}_t + A_1 e^{A_1 t} \mathbf{u}. \quad (4.3.3)$$

Using the fact that A and A_1 commute (see [5, Lemma 1]) and therefore A and $e^{A_1 t}$ commute, inserting (4.3.3) into (4.3.2) yields

$$\begin{aligned} \mathbf{v}_t &= e^{A_1 t} (F(\mathbf{u} - A\mathbf{u}) + A_1 e^{A_1 t} \mathbf{u}) = e^{A_1 t} f(\mathbf{u}) - e^{A_1 t} A\mathbf{u} + A_1 e^{A_1 t} \mathbf{u} \\ &= e^{A_1 t} F(\mathbf{u}) - A e^{A_1 t} \mathbf{u} + A_1 e^{A_1 t} \mathbf{u} = e^{A_1 t} F(\mathbf{u}) - A_2 e^{A_1 t} \mathbf{u} \\ &= e^{A_1 t} F(e^{-A_1 t} \mathbf{v}) - A_2 \mathbf{v}. \end{aligned}$$

We obtain

$$\mathbf{v}_t + A_2 \mathbf{v} = G(\mathbf{v}), \quad \mathbf{v}_0 = \mathbf{u}_{\text{init}} \quad (4.3.4)$$

with $G(\mathbf{v}) = e^{A_1 t} F(e^{-A_1 t} \mathbf{v})$.

For the time discretization, we consider an exponential integrator or exponential time differencing (ETD) scheme. An overview is given in [54]. They rely on using representations of the exact solutions of initial value problems by a variation-of-constants formula (similar to the expressions (2.2.6), (2.2.8) and (2.3.2) from Chapter 2) and subsequently approximating the integral and the matrix exponentials appearing in the formula. For the semilinear problem (4.3.4), the variation-of-constants formula (also called Duhamel's principle) is given by

$$\mathbf{v}(t_n) = e^{-t_n A_2} \mathbf{v}(t) + \int_0^{t_n} e^{-A_2(t_n-s)} G(\mathbf{v}(s)) \, ds \quad (4.3.5)$$

or, restricting the formula to a single time step,

$$\mathbf{v}(t_{n+1}) = e^{-k A_2} \mathbf{v}(t_n) + \int_{t_n}^{t_{n+1}} e^{-A_2(t_{n+1}-s)} G(\mathbf{v}(s)) \, ds. \quad (4.3.6)$$

After a change of variables with $s = k\tau$, $\tau \in [0, 1]$, we obtain

$$\mathbf{v}(t_{n+1}) = e^{-k A_2} \mathbf{v}(t_n) + k \int_0^1 e^{-k A_2(1-\tau)} G(\mathbf{v}(t_n + k\tau)) \, d\tau. \quad (4.3.7)$$

This formula is exact, and different ETD schemes stem from different approximations of the integral and the matrix exponential. We note that the following integral identities hold for a invertible, non-singular matrix A [4, Lemma 2.1.1]:

$$k \int_0^1 e^{-k A(1-\tau)} \, d\tau = A^{-1} (I - e^{-k A}), \quad (4.3.8)$$

$$k \int_0^1 e^{-k A(1-\tau)} \tau \, d\tau = k^{-1} A^{-2} (k A - I + e^{-k A}). \quad (4.3.9)$$

Two examples of simple ETD schemes are ETD1 and ETD2: They involve simply approximating G in (4.3.7) by a constant, i.e. $G(\mathbf{v}(t_n + k\tau)) \approx G(\mathbf{v}(t_n))$, or linearly, i.e. $G(\mathbf{v}(t_n + k\tau)) \approx G(\mathbf{v}(t_n)) + \tau(G(\mathbf{v}(t_{n+1})) - G(\mathbf{v}(t_n)))$. After applying (4.3.8) for the constant approximation and (4.3.9) for the linear approximation, this yields the ETD1 and ETD2 schemes

$$\mathbf{v}^{n+1} = e^{-k A_2} \mathbf{v}^n + A_2^{-1} (I - e^{-k A_2}) G(\mathbf{v}^n) \quad (4.3.10)$$

$$\mathbf{v}^{n+1} = e^{-kA_2} \mathbf{v}^n + A_2^{-1} (I - e^{-kA_2}) G(\mathbf{v}^n) + k^{-1} A_2^{-2} (kA_2 - I + e^{-kA_2}) (G(\mathbf{v}^{n+1}) - G(\mathbf{v}^n)). \quad (4.3.11)$$

The fully implicit ETD2 scheme (4.3.11) is then made explicit by using the ETD1 approximation (4.3.10) for \mathbf{v}^{n+1} , which we denote by \mathbf{v}_*^{n+1} , so that we obtain

$$\mathbf{v}^{n+1} = e^{-kA_2} \mathbf{v}^n + A_2^{-1} (I - e^{-kA_2}) G(\mathbf{v}^n) + k^{-1} A_2^{-2} (kA_2 - I + e^{-kA_2}) (G(\mathbf{v}_*^{n+1}) - G(\mathbf{v}^n)), \quad (4.3.12)$$

$$\mathbf{v}_*^{n+1} = e^{-kA_2} \mathbf{v}^n + A_2^{-1} (I - e^{-kA_2}) G(\mathbf{v}^n). \quad (4.3.13)$$

The next step is the approximation of the matrix exponentials. The second-order rational approximation with simple real distinct poles (RDP) originally proposed in [96] is given by

$$e^{-kA_2} \approx R_{\text{RDP}}(kA_2) := \left(I - \frac{5k}{12} A_2 \right) \left[\left(I - \frac{k}{3} A_2 \right) \left(I + \frac{k}{4} A_2 \right) \right]^{-1} \quad (4.3.14)$$

and is used in (4.3.12). In (4.3.13), the Padé-(0,1) first order approximation is given by

$$e^{-kA_2} \approx R_{01}(kA_2) := (I + kA_2)^{-1}. \quad (4.3.15)$$

The scheme with the inserted approximations is now given by

$$\mathbf{v}^{n+1} = R_{\text{RDP}}(kA_2) \mathbf{v}^n + A_2^{-1} (I - R_{\text{RDP}}(kA_2)) G(\mathbf{v}^n) \quad (4.3.16)$$

$$+ k^{-1} A_2^{-2} (kA_2 - I + R_{\text{RDP}}(kA_2)) (G(\mathbf{v}_*^{n+1}) - G(\mathbf{v}^n)), \quad (4.3.17)$$

$$\mathbf{v}_*^{n+1} = R_{01}(kA_2) \mathbf{v}^n + A_2^{-1} (I - R_{01}(kA_2)) G(\mathbf{v}^n). \quad (4.3.18)$$

After rearranging and rewriting terms (for the sake of brevity, we refer to [5, pp. 6]), this scheme called the ETD-RDP scheme can be rewritten as

$$\begin{aligned} \mathbf{v}^{n+1} &= \left(I + \frac{k}{3} A_2 \right)^{-1} [9\mathbf{v}^n + 2kG(\mathbf{v}^n) + kG(\mathbf{v}_*^{n+1})] \\ &\quad - \left(I + \frac{k}{4} A_2 \right)^{-1} \left[8\mathbf{v}^n + \frac{3k}{2} G(\mathbf{v}^n) + \frac{k}{2} G(\mathbf{v}_*^{n+1}) \right] \\ \mathbf{v}_*^{n+1} &= (I + kA_2)^{-1} (\mathbf{v}^n + kG(\mathbf{v}^n)). \end{aligned} \quad (4.3.19)$$

The last step are the substitutions $\mathbf{v}^n = e^{A_1 n k} \mathbf{u}^n$, $\mathbf{v}^{n+1} = e^{A_1 n k} e^{A_1 k} \mathbf{u}^{n+1}$, $G(\mathbf{v}^n) = e^{A_1 n k} F(\mathbf{u}^n)$, $G(\mathbf{v}^{n+1}) = e^{A_1 n k} e^{A_1 k} F(\mathbf{u}^n)$. Applying these on both sides and cancelling the matrix exponen-

tials yields

$$\begin{aligned}\mathbf{u}^{n+1} &= \left(I + \frac{k}{3}A_2\right)^{-1} [e^{-kA_1}\{9\mathbf{u}^n + 2kF(\mathbf{u}^n)\} + kF(\mathbf{u}_*^{n+1})] \\ &\quad - \left(I + \frac{k}{4}A_1\right)^{-1} \left[e^{-A_1k} \left\{ 8\mathbf{u}^n + \frac{3k}{2}F(\mathbf{u}^n) \right\} + \frac{k}{2}F(\mathbf{u}_*^{n+1}) \right] \\ \mathbf{u}_*^{n+1} &= (I + kA_2)^{-1} e^{-A_1k} (\mathbf{u}^n + kF(\mathbf{u}^n)).\end{aligned}$$

Making use of the identity (see [5, p. 6])

$$\left(I - \frac{5}{12}A_1\right) \left(I - \frac{k}{4}A_1\right)^{-1} \left(I - \frac{k}{4}A_1\right)^{-1} = 9 \left(I - \frac{k}{3}A_1\right)^{-1} - 8 \left(I - \frac{k}{4}A_1\right)^{-1},$$

and approximating e^{-A_1k} with the RDP approximation (4.3.14) in the first term and with the Padé approximation (4.3.15) in the second term, the ETD-RDP-IF scheme is given by

$$\begin{aligned}\mathbf{u}^{n+1} &= \left(I + \frac{k}{3}A_2\right)^{-1} [e^{-A_1k}\{9\mathbf{u}^n + 2kF(\mathbf{u}^n)\} + kF(\mathbf{u}_*^{n+1})] \\ &\quad - \left(I + \frac{k}{4}A_1\right)^{-1} \left[e^{-A_1k} \left\{ 8\mathbf{u}^n + \frac{3k}{2}F(\mathbf{u}^n) \right\} + \frac{k}{2}F(\mathbf{u}_*^{n+1}) \right] \\ \mathbf{u}_*^{n+1} &= (I + kA_2)^{-1} e^{-A_1k} (\mathbf{u}^n + kF(\mathbf{u}^n)).\end{aligned}\tag{4.3.20}$$

We conclude this section by repeating the ETD-RDP scheme (4.3.19) for one spatial dimension in Algorithm 4.2 and the ETD-RDP-IF scheme (4.3.20) for two spatial dimensions in Algorithm 4.3.

Algorithm 4.2: Deterministic finite differences ETD-RDP scheme

$$\mathbf{u}^{n+1} = \left(I_p + \frac{k}{3}DA_p\right)^{-1} [9\mathbf{u}^n + 2kF(\mathbf{u}^n) + kF(\mathbf{u}_*^{n+1})] \tag{4.3.21}$$

$$- \left(I_p + \frac{k}{4}DA_p\right)^{-1} \left[8\mathbf{u}^n + \frac{3k}{2}F(\mathbf{u}^n) + \frac{k}{2}F(\mathbf{u}_*^{n+1}) \right], \tag{4.3.22}$$

$$\mathbf{u}_*^{n+1} = (I_p + kDA_p)^{-1} (\mathbf{u}^n + kF(\mathbf{u}^n)), \quad n = 0, \dots, M-1, \tag{4.3.23}$$

where I_p denotes the $p \times p$ identity matrix.

Algorithm 4.3: Deterministic finite differences ETD-RDP-IF scheme

$$\begin{aligned}
\mathbf{u}^{n+1} = & \left(I_{p^2} + \frac{k}{3} DA_2 \right)^{-1} \left[\left\{ 9 \left(I_{p^2} + \frac{k}{3} DA_1 \right)^{-1} - 8 \left(I_{p^2} + \frac{k}{4} DA_1 \right)^{-1} \right\} \right. \\
& \cdot \{ 9\mathbf{u}^n + 2kF(\mathbf{u}^n) \} + kF(\mathbf{u}_*^{n+1}) \Big] \\
& - \left(I_{p^2} + \frac{k}{4} DA_2 \right)^{-1} \left[\left\{ 9 \left(I_{p^2} + \frac{k}{3} DA_1 \right)^{-1} - 8 \left(I_{p^2} + \frac{k}{4} DA_1 \right)^{-1} \right\} \right. \\
& \cdot \left\{ 8\mathbf{u}^n + \frac{3k}{2} F(\mathbf{u}^n) \right\} + \frac{k}{2} F(\mathbf{u}_*^{n+1}) \Big], \\
\mathbf{u}_*^{n+1} = & (I_{p^2} + kDA_{p^2})^{-1} (I_{p^2} + kDA_1)^{-1} (\mathbf{u}^n + kF(\mathbf{u}^n)), \quad n = 0, \dots, M-1,
\end{aligned}$$

where $A_1 := I_p \otimes A_p$, $A_2 := A_p \otimes I_p$.

4.3.3 ETDRK4

The ETDRK4 scheme has been shown in [62] to be a powerful method for solving reaction-diffusion equations which is superior to standard finite difference schemes. Before going into details on the spatial discretization, we give a few details on the ETDRK4 time stepping scheme.

Before we apply the time-stepping scheme to the initial value problem (4.3.1), we proceed as in [25] and first develop the scheme in the scalar case and consider the ordinary differential equation

$$\frac{du}{dt} = cu + F(u, t)$$

where c is a constant and F is a (potentially nonlinear) function. The starting point is, as in the previous section for the ETD-RDP scheme, the variation of constants formula (4.3.5) which we restate here for the $n+1$ -st time step for u :

$$u(t_{n+1}) = e^{ck} u(t_n) + \int_0^k e^{-(k-\tau)c} F(t_n + \tau) d\tau. \quad (4.3.24)$$

As in Section 4.3.2, the time stepping is derived from approximations applied to (4.3.24). The ETD1 scheme mentioned in the previous section with the approximation $F = F(u_n) + \mathcal{O}(k)$ is given by

$$u_{n+1} = e^{ck} u_n + c^{-1} F(u_n) (e^{ck} - 1). \quad (4.3.25)$$

In [25], a set of ETD schemes called ETDRK schemes based on Runge-Kutta time stepping are derived. They rely on the approximation of $F(u(t))$ at different interpolating points in time, for which the above schemes such as ETD1 are used. For example, u_{n+1} is approximated by the ETD1 expression (4.3.25)

$$a_n := u_n e^{ck} + c^{-1} F(u_n)(e^{ck} - 1) \quad (4.3.26)$$

and on the interval $t_n \leq t \leq t_{n+1}$, the linear approximation

$$F = F(u_n) + (t - t_n)(F(a_n) - F(u_n))/k + \mathcal{O}(h^2)$$

is applied and substituted into (4.3.24). After using (4.3.9) we obtain the ETDRK2 scheme given by

$$u_{n+1} = a_n + k^{-1} c^{-2} (F(a_n) - F_n)(e^{ck} - 1 - ck).$$

While a direct extension in the same way for a fourth order RK method yields only a third order scheme, Cox and Matthews show that after some modifications and verification using a computer algebra system, the fourth-order RK scheme (ETDRK4) is given by Algorithm 4.4. We now consider again the partial differential equation (4.3.1). In order to apply the ETDRK4 scheme in Algorithm 4.4, the equation first needs to be discretized in space. To this end, we use a spectral approach, making use of the discrete Fourier transform (DFT) which can be implemented using the well-known fast Fourier transform (FFT). Using the shorthand $u(x_n) =: \mathbf{u}_n$, $n = 0, \dots, p$, the DFT and inverse DFT formulae are given by

$$\hat{\mathbf{u}}_j = \sum_{n=0}^{p-1} e^{-2\pi i j x_n} \mathbf{u}_n, \quad j = -\frac{p}{2} + 1, \dots, \frac{p}{2}, \quad \mathbf{u}_n = \frac{1}{p} \sum_{j=-p/2+1}^{p/2} e^{2\pi i j x_n} \hat{\mathbf{u}}_j, \quad n = 0, \dots, p-1.$$

We will later also denote $\hat{\mathbf{u}} =: \mathcal{F}(\mathbf{u})$, especially when emphasizing that a transformation from physical space to Fourier space is performed, rather than a mere manipulation of terms in Fourier space. The derivatives on the spatial grid can now be approximated conveniently by noting that differentiation $\mathbf{u}' = (u'(x_n))_{n=0, \dots, p}$ in the physical space corresponds to a simple multiplication in Fourier space: $\widehat{\mathbf{u}'} = (i j \hat{u}_j)_{j=-\frac{p}{2}+1, \dots, \frac{p}{2}}$. In particular, the Laplace operator in Fourier space ends up being diagonal: $\widehat{\Delta \mathbf{u}} = L \cdot \hat{\mathbf{u}}$ for a diagonal matrix L with $L_{jj} = -j^2$, $j = -\frac{p}{2} + 1, \dots, \frac{p}{2}$. For the ETDRK4 scheme, the DFT is applied to the discretized function \mathbf{u} , the time stepping using the fourth-order Runge-Kutta method is carried out in Fourier space and after the last time step the solution is transformed back to physical space. One subtlety

to note is that in each time step, for each evaluation using the nonlinear function F , the argument $\hat{\mathbf{u}}^n$ needs to be transformed to physical space, then F is evaluated, and $F(\mathcal{F}^{-1}(\hat{\mathbf{u}}^n))$ is transformed back into Fourier space (see the scheme below). Taking this into account, we introduce the shorthand $\hat{F}(\hat{\mathbf{u}}^n) := \mathcal{F}(F(\mathcal{F}^{-1}(\hat{\mathbf{u}}^n)))$. The ETDRK4 scheme Algorithm 4.4 then becomes the discretized spectral ETDRK4 scheme and it is given by Algorithm 4.5. A problem with this scheme is that it can lead to numerical instabilities resulting from cancellation errors in the expressions of a_n , b_n and c_n (see Algorithm 4.5, [63, p. 6]). Instead of evaluating the critical expressions directly, Cauchy's integral formula

$$f(L) = \frac{1}{2\pi i} \int_{\Gamma} f(t)(tI_p - L)^{-1} dt \quad (4.3.31)$$

is used, where Γ is a contour which encloses the eigenvalues of L . The trapezoidal rule is suitable to evaluate this integral, since it converges exponentially for complex contour integrals [29]. In our case, L is diagonal and the contours for (4.3.31) to evaluate a_n , b_n and c_n may simply be chosen elementwise, so that for each diagonal element L_{ii} of L , $i = 1, \dots, p$, we pick one circle around L_{ii} . The approximated integral (4.3.31) then simplifies to

$$\frac{1}{R} \sum_{i=1}^R f(L_{ii} + r_i), \quad (4.3.32)$$

where r_1, \dots, r_R are the complex roots of unity of order R sitting on the unit circle shifted by L_{ii} .

Furthermore, in order to avoid errors caused by aliasing in Fourier space, anti-aliasing is needed in the program code. We refer to [62, p. 11] on how to do this properly.

Algorithm 4.4: Deterministic ETDRK4 time stepping scheme

$$\begin{aligned}
u_{n+1} &= u_n e^{ck} + k^{-2} c^{-3} \{ F(u_n) [-4 - kc + e^{ck} (4 - 3ck + c^2 k^2)] \\
&\quad + 2(F(a_n) + F(b_n)) [2 + ck + e^{ck} (-2 + ck)] \\
&\quad + F(c_n) [-4 - 3ck - h^2 c^2 + e^{-ck} (4 - ck)] \}, \\
a_n &= e^{-ck/2} u_n + c^{-1} (e^{-ck/2} - 1), \\
b_n &= e^{-ck/2} u_n + c^{-1} (e^{-ck/2} - 1) F(a_n) \\
c_n &= e^{-ck/2} a_n + c^{-1} (e^{-ck/2} - 1) (2F(b_n) - F(u_n)).
\end{aligned}$$

Algorithm 4.5: Deterministic discretized Spectral ETDRK4 scheme

$$\widehat{\mathbf{u}}^0 = \mathcal{F}(\mathbf{u}^0), \quad (4.3.27)$$

$$\begin{aligned}
\widehat{\mathbf{u}}^{n+1} &= e^{Lh} \widehat{\mathbf{u}}^n + h^{-2} L^{-3} \{ [-4 \cdot I_p - Lh + e^{Lh} (4 \cdot I_p - 3Lh + (Lh)^2)] \hat{F}(\widehat{\mathbf{u}}^n) \\
&\quad + 2[2 \cdot I_p + Lh + e^{Lh} (-2 \cdot I_p + Lh)] (\hat{F}(a_n) + \hat{F}(b_n)) \\
&\quad + [-4 - 3Lh - (Lh)^2 + e^{Lh} (4 \cdot I_p - Lh)] \hat{F}(c_n) \},
\end{aligned}$$

$$a_n = e^{Lh/2} \widehat{\mathbf{u}}^n + L^{-1} (e^{Lh/2} - I_p) \hat{F}(\widehat{\mathbf{u}}^n), \quad (4.3.28)$$

$$b_n = e^{Lh/2} \widehat{\mathbf{u}}^n + L^{-1} (e^{Lh/2} - I_p) \hat{F}(a_n), \quad (4.3.29)$$

$$c_n = e^{Lh/2} a_n + L^{-1} (e^{Lh/2} - I_p) (2\hat{F}(b_n) - \hat{F}(\widehat{\mathbf{u}}^n)), \quad n = 0, \dots, M-1, \quad (4.3.30)$$

$$\mathbf{u}^M = \mathcal{F}^{-1}(\widehat{\mathbf{u}}^M).$$

Chapter 5

Solving random differential equations using polynomial chaos with exponential time differencing

Throughout this work, we will be using p points in each spatial dimension, M time steps, a time step size $k = T/M$, a spatial step size $h = p^{-1}$ and $N + 1$ PCE coefficients (with the index ranging from 0 to N). For the random equations (4.0.4), we will assume that the random input is uniformly distributed, so we will always use Legendre polynomials in the gPCEs. The cases for other random distributions and the corresponding family of orthogonal polynomials according to Table 4.1.1 work in an analogous fashion.

5.1 Intrusive PCE schemes

We have already given an introduction to intrusive PCE in Section 4.1. The aim is now to develop the intrusive PCE numerical schemes needed to solve (4.2.7). To demonstrate how iPCE schemes are developed, we will consider in this section the nonlinear function $F(\omega, u(x, t, \omega)) = K(\omega)u(x, t, \omega)^3$ with a uniformly distributed constant $K(\omega) = \xi(\omega) \sim \mathcal{U}[a, b]$. We stress here that we will henceforth use K and ξ interchangeably, and the orthogonal polynomials will be the Legendre polynomials supported on $[a, b] \subset \mathbb{R}$. Among others, this nonlinear function F will also be treated in Section 5.2. We describe the explicit Euler scheme in 4.3.1, the ETD-RDP scheme in Section 5.1.2 and the ETDRK4 scheme in Section 5.1.3.

For both schemes, it is a necessary prerequisite to discuss the gPCE for the term u^3 given the

PCE of u . Given the gPCE (4.1.12), we have

$$\begin{aligned} u^3(x, t, \omega) &= \left(\sum_{i=0}^{\infty} u_i(x, t) P_i(\xi) \right) \cdot \left(\sum_{j=0}^{\infty} u_j(x, t) P_j(\xi) \right) \cdot \left(\sum_{k=0}^{\infty} u_k(x, t) P_k(\xi) \right) \\ &= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} u_i(x, t) u_j(x, t) u_k(x, t) P_i(\xi) P_j(\xi) P_k(\xi). \end{aligned}$$

After a Galerkin projection (multiplying with P_η for $\eta \in \mathbb{N}_0$ and taking $\mathbb{E}[\cdot]$) and making an approximation by truncating the series to $N + 1$ terms each, plugging the gPCE into (4.2.7) we obtain for $\eta = 0, \dots, N$ (see also [30, pp. 4])

$$\begin{cases} \frac{\partial}{\partial t} u_\eta(x, t) = D\Delta u_\eta(x, t) + \mathbb{E} \left[K \sum_{i,j,k=0}^N u_i(x, t) u_j(x, t) u_k(x, t) P_i(\xi) P_j(\xi) P_k(\xi) P_\eta(\xi) \right] \\ u_0(x, 0) = u_{\text{init}}(x), \quad u_\eta(x, 0) = 0, \quad \eta > 0. \end{cases}$$

Extracting the deterministic coefficient functions, we have

$$\begin{cases} \frac{\partial}{\partial t} u_\eta(x, t) = D\Delta u_\eta(x, t) + \sum_{i,j,k=0}^N u_i(x, t) u_j(x, t) u_k(x, t) \mathbb{E} [K P_i(\xi) P_j(\xi) P_k(\xi) P_\eta(\xi)] \\ u_0(x, 0) = u_{\text{init}}(x), \quad u_\eta(x, 0) = 0, \quad \eta > 0. \end{cases}$$

and, using from now on the notation $\mathbf{K}_{ijk\eta} := \mathbb{E}[K P_i(\xi) P_j(\xi) P_k(\xi) P_\eta(\xi)]$, it is

$$\begin{cases} \frac{\partial}{\partial t} u_\eta(x, t) = D\Delta u_\eta(x, t) + \sum_{i,j,k=0}^N \mathbf{K}_{ijk\eta} u_i(x, t) u_j(x, t) u_k(x, t) \\ u_0(x, 0) = u_{\text{init}}(x), \quad u_\eta(x, 0) = 0, \quad \eta > 0. \end{cases} \quad (5.1.1)$$

The tensor \mathbf{K} will be computed as a pre-processing step and used throughout a simulation.

Remark 5.1.1 *In our implementations, we use an equivalent way of computing the gPCE of u^3 : We make use of an addition theorem for Legendre polynomials [1] stating that for $\alpha, \beta \in \mathbb{N}$ and $\alpha \wedge \beta = \min(\alpha, \beta)$,*

$$P_\alpha(x) P_\beta(x) = \sum_{i \leq \alpha \wedge \beta} C(\alpha, \beta, p) P_{\alpha+\beta-2p}(x), \quad \text{where} \quad (5.1.2)$$

$$C_{\alpha, \beta, p} = \frac{A_p A_{\alpha-p} A_{\beta-p}}{A_{\alpha+\beta-p}} \cdot \frac{2\alpha + 2\beta - 4p + 1}{2\alpha + 2\beta - 2p + 1} \cdot \sqrt{\frac{(2\alpha + 1)(2\beta + 1)}{2(\alpha + \beta - 2p + 1)}}, \quad (5.1.3)$$

$$A_r := \frac{(1/2)_r}{r!}, \quad (a)_r := a \cdot (a+1) \cdots (a+r-1), \quad (a)_0 = 1. \quad (5.1.4)$$

Using (5.1.2) to expand the gPCE of u^3 yields, after applying a Cauchy product twice (and dropping x , t and ω),

$$uv^2 = \left[\sum_{l=0}^{\infty} u_l P_l \right] \cdot \left[\sum_{i=0}^{\infty} v_i P_i \right]^2 \quad (5.1.5)$$

$$= \left[\sum_{l=0}^{\infty} u_l P_l \right] \cdot \left[\sum_{i=0}^{\infty} \sum_{j=0}^i v_j v_{i-j} P_j P_{i-j} \right] \quad (5.1.6)$$

$$= \left[\sum_{l=0}^{\infty} u_l P_l \right] \cdot \left[\sum_{i=0}^{\infty} \sum_{j=0}^i v_j v_{i-j} \sum_{p=0}^{j \wedge (i-j)} C_{j,i-j,p} P_{i-2p} \right] \quad (5.1.7)$$

$$=: \left[\sum_{l=0}^{\infty} u_l P_l \right] \cdot \left[\sum_{i=0}^{\infty} a_i \right] \quad (5.1.8)$$

$$= \sum_{l=0}^{\infty} \sum_{m=0}^l a_m u_{l-m} P_{l-m} \quad (5.1.9)$$

$$= \sum_{l=0}^{\infty} \sum_{m=0}^l u_{l-m} P_{l-m} \sum_{j=0}^m v_j v_{m-j} \sum_{p=0}^{j \wedge (m-j)} C_{j,m-j,p} P_{m-2p} \quad (5.1.10)$$

$$= \sum_{l=0}^{\infty} \sum_{m=0}^l \sum_{j=0}^m \sum_{p=0}^{j \wedge (m-j)} u_{l-m} v_j v_{m-j} C_{j,m-j,p} P_{l-m} P_{m-2p} \quad (5.1.11)$$

$$= \sum_{l=0}^{\infty} \sum_{m=0}^l \sum_{j=0}^m \sum_{p=0}^{j \wedge (m-j)} u_{l-m} v_j v_{m-j} C_{j,m-j,p} \sum_{n=0}^{(l-m) \wedge (m-2p)} C_{l-m,m-2p,n} P_{l-2p-2n} \quad (5.1.12)$$

$$= \sum_{l=0}^{\infty} \sum_{m=0}^l \sum_{j=0}^m \sum_{p=0}^{j \wedge (m-j)} \sum_{n=0}^{(l-m) \wedge (m-2p)} u_{l-m} v_j v_{m-j} C_{j,m-j,p} C_{l-m,m-2p,n} P_{l-2p-2n}. \quad (5.1.13)$$

which after truncating to $N+1$ terms and performing a Galerkin step with P_η yields

$$\sum_{\ell=0}^N \sum_{m=0}^{\ell} \sum_{j=0}^m \sum_{p=0}^{j \wedge (m-j)} \sum_{n=0}^{(\ell-m) \wedge (m-2p)} u_{\ell-m} v_j v_{m-j} C_{j,m-j,p} C_{\ell-m,m-2p,n} \mathbb{1}_{\{\eta=\ell-2p-2n\}}. \quad (5.1.14)$$

This amounts to a number \tilde{N} of summands shown in Table 5.1.1 for low N , of which $\tilde{N}/(N+1)$ are nonzero. For equations with a u^3 term, this has a severe impact on the runtime of iPCE for large N (see Remark 5.1.2).

Remark 5.1.2 If the random input ξ is multivariate with dimension $P \in \mathbb{N}$, the underlying base of orthogonal polynomial has, for up to degree N , $\hat{N} + 1 := \frac{(N+P)!}{N!P!}$ elements [102, Eq. (5.3)] (where \hat{N} denotes the number of expansion terms corresponding to non-constant polynomials). This means that the system (5.1.1) is $\hat{N} + 1$ -times bigger than the deterministic

N	0	1	2	3	4	5	6	7	8
\tilde{N}	1	8	39	124	335	762	1589	3016	5418

Table 5.1.1: Number \tilde{N} of summands in (5.1.14) for low N

system. We pointed out in Remark 5.1.1 that especially equations involving a u^n term for $n \geq 3$ can start to pose problems due to the rapidly increasing number of necessary summands in the gPCE. Figure 5.1.1 shows how the runtimes increase for iPCE depending on N for the equations (5.2.1) investigated in Section 5.2. Especially the equation involving a u^3 term shows a rapid increase in computational effort as N grows.

On the other hand, suppose that a one-dimensional quadrature formula (such as Gauss-Legendre) has an error bound for $f \in \mathcal{C}^r([-1, 1])$ for $q \in \mathbb{N}$ quadrature nodes

$$|E_q^1 f| = \mathcal{O}(q^{-r}).$$

Then, it is known that in d dimensions, the full grid quadrature error $|\tilde{E}_q^d f|$ and the error for Smolyak sparse quadrature $|E_q^d f|$ for $f \in H^r([-1, 1]^d)$ is given by

$$|\tilde{E}_q^d f| = \mathcal{O}(q^{-r/d}), \quad |E_q^d f| = \mathcal{O}(\log(q)^{(d-1)(r+1)} q^{-r})$$

(see e.g. [61, p. 39]). This means that in higher dimensions, provided that the integrand is sufficiently smooth, the number of necessary function evaluations can be drastically reduced by using a sparse grid, and the curse of dimensionality is much less severe than in the iPCE case as outlined above.

We will henceforth use the following notation: For $\eta = 0, \dots, N$ and $n = 0, \dots, M$, it is

$$\mathbf{u}_\eta^n := (u_\eta(x_0, t_n), u_\eta(x_1, t_n), \dots, u_\eta(x_p, t_n))^\top.$$

Also, $\Delta \mathbf{u}_\eta^n = (\Delta u_\eta(x_i, t_n))_{i=0, \dots, p}^\top$ refers to Δu_η evaluated at the grid points. We also denote by

$$\mathbf{U}^n := \begin{pmatrix} \mathbf{u}_0^n \\ \mathbf{u}_1^n \\ \vdots \\ \mathbf{u}_N^n \end{pmatrix} \in \mathbb{R}^{(N+1) \cdot p^d} \quad (5.1.15)$$

the vector of all stacked discretized PCE coefficient functions $\mathbf{u}_0^n, \dots, \mathbf{u}_N^n$ at time t_n .

5.1.1 Explicit Euler

For a single function u_η in the PCE, the explicit Euler scheme for solving (4.0.4) is given by

$$\mathbf{u}_\eta^{n+1} = \mathbf{u}_\eta^n + k \cdot \left[DA_p \mathbf{u}_\eta^n + \sum_{i,j,k=0}^N \mathbf{K}_{ijk\eta} \mathbf{u}_i^n \odot \mathbf{u}_j^n \odot \mathbf{u}_k^n \right], \quad n = 0, \dots, M-1,$$

where by \odot we denote a componentwise product. Combining all coefficient functions $\mathbf{u}_0, \dots, \mathbf{u}_N$, we obtain Algorithm 5.1.

Algorithm 5.1: iPCE Explicit Euler scheme

$$\mathbf{U}^{n+1} = \mathbf{U}^n + k \cdot [(I_{N+1} \otimes DA_p) \mathbf{U}^n + \mathbf{F}(\mathbf{U}^n)], \quad (5.1.16)$$

$$\mathbf{F}(\mathbf{U}^n) = \left[\sum_{i,j,k=0}^N \mathbf{K}_{ijk\eta} \mathbf{u}_i^n \odot \mathbf{u}_j^n \odot \mathbf{u}_k^n \right]_{\eta=0,\dots,N}, \quad n = 0, \dots, M-1. \quad (5.1.17)$$

5.1.2 ETD-RDP and ETD-RDP-IF

Applying the scheme (4.3.21) to the PCE system (4.2.7) with a finite difference discretization and making use of the fact that $(I_{N+1} \otimes (I_p + \frac{k}{3} DA_p))^{-1} = I_{N+1} \otimes (I_p + \frac{k}{3} DA_p)^{-1}$, the resulting scheme is

Algorithm 5.2: iPCE ETD-RDP scheme

$$\mathbf{U}^{n+1} = \left(I_{N+1} \otimes (I_p + \frac{k}{3} DA_p)^{-1} \right) [9\mathbf{U}^n + 2k\mathbf{F}(\mathbf{U}^n) + k\mathbf{F}(\mathbf{U}_*^{n+1})] \quad (5.1.18)$$

$$- \left(I_{N+1} \otimes (I_p + \frac{k}{4} DA_p)^{-1} \right) [8\mathbf{U}^n + \frac{3k}{2}\mathbf{F}(\mathbf{U}^n) + \frac{k}{2}\mathbf{F}(\mathbf{U}_*^{n+1})],$$

$$\mathbf{U}_*^{n+1} = (I_{N+1} \otimes (I_p + k DA_p)^{-1}) [\mathbf{U}^n + k\mathbf{F}(\mathbf{U}^n)], \quad n = 0, \dots, M-1 \quad (5.1.19)$$

where $\mathbf{F}(\mathbf{U}^n)$ is given by the column vector (5.1.17).

We now discuss the application of ETD-RDP-IF in the case where the spatial dimension is greater than one. For simplicity of the presentation, we restrict ourselves here to the case of two spatial dimensions. In this case, the finite difference discretization of the Laplacian with periodic boundary conditions is given by $A = A_1 + A_2 := (D \cdot I_p) \otimes A_p + A_p \otimes (D \cdot I_p)$. We

use the notation $\mathbf{U}^n = (\mathbf{u}_0, \dots, \mathbf{u}_N)^\top$ from above and denote

$$\mathbf{A} := I_{N+1} \otimes A, \quad \mathbf{A}_1 := I_{N+1} \otimes A_1, \quad \mathbf{A}_2 := I_{N+1} \otimes A_2, \quad \mathbf{I} := I_{N+1} \otimes I_{p^2} = I_{p^2 \cdot (N+1)},$$

so in particular it is $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$. In order to solve the discretized PCE system

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A} \mathbf{U} = \mathbf{F}(\mathbf{U}), \quad \mathbf{u}_0^0 = \mathbf{u}_{\text{init}}, \quad (5.1.20)$$

with \mathbf{F} as above, we apply a dimensional splitting technique. We introduce the new time-dependent function $\mathbf{V} = e^{\mathbf{A}_1 t} \mathbf{U}$. The term $I_{N+1} \otimes e^{\mathbf{A}_1 t}$ is called the integrating factor for PCE. We will also use that \mathbf{A} and \mathbf{A}_1 commute (see [5, Lemma 1]), and therefore also \mathbf{A} and $e^{\mathbf{A}_1 t}$ commute. For the dimensional splitting, we define $\mathbf{V}(t) := e^{\mathbf{A}_1 t} \mathbf{U}(t)$. Carrying out the steps as in [5, pp. 3] for the PCE case, the time derivative of \mathbf{V} is

$$\frac{\partial \mathbf{V}}{\partial t} = e^{\mathbf{A}_1 t} \frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_1 e^{\mathbf{A}_1 t} \mathbf{U}$$

and inserting (5.1.20), we obtain

$$\begin{aligned} \frac{\partial \mathbf{V}}{\partial t} &= e^{\mathbf{A}_1 t} (\mathbf{F}(\mathbf{U}) - \mathbf{A} \mathbf{U}) + \mathbf{A}_1 e^{\mathbf{A}_1 t} \mathbf{U} = e^{\mathbf{A}_1 t} \mathbf{F}(\mathbf{U}) - e^{\mathbf{A}_1 t} \mathbf{A} \mathbf{U} + \mathbf{A}_1 e^{\mathbf{A}_1 t} \mathbf{U} \\ &= e^{\mathbf{A}_1 t} \mathbf{F}(\mathbf{U}) - \mathbf{A} e^{\mathbf{A}_1 t} \mathbf{U} + \mathbf{A}_1 e^{\mathbf{A}_1 t} \mathbf{U} = e^{\mathbf{A}_1 t} \mathbf{F}(\mathbf{U}) - \mathbf{A}_2 e^{\mathbf{A}_1 t} \mathbf{U} = e^{\mathbf{A}_1 t} \mathbf{F}(e^{-\mathbf{A}_1 t} \mathbf{V}) - \mathbf{A}_2 \mathbf{V}, \end{aligned}$$

so the system

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{A}_2 \mathbf{V} = \mathbf{G}(\mathbf{V}), \quad \mathbf{v}_0^0 = \mathbf{u}_{\text{init}}$$

with $\mathbf{G}(\mathbf{V}) := e^{\mathbf{A}_1 t} \mathbf{F}(e^{-\mathbf{A}_1 t} \mathbf{V})$ must be solved. The next step is to apply the ETD-RDP scheme given in (5.1.19) to \mathbf{V} . The derivation in the PCE case works analogously as described in [5, p. 5–7]. In addition to the ETD-RDP scheme (5.1.19), the unwinding of the integrating factor is needed. With the substitutions $\mathbf{V}^n = e^{\mathbf{A}_2 n k} \mathbf{U}^n$, $\mathbf{V}^{n+1} = e^{\mathbf{A}_2 n k} e^{\mathbf{A}_2 k} \mathbf{U}^{n+1}$, $\mathbf{G}(\mathbf{V}^n) = e^{\mathbf{A}_2 n k} \mathbf{F}(\mathbf{U}^n)$ and $\mathbf{G}(\mathbf{V}^{n+1}) = e^{\mathbf{A}_2 n k} e^{\mathbf{A}_2 k} \mathbf{F}(\mathbf{U}^{n+1})$, (5.1.19) becomes the full ETD-RDP-IF scheme for iPCE shown in Algorithm 5.3. Note that this scheme looks identical to the one presented in [5, Eq. (19)], but we have a different notation for \mathbf{A}_1 , \mathbf{A}_2 and the function \mathbf{F} .

Algorithm 5.3: iPCE ETD-RDP-IF scheme

$$\begin{aligned}
 \mathbf{U}^{n+1} &= \left(\mathbf{I} + \frac{k}{3} D\mathbf{A}_2 \right)^{-1} \left[\left\{ 9 \left(\mathbf{I} + \frac{k}{3} D\mathbf{A}_1 \right)^{-1} - 8 \left(\mathbf{I} + \frac{k}{4} D\mathbf{A}_1 \right)^{-1} \right\} \right. \\
 &\quad \cdot \{ 9\mathbf{U}^n + 2k\mathbf{F}(\mathbf{U}^n) \} + k\mathbf{F}(\mathbf{U}_*^{n+1}) \Big] \\
 &\quad - \left(\mathbf{I} + \frac{k}{4} D\mathbf{A}_2 \right)^{-1} \left[\left\{ 9 \left(\mathbf{I} + \frac{k}{3} D\mathbf{A}_1 \right)^{-1} - 8 \left(\mathbf{I} + \frac{k}{4} D\mathbf{A}_1 \right)^{-1} \right\} \right. \\
 &\quad \cdot \{ 8\mathbf{U}^n + \frac{3k}{2} \mathbf{F}(\mathbf{U}^n) \} + \frac{k}{2} \mathbf{F}(\mathbf{U}_*^{n+1}) \Big], \\
 \mathbf{U}_*^{n+1} &= (\mathbf{I} + kD\mathbf{A}_2)^{-1} (\mathbf{I} + kD\mathbf{A}_1)^{-1} (\mathbf{U}^n + k\mathbf{F}(\mathbf{U}^n)), \quad n = 0, \dots, M-1.
 \end{aligned}$$

5.1.3 ETDRK4

For $\eta = 0, \dots, N$, we denote $\widehat{\mathbf{u}}_\eta = \mathcal{F}(\mathbf{u}_\eta)$ and for the stacked discretized PCE functions, we write

$$\mathbf{U}^n = \begin{pmatrix} \mathbf{u}_0^n \\ \vdots \\ \mathbf{u}_N^n \end{pmatrix}, \quad \widehat{\mathbf{U}}^n = \mathcal{F}_{\text{comp}}(\mathbf{U}^n) := \begin{pmatrix} \mathcal{F}(\mathbf{u}_0^n) \\ \vdots \\ \mathcal{F}(\mathbf{u}_N^n) \end{pmatrix}, \quad \mathcal{F}_{\text{comp}}^{-1}(\widehat{\mathbf{U}}^n) := \begin{pmatrix} \mathcal{F}^{-1}(\widehat{\mathbf{u}}_0^n) \\ \vdots \\ \mathcal{F}^{-1}(\widehat{\mathbf{u}}_N^n) \end{pmatrix}$$

and we stress the fact that the discrete Fourier transform is taken separately for each base function. For the vector \mathbf{U}^n of stacked discretized PCE base functions, $\mathbf{F}(\mathbf{U}^n)$ is understood in the sense of (5.1.17). We also use the matrix L from Section 4.3.3 and denote $\mathbf{L} := I_{N+1} \otimes L$. This matrix is still diagonal and therefore very easy to invert. As before in (4.3.27), we introduce the shorthand $\hat{\mathbf{F}}(\hat{\mathbf{U}}) := \mathcal{F}_{\text{comp}}(\mathbf{F}(\mathcal{F}_{\text{comp}}^{-1}(\hat{\mathbf{U}})))$. The ETDRK4 scheme for the intrusive PCE system is then given by Algorithm 5.4. In Section 4.3.3, we recalled a contour integral method in order to avoid cancellation errors arising in the evaluation of the expressions a_n , b_n and c_n in equations (4.3.28) to (4.3.30). In this scheme, we apply this technique to each base function, i.e. the mean (4.3.32) is computed separately for each PCE base function. Likewise, anti-aliasing is applied to each base function.

Remark 5.1.3 *In order to deal with random PDEs of the form*

$$\frac{\partial u(x, t, \omega)}{\partial t} = D(\omega) \Delta u(x, t, \omega) + F(u(x, t, \omega)),$$

Algorithm 5.4: iPCE ETDRK4 scheme

$$\begin{aligned}
 \widehat{\mathbf{U}}^0 &= \mathcal{F}_{\text{comp}}(\mathbf{U}^0), \\
 \widehat{\mathbf{U}}^{k+1} &= \exp\left(\frac{h}{2}\mathbf{L}\right)\widehat{\mathbf{U}}^k \\
 &\quad + (h^{-2}\mathbf{L}^{-3}) \left\{ \left[-4\mathbf{I} - h\mathbf{L} + e^{h\mathbf{L}}(4\mathbf{I} - 3h\mathbf{L} + h^2\mathbf{L}^2) \right] \widehat{\mathbf{F}}(\widehat{\mathbf{U}}^k) \right. \\
 &\quad + 2[2\mathbf{I} + h\mathbf{L} + e^{h\mathbf{L}}(-2\mathbf{I} + h\mathbf{L})] \left(\widehat{\mathbf{F}}(\widehat{\mathbf{a}}^k) + \widehat{\mathbf{F}}(\widehat{\mathbf{b}}^k) \right) \\
 &\quad \left. + \left[-4\mathbf{I} - 3h\mathbf{L} - h^2\mathbf{L}^2 + e^{h\mathbf{L}}(4\mathbf{I} - h\mathbf{L}) \right] \widehat{\mathbf{F}}(\widehat{\mathbf{c}}^k) \right\}, \\
 \mathbf{U}^M &= \mathcal{F}_{\text{comp}}^{-1}(\widehat{\mathbf{U}}^M),
 \end{aligned}$$

where $\mathbf{I} := I_{(N+1)p^2}$ and

$$\begin{aligned}
 \widehat{\mathbf{a}}^k &= e^{\mathbf{L}h/2}\widehat{\mathbf{U}}^k + \mathbf{L}^{-1}(e^{\mathbf{L}h/2} - \mathbf{I})\widehat{\mathbf{F}}(\widehat{\mathbf{U}}^k), \\
 \widehat{\mathbf{b}}^k &= e^{\mathbf{L}h/2}\widehat{\mathbf{U}}^k + \mathbf{L}^{-1}(e^{\mathbf{L}h/2} - \mathbf{I})\widehat{\mathbf{F}}(\widehat{\mathbf{a}}^k), \\
 \widehat{\mathbf{c}}^k &= e^{\mathbf{L}h/2}\widehat{\mathbf{a}}^k + \mathbf{L}^{-1}(e^{\mathbf{L}h/2} - \mathbf{I})(2\widehat{\mathbf{F}}(\widehat{\mathbf{b}}^k) - \widehat{\mathbf{F}}(\widehat{\mathbf{U}}^k)).
 \end{aligned}$$

Runtimes for iPCE for different N compared to $N = 0$

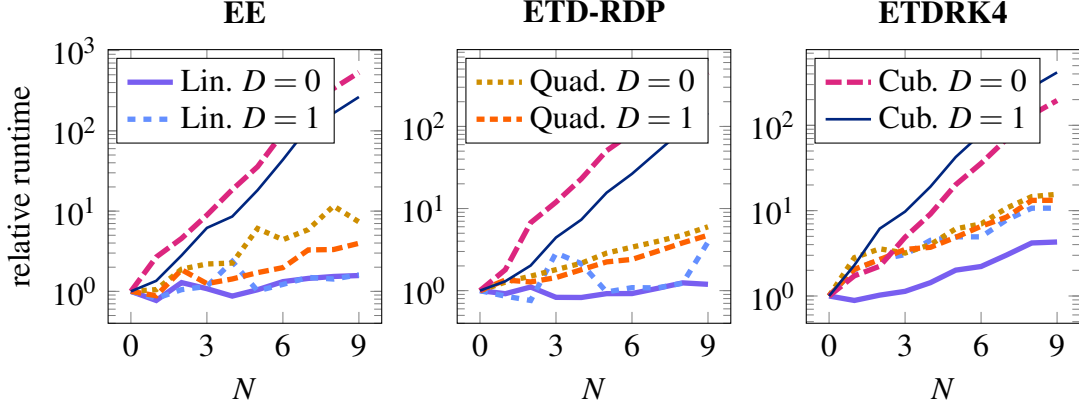


Figure 5.1.1: Relative runtimes R_N for $N = 0, \dots, 9$, where $R_0 := 1$. Each plot shows six curves for six different equations investigated in Section 5.2: Equation (5.2.1) with linear, quadratic or cubic F , with $D = 0$ or $D = 1$. Each data point is an average over the runtimes of ten identical iPCE simulations with $T = 0.1$ and $M = 10$ time steps.

i.e. equations where D is random and the function F is not random, the ETDRK4 intrusive PCE scheme described above is not very suitable, because in this case the discretized Laplacian $D\Delta$ is not a tridiagonal matrix anymore (the shape of the matrix depends on the distribution of the random variable D), and the Fourier transform L of the Laplacian is not diagonal anymore, which can severely impact the performance of the scheme.

5.2 Numerical experiments.

We now apply non-intrusive PCE as explained in Section 4.2.1 and intrusive PCE as explained in Section 5.1 to the equation (4.0.4) which we repeat here:

$$\begin{cases} \frac{\partial u(x,t,\omega)}{\partial t} &= D\Delta u(x,t,\omega) + F(\omega, u(x,t,\omega)), \\ u(x,0,\omega) &= u_{\text{init}}(x). \end{cases} \quad (5.2.1)$$

For a random constant $K \sim \mathcal{U}[1, 2]$, we will use $F(\omega, u(x,t,\omega)) = K(\omega)u(x,t,\omega)$ in Section 5.2.1, $F(\omega, u(x,t,\omega)) = K(\omega)u(x,t,\omega)^2$ in Section 5.2.2 and furthermore $F(\omega, u(x,t,\omega)) = K(\omega)u(x,t,\omega)^3$ in Section 5.2.3. In all three cases, we test the two cases $D = 1$ and $D = 0$, i.e. a random ODE and a PDE with a diffusion term. In all three cases, we pick $u_{\text{init}}(x) = \cos(\pi x)$. Finally, we will discuss a more complex problem, the Gray-Scott system, in Section 5.2.6. Throughout all simulations, we work on the spatial domain $(-1, 1)^d$, and spatial resolution

$p = 128$.

In all error plots, we show the relative L^2 error which, for the exact mean or a reference solution mean $\mathbb{E}[u]$ and the approximated solution $\mathbb{E}[u]_{\text{approx}}$, is given by $e_{\text{rel}}(t) = \|\mathbb{E}[u](\cdot, t) - \mathbb{E}[u]_{\text{approx}}(\cdot, t)\|_{L^2} / \|\mathbb{E}[u](\cdot, t)\|_{L^2}$. For the random equation with linear term in Section 5.2.1, we know the exact solution, for the other equations in Sections 5.2.2, 5.2.3 and 5.2.6 we use a reference solution obtained from using non-intrusive PCE with Gauß quadrature with a high number of quadrature points (see Table 5.2.1). We also note that for the error of niPCE with naive, randomly sampled MC, we ran ten simulations producing ten error graphs over which we took the mean.

5.2.1 One-dimensional random equation with linear term

The equation

$$\frac{\partial u(x, t, \omega)}{\partial t} = D\Delta u(x, t, \omega) - K(\omega)u(x, t, \omega), \quad u(x, 0, \omega) = \cos(\pi x), \quad (5.2.2)$$

$x \in (-1, 1)$, $t \in (0, T]$, has the exact solution $u(x, t, \omega) = \exp(-(K(\omega) + D\pi^2)t) \cos(\pi x)$. If $K \sim \mathcal{U}[a, b]$, the expected value is given by

$$\begin{aligned} \mathbb{E}[u(x, t, \cdot)] &= \frac{1}{b-a} \int_a^b \exp(-(\xi + D\pi^2)t) \cos(\pi x) \, d\xi \\ &= \frac{1}{b-a} \frac{\cos(\pi x)}{t} \exp(-(D\pi^2 + a)t) - \exp(-(D\pi^2 + b)t), \end{aligned} \quad (5.2.3)$$

and in our tests we choose again $a = 1$ and $b = 2$. The variance is

$$\begin{aligned} \text{Var}[u(x, t, \cdot)] &= \mathbb{E}[u(x, t, \cdot)^2] - \mathbb{E}[u(x, t, \cdot)]^2 \\ &= \frac{1}{b-a} \int_a^b \exp(-2(\xi + D\pi^2)t) \cos^2(\pi x) \, d\xi - \mathbb{E}[u(x, t, \cdot)]^2 \\ &= \frac{1}{b-a} \frac{\cos^2(\pi x)}{2t} [\exp(-2(D\pi^2 + a)t) - \exp(-2(D\pi^2 + b)t)] \\ &\quad - \left[\frac{1}{b-a} \frac{\cos(\pi x)}{t} \exp(-(D\pi^2 + a)t) - \exp(-(D\pi^2 + b)t) \right]^2. \end{aligned} \quad (5.2.4)$$

In Figure 5.2.1, we show the time-dependent error plots for solving (5.2.2) with $D = 0$, comparing to the exact solution. For iPCE, it can be seen that both for EE and for ETD-RDP, increasing the number of non-constant polynomials beyond $N = 1$ makes no difference. Due to the greater accuracy of ETD-RK4, the error keeps decreasing until $N = 4$, and the spectral convergence of iPCE is clearly seen. In the plots for niPCE, it can be seen that the error

strongly depends on the method of numerical quadrature to compute (4.2.3). By choice of the quadrature points and weights alone, niPCE with GQ achieves better accuracy than iPCE in all scenarios. While, therefore, niPCE is a clear winner in this case, the situation is different for $D = 1$, where it can be seen in Figure 5.2.2 that for ETD-RDP and ETDRK4, a lower error can be achieved with iPCE for comparable runtimes (see Table 5.2.1).

In Figure 5.2.3, the error plots for the variance are shown, computed according to formula (4.2.4). In the case of iPCE, the iPCE scheme is run and $\sum_{i=1}^N |u_i(x, T)|^2$ is computed. In the case of niPCE, formula (4.2.3) is applied with $q = 10$. The EE, ETD-RDP and ETDRK4 schemes again form a clear hierarchy from least to most accurate. It can also be seen that the error for ETDRK4 iPCE, $N = 5$, is very similar to that of ETDRK4 niPCE, indicating that the remaining error might not be due to the choice of the numerical scheme ETDRK4 but rather due to approximation steps regarding the gPCE, such as in (5.1.14). Also, MC and QMC yield high errors, which is due to the fact that the norm of the variance is very low compared to the mean, which makes the sampling error in (4.2.3) much more significant. For instance, for the canonical normed Legendre polynomial P_1 , for a random sequence $\{x_j^{\text{MC}}\}_{j=1,\dots,100}$, a Sobol sequence $\{x_j^{\text{QMC}}\}_{j=1,\dots,100}$ and Gauss-Legendre quadrature nodes $\{x_j^{\text{GQ}}\}_{j=1,\dots,100}$ it is (with the corresponding weights as described in (4.2.3))

$$\sum_{j=1}^{100} w_j^{\text{MC}} P_1(x_j^{\text{MC}}) \approx -0.00898, \quad (5.2.5)$$

$$\sum_{j=1}^{100} w_j^{\text{QMC}} P_1(x_j^{\text{QMC}}) \approx -0.00622, \quad (5.2.6)$$

$$\sum_{j=1}^{100} w_j^{\text{GQ}} P_1(x_j^{\text{GQ}}) \approx 2.02 \cdot 10^{-17} \quad (5.2.7)$$

(the first number is an average over 100 Monte Carlo samples) due to MC and QMC being non-symmetric rules, while Gauss-Legendre quadrature nodes are symmetric about the origin. In particular for small t , due to the deterministic initial condition the variance is very close to zero, which, due to sampling errors as demonstrated in (5.2.5), leads to a very large relative error.

5.2.2 One-dimensional random equation with quadratic term

We consider the equation with a quadratic term

$$\frac{\partial u(x, t, \omega)}{\partial t} = D\Delta u(x, t, \omega) - K(\omega)u(x, t, \omega)^2, \quad u(x, 0, \omega) = \cos(\pi x), \quad (5.2.8)$$

Equation with linear term, $D = 0$, mean

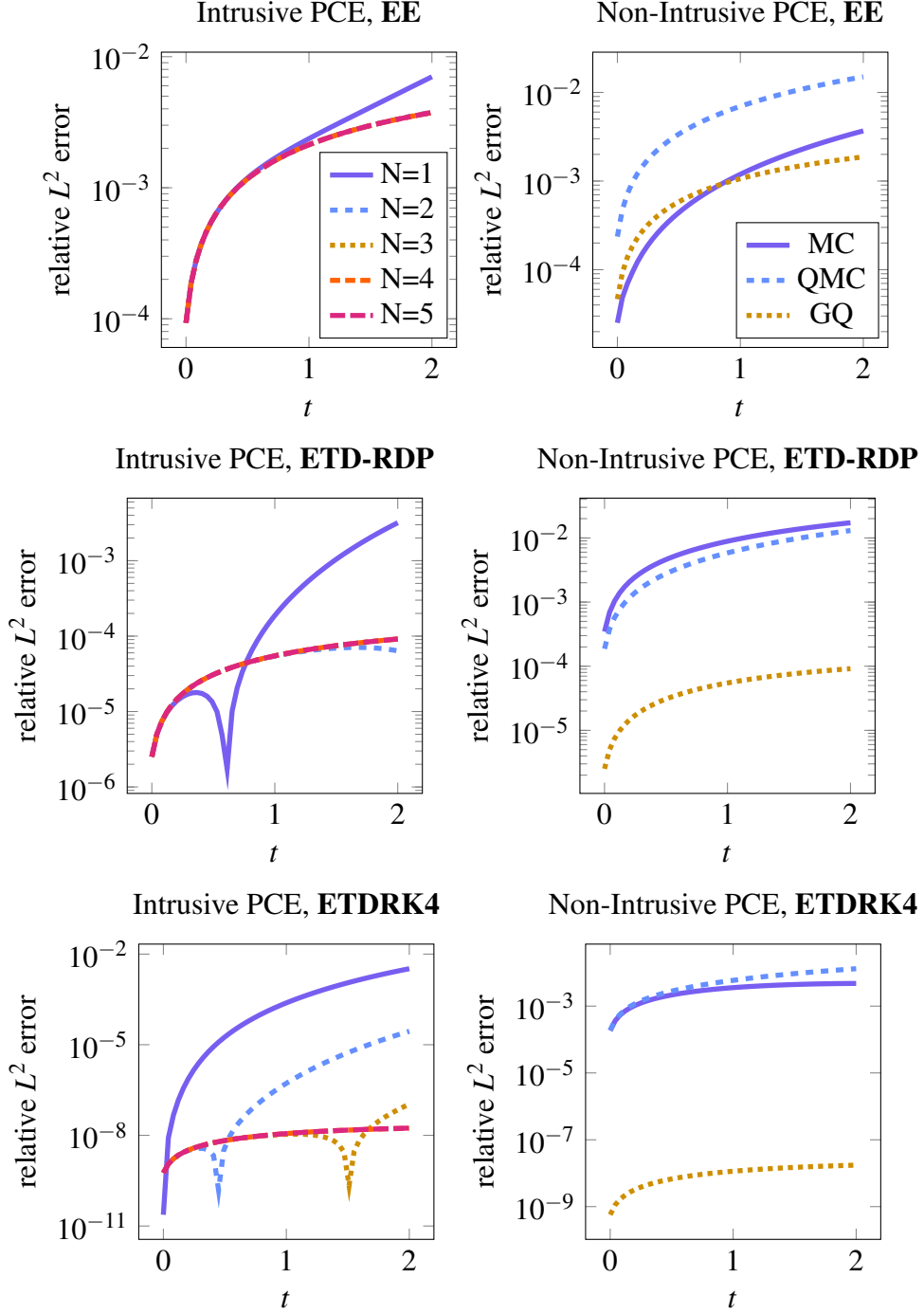


Figure 5.2.1: Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.2) with linear term with diffusion constant $D = 0$. The exact mean is given by (5.2.3).

Equation with linear term, $D = 1$, mean

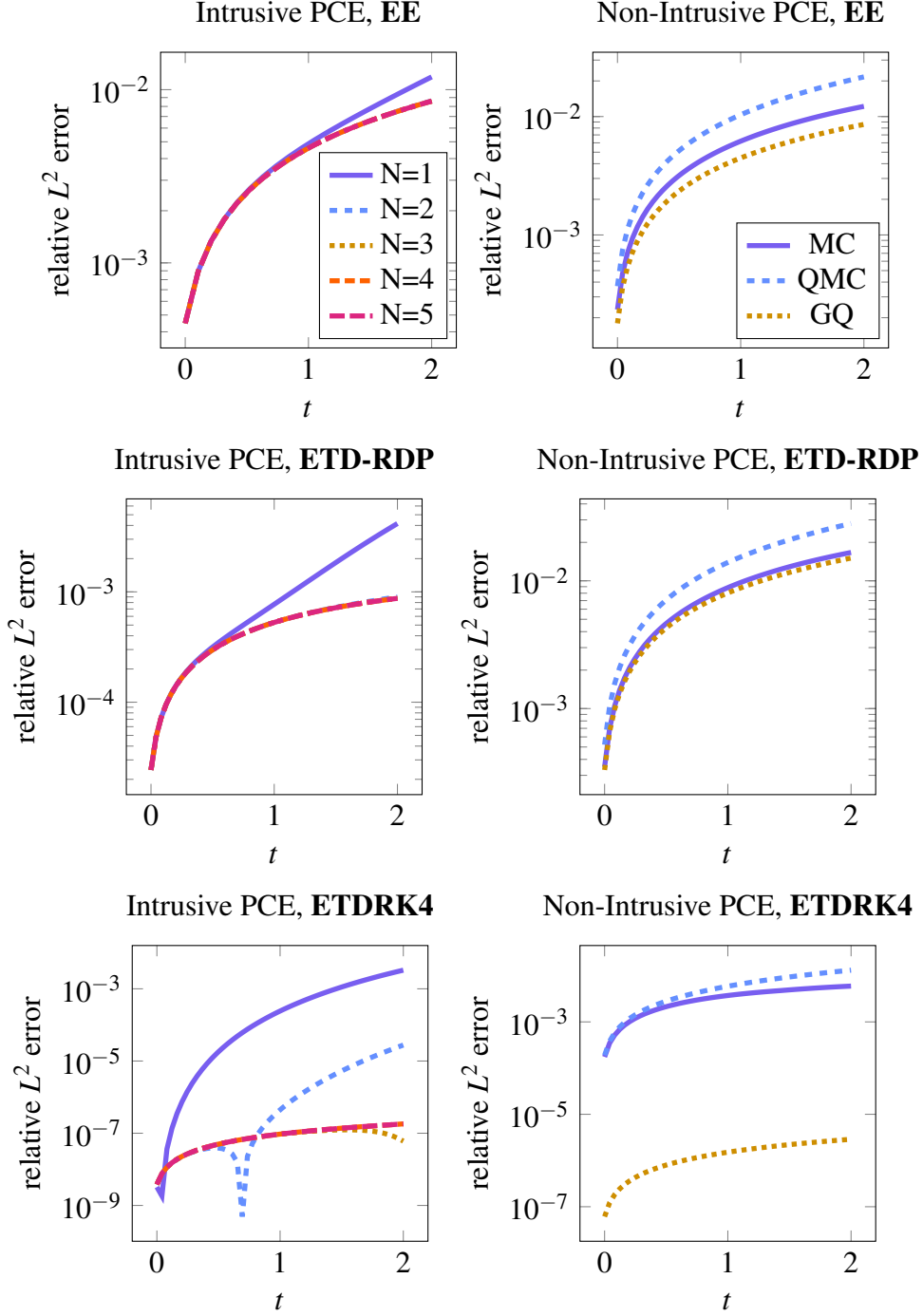


Figure 5.2.2: Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.2) with linear term with diffusion constant $D = 1$. The exact mean is given by (5.2.3).

Equation with linear term, $D = 0$, variance

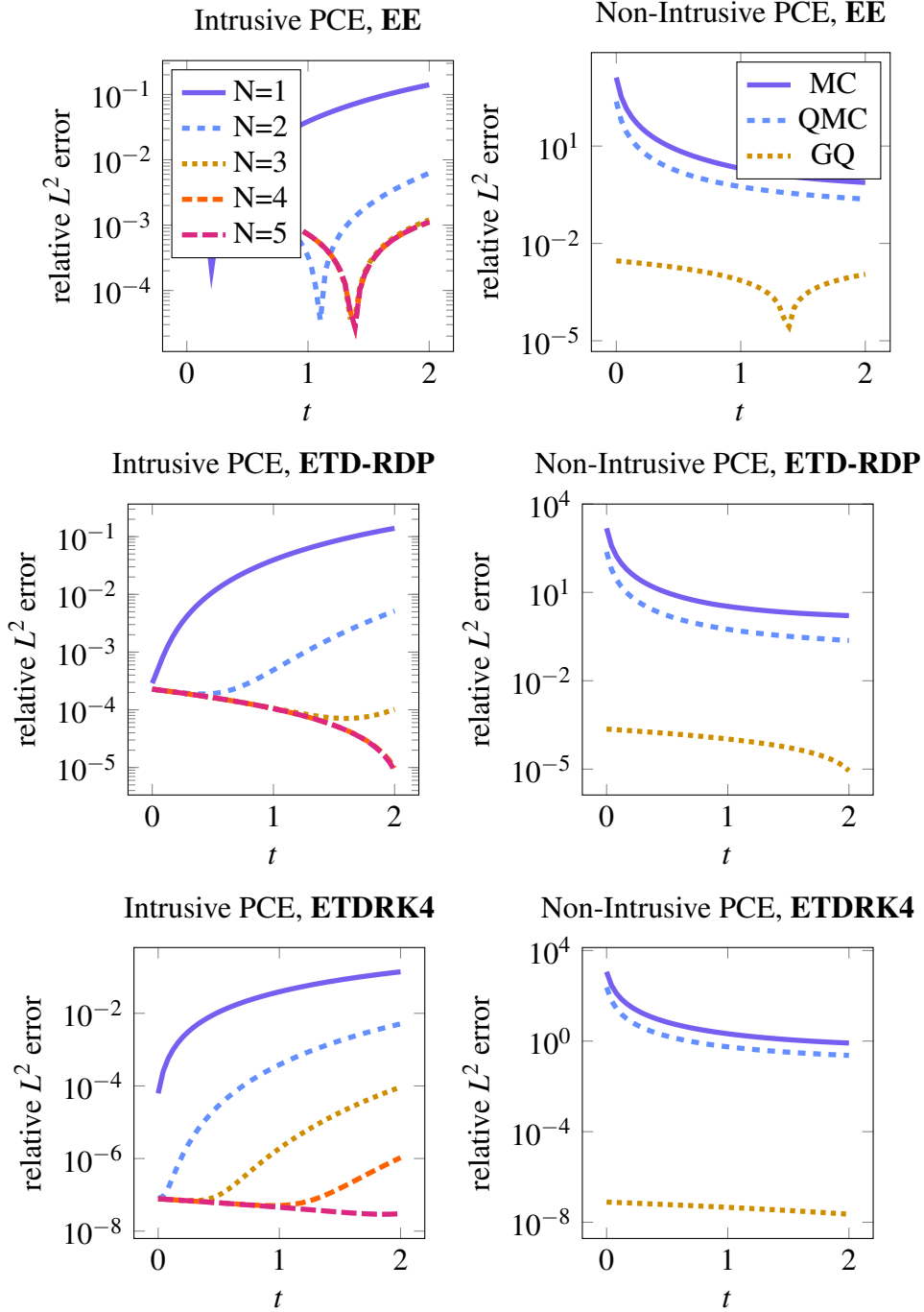


Figure 5.2.3: Plots showing the relative time-dependent error for the same setup as in Figure 5.2.1, but for the variance computed according to equation (4.2.4) with $q = 10$ coefficient functions using formula (4.2.3). The large errors for MC and QMC are due to the variance being close to zero, so that the sampling error introduced in (4.2.3) plays a much bigger role (see (5.2.5)). The exact variance is given by (5.2.4).

$x \in (-1, 1)$, $t \in (0, T]$. Numerical simulations show that for $D = 0$, this equation's solution diverges as early as about $t = 0.5$ for $K = 2$, which is why the errors in Figure 5.2.4 also show a sharp increase towards that point in time. In this case, despite the nonlinearity now being quadratic, iPCE is competitive with niPCE for all three schemes. This is also true for the case $D = 1$ for ETD-RDP and for ETDRK4, as can be seen in Figure 5.2.5.

5.2.3 One-dimensional random equation with cubic term

We consider the equation with a cubic term

$$\frac{\partial u(x, t, \omega)}{\partial t} = D\Delta u(x, t, \omega) - K(\omega)u(x, t, \omega)^3, \quad u(x, 0, \omega) = \cos(\pi x), \quad (5.2.9)$$

$x \in (-1, 1)$, $t \in (0, T]$. The plots in Figure 5.2.6 show that in the case $D = 0$, the iPCE schemes can produce somewhat competitive errors compared to the niPCE schemes for the EE and ETD-RDP cases, but in the ETDRK4 scheme error stays small more consistently in the niPCE case.

For the $D = 1$ case, it is seen in Figure 5.2.7 that both ETD-RDP and ETDRK4 schemes can produce competitive errors in the iPCE case. However, this has to be put into context taking into account the low stability of iPCE in this case: The step sizes for the iPCE schemes have to be increased substantially in order to produce these errors, resulting in much larger computation times (see Table 5.2.2, case $D = 1$, cubic).

5.2.4 Two-dimensional random equation with linear term

The equation

$$\frac{\partial u(\mathbf{x}, t, \omega)}{\partial t} = D\Delta u(\mathbf{x}, t, \omega) - K(\omega)u(\mathbf{x}, t, \omega), \quad u(\mathbf{x}, 0, \omega) = \cos(\pi x_1) \cos(\pi x_2) \quad (5.2.10)$$

with $\mathbf{x} = (x_1, x_2)^\top \in (-1, 1)^2$, $t \in (0, T]$, has the exact solution $u(\mathbf{x}, t, \omega) = \exp(-(K(\omega) + 2D\pi^2)t) \cos(\pi x_1) \cos(\pi x_2)$. If $K \sim \mathcal{U}[a, b]$, the expected value is given by

$$\begin{aligned} \mathbb{E}[u(\mathbf{x}, t, \cdot)] &= \frac{1}{b-a} \int_a^b \exp(-(\xi + 2D\pi^2)t) \cos(\pi x_1) \cos(\pi x_2) d\xi \\ &= \frac{1}{(b-a)t} \cos(\pi x_1) \cos(\pi x_2) \exp(-(2D\pi^2 + a)t) - \exp(-(2D\pi^2 + b)t). \end{aligned} \quad (5.2.11)$$

Equation with quadratic term, $D = 0$, mean

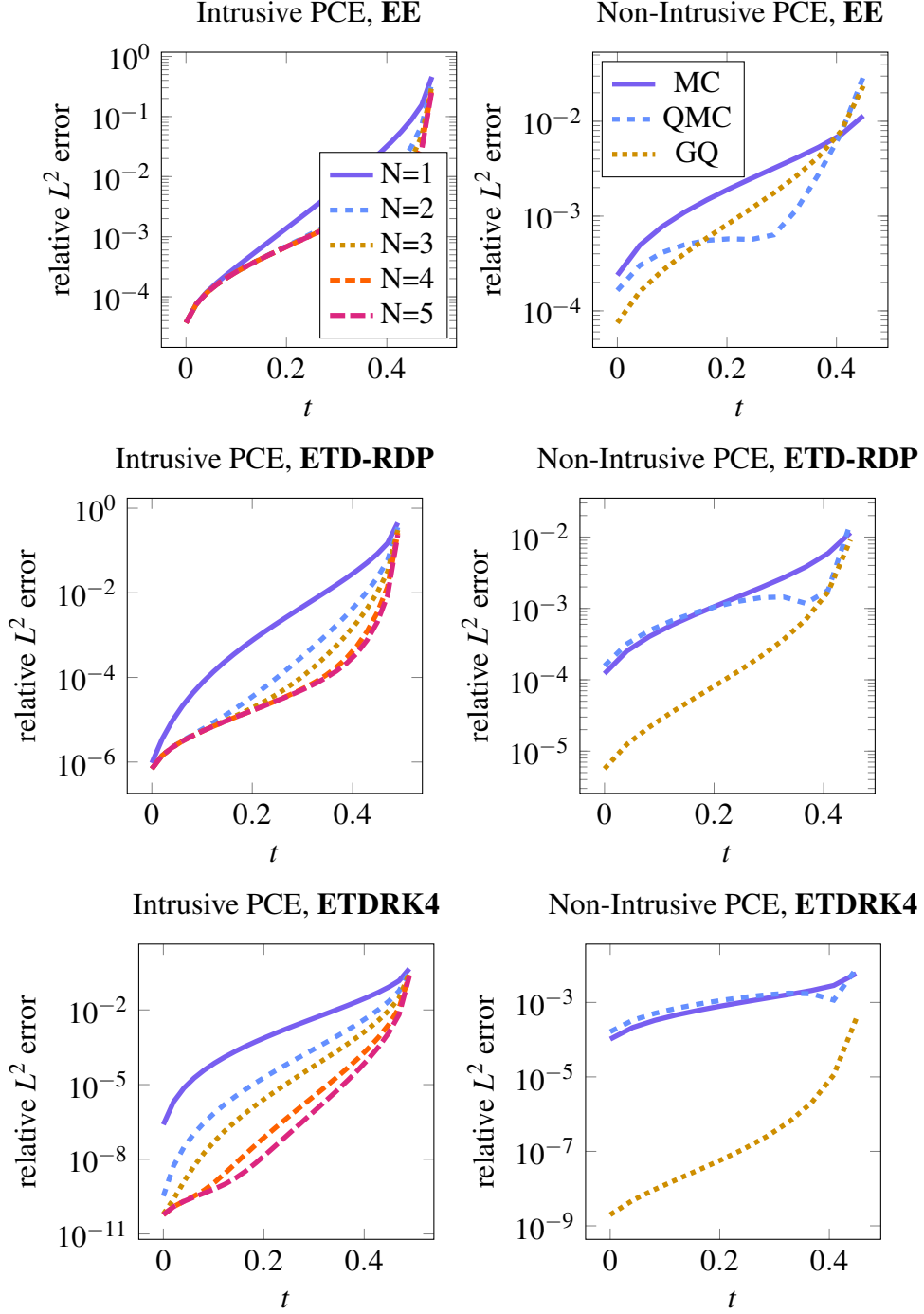


Figure 5.2.4: Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.8) with quadratic term with diffusion constant $D = 0$.

Equation with quadratic term, $D = 1$, mean

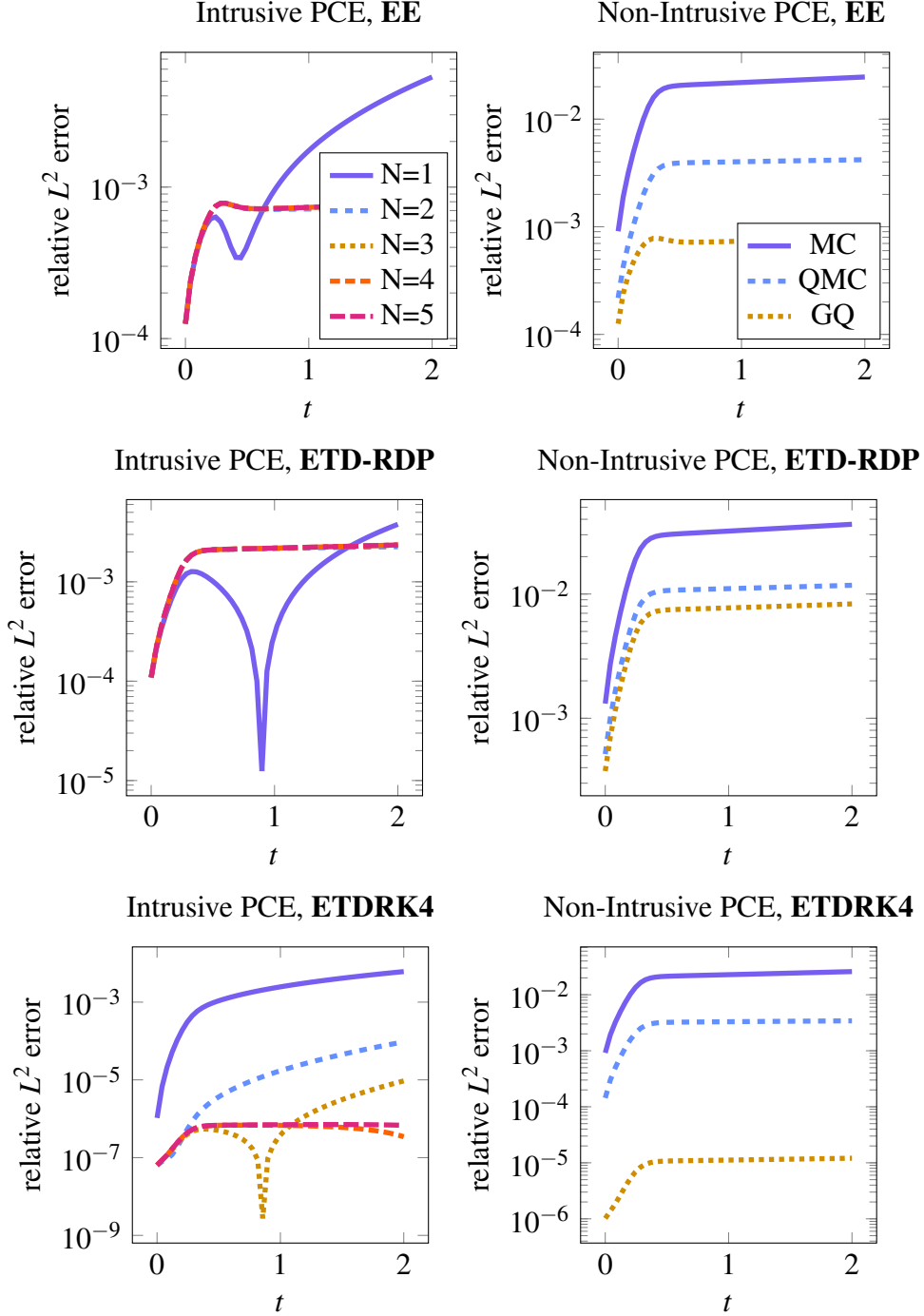


Figure 5.2.5: Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.8) with quadratic term with diffusion constant $D = 1$.

Figure 5.2.9 shows the time-dependent errors for the case $D = 0$, using the same parameters and time step numbers as in the 1D cases shown in Figure 5.2.1 and 5.2.2.

5.2.5 Performance plots

In Figure 5.2.8, for the different equations we show how the error for iPCE and niPCE compare for different numbers of time steps M . For the niPCE errors, $q = 10$ Gauss-Legendre quadrature points were used. It can be seen that in most cases, the iPCE and niPCE errors are very similar up to a certain M , where the error lines split up and iPCE does not go below a certain threshold. This threshold can, in some cases, be higher than the niPCE error, so that the iPCE error appears as a constant line, independent of M (as in the quadratic and cubic case for $D = 0$). This additional error for iPCE may be caused by the approximation of the iPCE product terms given in (5.1.14).

5.2.6 Random Gray-Scott model

We consider the random Gray-Scott model

$$\begin{cases} \frac{\partial u(x,t,\omega)}{\partial t} = D_u \Delta u(x,t,\omega) - u(x,t,\omega)v(x,t,\omega)^2 + F(1 - u(x,t,\omega)), \\ \frac{\partial v(x,t,\omega)}{\partial t} = D_v \Delta v(x,t,\omega) + u(x,t,\omega)v(x,t,\omega)^2 - (F + k(\omega))v(x,t,\omega). \end{cases} \quad (5.2.12)$$

for $x \in (-1, 1)$, $t \in (0, T]$ for some $T > 0$, $D_u = 2 \cdot 10^{-5}$, $D_v = 10^{-5}$, a constant F and a uniformly distributed random variable k . We take the initial condition from [40, Figure 9] which is given by

$$u_{\text{init}}(x) = 1 - \frac{5}{3\sqrt{2\pi}} \exp(-6(x - \mu)^2), \quad v_{\text{init}}(x) = 0.37 \cdot \frac{7.5}{2\sqrt{2}\Gamma(\frac{1}{3})} \exp\left(\frac{-7(x - \mu)^3}{\sqrt{2}}\right)$$

where $\mu = 0$ is the midpoint of the interval. For the two-dimensional plots shown in Figures 5.2.11 and 5.2.12, we use as an initial condition a function with four off-center local extrema:

$$v(x, y, 0) = \frac{1}{4} \sum_{i=1}^4 \exp(-150((x - x_i)^2 + (y - y_i)^2)), \quad u(x, y, 0) = 1 - v(x, y, 0), \quad (5.2.13)$$

where $(x_1, y_1) = (2/7, 2/7)$, $(x_2, y_2) = (-2/7, 2/7)$, $(x_3, y_3) = (2/7, -2/7)$ and $(x_4, y_4) = (-2/7, -2/7)$.

Remark 5.2.1 *The Gray-Scott model has been extensively studied and is known to show a wide variety of pattern formation behavior. The patterns are called Turing patterns, for*

Equation with cubic term, $D = 0$, mean

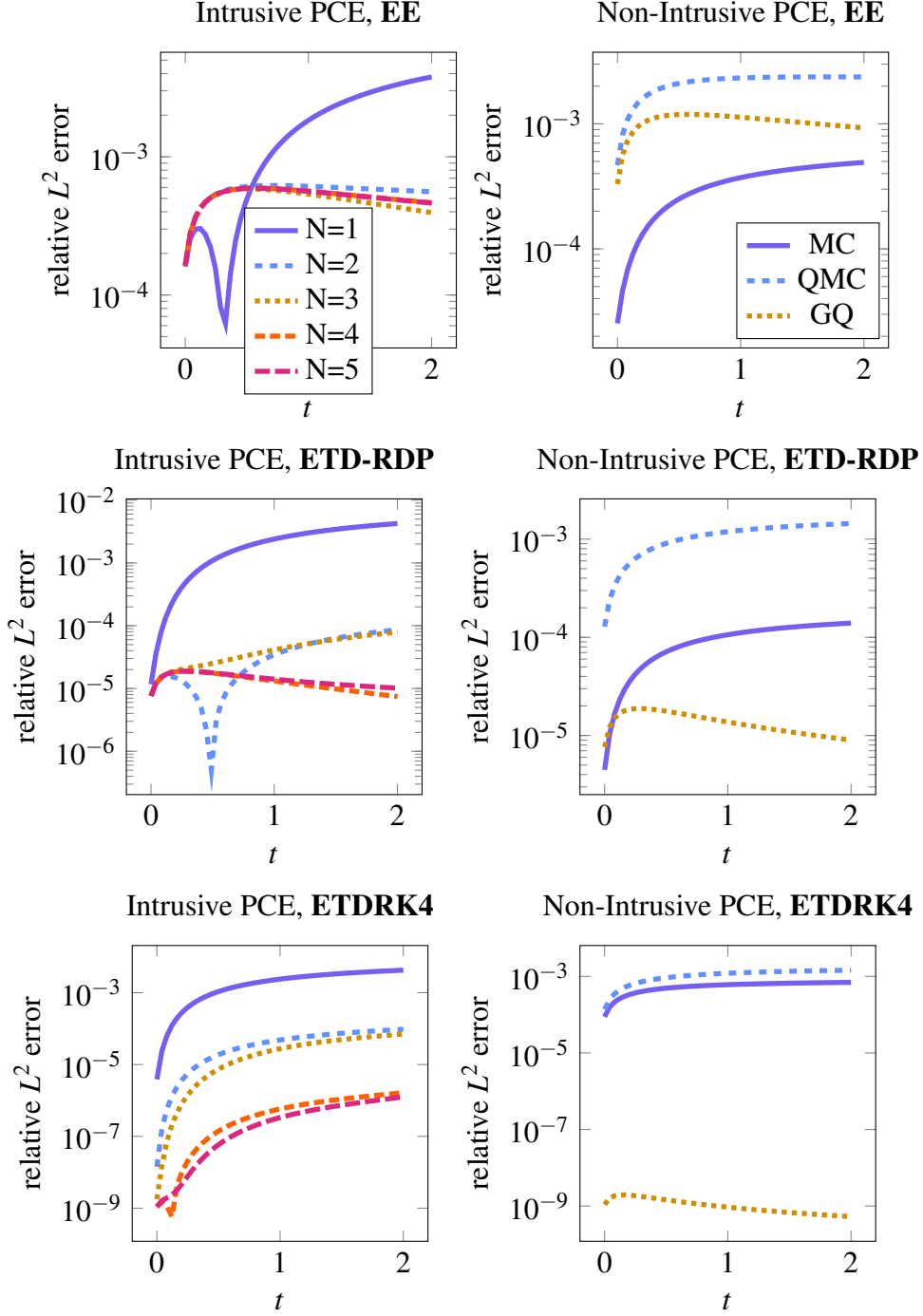


Figure 5.2.6: Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.9) with cubic term with diffusion constant $D = 0$.

Equation with cubic term, $D = 1$, mean

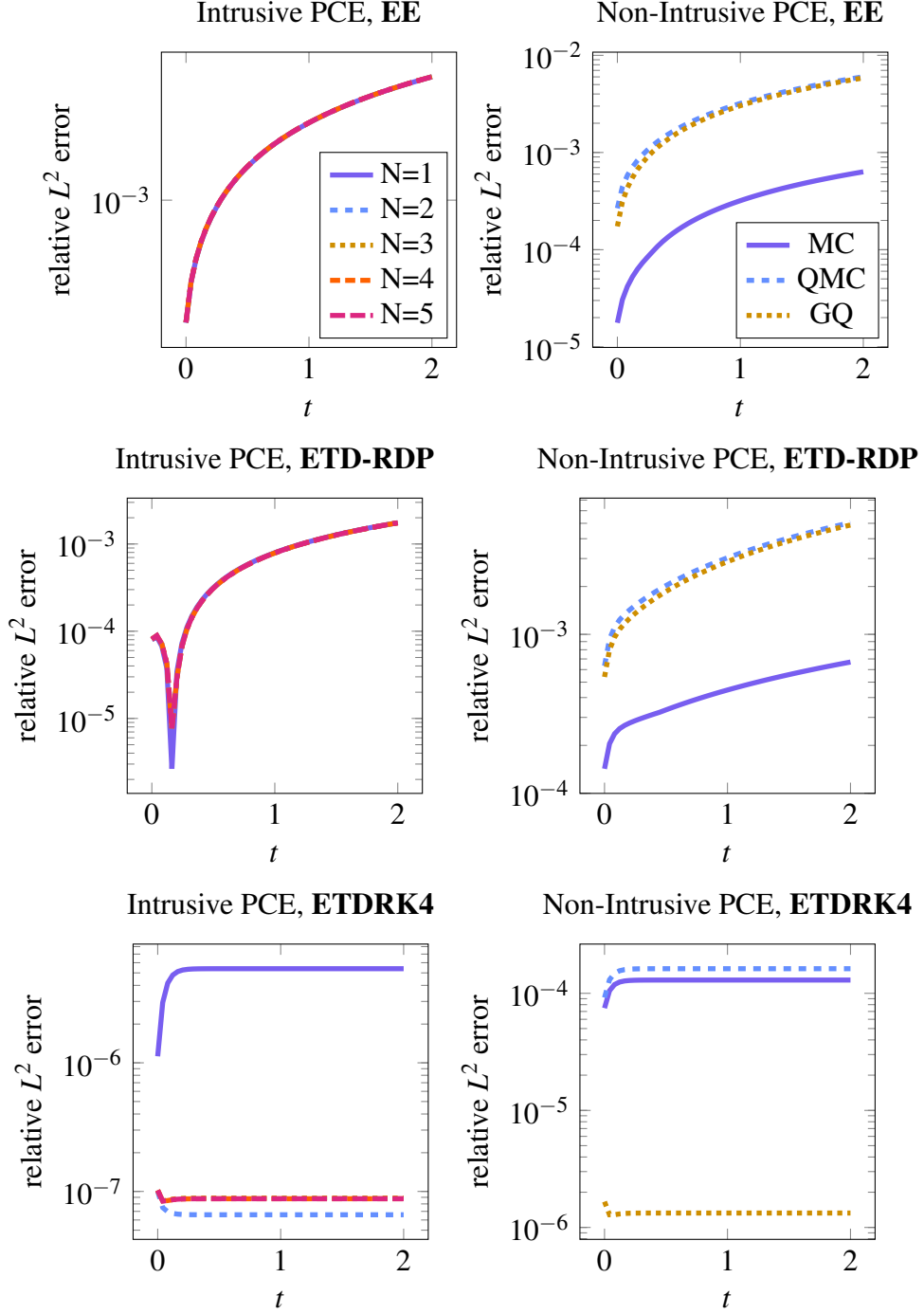


Figure 5.2.7: Plots showing the relative time-dependent error for iPCE schemes (Algorithms 5.1, 5.2, 5.4, left-hand side) and niPCE schemes (right-hand side) for equation (5.2.9) with cubic term with diffusion constant $D = 1$.

Performance plots for equations with linear, quadratic and cubic terms

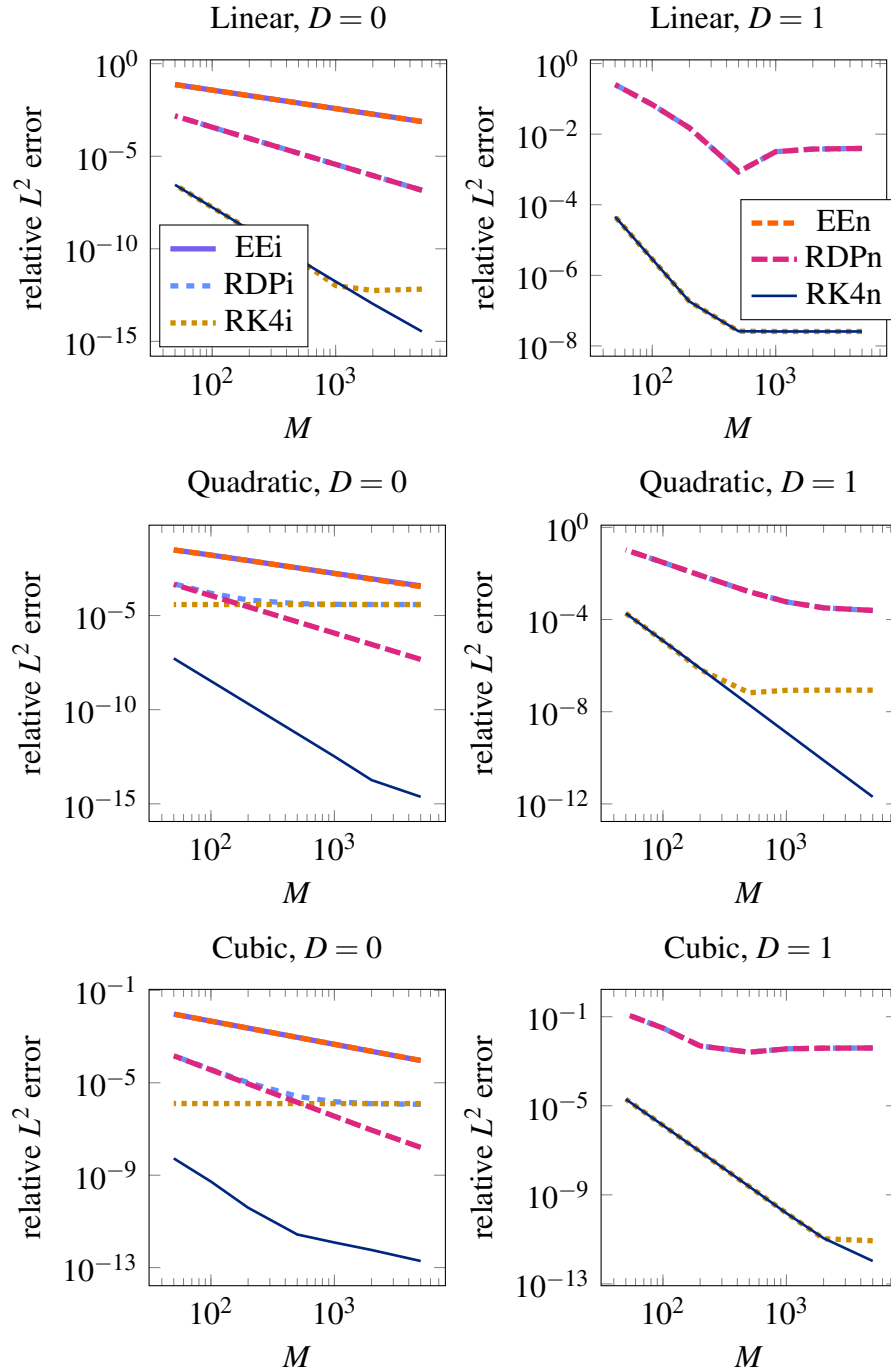


Figure 5.2.8: Plots showing the error for the different equations at time $T = 2$ (except for equation 5.2.8 with the quadratic term and $D = 0$, where $T = 0.4$). The shorthands EEi, EEn in the legend stand for EE iPCE and EE niPCE, respectively, and analogously with RDP for ETD-RDP and RK4 for ETD-RK4. For $D = 1$, only ETD-RDP and ETD-RK4 are shown, since EE requires higher M in order to be stable.

an overview we refer to [84, 78]. Depending on the parameter values of k and F , u and v may enter trivial or non-trivial homogeneous steady states, or non-homogeneous states called Turing patterns. A necessary condition in the k - F parameter space in order for Turing pattern formation to occur is [76, Eq. (19)]

$$[2(F + k) - (v_0^2 + F)]^2 > 8(F + k)(v_0^2 - F), \quad (5.2.14)$$

where v_0 is either the trivial steady state (the ‘red state’) $v_R = 0$ or, if $d := 1 - 4(F + k)^2/F > 0$ holds, one of the two non-trivial steady states [76, Eqs. (7), (8)] (with v_B also called the ‘blue state’)

$$v_B = \frac{1}{2}\alpha(1 + \sqrt{d}), \quad v_1 = \frac{1}{2}\alpha(1 - \sqrt{d}).$$

Since now there are two coupled functions instead of one, the schemes need to take into account these two functions, but for the sake of brevity we will not spell out all the schemes again, but refer to [5, 62] (the intrusive PCE schemes given in Section 5.1 can then easily be extended to two functions).

We pick for all simulations (1D and 2D) $F = 0.04$ and $k(\omega) \sim \mathcal{U}[0.058, 0.062]$, which falls into the region of complex pattern formation described by (5.2.14). It is seen in Figure 5.2.10 that the iPCE schemes fail to compute correct solutions for the Gray-Scott model. Furthermore, the needed time step numbers in order for the iPCE schemes are even higher than for Equation 5.2.9 with a single cubic term. This leaves niPCE as the only viable option to treat the random Gray-Scott system. We show an example of such a simulation in two spatial dimensions for a mean $\mathbb{E}[u(x, t, \cdot)]$ in Figure 5.2.12 with the correct result produced by niPCE juxtaposed by iPCE simulations which fail to reproduce the pattern. This is likely due to the challenge of sharp dependency in the random variable, as in the seemingly small interval $k \in [0.058, 0.062]$, a wide range of patterns emerge (see also [78]). Furthermore, long-term integration of the Gray-Scott system is a challenge even in the deterministic case, as show in Figure 5.2.11 for $T = 5000$, where it is seen that a high level of accuracy in both space and time is needed in order to produce the correct pattern shown on the bottom right. Especially in the cases of EE and ETD-RDP-IF, for $p = 256$ the error of the second-order spatial finite difference approximation appears to dominate, causing the pattern to look rather different.

Given the difficulties of long-term integration for the Gray-Scott model and the numerical challenges and substantially increased computation time for iPCE, it appears to be advisable to use niPCE for systems with strong nonlinearities such as Gray-Scott.

iPCE errors for 2D random equation with linear term, $D = 1$

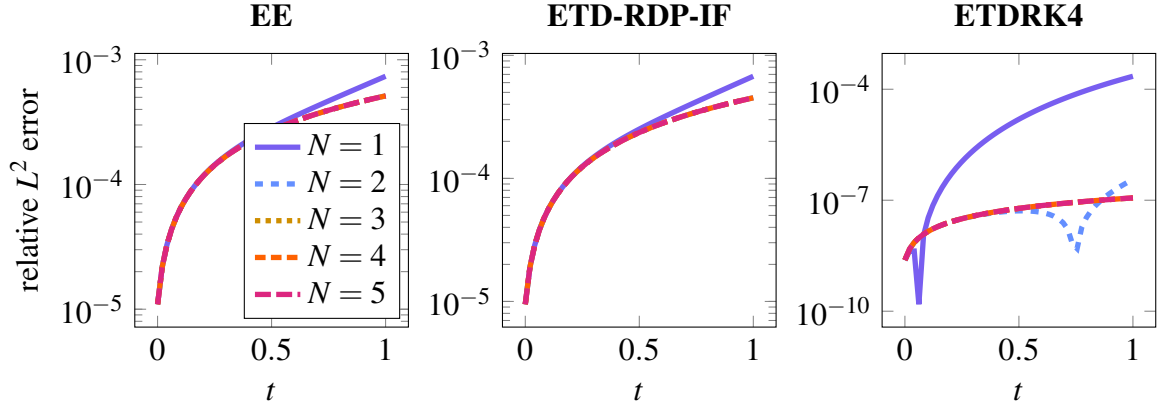


Figure 5.2.9: Time-dependent L^2 errors for equation (5.2.10), using the iPCE algorithms 5.1, 5.3 and 5.4. We used the same parameters and numbers of time steps as in the 1D case from Figure 5.2.1 and Table 5.2.2. The exact expected value is given by (5.2.11). The spatial resolution in the ETD-RDP case was picked in this simulation as $p = 512$. For lower p , the error caused by the finite difference discretization dominates and the curves for different N are identical.

Errors for a random Gray-Scott system

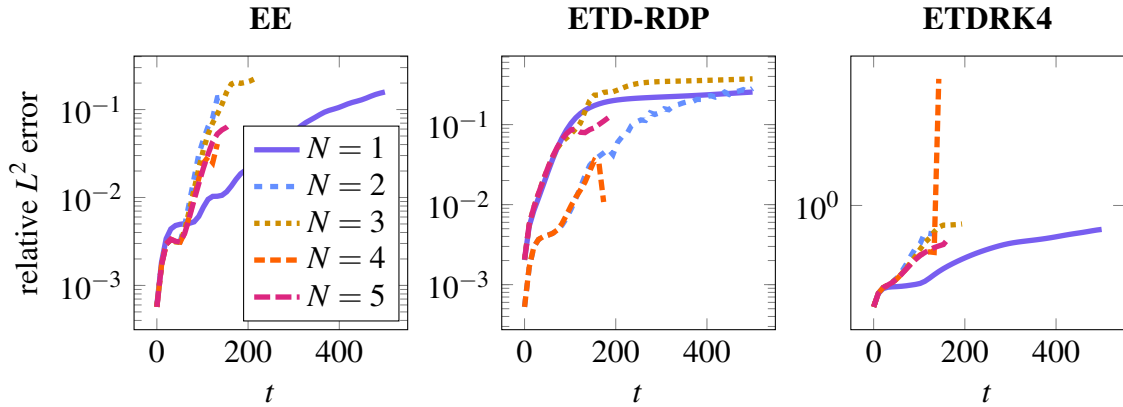


Figure 5.2.10: Time-dependent error plot for a random 1D Gray-Scott system. None of the iPCE schemes work in this case: Initially, the solution is relatively accurate, but as pattern formation starts to occur, iPCE breaks down. For a visual example in two dimensions, see Figure 5.2.12.

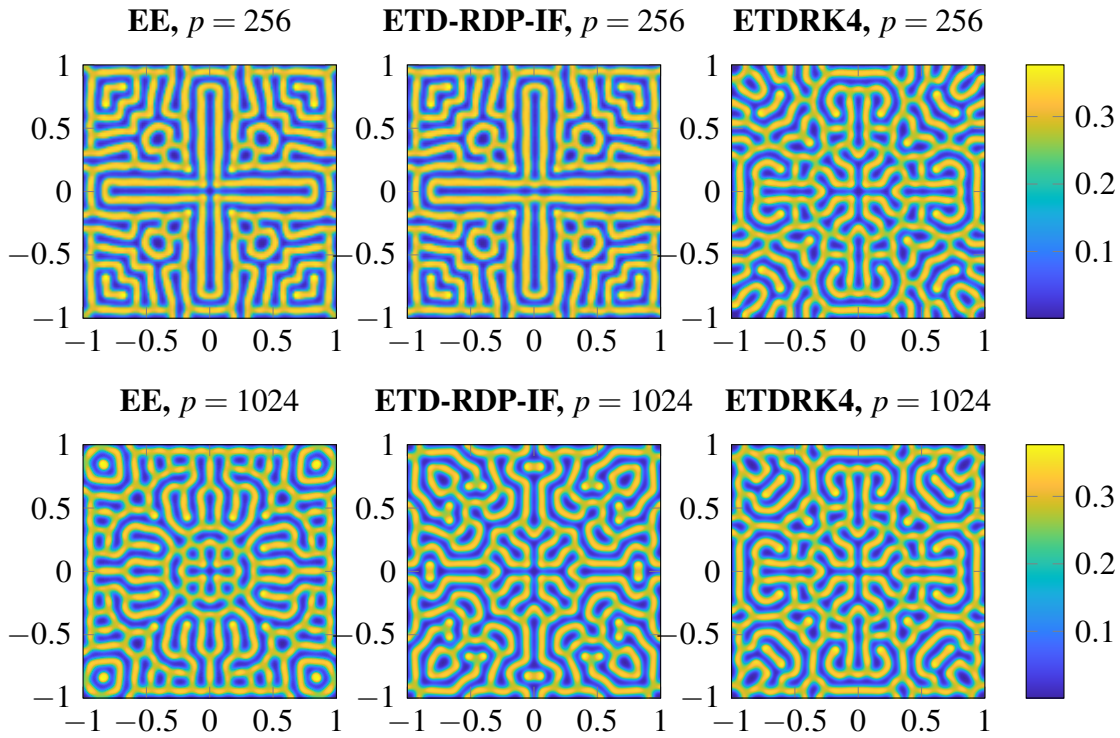


Figure 5.2.11: Deterministic simulations of Gray-Scott patterns for different schemes and resolutions for the same initial condition at $T = 5000$ (using Algorithms 4.1, 4.3, and 4.5). It becomes apparent that after a long-term integration, Gray-Scott patterns are extremely sensitive to the spatial resolution (and number of time steps), and for the correct solution, a high spatial resolution is needed, along with a scheme such as ETDRK4 which performs a spectral approximation in space. The numbers of time steps used are $M = 100000$ for EE and $M = 10000$ for ETD-RDP-IF and ETDRK4.

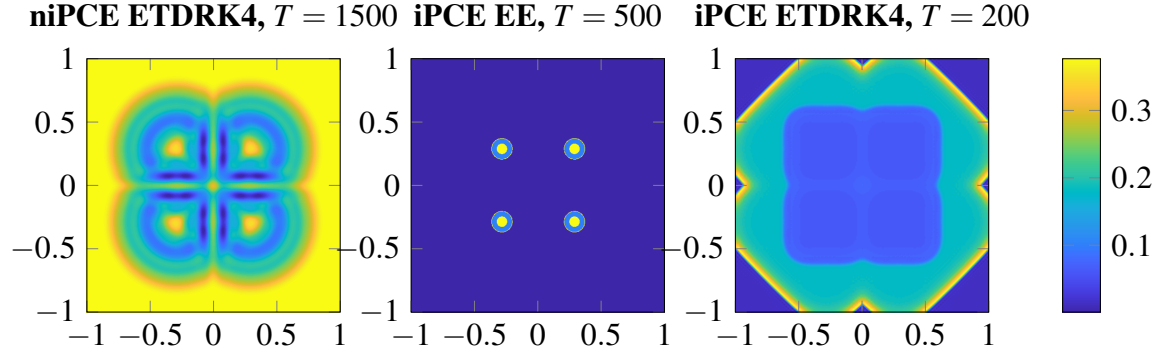


Figure 5.2.12: Simulations for $\mathbb{E}[u(x, T, \cdot)]$ of the random Gray-Scott system (5.2.12) with $F = 0.04$ and $k(\omega) \sim \mathcal{U}[0.058, 0.062]$. A niPCE simulation with ETD RK4 (left-hand side) and GQ gives an impression of superimposed patterns for different k . Over time, the four off-center bumps of the initial condition (5.2.13) expand and connect, forming intricate patterns as observed on the left-hand side. The EE iPCE simulation fails to correctly propagate the patterns, and the four rings from the initial condition stop expanding. For the ETD RK4 iPCE simulation, the edges of the rings propagate too fast, and no pattern formation is observed.

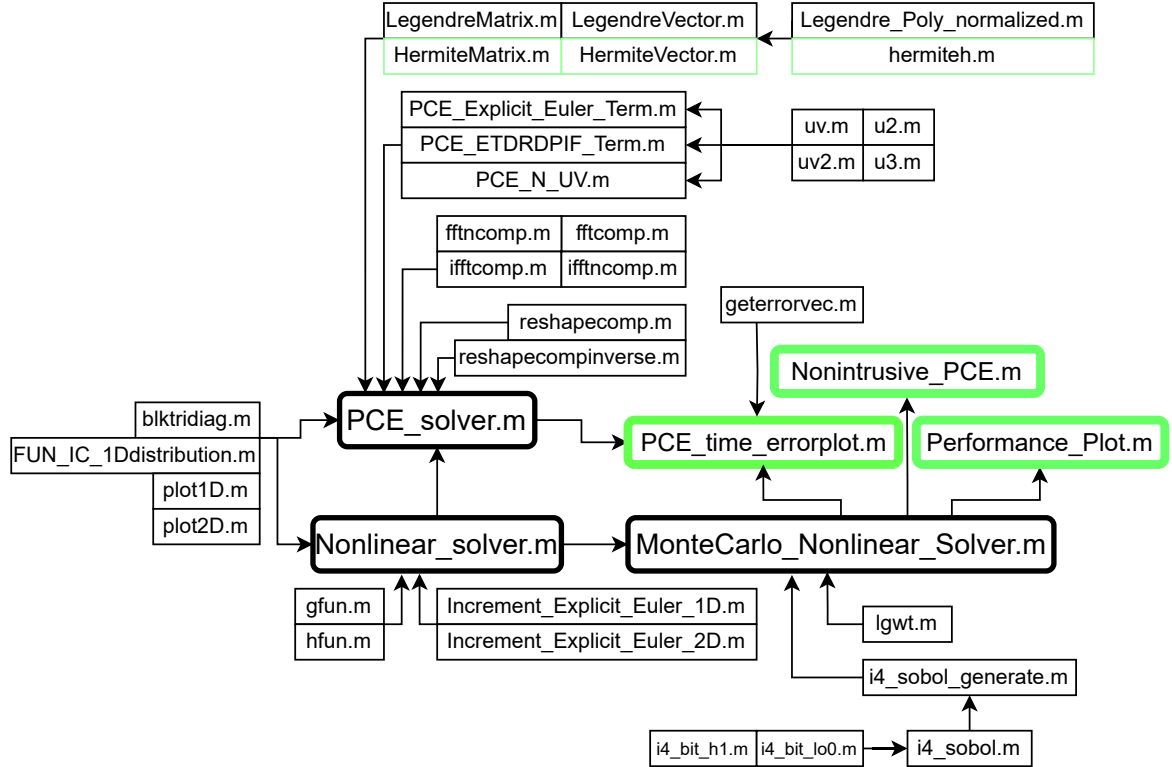


Figure 5.2.13: A diagram showing all the different MATLAB programs in the repository and how they are connected. The error plots in Figures 5.2.1 to 5.2.7, as well as Figure 5.2.9 are generated by `PCE_time_errorplot.m` for iPCE and by `Nonintrusive_PCE.m` for niPCE. The program `Performance_Plot.m` generates the plots in Figure 5.2.8. The programs in the repository contain more detailed descriptions.

$D = 0$, linear			$D = 1$, linear		
	Intrusive	Non-Intrusive		Intrusive	Non-Intrusive
EE	0.3225	0.6717	EE	0.4386	6.4825
ETD-RDP	0.4112	0.2256	ETD-RDP	0.2741	0.3698
ETDRK4	0.4322	0.4244	ETDRK4	0.7304	0.4809
$D = 0$, quadratic			$D = 1$, quadratic		
	Intrusive	Non-Intrusive		Intrusive	Non-Intrusive
EE	0.5090	0.7563	EE	3.5583	5.8424
ETD-RDP	0.3143	0.2550	ETD-RDP	0.4906	0.3265
ETDRK4	0.7242	0.6093	ETDRK4	1.1047	0.4916
$D = 0$, cubic					
	Intrusive	Non-Intrusive		Intrusive	Non-Intrusive
EE	12.6065	0.4934			
ETD-RDP	5.2208	0.4339			
ETDRK4	6.1375	0.5221			
$D = 1$, cubic					
	Intrusive	Non-Intrusive		Intrusive	Non-Intrusive
EE	247.6447	0.3319			
ETD-RDP	9.8553	0.5722			
ETDRK4	12.4297	0.5626			

Table 5.2.1: Runtimes for the created plots, all times in seconds, for iPCE with $N = 5$ and for niPCE with 50 realizations

$D = 0$, linear				$D = 1$, linear			
	iPCE M	niPCE $M \quad q$			iPCE M	niPCE $M \quad q$	
EE	1000	2000	50	EE	20000	20000	50
ETD-RDP	200	200	50	ETD-RDP	400	200	50
ETDRK4	100	100	50	ETDRK4	200	100	50
ETDRK4 ref.		-	-	ETDRK4 ref.		1000	200
$D = 0$, quadratic				$D = 1$, quadratic			
	iPCE M	niPCE $M \quad q$			iPCE M	niPCE $M \quad q$	
EE	1000	2000	50	EE	10000	20000	50
ETD-RDP	200	200	50	ETD-RDP	400	200	50
ETDRK4	100	100	50	ETDRK4	200	100	50
ETDRK4 ref.		1000	200	ETDRK4 ref.		1000	200
$D = 0$, cubic				$D = 1$, cubic			
	iPCE M	niPCE $M \quad q$			iPCE M	niPCE $M \quad q$	
EE	1000	500	50	EE	20000	20000	50
ETD-RDP	200	200	50	ETD-RDP	400	200	50
ETDRK4	100	100	50	ETDRK4	200	100	50
ETDRK4 ref.		1000	200	ETDRK4 ref.		1000	200

Table 5.2.2: Numbers of step sizes M and of samples (for niPCE) for each simulation shown in Figures 5.2.1 to 5.2.7. ‘ETDRK4 ref.’ refers to the reference solution used for that simulation. For the linear equation with $D = 0$, the exact solution is known.

Chapter 6

Summary and Outlook

6.1 SPDEs on two-dimensional domains

After providing the necessary mathematical framework in Chapter 2, in Chapter 3, we implemented and derived error bounds for the exponential Euler scheme for semilinear parabolic SPDEs on a curvilinear two-dimensional domain. For the spectral approximation in space, the necessary Dirichlet eigenvalues and eigenfunctions were numerically approximated using a boundary element method and Beyn's contour integral algorithm.

The advantages of the presented method are its flexibility in choosing the two-dimensional domain and the possibility to freely choose the points inside the domain at which the eigenfunctions are computed. This renders the solution of SPDEs with high resolution on a big class of domains possible. Due to the general setting of the spectral method, the scheme could be extended to a larger class of differential operators such as general second-order elliptic differential operators, whose Dirichlet eigenfunctions also form a complete orthogonal basis in $L^2(D)$ (see [15]). Difficulties might arise while using a boundary integral equation method, as explicit knowledge of the fundamental solution of the differential operator is required. The scheme could also be applied to three-dimensional domains with a \mathcal{C}^2 boundary: In order to do this, the fundamental solution needs to be changed to the one for the three-dimensional Helmholtz operator, the curved elements are now two-dimensional (see [68, Section 5]) and the linear system resulting from applying Beyn's contour integral algorithm is larger. However, for a small error, a very large number of eigenfunctions would have to be computed due to Weyl's law (3.1.1) and the error bound (3.3.15).

The requirements on the nonlinear operator F could be relaxed by implementing a different scheme like the one from [58], where the requirements on F are less strict. Future work might also include deriving error bounds for higher moments such as $\mathbb{E}[\|U(t_k) - V_k^{(4)}\|^2]^{1/2}$. Another avenue for future research could be the solution of SPDEs on general surfaces

or Riemannian manifolds by a spectral method like the one used in this work. There are, however, several obstacles: While the numerical approximation of eigenfunctions of the Laplacian on manifolds (the Laplace-Beltrami operator) has been investigated (see e.g. [80]), the approximations are not nearly as accurate as the ones used in this work for the Laplacian on a planar domain. Therefore, a better error bound, more computing power or more efficient approximation methods for Laplace-Beltrami eigenfunctions are needed. A rigorous mathematical treatment in the context of Riemannian geometry is beyond the scope of this thesis, for references on random fields and stochastic analysis on manifolds see [56].

6.2 Polynomial chaos with exponential time differencing

In Chapter 5, we investigated how the ETD-RDP-IF and ETDRK4 schemes can be implemented in an iPCE scheme and compared the performance of these two schemes and an EE scheme to a niPCE approach.

While niPCE using Monte Carlo methods or Gaussian quadrature is in most cases superior to iPCE, we also found that in some cases such as the model equation with a quadratic term, iPCE results in lower errors for comparable runtimes. For complex pattern formation dynamics such as the Gray-Scott model, iPCE breaks down for all three schemes which leaves only the non-intrusive variance as a viable option. While not implemented in full detail, it is also apparent that the curse of dimensionality poses a bigger problem to iPCE than it does to niPCE since solving bigger iPCE systems scales much worse than using sparse grids for niPCE.

Future work could include methods to remedy the shortcomings of iPCE for complex dynamical systems. In this work, it was seen that a direct iPCE implementation of algorithms which are powerful in the deterministic case is not sufficient. Works such as [13] have introduced an asynchronous time integration method for iPCE to deal with sharp dependencies in the random variable. In [99], a multi-element PCE method is proposed to handle long-term integration, which could be investigated in combination with the presented exponential time differencing schemes. Recent work [34] also shows that B-splines can be used instead of classical orthogonal polynomial bases to further reduce the error. One possibility for achieving speedups for the ETD-RDP-IF scheme is the use of parallelization as has been demonstrated in [5, pp. 8]. It could be carried out in an analogous fashion for the iPCE scheme.

Another topic which could be investigated are non-polynomial nonlinear functions, which can be handled using truncations of Taylor series [30].

Bibliography

- [1] W. A. Al-Salam. On the product of two Legendre polynomials. *Mathematica Scandinavica*, pages 239–242, 1956.
- [2] A. Alabert and I. Gyöngy. On numerical approximation of stochastic burgers' equation. In *From Stochastic Calculus to Mathematical Finance*, pages 1–15. Springer, 2006.
- [3] E. J. Allen, S. J. Novosel, and Z. Zhang. Finite element and difference approximation of some linear stochastic partial differential equations. *Stochastics and Stochastic Reports*, 64:117–142, 1998.
- [4] E. O. Asante-Asamani. *An Exponential Time Differencing Scheme with a Real Distinct Poles Rational Function for Advection-Diffusion Reaction Equations*. PhD thesis, University of Wisconsin Milwaukee, 2016.
- [5] E. O. Asante-Asamani, A. Kleefeld, and B. A. Wade. A second-order exponential time differencing scheme for non-linear reaction-diffusion systems with dimensional splitting. *Journal of Computational Physics*, 415:109490, 2020.
- [6] R. Askey and J. A. Wilson. *Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials*. Memoirs of the American Mathematical Society, no. 319. American Mathematical Society, 1985.
- [7] K. E. Atkinson. The numerical solution of integral equations of the second kind. *Cambridge Monographs on Applied and Computational Mathematics*, 1996.
- [8] D. Benson. *Music: A Mathematical Offering*. Cambridge University Press, 2008.
- [9] W.-J. Beyn. An integral method for solving nonlinear eigenvalue problems. *Linear Algebra and its Applications*, 436(10):3839–3863, 2012.
- [10] D. Blömker. Nonhomogeneous noise and Q-Wiener processes on bounded domains. *Stochastic Analysis and Applications*, 23(2):255–273, 2005.
- [11] D. Blömker and A. Jentzen. Galerkin approximations for the stochastic Burgers equation. 2009.
- [12] S. Bonaccorsi and G. Guatteri. Classical solutions for SPDEs with Dirichlet boundary conditions. In *Seminar on Stochastic Analysis, Random Fields and Applications III: Centro Stefano Franscini, Ascona, September 1999*. Springer, 2002.

- [13] P. Bonnaire, P. Pettersson, and C. F. Silva. Intrusive generalized polynomial chaos with asynchronous time integration for the solution of the unsteady Navier–Stokes equations. *Computers & Fluids*, 223:104952, 2021.
- [14] C.-E. Bréhier and G. Vilmart. High order integrator for sampling the invariant distribution of a class of parabolic stochastic PDEs with additive space-time noise. *SIAM Journal on Scientific Computing*, 38(4):2283–2306, 2016.
- [15] F. E. Browder. On the eigenfunctions and eigenvalues of the general linear elliptic differential operator. *Proceedings of the National Academy of Sciences*, 39(5):433–439, 1953.
- [16] O. P. Bruno and F. Reitich. High-order methods for high-frequency scattering applications. In H. Ammari, editor, *Modeling and Computations in Electromagnetics*, pages 129–163. Springer, 2008.
- [17] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.
- [18] S. N. Chandler-Wilde, D. P. Hewett, S. Langdon, and A. Twigger. A high frequency boundary element method for scattering by a class of nonconvex obstacles. *Numerische Mathematik*, 129(4):647–689, 2015.
- [19] P.-L. Chow. *Stochastic Partial Differential Equations*. CRC Press, 2007.
- [20] J. Clausnitzer and A. Kleefeld. Comparing intrusive and non-intrusive polynomial chaos for a class of exponential time differencing schemes. *arXiv:2311.16921*, 2023.
- [21] J. Clausnitzer and A. Kleefeld. A spectral Galerkin exponential Euler time-stepping scheme for parabolic SPDEs on two-dimensional domains with a \mathcal{C}^2 boundary. *Discrete and Continuous Dynamical Systems B*, 2023.
- [22] D. Cohen and A. Lang. Numerical approximation and simulation of the stochastic wave equation on the sphere. *Calcolo*, 59(3):32, 2022.
- [23] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer, 2013.
- [24] R. Courant and D. Hilbert. *Mathematical Methods of Physics*. Interscience Publishers, New York, 1953.
- [25] S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002.
- [26] G. da Prato and J. Zabczyk. *Stochastic Equations in Infinite Dimensions*. Cambridge University Press, 2nd edition, 2014.
- [27] R. C. Dalang, D. Khoshnevisan, C. Mueller, D. Nualart, and Y. Xiao. *A Minicourse on Stochastic Partial Differential Equations*. Springer, 2009.

- [28] A. M. Davie and J. G. Gaines. Convergence of numerical schemes for the solution of parabolic stochastic partial differential equations. *Mathematics of Computation*, 70 (233):121–134, 2000.
- [29] P. J. Davis. On the numerical integration of periodic analytic functions. In R. E. Langer, editor, *On Numerical Integration: Proceedings of a Symposium, Madison, 1958*, pages 45–59. University of Wisconsin Press, 1959.
- [30] B. J. Deusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, and O. P. Le Maître. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal on Scientific Computing*, 26(2):698–719, 2004.
- [31] G. Di Nunno, S. Ortiz-Latorre, and A. Petersson. SPDE bridges with observation noise and their spatial approximation. *Stochastic Processes and their Applications*, 158:170–207, 2023.
- [32] M. Dozzi and J. A. López-Mimbela. Finite-time blowup and existence of global positive solutions of a semi-linear SPDE. *Stochastic Processes and their Applications*, 120(6):767–776, 2010.
- [33] N. Dunford and J. T. Schwartz. *Linear operators, part 1: General theory*, volume 10. John Wiley & Sons, 1988.
- [34] C. Eckert, M. Beer, and P. D. Spanos. A polynomial chaos method for arbitrary random inputs using B-splines. *Probabilistic Engineering Mechanics*, 60:103051, 2020.
- [35] M. Eldred and J. Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 976, 2009.
- [36] K.-J. Engel and R. Nagel. One-parameter semigroups for linear evolution equations. In *Semigroup forum*, volume 194. Springer, 2000.
- [37] G. Evans. *Practical Numerical Integration*. Wiley, 1993.
- [38] L. C. Evans. *Partial Differential Equations*, volume 19. American Mathematical Society, 2022.
- [39] F. Flandoli. On the semigroup approach to stochastic evolution equations. *Stochastic Analysis and Applications*, 10(2):181–203, 1992.
- [40] D. L. Gandy and M. R. Nelson. Analyzing pattern formation in the Gray–Scott model: An XPPAUT tutorial. *SIAM Review*, 64(3):728–747, 2022.
- [41] L. Gawarecki and V. Mandrekar. *Stochastic Differential Equations in Infinite Dimensions: With Applications to Stochastic Partial Differential Equations*. Springer Science & Business Media, 2010.
- [42] M. Gerritsma, J.-B. Van der Steen, P. Vos, and G. E. Karniadakis. Time-dependent generalized polynomial chaos. *Journal of Computational Physics*, 229(22):8333–8363, 2010.

- [43] R. Ghanem. Probabilistic characterization of transport in heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 158(3-4):199–220, 1998.
- [44] R. Ghanem and P. D. Spanos. Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, 57(1):197–202, 03 1990.
- [45] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: A spectral approach*. Springer, 1991.
- [46] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230, 1969.
- [47] M. Gradinaru, I. Nourdin, and S. Tindel. Itô’s and Tanaka’s-type formulae for the stochastic heat equation: The linear case. *Journal of Functional Analysis*, 228:114–143, 2005.
- [48] D. S. Grebenkov and B.-T. Nguyen. Geometrical structure of Laplacian eigenfunctions. *SIAM Review*, 55(4), 2013.
- [49] W. Grecksch and P. E. Kloeden. Time-discretised Galerkin approximations of parabolic stochastic PDEs. *Bulletin of the Australian Mathematical Society*, 54:79–85, 1996.
- [50] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.
- [51] J. H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12):701–702, 1964.
- [52] E. Hausenblas. Approximation for semilinear stochastic evolution equations. *Potential Analysis*, 18(2):141–186, 2003.
- [53] E. Hlawka. Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. *Annali di Matematica Pura ed Applicata*, 54(1):325–333, 1961.
- [54] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numerica*, 19: 209–286, 2010.
- [55] J. D. Hoffman and S. Frankel. *Numerical Methods for Engineers and Scientists*. CRC Press, 2018.
- [56] E. P. Hsu. *Stochastic analysis on manifolds*. Number 38. American Mathematical Society, 2002.
- [57] A. Jentzen. Pathwise numerical approximation of SPDEs with additive noise under non-global Lipschitz coefficients. *Potential Analysis*, 31:375–404, 2009.
- [58] A. Jentzen. Higher order pathwise numerical approximations of SPDEs with additive noise. *SIAM Journal on Numerical Analysis*, 49(2):642–667, 2011.
- [59] A. Jentzen and P.E. Kloeden. Overcoming the order barrier in the numerical approximation of stochastic partial differential equations with additive space-time noise. *Proceedings of the Royal Society A*, 465:649–667, 2009.

- [60] A. Jentzen, P. E. Kloeden, and G. Winkel. Efficient simulation of nonlinear parabolic SPDEs with additive noise. *The Annals of Applied Probability*, 21(3):908–950, 2011.
- [61] V. Kaarnioja. Smolyak Quadrature. Master’s thesis, University of Helsinki, 2013.
- [62] A.-K. Kassam. Solving reaction-diffusion equations 10 times faster. *Oxford University, Numerical Analysis Group Research Report No. 16*, 2003.
- [63] A.-K. Kassam and L. N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.
- [64] M. Katsoulakis, G. Kossioris, and O. Lakkis. Noise regularization and computations for the 1-dimensional stochastic Allen-Cahn problem. *Interfaces and Free Boundaries*, 9:1, 11 2011.
- [65] Navjot Kaur and Kavita Goyal. An adaptive wavelet optimized finite difference b-spline polynomial chaos method for random partial differential equations. *Applied Mathematics and Computation*, 415:126738, 2022.
- [66] M. V. Keldysh. On the characteristic values and characteristic functions of certain classes of non-self-adjoint equations. In *Dokl. Akad. Nauk SSSR*, volume 77, pages 11–14, 1951.
- [67] A. Kleefeld. The hot spots conjecture can be false: some numerical examples. *Advances in Computational Mathematics*, 47(6):85, 2021.
- [68] A. Kleefeld and T.-C. Lin. The nonlinear Landweber method applied to an inverse scattering problem for sound-soft obstacles in 3D. *Computer Physics Communications*, 182(12):2550–2560, 2011.
- [69] A. Lang and I. Motschan-Armen. Euler-Maruyama approximations of the stochastic heat equation on the sphere. *arXiv:2307.07564*, 2023.
- [70] O. P. Le Maître, L. Mathelin, O. M. Knio, and M. Y. Hussaini. Asynchronous time integration for polynomial chaos expansion of uncertain periodic dynamics. *Discrete and Continuous Dynamical Systems*, 28(1):199–226, 2010.
- [71] Y. Li and J. A. Yorke. Period three implies chaos. *The American Mathematical Monthly*, 82:985–992, 1975.
- [72] W. Liu and M. Röckner. *Stochastic Partial Differential Equations: An Introduction*. Springer, 2015.
- [73] G. J. Lord and A. Tambue. Stochastic exponential integrators for the finite element discretization of SPDEs for multiplicative and additive noise. *IMA Journal of Numerical Analysis*, 33(2):515–543, 2013.
- [74] L. Lorenzi, A. Lunardi, G. Metafuno, and D. Pallara. *Analytic Semigroups and Reaction-Diffusion Problems*. Internet Seminar, 2005.
- [75] S. V. Lototsky and B. L. Rozovsky. *Stochastic Partial Differential Equations*. Springer, 2017.

- [76] W. Mazin, K. E. Rasmussen, E. Mosekilde, P. Borckmans, and G. Dewel. Pattern formation in the bistable Gray-Scott model. *Mathematics and Computers in Simulation*, 40(3-4):371–396, 1996.
- [77] B. McMillan. Norbert Wiener and chaos. *A Century of Mathematics in America*, pages 479–491, 1989.
- [78] R. Munafo. Pearson’s classification (extended) of Gray-Scott system parameter values. <http://www.mrob.com/pub/comp/xmorphia/pearson-classes.html>, 1996–2023. Accessed: 2023-09-13.
- [79] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41:35–52, 2009.
- [80] A. Nasikun, C. Brandt, and K. Hildebrandt. Fast approximation of Laplace-Beltrami eigenproblems. *Computer Graphics Forum*, 37(5):121–134, 2018.
- [81] H. Niederreiter. Quasi-Monte Carlo methods and pseudo-random numbers. *Bulletin of the American Mathematical Society*, 84(6):957–1041, 1978.
- [82] B. Øksendal. *Stochastic Differential Equations*. Springer, 2000.
- [83] G. Onorato, G. J. A. Loeven, G. Ghorbaniasl, H. Bijl, and C. Lacor. Comparison of intrusive and non-intrusive polynomial chaos methods for CFD applications in aeronautics. In *V European conference on computational fluid dynamics ECCOMAS, Lisbon, Portugal*, pages 14–17, 2010.
- [84] J. E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.
- [85] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd edition: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [86] P. Pušnik. *Strong convergence rates for full-discrete numerical approximations of stochastic partial differential equations with non-globally Lipschitz continuous nonlinearities*. PhD thesis, ETH Zürich, 2020.
- [87] M. T. Reagan, H. N. Najm, R. G. Ghanem, and O. M. Knio. Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection. *Combustion and Flame*, 132(3):545–555, 2003.
- [88] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, Inc., USA, 3rd edition, 1987.
- [89] T. Shardlow. Numerical methods for stochastic parabolic PDEs. *Numerical Functional Analysis and Optimization*, 20(1-2):121–145, 1999.
- [90] E. Sinestrari. On the abstract Cauchy problem of parabolic type in spaces of continuous functions. *Journal of Mathematical Analysis and Applications*, 107:16–66, 1985.
- [91] I. M. Sobol’. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

- [92] J. Son and Y. Du. Comparison of intrusive and nonintrusive polynomial chaos expansion-based approaches for high dimensional parametric uncertainty quantification and propagation. *Computers & Chemical Engineering*, 134:106685, 2020.
- [93] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.
- [94] J. Sun and A. Zhou. *Finite Element Methods for Eigenvalue Problems*. Chapman and Hall/CRC, 2016.
- [95] C. Tudor and W. Grecksch. *Stochastic Evolution Equations: A Hilbert Space Approach*. Walter de Gruyter GmbH, 1995.
- [96] D. A. Voss and A. Q. M. Khaliq. Parallel LOD methods for second order time dependent PDEs. *Computers & Mathematics with Applications*, 30(10):25–35, 1995.
- [97] J. B. Walsh. *An Introduction to Stochastic Partial Differential Equations*. Springer, 1986.
- [98] J. B. Walsh. Finite element methods for parabolic stochastic PDEs. *Potential Analysis*, 23:1–43, 2005.
- [99] X. Wan and G. E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, 2005.
- [100] X. Wan and G. E. Karniadakis. Long-term behavior of polynomial chaos in stochastic flow simulations. *Computer methods in applied mechanics and engineering*, 195 (41-43):5582–5596, 2006.
- [101] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4): 897–936, 1938.
- [102] D. Xiu and G. E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.
- [103] S. Zheng. *Nonlinear Evolution Equations*. CRC Press, 2004.