AN OPTIMAL SAMPLING RULE FOR NONINTRUSIVE POLYNOMIAL CHAOS EXPANSIONS OF EXPENSIVE MODELS

Michael Sinsbeck[*] & *Wolfgang Nowak*

Institute for Modeling Hydraulic and Environmental Systems (LS3)/SimTech, University of Stuttgart, Stuttgart, Germany

Original Manuscript Submitted: 08/12/2013; Final Draft Received: 02/03/2015

In this work we present the optimized stochastic collocation method (OSC). OSC is a new sampling rule that can be applied to polynomial chaos expansions (PCE) for uncertainty quantification. Given a model function, the goal of PCE is to find the polynomial from a given polynomial space that is closest to the model function with respect to the L_2-norm induced by a given probability measure. Many PCE methods approximate the involved projection integral by discretization with a finite set of integration points. Our key idea is to choose these integration points through numerical optimization based on an operator norm derived from the discretized projection operator. OSC is a generalization of Gaussian quadrature: both methods coincide for one-dimensional integration and under appropriate problem settings in multidimensional problems. As opposed to many established integration rules, OSC does not generally lead to tensor grids in multidimensional problems. With OSC, the user can specify the number of integration points independently of the problem dimension and PCE expansion order. This allows one to reduce the number of model evaluations and still achieve a high accuracy. The input parameters can follow any kind of probability distribution, as long as the statistical moments up to a certain order are available. Even statistically dependent parameters can be handled in a straightforward and natural fashion. Moreover, OSC allows reusing integration points, if results from earlier model evaluations are available. Gauss-Kronrod and Stroud integration rules can be reproduced with OSC for the respective special cases.

KEY WORDS: *uncertainty quantification, polynomial chaos, stochastic collocation, arbitrary distribution, dependent parameters, nested quadrature rules*

1. INTRODUCTION

Consider a model function $M : \Omega \rightarrow \mathbb{R}$ and a random variable X with values in Ω. We consider a polynomial chaos [1–4] approximation of the form

$$M \approx P = \sum_{i=1}^{p} a_i \Psi_i, \tag{1}$$

where p is the number of terms, $a_1, \ldots, a_p \in \mathbb{R}$ are expansion coefficients, and $\Psi_1, \ldots, \Psi_p$ are the basis functions of a previously selected space $\mathcal{P}$ of polynomials on Ω.

The polynomial P is constructed, such that $P(X)$ is a good approximation of $M(X)$, and finding P is a matter of finding the expansion coefficients $a_1, \ldots, a_p$. In this work, we only consider nonintrusive methods, which determine the expansion coefficients with the information obtained from a finite number of deterministic function evaluations of M [5, 6]. We call the points at which M is evaluated *sample points* or *integration points*.

[*] Correspond to Michael Sinsbeck, E-mail: michael.sinsbeck@iws.uni-stuttgart.de

A general analytic expression for the expansion coefficients comes from an orthogonal projection of M onto $\mathcal{P}$. As this involves integration over Ω, one approach to calculating the coefficients is by approximating this integral with numerical quadrature rules [5, 7]. A second approach is regression, which requires the polynomial P to minimize the deviation from M in a finite set of sample points [8–11]. If the number of sample points and the number of terms in the polynomial approximation coincide, then the regression reduces to an interpolation. Regression and interpolation methods can also be understood as collocation methods [12, 13].

In all these cases, a key step is to select integration or sample points. The number of points should be as small as possible while the approximation in Eq. (1) should be as accurate as possible. For one-dimensional parameter spaces Ω, the integration points from Gaussian quadrature are regarded optimal [14, 15]. Gaussian quadrature, however, is not easily generalized to multidimensional domains [16, 17] and, to the best knowledge of the authors, for multidimensional parameter spaces, a single best sampling or integration method has not been found. A good sampling method should yield points that are well spread in the areas of high probability (densities). Additionally, it is desirable to have points that are more dense in the outer parts of the parameter space, because polynomial approximation on equally spaced points quickly leads to instabilities, an effect known as the Runge phenomenon [18].

Many different sampling rules for multidimensional spaces are available. The most common ones are tensor grid methods, sparse grid methods, the probabilistic collocation method (PCM), random sampling, quasi-Monte Carlo sampling (QMC), and monomial cubature rules. Details are provided in the following.

Tensor grids and derived methods require that the set of admissible parameter values is a Cartesian product $\Omega = \Omega_1 \times \cdots \times \Omega_d$ of one-dimensional sets and that the components of X are independently distributed. For each individual one-dimensional set $\Omega_i \subseteq \mathbb{R}$, integration points are determined according to a one-dimensional quadrature rule. Then all possible combinations of the points for all parameters are formed. If we select n_i sample points along each dimension $i \in \{1, \ldots, d\}$, then the tensor grid consists of $\prod_{i=1}^{d} n_i$ points. Tensor-grid quadrature rules inherit properties from one-dimensional quadrature rules, and it is straightforward to derive error estimates. The number of integration points, however, grows exponentially with the dimension, which is called the *curse of dimensionality*. Thus, for high-dimensional problems, tensor-grid methods quickly become infeasible. Also, tensor grids implicitly assume that the input parameters are statistically independent. If parameters are dependent, then tensor grids might place integration points in areas of Ω that are not relevant for the projection.

Sparse grid methods are based on tensor grids [19, 20]. A sparse grid is a combination of multiple tensor grids, such that functional features in each coordinate direction can be captured well, while keeping the number of points lower than in the full tensor grid. The lower number of points is achieved by investing fewer points in cross terms between the coordinates. For a fixed dimension and with increasing number of points, sparse grids have almost the same convergence behavior for integration as full tensor grids. However with increasing dimension, sparse grids of low order have increasingly large errors. This means that sparse grids in high dimensions are well suited only, if the number of points is high as well. Sparse grids have been used as sampling methods for polynomial chaos expansion (PCE), e.g., [5, 21]. Recent work shows that, rather than writing the PCE approximation as an integration problem and computing the integrals by a sparse grid quadrature, it is better to apply the Smolyak algorithm to the projection operator directly [22].

The PCM is a heuristic, based on the full tensor grid [9, 23–25]. Because of its simplicity it is widely used, e.g., [15, 26–28]. Aiming for an approximation of M with p terms in Eq. (1), the PCM selects p sample points from the full tensor grid, namely those with the highest weights in the associated full-grid quadrature rule, and performs a polynomial interpolation. In the selection of the points, it has also to be taken into account that the polynomial interpolation must be well-posed on these points. PCM reduces the number of sample points to the minimum. At the same time, the sample points tend to cluster in high-probability regions. This is typically in the center of the parameter space, while the corners stay empty. The clustering in the center can lead to problems with the Runge phenomenon, and does not adequately resemble the probability measure. The PCM would generally be able to select sample points for statistically interdependent parameters, but the authors are not aware of any publication that has followed this possible path.

As a nongrid-based method, sampling based on monomial cubature rules has been proposed [29, 30]. The idea is to select integration points according to multidimensional quadrature rules with high polynomial degree and a low number of integration points. In [29], four different quadrature rules are presented. These are of order 5 and 7

and are restricted to normally distributed input variables. For higher orders or different distributions than the normal distribution, no quadrature rule is explicitly given.

As opposed to the deterministic methods stated above, a couple of nondeterministic sampling methods have been suggested. Random sampling simply means that a number of points (now called realizations) is chosen according to the distribution of the input parameters in the sense of Monte Carlo (MC) methods [8]. This has the advantage that any sort of input distribution can be handled, as long as an efficient sampling method is available. To get reliable results despite the randomness of this method, different authors suggest using more than the minimal number of realizations [8]. This is called oversampling and leads to statistical regression between M and P.

As a variance reduction method within MC, it is also possible to apply quasi-Monte Carlo methods, such as Hammersley sampling [8, 31]. The Hammersley points generally have a lower discrepancy than randomly selected points. They are defined for the uniform distribution on hyper-cubes of arbitrary dimension.

In this paper, we present the optimized stochastic collocation method (OSC), which is a method of choosing integration points optimally. It is based on the same idea as the monomial cubature rules by [29, 30], namely sampling according to a multidimensional quadrature rule. The main difference is that the user does not have to select a quadrature rule manually. The optimal integration points and weights are determined by the method through formal minimization of an integration operator error norm.

A similar approach is found in [32]. Here, the model function is treated as a Gaussian random field with known mean and covariance structure. Under this assumption, the quadrature rule that minimizes the second moment of the integration error is constructed via numerical optimization. This is highly similar to optimal spatial design under the assumption of second-order geostatistics [33, 34]. Our approach only considers polynomial functions and minimizes the supremum of integration errors over a certain space of polynomials. That way, our method is related more closely to Gaussian quadrature. We obtain a strict quality measure over a small polynomial space, while the approach in [32] obtains a (soft) probabilistic quality measure over a possibly infinite-dimensional function space.

The idea of finding a quadrature rule by minimizing an operator norm is not new. The idea has already been pursued in the 1970s and it has been emphasized that a minimization problem such as the one presented in this work is so involved that no general analytical solution has been found [35]. Thus, we cannot expect to find an explicit expression for the nodes and weights. Instead, we have to employ a numerical optimization algorithm to find the minimum. Numerical optimizations of point clouds have recently been pursued by [32, 36, 37] and are common practice in the field of optimal spatial design, e.g., [33, 38].

The paper is organized as follows. In Section 2 we present the computational details of nonintrusive PCE approximations. Then, in Section 3 the OSC is derived, and in Section 4, various properties of the method are discussed. The efficiency of the method is demonstrated in Section 5 in various numerical examples. Finally, a summary and conclusions are given in Section 6.

2. NONINTRUSIVE PCE

Let $\Omega \in \mathbb{R}^d$ be a set of input parameter values. The uncertainty of these parameters is modeled as a multivariate random variable (a random vector) X with values in Ω and probability measure Γ. The measure Γ can be, e.g., a continuous measure with a probability density function (pdf) or a discrete measure in the form of a countable set of points with associated probabilities. The input parameters can be statistically dependent. In this case, the measure Γ is nonseparable.

Next, we introduce $L_2(\Omega, \mathcal{A}, \Gamma)$, the space of all real-valued, square integrable functions on Ω with respect to measure Γ, where $\mathcal{A}$ denotes the Borel σ-algebra of Ω. For brevity, we refer to this space as L_2. It is equipped with the inner product

$$\langle f, g \rangle_{L_2} := \int_\Omega f(x)\, g(x)\, \mathrm{d}\Gamma(x) \tag{2}$$

and the corresponding norm $\|f\|_{L_2} = \sqrt{\langle f, f \rangle_{L_2}}$. In the following, this inner product and norm will be referred to as $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, respectively.

PCE methods aim to approximate a model function M. We assume $M \in L_2$. Mostly, M is given in the form of numerical software that solves partial differential equations. We consider the model output component-wise and thus, without loss of generality, we can assume the output to be a scalar. The evaluation of M for a particular parameter vector is assumed to be computationally expensive. In practice, one evaluation of M might take up to several days of computation, even when using parallelization techniques and high-performance computing. Additionally, we assume that M is sufficiently smooth, such that it is reasonable to approximate it by a polynomial.

2.1 Orthonormal Basis and Ansatz Space

Let $\Psi_1, \Psi_2, \dots$ be a polynomial orthonormal basis (ONB) of L_2, i.e., each element Ψ_i is a polynomial, and for all $i, j \in \mathbb{N}$ we have

$$\langle \Psi_i, \Psi_j \rangle = \delta_{ij}, \tag{3}$$

where δ denotes the Kronecker delta.

There exist measures Γ for which L_2 does not have an ONB of polynomials. In these cases, the space of polynomials is not dense in L_2. For more details on this aspect and a list of conditions to avoid this case, see [39].

In practice, the ONB is constructed with the following properties:

- The first polynomial is the constant $\Psi_1(x) = 1$. This yields a convenient expression for the expectation of the polynomials

$$\mathbb{E}\left[\Psi_i(X)\right] = \int_\Omega \Psi_i(x)\, \mathrm{d}\Gamma(x) = \langle \Psi_i, \Psi_1 \rangle = \delta_{i1}. \tag{4}$$

- Each polynomial Ψ_i contains exactly one additional monomial x^α that is not contained in the previous polynomials $\Psi_1, \dots, \Psi_{i-1}$. Very often, the polynomials are ordered by degree.

In the following we assume that the considered ONBs have these properties. For approximating the model function M as in Eq. (1) we now select a number of terms p and define the ansatz space $\mathcal{P}$ as the span of the first p polynomials:

$$\mathcal{P} = \operatorname{span}\left\{\Psi_1, \dots, \Psi_p\right\}. \tag{5}$$

In this work we choose and fix the ansatz space before the model function is evaluated. More recent methods, called sparse PCE methods [40, 41], try to adaptively construct a basis of polynomials based on the model response, and we will come back to this aspect later on.

For the numerical construction of the ONB, one can start with any basis of $\mathcal{P}$ and orthonormalize it using the Gram-Schmidt process [42]. Alternatively, it is possible to find the orthonormal polynomials by solving linear systems [43]. In any case, for the construction of an ONB it is sufficient to know the statistical moments of X up to a certain order.

Apart from that, if Γ is separable, i.e., the input parameters are statistically independent, then the construction of the ONB can be done for each dimension separately. If orthonormal polynomials of high degree are needed, one might run into numerical stability problems. To improve stability, it is suggested to use algorithms based on three-term recurrence relations for orthogonal polynomials, see [44].

2.2 Truncation Error and Approximation Error

Since $M \in L_2$, we can expand it with respect to the basis $\{\Psi_1, \Psi_2, \dots\}$:

$$M = \sum_{i=1}^{\infty} a_i \Psi_i. \tag{6}$$

The best approximation (in terms of the L_2 norm) of M in $\mathcal{P}$ [Eq. (5)] is the orthogonal projection of M onto $\mathcal{P}$ and can be obtained by truncating this expansion, since the ansatz polynomials are mutually orthogonal. The best approximation thus is

$$P = \sum_{i=1}^{p} a_i \Psi_i. \tag{7}$$

When we calculate P by numerical techniques, we only get approximations of the expansion coefficients and thus obtain an approximated polynomial

$$\tilde{P} = \sum_{i=1}^{p} \tilde{a}_i \Psi_i. \tag{8}$$

The difference between $\tilde{P}$ and M can now be split into two parts: a truncation error and an approximation error

$$\begin{aligned} M - \tilde{P} &= \underbrace{M - P}_{\text{truncation error}} + \underbrace{P - \tilde{P}}_{\text{approximation error}} \\ &= \sum_{i=p+1}^{\infty} a_i \Psi_i + \sum_{i=1}^{p} \left(a_i - \tilde{a}_i\right) \Psi_i. \end{aligned} \tag{9}$$

The truncation error depends on the model function and on the terms that are used for expansion. The approximation error additionally depends on the numerical method used to determine the coefficients. Thanks to the orthogonality of the ONB, the two errors are orthogonal and their squared L_2 norm is additive:

$$\left\| M - \tilde{P} \right\|^2 = \sum_{i=p+1}^{\infty} a_i^2 + \sum_{i=1}^{p} \left(a_i - \tilde{a}_i\right)^2. \tag{10}$$

As said before, the ansatz space is chosen *a priori*. Thus, for a fixed model function, the truncation error is constant. Different sampling methods can then be compared by the approximation error they introduce.

2.3 The Discretized Projection Operator

Following a nonintrusive approach, a list of n sample points $x^{(1)}, \ldots, x^{(n)}$ in Ω is chosen, and the expansion coefficients are calculated using only the corresponding model response values $M\left(x^{(1)}\right), \ldots, M\left(x^{(n)}\right)$. We refer to the calculation rule that approximates the coefficients from a finite number of model evaluations as *integration rule* (even though it can be based on a regression approach). The corresponding operator that maps the model function M to a polynomial $\tilde{P}$ will be referred to as *discretized projection operator*.

Two choices for the discretized projection operator are common: *quadrature rules* and *regression*. By inserting the expansion from Eq. (6) into an inner product with one of the basis functions and using the orthogonality property of the basis, one obtains the analytical formula for the coefficients

$$a_i = \langle M, \Psi_i \rangle = \int_{\Omega} M(x)\, \Psi_i(x)\, \mathrm{d}\Gamma(x). \tag{11}$$

A quadrature rule approximates this integral with a discretized version of the form

$$\tilde{a}_i = \sum_{j=1}^{n} w_j M\left(x^{(j)}\right) \Psi_i\left(x^{(j)}\right), \tag{12}$$

with a list of integration points $x^{(1)}, \ldots, x^{(n)}$ and appropriately chosen weights $w_1, \ldots, w_n$ [45–47].

The regression approach starts from the fact that the best approximation minimizes the L_2-norm error between M and P:

$$\|P - M\|^2 = \int_{\Omega} \left(P(x) - M(x)\right)^2 \mathrm{d}\Gamma(x). \tag{13}$$

After replacing the integral by a quadrature rule, minimizing the expression

$$\sum_{j=1}^{n} w_j \left[\tilde{P}\left(x^{(j)}\right) - M\left(x^{(j)}\right)\right]^2 \tag{14}$$

leads to expansion coefficients $\tilde{a}_1, \ldots, \tilde{a}_p$. Again, $x^{(1)}, \ldots, x^{(n)}$ and $w_1, \ldots, w_n$ are integration points and weights that are chosen according to a quadrature rule. Older approaches used uniform weights [10] for regression, but more recent publications show that results can be improved by selecting appropriate weights [11]. In this work, we restrict the weights for regression to be non-negative. That way, the expression in Eq. (14) is guaranteed to be nonnegative, which is reasonable for the approximation of a norm. If weights were negative, the deviation between $\tilde{P}$ and M would be maximized in some points and minimization of Eq. (14) would not necessarily be well-posed.

If we perform regression with $n = p$, and if the sample points are spread properly in Ω, then the residues at the sampling points can be reduced to zero and the regression becomes an interpolation with

$$P\left(x^{(j)}\right) = M\left(x^{(j)}\right), \quad \text{for all } j = 1, \ldots, n. \tag{15}$$

The weights are irrelevant in this case, as long as they are positive.

The regression and interpolation approach are also called collocation method or stochastic collocation method. The same equations can be obtained by writing the equations in the model function in a weak formulation and using Dirac distributions as weighting functions. The corresponding sample points are then called collocation points [5, 12, 13].

The OSC method, presented in the following, yields a set of integration points and corresponding weights to be used in Eqs. (12) or (14).

3. THE OPTIMIZED STOCHASTIC COLLOCATION METHOD

In this section, we present the optimized stochastic collocation methods (OSC). As pointed out in Section 2.3, we can calculate the expansion coefficients accurately, once we are able to numerically evaluate integrals as in Eqs. (11) and (13). The OSC is a method of choosing integration points and weights for a quadrature rule. OSC is formulated as an optimization problem with an objective function that is adapted to the efficient approximation of PCE coefficients.

First we define the exact integral operator

$$I : L_2 \to \mathbb{R} : f \mapsto \int_\Omega f(x)\, \mathrm{d}\Gamma(x). \tag{16}$$

Now we try to find the quadrature formula that is closest to I in some sense. Quadrature formulas are of the form

$$\int_\Omega f(x)\, \mathrm{d}\Gamma(x) \approx \sum_{j=1}^{n} w_j f\left(x^{(j)}\right), \tag{17}$$

where $w_1, \ldots, w_n$ are real-valued weights and $x^{(1)}, \ldots, x^{(j)} \in \Omega$ are the integration points.

For a given list of points $\mathbf{x} = \left(x^{(1)}, \ldots, x^{(n)}\right)$ and weights $w = (w_1, \ldots, w_n)$, we thus define the discrete quadrature operator

$$Q_{(\mathbf{x},w)} : L_2 \to \mathbb{R} : f \mapsto \sum_{j=1}^{n} w_j f\left(x^{(j)}\right). \tag{18}$$

A similar representation of integration and quadrature as operators has first been given in [7].

3.1 Minimizing the Quadrature Operator's Error Norm

The key idea behind OSC is to find integration points and weights such that the discrete operator $Q_{(\mathbf{x},w)}$ [Eq. (18)] resembles the exact integration operator I [Eq. (16)] as close as possible. The distance between the two operators is measured with an operator norm.

In order to define an operator norm for I and $Q_{(\mathbf{x},w)}$, these operators have to be bounded. Therefore, we restrict both I and $Q_{(\mathbf{x},w)}$ to a finite-dimensional test space $\mathcal{T}$, a subspace of L_2. Once both operators are bounded, we can introduce the desired operator norm. The space of all bounded linear operators from $\mathcal{T}$ to $\mathbb{R}$ is denoted by $\mathfrak{L}(\mathcal{T},\mathbb{R})$. For an operator $A \in \mathfrak{L}(\mathcal{T},\mathbb{R})$, the induced operator norm on $\mathfrak{L}(\mathcal{T},\mathbb{R})$ is defined as

$$\|A\|_{\mathfrak{L}(\mathcal{T},\mathbb{R})} = \sup_{f\in\mathcal{T}} \frac{\|Af\|_{\mathbb{R}}}{\|f\|_{L_2}}. \tag{19}$$

In Section 3.3, we show how this operator norm can be evaluated in practice.

We are now ready to formulate the OSC. The procedure for determining optimal integration points $\mathbf{x}_{\mathrm{osc}}$ and weights w_{osc} by OSC is

1. Choose a finite-dimensional test space $\mathcal{T} \subseteq L_2$ and the number n of integration points.
2. Determine the optimal integration points and weights according to

$$(\mathbf{x}_{\mathrm{osc}}, w_{\mathrm{osc}}) = \underset{\substack{\mathbf{x}\in\Omega^n \\ w\in[0,\infty)^n}}{\operatorname{argmin}} \left\|I - Q_{(\mathbf{x},w)}\right\|^2_{\mathfrak{L}(\mathcal{T},\mathbb{R})}. \tag{20}$$

Remarks:

- This approach does not require us to discretize the optimization. While in the field of optimal spatial design, similar optimization problems are usually discretized and solved on a grid of candidate points, we regard the problem as a continuous optimization problem.
- The objective function is a multivariate polynomial. The functional form will be discussed in more detail in Section 3.3. The smoothness of the objective function suggests the use of gradient-based optimization algorithms.
- Minimizing the squared norm in step 2 is equivalent to minimizing the norm itself, because the norm is non-negative. By considering the square, the objective function becomes a sum of squares. This structure can be exploited by optimization algorithms, see Section 3.3.
- It may seem that, when the multidimensional integration problem is transferred into a multidimensional optimization problem, the level of difficulty remains the same. However, the optimization can be done without evaluating the model function. Under the assumption that the model function is computationally expensive, the optimization can still be beneficial. This will be further discussed in Section 4.3.
- The optimization has $n\,(d+1)$ degrees of freedom. For a fixed polynomial degree and increasing dimension, the number of degrees of freedom grows faster than the number of polynomial terms. This is an important aspect in the discussion in Section 4.3.
- The ansatz space $\mathcal{P}$ is not entering the optimization procedure in a direct fashion. However, the choice of $\mathcal{T}$ and n can only be done in a meaningful way, if $\mathcal{P}$ is selected first. This is discussed in Section 3.2.
- The integration points in the optimization are constrained to Ω^n. If Ω is just the support of the measure Γ, then it might have an irregular shape. In this case, and if the numerical software behind M admits it, it is advantageous to select Ω larger, such that it is a Cartesian product of intervals $\Omega = \Omega_1 \times \cdots \times \Omega_d$. We call this an augmented support of Γ. This has two advantages. First, the constraints are easier to implement in an optimization algorithm. Second, a bigger domain Ω potentially allows a smaller minimum in the optimization, and may help to increase the degree of the quadrature rule for the given n. An example is provided in the Appendix.
- The weights are forced to be non-negative. This has two reasons. First, as noted in Section 2.3, non-negative weights guarantee that the approximation in Eq. (14) is non-negative. Second, numerical tests showed that, if we allow negative weights, then the objective function has many local minima that are troublesome for the appropriate choice of optimization algorithms. The solutions in these local minima often have two or more sample points very close together with weights of large magnitude and opposite sign. By enforcing non-negative weights, the integration points are forced to spread in the domain.

In this work we do not address the issues of existence and uniqueness of the optimum. In practice, we will often be sufficiently satisfied if we achieve a suitably low value of the error norm.

3.2 Choice of Test Space and Number of Integration Points

In this section we provide details on an appropriate choice of $\mathcal{T}$ and n. As noted before, the choice of $\mathcal{T}$ and n has to be adapted to the structure of the ansatz space $\mathcal{P}$. It is intuitively straightforward and also reasonable to select the test space to be a space of polynomials, because the PCE methodology is based on polynomial approximation. Thus, one wants to recognize the polynomial components of M as well as possible. This is similar to Gaussian quadrature rules that also consider polynomials only.

Before we discuss how to select the dimension of $\mathcal{T}$, we provide some basic considerations on relations between $\mathcal{T}$ and n. A trivial lower bound for the number of integration points is $n \geq p$, with $p = \dim \mathcal{P}$. Otherwise, the image of the discretized projection operator is only a lower-dimensional subspace of the ansatz space $\mathcal{P}$, and the n sample points cannot even distinguish the p different ansatz functions. This leads to an effect called internal aliasing [48], which means that not even the elements of $\mathcal{P}$ can be projected correctly. A trivial upper bound for the number of integration points is $n \leq t$, with $t := \dim \mathcal{T}$. With t integration points it is always possible to reduce the error norm in Eq. (20) to zero, see e.g., [49, 50].

In practice, we can select n much smaller than this upper bound. This is made plausible by the following consideration: In order to reduce the objective function Eq. (20) to zero, one has to satisfy t equations. The number of degrees of freedom in the OSC optimization is $n\,(d+1)$. If we choose $\mathcal{T}$ and n, such that

$$t = n\,(d+1)\,, \tag{21}$$

we can hope to just have enough integration points to be able to satisfy all t conditions. We call Eq. (21) the *degrees of freedom condition* (DOF condition). This condition, however, has to be understood as a rule of thumb. Both cases exist where n has to be chosen greater or can be chosen smaller. Two examples are provided in the Appendix.

Now, we discuss two possible options to choose the test space, together with appropriate selections of n.

1. The first approach we call the rigorous approach. Recall the two types of integrals we want to approximate [Eqs. (11) and (13)]. A good starting point is to require $\tilde{P} = M$, if $M \in \mathcal{P}$. This means that all integrals of products of two elements of $\mathcal{P}$ have to be exact. We obtain the test space

$$\mathcal{T} = \operatorname{span}\left\{\Psi_i\Psi_j :\ 1 \leq i, j \leq p\right\}. \tag{22}$$

   We then select n large enough so that the operator norm is reduced to zero. A practical procedure is to first select n according to the DOF condition and numerically perform the optimization. If the smallest found value of the objective function is not small enough, then n can gradually be increased until it is large enough to reduce the objective function to a sufficiently low value.

2. The second approach we call the minimal approach. We set the number of integration points to its minimum, which is $n = p$. Then we select an appropriate test space. Now the quadrature rule is, in general, not able to calculate all necessary integrals exactly. The test space can be chosen either as in Eq. (22) or according to the DOF condition:

$$\mathcal{T} = \operatorname{span}\left\{\Psi_1, \ldots, \Psi_{n(d+1)}\right\}. \tag{23}$$

   The former treats all coordinate directions of Ω according to their importance in $\mathcal{P}$. The latter simply cuts off the ONB after $n\,(d+1)$ terms, which means that some coordinate directions are slightly preferred over others. The word *minimal* in *minimal approach* refers to the minimality of n, not of $\mathcal{T}$.

3.3 Implementation Details

In this section, we derive an expression for evaluating the squared operator error norm $\left\|I - Q_{(\mathbf{x},w)}\right\|^2_{\mathfrak{L}(\mathcal{T},\mathbb{R})}$ in Eq. (20) for any set of integration points $\mathbf{x}$ and the weights w.

As before, we define $t = \dim \mathcal{T}$ and represent the elements of $\mathcal{T}$ as coordinate vectors with respect to the ONB $\Psi_1, \ldots, \Psi_t$. Trivially, $\mathbb{R}$ is a space of dimension 1 and its elements are represented as themselves. Formally, the corresponding basis is 1, which also constitutes an ONB. Accordingly, both operators I and $Q_{(\mathbf{x},w)}$ can be represented as $1 \times t$ matrices. We denote the matrix representation of an operator A with respect to basis Ψ as ${}_{\Psi}A$. Since we represent everything with respect to ONBs, the operator norm in $\mathfrak{L}(\mathcal{T}, \mathbb{R})$ can be computed as the 2-norm of its matrix representation, which is degenerated to a vector in this case.

For the matrix representation of I we recall Eq. (4) and find

$$\begin{aligned} I\Psi_1 &= 1 \\ I\Psi_i &= 0, \quad \text{for } i > 1, \end{aligned} \tag{24}$$

and then

$${}_{\Psi}I = \mathrm{e}_1 = (1, 0, \ldots, 0)\,. \tag{25}$$

For $Q_{(\mathbf{x},w)}$ we find

$$Q_{(\mathbf{x},w)}\Psi_i = \sum_{j=1}^{n} w_j \Psi_i\left(x^{(j)}\right) \tag{26}$$

and then

$${}_{\Psi}Q_{(\mathbf{x},w)} = \left(\mathbf{\Psi}\left(\mathbf{x}\right) w\right)^{\top}, \tag{27}$$

with the $t \times n$ matrix

$$\mathbf{\Psi}\left(\mathbf{x}\right) := \begin{bmatrix} \Psi_1\left(x^{(1)}\right) & \ldots & \Psi_1\left(x^{(n)}\right) \\ \vdots & & \vdots \\ \Psi_t\left(x^{(1)}\right) & \ldots & \Psi_t\left(x^{(n)}\right) \end{bmatrix}.$$

We can calculate the squared operator norm of $I - Q_{(\mathbf{x},w)}$ with

$$\left\| I - Q_{(\mathbf{x},w)} \right\|^2_{\mathfrak{L}(\mathcal{T},\mathbb{R})} = \left\| \mathrm{e}_1 - \mathbf{\Psi}\left(\mathbf{x}\right) w \right\|_2^2, \tag{28}$$

which is a sum of squares. This structure can efficiently be exploited with Gauss-Newton-type optimization algorithms. The optimization can be accelerated by analytically implementing the partial derivatives of the objective function.

4. DISCUSSION OF THE METHOD

In this section we address various properties of OSC. These are related to the possibility to recycle integration points in the sense of nested integration (Section 4.1), the consistency with known integration rules (Section 4.2), and some limitations related to the required numerical optimization (Section 4.3).

4.1 Recycling of Integration Points

With OSC it is possible to recycle integration points, e.g., in the sense of nested integration rules. Assume we have already performed some model evaluations in earlier work. A fixed list of integration points $x^{(1)}, \ldots, x^{(n)}$ and the model responses $M\left(x^{(1)}\right), \ldots, M\left(x^{(n)}\right)$ are given. Next we want to add m more integration points $x^{(n+1)}, \ldots, x^{(n+m)}$, such that the quadrature rule with all $n + m$ integration points is an optimal extension of the given n points. Relevant situations where this occurs are listed below.

The reuse of integration points by OSC is straightforward. We use exactly the same objective function as before [Eq. (20)], only we fix the first n integration points in the optimization. The number of degrees of freedom is now $n + m\,(d + 1)$: Each of the new points has $d + 1$ degrees of freedom, while the recycled points are free only in their weights. Intuitively speaking, this means that by recycling points we can increase the order of the quadrature rule, but the contribution of recycled points is reduced by the factor $(d+1)$. Therefore, recycling of points becomes less effective in higher dimensions. The test space $\mathcal{T}$ can be selected analogously to the suggestions in Section 3.2. In this case, the DOF condition has a slightly different form:

$$t = n + m\,(d+1)\,. \tag{29}$$

Two possible applications for the reuse of integration points are

- A first application is, when the test space $\mathcal{T}$ changes. A lower-order PCE can be used as part of an error estimator, to predict whether a higher-order PCE is necessary. In this approach, PC expansions of increasing degree are constructed until an error estimator indicates that the degree is high enough. For these methods, it is highly desirable to reuse points. This idea is analogous to nestedness in quadrature rules. By reusing points, we construct a set of nested quadrature rules.

  A possible use is in sparse PCE expansions. For example, in [41] an adaptive sparse polynomial chaos approximation is proposed, using a sequential experimental design. Such sequential design could be improved by incorporating information about previous sample points in each iteration.

- A second application is when the measure Γ of the parameter distribution changes. This is the case, for example, when Bayes' theorem is applied for parameter inference [51]. After incorporating measurement data into the prior knowledge of the distribution, one obtains a posterior distribution that differs from the prior. The old integration points are generally not placed optimally with respect to the new measure and it is desirable to add more points, if the modes of prior and posterior lie far apart [52]. Another situation where the applied measure changes is, e.g., the so-called shifted PCE and the windowed PCE, see [53].

4.2 OSC as a Generalization of Known Quadrature Rules

A couple of known quadrature rules are special cases of OSC, i.e., they minimize the objective function in Eq. (20) for certain choices of the test space. We present a selection of three such known types of quadratures rules. In numerical tests, which are not further reported here, the authors were able to confirm that these quadrature rules are not only theoretical minima of the objective function, but can also practically be found by numerical optimization.

The first type of quadrature rules we examine is Gaussian quadrature (GQ). GQ is a one-dimensional quadrature rule and it is famous for maximizing its integration order. With n integration points, it is possible to exactly integrate the first $2n$ monomials $1, x, x^2, \ldots, x^{2n-1}$. This number of monomials satisfies the DOF condition. To reproduce the integration points and weights via OSC, we select a number n and then choose the test space according to the DOF condition. By definition, the integration points of GQ minimize the objective function in Eq. (20). We do not have to restrict ourselves to a specific GQ-rule, e.g., Gauss-Legendre, Gauss-Hermite, etc. Instead, the measure Γ can be chosen freely, as long as the ONB of the test space can be constructed. OSC is also capable of reproducing tensor products of Gaussian quadrature, if the underlying measure is separable and the ansatz space is chosen to be a tenspor product polynomial space. For nonseparable measures (i.e., for statistically dependent input parameters), OSC deviates from GQ rules, and provides more problem-adapted nontensor clouds of integration points, see Section 5.2.

Second, we examine Kronrod extensions and Gauss-Kronrod quadrature rules. Given a one-dimensional quadrature rule with n integration points, its Kronrod extension is the nested quadrature rule with additional $n+1$ points that has the highest possible degree [54]. A Gauss-Kronrod quadrature rule is a Kronrod extension of a GQ rule. When constructing a Kronrod extension, there are $3n+2$ degrees of freedom. Each old integration point yields one degree of freedom, namely its weight, while the $n+1$ new integration points are free in location and weights and thus have two degrees of freedom each. To construct a Kronrod extension with OSC, we make use of the ability to recycle integration points. Starting from an n-point integration rule, we select $\mathcal{T} = \{\Psi_1, \ldots, \Psi_{3n+2}\}$, which is according to the modified DOF-condition [Eq. (29)] with $m = n+1$. If the Kronrod extension exists, then by definition it globally minimizes our objective function [Eq. (20)]. Newton's method has already been used to find Gauss-Kronrod rules [55] and general Kronrod extensions [56]. OSC coincides with the approach in these two papers, if a Gauss-Newton algorithm is used for the minimization of Eq. (20). The main difference is that OSC is formulated for arbitrary dimension and arbitrary number of additional integration points, while Kronrod extensions are one-dimensional by definition and always add $n+1$ points to an n-point quadrature rule.

Last, we want to draw attention to multidimensional monomial quadrature rules in general. A monomial quadrature rule is a quadrature rule that integrates polynomials up to a certain degree exactly. Famous work on such rules goes back to Radon in 1948 [57] and Stroud in 1971 [49]. By construction, a monomial quadrature rule attains an operator norm of 0 in Eq. (20), if the test space is chosen correctly. This means that, if the numerical optimization is successful, then OSC can either reproduce these quadrature rules from literature, or we would find a quadrature rule that is different, but achieves the same polynomial degree with the same number of points. Numerical tests within our study revealed that in some cases there exists a continuum of quadrature rules that minimize the operator norm to 0, e.g., when the stochastic domain Ω and the measure Γ are rotationally symmetric.

4.3 Limitations of the Method

The OSC has an important limitation: one needs to solve a high-dimensional optimization problem to obtain the integration points and weights. The dimensionality of the optimization problem is $n\,(d+1)$. Thus, for many integration points in high dimensions, the optimization problem is increasingly difficult to solve. *The applicability of OSC is restricted by the availability of efficient and robust optimization algorithms.*

For a fixed polynomial degree and increasing dimension, the dimensionality of the optimization grows faster than the number of needed sample points. That means, no matter how expensive the model function is, with increasing dimension, there will be a point at which the optimization becomes more time-consuming than additional model evaluations.

In such case, it might be more efficient to use a simpler integration rule and accept a nonminimal number of integration points. Most other rules, like sparse grid rules, are much simpler in their construction than OSC and the set of integration points can be determined explicitly and easily.

The benefits of OSC become substantive for expensive model functions, where even a large optimization effort is overwhelmed by the computational savings or additional accuracy brought by the optimized integration rule.

Another issue is the practical problem of finding the global minimum. The objective function is a multivariate polynomial and it can be expected to have local minima. Our test cases confirmed this. The problem of local minima can be tackled by using multi-start optimization algorithms or global search algorithms. However, even if the optimization is repeated a couple of times, there is no guarantee that the global optimum has been found, unless the objective function has been reduced to its lowest attainable value, i.e., zero.

In Section 5.4, we report an experiment about the practical calculation time of the optimization and the necessary number of multistarts.

5. NUMERICAL EXPERIMENTS

As explained in Section 2.2, the total L_2-norm error between the model function and the surrogate polynomial can be split up into a truncation error and an approximation error. By changing the integration rule, but keeping the ansatz space for the expansion fixed, we can only influence the approximation error. In the following experiments, we assume that the model function M and the ansatz space $\mathcal{P}$ is already given. We then compare OSC to different existing integration rules with respect to their approximation error. The MATLAB code used to perform the experiments can be obtained from the first author upon request.

5.1 Two-Dimensional Independent Input

In the first experiment, X is uniformly distributed in the domain $[-1,1]^2$ and we consider the model function $M\left(x_1,x_2\right) = \exp\left(x_1+x_2\right)$. We approximate the model function by a polynomial of total degree 4, i.e., the expansion has $p=15$ terms. We compare (1) OSC to (2) full tensor grids with points from Gauss-Legendre quadrature, (3) PCM, (4) random sampling and (5) Hammersley sampling. The minimal integration rule for the tensor grid has 25

sample points. For all other methods, it has $n = p = 15$ points. The point locations of these minimal rules are shown in Fig. 1.

With each sampling method, we construct integration rules of different sample sizes and calculate the approximation error for the first 15 terms. In this and all following experiments, polynomial coefficients were calculated using the regression approach. The results from the quadrature approach are not shown in this chapter. In almost all experiments, regression was equally accurate or better. The approximation error is shown in Fig. 2 on the left. OSC method and the tensor grid show similar convergence behavior. However, OSC can be computed with fewer points than the minimal tensor grid; in this case, 15 points instead of 25. The tensor grid starts with a very small approximation error, which in this case is not of much use, because the truncation error is relatively large. This can be seen in Fig. 2 on the right, where the total error is shown. Note the huge difference in scale between the two plots. The PCM, random sampling and Hammersley sampling, converge at a slower rate.

In practice, it does not make much sense to increase the number of sample points without increasing the number of expansion terms as well. Otherwise, the truncation error dominates the total error as seen in Fig. 2 on the right. For a reasonable result, the two errors should decay at approximately the same rate. In experiment two, we repeat the above experiment but increase both the number of expansion terms p and the number of sample points n. We approximate by polynomials of total degree between 1 and 8. With increasing expansion degree, the truncation error decreases, while the approximation error potentially increases, as more coefficients have to be computed. This is evened out by using more and more sample points. For PCM, random sampling, and Hammersley sampling, the minimal number of points $n = p$ is chosen. The OSC is constructed with the rigorous approach (see Section 3.2), which results in slightly more points than the minimum. The minimal tensor grid for each degree has approximately twice as many points as there are terms in the expansion, because we are employing a basis of polynomials up to a certain total order, not a tensor-product polynomial space.

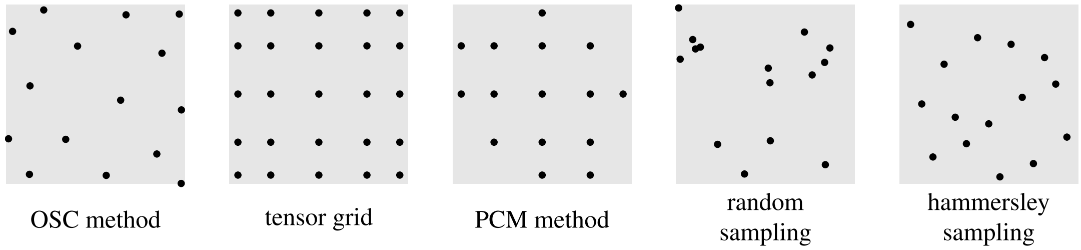

FIG. 1: Visual comparison of different sampling methods for two uniformly distributed random variables on $[-1, 1]^2$.

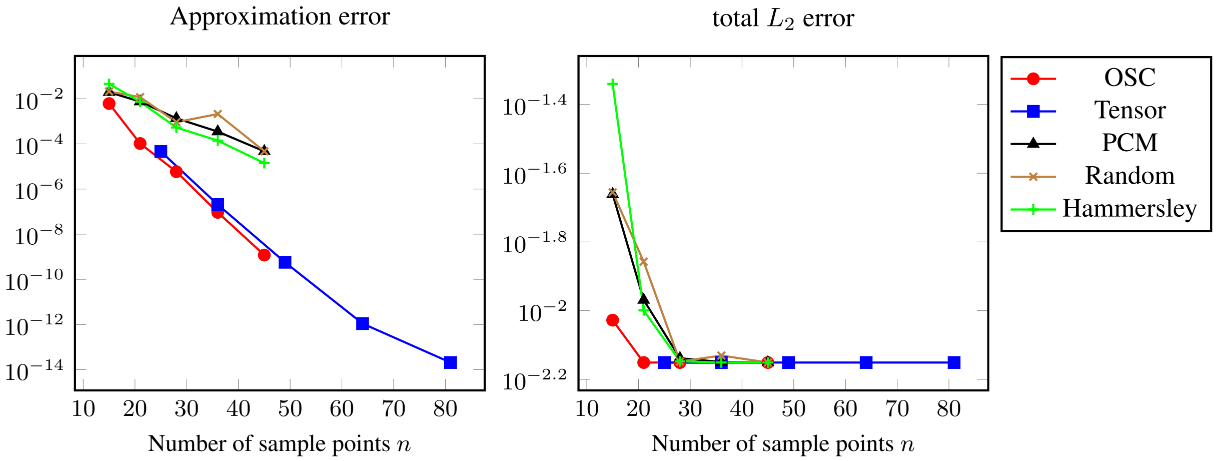

FIG. 2: Approximation error and total error observed in experiment one for the calculation of the 15 expansion coefficients for a 4-th order expansion of the model function $M(x_1, x_2) = \exp(x_1 + x_2)$.

The approximation error and the total error observed in experiment two are shown in Fig. 3. Now the approximation error of the individual data points are not directly comparable, as each expansion has a different number of terms. If we compare data points that belong to the same degree of expansion (e.g., the last point of each plot) then we see that tensor grid rules are much more accurate (an error of 10^{-9} versus 10^{-6}) but at the same time need more points. The plot in Fig. 3 on the right shows that the OSC overall is more useful. This is because the high approximation accuracy of the tensor grid rule is compromised by the relatively large truncation error. The additional sample points in the tensor grid are spent for an additional integration accuracy that is not worth the effort.

In experiment three, we repeat experiment two with the model function $M(x_1, x_2) = 1/\left(1 + 5x_1^2 + 5x_2^2\right)$. This function is similar to the one-dimensional Runge function [18], which is famous for the fact that polynomial interpolation is instable on an equidistant grid. The two errors are shown in Fig. 4. Here we see that the three sampling rules (PCM, random sampling, and Hammersley sampling) have increasing integration error. This is in line with the fact that the Runge function is difficult to interpolate. The sample points are not well-spread in the domain to approximate the growing number of coefficients reliably. These three methods are efficient in some cases, but not robust in the current example.

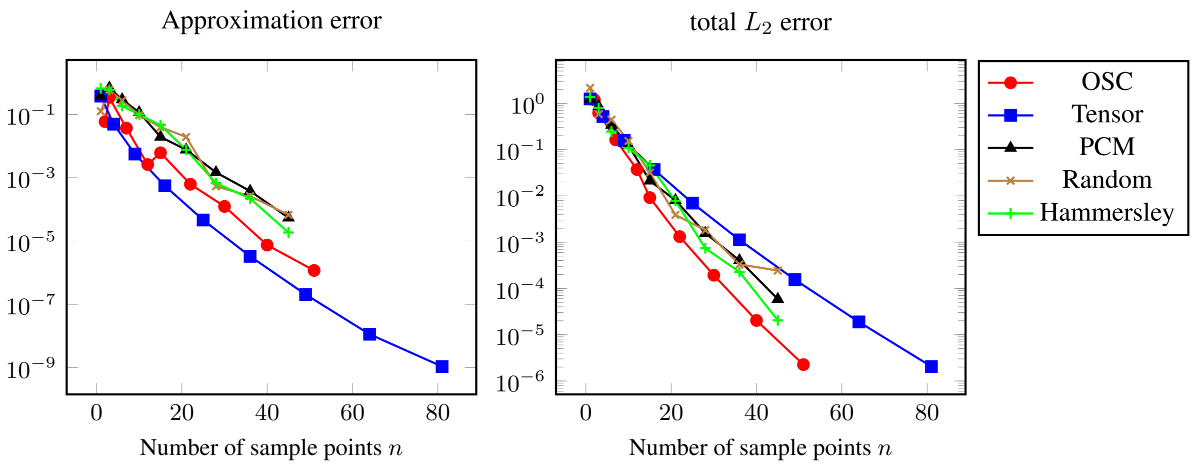

FIG. 3: Approximation error and total error observed in experiment two for expansions for degree from 0 to 8 for the model function $M(x_1, x_2) = \exp(x_1 + x_2)$.

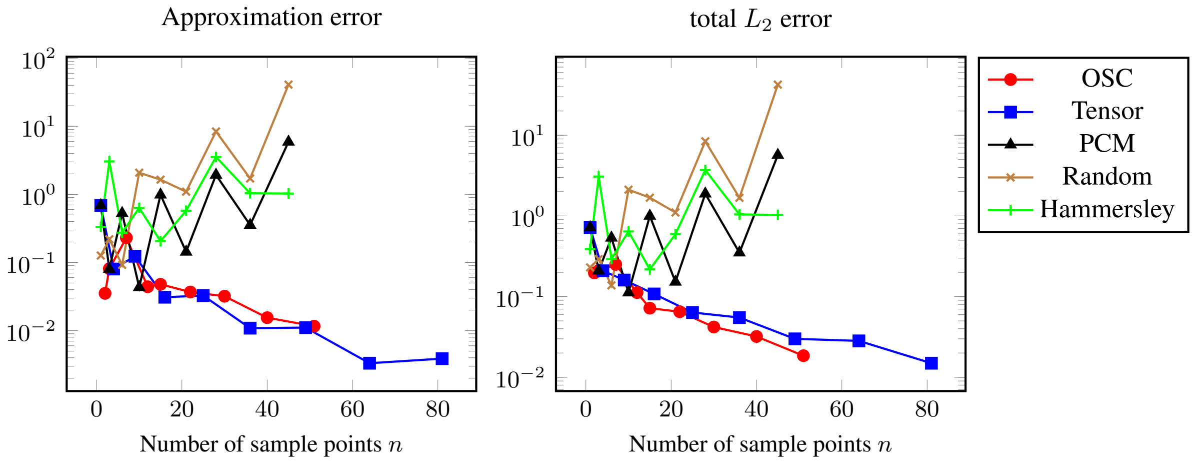

FIG. 4: Approximation error and total error observed in experiment three for expansions for degree from 0 to 8 for the Runge-type model function.

In summary, these three experiments demonstrated that tensor grid integration rules are very robust and accurate, but have too many sample points. The other methods get along with fewer points, but are not necessarily robust. The OSC method, however, is both robust and has a small number of points.

The authors are aware that, in all of these examples, the tensor grid integration rule has been used for a case that it is not optimal for. Many polynomial terms were neglected that could have been integrated exactly with a tensor grid. This demonstrates that, for an ansatz space that is not of tensor product form, tensor grids can be beaten in efficiency by a more adapted integration rule, such as OSC.

If the ansatz space was chosen to be of tensor product form (i.e., including all products of one-dimensional polynomials up to a certain one-dimensional order), then without doubt tensor product rules with points from Gaussian quadrature are very efficient and robust. Would a tensor grid be more efficient than OSC in this case? No, it would not, because by construction, OSC coincides with a tensor grid in this case: A tensor-product rule with points from Gaussian quadrature minimizes the objective function in Eq. (20).

5.2 Two-Dimensional Dependent Input

In this experiment (experiment four), we demonstrate that the OSC can handle input parameters with a dependent distribution. For this case, we select

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} R \cdot \cos(\theta) - 1 \\ R \cdot \sin(\theta) - 1 \end{bmatrix} \tag{30}$$

with two uniformly distributed random variables $R \sim \mathcal{U}(1, 2)$ and $\theta \sim \mathcal{U}(0, \pi/2)$. The values of X are again in the interval $[-1, 1]^2$, and the support of the distribution is a quarter of a ring around the point $(-1, -1)$ with radii 1 and 2. The distribution was discretized by a sample of size $10\,000$. All subsequent calculations (construction of ONB, calculation of error measures) are based on this sample, so we regard the sample as a discrete distribution. Thus, in the error measures, we do not get an additional Monte-Carlo error. Please note that the representation of X in Eq. (30) by a sample is not a requirement of OSC, but merely serves to define a fully accurate reference solution by Monte Carlo.

Figure 5 shows the resulting sampling points determined by different sampling methods. Tensor-grid-based methods have difficulties to resemble the input distribution properly. With two exceptions, the sample points of OSC all lie within the support of the distribution of X. This is noteworthy, since in the optimization no explicit constraints were imposed, following the idea of an augmented support for the integration (see remark in Section 3.1). The two outliers can be explained by the fact that the functions of the test space $\mathcal{T}$ are defined outside the support as well. Since the functions are smooth, even points outside the support are informative about the model function M.

As a fifth experiment, we perform a PCE with increasing order for the model function $M(x_1, x_2) = \exp(x_1 + x_2)$ (as in experiments one and two) and otherwise use the problem setting defined in experiment four. For polynomial orders from 0 to 8, the two errors are shown in Fig. 6. OSC clearly outperforms the other sampling methods. Compared to the tensor grid, OSC needs fewer sample points and is more accurate.

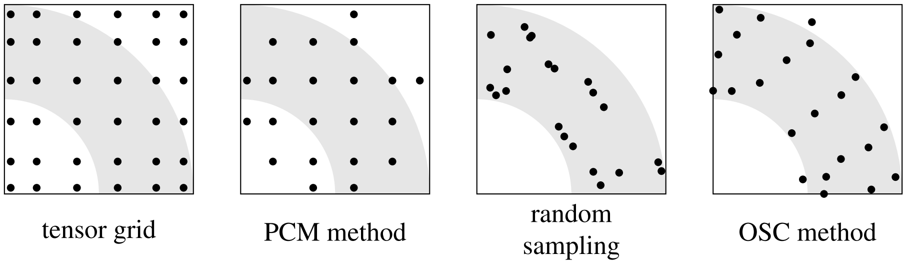

FIG. 5: Sampling points generated by different methods for a dependent distribution in experiment four. The gray area indicates the support of the distribution. The probability density function is not uniform on this area.

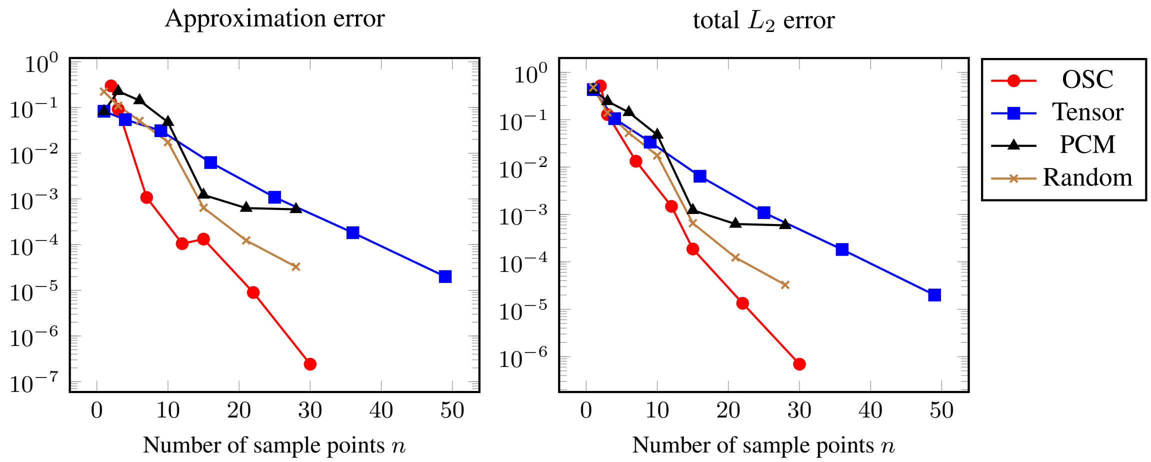

FIG. 6: Approximation error and total error observed in experiment five for expansions for degree from 0 to 8 for the model function $M(x_1, x_2) = \exp(x_1 + x_2)$ with a dependent input distribution.

In the example used for experiments four and five, the dependency in the parameter distribution can be removed by transforming the model function to the (R, θ) space. In theory, any continuous dependent distribution can be transformed to an independent distribution (e.g., via the Rosenblatt transform [58]). In practice, however, this can be tedious, if the distribution is given as a pdf. For discrete distributions, such a transform exists, too [59], but its inverse transform is not continuous and, in most cases, the transform removes the spatial information from the distribution.

5.3 Nested Integration

In experiment six, we demonstrate how OSC is able to produce nested point sets. We mimic a case of Bayesian inference, i.e., we use two different distributions. In the first step, we generate a set of points for a prior distribution, which we select to be $\mathcal{N}(0, 1)$ for both parameters independently. In the second step, we add more points and use a different distribution, namely $\mathcal{N}(1, 0.5^2)$. Such types of distributions could occur when Bayesian inference with a linear model and normally distributed measurement errors is performed. In Fig. 7, the two distributions are indicated by the gray shading in the background. The distributions are chosen such that some of the prior integration points lie in the high-probability region of the posterior, which means that some interactions between the prior and the posterior point cloud can be expected.

We generate 12 integration points for the prior with a test space of polynomials up to degree 7. For the posterior, we add 11 points with a test space of polynomials up to degree 8. The integration points are shown in Fig. 7. On the

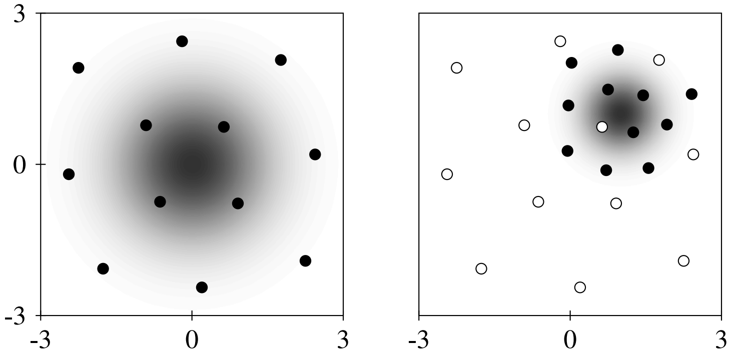

FIG. 7: Prior (left) and posterior (right) distribution and accordingly placed integration points for experiment six.

left, the 12 prior integration points are shown. On the right, the prior integration points are shown in white and the newly added points are shown in black.

As expected, the newly added points interact with the prior points and leave gaps around them. It is clear that the 23-point rule is not the best integration rule one can obtain with 23 points, but it is the best possible extension of the preset 12 points. If the posterior distribution was known beforehand, then then we could have generated a 15-point rule with the same test space. The recycling of the old points saved us $12/(d+1) = 4$ points [Eq. (29)].

5.4 Optimization Effort and Robustness

The dimension of the optimization problem is $n\,(d+1)$, and increases with both dimension and number of points. In experiment seven, we perform the optimization in different dimensions (number of random input variables) and for various numbers of integration points to provide a rough idea about the computation time.

For all of the following calculations, we assume a uniformly distributed input X in a unit cube $[-1,1]^d$. For various dimensions, we construct ansatz spaces of different total degrees. An ansatz space of polynomials up to total degree g in d dimensions has a basis of

$$p = \frac{(d+g)!}{d!g!} \tag{31}$$

ansatz functions. For each case, we perform optimizations with the minimal number of points $n = p$ and the test space according to Eq. (22) (see Section 3.2). To get a robust estimate, the computation time is averaged over 10 optimization runs for each case. The initial conditions are point clouds drawn randomly from the distribution of X. Our implementation uses the objective function as derived in Section 3.3. Additionally, the partial derivatives of the objective function are implemented analytically. For the optimization, MATLAB's function `lsqnonlin` was used, which is based on the interior trust region algorithm described in [60]. The optimization was performed on one core of a desktop computer with 3.10 GHz. Results are summarized in Table 1. Figure 8 shows a plot of the optimization time in terms of degrees of freedom. Data points from different dimensions are put together in this plot. In the range of up to 100 degrees of freedom, the optimization time increases approximately linear (with a slope of approximately 1 in the log-log plot). Beyond that point, the slope increases, indicating roughly a quadratic or faster increase. However, our data are sparse in this area.

In the explored range, optimization time is on the order of seconds, and only one of the cases takes more than one hour. This has to be compared with the evaluation time of the model function M per integration point. Depending on the complexity of M, this can easily be in the range of days, even on much faster and parallel architectures.

Finally, in experiment eight, we investigate the numerical robustness of the optimization. How sensitive is the optimization result to the random initial conditions? We compare three cases, all of dimension 2 and all with a test

TABLE 1: Optimization time for OSC for different polynomial degrees in different numbers of dimension, observed in experiment seven

Dimension	Degree	Number of integration points	Degrees of freedom	Optimization time [s]
1	1	2	4	0.02
	5	6	12	0.07
	10	11	22	0.21
2	1	3	9	0.03
	5	21	63	0.26
	10	66	198	2.14
5	1	6	36	0.29
	5	252	1 512	4 020.80
10	1	11	121	51.17
	2	66	726	2 171.10

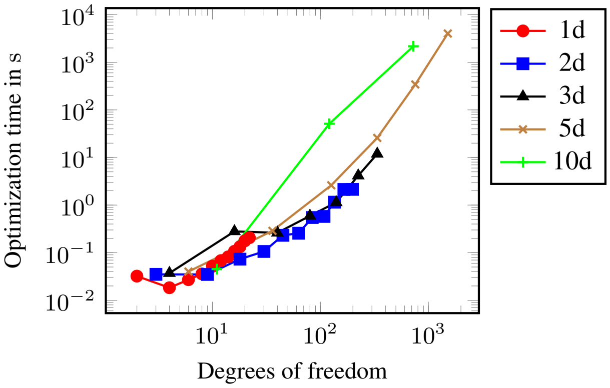

FIG. 8: Optimization time in terms of degrees of freedom.

space of polynomials up to degree 8. In the first case, the underlying distribution is uniform on the unit square. In the second case, we select a bimodal Gaussian mixture distribution: With a probability of $50/50$ the random parameter is sampled from one of two normal distributions $\mathcal{N}((0,0),I)$ and $\mathcal{N}((3,3),I)$. In the last case, the parameters are again uniformly distributed, but this time we generate 6 random points and insert them into the optimization as fixed points for point recycling (see Section 4.1), and ask the optimization to augment the initial 6 points.

For each of the three cases, we repeat the optimization 1000 times and record the optimization time and the achieved value of the objective function. Figure 9 shows scatter plots of this data. In the left plot, we can see that the objective function has three distinct local minima. The number of runs that reached the global minimum is 420 of 1000. In the second case, the objective function seems to have many more local minima and the number of successful runs is 278 of 1000. In the third case, very many runs did not find the global minimum and only 55 of 1000 runs reached the optimal point.

These results indicate that, for multimodal distributions and with point recycling, the optimization becomes practically more difficult. Local search algorithms do not guarantee to find the global minimum. Consequently, the computational effort for finding optimal points is the product of the average optimization time and the number of necessary multistarts. Also, in practice we have to accept that we can only improve quadrature rules, but do not necessarily find the best possible one. In these cases, global search techniques could be beneficial. For a practical guarantee for improvement, one can insert the closest alternative or best-in-class integration rule, and ask OSC for an adaptation to the more specific problem at hand.

In all three cases reported in Fig. 9, the computation time varied mainly within one order of magnitude with a few outliers in case 2 and 3, which are slower by a factor of about 100 compared to the fastest run.

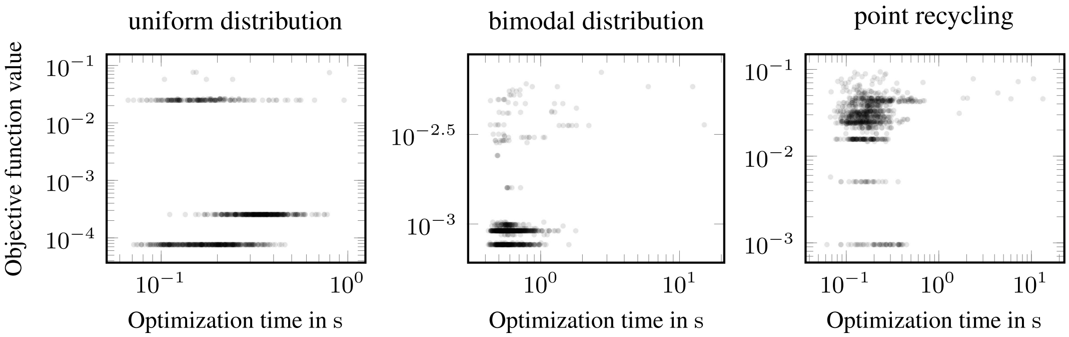

FIG. 9: Scatter plot of achieved value of the objective function versus calculation time.

6. CONCLUSIONS

In this work we introduced OSC, a new integration rule for the efficient, nonintrusive construction of polynomial chaos expansion methods. The method is adapted to the task of calculating PCE coefficients and yields an optimized set of integration points and weights. The integration points can be used for a quadrature approach or a weighted regression and are not generally of tensor grid structure. The OSC method can handle statistically-dependent input parameters and, by reusing integration points, nested integration rules can be constructed.

For the model functions tested here, OSC showed to be more robust than known sampling and integration methods with minimal number of points. Moreover, tensor grid rules are a special case of OSC, if the expansion is based on a tensor product polynomial space with a separable measure. For expansions with other types of polynomial spaces or with statistically dependent input parameters, OSC deviates from tensor grid rules and yields more efficient results.

In high dimensions, the optimization for finding the integration rule can take a considerable amount of time, but the number of necessary model evaluations can be reduced. Thus the OSC method shows its strength when applied to computationally expensive models, where each single model evaluation matters. The method can reduce the number of model evaluations to the minimum and it can reuse given information, if available.

In Section 1, we mentioned that sparse grids are used most effectively, if the Smolyak algorithm is applied to the projection operator itself, rather than to the integration operator. A similar step could be done with OSC. While in this work we optimized nodes and weights to get the best possible quadratur operator, one could also try to find the nodes and weights, that yield the best projection operator. This is a subject of ongoing research.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of the German Research Foundation (DFG) for the CODECS project (DFG grant nO 805/3-1) and within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

REFERENCES

1. Wiener, N., The homogeneous chaos, *Am. J. Math.*, 60(4):897–936, 1938.
2. Ghanem, R. and Spanos, P., *Stochastic Finite Elements: A Spectral Approach*, Springer, New York, 1991.
3. Xiu, D. and Karniadakis, G. E., The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, 24:619–644, 2002.
4. Xiu, D., *Numerical Methods for Stochastic Computations—A Spectral Method Approach*, Princeton University Press, Princeton, 2010.
5. Xiu, D. and Hesthaven, J. S., High-order collocation methods for differential equations with random inputs, *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.
6. Hosder, S., Walters, R., and Perez, R., A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations, In *44th AIAA Aerospace Sciences Meeting*, Reno, Nevada, 2006.
7. Xiu, D., Efficient collocational approach for parametric uncertainty analysis, *Commun. Comput. Phys.*, 2(2):293–309, 2007.
8. Hosder, S. and Walters, R., Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables, In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii, 2007.
9. Isukapalli, S. S., Roy, A., and Georgopoulos, P., Stochastic response surface methods (SRSMs) for uncertainty propagation: application to environmental and biological systems, *Risk Analysis*, 18(3):351–363, 1998.
10. Choi, S.-K., Canfield, R., Grandhi, R., and Pettit, C., Polynomial chaos expansion with latin hypercube sampling for estimating response variability, *AIAA J.*, 42(6):1191–1198, 2004.
11. Xiong, F., Chen, W., Xiong, Y., and Yang, S., Weighted stochastic response surface method considering sample weights, *Struct. Multidisciplinary Optimization*, 43(6):837–849, 2011.
12. Mathelin, L. and Hussaini, M. Y., A stochastic collocation algorithm for uncertainty analysis, Tech. Rep. February, NASA, 2003.

13. Babuška, I., Nobile, F., and Tempone, R., A stochastic collocation method for elliptic partial differential equations with random input data, *SIAM J. Numer. Anal.*, 45(3):1005–1034, 2007.
14. Villadsen, J. and Michelsen, M. L., *Solution of Differential Equation Models by Polynomial Approximation*, Prentice-Hall International, Englewood Cliffs, NJ, 1978.
15. Sudret, B., Global sensitivity analysis using polynomial chaos expansions, *Reliability Eng. Syst. Safety*, 93(7):964–979, 2008.
16. Haber, S., Numerical evaluation of multiple integrals, *SIAM Rev.*, 12(4):481–526, 1970.
17. Jäckel, P., A note on multivariate Gauss-Hermite quadrature, *Working Paper*, May 16, 2005.
18. Runge, C., Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, *Z. Math. Phys.*, 46:224–243, 1901.
19. Gerstner, T. and Griebel, M., Dimension-adaptive tensor-product quadrature, *Computing*, 71(1):65–87, 2003.
20. Barthelmann, V., Novak, E., and Ritter, K., High dimensional polynomial interpolation on sparse grids, *Adv. Comput. Math.*, 12:273–288, 2000.
21. Nobile, F., Tempone, R., and Webster, C. G., A sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Num. Anal.*, 46(5):2309–2345, 2008.
22. Constantine, P. G., Eldred, M. S., and Phipps, E. T., Sparse pseudospectral approximation method, *Comput. Methods Appl. Mech. Eng.*, 229-232:1–12, 2012.
23. Webster, M., Tatang, M. A., and McRae, G. J., Application of the probabilistic collocation method for an uncertainty analysis of a simple ocean model, *MIT Joint Program on the Science and Policy of Global Change*, 1996.
24. Tatang, M. A., Pan, W., Prinn, R. G., and McRae, G. J., An efficient method for parametric uncertainty analysis of numerical geophysical models, *J. Geophys. Res.*, 102(D18):21925–21932, 1997.
25. Isukapalli, S. S., Uncertainty analysis of transport-transformation models, PhD thesis, State University of New Jersey, 1999.
26. Li, H. and Zhang, D., Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods, *Water Resources Res.*, 43(9):1–13, 2007.
27. Li, W., Lu, Z., and Zhang, D., Stochastic analysis of unsaturated flow with probabilistic collocation method, *Water Resources Res.*, 45(8):W08425, 2009.
28. Oladyshkin, S., Class, H., Helmig, R., and Nowak, W., A concept for data-driven uncertainty quantification and its application to carbon dioxide storage in geological formations, *Adv. Water Resources*, 34(11):1508–1518, 2011.
29. Wei, D., Cui, Z., and Chen, J., Uncertainty quantification using polynomial chaos expansion with points of monomial cubature rules, *Comput. Struct.*, 86(23-24):2102–2108, 2008.
30. Sun, G. R. and Xiong, F. F., Efficient sampling approaches for stochastic response surface method, *Adv. Mater. Res.*, 538-541:2481–2487, June 2012.
31. Hammersley, J., Monte Carlo methods for solving multivariable problems, *Ann. New York Acad. Sci.*, 86(3):844–874, 1960.
32. Minka, T., Deriving quadrature rules from Gaussian processes, Tech. Rep., Statistics Department, Carnegie Mellon University, 2000.
33. Müller, W., *Collecting Spatial Data*, Springer, Heidelberg, third edition, 2007.
34. Nowak, W., Measures of parameter uncertainty in geostatistical estimation and geostatistical optimal design, *Math. Geosci.*, 42(2):199–221, October 2009.
35. Charushnikov, V., On the problem of optimizing the coefficients of cubature formulas, *Siberian Math. J.*, 11(4):714–718, 1970.
36. Mousavi, S. E., Xiao, H., and Sukumar, N., Generalized Gaussian quadrature rules on arbitrary polygons, *Int. J. Num. Methods Eng.*, 82(1):99–113, 2010.
37. Pajonk, O., *Numerical Evaluation of Functionals Based on Variance Minimisation*, Diploma Thesis, Institute of Scientific Computing, Technische Universität Braunschweig, Germany, 2008.
38. Ucinski, D., *Optimal Measurement Methods for Distributed Parameter System Identification*, CRC Press, Boca Raton, 2005.
39. Ernst, O. G., Mugler, A., Starkloff, H.-J., and Ullmann, E., On the convergence of generalized polynomial chaos expansions, *ESAIM: Math. Model. Num. Anal.*, 46(2):317–339, 2011.
40. Blatman, G. and Sudret, B., Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression

approach, *Comptes Rendus Mécanique*, 336(6):518–523, 2008.

41. Blatman, G. and Sudret, B., An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, *Probab. Eng. Mech.*, 25(2):183–197, 2010.
42. Witteveen, J. A. S. and Bijl, H., Modeling arbitrary uncertainties using Gram-Schmidt polynomial chaos, In *44th AIAA Aerospace Sciences Meeting*, Reno, Nevada, 2006.
43. Oladyshkin, S. and Nowak, W., Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion, *Reliab. Eng. Syst. Safety*, 106:179–190, 2012.
44. Gautschi, W., Orthogonal polynomials (in MATLAB), *J. Comput. Appl. Math.*, 178(1-2):215–234, 2005.
45. Ghiocel, D. and Ghanem, R., Stochastic finite-element analysis of seismic soil-structure interaction, *J. Eng. Mech.*, 128(1):66–77, 2002.
46. Le Maître, O. P., Reagan, M. T., Najm, H. N., Ghanem, R., and Knio, O. M., A stochastic projection method for fluid flow—II. Random process, *J. Comput. Phys.*, 181(1):9–44, 2002.
47. Reagan, M. T., Najm, H. N., Ghanem, R., and Knio, O. M., Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection, *Combust. Flame*, 132(3):545–555, 2003.
48. Conrad, P. R. and Marzouk, Y. M., Adaptive Smolyak Pseudospectral Approximations, *SIAM J. Sci. Comput.*, 35(6):A2643–A2670, 2013.
49. Stroud, A., *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Englewood Cliffs, NJ, 1971.
50. Cools, R., Advances in multidimensional integration, *J. Comput. Appl. Math.*, 149(1):1–12, 2002.
51. Tarantola, A., *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, Philadelphia, 2005.
52. Oladyshkin, S., Class, H., and Nowak, W., Bayesian updating via bootstrap filtering combined with data-driven polynomial chaos expansions: methodology and application to history matching for carbon dioxide storage in geological formations, *Comput. Geosci.*, 17(4):671–687, 2013.
53. Paffrath, M. and Wever, U., Adapted polynomial chaos expansion for failure detection, *J. Comput. Phys.*, 226(1):263–281, 2007.
54. Patterson, T., The Optimum addition of points to quadrature formulae, *Math. Comput.*, 22(104):847–856, 1968.
55. Calio, F., Gautschi, W., and Marchetti, E., On computing Gauss-Kronrod quadrature formulae, *Math. Comput.*, 47(176):639, 1986.
56. Gautschi, W. and Notaris, S., Newton's method and Gauss-Kronrod quadrature, *Num. Integration*, 85:60–71, 1988.
57. Radon, J., Zur mechanischen Kubatur, *Monatshefte Math.*, 52(4):286–300, 1948.
58. Rosenblatt, M., Remarks on a multivariate transformation, *Ann. Math. Stat.*, 23(3):470–472, 1952.
59. Brockwell, A. E., Universal residuals: A multivariate transformation, *Stat. Probab. Lett.*, 77(14):1473–1478, 2007.
60. Coleman, T. and Li, Y., An interior trust region approach for nonlinear minimization subject to bounds, *SIAM J. Optimization*, 6(2):418–445, 1996.

APPENDIX A. AUGMENTED SUPPORT

We provide a simple example to show that, by setting Ω to a larger set than the support of Γ, one can increase the degree of a quadrature rule.

Let Γ be the uniform distribution on $[-2,-1] \cup [1,2]$. We seek to construct a quadrature rule that is exact for all linear functions. If we select the domain to be just the support of Γ, namely $\Omega = [-2,-1] \cup [1,2]$, then at least two integration points are needed. Additionally, an optimization on such a domain is tedious. If, however, we select $\Omega = [-2,2]$ and if M can be evaluated everywhere on this interval, then we can find a quadrature rule with only one integration point, namely the expected value, which is 0 here.

APPENDIX B. THE DOF CONDITION CAN BOTH OVER- AND UNDERESTIMATE THE NUMBER OF NECESSARY INTEGRATION POINTS

Here we give two examples of test spaces, where the number of integration points needed for exact integration differs from the number suggested by the DOF condition.

The first example is described in [57]. For a uniform distribution with $d = 2$ and $\mathcal{T} = \mathrm{span}\{1, x_1, x_2, x_1^2, x_1 x_2, x_2^2\}$, it is $t = 6$ and the DOF condition suggests choosing $n = 2$. However, no matter where the two points are placed in the domain, there always exists one strictly non-negative function in $\mathcal{T}$ that is zero in both points. Thus, no quadrature rule with $n = 2$ exists that is exact for all functions in $\mathcal{T}$.

For the second example, assume a two-dimensional tensor product of Gaussian quadrature rules with two points in each direction on a separable measure, $d = 2$, $n = 4$ and the DOF condition suggests that such a quadrature rule is exact for a test space of dimension $n\,(d + 1) = 12$. In fact this quadrature rule is exact for all polynomials up to order 3 in each coordinate direction, which yields $t = 16$, which is a higher order of accuracy than the DOF condition suggests. Tensor products of Gaussian quadrature rules have the property to be exact for $t = n \cdot 2^d$ test functions, because the factor of 2 multiplies for each dimension.