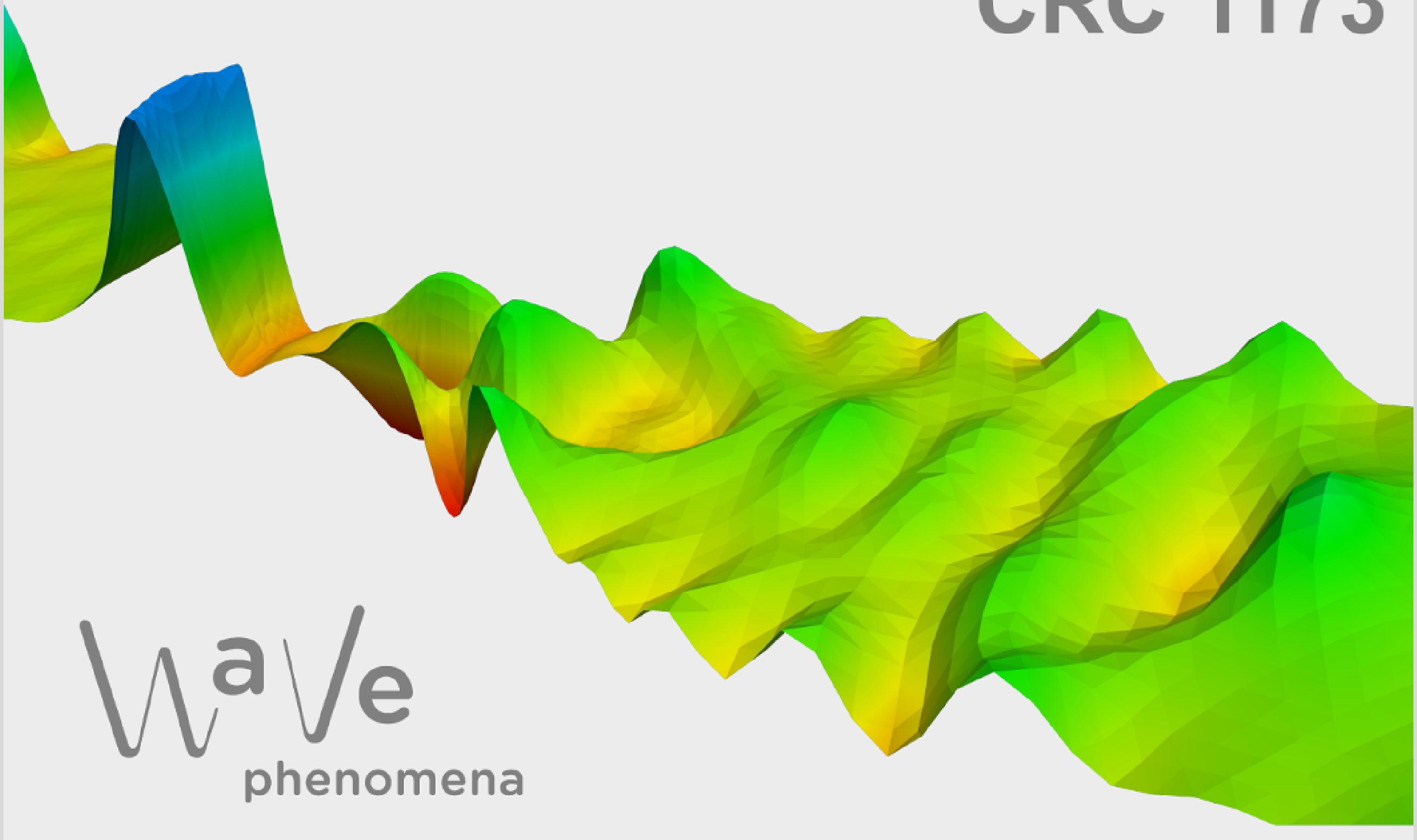
Stochastic Galerkin-collocation splitting for PDEs with random parameters

Tobias Jahnke, Benny Stein

CRC Preprint 2018/28, October 2018



Participating universities

Universität Stuttgart

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Funded by

DFG

ISSN 2365-662X

STOCHASTIC GALERKIN-COLLOCATION SPLITTING FOR PDES WITH RANDOM PARAMETERS

TOBIAS JAHNKE AND BENNY STEIN

Abstract. We propose a numerical method for time-dependent, semilinear partial differential equations (PDEs) with random parameters and random initial data. The method is based on an operator splitting approach. The linear part of the right-hand side is discretized by a stochastic Galerkin method in the stochastic variables and a pseudospectral method in the physical space, whereas the nonlinear part is approximated by a stochastic collocation method in the stochastic variables. In this setting both parts of the random PDE can be propagated very efficiently. The Galerkin method and the collocation method are combined with sparse grids in order to reduce the computational costs. This approach is discussed in detail for the Lugiato-Lefever equation, which serves as a motivating example throughout, but also applies to a much larger class of random PDEs. For such problems our method is computationally much cheaper than standard stochastic Galerkin methods, and numerical tests show that it outperforms standard stochastic collocation methods, too.

Keywords. Uncertainty quantification, splitting methods, spectral methods, (generalized) polynomial chaos, Lugiato-Lefever equation, nonlinear Schrödinger equation, sparse grids, stochastic Galerkin method, stochastic collocation

1. Introduction

Partial differential equations (PDEs) provide well-established models for many processes and phenomena in science and technology. In many real-life applications, however, some part of the information that is required to solve the problem – parameters, coefficient functions, initial or boundary data – is not available or cannot be measured with the desired accuracy. This is the reason why numerical methods for PDEs with *uncertain* or *random* data have received ever-increasing attention in the last decades. Modelling randomness in the data and understanding the influence of this randomness on the solution of the problem are central goals of *uncertainty quantification*. Under this term, a new research area has emerged combining stochastics, analysis, approximation theory, numerical mathematics, with applications in many fields.

The main difficulty is that the solution of such a random[1] PDE is a function which does not only depend on the spatial variable x and (for time-dependent problems) on time t, but also on an additional vector y which represents the randomness of the data. A very popular way to deal with the randomness are *generalized polynomial chaos expansions* (PCE): the solution is expanded in a series of multivariate orthogonal polynomials with respect to y, and the goal is then to approximate the deterministic coefficient functions which only depend on t and x. There are several approaches to do this. Stochastic *collocation* methods rely on solving the same deterministic PDE with different parameters for a number of carefully chosen

Date: October 8, 2018.

We gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) through CRC 1173.

[1]The term random PDE denotes PDEs with random data. This is different from stochastic PDEs where randomness is inherent in the dynamics.

vectors $y^{(j)}$. From these sample solutions an approximation of the unknown coefficients of the truncated PCE can be obtained, e.g., via high-dimensional quadrature formulas. Such schemes are discussed, e.g., in [1, 38, 37, 53]. Stochastic *Galerkin* methods are based on a different idea. The problem is considered in a space spanned by finitely many of the orthonormal polynomials in y, and imposing the classical Galerkin condition leads to a system of PDEs for the coefficient functions. This approach was discussed and analyzed, e.g., in [47, 54, 36, 20, 2]. The resulting system is usually *coupled* and large, and approximating its solution poses a considerable numerical challenge unless only a very small number of polynomials is considered. This is clearly a disadvantage of the stochastic Galerkin method in comparison to the stochastic collocation method where only a number of decoupled PDEs has to be solved, which can be done in parallel. Moreover, handling PDEs with nonlinear terms is cumbersome in the Galerkin ansatz. On the other hand, the number of equations in the Galerkin method is usually noticeably smaller than for the collocation approach if one uses the same polynomial ansatz space; see the discussion below in Section 3.5. A brief comparison of Galerkin and collocation methods can be found in [51, Sec. 6.1]; see also [3] for a more detailed discussion.

The evolution equation which has initiated our research and which serves as a motivating example is the Lugiato-Lefever equation which models the generation of frequency combs in a ring resonator. Such frequency combs are crucial for increasing the data transmission rate through optical fibers. The Lugiato-Lefever equation is a semilinear Schrödinger equation with cubic nonlinearity and additional damping and forcing terms. It involves three parameters which represent dispersion, detuning and forcing. These parameters and also the initial data are uncertain in the sense that they can only be measured up to a limited accuracy. Our main objective is the construction of an efficient numerical method for a class of random PDEs which contains the Lugiato-Lefever equation as one special case.

Our method makes use of the observation that solving only the linear part or only the nonlinear part of the Lugiato-Lefever equation is numerically much easier than a direct approximation of the full equation. In absence of the nonlinear part, the stochastic Galerkin method leads to a system of PDEs which can be *decoupled.* Hence, the linear part of the Lugiato-Lefever equation can be propagated at rather low numerical costs. For the nonlinear part, however, stochastic collocation is much more favorable, because it leads to a set of ordinary differential equations (ODEs) with explicitly known solution. This suggests to approximate the full PDE including both the linear and the nonlinear terms by applying the Strang splitting method, which allows us to treat each of the two parts in its "natural" setting, and to solve it in a very efficient way.

Numerical methods for elliptic PDEs with random coefficients have been proposed and analyzed, e.g., in [12, 13, 47] and many other references. A number of authors have considered parabolic [24, 28] or hyperbolic [37, 22, 34, 26] PDEs. For an overview of uncertainty quantification and corresponding numerical methods, we refer the reader to the books [46, 52, 29, 44]. Splitting methods are very popular for time-integration of PDEs, but it seems that their usefulness for uncertainty quantification has so far not been fully exploited. In [50, 10], operator splitting has been applied to solve hyperbolic PDEs with random coefficients, but in both cases this was done in order to decompose the problem into globally hyperbolic subproblems, whereas our motivation to use the splitting approach is to reduce the computational costs.

In Section 2 we introduce the Lugiato-Lefever equation as a motivating model problem. Our method does not only apply to this PDE, but to a more general problem class which is specified in Section 3.1. In Sections 3.2–3.5, we briefly review important building blocks for our approach, namely polynomial chaos expansions, stochastic Galerkin and collocation methods, and sparse grids. The new splitting method is presented in Section 4. Numerical experiments are shown in Section 5. In these examples the new splitting method turns out to be more efficient than a pure stochastic collocation method, which underlines the advantage of our approach.

2. A motivating example: The Lugiato-Lefever equation

2.1. **The Lugiato-Lefever equation with deterministic data.** The Lugiato-Lefever equation

$$\partial_t u(t,x) = \mathrm{i}a\partial_x^2 u(t,x) - (1+\mathrm{i}b)u(t,x) + f + \mathrm{i}|u(t,x)|^2 u(t,x), \qquad t \geq 0, \quad x \in \mathbb{T}, \tag{1a}$$

$$u(0,x) = u_0(x) \tag{1b}$$

is a cubic semilinear Schrödinger equation on the one-dimensional torus $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$ with additional damping, detuning and forcing terms; cf. [33]. The quantities a, b and f are real-valued parameters. The Lugiato-Lefever equation has been proposed as a model for the emergence of frequency combs in a ring resonator; cf. [9, 23]. The damping term $-u$ in (1a) describes radiation into an optical waveguide to which the ring resonator is coupled. The forcing term f represents an external pump which is tuned to a resonance wavelength. If the parameters a, b and f and the initial data in (1b) are chosen in a suitable way, then the solution of (1a) converges to a soliton-like steady state $u_\star(x) = \lim_{t\to\infty} u(t,x)$ with Fourier coefficients

$$\hat{u}_k = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{T}} \mathrm{e}^{-\mathrm{i}kx} u_\star(x)\,\mathrm{d}x, \qquad k \in \mathbb{Z}. \tag{2}$$

Then, the graph of $k \mapsto \log(|\hat{u}_k|)$ has a particular structure which is called a frequency comb, see Figure 3A in Section 5 for an example. Understanding this effect via numerical simulations is important for signal processing, because the frequency combs generated by such a ring resonator can be used as optical sources for high-speed data transmission. The central probem is that technically suitable frequency combs occur only for special choices of the parameters a, b and f (cf. [35]), and that for a large set of initial data the solution converges to a steady state which is spatially constant and hence technically useless.

Before we proceed to the case of random data we briefly sketch how the Lugiato-Lefever equation can be solved numerically when the parameters and the initial data are *known.* Splitting methods are particularly attractive for this task. These time-stepping methods approximate the solution of (1) by solving the three PDEs

$$(S_1) \quad \partial_t v = \mathrm{i}a\partial_x^2 v, \qquad (S_2) \quad \partial_t v = -(1+\mathrm{i}b)v + f \qquad \text{and} \qquad (S_3) \quad \partial_t v = \mathrm{i}|v|^2 v \tag{3}$$

one after the other in each time-step. The advantage of such a decomposition is that each of the subproblems can be solved very efficiently. For every time interval $[t_0, t_{\mathrm{end}}]$ the solution of (S_1) with given initial data in $L^2(\mathbb{T})$ can be computed via the Fourier transform. For numerical computations the infinite Fourier series has to be replaced by a finite sum, and the Fourier transform is replaced by the discrete fast Fourier transform. Strictly speaking, this yields only an approximation of the solution of (S_1) in finitely many grid points, but the error of this approximation is well understood and can be made arbitrarily small, see e.g. [32, Thm. III.1.8]. The subproblems (S_2) and (S_3) do not involve any spatial derivatives and are thus equivalent to ODEs in each point x. The exact solution of (S_2) is given by the variation-of-constants formula and the exact solution of (S_3) is explicitly stated in equation (37) below.

For $j \in \{1,2,3\}$ let $\Phi_t^{(j)}$ be the flow of the differential equation (S_j), i.e. $v(t) = \Phi_{t-t_0}^{(j)}(v_0)$ is the solution of (S_j) at time $t \geq t_0$ with initial condition $v(t_0) = v_0$. Then, the Strang splitting with step-size $\tau > 0$ is defined by

$$u_{n+1} = \Phi^{(1)}_{\tau/2} \circ \Phi^{(2)}_{\tau/2} \circ \Phi^{(3)}_{\tau} \circ \Phi^{(2)}_{\tau/2} \circ \Phi^{(1)}_{\tau/2}(u_n), \qquad n = 0, 1, 2, \ldots$$

This method provides computationally cheap (second-order in time) approximations $u_n \approx u(t_n, \cdot)$ at times $t_n = n\tau$. A Strang splitting scheme for the Lugiato-Lefever equation was analyzed in detail in [25]. In this reference, the right-hand side of the PDE was split into two instead of three parts, but we will see in Section 4 that splitting into three parts is more favorable when the coefficients are random.

2.2. **The Lugiato-Lefever equation with random data.** Henceforth we consider the situation that the parameters a, b and f in (1a) are not given exactly but only up to some uncertainty. To account for the randomness we introduce a new variable

$$y = (y_1, \ldots, y_d) \in \Gamma^{(1)} \times \cdots \times \Gamma^{(d)} = \Gamma, \qquad y_i \in \Gamma^{(i)} \subseteq \mathbb{R}$$

on a suitable subset $\Gamma \subseteq \mathbb{R}^d$ and suppose that y is the realization of a random variable $Y\colon \Omega \to \Gamma$ on a probability space $(\Omega, \Sigma, \mathbb{P})$. We assume that the parameters $a = a(y)$, $b = b(y)$, $f = f(y)$ and the initial data $u_0 = u_0(x, y)$ depend on y in a known, deterministic way. Hence, instead of (1) we have to solve the random PDE

$$\partial_t u(t,x,y) = \mathrm{i}a(y)\partial_x^2 u(t,x,y) - (1 + \mathrm{i}b(y))u(t,x,y) + f(y) + \mathrm{i}|u(t,x,y)|^2 u(t,x,y), \tag{4a}$$

$$u(0,x,y) = u_0(x,y). \tag{4b}$$

For every *given* realization $y = Y(\omega)$, $\omega \in \Omega$, the problem (4) can be solved in the same way as the original problem (1), but our goal is to solve (4) when only the probability density of Y is known.

The numerical method which will be constructed in Section 4 is not at all restricted to the Lugiato-Lefever equation (4). Our approach can be applied to all PDEs of a certain type which will be specified in the following section. We return to the special case of the Lugiato-Lefever equation later in Sections 4 and 5.

3. Problem setting and classical UQ methods

3.1. **Problem setting.** We consider time-dependent, semilinear evolution equations of the form

$$\partial_t u(t,x,y) = \alpha(y)\mathcal{A}u(t,x,y) + \beta(y)\mathcal{B}u(t,x,y) + \gamma(y)g(x) + F(u(t,x,y)), \tag{5a}$$

$$u(0,x,y) = u_0(x,y), \tag{5b}$$

for $t \geq 0$, $x \in S = \mathbb{T}^N$ and $y = (y_1, \ldots, y_d) \in \Gamma^{(1)} \times \cdots \times \Gamma^{(d)} = \Gamma \subseteq \mathbb{R}^d$. Henceforth, the following assumptions are made. The operator $\mathcal{A}\colon L^2(S) \supseteq \mathrm{D}(\mathcal{A}) \to L^2(S)$ generates a strongly continuous semigroup $(\mathrm{e}^{t\mathcal{A}})_{t\geq 0}$ and $\mathcal{B}\colon L^2(S) \to L^2(S)$ is a bounded operator which maps $\mathrm{D}(\mathcal{A})$ into itself. Both operators act on $u(t, \cdot, y)$ for fixed t and y. The nonlinearity $F\colon \mathrm{D}(\mathcal{A}) \to \mathrm{D}(\mathcal{A})$ is locally Lipschitz continuous. For every $y \in \Gamma$ the parameters $\alpha(y)$ and $\gamma(y)$ are real scalars, and if $\mathcal{A}$ does not generate a group, then $\alpha(y)$ must be positive. The parameter $\beta(y)$ can be written as $\beta(y) = \beta_0 + \beta_1(y)$ with $\beta_1(y) \in \mathbb{R}$ and a constant $\beta_0 \in \mathbb{C}$. In most applications, β_0 is simply zero such that $\beta(y)$ is real, but a nonzero β_0 appears, for example, in the Lugiato-Lefever equation. The forcing term and the initial data have the regularity $g \in \mathrm{D}(\mathcal{A})$ and $u_0(\cdot, y) \in \mathrm{D}(\mathcal{A})$ for every $y \in \Gamma$. These assumptions imply local wellposedness of problem (5) in the classical sense for each $y \in \Gamma$, i.e. there exists $T = T(y) > 0$ and

$$u(\cdot, \cdot, y) \in C([0,T), \mathrm{D}(\mathcal{A})) \cap C^1([0,T), L^2(S)) \tag{6}$$

which solves (5). This follows from [42, Thm. 6.1.7] and the remark thereafter.

A crucial assumption for our approach is that for known constants $\alpha \in \mathbb{R}$, $\beta \in \mathbb{C}$ and $\gamma \in \mathbb{R}$ each of the three deterministic subproblems

$$\partial_t v(t,x) = \alpha \mathcal{A} v(t,x), \qquad v(t_0,\cdot) \in \mathrm{D}(\mathcal{A}), \tag{7a}$$

$$\partial_t v(t,x) = \beta \mathcal{B} v(t,x) + \gamma g(x), \qquad v(t_0,\cdot) \in L^2(S), \tag{7b}$$

$$\partial_t v(t,x) = F(v(t,x)), \qquad v(t_0,\cdot) \in \mathrm{D}(\mathcal{A}) \tag{7c}$$

can be solved numerically at low costs for $t \geq t_0$ (again with $\alpha > 0$ if $\mathcal{A}$ does not generate a group). This assumption seems to be somewhat restrictive, but actually there are many applications where this is indeed true. One example is the linear Schrödinger equation, which is obtained for

$$\mathrm{D}(\mathcal{A}) = H^2(S), \qquad \mathcal{A}u = \mathrm{i}\Delta u, \qquad \mathcal{B}u = Vu, \qquad g \equiv 0, \qquad F(u) \equiv 0,$$

with a bounded potential $V : S \to \mathbb{R}$ and $\beta(y) = \beta_1(y) \in \mathbb{R}$. Another example is the cubic nonlinear Schrödinger equation where

$$\mathrm{D}(\mathcal{A}) = H^s(S), \qquad \mathcal{A}u = \mathrm{i}\Delta u, \qquad \mathcal{B}u = 0, \qquad g \equiv 0, \qquad F(u) = \mathrm{i}\nu|u|^2 u$$

with $s > \frac{N}{2}$ and $\nu \in \mathbb{R}$. The Lugiato-Lefever equation (4) is recovered for $N = 1$ and

$$\mathrm{D}(\mathcal{A}) = H^2(S), \qquad \mathcal{A}u = \mathrm{i}\partial_x^2 u, \qquad \mathcal{B}u = \mathrm{i}u, \qquad g \equiv 1, \qquad F(u) = \mathrm{i}|u|^2 u$$

if we set $\alpha(y) = a(y)$, $\beta(y) = \mathrm{i} - b(y)$, and $\gamma(y) = f(y)$ with $a(y),\ b(y),\ f(y) \in \mathbb{R}$ from (4). Yet another example are reaction-diffusion equations such as, e.g.,

$$\mathrm{D}(\mathcal{A}) = H^2(S), \qquad \mathcal{A}u = \partial_x^2 u, \qquad \mathcal{B}u = -u,$$

with suitable functions F, g and real $\beta(y) = \beta_1(y) \in \mathbb{R}$. In all these examples, the subproblems involving spatial derivatives can be solved via the Fourier transform because $S = \mathbb{T}^N$. We remark, however, that the restriction to the N-dimensional torus $S = \mathbb{T}^N$ is made in order to keep the exposition simple. Other domains and boundary conditions can also be handled by our method as long as the assumptions specified above are fulfilled. The requirement that $\mathcal{B}$ is bounded may also be weakened, but then it is more difficult to keep track of the correct function spaces for each of the subproblems. Our method can also be applied in cases where the nonlinear operator F depends explicitly on y as long as the corresponding subproblem can be solved efficiently.

As before, y is interpreted as a realization of a vector-valued random variable $Y \colon \Omega \to \Gamma$, $Y = (Y_1, \ldots, Y_d)$, i.e. $y = Y(\omega)$ for some $\omega \in \Omega$, where $(\Omega, \Sigma, \mathbb{P})$ is a probability space. Throughout it is assumed that $Y_1, \ldots, Y_d$ are stochastically independent, and that each Y_i has a known probability density function $\rho^{(i)} : \Gamma^{(i)} \to \mathbb{R}$. Typical choices are $\rho^{(i)}(y_i) = \frac{1}{2}$ on $[-1,1]$ for uniformly distributed Y_i and $\rho^{(i)}(y_i) = \frac{1}{\sqrt{2\pi}} \exp(-y_i^2/2)$ on $(-\infty, \infty)$ for normally distributed Y_i.

If one is only interested in some deterministic quantity of interest (such as, e.g., the expected value or variance) of the solution, one may solve (5) for a huge number of vectors $y^{(j)} \in \Gamma$, $j = 1, \ldots, Q$ sampled from the distribution of Y. Afterwards, one may compute the quantity of interest via the standard *Monte Carlo* estimators. The convergence rate of the naive Monte Carlo method is known to be very slow. Although improvements such as multi-level or quasi-Monte Carlo methods are available [21, 4, 8], we use a Monte Carlo method only for the purpose of computing reference solutions.

3.2. **Generalized polynomial chaos expansions.** From now on, we are only interested in random variables which have finite second moments. For $i \in \{1, \ldots, d\}$ let

$$L^2_{\rho^{(i)}}(\Gamma^{(i)}) = \left\{ v\colon \Gamma^{(i)} \to \mathbb{C} \;\middle|\; \|v\|^2_{\rho^{(i)}} < \infty \right\}$$

be the Hilbert space of measurable, square-integrable functions on $\Gamma^{(i)}$ with norm $\|v\|^2_{\rho^{(i)}} = \langle v, v\rangle_{\rho^{(i)}}$ induced by the weighted inner product

$$\langle v, w\rangle_{\rho^{(i)}} := \int\limits_{\Gamma^{(i)}} v(y_i)\overline{w(y_i)}\rho^{(i)}(y_i)\mathrm{d}y_i. \tag{8}$$

As usual, two functions are identified if they coincide outside of a set of Lebesgue measure zero. Since $Y_1, \ldots, Y_d$ are independent by assumption, the probability density function of Y is given by the product

$$\rho(y) = (\rho^{(1)} \otimes \cdots \otimes \rho^{(d)})(y) = \prod_{i=1}^{d} \rho^{(i)}(y_i), \qquad y = (y_1, \ldots, y_d).$$

The space $L^2_\rho(\Gamma) = \left\{ v\colon \Gamma \to \mathbb{C} \;\middle|\; \|v\|_\rho < \infty \right\}$ with norm $\|\cdot\|_\rho$ induced by the inner product

$$\langle v, w\rangle_\rho := \int\limits_{\Gamma} v(y)\overline{w(y)}\rho(y)\mathrm{d}y$$

is again a Hilbert space. Henceforth we assume that the solutions from (6) satisfy $T := \inf_{y\in\Gamma} T(y) > 0$ and that $y \mapsto u(t, x, y)$ belongs to $L^2_\rho(\Gamma)$ for fixed values of $t \in [0, T)$ and $x \in S$.

For $i = 1, \ldots, d$ let $(\phi^{(i)}_j)_{j\in\mathbb{N}_0}$ be a complete set of real-valued orthonormal polynomials on $L^2_{\rho^{(i)}}(\Gamma^{(i)})$ with the properties $\deg(\phi^{(i)}_j) = j$ and $\phi^{(i)}_0 \equiv 1$. Such polynomials can be computed efficiently by the usual three-term recursions. Multivariate orthonormal polynomials in $L^2_\rho(\Gamma)$ can be constructed via tensorisation: If we let

$$\phi_{\mathbf{k}}(y) = (\phi^{(1)}_{k_1} \otimes \cdots \otimes \phi^{(d)}_{k_d})(y) = \prod_{i=1}^{d} \phi^{(i)}_{k_i}(y_i), \qquad y = (y_1, \ldots, y_d) \tag{9}$$

for a multi-index $\mathbf{k} = (k_1, \ldots, k_d) \in \mathbb{N}^d_0$, then by construction we have $\langle \phi_{\mathbf{j}}, \phi_{\mathbf{k}}\rangle_\rho = \delta_{\mathbf{jk}}$ for $\mathbf{j}, \mathbf{k} \in \mathbb{N}^d_0$, where $\delta_{\mathbf{jk}}$ is the Kronecker delta. Now we expand the solution u of (6) in a generalized polynomial chaos expansion (PCE)

$$u(t, x, y) = \sum_{\mathbf{k}\in\mathbb{N}^d_0} u_{\mathbf{k}}(t, x)\phi_{\mathbf{k}}(y), \qquad u_{\mathbf{k}}(t, x) = \langle u(t, x, \cdot), \phi_{\mathbf{k}}\rangle_\rho, \tag{10}$$

where $\{\phi_{\mathbf{k}} \mid \mathbf{k} \in \mathbb{N}^d_0\}$ is a complete set of orthogonal polynomials in $L^2_\rho(\Gamma)$. The equality (10) has to be understood in the usual sense that for a.e. $x \in S$ and $t \geq 0$, the left-hand side is the limit of the series on the right-hand side w.r.t. $\|\cdot\|_\rho$. This expansion was introduced by Xiu and Karniadakis in [55] as an extension to Wiener's classical polynomial chaos expansion from [49]. The convergence of generalized PCEs can be established in many cases of interest, since criteria are available which can be verified for most of the usual distributions encountered in practice, see [16]. From the PCE (10) the expectation and the variance of $\omega \mapsto u(t, x, Y(\omega))$

can easily be computed: Setting $\mathbf{0} = (0,\dots,0) \in \mathbb{N}_0^d$, we arrive at

$$\mathbb{E}[u(t,x,Y)] = \int_\Gamma u(t,x,y)\rho(y)dy = \sum_{\mathbf{k}\in\mathbb{N}_0^d} u_{\mathbf{k}}(t,x) \int_\Gamma \phi_{\mathbf{k}}(y)\rho(y)dy = u_{\mathbf{0}}(t,x)$$

and

$$\mathbb{V}[u(t,x,Y)] = \int_\Gamma \Big| \sum_{\mathbf{k}\in\mathbb{N}_0^d\setminus\{\mathbf{0}\}} u_{\mathbf{k}}(t,x)\phi_{\mathbf{k}}(y)\Big|^2 \rho(y)dy = \sum_{\mathbf{k}\in\mathbb{N}_0^d\setminus\{\mathbf{0}\}} |u_{\mathbf{k}}(t,x)|^2$$

due to $u_{\mathbf{0}}(t,x) = \phi_{\mathbf{0}}(y)u_{\mathbf{0}}(t,x)$ and

$$\int_\Gamma \phi_{\mathbf{k}}(y)\rho(y)\mathrm{d}y = \langle \phi_{\mathbf{k}}, \phi_{\mathbf{0}}\rangle_\rho = \delta_{\mathbf{k}\mathbf{0}}.$$

From (10), one can also derive similar formulas for higher-order moments and other statistical quantities of interest, such as the covariance function and the global sensitivity coefficients, see e.g. [51].

Since only finitely many terms can be computed in practice, we consider the truncated PCE

$$\sum_{\phi\in\Pi} u_\phi(t,x)\phi(y) \approx u(t,x,y), \qquad u_\phi(t,x) = \langle u(t,x,\cdot), \phi\rangle_\rho. \tag{11}$$

Here, Π is a set of multivariate orthogonal polynomials which corresponds to a finite set of multi-indices in $\mathbb{N}_0^d$ which is *admissible.* We adopt this notion from [14, Sec. 2.2]. A subset $\mathcal{K} \subseteq \mathbb{N}_0^d$ is called admissible if it is finite and for each $\mathbf{k} \in \mathcal{K}$ the backward neighborhood of $\mathbf{k}$ belongs to $\mathcal{K}$. The backward neighborhood of a multi-index $\mathbf{k} = (k_1,\dots,k_d) \in \mathbb{N}_0^d$ is the set $\{\mathbf{k} - \mathbf{e}_i \mid i = 1,\dots,d \text{ with } k_i \geq 1\}$, where $\mathbf{e}_i$ is the i-th canonical unit vector in $\mathbb{R}^d$. Typical choices for the set Π are:

(i) All polynomials up to a given degree p in all single variables, i.e. $\Pi_p^f = \{\phi_{\mathbf{k}} \mid |\mathbf{k}|_\infty \leq p\}$.
(ii) All polynomials up to a given total degree q, i.e. $\Pi_q^t = \{\phi_{\mathbf{k}} \mid |\mathbf{k}|_1 \leq q\}$.

We use the more flexible description with general Π since it allows us to include other polynomial sets which occur as *half-exact sets* of sparse grid quadrature formulas, see Section 3.5 or [14] for details.

Next, we explain two well-established classes of methods which rely on the expansion (11), namely stochastic Galerkin and stochastic collocation methods. In principle, both approaches may be used to approximate the solution of (5), and an ansatz space of the form span(Π) can be used for the discretization in y. We use both methods as building blocks for the splitting method introduced in Section 4 later on.

3.3. **The stochastic Galerkin approach.**

We aim to approximate the coefficient functions $u_\phi(t,x)$ from (11) by

$$\widetilde{u}_\phi(t,x) \approx u_\phi(t,x), \qquad \widetilde{u}(t,x,y) = \sum_{\phi\in\Pi} \widetilde{u}_\phi(t,x)\phi(y) \approx u(t,x,y)$$

such that for all $t \geq 0$, $x \in S$ and $\phi \in \Pi$ the Galerkin condition

$$\begin{aligned}\langle \partial_t \widetilde{u}(t,x,\cdot), \phi\rangle_\rho = \langle \alpha \sum_{\psi\in\Pi} \mathcal{A}\widetilde{u}_\psi(t,x)\psi, \phi\rangle_\rho + \langle \beta \sum_{\psi\in\Pi} \mathcal{B}u_\psi(t,x)\psi, \phi\rangle_\rho \\ + g(x)\langle \gamma, \phi\rangle_\rho + \langle F(\widetilde{u}(t,x,\cdot)), \phi\rangle_\rho\end{aligned} \tag{12}$$

holds. The left-hand side of (12) reduces to

$$\langle \partial_t \widetilde{u}(t,x,\cdot), \phi\rangle_\rho = \sum_{\psi\in\Pi} \partial_t \widetilde{u}_\psi(t,x)\langle \psi,\phi\rangle_\rho = \partial_t \widetilde{u}_\phi(t,x).$$

Together with the right-hand side of (12), we arrive at

$$\begin{aligned} \partial_t \widetilde{u}_\phi(t,x) &= \sum_{\psi\in\Pi} \mathcal{A}\widetilde{u}_\psi(t,x)\langle \alpha\psi,\phi\rangle_\rho + \sum_{\psi\in\Pi} \mathcal{B}u_\psi(t,x)\langle \beta\psi,\phi\rangle_\rho \\ &\quad + g(x)\langle \gamma,\phi\rangle_\rho + \langle F(\widetilde{u}(t,x,\cdot)),\phi\rangle_\rho. \end{aligned} \tag{13}$$

In order to obtain a system of differential equations for the coefficients $\widetilde{u}_\phi(t,x)$, the function $\widetilde{u}(t,x,\cdot)$ in the last term has to be substituted by (10). This leads to a complicated expression which, in general, cannot be simplified. In the special case of the LLE with $F(u) = \mathrm{i}|u|^2u$, one would obtain

$$\mathrm{i}\langle |\widetilde{u}|^2\widetilde{u},\phi\rangle_\rho = \mathrm{i}\sum_{\psi_1\in\Pi}\sum_{\psi_2\in\Pi}\sum_{\psi_3\in\Pi} \widetilde{u}_{\psi_1}\overline{\widetilde{u}_{\psi_2}}\widetilde{u}_{\psi_3}\langle \psi_1\psi_2\psi_3,\phi\rangle_\rho$$

for the last term in (13). Hence, each $\phi\in\Pi$ would require the computation of a triple sum. The computational costs of this operation are far too expensive in almost every setting. This is a crucial disadvantage of the standard Galerkin approach. The method which we present later in Section 4 does not have this downside.

Suppose for a moment that the last term on the right-hand side of (13) can be computed in some way or the other. Now we switch from polynomials ϕ and ψ to single indices j, k to enumerate the set Π. Formally, we set $P = \mathrm{card}(\Pi)$ and choose a bijection $\eta\colon \{1,\dots,P\}\to\Pi$. Then we set $\widetilde{u}_j(t,x) := \widetilde{u}_{\eta(j)}(t,x)$ for all $j = 1,\dots,P$. The coefficient vector $\widetilde{\mathbf{u}}(t,x) := \left(\widetilde{u}_j(t,x)\right)_j$ is now the solution of

$$\partial_t\widetilde{\mathbf{u}}(t,x) = \mathbf{M}_\alpha\mathcal{A}\widetilde{\mathbf{u}}(t,x) + \mathbf{M}_\beta\mathcal{B}\widetilde{\mathbf{u}}(t,x) + g(x)\mathbf{V}_\gamma + \mathbf{V}_F(\widetilde{\mathbf{u}}(t,x)) \tag{14}$$

with matrices and vectors

$$\mathbf{M}_\alpha = \left(\langle \alpha\phi_j,\phi_k\rangle_\rho\right)_{j,k=1}^P, \qquad \mathbf{M}_\beta = \left(\langle \beta\phi_j,\phi_k\rangle_\rho\right)_{j,k=1}^P, \tag{15a}$$

$$\begin{aligned} \mathbf{V}_\gamma &= \left(\langle \gamma,\phi_j\rangle\right)_{j=1}^P, & \mathbf{V}_F(\widetilde{\mathbf{u}}(t,x)) &= \left(\langle F(\widetilde{u}(t,x,\cdot)),\phi_j\rangle_\rho\right)_{j=1}^P, \\ \mathcal{A}\widetilde{\mathbf{u}}(t,x) &= \left(\mathcal{A}\widetilde{u}_j(t,x)\right)_{j=1}^P, & \mathcal{B}\widetilde{\mathbf{u}}(t,x) &= \left(\mathcal{B}\widetilde{u}_j(t,x)\right)_{j=1}^P. \end{aligned} \tag{15b}$$

All of the quantities in (15a) and (15b) contain certain inner products, which have to be approximated by a quadrature rule. The same holds for the initial values $\widetilde{\mathbf{u}}(0,x) = \left(\widetilde{u}_j(0,x)\right)_{j=1}^P$, which are computed by projecting the given initial value u_0 from (5b) into the ansatz space $\mathrm{span}(\Pi)$. The quadrature rule should provide the exact matrices $\mathbf{M}_\alpha$ and $\mathbf{M}_\beta$ at least if α and β are constant, which means that all the products $\phi\psi$ for ϕ, $\psi\in\Pi$ should be integrated exactly. Therefore the quadrature formula should depend on the set Π. If the quadrature rule is *not* a tensor product of one-dimensional quadrature formulas, then the two types of errors involved so far, namely truncation and quadrature errors, influence each other. This leads to a phenomenon called *internal aliasing*, which produces wildly inaccurate approximations in most cases. As shown in [15], this can be avoided if the polynomials in Π and the quadrature rule are chosen in a special way and not independently of each other. Although we do not explain here *why* the phenomenon of internal aliasing occurs when quadrature is applied in a naive way, we point out that in this paper we always choose the quadrature rule in such a way that internal aliasing does *not* occur. Details will be explained in Section 4.

Equation (14) is a coupled system of P time-dependent nonlinear PDEs on $S = \mathbb{T}^N$. After a space discretization for (14) using a grid with M^N points, every function $\widetilde{u}_j$ is described by a vector of M^N function values. For the classical polynomial sets Π_K^f and Π_K^t, we have $P = (K+1)^d$ and $P = \binom{K+d}{d}$, respectively. One then obtains ODE systems of size $(K+1)^d M^N$ and $\binom{K+d}{d} M^N$. Already for relatively small values of K, these systems are very large even in cases where the dimension N of the "physical" domain $S = \mathbb{T}^N$ is rather small, say $N = 1$ or $N = 2$. Note that P grows with d and therefore depends on the stochastic dimension, too.

We conclude that a naive stochastic Galerkin approach is often not feasible in practice for the Lugiato-Lefever equation (4) or other problems of the type (5). This is due to the nonlinearity, which does not really fit into the Galerkin approach, and the large size of the resulting discrete system.

3.4. **The stochastic collocation approach.** The starting point for stochastic collocation methods is again the PCE from (10) and its truncated version (11). Instead of deriving a coupled system of PDEs as in the Galerkin method, we choose a number of collocation points $y^{(1)}, \dots, y^{(Q)} \in \Gamma$ and compute numerical approximations $v_j(t,x) \approx u(t,x,y^{(j)})$, at least for discrete values of t and x. Then, we have to determine the coefficients $u_\phi(t,x)$ in such a way that the truncated PCE (11) corresponds in some sense to the sample solutions $v_j(t,x) \approx u(t,x,y^{(j)})$. This can be done via interpolation or via a (pseudo-)spectral approximation procedure. We prefer the latter, since the choice of nodes and their manipulation is more cumbersome for Lagrange interpolation, as e.g. explained in [52, Section 7.2].

Let $\phi_1, \dots, \phi_P$ be an enumeration of the elements of Π. *Spectral approximation* uses evaluations of a finite PCE

$$\widetilde{u}(t,x,y) = \sum_{p=1}^{P} \widetilde{u}_p(t,x)\phi_p(y)$$

in the collocation points $y^{(1)}, \dots, y^{(Q)}$ to obtain a (possibly overdetermined) linear system

$$(\widetilde{u}(t,x,y^{(j)}))_{j=1}^{Q} = \Psi \cdot (\widetilde{u}_p(t,x))_{p=1}^{P}, \qquad \Psi = (\Psi_{q,p}) \in \mathbb{R}^{Q\times P}, \qquad \Psi_{q,p} = \phi_p(y^{(q)}).$$

After substituting $\widetilde{u}(t,x,y^{(j)})$ by the numerical approximations $v_j(t,x)$ the corresponding least-squares problem has to be solved for each discrete value of t and x. The matrix Ψ does not depend on t or x and is thus the same in each of these least-squares problems.

Pseudospectral approximation uses a different approach. Here, the integral in

$$u_\phi(t,x) = \langle u(t,x,\cdot), \phi\rangle_\rho = \int_\Gamma u(t,x,y)\phi(y)\rho(y)\mathrm{d}y$$

is approximated by a quadrature rule

$$\int_\Gamma u(t,x,y)\phi(y)\rho(y)\mathrm{d}y \approx \sum_{j=1}^{Q} u(t,x,y^{(j)})\phi(y^{(j)})\omega^{(j)}, \quad \phi \in \Pi,$$

with nodes $y^{(1)}, \dots, y^{(Q)}$ and weights $\omega^{(1)}, \dots, \omega^{(Q)}$. Hence, the collocation points where the sample solutions are computed must be chosen to be the nodes of a suitable quadrature formula. Finally, substituting again $v_j(t,x)$ for the unknown solution $u(t,x,y^{(j)})$ yields the approximation

$$u_\phi(t,x) \approx \sum_{j=1}^{Q} v_j(t,x)\phi(y^{(j)})\omega^{(j)}, \quad \phi \in \Pi.$$

In contrast to the stochastic Galerkin method, where a *coupled* system of PDEs has to be solved, the approximations $v_j(t,x) \approx u(t,x,y^{(j)})$ are independent from each other and can be computed in parallel. Moreover, a standard method can be used for this task because each $v_j(t,x)$ is the numerical solution of a PDE with *known* data. These are important advantages of the stochastic collocation method.

It is preferable to take a quadrature rule of high order to obtain a sufficiently accurate approximation. The number Q of collocation points, however, is correlated with the dimension d of the stochastic space, and in applications with large values of d, it is unavoidable to use a *sparse grid* quadrature formula. To avoid internal aliasing, the set Π and the quadrature rule are chosen appropriately, as for the Galerkin approach. This topic is covered in detail in the next subsections.

3.5. **Sparse grids.** Full tensor grids in d dimensions involve so many grid points that numerical computations are too expensive unless d is very small. For this reason sparse grids are often unavoidable both for stochastic Galerkin and for collocation methods. For an introduction to sparse grids the reader is referred to [19, 48, 40, 39, 6]. What we present in this subsection is not new but crucial for understanding the efficiency of the Galerkin-collocation splitting method which will be constructed in Section 4.

Let us begin by introducing notation, starting with the one-dimensional case. Afterwards, we describe full grids and then how one obtains sparse grids. Let $i \in \{1,\dots,d\}$ be a (fixed) dimension. For $f\colon \Gamma^{(i)} \to \mathbb{C}$ we define a quadrature formula of a certain *level* $m \in \mathbb{N}$ with $p^{(i)}(m) \in \mathbb{N}$ nodes

$$\mathcal{Q}_m^{(i)}(f) = \sum_{j=1}^{p^{(i)}(m)} \omega_j^{(i)} f(y_j^{(i)}) \approx \int_{\Gamma^{(i)}} f(y^{(i)})\rho^{(i)}(y^{(i)})\mathrm{d}y^{(i)}.$$

Here, $(y_j^{(i)})_{j=1}^{p^{(i)}(m)}$ are the nodes and $(\omega_j^{(i)})_{j=1}^{p^{(i)}(m)}$ are the weights. (Actually, nodes and weights should also have an index m, which we omit here in order to keep the notation as simple as possible.) The product formula $\mathcal{Q}_m^{(i)}$ has *degree of exactness*[2] $q^{(i)}(m) \in \mathbb{N}$ if $\mathcal{Q}_m^{(i)}$ is exact for all polynomials whose degree in $y^{(i)}$ is not larger than $q^{(i)}(m)$. We assume that the degree of exactness of $\mathcal{Q}_m^{(i)}$ and thus also the number of nodes $p^{(i)}(m)$ increases with m. The function $m \mapsto p^{(i)}(m)$ is called the *growth rule*. Typical growth rules are

$$p_{\text{lin}}(m) = m \qquad \text{and} \qquad p_{\text{exp}}(m) = \begin{cases} 2^{m-1}+1, & \text{for } m \neq 1, \\ 1, & \text{for } m = 1. \end{cases} \tag{16}$$

Another growth rule which appears later on is[3]

$$p_{\text{slow}}(m) = 2\left\lceil \tfrac{m+1}{2} \right\rceil - 1. \tag{17}$$

Here, $\lceil x \rceil$ denotes the least integer greater than or equal to x.

Let us briefly explain why one may be interested in other growth rules than p_{lin}. Consider a one-dimensional Newton-Cotes formula with M equidistant nodes $y_j = \frac{j}{M-1}$, $j = 0,\dots,M-1$, on the interval $[0,1]$. If we choose the number of nodes according to p_{exp}, then $p_{\text{exp}}(m+1) = 2^m + 1 = 2\cdot(2^{m-1}+1) - 1 = 2\cdot p_{\text{exp}}(m) - 1$. Hence, the quadrature nodes of level m are completely contained in the nodes for level $m+1$, and one obtains a *nested* family of quadrature rules. This is beneficial for adaptive quadrature or – as we will see soon – in the

[2]The degree of exactness is not to be confused with the *order* of a quadrature rule. By definition, a one-dimensional quadrature rule with degree of exactness q has the order $q+1$.

[3]The slow growth rule p_{slow} for, e.g., Gauss quadrature formulas was suggested by Burkardt and Webster in [27] as a simple nesting improvement for the standard Gauss rules.

construction of sparse grids. For p_{lin}, however, the family is not nested. The growth rule for a fully-nested family of Clenshaw-Curtis quadrature rules is g_{exp}, too; cf. [11].

In principle, one could construct multi-dimensional quadrature formulas via tensorisation: For $f\colon \Gamma \to \mathbb{C}$ and $m = (m_1, \dots, m_d) \in \mathbb{N}^d$ let

$$(\mathcal{Q}^{(1)}_{m_1} \otimes \cdots \otimes \mathcal{Q}^{(d)}_{m_d})(f) = \sum_{j_1=1}^{p^{(1)}(m_1)} \cdots \sum_{j_d=1}^{p^{(d)}(m_d)} \omega^{(1)}_{j_1} \cdots \omega^{(d)}_{j_d} f(y^{(1)}_{j_1}, \dots, y^{(d)}_{j_d}) \approx \int_\Gamma f(y)\rho(y)\mathrm{d}y.$$

Note that this approximation requires $p^{(1)}(m_1) \cdots p^{(d)}(m_d)$ function evaluations. We call such a quadrature rule a *tensor product quadrature rule* and the corresponding grid a *full grid.* Now we define the *half-exact set* $\Pi^f_{\mathbf{m}}$ via

$$I(\Pi^f_{\mathbf{m}}) = \{\mathbf{k} \in \mathbb{N}^d_0 \mid k_i \le \lfloor \tfrac{q^{(i)}(m_i)}{2} \rfloor \text{ for } i = 1, \dots, d\}, \qquad \Pi^f_{\mathbf{m}} = \{\phi_{\mathbf{k}} \mid \mathbf{k} \in I(\Pi^f_{\mathbf{m}})\},$$

where $\phi_{\mathbf{k}}$ is again the multivariate orthonormal polynomial defined in (9). The upper index f stands for "full grid" and $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. The admissible set Π as introduced in (11) and below corresponds to the set $\Pi^f_{\mathbf{m}}$ here. The term *half-exact set* comes from the fact that the square of a polynomial $\phi_{\mathbf{k}}$, $\mathbf{k} \in \mathbb{N}^d_0$, is exactly integrated by $\mathcal{Q}^{(1)}_{m_1} \otimes \cdots \otimes \mathcal{Q}^{(d)}_{m_d}$ if and only if $\phi_{\mathbf{k}} \in \Pi^f_{\mathbf{m}}$ or equivalently if $\mathbf{k} \in I(\Pi^f_{\mathbf{m}})$.

As before, we denote the cardinality of $\Pi = \Pi^f_{\mathbf{m}}$ by $P = P(\mathbf{m})$ and the total number of nodes which belong to the quadrature rule by $Q = Q(\mathbf{m})$. Recall that these quantities determine the number of deterministic PDEs which have to be solved for the stochastic Galerkin and stochastic collocation methods, respectively. Example 6.1 in the Appendix shows that even on full grids, P is usually smaller than Q (unless Gauss quadrature formulas are used). We will see below that this effect is even more pronounced on sparse grids, which is crucial for understanding the efficiency of our method.

Let us now discuss the construction of sparse grids, which is due to Smolyak in [45]. Smolyak's idea was to combine certain full tensor grids with few nodes to obtain a quadrature formula which is exact for all polynomials which correspond to a certain admissible set. Smolyak's formula is explicitly given by

$$\mathcal{Q}_\ell(f) = \sum_{\mathbf{m} \in \mathcal{I}_\ell} (-1)^{\ell+d-|\mathbf{m}|} \binom{d-1}{\ell+d-|\mathbf{m}|} (\mathcal{Q}^{(1)}_{m_1} \otimes \cdots \otimes \mathcal{Q}^{(d)}_{m_d})(f), \tag{18}$$

where $\mathcal{I}_\ell = \{\mathbf{m} \in \mathbb{N}^d \mid \ell + 1 \le |\mathbf{m}| \le \ell + d\}$ is the Smolyak multi-index set of level ℓ. Now we define

$$\Pi_\ell \coloneqq \bigcup_{\mathbf{m} \in \mathcal{I}_\ell} \Pi^f_{\mathbf{m}}. \tag{19}$$

It has been shown in [14, Cor. 3.3] that

$$\Pi_\ell \subseteq \{\phi_{\mathbf{k}} \mid \mathcal{Q}_\ell(\phi_{\mathbf{k}}\phi_{\mathbf{k}}) = \langle \phi_{\mathbf{k}}, \phi_{\mathbf{k}} \rangle_\rho = 1\}. \tag{20}$$

This set Π_ℓ is used as a basis for the ansatz space of our method later on, which is why it is central to what follows. We remark that it is usually *not* true that $\mathcal{Q}_\ell(\phi_{\mathbf{k}}\phi_{\mathbf{j}}) = \delta_{\mathbf{kj}}$ for *all* $\phi_{\mathbf{k}}$, $\phi_{\mathbf{j}} \in \Pi_\ell$. Hence, a naive orthogonal expansion of a basis function $\phi_{\mathbf{k}}$ does in general not reproduce this function, i.e.

$$\phi_{\mathbf{k}} \neq \sum_{\phi_{\mathbf{j}} \in \Pi} \mathcal{Q}_\ell(\phi_{\mathbf{k}}\phi_{\mathbf{j}})\phi_{\mathbf{j}} \qquad \text{in contrast to} \qquad \phi_{\mathbf{k}} = \sum_{\phi_{\mathbf{j}} \in \Pi} \langle \phi_{\mathbf{k}}, \phi_{\mathbf{j}} \rangle_\rho \, \phi_{\mathbf{j}}. \tag{21}$$

The use of such a quadrature rule for spectral expansions must thus be modified; this will be explained at the end of this subsection.

Which choices are left? In each dimension, we can select a family of quadrature formulas. Moreover, we have to choose a growth rule $m_i \mapsto p^{(i)}(m_i)$ for each $i = 1, \dots, d$. This choice usually depends on the family of quadrature formulas, since e.g. a Clenshaw-Curtis family is often used with an exponential growth rule to obtain a nested family of sparse grid rules, as explained before. Last but not least, we choose the level ℓ depending on the desired accuracy. These choices then determine the set of nodes and weights of the sparse grid formula, the half-exact set and in particular the polynomial basis Π_ℓ used for spectral expansions.

Let us give two examples of sparse grids and their half-exact sets. In Figure 1, a two-dimensional sparse grid built from one-dimensional Clenshaw-Curtis quadrature rules on $\Gamma = [-1,1] \times [-1,1]$ with uniform weight is shown. We choose level $\ell = 5$ here. The growth rule is p_{exp}, which roughly speaking leads to increased accuracy for functions which primarily depend on one of the two variables, and less accuracy for functions containing "mixed terms".

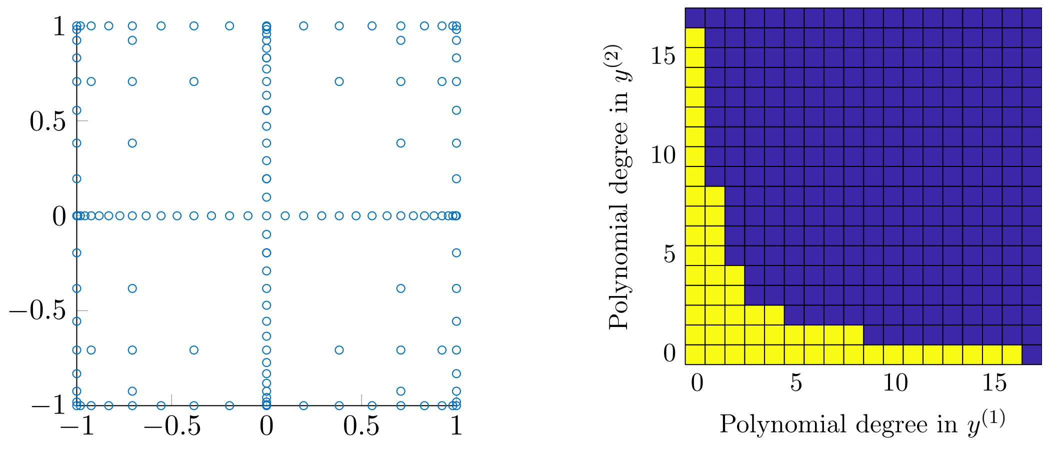


FIGURE 1. Two-dimensional Clenshaw-Curtis sparse grid of level $\ell = 5$ with exponential growth rule; left picture: nodes, right picture: Π_ℓ (yellow)

In Figure 2, a two-dimensional sparse grid built from one-dimensional Gauss-Hermite quadrature rules on $(-\infty,\infty) \times (-\infty,\infty)$ with Gaussian weight is shown. We choose level $\ell = 8$ here. The growth rule is p_{lin}, which leads – compared to the previous example – to a half-exact set with more mixed terms but less terms which just depend on one of the variables.

Now examine the values of P and Q for sparse grids. These values cannot be easily computed from the Smolyak multi-index set, the growth rules and the type of quadrature formula, since different full grids in Smolyak's formula may have common nodes; cf. Example 6.2 in the Appendix. In Tables 1 – 2, we compare the values of P and Q for different sparse grid rules in 2 and 5 dimensions. Let us recall that a stochastic Galerkin approach as explained in Section 3.3 leads to a system of P coupled PDEs, whereas a stochastic collocation approach as in Section 3.4 leads to Q decoupled PDEs.

Clenshaw-Curtis quadrature rules have the possible downside of requiring in general more quadrature nodes than Gauss quadrature rules to achieve a given polynomial exactness due to the lower order. This is not the case in practice if one uses fully nested Clenshaw-Curtis grids.

We present a third option in Table 3, namely Gauss quadrature rules with growth rule p_{slow} from (17). Observe how the choice of growth rule p_{slow} for Gauss rules greatly reduces the total number of collocation points Q compared to p_{lin}, although the one-dimensional quadrature formulas use the same or a higher number of points (since $p_{\text{lin}}(m) \le p_{\text{slow}}(m)$ for all $m \in \mathbb{N}$). This improvement originates from the fact that many collocation points occur on several grids

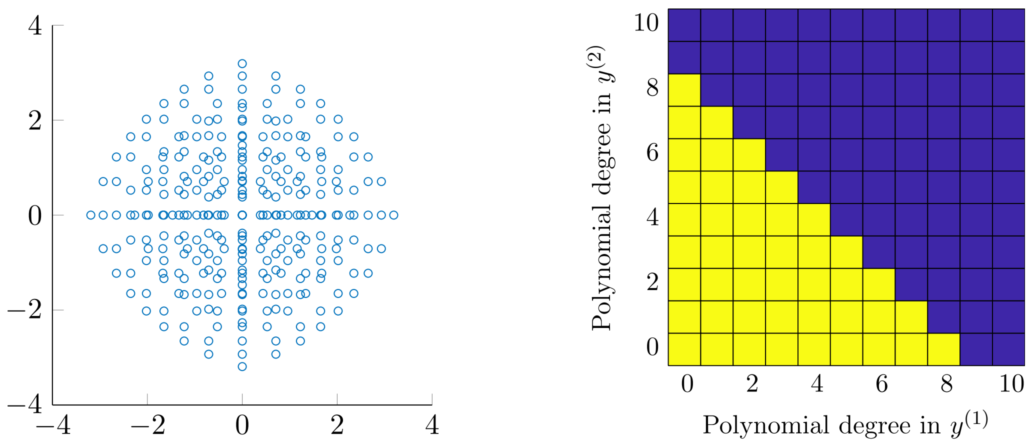


FIGURE 2. Two-dimensional Gauss-Hermite sparse grid of level $\ell = 8$ with linear growth rule; left picture: nodes, right picture: Π_ℓ (yellow)

	ℓ	1	2	3	4	5	6	7	8
$d=2$	P	3	6	10	15	21	28	36	45
	Q	5	13	29	54	90	139	203	284
	P/Q	0.60	0.46	0.34	0.28	0.24	0.20	0.18	0.16
$d=5$	P	6	21	56	126	252	462	792	1287
	Q	11	61	241	785	2239	5761	13657	30267
	P/Q	0.55	0.34	0.23	0.16	0.11	0.083	0.061	0.045

TABLE 1. Values of P and Q for Gauss quadrature with p_{lin}

	ℓ	1	2	3	4	5	6	7	8
$d=2$	P	3	6	12	25	53	113	241	513
	Q	5	13	29	65	149	350	799	1781
	P/Q	0.60	0.46	0.41	0.38	0.36	0.32	0.30	0.29
$d=5$	P	6	21	61	166	437	1122	2822	6977
	Q	11	61	241	801	2449	7205	20673	57737
	P/Q	0.55	0.34	0.25	0.21	0.18	0.16	0.14	0.12

TABLE 2. Values of P and Q for Clenshaw-Curtis quadrature with p_{exp}

corresponding to different levels when p_{slow} is chosen. A comparison of slow growth Gauss quadrature and Clenshaw-Curtis quadrature shows that this modification is even better in the sense of a higher P/Q ratio, i.e. less quadrature nodes for a given size of the polynomials basis.

We stress that the value of P is not the only relevant information which determines the polynomial approximation space, since the selection of polynomials (more or less isotropic, different accuracies in each direction) matters, too. Compare again Figures 1 and 2.

The choice of quadrature rules (Newton-Cotes, Gauss, Clenshaw-Curtis, etc.) is difficult and discussed extensively in the literature. Gauss quadrature formulas have the highest possible accuracy for the same number of nodes, but do not offer nested nodes for quadrature formulas of different order, which would be beneficial in the sparse grid setting. In high dimensions or

	ℓ	1	2	3	4	5	6	7	8
$d=2$	P	5	9	13	21	25	37	41	57
	Q	5	9	18	36	49	89	106	176
	P/Q	1	1	0.76	0.64	0.56	0.46	0.42	0.35
$d=5$	P	11	51	141	301	583	1023	1673	2633
	Q	11	51	155	427	1069	2339	4963	9815
	P/Q	1	1	0.93	0.77	0.62	0.52	0.42	0.34

TABLE 3. Values of P and Q for Gauss quadrature with p_{slow}

for high levels, this leads to considerably more nodes than the use of e.g. Clenshaw-Curtis quadrature formulas. A disadvantage of Clenshaw-Curtis quadrature is that it is not available for weighted integrals over infinite domains, which appear frequently when one of the basic random variables is normally distributed. A nested variation of Gauss quadrature formulas with a degree of exactness between Clenshaw-Curtis and Gauss is available for Gaussian and uniform weights (corresponding to normally and uniformly distributed random variables), the *Gauss-Patterson* formulas; cf. Section 5. Several authors have suggested to use these quadrature formulas for stochastic computations; cf. [19, 31, 5].

After this comparison between values of P and Q in different configurations, we conclude that collocation methods have the following disadvantage compared to Galerkin methods: In order to obtain numerical solutions with the same accuracy (meaning that we use the same polynomial ansatz space $\text{span}(\Pi)$), the number Q of quadrature nodes in the collocation method is typically large compared to the number of polynomials $P = \text{card}(\Pi)$ in the Galerkin approach. This observation is crucial for the splitting idea presented in the next section. One may use the same value for P and Q for product formulas of one-dimensional Gauss quadrature rules, but these are not feasible in practice for large values of d due to the *curse of dimension.*

3.6. **Sparse pseudospectral approximation.** In Section 3.4, we have pointed out that pseudospectral approximation on sparse grids suffers from internal aliasing if the set of basis polynomials and the quadrature rule are chosen in a naive way. Now we explain how this can be avoided. As before, one seeks an approximation $\widetilde{f}$ for a function $f\colon \Gamma \to \mathbb{C}$ of the form

$$f(y) = \sum_{\phi\in\mathbb{N}_0^d} f_\phi \phi(y) \approx \sum_{\phi\in\Pi} \widetilde{f}_\phi \phi(y) = \widetilde{f}(y), \qquad y \in \Gamma.$$

The true (but in general unavailable) coefficients f_ϕ are replaced by certain $\widetilde{f}_\phi$. As we have seen in Section 3.4, pseudospectral approximation means that for every $\phi \in \Pi$, one uses the relation

$$f_\phi = \langle f, \phi\rangle_\rho = \int_\Gamma f(y)\phi(y)\rho(y)\mathrm{d}y$$

and approximates the integral on the right-hand side by a quadrature formula. Now we assume, however, that values of f are only given on a sparse grid such that standard quadrature formulas cannot be applied. The naive approach would be to approximate the integral on the right-hand side by

$$\mathcal{Q}_\ell(f\phi) \approx \int_\Gamma f(y)\phi(y)\rho(y)\mathrm{d}y, \tag{22}$$

where $\mathcal{Q}_\ell$ is the sparse grid quadrature operator from (18). This is a computable quantity since we have assumed that the values of f are available on the nodes of the sparse grid. This approach, however, has to be modified because of (21). The following modification has been suggested in [15], but our presentation is different. Let $\mathbf{m} \in \mathcal{I}_\ell$ be the multi-index of one fixed full grid, and let $\omega_{\mathbf{m}}(y_q)$ denote the weight from quadrature rule $\mathcal{Q}_{\mathbf{m}}$ at the node y_q for $q \in \{1, \dots, Q\}$. (If y_q is not a node for $\mathbf{m}$, then $\omega_{\mathbf{m}}(y_q)$ is zero.) We rewrite the definition of $\mathcal{Q}_\ell(f\phi)$ to

$$\mathcal{Q}_\ell(f\phi) = \sum_{\mathbf{m}\in\mathcal{I}_\ell} c(\mathbf{m})(\mathcal{Q}^{(1)}_{m_1} \otimes \cdots \otimes \mathcal{Q}^{(d)}_{m_d})(f\phi) = \sum_{q=1}^{Q} \sum_{\mathbf{m}\in\mathcal{I}_\ell} c(\mathbf{m})\omega_{\mathbf{m}}(y_q)\phi(y_q)f(y_q) \tag{23}$$

with

$$c(\mathbf{m}) = (-1)^{\ell+d-|\mathbf{m}|}\binom{d-1}{\ell+d-|\mathbf{m}|}.$$

We choose an enumeration $\phi_1, \dots, \phi_P$ of the polynomials in Π and set $\widetilde{\Xi} = (\widetilde{\Xi}_{p,q}) \in \mathbb{R}^{P\times Q}$ with

$$\widetilde{\Xi}_{p,q} = \sum_{\mathbf{m}\in\mathcal{I}_\ell} c(\mathbf{m})\omega_{\mathbf{m}}(y_q)\phi_p(y_q).$$

This yields $\widetilde{\Xi}(f(y_q))_{q=1}^Q = (\mathcal{Q}_\ell(f\phi_p))_{p=1}^P \approx (f_{\phi_p})_{p=1}^P$. In order to avoid the problem (21), the matrix $\widetilde{\Xi}$ has to be replaced by $\Xi = (\Xi_{p,q}) \in \mathbb{R}^{P\times Q}$ defined by

$$\Xi_{p,q} = \sum_{\substack{\mathbf{m}\in\mathcal{I}_\ell \\ \phi_p\in\Pi^f_{\mathbf{m}}}} c(\mathbf{m})\omega_{\mathbf{m}}(y_q)\phi_p(y_q). \tag{24}$$

This matrix is called the *pseudospectral weight matrix*, and according to [15] the correct sparse pseudospectral approximation method (SPAM) is now given by

$$(\widetilde{f}_{\phi_p})_{p=1}^P \coloneqq \Xi(f(y_q))_{q=1}^Q \approx (f_{\phi_p})_{p=1}^P.$$

Hence, the coefficients $\widetilde{f}_\phi$ are obtained from the function values $f(y_1), \dots, f(y_Q)$ by multiplication with the pseudospectral weight matrix.

For later reference, we define the sparse pseudospectral operator

$$\mathcal{S}_\ell f = \sum_{p=1}^{P} \widetilde{f}_{\phi_p}\phi_p \tag{25}$$

for $f \in C(\Gamma, \mathbb{C})$. The SPAM is consistent in the sense that

$$\mathcal{S}_\ell(\phi_{\mathbf{k}}) = \phi_{\mathbf{k}} \qquad \text{in contrast to} \qquad \sum_{\phi_{\mathbf{j}}\in\Pi} \mathcal{Q}_\ell(\phi_{\mathbf{k}}\phi_{\mathbf{j}})\phi_{\mathbf{j}} \neq \phi_{\mathbf{k}}\ ;$$

cf. (21).

4. Galerkin-collocation splitting (GCS) method

Let us recall the observations from the previous sections: The Galerkin approach is well-suited for the linear parts of problem (5) and probably more beneficial due to the reduced number of equations, whereas the collocation approach has no difficulty in dealing with the nonlinearity. Let us now introduce the idea of combining both approaches via the concept of Strang splitting. To the best of our knowledge, such a method has not been proposed so far.

We assume again that the solution u of (4) admits a PCE of the form

$$u(t,x,y) = \sum_{\mathbf{k}\in\mathbb{N}_0^d} u_{\mathbf{k}}(t,x)\phi_{\mathbf{k}}(y), \qquad u_{\mathbf{k}}(t,x) = \langle u(t,x,\cdot), \phi_{\mathbf{k}}\rangle_\rho \tag{26}$$

for each $t \geq 0$ and a.e. $x \in S$. We choose the ansatz space span(Π) for a finite set of multivariate orthonormal polynomials Π as explained in (19) to obtain a numerical approximation

$$\widetilde{u}(t,x,y) = \sum_{\phi\in\Pi} \widetilde{u}_\phi(t,x)\phi(y) \approx \sum_{\mathbf{k}\in\mathbb{N}_0^d} u_{\mathbf{k}}(t,x)\phi_{\mathbf{k}}(y) = u(t,x,y).$$

The main idea is now to split the random PDE (5) into the three subproblems

$$\partial_t v(t,x,y) = \alpha(y)\mathcal{A}v(t,x,y), \qquad t \geq t_0, \quad v(t_0,x,y) = v_0(x,y), \tag{27a}$$

$$\partial_t v(t,x,y) = \beta(y)\mathcal{B}v(t,x,y) + \gamma(y)g(x), \qquad t \geq t_0, \quad v(t_0,x,y) = v_0(x,y), \tag{27b}$$

$$\partial_t v(t,x,y) = F(v(t,x,y)), \qquad t \geq t_0, \quad v(t_0,x,y) = v_0(x,y), \tag{27c}$$

with $v_0(\cdot,y) \in \mathrm{D}(\mathcal{A})$ for every $y \in \Gamma$, and then to apply a stochastic Galerkin method to the linear parts (27a), (27b), and a stochastic collocation method to the nonlinear part (27c).

To obtain an approximation

$$\widetilde{v}(t,x,y) = \sum_{\phi\in\Pi} \widetilde{v}_\phi(t,x)\phi(y) \approx \sum_{\mathbf{k}\in\mathbb{N}_0^d} v_{\mathbf{k}}(t,x)\phi_{\mathbf{k}}(y) = v(t,x,y) \tag{28}$$

for the solution of (27a) in the space span(Π), we use the Galerkin ansatz, which yields

$$\langle \partial_t \widetilde{v}(t,x,\cdot), \phi\rangle_\rho = \sum_{\psi\in\Pi} \mathcal{A}\widetilde{v}_\phi(t,x)\langle \alpha\psi, \phi\rangle_\rho \qquad \text{for all } \phi \in \Pi \tag{29}$$

with initial data

$$\widetilde{v}(t_0,x,y) = \sum_{\phi\in\Pi} \widetilde{v}_\phi(t_0,x)\phi(y), \qquad \widetilde{v}_\phi(t_0,x) = \langle v_0(x,\cdot), \phi\rangle_\rho. \tag{30}$$

In order to keep the presentation simple, we pretend for the moment that all inner products can be computed exactly.

Now we use again single indices $j = 1,\dots,P$ with $P = |\Pi|$ to enumerate the polynomials in Π such that $\Pi = \{\phi_1,\dots,\phi_P\}$. If we set $a_{jk} = \langle \alpha\phi_j, \phi_k\rangle_\rho$ and $\mathbf{M}_\alpha = (a_{jk})_{j,k=1}^P$ as in (15a), then (29) implies

$$\partial_t \widetilde{\mathbf{v}}(t,x) = \mathbf{M}_\alpha \mathcal{A}\widetilde{\mathbf{v}}(t,x) \tag{31}$$

with

$$\widetilde{\mathbf{v}}(t,x) = (\widetilde{v}_j(t,x))_{j=1}^P \qquad \text{and} \qquad \mathcal{A}\widetilde{\mathbf{v}}(t,x) = (\mathcal{A}\widetilde{v}_j(t,x))_{j=1}^P.$$

Since $\alpha(y)$ is real-valued, the matrix $\mathbf{M}_\alpha$ is symmetric. Hence, we may decompose $\mathbf{M}_\alpha = Q_\alpha\Lambda_\alpha Q_\alpha^\top$ with $\Lambda_\alpha = \mathrm{diag}(\lambda_1^\alpha,\dots,\lambda_P^\alpha)$ and an orthogonal matrix Q_α. The matrices Λ_α and Q_α have to be computed only once at the beginning of the time integration process because $\alpha(y)$ does not depend on time. Via the transformation $\mathbf{w} = (w_1,\dots,w_P)^\top = Q_\alpha^\top\widetilde{\mathbf{v}}$, the system (31) is equivalent to the decoupled equations

$$\partial_t w_j(t,x) = \lambda_j^\alpha \mathcal{A}w_j(t,x), \qquad j = 1,\dots,P. \tag{32}$$

Each of these equations has the form (7a), and we have assumed in Section 3.1 that the solution

$$w_j(t,\cdot) = \exp\left((t-t_0)\lambda_j^\alpha\mathcal{A}\right) w_j(t_0,\cdot) \tag{33}$$

can be approximated numerically at low costs. In case of the Lugiato-Lefever equation, for example, we have $\mathcal{A} = \mathrm{i}\partial_x^2 u$, $\alpha = a$, and if we discretize space via trigonometric polynomials, then $\exp((t-t_0)\lambda_j^\alpha \mathcal{A})$ is represented by the exponential of a diagonal matrix; cf. [32, 17]. Note that the number of independent deterministic PDEs in (32) is equal to P, the dimension of $\operatorname{span}(\Pi)$, and that these decoupled PDEs may be solved in parallel.

To approximate v in (27b), we mimic this procedure. Imposing the Galerkin condition yields

$$\langle \partial_t \widetilde{v}(t,x,\cdot), \phi_j \rangle_\rho = \sum_{k=1}^P \mathcal{B}\widetilde{v}_k(t,x) \langle \beta\phi_k, \phi_j\rangle_\rho + g(x)\langle \gamma, \phi_j\rangle_\rho,$$

Abbreviating $b_{jk} = \langle \beta\phi_k, \phi_j\rangle_\rho$, $\mathbf{M}_\beta = (b_{jk})_{j,k=1}^P$ and $\mathbf{V}_\gamma = (\langle \gamma, \phi_j\rangle_\rho)_{j=1}^P$, we obtain

$$\partial_t \widetilde{\mathbf{v}}(t,x) = \mathbf{M}_\beta \mathcal{B}\widetilde{\mathbf{v}}(t,x) + g(x)\mathbf{V}_\gamma \tag{34}$$

with

$$\widetilde{\mathbf{v}}(t,x) = (\widetilde{v}_j(t,x))_{j=1}^P \qquad \text{and} \qquad \mathcal{B}\widetilde{\mathbf{v}}(t,x) = (\mathcal{B}\widetilde{v}_j(t,x))_{j=1}^P.$$

Recall that by assumption $\beta(y) = \beta_0 + \beta_1(y)$ with $\beta_0 \in \mathbb{C}$ and $\beta_1(y) \in \mathbb{R}$. Hence, it follows that

$$\mathbf{M}_\beta = \beta_0 I + \mathbf{M}_{\beta_1} \qquad \text{with} \qquad \mathbf{M}_{\beta_1} = (\langle \beta_1\phi_j, \phi_k\rangle_\rho)_{j,k=1}^P.$$

The symmetric matrix $\mathbf{M}_{\beta_1}$ has an eigendecomposition $\mathbf{M}_{\beta_1} = Q_{\beta_1}\Lambda_{\beta_1}Q_{\beta_1}^\top$ with an orthogonal matrix Q_{β_1}. This yields the eigendecomposition $\mathbf{M}_\beta = Q_\beta \Lambda_\beta Q_\beta^\top$ with $Q_\beta = Q_{\beta_1}$ and $\Lambda_\beta = \operatorname{diag}(\lambda_1^\beta, \dots, \lambda_P^\beta) = \beta_0 I + \Lambda_{\beta_1}$. After the transformation $\mathbf{w} = (w_1, \dots, w_P)^\top = Q_\beta^\top \widetilde{\mathbf{v}}$, the PDE system (34) decouples to

$$\partial_t w_j(t,x) = \lambda_j^\beta \mathcal{B} w_j(t,x) + g(x)(Q_\beta^\top \mathbf{V}_\gamma)_j, \qquad j = 1, \dots, P,$$

where $(Q_\beta^\top \mathbf{V}_\gamma)_j$ denotes the j-th entry of $Q_\beta^\top \mathbf{V}_\gamma$. Since each of these equations is of the type (7b), the solution

$$w_j(t,x) = \exp\left((t-t_0)\lambda_j^\beta \mathcal{B}\right) w_j(t_0,x) + \int_{t_0}^t \exp\left((t-s)\lambda_j^\beta \mathcal{B}\right) g(x)(Q_\beta^\top \mathbf{V}_\gamma)_j \mathrm{d}s \tag{35}$$

can be approximated efficiently according to the assumption made in Section 3.1. In case of the Lugiato-Lefever equation, for example, we have $\mathcal{B}u = \mathrm{i}u$, $\beta_0 = \mathrm{i}$, $\beta_1 = b$, $\gamma = f$ and $g \equiv 1$ such that

$$w_j(t,x) = \exp\left((t-t_0)\mathrm{i}\lambda_j^\beta\right) w_j(t_0,x) + (\mathrm{i}\lambda_j^\beta)^{-1}\left(\exp\left((t-t_0)\mathrm{i}\lambda_j^\beta\right) - 1\right)(Q_\beta^\top \mathbf{V}_\gamma)_j.$$

Note that $\lambda_j^\beta \neq 0$ for all $j = 1, \dots, P$ due to $\beta_0 = \mathrm{i}$.

For (27c), we use the collocation approach instead of the Galerkin approach. As we will see below, it is important that we use exactly the nodes $y_1, \dots, y_Q \in \Gamma$ from which the quadrature rule with half-exact set Π was built. For these y_q we solve the Q deterministic differential equations[4]

$$\partial_t v(t,x,y_q) = F(v(t,x,y_q)), \qquad v(t_0,x,y_q) = v_0(x,y_q), \qquad q = 1, \dots, Q. \tag{36}$$

[4] The integer Q is not to be confused with the orthogonal matrices Q_α and Q_β from above. The collocation points $y_1, \dots, y_Q \in \Gamma$ should not be confused with the components of a vector $y = (y_1, \dots, y_d) \in \Gamma$ from before. The latter does not appear in Section 4 anymore.

In the splitting scheme, the function $v_0(x,y)$ is not explicitly given, but the coefficients of its truncated PCE

$$v_0(x,y) = \sum_{p=1}^{P} v_p(t_0,x)\phi(y)$$

are available. To evaluate this sum at the points $y_1,\dots,y_Q$, only the values $\phi(y_1),\dots,\phi(y_Q)$ have to be computed for each $\phi \in \Pi$. Now each of the initial value problems (36) has the same structure as (7c), and approximating its solution numerically is cheap according to our assumption in Section 3.1. In case of the Lugiato-Lefever equation (i.e. for $F(v) = \mathrm{i}|v|^2 v$) the exact solution of (36) is explicitly known, namely

$$v(t,x,y_q) = \exp\left((t-t_0)|v_0(x,y_q)|^2\right) v_0(x,y_q), \qquad q = 1,\dots,Q, \quad x \in S. \tag{37}$$

From $v(t,x,y_q)$, $q = 1,\dots,Q$, at the new time $t > t_0$, the coefficients $\widetilde{v}_p(t,x)$ with

$$\widetilde{v}(t,x,y_q) = \sum_{p=1}^{P} \widetilde{v}_p(t,x)\phi_p(y_q) \approx v(t,x,y_q), \qquad q = 1,\dots,Q$$

can be obtained by sparse pseudospectral approximation as explained in Section 3.5. Using the notation from this section, we have

$$(\widetilde{v}_p(t,x))_{p=1}^{P} = \Xi(v(t,x,y_q))_{q=1}^{Q},$$

or, equivalently, $\widetilde{v}(t,x,\cdot) = \mathcal{S}_\ell v(t,x,\cdot)$, where $\mathcal{S}_\ell$ is the sparse pseudospectral operator defined in (25).

The inner products in (30) and in the definitions of $\mathbf{M}_\alpha$, $\mathbf{M}_\beta$ and $\mathbf{V}_\gamma$ can be computed as follows. To determine the initial data (30) has to be replaced by $\widetilde{v}(t_0,x,\cdot) = \mathcal{S}_\ell v_0(x,\cdot)$ with $\mathcal{S}_\ell$ from (25). The entries of $\mathbf{M}_\alpha$ can be approximated via sparse pseudospectral approximation, too. We do not do this directly, since from the definition of Π as a half-exact set it is clear that for fixed $\psi \in \Pi$ the sum

$$\sum_{\phi\in\Pi} \langle \alpha\psi, \phi\rangle_\rho \phi \qquad \text{and} \qquad \mathcal{S}_\ell(\alpha\psi)$$

will almost never coincide if α is non-constant due to the degrees of the participating polynomials. Therefore, we suggest a small modification here. First, compute the pseudospectral approximation $\mathcal{S}_\ell(\alpha)$ itself without the ψ from before, i.e.

$$\alpha(y) \approx \mathcal{S}_\ell\alpha(y) = \sum_{p=1}^{P} \widetilde{\alpha}_p\phi_p(y),$$

and then we use the approximation

$$\langle \alpha\phi_j, \phi_k\rangle_\rho \approx \langle \mathcal{S}_\ell(\alpha)\phi_j, \phi_k\rangle_\rho = \sum_{i=1}^{P} \widetilde{\alpha}_i \langle \phi_i\phi_j, \phi_k\rangle_\rho.$$

For the integrals in the entries of the tensor $(\langle\phi_i\phi_j,\phi_k\rangle_\rho)_{i,j,k=1}^{P}$, we now use the exact formulas which are available for most of the "typical" families of orthogonal polynomials, e. g. Legendre and Hermite polynomials. A list of available formulas can be found in the Appendix in [30]. Now it is clear that for $\alpha \in \operatorname{span}(\Pi)$, the matrix $\mathbf{M}_\alpha$ can be computed exactly since $\mathcal{S}_\ell$ is exact on $\operatorname{span}(\Pi)$. In a similar fashion, one obtains computable approximations for $\mathbf{M}_\beta$ and $\mathbf{V}_\gamma$.

The steps of the Galerkin-collocation splitting (GCS) method are summarized in the following. It is assumed that the level ℓ and for each dimension a family of quadrature rules and a growth rule have been chosen. Recall that this determines the nodes and weights of the sparse grid formula, the half-exact set and the polynomial basis $\Pi = \Pi_\ell = \{\phi_1, \dots, \phi_P\}$ of the ansatz space. Moreover, we suppose that the length of the time interval $[0, T]$ and the number of time-steps $N_t \in \mathbb{N}$ have been chosen. The method computes approximations

$$U^n(x, y) = \sum_{p=1}^{P} U_p^n(x)\phi_p(y) = \widetilde{u}(t_n, x, y) \approx u(t_n, x, y)$$

with step-size $\tau = T/N_t$ at times $t_n = n\tau$ for $n = 0, 1, \dots, N_t$. As before, the coefficients are collected in a vector $\mathbf{U}^n(x) := (U_1^n(x), \dots, U_P^n(x))^\top$. For simplicity, we only consider the semidiscretization in time. Of course, discretizing also the spatial variable is unavoidable in practice to compute numerical results, but the choice of a suitable space discretization depends on the particular properties of the operators $\mathcal{A}$ and $\mathcal{B}$ in Section 3.1. We do not explicitely state the x-dependency of the current approximation $\mathbf{U}^n$. We define the numerical flows

$$\begin{aligned}\Phi_\tau^{(1)}\mathbf{U}^n &= \mathrm{e}^{\tau \mathbf{M}_\alpha \mathcal{A}}\mathbf{U}^n,\\ \Phi_\tau^{(2)}\mathbf{U}^n &= \mathrm{e}^{\tau \mathbf{M}_\beta \mathcal{B}}\mathbf{U}^n + \int_0^\tau \mathrm{e}^{(\tau - s)\mathbf{M}_\beta \mathcal{B}} g \mathbf{V}_\gamma \,\mathrm{d}s,\end{aligned}$$

and $\Phi_\tau^{(3)}\mathbf{U}^n = (v(\tau, x, y_q))_{q=1}^Q$, where $v = v(t, x, y_q)$ solves the PDE

$$\begin{aligned}\partial_t v(t, x, y_q) &= F(v(t, x, y_q)), \qquad t \geq 0,\\ v(0, x, y_q) &= U^n(x, y_q)\end{aligned}$$

for any $q = 1, \dots, Q$. Furthermore, we set $\Psi = (\phi_p(y_q))_{q,p} \in \mathbb{R}^{Q \times P}$.

Then the n-th approximation of our method can be written as

$$\mathbf{U}^n = \left(\Phi_{\tau/2}^{(1)} \circ \Phi_{\tau/2}^{(2)} \circ \Xi \circ \Phi_\tau^{(3)} \circ \Psi \circ \Phi_{\tau/2}^{(2)} \circ \Phi_{\tau/2}^{(1)}\right)^n \mathbf{U}^0$$

with $\mathbf{U}^0(x) = \Xi(u_0(x, y_1), \dots, u_0(x, y_q))^\top$ and Ξ from (24).

5. Numerical examples for the Lugiato-Lefever equation

In order to show how the GCS method performs in practice we apply it to the Lugiato-Lefever equation with random parameters (4). As before this equation is considered on the one-dimensional torus (i.e. $x \in S = \mathbb{T}$), which is the natural setting for applications in electrical engineering. The spatial derivatives are computed by spectral collocation with $M = 512$ grid points x_k; see Section 2.4 in [17] or Section III.1.3 in [32] for details. All computations were carried out in MATLAB® on a notebook with Intel® Core$^{\mathrm{TM}}$ i7-6700HQ CPU with 4 physical cores. All methods implemented use parallelization where it is possible.

The approximation of the solution u computed by the new method is denoted by u_{GCS}. For simplicity, we use the notation $u_{\mathrm{GCS}}(t, x, y)$ with three arguments despite the fact that values for u_{GCS} are actually only computed for discrete times $t \in \{t_0, t_1, \dots, t_{N_t}\}$ at discrete grid points $x \in \{x_1, \dots, x_M\}$.

5.1. Experiments for the random equation.

The first example illustrates the impact of randomness in the data on the solution. This shows, in particular, that solving (4) instead of the deterministic PDE (1) is unavoidable when parameters and initial data are uncertain. The parameters and initial data used in this numerical example are

$$\begin{aligned}a(y_1, y_2) &= (1 + \delta y_1 y_2)\bar{a}, & b(y_1, y_2) &= (1 + \tfrac{\delta}{3}(y_1 + y_2^2))\bar{b},\\ f(y_1, y_2) &= (1 + \delta y_2^2)\bar{f}, & u_0(x, y_1, y_2) &= (1 + 0.2 y_1)\tfrac{1}{2}\exp(\sin(2x)),\end{aligned} \tag{38}$$

with $d = 2$, $y = (y_1, y_2)$, $\bar{a} = 0.1025$, $\bar{b} = 0.5$, $\bar{f} = 1.25$ and noise parameter $\delta = 0.5$. These values are chosen in such a way that the solution converges to a frequency comb (i.e. to a steady state which is not constant in x) after a sufficiently large time interval. The random variables Y_1, Y_2 which correspond to y_1 and y_2 are chosen to be uniformly distributed on $[-1, 1]$. The corresponding orthonormal polynomials are (up to a normalization) the Legendre polynomials. We have $\mathbb{E}a = \bar{a}$, $\mathbb{E}b = \bar{b}(1+\frac{\delta}{9})$, $\mathbb{E}f = \bar{f}(1+\frac{\delta}{3})$ and $\mathbb{E}[u_0(x, Y)] = \frac{1}{2}\exp(\sin(2x))$.

To discretize the stochastic variable y, we use a sparse grid of level $\ell = 7$ built from one-dimensional Clenshaw-Curtis (CC) quadrature rules with exponential growth rule and the corresponding basis of orthonormal polynomials Π_ℓ as explained in Section 3.5. The GCS method is applied with $N_t = 5000$ time-steps on the interval $[0, T]$ with $T = 15$. The absolute value of the expectation and the standard deviation of the numerical approximation u_{GCS} are shown in Figure 3b – 3e. It can be seen that $|\mathbb{E}[u_{\mathrm{GCS}}(t, x, Y)]|$ changes rapidly at the beginning of the time interval before it converges to a steady state with four pronounced peaks (Figure 3d). The information which is most relevant for electrical engineers is the frequency comb $k \mapsto \log(|\hat{u}_k|)$ where $\hat{u}_k$ are the Fourier coefficients of such a steady state; see Section 2.1. The frequency comb of $\mathbb{E}[u_{\mathrm{GCS}}(T, x, Y)]$ is depicted in Figure 3a.

The expectation $\mathbb{E}[u_{\mathrm{GCS}}(t, x, Y)] \approx \mathbb{E}[u(t, x, Y)]$ is certainly a relevant quantity in many applications, but it only represents the mean behaviour of the system, whereas the solution of (4) provides full information for *every single y*. For several samples $y \in \{y^{(1)},\ y^{(2)},\ y^{(3)},\ \bar{y}\}$ the time evolution of the approximation $u_{\mathrm{GCS}}(\cdot, \cdot, y)$ and the corresponding final states are shown in Figure 4. These plots illustrate that $u_{\mathrm{GCS}}(\cdot, \cdot, y)$ may change considerably when y varies. This behaviour is quantified by the standard deviation shown in Figures 3c and 3e.

For a first plausibility check of our results we use the fact that evaluating the exact solution of (4) for a *fixed* vector $y = y_\star$ coincides with the exact solution of the deterministic PDE (1) with deterministic parameters $a = a(y_\star), b = b(y_\star)$ and $f = f(y_\star)$. Hence, it can be expected that evaluations of the approximation $u_{\mathrm{GCS}}(\cdot, \cdot, y)$ for $y \in \{y^{(1)},\ y^{(2)},\ y^{(3)},\ \bar{y}\}$ agree up to a small numerical error with the corresponding approximations of (1). This is indeed the case: The plots (not shown) obtained by solving (1) numerically for $y \in \{y^{(1)},\ y^{(2)},\ y^{(3)},\ \bar{y}\}$ are visually almost indistinguishable from the ones in Figure 4.

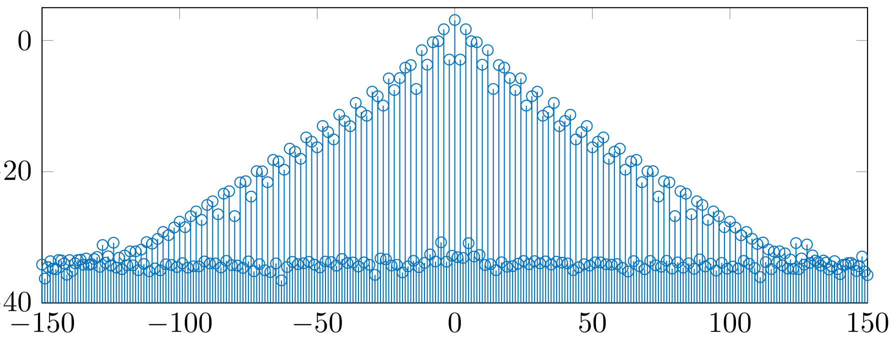
The sample $\bar{y} = (0, \frac{1}{\sqrt{3}})$ in Figure 4h is particularly interesting, because evaluations with $\bar{y}$ coincide with the expectation of the parameters and the initial data, respectively:

$$a(\bar{y}) = \mathbb{E}a(Y), \qquad b(\bar{y}) = \mathbb{E}b(Y), \qquad f(\bar{y}) = \mathbb{E}f(Y), \qquad u_0(x, \bar{y}) = \mathbb{E}[u_0(x, Y)]. \tag{39}$$

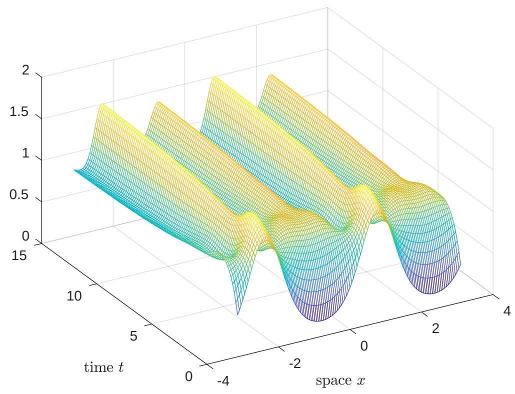
Solving (1) with the expected data (39) is, of course, much simpler than solving (4), but the outcome differs significantly[5] from the expectation $\mathbb{E}[u_{\mathrm{GCS}}(t, x, Y)]$; compare Figures 4h and 3d. This underlines that considering the random PDE (4) instead of its deterministic version (1) cannot be avoided even in cases where only the expectation of the solution is sought after.

5.2. **Convergence and efficiency.** In order to study the convergence behaviour of the GCS method we compare the final approximation $u_{\mathrm{GCS}}(T, \cdot, \cdot)$ with a Monte Carlo reference solution $u_{\mathrm{MC}}(T, \cdot, \cdot)$ obtained with $N_{\mathrm{MC}} = 10000$ samples $y_{\mathrm{MC}}^{(j)}$ from the joint distribution of Y. As for u_{GCS} we use the notation $u_{\mathrm{MC}}(t, x, y)$ although u_{MC} is only defined if $t \in \{t_0, t_1, \dots, t_{N_t}\}$, if $x \in \{x_1, \dots, x_M\}$ and if $y = y_{\mathrm{MC}}^{(j)}$ for some $j = 1, \dots, N_{\mathrm{MC}}$. Each realization $u_{\mathrm{MC}}(\cdot, \cdot, y_{\mathrm{MC}}^{(j)})$ is computed by solving the deterministic PDE (1) numerically with the Strang splitting method discussed in Section 2.1. Since we focus on the error induced by discretizing the random

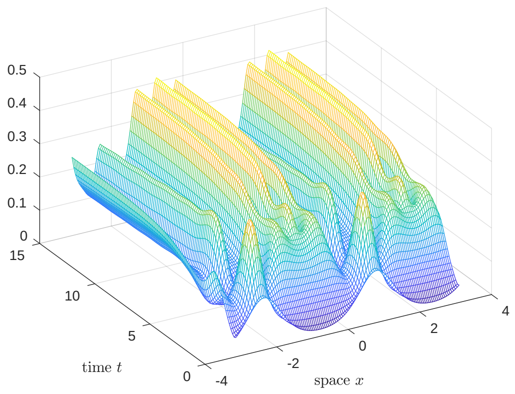
[5]This difference is not a numerical artefact, i.e. it is *not* caused by the fact that both the deterministic and the random PDE are approximated by numerical methods.



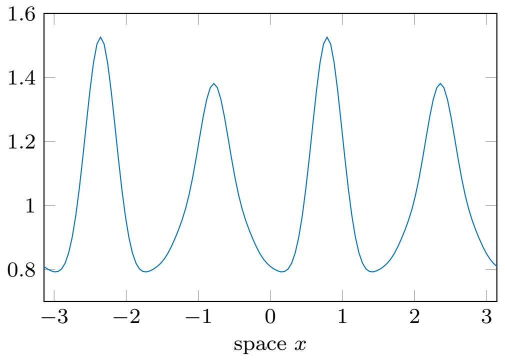
(A) Logarithm of the Fourier modes of $\mathbb{E}u_{\mathrm{GCS}}(T)$



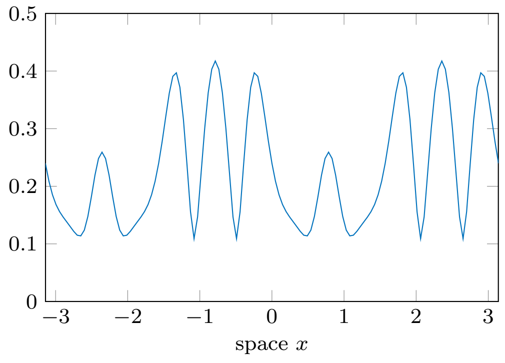
(B) $|\mathbb{E}[u_{\mathrm{GCS}}(t,x,Y)]|$



(C) $\sigma[u_{\mathrm{GCS}}(t,x,Y)]$



(D) $|\mathbb{E}[u_{\mathrm{GCS}}(T,x,Y)]|$



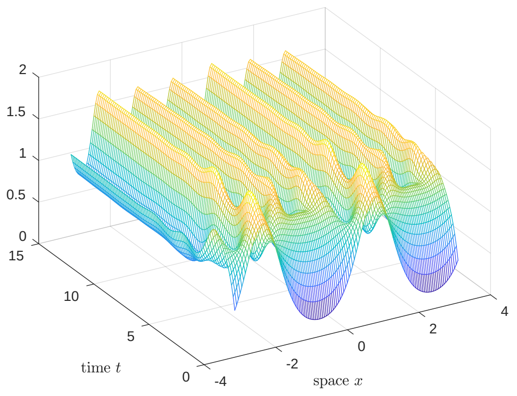
(E) $\sigma[u_{\mathrm{GCS}}(T,x,Y)]$

FIGURE 3. Solution to the random PDE (4) with the parameters from (38)

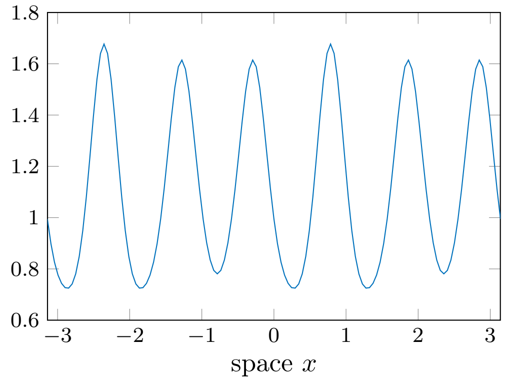
variables, we use the same spatial and temporal discretization for the Strang splitting and the GCS method, namely $M = 512$ grid points in space and $N_t = 5000$ time-steps.

The approximation error is measured in the space $L^2_\rho(\Gamma, L^2(\mathbb{T}))$, i.e. by taking the expectation with respect to y and the $L^2(\mathbb{T})$ norm with respect to x. Of course, both the expectation and the norm have to be replaced by their discrete counterparts, i.e.

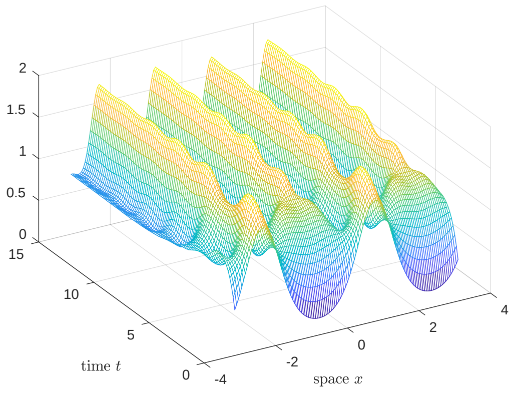
$$\frac{2\pi}{N_{\mathrm{MC}} M} \sum_{j=1}^{N_{\mathrm{MC}}} \sum_{k=1}^{M} |u_{\mathrm{MC}}(T, x_k, y_{\mathrm{MC}}^{(j)}) - u_{\mathrm{GCS}}(T, x_k, y_{\mathrm{MC}}^{(j)})|^2 \approx \mathbb{E}\left[\int_{\mathbb{T}} |u(T,x,\cdot) - u_{\mathrm{GCS}}(T,x,\cdot)|^2 \mathrm{d}x\right].$$



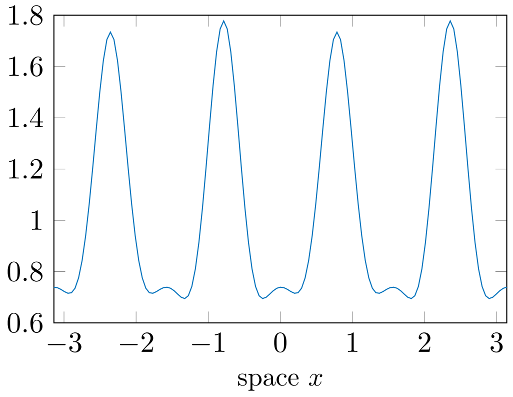
(A) $|u_{\mathrm{GCS}}(t,x,y^{(1)})|,\ y^{(1)}=(0.6,-0.8)$



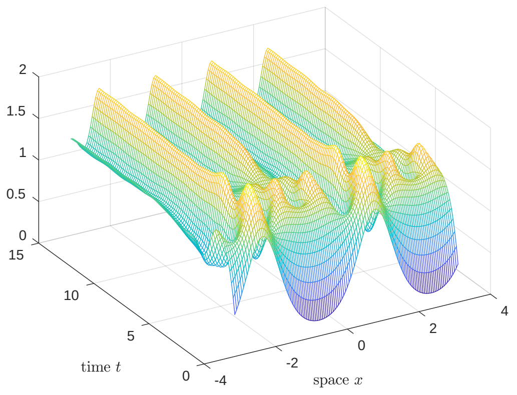
(B) $|u_{\mathrm{GCS}}(T,x,y^{(1)})|\ y^{(1)}=(0.6,-0.8)$



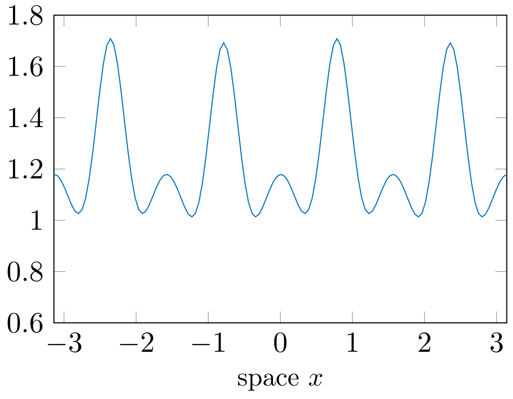
(C) $|u_{\mathrm{GCS}}(t,x,y^{(2)})|,\ y^{(2)}=(0.45,0.7)$



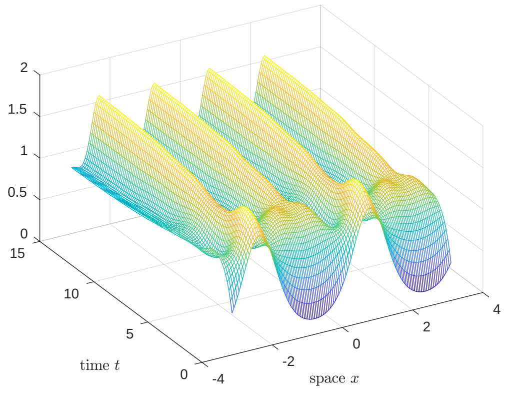
(D) $|u_{\mathrm{GCS}}(T,x,y^{(2)})|,\ y^{(2)}=(0.45,0.7)$



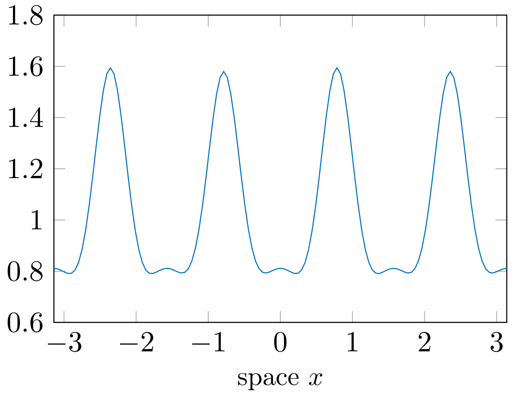
(E) $|u_{\mathrm{GCS}}(t,x,y^{(3)})|,\ y^{(3)}=(0,1)$



(F) $|u_{\mathrm{GCS}}(T,x,y^{(3)})|,\ y^{(3)}=(0,1)$



(G) $|u_{\mathrm{GCS}}(t,x,\bar{y})|,\ \bar{y}=(0,\frac{1}{\sqrt{3}})$



(H) $|u_{\mathrm{GCS}}(T,x,\bar{y})|,\ \bar{y}=(0,\frac{1}{\sqrt{3}})$

FIGURE 4. Solution to the random PDE (4) via GCS with the parameters from (38), evolution of deterministic solutions for different samples

This is possible since the PCE of $u_{\text{GCS}}(T,\cdot,\cdot)$ can be evaluated for arbitrary y and in particular at the samples $y_{\text{MC}}^{(j)}$.

We study the convergence of the method with respect to the level ℓ. This should be the crucial parameter which determines the accuracy of the approximations in random space. We expect spectral convergence since stochastic Galerkin and collocation methods usually achieve this type of convergence if coefficients and initial data are smooth, which is the case here. All parameters are as before in (38) except that we use $\delta = 0.3$ and $T = 3$. (This corresponds to less noisy parameters than before and a smaller time interval, both resulting in a smaller error. In the former case the convergence rates in the experiments are just slower.) Results are given in Figure 5. We also show the error and runtime of a competitive collocation method which uses the points of the same sparse grid as collocation points. Clearly, both methods achieve spectral convergence with more or less identical values for the L^2-error, see Figure 5A. The remarkable difference is that the GCS method is twice as fast for levels $\ell \geq 7$, as can be seen in Figure 5B. We remark that for both methods the runtime just measures the time elapsed between the beginning and the end of the time integration process. We do not measure the time for setting up the sparse grid, the Galerkin matrices, the pseudospectral weight matrix and everything else which has to be done only once before the time integration. This setup process is negligible for both methods in our practical applications.

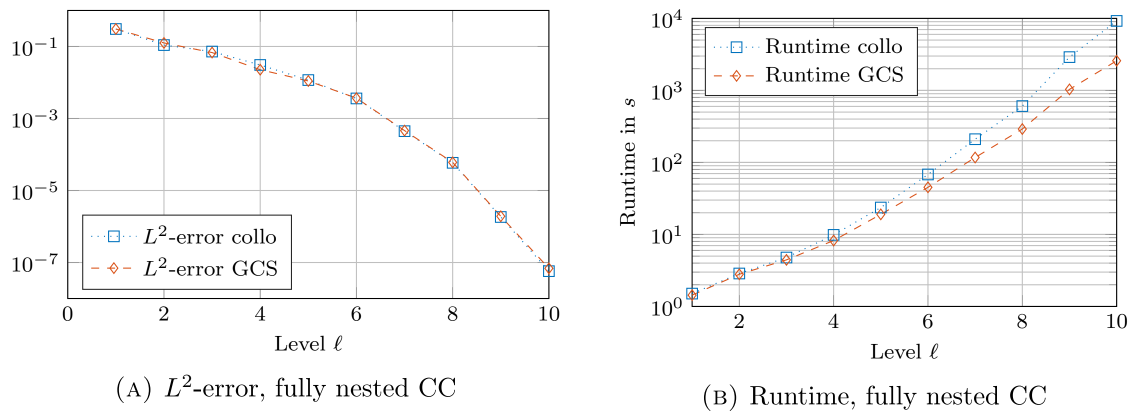


FIGURE 5. Comparison of collocation and GCS methods using Clenshaw-Curtis quadrature rules with exponential growth rule to solve problem (4)

Now we change the sparse grid to be built from Gauss-Legendre (GL) quadrature formulas with linear growth rule with the corresponding choice of Π_ℓ. Results are shown in Figures 6A and 6B. Clearly, the GCS method outperforms the collocation method up to a factor of 3. One could argue that this comparison is not fair due to the fact that the sparse grid is in no way nested and therefore the number of collocation points Q is disproportionately large in contrast to the number P of basis polynomials in the GCS method. Recall, however, that this also affects the propagation of the nonlinear part in the GCS method.

Next, we present a slight modification of *slow growth* sparse Gauss-Legendre grids. Here we replace the growth rule p_{lin} by p_{slow} from (17) for both the GCS and the collocation method. This modification leads to a partially nested version of Gauss-Legendre sparse grids. Gauss-Legendre grids with an odd number of points share the midpoint 0 and this node is therefore contained in all of the full grids which are combined in Smolyak's formula (18). Moreover, the one-dimensional grids corresponding to an even index m and the next odd index $m+1$ are the same and therefore even more nodes can be reused in the sparse grid formula with increasing

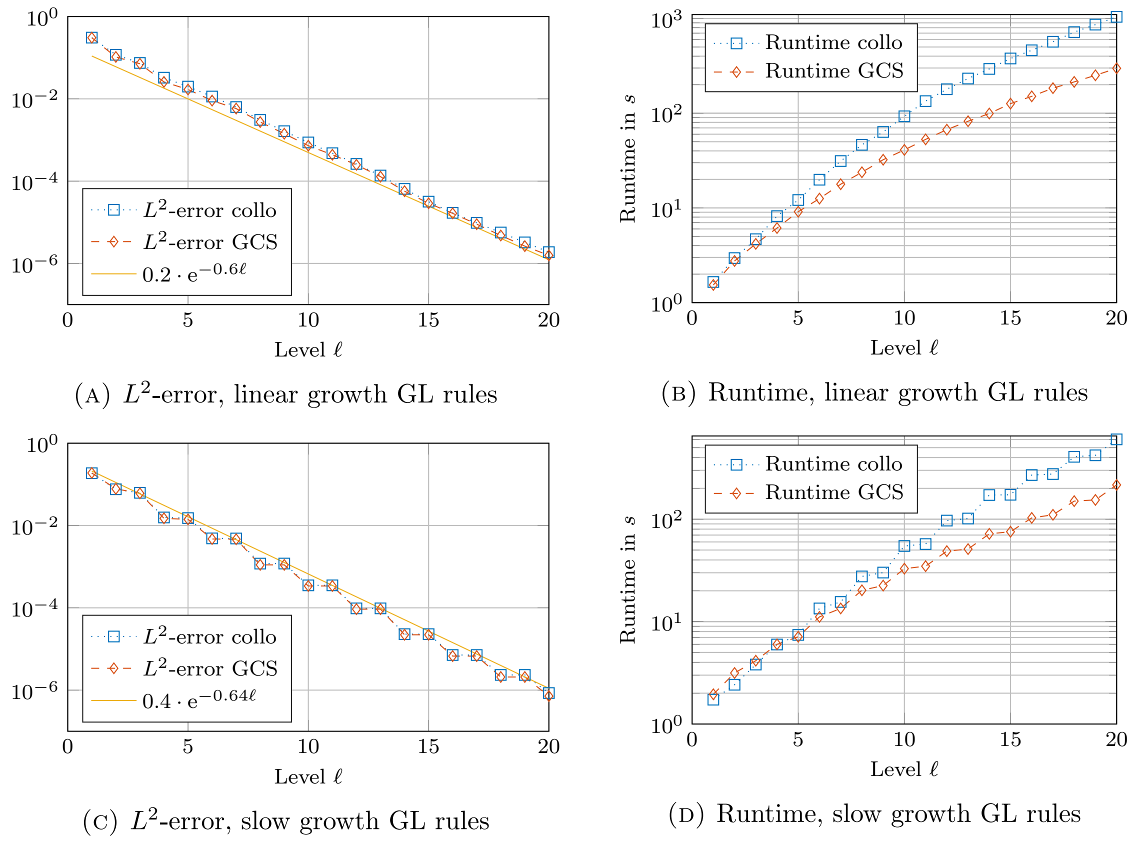


FIGURE 6. Comparison of both methods with sparse grids built from Gauss-Legendre quadrature formulas (linear and slow growth versions)

level and dimension. Using this setting, we obtain the results in Figures 6C and 6D. Both the GCS and the collocation method are significantly faster with this sparse grid while keeping the accuracy. Still the GCS is up to twice as fast.

The "steps" in the order plot for slow growth Gauss-Legendre grids can be explained by the chosen growth rule p_{slow}: Only quadrature rules with an odd number of nodes are combined here. This is the reason why the sparse grids of even level usually achieve roughly the same accuracy as the next odd-level rule. The approximations on two subsequent levels are not identical, however, because the Smolyak multi-index sets are different and therefore the combination of the involved full grids is different, too.

Gauss-Patterson quadrature. As a third family of quadrature formulas for which we test the GCS method, we use the Gauss-Patterson (GP) quadrature formulas. These have received increased attention for collocation methods, as they combine the advantages of high-order Gauss quadrature with the nesting property of Clenshaw-Curtis grids.

We do not present the construction of these quadrature formulas, but give the main idea: In the case of a uniform weight, one starts with the Gauss-Legendre quadrature formulas with 1 and 3 nodes and then recursively adds $n+1$ points to the n-point quadrature formula in such a way that the highest polynomial exactness is achieved. The Gauss-Patterson (GP) quadrature formula with $2^m - 1$ nodes then has degree of exactness $3 \cdot 2^{m-1} - 1$. Using this procedure, one obtains quadrature formulas with $1, 3, 7, 15, 31, 63, \ldots$ nodes having degrees of exactness

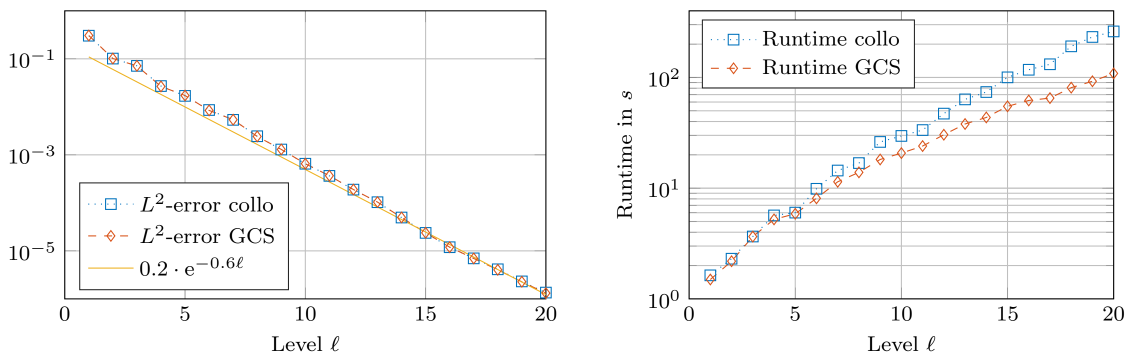
$1, 5, 11, 23, 47, 95, \ldots$. This was suggested in [41], see also [43, Sec. 4]. It is possible to add some intermediate quadrature rules with n nodes to this list which maintain the nestedness and have an "intermediate" degree of exactness n. This is possible for $n = 13, 25, 27, 29, 49, \ldots$.[6] Hence, these formulas are not available for an arbitrary number of quadrature nodes, but only for the values

$$1, 3, 7, 13, 15, 25, 27, 29, 31, 49, \ldots \qquad \text{for } [-1, 1] \text{ with uniform weight.}$$

The growth rule we always choose for this family is such that each of the formulas in the family appears once, i.e. $p(1) = 1,\ p(2) = 3,\ p(3) = 7,\ p(4) = 13$, and so on.

A similar procedure is possible for integrals over $(-\infty, \infty)$ with Gaussian weight, too. Here, one starts with the 1- and 3-point Gauss-Hermite quadrature formulas and then extends these formulas keeping the property of nestedness. Such a family was developed in [18].

Let us now turn back to our original problem and use these quadrature formulas for our problem. The result is given in Figure 7.



(A) L^2-error, GP (B) Runtime, GP

FIGURE 7. Comparison of both methods with Gauss-Patterson quadrature formulas

The error is comparable with the error for slow growth Gauss grids from before in Figure 6C. Both the collocation and the GCS method are more efficient in terms of CPU time here. The difference between the collocation method and the GCS is less pronounced, but still very clearly present. All of the computations above are included in the work-precision diagram in Figure 8 below.

As a result, we conclude that the Gauss-Patterson rules perform best in our test problem, followed by slow growth Gauss rules. The error shows the same convergence behavior for each of the quadrature formulas although the ansatz spaces are different. For each of the cases, we observe that the splitting approach outperforms the straight-forward collocation method. As we have argued in Section 3.3, the standard Galerkin-only approach (14), (15) for the problem class (5) is in no way competitive, which is why we did not include it in our numerical tests.

5.3. **Parameters in a five-dimensional random space with various distributions.** Now we extend the above example by introducing three additional random variables $Y_3,\ Y_4 \sim \mathcal{N}(0, 1)$, $Y_5 \sim \mathrm{Beta}(1, \frac{3}{2})$ on which the parameters depend. Hence, the stochastic dimension is now $d = 5$. The orthonormal polynomials corresponding to the normally and Beta distributed

[6]The family introduced here is sometimes called the Genz-Gauss-Patterson (GGP_E) family. Under this name, it appears on the list available under the link [7].

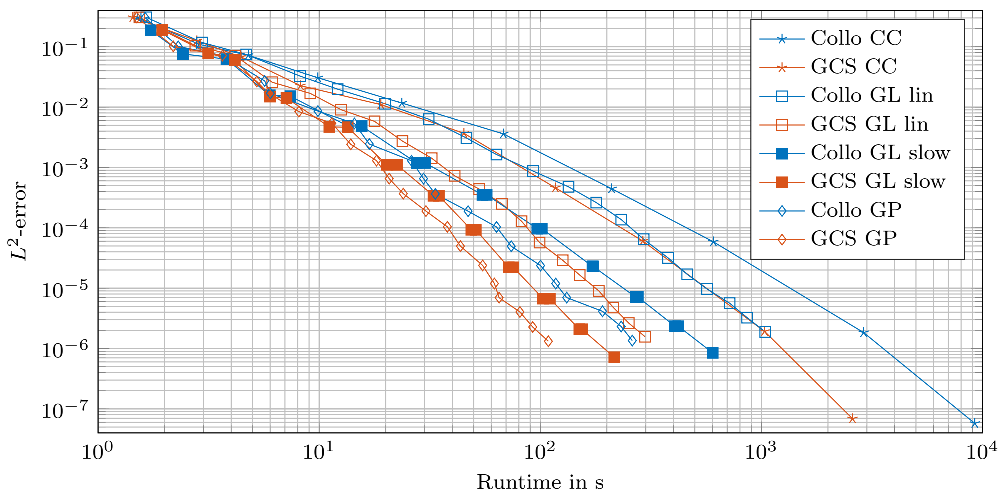


FIGURE 8. Work-precision diagram, all tests from this section

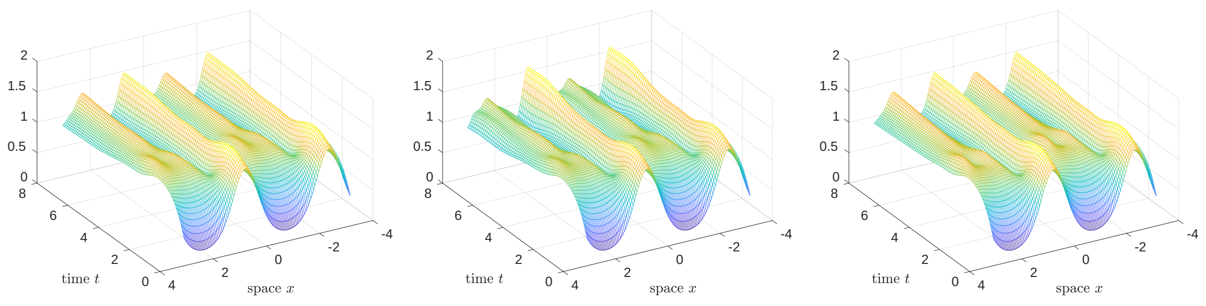
random variables are (up to a normalization) the Hermite and Jacobi polynomials, respectively. As before, $Y_1,\ Y_2 \sim \mathcal{U}(-1,1)$. The parameters are explicitly given by

$$a(y) = (1 + \delta y_1 y_2(1 + y_5 - \mathbb{E}[Y_5]))\bar{a}, \qquad b(y) = (1 + \tfrac{\delta}{3}(y_1 + y_2^2 + y_3))\bar{b},$$
$$f(y) = (1 + \delta y_2^2(1 + \tfrac{1}{2}y_4^2))\bar{f}, \qquad u_0(x, y) = (1 + 0.2y_1)^{\frac{1}{2}} \exp(\sin(2x)). \tag{40}$$

If we replaced y_i by $\mathbb{E}[Y_i]$ for $i = 3, 4, 5$, we would obtain the parameters from (38) before. The noise parameter is $\delta = 0.5$. Now we apply three methods to the problem (4) with parameters (40), namely GCS, collocation and a Monte Carlo sampling method with 1000 samples drawn from the distribution of Y. All of them are discretized in time by the Strang splitting method and in space by Fourier collocation, as in the section before. The time interval is $[0, 8]$ and the step-size in time is $\tau = 10^{-3}$. The Fourier collocation method uses $M = 2^{10}$ points in space. The sparse grid to be used for both the GCS and the collocation method is built from one-dimensional Gauss quadrature formulas with linear growth rule (level $\ell = 4$). Note that *Gauss quadrature* means the Gauss quadrature formula for the probability density function in the corresponding dimension, i.e. Gauss-Legendre in the first and second variable, Gauss-Hermite in the third and fourth variable, and Gauss-Jacobi in the fifth variable. The quantities P and Q from the previous sections are $P = 126$, $Q = 823$. We compare the computed expectations and standard deviations for the three methods, see Figures 9 and 10.

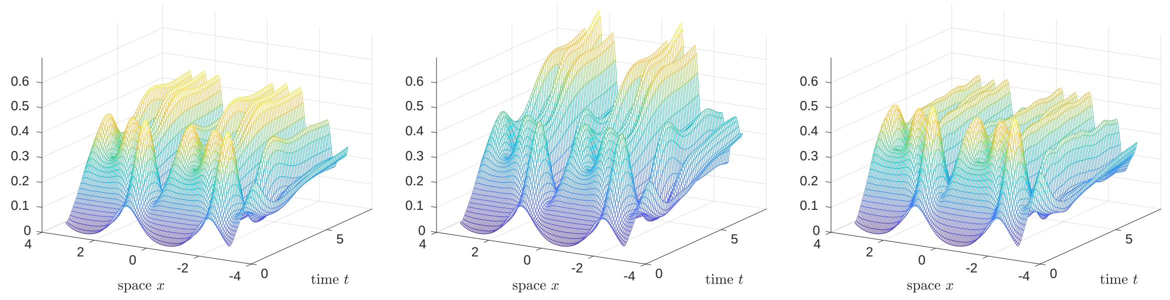
For each of the three methods, the expected value tends to a steady state with four pronounced peaks. We have already seen this in the previous subsection in Figure 3B and not surprisingly, the picture is nearly the same here – at least for the GCS and the MC method. The collocation method has a slightly different shape at the first and third peak and a slightly larger amplitude at the second and fourth peak. The behavior of the standard deviation in Figure 10 is even more remarkable.

First, we observe that GCS and MC produce a similar picture here, and that both yield a standard deviation with becomes stationary at the end of the time interval. Again, the collocation method yields a different picture: The standard deviation does not become stationary in this time interval, but increases even at time 8. This behavior seems to be wrong, since the other two methods suggest that the solution has already reached its steady state at this point. So far, the GCS solution seems to correspond very well to the Monte Carlo solution, in contrast to the collocation method.



(A) $|\mathbb{E}[u_{\mathrm{GCS}}(t,x,Y)]|$ (B) $|\mathbb{E}[u_{\mathrm{collo}}(t,x,Y)]|$ (C) $|\mathbb{E}[u_{\mathrm{MC}}(t,x,Y)]|$

FIGURE 9. Expected value of the solution of the random PDE (4) with the parameters from (40), computed with three different methods: GCS, collocation and Monte Carlo



(A) $\sigma[u_{\mathrm{GCS}}(t,x,Y)]$ (B) $\sigma[u_{\mathrm{collo}}(t,x,Y)]$ (C) $\sigma[u_{\mathrm{MC}}(t,x,Y)]$

FIGURE 10. Standard deviations for the solutions from Figure 9

Additionally, we now compare how the three methods can predict the solution for two special realisations of $Y = (Y_1, \dots, Y_5)$ which are given by

$$y^{(1)} = (0.6, -0.8, -1, 1, 0.6),\ y^{(2)} = (0.45, 0.95, 1, 1, 0.45) \in \Gamma = [-1,1]^2 \times (-\infty,\infty)^2 \times [0,1].$$

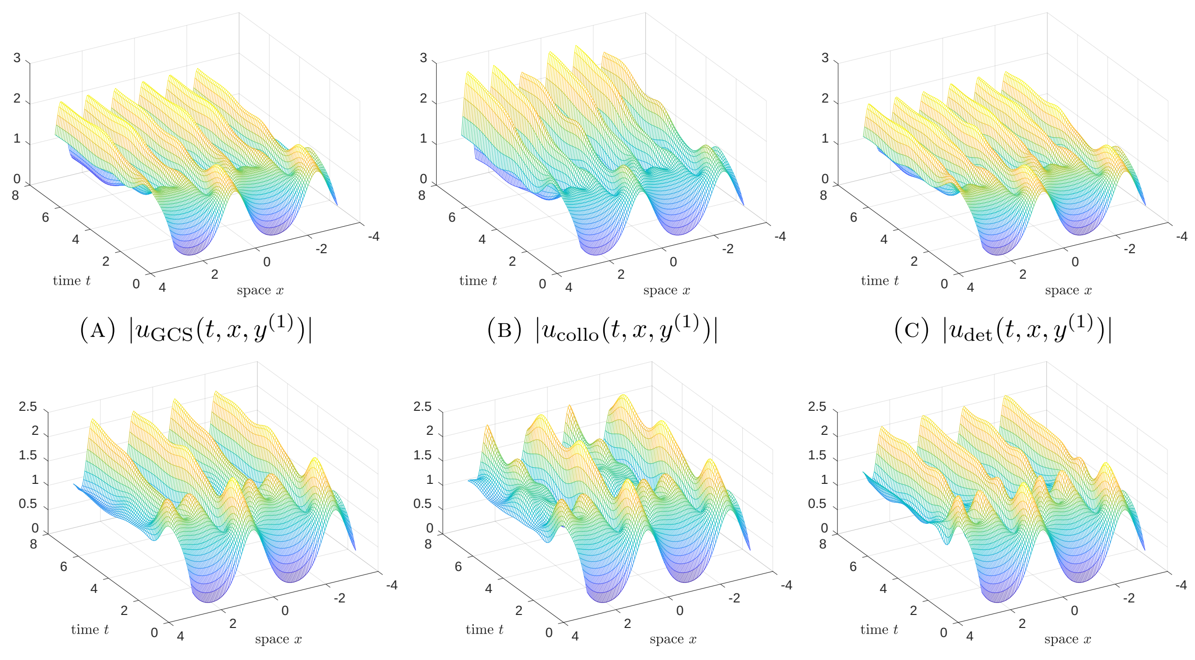
Note that the numerical solutions produced by the GCS and collocation methods can be evaluated at any given sample[7], whereas the Monte Carlo solution cannot predict the behavior of the solution for a given realization unless the realization coincides with one of the Monte Carlo samples. For the purpose of having reference solutions for the other two methods, we can of course use the solver for the deterministic problem with the parameters for the given realizations $y^{(1)}$ and $y^{(2)}$. Results are given in Figure 11 (the use of the deterministic solver is indicated by the subscript det).

We observe that for $y^{(1)}$, the deterministic solver and the GCS solver yield very similar results. Both methods reach a steady state with 6 pronounced peaks after time 6, whereas the collocation method yields a solution which still increases at time 8 and has larger peaks throughout. For $y^{(2)}$, the deterministic solver and the GCS method slightly differ, but are qualitatively comparable. The steady state reached at the end of the time interval has 4 pronouced peaks with approximate height 2. The collocation method yields a solution with more complicated dynamics, in particular the peaks given by the other methods do not gradually increase. For these two realizations, we observe that the GCS method produces much

[7]For the collocation method, we use *sparse* pseudospectral approximation as explained in the end of Section 3 to computed a PCE from the solutions in the collocation points.

more reliable solutions than the collocation method. This is underlined by the fact that the L^2-error of the solution $u(T, x, y)$ (now again with full y-dependency) at the end of the time interval is 0.63 for the GCS method and 0.78 for the collocation method, both compared to the Monte Carlo solution as described in the previous subsection. The sampled L^2-norm of the Monte Carlo solution itself is 2.83. Clearly, the error is not as small as desired, but we underline that this is a 5-dimensional problem with large noise parameter $\delta = 0.5$ and a rather long time interval $[0, 8]$. The time integration process in this example took only 387 seconds of wall-clock time for the GCS and 749 seconds for the collocation method. We also note that it is possible to construct samples $y \in \Gamma$ for which the collocation method performs better: In the special case that $y \in \Gamma$ is one of the collocation points used by the collocation method, the collocation method would compute the same solution as the deterministic solver. This is not surprising and follows directly from the construction of these methods.

The numerical examples presented in this section demonstrate that the GCS method is superior to standard Galerkin and collocation methods, at least in case of our model problem. Our next goal is to develop a rigorous stability and convergence analysis for this method.



(A) $|u_{\mathrm{GCS}}(t, x, y^{(1)})|$ (B) $|u_{\mathrm{collo}}(t, x, y^{(1)})|$ (C) $|u_{\mathrm{det}}(t, x, y^{(1)})|$

(D) $|u_{\mathrm{GCS}}(t, x, y^{(2)})|$ (E) $|u_{\mathrm{collo}}(t, x, y^{(2)})|$ (F) $|u_{\mathrm{det}}(t, x, y^{(2)})|$

FIGURE 11. Results of the GCS and collocation methods for two special realizations: $y^{(1)}$ (first row) and $y^{(2)}$ (second row), last column: results of solving the deterministic PDE (1) with parameters $a = a(y^{(i)}),\ b = b(y^{(i)}),\ f = f(y^{(i)})$ for $i = 1, 2$

6. Appendix

6.1. **Comparison of P and Q on full grids.**

Here we use the notations from Section 3.5. In particular, let P be the cardinality of $\Pi = \Pi^f_{\mathbf{m}}$ and let Q be the total number of nodes which belong to the quadrature rule to be specified in the following. Let us choose the linear growth rule $p(m) = m$ and Gauss quadrature rules on the one-dimensional interval $\Gamma = [-1, 1]$ (with uniform weight). The degree of exactness of the level m quadrature rule is well-known, namely $q(m) = 2m - 1$. The half-exact set of this quadrature rule is given by

$$\Pi^f_m = \{\phi_k \mid k = 0, \dots, \lfloor \tfrac{q(m)}{2} \rfloor = m - 1\},$$

where ϕ_k denotes the k-th Legendre (orthonormal) polynomial. This set has cardinality $P = m$. We also have $Q = m$ here. In d dimensions, the situation is similar: The half-exact set of the full grid which corresponds to the multi-index $\mathbf{m} = (m_1, \dots, m_d) \in \mathbb{N}_0^d$ is given by

$$\Pi^f_{\mathbf{m}} = \{\phi_{k_1} \cdots \phi_{k_d} \mid k_i = 0, \dots, \lfloor \tfrac{q(m_i)}{2} \rfloor = m_i - 1\}.$$

Therefore $P = Q = m_1 \cdots m_d$ in this special case.

Now we show an example where P is smaller than Q for a tensor product quadrature rule. If we change the growth rule to $p = p_{\text{exp}}$ from (16) and replace Gauss quadrature by Clenshaw-Curtis quadrature, then the degree of exactness is $q(m) = 2^{m-1}$ for $m \geq 1$. The growth rule is chosen in such a way that the level m quadrature nodes are contained in the level $m + 1$ quadrature nodes for every $m \in \mathbb{N}$. Now the situation is different, since we have $Q = p(m) = 2^{m-1} + 1$, whereas for $m \geq 2$ the half-exact set in one dimension is

$$\Pi^f_m = \{\phi_k \mid k = 0, \dots, \lfloor \tfrac{q(m)}{2} \rfloor = 2^{m-2}\}$$

and therefore contains only $P = 2^{m-2} + 1 < Q$ elements.

6.2. **Comparison of P and Q on sparse grids.**

In this example we illustrate the difference between P and Q on sparse grids. We consider a two-dimensional sparse grid quadrature rule for $\Gamma = [-1, 1] \times [-1, 1]$ with uniform weight built from one-dimensional Clenshaw-Curtis grids according to the growth rule $p(m) = p_{\text{exp}}(m)$ in both dimensions. We choose level $\ell = 2$, such that the Smolyak multi-index set is $\mathcal{I}_\ell = \{(2, 1), (1, 2), (2, 2), (3, 1), (1, 3)\}$. Hence, the Smolyak formula combines 5 full grids here. Their nodes are shown in Figure 12 (with $\xi = \frac{1}{\sqrt{2}}$).

Grid index	Corresponding nodes
$(2,1)$	$(1,0),\ (-1,0),\ (0,0)$
$(1,2)$	$(0,1),\ (0,-1),\ (0,0)$
$(2,2)$	$(-1,-1),\ (-1,0),\ (-1,1),$ $(0,-1),\ (0,0),\ (0,1),$ $(1,-1),\ (1,0),\ (1,1)$
$(3,1)$	$(-1,0),\ (-\xi,0),\ (0,0),\ (\xi,0),\ (1,0)$
$(1,3)$	$(0,-1),\ (0,-\xi),\ (0,0),\ (0,\xi),\ (0,1)$

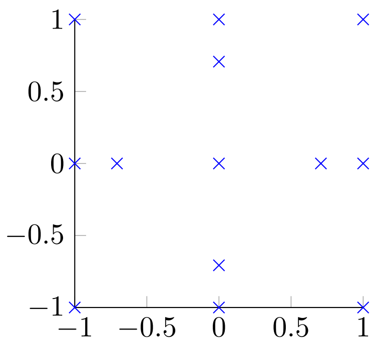


Figure 12. Contribution of different full grids to Smolyak's formula ($\xi = \frac{1}{\sqrt{2}}$)

In total (after removing duplicate nodes), we have $Q = 13$ quadrature points. It follows from (20) that the monomials

$$1,\ x,\ y,\ x^2,\ y^2,\ x^3,\ y^3,\ x^4,\ y^4,\ xy,\ x^2y,\ xy^2,\ x^2y^2$$

are exactly integrated by the sparse grid formula. From these polynomials, only the squares of

$$1,\ x,\ y,\ x^2,\ y^2,\ xy$$

are exactly integrated. Therefore, $P = 6 = \frac{Q-1}{2}$.
Finally, let us compare this to full grids: A full grid which integrates the squares of x^2 and y^2 exactly and which is built from one-dimensional Clenshaw-Curtis grids would require at least 5 points in each dimension, therefore $Q = 25$ points in total. The sparse grid rule above only requires roughly half as much. Its half-exact set would consist of $P = 9$ elements.

References

1. Ivo Babuška, Fabio Nobile, and Raúl Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Rev. **52** (2010), no. 2, 317–355. MR 2646806
2. Ivo Babuška, Raúl Tempone, and Georgios E. Zouraris, *Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation*, Comput. Methods Appl. Mech. Engrg. **194** (2005), no. 12-16, 1251–1294. MR 2121215
3. Joakim Bäck, Fabio Nobile, Lorenzo Tamellini, and Raul Tempone, *Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison*, Spectral and high order methods for partial differential equations, Lect. Notes Comput. Sci. Eng., vol. 76, Springer, Heidelberg, 2011, pp. 43–62. MR 3204803
4. Andrea Barth, Christoph Schwab, and Nathaniel Zollinger, *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numer. Math. **119** (2011), no. 1, 123–161. MR 2824857
5. Timothy Barth, *Non-intrusive uncertainty propagation with error bounds for conservation laws containing discontinuities*, Uncertainty quantification in computational fluid dynamics, Lect. Notes Comput. Sci. Eng., vol. 92, Springer, Heidelberg, 2013, pp. 1–57. MR 3202521
6. Hans-Joachim Bungartz and Michael Griebel, *Sparse grids*, Acta Numer. **13** (2004), 147–269. MR 2249147
7. John Burkardt, *1d quadrature rules for sparse grids*, `https://people.sc.fsu.edu/~jburkardt/presentations/sgmga_1d_rules.pdf` [Accessed: 03 July 2018].
8. Russel E. Caflisch, *Monte Carlo and quasi-Monte Carlo methods*, **7** (1998), 1–49. MR 1689431
9. Y. K. Chembo and Nan Yu, *Modal expansion approach to optical-frequency-comb generation with monolithic whispering-gallery-mode resonators*, Physical Review A **82** (2010), 033801.
10. Alina Chertock, Shi Jin, and Alexander Kurganov, *An operator splitting based stochastic galerkin method for the one-dimensional compressible euler equations with uncertainty*, Preprint.
11. C. W. Clenshaw and A. R. Curtis, *A method for numerical integration on an automatic computer*, Numer. Math. **2** (1960), 197–205. MR 0117885
12. Albert Cohen, Ronald DeVore, and Christoph Schwab, *Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs*, Found. Comput. Math. **10** (2010), no. 6, 615–646. MR 2728424
13. Albert Cohen, Ronald Devore, and Christoph Schwab, *Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's*, Anal. Appl. (Singap.) **9** (2011), no. 1, 11–47. MR 2763359
14. Patrick R. Conrad and Youssef M. Marzouk, *Adaptive Smolyak pseudospectral approximations*, SIAM J. Sci. Comput. **35** (2013), no. 6, A2643–A2670. MR 3129762
15. Paul G. Constantine, Michael S. Eldred, and Eric T. Phipps, *Sparse pseudospectral approximation method*, Comput. Methods Appl. Mech. Engrg. **229/232** (2012), 1–12. MR 2917505
16. Oliver G. Ernst, Antje Mugler, Hans-Jörg Starkloff, and Elisabeth Ullmann, *On the convergence of generalized polynomial chaos expansions*, ESAIM Math. Model. Numer. Anal. **46** (2012), no. 2, 317–339. MR 2855645
17. Erwan Faou, *Geometric numerical integration and Schrödinger equations*, Zurich Lectures in Advanced Mathematics, European Mathematical Society (EMS), Zurich, 2012.
18. Alan Genz and B. D. Keister, *Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight*, J. Comput. Appl. Math. **71** (1996), no. 2, 299–309. MR 1399898
19. Thomas Gerstner and Michael Griebel, *Numerical integration using sparse grids*, Numer. Algorithms **18** (1998), no. 3-4, 209–232. MR 1669959
20. Roger G. Ghanem and Pol D. Spanos, *Stochastic finite elements: a spectral approach*, Springer-Verlag, New York, 1991. MR 1083354
21. Michael B. Giles, *Multilevel Monte Carlo methods*, Acta Numer. **24** (2015), 259–328. MR 3349310

22. David Gottlieb and Dongbin Xiu, *Galerkin method for wave equations with uncertain coefficients*, Commun. Comput. Phys. **3** (2008), no. 2, 505–518. MR 2389809
23. T. Herr, K. Hartinger, J. Riemensberger, C.Y. Wang, E. Gavartin, R. Holzwarth, M.L. Gorodetsky, and T.J. Kippenberg, *Universal formation dynamics and noise of Kerr-frequency combs in microresonators*, Nature Photonics **6** (2012), 480–487.
24. Viet Ha Hoang and Christoph Schwab, *Sparse tensor Galerkin discretization of parametric and random parabolic PDEs—analytic regularity and generalized polynomial chaos approximation*, SIAM J. Math. Anal. **45** (2013), no. 5, 3050–3083. MR 3115458
25. Tobias Jahnke, Marcel Mikl, and Roland Schnaubelt, *Strang splitting for a semilinear Schrödinger equation with damping and forcing*, Journal of Mathematical Analysis and Applications **455** (2017), no. 2, 1051 – 1071.
26. Shi Jin and Zheng Ma, *The discrete stochastic Galerkin method for hyperbolic equations with non-smooth and random coefficients*, J. Sci. Comput. **74** (2018), no. 1, 97–121. MR 3742872
27. Clayton Webster John Burkardt, *Slow growth for gauss legendre sparse grids*, `https://people.sc.fsu.edu/~jburkardt/presentations/sgmga_gls.pdf` [Accessed: 03 July 2018].
28. Angela Kunoth and Christoph Schwab, *Analytic regularity and GPC approximation for control problems constrained by linear parametric elliptic and parabolic PDEs*, SIAM J. Control Optim. **51** (2013), no. 3, 2442–2471. MR 3064588
29. O. P. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification*, Scientific Computation, Springer, New York, 2010, With applications to computational fluid dynamics. MR 2605529
30. Matti Leinonen, *On applying stochastic galerkin finite element method to electrical impedance tomography*, 2015, pp. 38 + app. 129.
31. Meilin Liu, Zhen Gao, and Jan S. Hesthaven, *Adaptive sparse grid algorithms with applications to electromagnetic scattering under uncertainty*, Appl. Numer. Math. **61** (2011), no. 1, 24–37. MR 2735253
32. Christian Lubich, *From quantum to classical molecular dynamics: reduced models and numerical analysis*, Zurich Lectures in Advanced Mathematics, European Mathematical Society (EMS), Zürich, 2008. MR 2474331
33. L. A. Lugiato and R. Lefever, *Spatial dissipative structures in passive optical systems*, Phys. Rev. Lett. **58** (1987), 2209–2211.
34. G. Malenova, M. Motamed, O. Runborg, and R. Tempone, *A sparse stochastic collocation technique for high-frequency wave propagation with uncertainty*, SIAM/ASA J. Uncertain. Quantif. **4** (2016), no. 1, 1084–1110. MR 3544857
35. R. Mandel and W. Reichel, *A priori bounds and global bifurcation results for frequency combs modeled by the Lugiato-Lefever equation*, ArXiv e-prints (2016).
36. Hermann G. Matthies and Andreas Keese, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Comput. Methods Appl. Mech. Engrg. **194** (2005), no. 12-16, 1295–1331. MR 2121216
37. Mohammad Motamed, Fabio Nobile, and Raúl Tempone, *A stochastic collocation method for the second order wave equation with a discontinuous random speed*, Numer. Math. **123** (2013), no. 3, 493–536. MR 3018144
38. F. Nobile, R. Tempone, and C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal. **46** (2008), no. 5, 2309–2345. MR 2421037
39. Erich Novak and Klaus Ritter, *High-dimensional integration of smooth functions over cubes*, Numer. Math. **75** (1996), no. 1, 79–97. MR 1417864
40. ———, *Simple cubature formulas with high polynomial exactness*, Constr. Approx. **15** (1999), no. 4, 499–522. MR 1702807
41. T. N. L. Patterson, *The optimum addition of points to quadrature formulae*, Math. Comp. 22 (1968), 847–856; addendum, ibid. **22** (1968), no. 104, loose microfiche supp., C1–C11. MR 0242370
42. A. Pazy, *Semigroups of linear operators and applications to partial differential equations*, Applied Mathematical Sciences, vol. 44, Springer-Verlag, New York, 1983. MR 710486
43. Knut Petras, *Smolyak cubature of given polynomial degree with few nodes for increasing dimension*, Numer. Math. **93** (2003), no. 4, 729–753. MR 1961886
44. Mass Per Pettersson, Gianluca Iaccarino, and Jan Nordström, *Polynomial chaos methods for hyperbolic partial differential equations*, Mathematical Engineering, Springer, Cham, 2015, Numerical techniques for fluid dynamics problems in the presence of uncertainties. MR 3328389
45. S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. Nauk SSSR **148** (1963), no. 5, 1042–1053, Transl.: Soviet Math. Dokl. 4:240-243, 1963.

46. T. J. Sullivan, *Introduction to Uncertainty Quantification*, Texts in Applied Mathematics, vol. 63, Springer, Cham, 2015. MR 3364576
47. Radu Alexandru Todor and Christoph Schwab, *Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients*, IMA J. Numer. Anal. **27** (2007), no. 2, 232–261. MR 2317004
48. Grzegorz W. Wasilkowski and Henryk Woźniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, J. Complexity **11** (1995), no. 1, 1–56. MR 1319049
49. Norbert Wiener, *The Homogeneous Chaos*, Amer. J. Math. **60** (1938), no. 4, 897–936. MR 1507356
50. Kailiang Wu, Huazhong Tang, and Dongbin Xiu, *A stochastic Galerkin method for first-order quasilinear hyperbolic systems with uncertainty*, J. Comput. Phys. **345** (2017), 224–244. MR 3667611
51. Dongbin Xiu, *Fast numerical methods for stochastic computations: a review*, Commun. Comput. Phys. **5** (2009), no. 2-4, 242–272. MR 2513686
52. ———, *Numerical Methods for Stochastic Computations*, Princeton University Press, Princeton, NJ, 2010, A Spectral Method Approach. MR 2723020
53. Dongbin Xiu and Jan S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput. **27** (2005), no. 3, 1118–1139. MR 2199923
54. Dongbin Xiu and George Em Karniadakis, *Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos*, Comput. Methods Appl. Mech. Engrg. **191** (2002), no. 43, 4927–4948. MR 1932024
55. ———, *The Wiener-Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput. **24** (2002), no. 2, 619–644. MR 1951058

Department of Mathematics, Institute for Applied and Numerical Mathematics, Karlsruhe Institute of Technology, D-76128 Karlsruhe, Germany
Email address, Tobias Jahnke: `tobias.jahnke@kit.edu`
Email address, Benny Stein: `benny.stein@kit.edu`