MATHICSE
Mathematics Institute of Computational Science and Engineering
School of Basic Sciences - Section of Mathematics

MATHICSE Technical Report
Nr. 04.2015
March 2015

An adaptive sparse grid algorithm for elliptic PDEs with lognormal diffusion coefficient

Fabio Nobile, Lorenzo Tamellini, Francesco Tesei, Raúl Tempone

http://mathicse.epfl.ch

Phone: +41 21 69 37648

Address:
EPFL - SB - MATHICSE (Bâtiment MA)
Station 8 - CH-1015 - Lausanne - Switzerland

Fax: +41 21 69 32545

An adaptive sparse grid algorithm for elliptic PDEs with lognormal diffusion coefficient

F. Nobile, L. Tamellini, F. Tesei and R. Tempone

Abstract In this work we build on the classical adaptive sparse grid algorithm (*T. Gerstner and M. Griebel, Dimension-adaptive tensor-product quadrature*), obtaining an enhanced version capable of using non-nested collocation points, and supporting quadrature and interpolation on unbounded sets. We also consider several profit indicators that are suitable to drive the adaptation process. We then use such algorithm to solve an important test case in Uncertainty Quantification problem, namely the Darcy equation with lognormal permeability random field, and compare the results with those obtained with the quasi-optimal sparse grids based on profit estimates, which we have proposed in our previous works (cf. e.g. *Convergence of quasi-optimal sparse grids approximation of Hilbert-valued functions: application to random elliptic PDEs*). To treat the case of rough permeability fields, in which a sparse grid approach may not be suitable, we propose to use the adaptive sparse grid quadrature as a control variate in a Monte Carlo simulation. Numerical results show that the adaptive sparse grids have performances similar to those of the quasi-optimal sparse grids and are very effective in the case of smooth permeability fields. Moreover, their use as control variate in a Monte Carlo simulation allows to tackle efficiently also problems with rough coefficients, significantly improving the performances of a standard Monte Carlo scheme.

F. Nobile, L. Tamellini, F. Tesei
SB-MATHICSE-CSQI-EPFL, Station 8, CH-1015, Lausanne, Switzerland e-mail: fabio.nobile,lorenzo.tamellini,francesco.tesei@epfl.ch

R. Tempone
SRI Center for Uncertainty Quantification in Computational Science and Engineering, KAUST, Thuwal, Saudi Arabia e-mail: raul.tempone@kaust.edu.sa

1 Introduction

In this work we consider the problem of building a sparse grid approximation of a multivariate function $f(\mathbf{y}) : \Gamma \to V$ with global polynomials, where Γ is an N-dimensional hypercube $\Gamma = \Gamma_1 \times \Gamma_2 \times \ldots \times \Gamma_N$ (with $\Gamma_n \subseteq \mathbb{R}$, $n = 1,...,N$), and V is a Hilbert space [6, 3, 28, 25, 1]. We also assume that each Γ_n is endowed with a probability measure $\rho_n(y_n)dy_n$, so that $\rho(\mathbf{y})d\mathbf{y} = \prod_{n=1}^{N} \rho_n(y_n)dy_n$ is a probability measure on Γ. This setting is common in many optimization and Uncertainty Quantification problems, where sparse grids have been increasingly used to perform tasks such as quadrature, interpolation and surrogate modeling, since they allow for trivial parallelization and maximal reuse of legacy codes, with little or no expertise required by the end-user. While very effective for moderate dimensions (say $N \approx 10$), the basic sparse grid algorithms show a significant performance degradation when N increases (the so-called "curse of dimensionality" effect). The search for advanced sparse grid implementations, ideally immune to this effect, has thus become a very relevant research topic.

A general consensus has been reached on the fact that the "curse of dimensionality" should be tackled by exploiting the anisotropy of f, i.e. by assessing the amount of variability of f due to each parameter y_i and enriching the sparse grid approximation accordingly. Two broad classes of algorithms can be individuated to this end: those that discover the anisotropy structure "a-posteriori", i.e. at run-time, based on suitable indicators, and those based on "a-priori" theoretical estimates, possibly aided by some preliminary computations (we refer to the latter as "a-priori/a-posteriori" methods). A-priori algorithms based on a sharp theoretical analysis save the cost of the exploration of the anisotropy structure, while a-posteriori approaches are to a certain extent more flexible and robust. Focusing on the field of Uncertainty Quantification, examples of a-priori/a-posteriori algorithms can be found e.g. in [26, 4, 25], while the classical a-posteriori algorithm originally proposed in [17] has been further considered e.g. in [8, 27, 31].

A-posteriori sparse grid algorithms have always been used in the literature in combination with nested univariate quadrature rules, since this choice eases the computation of the anisotropy indicators, cf. [17]. In Uncertainty Quantification it is quite natural to choose univariate quadrature points according to the probability measures $\rho_n(y_n)dy_n$, see e.g. [1]: hence, one is left with the problem of computing good univariate nested quadrature rules for the probability measures at hand. While the case of the uniform measure has been thoroughly investigated and several choices of appropriate nested quadratures are available, like Leja, Gauss–Patterson or Clenshaw–Curtis points (see e.g. [24, 25] and references therein), non-uniform measures have been less explored. In the very relevant case of normal probability distribution a common choice is represented by Genz-Keister quadrature rules [16]; however, the cardinality of such quadrature rules increases very quickly when moving from one quadrature level to the following one, hence leading to an heavy computational burden when tensorized in a high-dimensional setting. The very recent work [23] develops instead generalized Leja quadrature rules for arbitrary measures on unbounded intervals: the main advantage of such quadrature rules over the Genz-Keister points is that two consecutive quadrature rules differ by one point only, rendering the Leja points more suitable for sparse grids construction. In this work we will approach the problem from a different perspective and propose a slight generalization of the classical a-posteriori adaptive algorithm that allows to use non-nested quadrature rules: this immediately permits to build adaptive sparse grids using gaussian-type quadrature nodes, which are readily available for practically every common probability measure. We will also consider different profit indicators and compare the performances of the corresponding adaptive schemes.

We will then test our version of the adaptive algorithm on a classical Uncertainty Quantification test problem, i.e. an elliptic PDE describing a Darcy flow in a porous medium, whose diffusion coefficient is modeled as a lognormal random field [5, 14, 7, 18, 9] and discretized by a Karhunen–Loève expansion. The covariance structure of the random field will be described by a tensor Matérn covariance model [11], which is a family of covariance structures parametrized by a scalar value ν that governs the smoothness of each realization of the random field and includes the Gaussian and the Exponential covariance structure as particular cases ($\nu = \infty$ and $\nu = 0.5$, respectively); more specifically, we will first consider the case $\nu = 2.5$, that results in fairly smooth random field realizations, and then move to the rough case $\nu = 0.5$, which leads to continuous but not differentiable field realizations. In both cases we will compare the performance of the adaptive sparse grid procedure with the "a-priori/a-posteriori" quasi-optimal sparse grid proposed in [5] for the same problem.

In the case $\nu = 2.5$, the lognormal random field can be very accurately described by including a moderate number of random variables in the Karhunen–Loève expansion, and a sparse grid approach to solve the Darcy problem is quite effective. Note however that we will not fix a-priori the number of random variables to be considered, but rather propose a version of the adaptive algorithm that progressively adds dimensions to the search space, thus formally working with $N = \infty$ random variables. Yet, even such dimension adaptive sparse grids (as well as the quasi-optimal ones) may suffer from a deterioration of the performance when the lognormal random field gets rougher. In particular, in the case $\nu = 0.5$, numerical experience seems to indicate that their performance might be asymptotically not better than a standard Monte Carlo method. Thus, in this case we will actually compare the performances of the adaptive and quasi-optimal sparse grids in the framework proposed in [29], in which they will be applied to a smoothed version of the problem (where a sparse grid approach can be effective), and the results used as control variates in a Monte Carlo approach.

The rest of the paper is organized as follows. We start by introducing the general construction of sparse grids in Section 2. Then, we discuss in detail the construction of the quasi-optimal and adaptive sparse grids in Section 3: in particular, we will setup a common framework for the two methods in the context of the resolution of discrete optimization problems, and specify the details of the two algorithms in Sections 3.1 and 3.2 respectively. The details of the Darcy problem are presented in Section 4, and in particular we will describe the dimension adaptive algorithm in Section 4.1 and the Monte Carlo Control Variate approach in Section 4.2. The numerical results are shown in Section 5, while Section 6 presents the conclusions of this work.

In what follows, $\mathbb{N}$ will denote the set of integer numbers including 0, and $\mathbb{N}_+$ that of integer numbers excluding 0. Given two vectors $\mathbf{v},\mathbf{w} \in \mathbb{N}^N$, $|\mathbf{v}|_0, |\mathbf{v}|_1, |\mathbf{v}|_2$ denote respectively the number of non-zero entries of $\mathbf{v}$, the sum of their absolute values and the euclidean norm of $\mathbf{v}$, and we write $\mathbf{v} \leq \mathbf{w}$ if and only if $v_j \leq w_j$ for every $1 \leq j \leq N$. Moreover, $\mathbf{0}$ will denote the vector $(0,0,\dots,0) \in \mathbb{N}^N$, $\mathbf{1}$ the vector $(1,1,\dots,1) \in \mathbb{N}^N$, and $\mathbf{e}_j$ the j-th canonical vector in $\mathbb{R}^N$, i.e. a vector whose components are all zero but the j-th, whose value is one. To close our introduction, we recall the definition of some functional spaces that will be useful in the following. In particular, we will need the weighted L^p spaces

$$L^p_\rho(\Gamma;V) = \left\{ f : \Gamma \to V \text{ s.t. } \int_\Gamma \|f(\mathbf{y})\|_V^p \rho(\mathbf{y}) d\mathbf{y} < \infty \right\}, \ \forall p \in (0,\infty),$$

and the space of continuous functions with weighted maximum norm

$$C^0_\pi(\Gamma;V) = \left\{ f : \Gamma \to V \text{ s.t. } f \text{ is continuous and } \max_\Gamma \|f(\mathbf{y})\|_V \pi(\mathbf{y}) < \infty \right\},$$

where $\pi = \prod_{n=1}^N \pi_n(y_n), \pi_n : \Gamma_n \to \mathbb{R}$, is a positive and smooth function. The reasons for introducing two different weight functions ρ and π will be clearer later on. Observe in particular that since V and $L^2_\rho(\Gamma)$ are Hilbert spaces, $L^2_\rho(\Gamma;V)$ is isomorphic to the tensor space $V \otimes L^2_\rho(\Gamma)$, and is itself an Hilbert space.

2 Sparse grid approximation of multivariate functions

As already mentioned in the introduction, we consider the problem of constructing a sparse grid approximation with global polynomials of the V-valued multivariate function f, defined over the hypercube Γ with associated probability measure $\rho(\mathbf{y})d\mathbf{y} = \prod_{n=1}^N \rho_n(y_n)dy_n$. More precisely, we will consider functions f that are continuous with respect to $\mathbf{y}$ and with finite variance, i.e. belonging to $L^2_\rho(\Gamma;V) \cap C^0_\pi(\Gamma;V)$ for some suitable weight π (which can be often taken equal to ρ, but not always, as indeed in certain instances of the stochastic Darcy problem we will consider in the numerical part of this paper, see e.g. [1, 19]). Observe that approximating f with global polynomials is a sound approach if f is not just continuous, but actually a smooth function of $\mathbf{y}$, see [1, 25]. Sparse grids based on piecewise polynomial approximations, which are suitable for non-smooth or even discontinuous functions, have been developed e.g. in [20, 15].

To begin with the construction of the sparse grid, we consider a sequence $\{\mathcal{U}_n^{m(i_n)}\}_{i_n \in \mathbb{N}}$ of univariate Lagrangian interpolant operators along each dimension Γ_n of the hypercube Γ,

$$\mathcal{U}_n^{m(i_n)} : C^0_{\pi_n}(\Gamma_n) \to \mathbb{P}_{m(i_n)-1}(\Gamma_n),$$

where $m(i_n)$ denotes the number of collocation points used by the i_n-th interpolant, and $\mathbb{P}_q(\Gamma_n)$ is the set of polynomials in y_n of degree at most q. The function $m : \mathbb{N} \to \mathbb{N}$ is called "level-to-nodes function" and is a strictly increasing function, with $m(0) = 0$ and $m(1) = 1$; consistently, we set $\mathcal{U}_n^0[f] = 0$. Next, for any $\mathbf{i} \in \mathbb{N}_+^N$ we define the tensor interpolant operator

$$\mathcal{T}_{\mathbf{i}}^m[f](\mathbf{y}) = \bigotimes_{n=1}^{N} \mathcal{U}_n^{m(i_n)}[f](\mathbf{y}), \tag{1}$$

and the *hierarchical surplus* operator

$$\boldsymbol{\Delta}^{m(\mathbf{i})} = \bigotimes_{n=1}^{N} \left(\mathcal{U}_n^{m(i_n)} - \mathcal{U}_n^{m(i_n - 1)} \right). \tag{2}$$

A sparse grid approximation is built as a sum of *hierarchical surplus* operators; more specifically, we consider a sequence of index sets $\mathbf{I}(w) \subset \mathbb{N}_+^N$ such that $\mathbf{I}(w) \subset \mathbf{I}(w+1)$, $\mathbf{I}(0) = \{\mathbf{1}\}$ and $\cup_{w \in \mathbb{N}} \mathbf{I}(w) = \mathbb{N}_+^N$, and we define the sparse grid approximation of $f(\mathbf{y})$ at level $w \in \mathbb{N}$ as

$$\mathcal{S}_{\mathbf{I}(w)}^m : L_\rho^2(\Gamma; V) \cap C_\pi^0(\Gamma; V) \to L_\rho^2(\Gamma; V), \quad \mathcal{S}_{\mathbf{I}(w)}^m[f](\mathbf{y}) = \sum_{\mathbf{i} \in \mathbf{I}(w)} \boldsymbol{\Delta}^{m(\mathbf{i})}[f](\mathbf{y}). \tag{3}$$

To ensure good approximation properties to the sparse approximation, the sum (3) must be telescopic, cf. [17]: to this end we require that

$$\forall \mathbf{i} \in \mathbf{I}, \quad \mathbf{i} - \mathbf{e}_j \in \mathbf{I} \text{ for } 1 \le j \le N \text{ such that } i_j > 1.$$

A set $\mathbf{I}$ satisfying the above property is said to be a *lower set* or a *downward closed set*, see e.g. [10]. The choice of the set $\mathbf{I}(w)$ plays a crucial role in devising effective sparse grid schemes: the next section will be entirely devoted to the discussion of two possible strategies to this end, namely the a-posteriori adaptive and the "a-priori/a-posteriori" quasi-optimal procedures that have been mentioned in the introduction.

Further insight into the structure of sparse grid operators can be obtained by rewriting (3) as a linear combination of tensor interpolant operators (1), see e.g. [30]. Assuming that $\mathbf{I}(w)$ is downward closed, we get indeed

$$\mathcal{S}_{\mathbf{I}(w)}^m[f](\mathbf{y}) = \sum_{\mathbf{i} \in \mathbf{I}(w)} c_{\mathbf{i}} \mathcal{T}_{\mathbf{i}}^m[f](\mathbf{y}), \qquad c_{\mathbf{i}} = \sum_{\substack{\mathbf{j} \in \{0,1\}^N \\ (\mathbf{i}+\mathbf{j}) \in \mathbf{I}(w)}} (-1)^{|\mathbf{j}|}. \tag{4}$$

Observe that many of the coefficients $c_{\mathbf{i}}$ in (4) may be zero: in particular $c_{\mathbf{i}}$ is zero whenever $\mathbf{i} + \mathbf{j} \in \mathbf{I}(w)$ $\forall \mathbf{j} \in \{0,1\}^N$. The set of all collocation points needed by (4) is actually called a *sparse grid*, and we denote its cardinality by $W_{\mathbf{I}(w),m}$. It is useful to introduce the operator $\mathrm{pts}(\mathcal{S})$ that returns the set of points associated to a tensor / sparse grid operator, and the operator $\mathrm{card}(\mathcal{S})$ that returns the cardinality of $\mathrm{pts}(\mathcal{S})$:

Table 1 Common choices of univariate collocation points for sparse grids.

Collocation points			
	measure	**nested**	$\mathbf{m}(\mathbf{i})$
Gauss–Legendre	uniform	no	i
Clenshaw–Curtis	uniform	yes	$2^{i-1}+1$
Gauss–Patterson	uniform	yes	2^i-1
Leja	uniform	yes	$m(i)=i$ or $m(i)=2i-1$
Gauss–Hermite	gaussian	no	i
Genz–Keister	gaussian	yes	tabulated: $m(i)=1,3,9,19,35$
generalized Leja	gaussian	yes	i

$$\mathrm{card}(\mathcal{T}_{\mathbf{i}}^m)=\prod_{n=1}^{N} m(i_n), \qquad \mathrm{card}(\mathcal{S}^m_{\mathbf{I}(w)})=W_{\mathbf{I}(w),m}\,. \tag{5}$$

Finally, consider a sequence of univariate quadrature operators built over the same set of points of $\{\mathcal{U}_n^{m(i_n)}\}_{i_n\in\mathbb{N}}$; it is then relatively straightforward to derive a sparse grid quadrature scheme $\mathcal{Q}^m_{\mathbf{I}(w)}[\cdot]$ starting from (4):

$$\int_\Gamma f(\mathbf{y})\rho(\mathbf{y})d\mathbf{y} \approx \int_\Gamma \mathcal{S}^m_{\mathbf{I}(w)}[f]\rho(\mathbf{y})d\mathbf{y} = \sum_{j=1}^{W^m_{\mathbf{I}(w)}} f(\mathbf{y}_j)\beta_j = \mathcal{Q}^m_{\mathbf{I}(w)}[f], \tag{6}$$

for suitable quadrature weights $\beta_j \in \mathbb{R}$.

Coming to the choice of the univariate collocation points used to build $\mathcal{U}_n^{m(i_n)}$, as mentioned in the introduction they should be chosen according to the probability measure $\rho_n(y_n)dy_n$ on Γ_n. Although the use of nested points seems to be particularly indicated for the hierarchical construction (3), as the $\boldsymbol{\Delta}^{m(\mathbf{i})}$ operator would entail evaluations only on the new points added going from the tensor grid $\mathcal{T}^m_{\mathbf{i-1}}$ to $\mathcal{T}^m_{\mathbf{i}}$, at this point any choice of univariate collocation points is allowed (see Table 1), and in particular Gauss interpolation / quadrature points, associated to the underlying probability density functions $\rho_n(y_n)$, have been widely used, cf. e.g. [2, 26, 13, 12]. Note however that non-nested interpolatory rules have not been used in the adaptive context, for reasons that will be clearer in a moment; the aim of this work is to extend the adaptive algorithm to non-nested quadrature rules.

3 On the choice of $\mathbf{I}(w)$

In this section we detail two possible strategies to design the sequence of sets $\mathbf{I}(w)$. To simplify the notation, let us assume that $V=\mathbb{R}$, i.e. f is a real-valued N-variate function over Γ, and that we are measuring the sparse grid approximation error by some sublinear functional[1] $\mathcal{E}[\cdot]$, e.g. a semi-norm on $L^p_\rho(\Gamma)$ (we will give three such

[1] A sublinear functional over a vector space X is a function $\Theta : X \to \mathbb{R}$ such that

examples in the following). Furthermore, assume that the sparse grid approximation converges ρ-a.e. to f, so that we can write $f = \mathcal{S}^m_{\mathbb{N}^N_+}[f] = \sum_{\mathbf{i}\in\mathbb{N}^N_+} \boldsymbol{\Delta}^{m(\mathbf{i})}[f]$. Then, we have

$$\mathcal{E}\left[f - \mathcal{S}^m_{\mathbf{I}(w)}[f]\right] = \mathcal{E}\left[\sum_{\mathbf{i}\notin\mathbf{I}(w)} \boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right] \leq \sum_{\mathbf{i}\notin\mathbf{I}(w)} \mathcal{E}\left[\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right]. \tag{7}$$

Since the exact value of $\mathcal{E}\left[\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right]$ may not be at disposal, we further define the *error contribution operator* $\Delta E(\mathbf{i})$ as any computable (and hopefully tight) approximation of $\mathcal{E}\left[\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right]$, namely $\Delta E(\mathbf{i}) \approx \mathcal{E}\left[\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right]$. Moreover, we also introduce the *work contribution* $\Delta W(\mathbf{i})$, i.e. the number of evaluations of f implied by the addition of the hierarchical surplus operator $\boldsymbol{\Delta}^{m(\mathbf{i})}[f]$ to the sparse grid approximation. Observe that this is actually a quite delicate issue when using non-nested points as discussed later on.

Upon having assigned an error and a work contribution to each hierarchical surplus operator, the selection of the sequence of sets $\mathbf{I}(w)$ can be rewritten as a "binary knapsack problem" [6, 22],

$$\max \sum_{\mathbf{i}\in\mathbb{N}^N_+} \Delta E(\mathbf{i}) x_{\mathbf{i}} \quad \text{s.t.} \sum_{\mathbf{i}\in\mathbb{N}^N_+} \Delta W(\mathbf{i}) x_{\mathbf{i}} \leq W_{max}(w) \text{ and } x_{\mathbf{i}} \in \{0,1\},$$

where $W_{\max}(w)$ is the maximum computational work allowed for the approximation level w. Note that we are not explicitly enforcing that the resulting sets $\mathbf{I}(w)$ be downward closed (which will have to be verified a-posteriori).

While the binary knapsack problem is known to be computationally intractable (NP-hard) its *linear programming relaxation*, in which fractional values of $x_{\mathbf{i}}$ are allowed, can be solved analytically by the so-called Dantzig algorithm [22]:

1. Assign a "profit" to each multi-index $\mathbf{i}$,
$$P(\mathbf{i}) = \frac{\Delta E(\mathbf{i})}{\Delta W(\mathbf{i})}; \tag{8}$$
2. sort multi-indices by decreasing profit;
3. set $x_{\mathbf{i}} = 1$, i.e. add $\mathbf{i}$ to $\mathbf{I}(w)$, until the constraint on the maximum work is fulfilled. In particular, whenever the multi-index $\mathbf{1} + \mathbf{e}_n$ enters the set $\mathbf{I}(w)$ we say that the random variable y_n is *activated*.

Note that only the last multi-index included in the selection is possibly taken not entirely (i.e. with $x_{\mathbf{i}} < 1$), whereas all the previous ones are taken entirely (i.e. with $x_{\mathbf{i}} = 1$). However, if this is the case, we assume that we could slightly adjust the computational budget, so that all $x_{\mathbf{i}}$ have integer values; observe that such integer

- $\Theta(\alpha x) = \alpha\Theta(x)$, $\forall \alpha > 0$ and $x \in X$;
- $\Theta(x+y) \leq \Theta(x) + \Theta(y)$, $\forall x, y \in X$.

solution is also the solution of the original binary knapsack problem with modified work constraint.

Both the quasi-optimal and the a-posteriori adaptive sparse grids strategies fit in this general framework. What changes between the two schemes are just the choice of the error indicator $\mathcal{E}[\cdot]$ and the way $\Delta W(\mathbf{i})$ and $\Delta E(\mathbf{i})$ are computed.

3.1 Quasi-optimal sparse grids

In this section we briefly summarizing the quasi-optimal sparse grids construction, see [25] for a thorough discussion. In this case, the error indicator $\mathcal{E}[\cdot]$ is the L^2_ρ-norm, so that (7) becomes

$$\left\| f - \mathcal{S}^m_{\mathbf{I}(w)}[f] \right\|_{L^2_\rho} \leq \sum_{\mathbf{i} \notin \mathbf{I}(w)} \left\| \boldsymbol{\Delta}^{m(\mathbf{i})}[f] \right\|_{L^2_\rho},$$

and we need to provide a computable approximation $\left\| \boldsymbol{\Delta}^{m(\mathbf{i})}[f] \right\|_{L^2_\rho} \approx \Delta E(\mathbf{i})$. Following [25, 5, 4], this can be obtained by further introducing the spectral expansion of f over a N-variate $\widetilde{\rho}$-orthonormal polynomial basis $\varphi_{\mathbf{q}}(\mathbf{y})$ [2], with $\widetilde{\rho}$ not necessarily equal to ρ; for example, in the case where y_n are uniform random variables, $\rho_n(y_n) = 1/|\Gamma_n|$, one is allowed to expand f on tensorized Chebyshev polynomials, which are orthonormal with respect to $\widetilde{\rho} = \prod_{n=1}^N \widetilde{\rho}_n$, with $\widetilde{\rho}_n(y_n) = 1/\sqrt{1-y_n^2}$. Next, let us denote by $f_{\mathbf{q}}$ the $\mathbf{q}$-th coefficient of the $\widetilde{\rho}$-expansion of f and by $\mathbb{M}_n^{m(i_n)}$ the "$C^0_\pi \to L^2_\rho$ Lebesgue constant" of the univariate interpolant operators $\mathcal{U}_n^{m(i_n)}$ for a suitable weight π, i.e.

$$\mathbb{M}_n^{m(i_n)} = \sup_{\|f\|_{C^0_\pi(\Gamma_n)}=1} \left\| \mathcal{U}_n^{m(i_n)}[f] \right\|_{L^2_\rho(\Gamma_n)}.$$

Then, assuming that the coefficients $f_{\mathbf{q}}$ are at least exponentially decreasing in each y_n, $|f_{\mathbf{q}}| \leq C \prod_n \exp(-g_n q_n)$, and that $\|\varphi_{\mathbf{q}}\|_{C^0_\pi} \leq C^{|\mathbf{q}|_0}$, following [25] we have that for a suitable constant C there holds

$$\left\| \boldsymbol{\Delta}^{m(\mathbf{i})}[f] \right\|_{L^2_\rho} \leq \Delta E(\mathbf{i}) = C(N) \left| f_{m(\mathbf{i}-\mathbf{1})} \right| \prod_{n=1}^N \mathbb{M}_n^{m(i_n)}. \tag{9}$$

Observe that in practical cases, the constant $\mathbb{M}_n^{m(i_n)}$ can be estimated numerically, and computable ansatzes for $f_{m(\mathbf{i}-\mathbf{1})}$ can be derived, so that it is possible to obtain numerical estimates of the quantities $\Delta E(\mathbf{i})$. Such computable ansatzes depend on the exponential coefficients $g_1, \ldots, g_N$, that can be conveniently precomputed with a numerical procedure that requires $\mathcal{O}(N)$ evaluations of f. We will return on this

[2] Here the n-th component of $\mathbf{q}$ denotes the polynomial degree with respect to y_n.

matter in the next sections, proposing an ansatz for the Darcy problem, as well as giving details on the numerical procedure needed to estimate $g_1,\dots,g_N$.

Concerning the work contributions $\Delta W(\mathbf{i})$, the definitions are different depending on whether the family of nodes considered is nested or non-nested (see [25] for details). In the former case, we can set

$$\Delta W(\mathbf{i}) = \prod_{n=1}^{N} \big(m(i_n) - m(i_n - 1)\big), \tag{10}$$

and there holds

$$W_{\mathbf{I}(w),m} = \sum_{\mathbf{i}\in\mathbf{I}(w)} \Delta W(\mathbf{i}),$$

i.e. the cardinality of the sparse grid is equal to the sum of the work contributions. On the contrary, when considering non-nested points the number of new evaluations of f needed by the addition of $\boldsymbol{\Delta}^{m(\mathbf{i})}$ will depend in general on the set $\mathbf{I}$ to which $\mathbf{i}$ is added to, i.e. if $\mathbf{I},\mathbf{I}'$ are two index sets such that both $\mathbf{I}\cup\{\mathbf{j}\}$ and $\mathbf{I}'\cup\{\mathbf{j}\}$ are downward closed, it can happen that

$$\text{card}(\mathcal{S}^m_{\mathbf{I}\cup\{\mathbf{j}\}}) \neq \text{card}(\mathcal{S}^m_{\mathbf{I}'\cup\{\mathbf{j}\}}), \tag{11}$$

and nodes that are present in the sparse grid built over $\mathbf{I}$ are not necessarily present in the one built over $\mathbf{I}\cup\{\mathbf{j}\}$, i.e.

$$\text{pts}(\mathcal{S}^m_{\mathbf{I}}) \not\subset \text{pts}(\mathcal{S}^m_{\mathbf{I}\cup\{\mathbf{j}\}}). \tag{12}$$

Therefore, we have to use the pessimistic estimate

$$\Delta W(\mathbf{i}) = \prod_{n=1}^{N} m(i_n) = \text{card}(\mathcal{T}^m_{\mathbf{i}}[f]), \tag{13}$$

i.e. the cardinality of the entire tensor grid associated to $\mathbf{i}$, which ensures

$$W_{\mathbf{I}(w),m} \leq \sum_{\mathbf{i}\in\mathbf{I}(w)} \Delta W(\mathbf{i}).$$

Once the numerical values of $\Delta E(\mathbf{i})$ and $\Delta W(\mathbf{i})$ are available, the profits (8) and the sequence of optimal sets $\mathbf{I}(w)$ can be computed right-away, and the sparse grid construction can proceed. Thus, this algorithm is said to be “a-priori”/“a-posteriori” since it relies on a-priori estimates whose constants $g_1 \dots, g_N$ need however to be tuned numerically.

3.2 An extended adaptive sparse grid algorithm

We now describe the adaptive sparse grid construction algorithm [17, 8, 27], and its extension to non-nested points and unbounded intervals. To this end, we introduce the concepts of *margin* and *reduced margin* of a multi-index set $\mathbf{I}$, and the concept of *neighbors* of a multi-index. The margin of $\mathbf{I}$, which we denote by $\mathbf{M}_{\mathbf{I}}$, contains all the multi-indices $\mathbf{i}$ that can be reached within "one-step forward" from $\mathbf{I}$, i.e.

$$\mathbf{M}_{\mathbf{I}} = \{\mathbf{i} \in \mathbb{N}_+^N : \exists \mathbf{j} \in \mathbf{I} : |\mathbf{i}-\mathbf{j}|_1 = 1\}.$$

The reduced margin of $\mathbf{I}$, denoted by $\mathbf{R}$, is the subset of the margin of $\mathbf{I}$ containing only those indices $\mathbf{i}$ such that "one-step backward" in any direction takes into $\mathbf{I}$, i.e.

$$\mathbf{R} = \{\mathbf{i} \in \mathbb{N}_+^N : \mathbf{i} - \mathbf{e}_j \in \mathbf{I},\ \forall\, j = 1,\dots,N : i_j > 1\} \subset \mathbf{M}_{\mathbf{I}}.$$

This means that the reduced margin of $\mathbf{I}$ contains all indices $\mathbf{i}$ such that $\mathbf{I} \cup \{\mathbf{i}\}$ is downward closed, provided that $\mathbf{I}$ itself is downward closed. Furthermore, given an index $\mathbf{i}$ on the boundary of $\mathbf{I}$, we call *neighbors* of $\mathbf{i}$ with respect to $\mathbf{I}$, $\mathrm{neigh}(\mathbf{i},\mathbf{I})$, the indices $\mathbf{j}$ not included in $\mathbf{I}$ that can be reached with "one step forward" from $\mathbf{i}$, so that $\mathbf{M}_{\mathbf{I}} = \bigcup_{\mathbf{i}\in\mathbf{I}} \mathrm{neigh}(\mathbf{i},\mathbf{I})$.

Instead of computing the profits and the sets $\mathbf{I}(w)$ beforehand as in the quasi-optimal algorithm, the idea of the adaptive algorithm is to compute the profits and the sets $\mathbf{I}(w)$ at run-time, proceeding iteratively in a greedy way. More specifically, given a multi-index set $\mathbf{I}$ and its reduced margin $\mathbf{R}$, the adaptive algorithm operates as follows:

1. the profits of $\mathbf{i} \in \mathbf{R}$ are computed;
2. the index $\mathbf{i}$ with the highest profit is moved from $\mathbf{R}$ to $\mathbf{I}$;
3. the reduced margin is updated and the algorithm moves to the next iteration, until some stopping criterion is met (usually, a check on the number of evaluations of f or on the values of the profits or error contributions of the multi-indices in $\mathbf{R}$).

Note that the profits of the indices $\mathbf{i} \in \mathbf{R}$ are computed by actually adding the hierarchical surpluses to the sparse grid operator (as will be clearer in a moment), hence the definition of "a-posteriori"; therefore, the outcome of the algorithm at each iteration is the sparse gird approximation built on $\mathbf{I} \cup \mathbf{R}$ and not on $\mathbf{I}$ only.

In this work, we consider two different error indicators $\mathcal{E}[\cdot]$, namely the absolute value of the expectation of $f - \mathcal{S}^m_{\mathbf{I}(w)}[f]$, which is a semi-norm on $L^1_\rho(\Gamma)$, and the weighted $C^0_\pi(\Gamma)$ norm, so that (7) becomes

$$\left|\mathbb{E}[f] - \mathbb{E}[\mathcal{S}^m_{\mathbf{I}}[f]]\right| \leq \sum_{\mathbf{i}\notin\mathbf{I}} \left|\mathbb{E}\left[\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right]\right|,$$

$$\|f - \mathcal{S}^m_{\mathbf{I}}[f]\|_{C^0_\pi(\Gamma)} \leq \sum_{\mathbf{i}\notin\mathbf{I}} \left\|\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right\|_{C^0_\pi(\Gamma)}.$$

To derive the error indicator $\Delta E(\mathbf{i})$ for the quantity $\mathcal{E}\left[\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right]$, let us consider an arbitrary set $\mathbf{I}$, and let $\mathbf{J} = \mathbf{I} \cup \{\mathbf{i}\}$, with both $\mathbf{I}, \mathbf{J}$ downward closed index sets. For the $L^1_\rho(\Gamma)$ seminorm, we immediately have

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right] &= \mathbb{E}\left[\ \mathcal{S}^m_{\mathbf{J}}[f] - \mathcal{S}^m_{\mathbf{I}}[f]\right] = \mathcal{Q}^m_{\mathbf{J}}[f] - \mathcal{Q}^m_{\mathbf{I}}[f],\\ &\Rightarrow\ \Delta E(\mathbf{i}) = |\mathcal{Q}^m_{\mathbf{J}}[f] - \mathcal{Q}^m_{\mathbf{I}}[f]|. \qquad (14)\end{aligned}$$

In the case of the $C^0_\pi(\Gamma)$ norm, the computation is different for nested and non-nested points. When using nested points we can set $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{S}^m_{\mathbf{J}}) \setminus \mathrm{pts}(\mathcal{S}^m_{\mathbf{I}})$, cf. eq. (5), and, since the sparse grid operator is interpolatory (see e.g [3, Prop. 6]), we have $f(\mathbf{y}) = \mathcal{S}^m_{\mathbf{J}}[f](\mathbf{y})$ for $\mathbf{y} \in \mathcal{N}\mathrm{ew}$. Thus

$$\begin{aligned}\left\|\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right\|_{C^0_\pi(\Gamma)} &= \left\|\ \mathcal{S}^m_{\mathbf{J}}[f] - \mathcal{S}^m_{\mathbf{I}}[f]\ \right\|_{C^0_\pi(\Gamma)}\\ &\approx \max_{\mathbf{y}\in\mathcal{N}\mathrm{ew}} \left|\Big(\mathcal{S}^m_{\mathbf{J}}[f](\mathbf{y}) - \mathcal{S}^m_{\mathbf{I}}[f](\mathbf{y})\Big)\pi(\mathbf{y})\right|\\ &= \max_{\mathbf{y}\in\mathcal{N}\mathrm{ew}} \left|\Big(f(\mathbf{y}) - \mathcal{S}^m_{\mathbf{I}}[f](\mathbf{y})\Big)\pi(\mathbf{y})\right| = \Delta E(\mathbf{i}). \qquad (15)\end{aligned}$$

On the other hand, sparse grids built with non-nested points are not interpolatory, and the set of points added to a sparse grid by $\boldsymbol{\Delta}^{m(\mathbf{i})}$ is not unique, cf. eq. (11), as it depends on the current index set $\mathbf{I}$ to which $\mathbf{i}$ is added. Thus, we define $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{T}_{m(\mathbf{i})})$ and approximate

$$\begin{aligned}\left\|\boldsymbol{\Delta}^{m(\mathbf{i})}[f]\right\|_{C^0_\pi(\Gamma)} &= \left\|\ \mathcal{S}^m_{\mathbf{J}}[f] - \mathcal{S}^m_{\mathbf{I}}[f]\ \right\|_{C^0_\pi(\Gamma)}\\ &\approx \max_{\mathbf{y}\in\mathcal{N}\mathrm{ew}} \left|\Big(\mathcal{S}^m_{\mathbf{J}}[f](\mathbf{y}) - \mathcal{S}^m_{\mathbf{I}}[f](\mathbf{y})\Big)\pi(\mathbf{y})\right| = \Delta E(\mathbf{i}). \qquad (16)\end{aligned}$$

We remark that the values of $\Delta E(\mathbf{i})$ defined in equations (14)-(16) do not depend on the set $\mathbf{I}$. This means that we can consider the indices of the reduced margin $\mathbf{R}$ in any order, and that the values of $\Delta E(\mathbf{i})$ need not be recomputed at each iteration.

As for the work contribution $\Delta W(\mathbf{i})$, we consider the same indicators defined in the quasi-optimal case, i.e. (10) for nested points and (13) for non-nested points, which is equivalent to setting the work contributions equal to the cardinality of the sets $\mathcal{N}\mathrm{ew}$ introduced above. A third option is to consider $\Delta W(\mathbf{i}) = 1$, i.e. driving the adaptivity only by the error contributions. This is the choice considered e.g. in [8, 27], while [17, 21] combine $\Delta E(\mathbf{i})$ and $\Delta W(\mathbf{i})$ in a different way. To summarize, we will drive the adaptive algorithm with any of the four profit definitions listed next, whose formulas differ depending on whether nested or non-nested points are used:

- **"deltaint"**: set $\Delta E(\mathbf{i})$ as in (14) and $\Delta W(\mathbf{i}) = 1$;
- **"deltaint / new points"** combine $\Delta E(\mathbf{i})$ as in (14) with $\Delta W(\mathbf{i})$ in (10) for nested points and in (13) for non-nested points;

- **“weighted Linf”** set $\Delta E(\mathbf{i})$ as in (15) and $\Delta W(\mathbf{i}) = 1$ for nested points, and $\Delta E(\mathbf{i})$ as in (16) and $\Delta W(\mathbf{i}) = 1$ for non-nested points;
- **“weighted Linf / new points”** combine $\Delta E(\mathbf{i})$ in (15) with $\Delta W(\mathbf{i})$ in (10) for nested points and $\Delta E(\mathbf{i})$ in (16) with $\Delta W(\mathbf{i})$ in (13) for non-nested points.

The pseudo-code of the algorithm is listed in Algorithm 1. Since nodes that are present in a given sparse grid are not necessarily present in the following ones when using non-nested points, cf. eq. (12), the full work count in this case is not simply $\mathrm{pts}(\mathcal{S})$ (as it would be for nested points), but should rather include all the points “visited” to reach that grid in the adaptive algorithm, which motivates lines L1-L2 in Algorithm 1. Observe however that all Gaussian quadrature rules associated to a symmetric weight (or probability density) are in a sense “partially nested”, meaning that rules with odd number of points place a quadrature node in the midpoint of the interval, implying that a non-negligible number of points can still be in common between two grids (e.g., the grid with 3 × 5 Gauss–Legendre points shares 5 of its 15 points with the grid 1 × 5).

4 Darcy Problem

As mentioned in the introduction, in this work we are concerned with the application of the adaptive sparse grid algorithm in the Uncertainty Quantification context. In particular, we focus on the numerical approximation of the solution of the stochastic version of the Darcy problem [5, 14, 7, 18] in which an unknown Darcy pressure p is obtained as solution of an elliptic PDE having a lognormal random field a as diffusion coefficient; a models the permeability of the medium in which the flow takes place and, since it is a quantity that often can not be properly estimated, it is modeled as a random field over a suitable probability space $(\Omega, \mathcal{F}_\Omega, \mathbb{P})$, where Ω is the set of possible outcomes ω, $\mathcal{F}_\Omega$ a σ-algebra and $\mathbb{P} : \mathcal{F}_\Omega \to [0,1]$ a probability measure. The mathematical formulation of the problem is the following:

Problem 1 *Given $D \in \mathbb{R}^d$, find a real-valued function $p : \overline{D} \times \Omega \to \mathbb{R}$, such that $\mathbb{P}$-almost surely (a.s) there holds:*

$$\begin{cases} -\operatorname{div}(a(\mathbf{x},\omega)\nabla p(\mathbf{x},\omega)) = f(\mathbf{x}) & \mathbf{x} \in D, \\ p(\mathbf{x},\omega) = g(\mathbf{x}) & \mathbf{x} \in \partial D_j^D,\ j = 1,\dots,k_D, \\ \nabla p(\mathbf{x},\omega) \cdot \mathbf{n} = 0 & \mathbf{x} \in \partial D_j^N,\ j = 1,\dots,k_N, \end{cases}$$

where the operators div *and* ∇ *imply differentiation with respect to the physical coordinates only, $a : \overline{D} \times \Omega \to \mathbb{R}$ is a given random field, $\mathbf{n}$ is the outward normal to the boundary, $\partial D_D = \cup_{j=1}^{k_D} \partial D_j^D$ denotes the Dirichlet boundary, $\partial D_N = \cup_{j=1}^{k_N} \partial D_j^N$ denotes the Neumann boundary and $\overline{\partial D_D} \cup \overline{\partial D_N} = \partial D$, $\overset{\circ}{\partial D}_D \cap \overset{\circ}{\partial D}_N = \emptyset$.*

More specifically, we set $a(\mathbf{x},\omega) = e^{\gamma(\mathbf{x},\omega)}$, being γ a mean-free stationary Gaussian random field having a tensor covariance function belonging to the so-called Matérn

Algorithm 1: Adaptive sparse grids algorithm.

```
Adaptive sparse grids(MaxPts, ProfTol, $\pi$, <ProfitName>)
    $\mathbf{I} = \{\mathbf{1}\}$, $\mathbf{G} = \{\mathbf{1}\}$, $\mathbf{R} = \varnothing$, $\mathbf{i} = \mathbf{1}$ ;
    $\mathcal{S}_{old} = \mathcal{S}^m_{\mathbf{I}}[u]$, $\mathcal{Q}_{old} = \mathcal{Q}^m_{\mathbf{I}}[f]$ ;
    $\mathcal{H} = \mathrm{pts}(\mathcal{S}_{old})$, NbPts=card($\mathcal{S}_{old}$), ProfStop=$\infty$ ;
    while NbPts < MaxPts and ProfStop > ProfTol do
        $\mathcal{N}\mathrm{g} = \mathrm{neigh}(\mathbf{i}, \mathbf{I})$
        for $\mathbf{j} \in \mathcal{N}\mathrm{g}$ and $\mathbf{I} \cup \{\mathbf{j}\}$ downward closed do
            $\mathbf{G} = \mathbf{G} \cup \{\mathbf{j}\}$ ;                 at the end of the for loop, $\mathbf{G} = \mathbf{I} \cup \mathbf{R}$
            $\mathcal{S} = \mathcal{S}^m_{\mathbf{G}}[f]$ ;                 $\mathbf{j}$ must be added to $\mathcal{S}$ to evaluate its profit.
            $\mathcal{Q} = \mathcal{Q}^m_{\mathbf{G}}[f]$ ;
            if using nested points then
                $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{S}) \setminus \mathrm{pts}(\mathcal{S}_{old})$ ;        i.e. the points added by $\mathbf{j}$ to $\mathcal{S}$
                NbPts = NbPts + card($\mathcal{N}\mathrm{ew}$) ;
                $\mathbf{v}$ = evaluations of $f$ on each $\mathbf{y} \in \mathcal{N}\mathrm{ew}$ ;        cf. eq. (15)
            else
                $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{T}_{m(\mathbf{i})})$
L1              $\mathcal{H} = \mathcal{H} \cup \mathrm{pts}(\mathcal{S})$ ;        add points of $\mathcal{S}$ to $\mathcal{H}$ (no repetitions)
L2              NbPts = card($\mathcal{H}$) ;        for non-nested points, card($\mathcal{H}$)>card($\mathcal{S}$)
                $\mathbf{v}$ = evaluations of $\mathcal{S}$ on each $\mathbf{y} \in \mathcal{N}\mathrm{ew}$ ;        cf. eq. (16)
            $\mathbf{v}_{old}$ = evaluations of $\mathcal{S}_{old}$ on each $\mathbf{y} \in \mathcal{N}\mathrm{ew}$ ;
            $\boldsymbol{\pi}$ = evaluations of $\pi$ on each $\mathbf{y} \in \mathcal{N}\mathrm{ew}$ ;
            $P(\mathbf{j})$ = Compute_profit($\mathcal{N}\mathrm{ew}, \mathbf{v}, \mathbf{v}_{old}, \boldsymbol{\pi}, \mathcal{Q}, \mathcal{Q}_{old}$,<ProfitName>)
            $\mathbf{R} = \mathbf{R} \cup \{\mathbf{j}\}$
            $\mathcal{S}_{old} = \mathcal{S}$, $\mathcal{Q}_{old} = \mathcal{Q}$ ;
        choose the $\mathbf{i}$ from $\mathbf{R}$ with highest profit;
        $\mathbf{I} = \mathbf{I} \cup \{\mathbf{i}\}$, $\mathbf{R} = \mathbf{R} \setminus \{\mathbf{i}\}$
        update ProfStop with a suitable criterion based on the values of $P$
    return $\mathcal{S}, \mathcal{Q}$

Compute_profit($\mathcal{N}\mathrm{ew}, \mathbf{v}, \mathbf{v}_{old}, \boldsymbol{\pi}, \mathcal{Q}, \mathcal{Q}_{old}$,<ProfitName>)
    switch ProfitName do
        case deltaint
            profit($\mathbf{i}$) $= |\,\mathcal{Q} - \mathcal{Q}_{old}\,|$ ;
        case deltaint/new points
            profit($\mathbf{i}$) $= \dfrac{|\,\mathcal{Q} - \mathcal{Q}_{old}\,|}{\mathrm{card}(\mathcal{N}\mathrm{ew})}$ ;
        case Weighted Linf
            profit($\mathbf{i}$) $= \max\{|\mathbf{v} - \mathbf{v}_{old}| \odot \boldsymbol{\pi}\}$ ;        $\odot$ denotes element-wise multiplication
        case Weighted Linf/new points
            profit($\mathbf{i}$) $= \dfrac{\max\{|\mathbf{v} - \mathbf{v}_{old}| \odot \boldsymbol{\pi}\}}{\mathrm{card}(\mathcal{N}\mathrm{ew})}$ ;
    return profit($\mathbf{i}$)
```

family [11], namely:

$$\mathrm{cov}_\nu(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{i=1}^{d} \frac{\left(\sqrt{2\nu}\frac{|x_i - x_i'|}{L_c}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\frac{|x_i - x_i'|}{L_c}\right)}{\Gamma(\nu)2^{\nu-1}}, \quad \nu \geq 0.5, \tag{17}$$

where σ^2 is the pointwise variance, L_c is a correlation length, Γ is the gamma function, K_ν is the modified Bessel function of the second kind and ν is a parameter that governs the regularity of the covariance function and, in turn, of the realizations of the random field. In particular, for ν=1/2 we obtain a tensor Exponential covariance function $\mathrm{cov}_\nu(\mathbf{x},\mathbf{x}') = \sigma^2 exp\{-|\mathbf{x}-\mathbf{x}'|_1/L_c\}$ which is only Lipschitz continuous and produces realizations of the random field $a(\mathbf{x},\omega_i)$, $\omega_i \in \Omega$, that are a.s. Hölder continuous $C^{0,s}(\overline{D})$ with parameter $s < 1/2$ [3]; on the other hand when $\nu \to \infty$ we obtain a Gaussian covariance function $\mathrm{cov}_\nu(\mathbf{x},\mathbf{x}') = \sigma^2 exp\{-|\mathbf{x}-\mathbf{x}'|_2^2/L_c^2\}$ which is analytic and generates infinitely differentiable realizations; in between, depending on ν, we have all the possible regularities; in general realizations with $\nu = n+\alpha$ with $n \in \mathbb{N}$ and $\alpha \in (0,1]$, are n times a.s. differentiable and have all the n-th derivatives a.s. Hölder continuous $C^{0,s}(\overline{D})$ with parameter $s < \alpha$ (see e.g. [29, Lemma C.2]). The well-posedness of Problem 1 has been studied e.g. in [7, 18]. The choice of the diffusion coefficient a just detailed guarantees that Problem 1 has a unique solution $p \in L^2_{\mathbb{P}}(\Omega,V)$, $V = H^1(D)$, see e.g. [7, 18] for details, under standard regularity assumptions on f,g, that will be fulfilled by the test case that we will detail later on.

To make Problem 1 suitable for the sparse grid methodology developed in the previous sections, we consider a truncated Karhunen-Loève (KL) expansion of the Gaussian random field $\gamma(\mathbf{x},\omega)$ with N i.i.d. standard normal random variables $\{y_i\}_{i=1}^N$ and approximate $a(\mathbf{x},\omega) = e^{\gamma(\mathbf{x},\omega)}$ accordingly, namely

$$\begin{gathered}\gamma(\mathbf{x},\omega) = \sum_{n=1}^{\infty} \sqrt{\lambda_n}\psi_n(\mathbf{x})y_n(\omega) \approx \sum_{n=1}^{N} \sqrt{\lambda_n}\psi_n(\mathbf{x})y_n(\omega) = \gamma(\mathbf{x},\mathbf{y}(\omega)),\\ a(\mathbf{x},\omega) = e^{\gamma(\mathbf{x},\omega)} \approx e^{\gamma(\mathbf{x},\mathbf{y}(\omega))} = a(\mathbf{x},\mathbf{y}(\omega));\end{gathered} \tag{18}$$

where the functions $\psi_n(\mathbf{x}) : D \to \mathbb{R}$, $n = 1,2,3...$, and the positive coefficients $\{\lambda_n\}_{n=1}^\infty$ are the solutions of the eigenvalue problem

$$\int_D \mathrm{cov}_\nu(\mathbf{x},\mathbf{x}')\psi(\mathbf{x})d\mathbf{x} = \lambda\,\psi(\mathbf{x}').$$

Once the random field has been (approximately) parametrized with a random vector $\mathbf{y} = (y_1,...,y_N)$ belonging to the probability space $(\Gamma,\mathcal{F}_\Gamma,\rho(\mathbf{y})d\mathbf{y})$, where $\Gamma = \mathbb{R}^N$ is the image of $\mathbf{y}$, $\mathcal{F}_\Gamma$ is the Borel σ-algebra and $\rho(\mathbf{y}) = (2\pi)^{-\frac{n}{2}} \exp\left(-\frac{|\mathbf{y}|_2^2}{2}\right)$ is the probability density function of $\mathbf{y}$, we can approximate Problem 1 with the following finite dimensional parametric problem:

Problem 2 *Find a real-valued function* $u : \overline{D}\times\Gamma \to \mathbb{R}$*, such that* $\rho(\mathbf{y})d\mathbf{y}$*-almost everywhere there holds:*

[3] A function $f : \overline{D} \subset \mathbb{R}^d \to \mathbb{R}$ is said to be Hölder continuous with parameter $s \in (0,1]$, $f \in C^{0,s}(\overline{D})$, if there exist non-negative real constants C and s such that

$$|f(\mathbf{x}) - f(\mathbf{y})| \le C|\mathbf{x}-\mathbf{y}|_2^s \quad \forall \mathbf{x},\mathbf{y} \in \overline{D}.$$

$$\begin{cases} -\operatorname{div}(a(\mathbf{x},\mathbf{y})\nabla p(\mathbf{x},\mathbf{y})) = f(\mathbf{x}) & \mathbf{x}\in D, \\ p(\mathbf{x},\mathbf{y}) = g(\mathbf{x}) & \mathbf{x}\in \partial D_j^D,\ j=1,...,k_D, \\ \nabla p(\mathbf{x},\mathbf{y})\cdot \mathbf{n} = 0 & \mathbf{x}\in \partial D_j^N,\ j=1,...,k_N. \end{cases}$$

Consistently with what we said about the infinite dimensional case, Problem 2 admits a unique solution $p\in L^2_\rho(\Gamma,V)$, and it is now ready to be solved numerically. In particular, in our analysis we will be interested in computing the expectation of some quantity of interest (QoI) related to the solution p of the Darcy problem, defined as $u(\omega)=L(p(\cdot,\omega))$, where L is a functional $L:V\to\mathbb{R}$ that we will detail later on. At this point it is also crucial to remark that solving the stochastic Darcy problem with sparse grids is a sound approach since it can be shown that the dependence of p on the random parameters $\mathbf{y}$ is smooth, and more precisely analytic, as shown in [1, 14]; moreover it can be shown that $p\in C^0_\pi(\Gamma,V)$ with $\pi(\mathbf{y})=\prod_{n=1}^N exp(-|y_n|\sqrt{\lambda_n}\,\|\psi_n\|_{L^\infty(D)})$, see [19]. Nonetheless, we will choose $\rho=\widetilde{\rho}=\pi$ in the computations, cf. equations (9), (15), (16). In particular, this means that we can use Hermite polynomials $\varphi_\mathbf{q}$ in the quasi-optimal approach, for which indeed $\|\varphi_\mathbf{q}\|_{C^0_\pi}\leq C$, and we will use the following ansatz for the Hermite coefficient of u:

$$|u_\mathbf{q}|\approx C\prod_{n=1}^N \frac{e^{-g_n q_n}}{\sqrt{q_n!}}\,, \tag{19}$$

cf. [5]. Concerning the truncation of the Karhunen–Loève expansion of γ, it is desirable to select the number of random variables N such that essentially the entire spatial variability is taken into account (say more than 99.9%), in order to obtain a negligible distance between the exact solution of Problem 1 $p(\mathbf{x},\omega)$ and the exact solution of Problem 2 $p(\mathbf{x},\mathbf{y})$, and, in turn, between $u(\omega)$ and $u(\mathbf{y})$. As a consequence, the problem will depend on a number of random variables N ranging from a few tens (for choices of ν that yield smooth realizations of a) to several hundreds (for $\nu\to 1/2$). This will require some adaptations of the adaptive algorithm introduced earlier, that will be detailed in the following sections.

4.1 Dimension-adaptive sparse grid algorithm

When considering a large number of random variables, generating and exploring the reduced margin of $\mathbf{I}$ might be computationally intensive. Thus, in the following we present a modified version of the adaptive sparse grid algorithm that starts by working over a parameter space $\widetilde{\Gamma}$ with a moderate dimensionality $\widetilde{N}$ and progressively increases $\widetilde{N}$. Crucially, such strategy actually relieves us from fixing a-priori a truncation for the Karhunen–Loève expansion, i.e. it allows to work with $N=\infty$ random variables.

To this end, we start by assuming that the importance of the random variables in the approximation of the QoI u follows to a good extent the Karhunen–Loève ordering: in other words, y_n may contribute less than y_{n+1} to the variability of the

QoI but there is a certain “dimensional buffer” N_b such that y_n is guaranteed to be more important than y_{n+N_b}. Then, the adaptive algorithm starts by considering $\widetilde{N} = N_b$ random variables only, and whenever a variable y_n with $n < \widetilde{N}$ is activated (cf. Section 3.2), the random variable $y_{\widetilde{N}+1}$ enters the approximation (i.e. the multi-index $\mathbf{i} = \mathbf{1} + \mathbf{e}_{\widetilde{N}+1}$ is included in the reduced margin) and the counter $\widetilde{N}$ is increased by one, so that there is always a buffer of N_b non-activated directions. This strategy is detailed in Algorithm 2.

4.2 *Monte Carlo method with Control Variate (MCCV)*

For values of ν close to $1/2$, the decay of the eigenvalues of the KL expansion of γ is so slow that a very large number of random variables will equally contribute to the variability of the QoI; therefore, even the dimension-adaptive sparse grid algorithm detailed in the previous section may not be effective. In such a case we propose to combine the sparse grid approximation with a Monte Carlo sampling following the ideas proposed in [29]. More precisely, we will introduce an auxiliary problem having a smoothed coefficient a^ε as random permeability, whose solution u^ε can be effectively approximated by a quasi-optimal or an adaptive sparse grid scheme. Then we will use u^ε as control variate in order to define a new QoI, namely u^{CV}, upon which we build a MC estimator.

The first step in order to apply this strategy is to define a proper smoothed random field. Thus, let $\gamma(x,\omega)$ and $\gamma^\varepsilon(x,\omega)$ be two random fields obtained respectively by considering a covariance function of the Matérn family and the convolution of $\gamma(x,\omega)$ with a smooth kernel (e.g. Gaussian),

$$\gamma^\varepsilon(\cdot,\omega) = \gamma(\cdot,\omega) * \phi_\varepsilon(\cdot) \quad \text{where} \quad \phi_\varepsilon(\mathbf{x}) = e^{-\frac{|\mathbf{x}|^2}{2\varepsilon^2}} / (2\pi\varepsilon^2)^{\frac{d}{2}},$$

and let $a^\varepsilon = e^{\gamma^\varepsilon}$. Using this definition, it is easy to see that the smoothed random field γ^ε has a covariance function defined as

$$\mathrm{cov}^\varepsilon_\nu(\mathbf{x},\mathbf{x}') = \mathbb{E}[\gamma^\varepsilon(\mathbf{x},\cdot)\gamma^\varepsilon(\mathbf{x}',\cdot)] = \phi_\varepsilon(\mathbf{x}) * \mathrm{cov}_\nu(\mathbf{x},\mathbf{x}') * \phi_\varepsilon(\mathbf{x}'). \tag{20}$$

Clearly, the smaller the parameter ε is, the more correlated the two random fields γ and γ^ε are, as it can be seen in Figure 1; consistently, $u^\varepsilon \to u$ when $\varepsilon \to 0$.

Next, let us assume for the moment that we know exactly the mean of the control variate $\mathbb{E}[u^\varepsilon]$ and define

$$\tilde{u}^{CV} = u - u^\varepsilon + \mathbb{E}[u^\varepsilon].$$

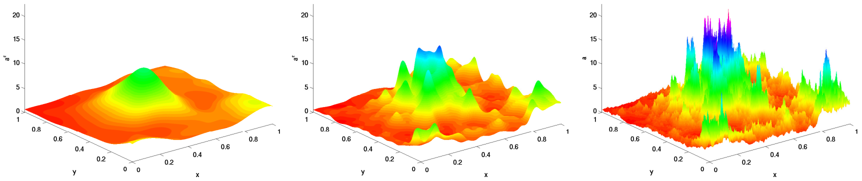
This new variable is such that $\mathbb{E}[\tilde{u}^{CV}] = \mathbb{E}[u]$ and

$$\mathbb{V}\mathrm{ar}(\tilde{u}^{CV}) = \mathbb{V}\mathrm{ar}(u) + \mathbb{V}\mathrm{ar}(u^\varepsilon) - 2\mathrm{cov}(u,u^\varepsilon), \tag{21}$$

showing that the more positively correlated the quantities of interest are, the larger the variance reduction achievable. Although we do not have the exact mean of $u^\varepsilon(\mathbf{y})$

Algorithm 2: Dimension Adaptive Algorithm

```
Note: To avoid ambiguities we write $\mathbf{v}^N$ to make clear that the vector $\mathbf{v}$ has $N$ components;
analogously $\mathbf{I}^N$ indicates that the multi-index set $\mathbf{I}$ is composed of $N$-dimensional vectors.
Dimension adaptive sparse grids(MaxPts, ProfTol, $\pi$, <ProfitName>,$N_b$)
    $\widetilde{N} = N_b$, $\mathbf{A}^{\widetilde{N}} = \mathbf{0}^{\widetilde{N}}$ ;        $\mathbf{A}$ is a Boolean vector indicating which variables are active
    $\mathbf{I}^{\widetilde{N}} = \{\mathbf{1}^{\widetilde{N}}\}$, $\mathbf{G}^{\widetilde{N}} = \{\mathbf{1}^{\widetilde{N}}\}$, $\mathbf{R}^{\widetilde{N}} = \varnothing$, $\mathbf{i}^{\widetilde{N}} = \mathbf{1}^{\widetilde{N}}$, $\mathcal{S}_{old} = \mathcal{S}^m_{\mathbf{I}^{\widetilde{N}}}[f]$, $\mathcal{Q}_{old} = \mathcal{Q}^m_{\mathbf{I}^{\widetilde{N}}}[f]$ ;
    $\mathcal{H} = \mathrm{pts}(\mathcal{S}_{old})$, NbPts=card($\mathcal{S}_{old}$), ProfStop=$\infty$ ;
    while NbPts < MaxPts and ProfStop > ProfTol do
        $\mathcal{N}\mathrm{g} = \mathrm{neigh}(\mathbf{i}^{\widetilde{N}}, \mathbf{I}^{\widetilde{N}})$ ;
        for $\mathbf{j} \in \mathcal{N}\mathrm{g}$ and $\mathbf{I}^{\widetilde{N}} \cup \{\mathbf{j}\}$ is downward closed do
            $\mathbf{G}^{\widetilde{N}} = \mathbf{G}^{\widetilde{N}} \cup \{\mathbf{j}\}$ ;
            $\mathcal{S} = \mathcal{S}^m_{\mathbf{G}^{\widetilde{N}}}[f]$ ;
            $\mathcal{Q} = \mathcal{Q}^m_{\mathbf{G}^{\widetilde{N}}}[f]$ ;
            if using nested points then
                $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{S}) \setminus \mathrm{pts}(\mathcal{S}_{old})$,  NbPts = NbPts + card($\mathcal{N}$ew) ;
                $\mathbf{v}$ = evaluations of $f$ on each $\mathbf{y} \in \mathcal{N}$ew ;
            else
                $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{T}_{m(\mathbf{i})})$, $\mathcal{H} = \mathcal{H} \cup \mathrm{pts}(\mathcal{S})$,  NbPts = card($\mathcal{H}$) ;
                $\mathbf{v}$ = evaluations of $\mathcal{S}$ on each $\mathbf{y} \in \mathcal{N}$ew ;
            $\mathbf{v}_{old}$ = evaluations of $\mathcal{S}_{old}$ on each $\mathbf{y} \in \mathcal{N}$ew ;
            $\boldsymbol{\pi}$ = evaluations of $\pi$ on each $\mathbf{y} \in \mathcal{N}$ew ;
            $P(\mathbf{j})$ = Compute_profit($\mathcal{N}$ew, $\mathbf{v}, \mathbf{v}_{old}, \boldsymbol{\pi}, \mathcal{Q}, \mathcal{Q}_{old}$,<ProfitName>)
            $\mathbf{R}^{\widetilde{N}} = \mathbf{R}^{\widetilde{N}} \cup \{\mathbf{j}\}$
            $\mathcal{S}_{old} = \mathcal{S}$, $\mathcal{Q}_{old} = \mathcal{Q}$ ;
        choose $\mathbf{k}^{\widetilde{N}}$ from $\mathbf{R}^{\widetilde{N}}$ with highest profit; $\mathbf{i}^{\widetilde{N}} = \mathbf{k}^{\widetilde{N}}$ ;
        if $\exists n = 1,\dots,\widetilde{N}$ s.t. $A_n = 0$ and $k_n > 1$ then
            $A_n = 1, \widetilde{N} = \widetilde{N} + 1$ ;        activate n-th variable and update $\widetilde{N}$
            extend the containers $\mathbf{I}, \mathbf{R}, \mathbf{G}, \mathbf{k}, \mathbf{A}$ by adding the new direction.
            $\mathbf{G}^{\widetilde{N}} = \mathbf{G}^{\widetilde{N}} \cup \{\mathbf{1}^{\widetilde{N}} + \mathbf{e}^{\widetilde{N}}_{\widetilde{N}}\}$; $\mathcal{S} = \mathcal{S}^m_{\mathbf{G}^{\widetilde{N}}}[f]$; $\mathcal{Q} = \mathcal{Q}^m_{\mathbf{G}^{\widetilde{N}}}[f]$ ;
            if using nested points then
                $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{S}) \setminus \mathrm{pts}(\mathcal{S}_{old})$,  NbPts = NbPts + card($\mathcal{N}$ew) ;
                $\mathbf{v}$ = evaluations of $f$ on each $\mathbf{y} \in \mathcal{N}$ew ;        cf. eq. (15)
            else
                $\mathcal{N}\mathrm{ew} = \mathrm{pts}(\mathcal{T}_{m(\mathbf{i})})$, $\mathcal{H} = \mathcal{H} \cup \mathrm{pts}(\mathcal{S})$,  NbPts = card($\mathcal{H}$);
                $\mathbf{v}$ = evaluations of $\mathcal{S}$ on each $\mathbf{y} \in \mathcal{N}$ew ;        cf. eq. (16)
            $\mathbf{v}_{old}$ = evaluations of $\mathcal{S}_{old}$ on each $\mathbf{y} \in \mathcal{N}$ew ;
            $\boldsymbol{\pi}$ = evaluations of $\pi$ on each $\mathbf{y} \in \mathcal{N}$ew ;
            $P(\mathbf{1}^{\widetilde{N}} + \mathbf{e}^{\widetilde{N}}_{\widetilde{N}})$ = Compute_profit($\mathcal{N}$ew, $\mathbf{v}, \mathbf{v}_{old}, \boldsymbol{\pi}, \mathcal{Q}, \mathcal{Q}_{old}$,<ProfitName>)
            $\mathbf{R}^{\widetilde{N}} = \mathbf{R}^{\widetilde{N}} \cup \{\mathbf{1}^{\widetilde{N}} + \mathbf{e}^{\widetilde{N}}_{\widetilde{N}}\}$,
            $\mathbf{i}^{\widetilde{N}} = \mathrm{argmax}(\max(P(\mathbf{1}^{\widetilde{N}} + \mathbf{e}^{\widetilde{N}}_{\widetilde{N}}), P(\mathbf{k}^{\widetilde{N}})))$ ;        select $\mathbf{i}^{\widetilde{N}}$ with highest profit
        $\mathbf{I}^{\widetilde{N}} = \mathbf{I}^{\widetilde{N}} \cup \{\mathbf{i}^{\widetilde{N}}\}$, $\mathbf{R}^{\widetilde{N}} = \mathbf{R}^{\widetilde{N}} \setminus \{\mathbf{i}^{\widetilde{N}}\}$
        update ProfStop with a suitable criterion based on the values of $P(\mathbf{j})$
    return $\mathcal{S}$, $\mathcal{Q}$
```



(a) $\varepsilon = 1/2^4$. (b) $\varepsilon = 1/2^6$. (c) Original field.

Fig. 1 Three different regularizations of the same realization of a. $\nu = 0.5$, $L_c = 0.5$, $\sigma = 1$.

at our disposal, we can successfully compute it with a sparse grid method, since $a^\varepsilon(\mathbf{x},\mathbf{y})$ has smooth realizations, and hence the coefficients of the KL expansion are rapidly decreasing, as long as the smoothing parameter ε remains sufficiently large. The final variable on which we will actually apply our MC algorithm is therefore

$$u^{CV} = u - u^\varepsilon + \mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon], \tag{22}$$

and the associated MC control variate estimator (MCCV) is defined as

$$\hat{u}^{MCCV}_M = \frac{1}{M}\sum_{i=1}^{M} u^{CV}(\omega_i) = \frac{1}{M}\sum_{i=1}^{M}\left(u(\omega_i) - u^\varepsilon(\omega_i)\right) + \mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon], \tag{23}$$

where $u^{CV}(\omega_i)$ are i.i.d. realizations of the control variate and M is the sample size. Note that

$$\mathbb{V}\mathrm{ar}(u^{CV}) = \mathbb{V}\mathrm{ar}(\tilde{u}^{CV}). \tag{24}$$

Observe that care must be taken from a computational point of view when generating the samples $u_i(\omega) - u_i^\varepsilon(\omega)$. We propose to generate realizations of $u - u^\varepsilon$ starting from the Fourier expansions of γ and γ^ε: indeed, the Fourier expansion is very convenient when expansions over several random variables are needed, as the basis functions are known analytically; moreover, the Fourier expansions of γ and γ^ε share the same basis functions and differ only by the coefficients. On the other hand, to compute $\mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon]$, it is more convenient to start from a Karhunen–Loève expansion of γ^ε, that needs less variables than a Fourier expansion but whose basis functions need to be determined solving an eigenvalue problem (which is however doable for γ^ε given that it is a smooth field). In other words, two expansions of γ^ε and one expansion of γ will be used simultaneously. In particular, we have considered the following truncated Fourier expansion over an hypercube of size $(2L)^d$, containing the domain D, with $L = \max(6L_c, diam(D))$:

$$\gamma(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{k}\in\mathbf{K}} \sqrt{c_\mathbf{k}} \sum_{\mathbf{n}\in\{0,1\}^d} y^{\mathbf{n}}_{\mathbf{k}}(\omega) \prod_{l=1}^{d} \cos\left(\frac{\pi k_l}{L} x_l\right)^{n_l} \sin\left(\frac{\pi k_l}{L} x_l\right)^{1-n_l},$$

where $\mathbf{K} \subset \mathbb{N}^d$ is a suitable multi-index set having cardinality K, the resulting vector of i.i.d. standard normal random variables is $\mathbf{y} = \{y_{\mathbf{k}}^{\mathbf{n}},\, \mathbf{k} \in \mathbf{K},\, \mathbf{n} \in \{0,1\}^d\}$, and the coefficients $c_{\mathbf{k}}$ are the positive coefficients of the cosine expansion of the covariance function $\mathrm{cov}_\nu(\mathbf{x},\mathbf{x}')$ on $[-L,L]^d$, namely

$$\mathrm{cov}_\nu(\mathbf{x},\mathbf{x}') = \sum_{\mathbf{k}\in\mathbb{N}^d} c_{\mathbf{k}} \prod_{l=1}^{d} \cos\left(\frac{\pi k_l}{L}(x_l - x_l')\right).$$

Consistently, the realization $u_i^\varepsilon(\omega)$ will be computed starting from a truncated Fourier expansion of γ^ε over the same index-set $\mathbf{K}$ and using the same realization $\mathbf{y}(\omega_i)$ used to generate $u_i(\omega)$.

Concerning the mean square error associated to the estimator (23), namely $e(\hat{u}_M^{MCCV})^2 = \mathbb{E}\left[(\hat{u}_M^{MCCV} - \mathbb{E}[u])^2\right]$, the following result holds:

Proposition 1. *The mean square error of the estimator (23) can be split as*

$$e(\hat{u}_M^{MCCV})^2 = \frac{\mathbb{V}ar(u^{CV})}{M} + \Big(\mathbb{E}[u^\varepsilon] - \mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon]\Big)^2. \qquad (25)$$

Proof. We have

$$\begin{aligned} e(\hat{u}_M^{MCCV})^2 &= \mathbb{E}[(\hat{u}_M^{MCCV} - \mathbb{E}[u])^2] = \mathbb{E}\Big[\Big(\sum_{i=1}^{M} \frac{u_i - u_i^\varepsilon}{M} + \mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon] \pm \mathbb{E}[u^\varepsilon] - \mathbb{E}[u]\Big)^2\Big] \\ &= \mathbb{E}\Big[\Big(\frac{1}{M}\sum_{i=1}^{M}(u_i - u_i^\varepsilon - \mathbb{E}[u] + \mathbb{E}[u^\varepsilon])\Big)^2\Big] + \mathbb{E}[(\mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon] - \mathbb{E}[u^\varepsilon])^2] \\ &= \frac{\mathbb{V}ar(u^{CV})}{M} + \Big(\mathbb{E}[u^\varepsilon] - \mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon]\Big)^2. \end{aligned}$$

□

The first term on the right hand side of (25) represents the variance of the estimator $\hat{u}_M^{MCCV}$, i.e. the error coming from the MCCV sampling, and it is expected to be significantly smaller than the variance of the standard MC estimator thanks to the presence of the control variate, cf. equation (24) and (21); the second term represents instead the error due to the approximation of the mean of the smoothed quantity of interest u^ε with a sparse grid scheme. As already hinted, when ε goes to 0 the term $\mathbb{V}\mathrm{ar}(u^{CV})/M$ vanishes, and more precisely, the following result (which is a simplified version of Theorem 5.1 in [29]) holds:

Proposition 2. *Let γ and γ^ε be two Gaussian random fields having covariance functions respectively defined as in (17) and (20); assume $\partial D_D = \partial D$, $f \in L^2(D)$ and $L \in H^{-1}(D)$. Then, $\mathbb{P}$-a.s. in Ω it holds*

$$|u - u^\varepsilon|(\omega) \leq C(\nu,\omega)\varepsilon^{\min(2,\nu)},$$

where the constant $C(\nu,\omega)$ is $L^q_{\mathbb{P}}$-integrable for any $q > 0$ so the bound can also be expressed as

$$\|u - u^\varepsilon\|_{L^q_{\mathbb{P}}(\Omega)} \leq C(\nu,q)\varepsilon^{\min(2,\nu)}.$$

In particular, $\mathbb{V}ar(u^{CV}) \leq C^2(\nu,2)\varepsilon^{2\min(2,\nu)}$.

On the other hand an accurate approximation of $\mathbb{E}[u^{\varepsilon}]$ by a sparse grid scheme might becomes non-advantageous if $\varepsilon \to 0$. The parameter ε should therefore be chosen so as to have a good variance reduction while still keeping a manageable sparse grid approximation problem.

Remark 1 *In this work we do not address the issue of the spatial approximation of Problem 2. In general all the results previously presented still hold if a finite dimensional subspace $V_h \subset V$, e.g. a finite element space, is considered in order to approximate functions in V.*

5 Numerical results

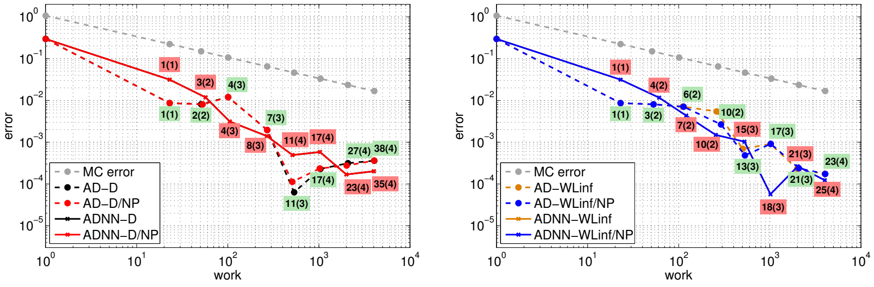
In this section we present the convergence results obtained for the Darcy problem on the unit square $D = (0,1)^2$ with $f = 0$, Dirichlet boundary conditions $g(\mathbf{x}) = 1 - x_1$ on $\partial D_D = \{\mathbf{x} \in \partial D : x_1 = 0 \text{ or } x_1 = 1\}$, and homogeneous Neumann conditions on the remaining part of ∂D; the spatial approximation of the Darcy problem is done by piecewise linear finite elements defined on a structured mesh.

We will consider two cases: first we will solve Problem 2 with a smooth random field a, corresponding to the choice $\nu = 2.5$ in (17), and then we will move to the rough random field corresponding to $\nu = 0.5$, in which case we will consider the MCCV approach. In both cases we set $\sigma = 1$ and $L_c = 0.5$, while the mesh over $\overline{D}$ consists of 33×33 vertices in the case $\nu = 2.5$ and 65×65 vertices in the case $\nu = 0.5$; both meshes have been verified to be sufficiently refined for our purposes. In particular we will be interested in approximating the expected value of the functional

$$u(\omega) = \int_0^1 a(1,x_2,\omega)\frac{\partial p}{\partial x_1}(1,x_2,\omega)dx_2, \tag{26}$$

which represents the mass flow on the outlet. The aims of this section are:

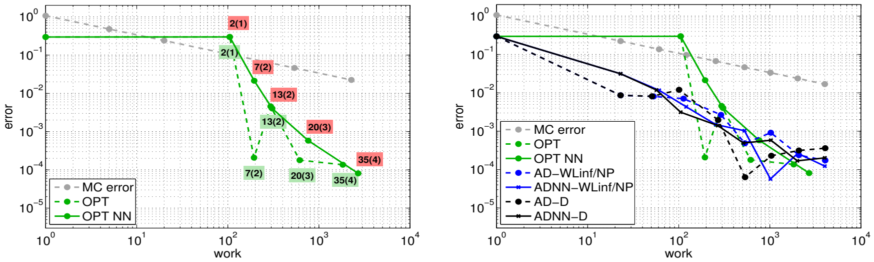
1. establish whether using non-nested points in an adaptive sparse grid framework might be convenient or not;
2. verify the performance of adaptive sparse grids built with different profit indicators;
3. compare the performance of the adaptive sparse grids with that of the quasi-optimal sparse grids (note that our previous numerical experiences suggest that indeed these two sparse grid constructions behave similarly when used to solve UQ problems depending on uniform random variables if nested univariate points are used, see [4, 25]);
4. test the effectiveness of using an adaptive (or quasi-optimal) sparse grid construction as control variate in a MC framework in order to tackle also problems depending on rough coefficients.



(a) Profit indicators: deltaint (AD-D and ADNN-D), deltaint / new points (AD-D/NP and ADNN-D/NP).

(b) Profit indicator: Weighted Linf (AD-WLinf and ADNN-WLinf), Weighted Linf / new points (AD-WLinf/NP and ADNN-WLinf/NP).

Fig. 2 Case $\nu = 2.5$, adaptive sparse grids error.



(a) Quasi-optimal sparse grids error.

(b) Comparison adaptive / quasi-optimal.

Fig. 3 Case $\nu = 2.5$, quasi-optimal sparse grids error (left) and a comparison between adaptive and quasi-optimal schemes (right).

Smooth case: $\nu = 2.5$

In this case we deal with an input random field with twice differentiable realizations; therefore the eigenvalues of the Karhunen–Loève expansion decay quickly enough to justify the use of the N-adaptive sparse grid algorithm to approximate the QoI.

For this test, we consider as a reference solution the approximation of the QoI obtained with a quasi-optimal sparse grid with approximately 8300 quadrature points base of Gauss–Hermite abscissas, for which 45 out of the first 50 random variables of the KL expansion are active: observe that this is sufficient to take into account 99.99% of the total variability of the permeability field, i.e. there is essentially no KL truncation error. We monitor the convergence of the error measured as

$$err(w) \approx |\mathcal{Q}^m_{\mathbf{I}(w)}[u] - \mathcal{Q}^m_{\mathbf{I}(w_{ref})}[u]|,$$

i.e. the absolute value of the sparse grid quadrature error, where $\mathbf{I}(w), w = 0,1,2,\dots,$ are the sequences of multi-index sets generated either by the adaptive or the quasi-optimal sparse grid scheme and $\mathbf{I}(w_{ref})$ is the multi-index set corresponding to the above-mentioned reference solution. More specifically, the sets $\mathbf{I}(w)$ for the adaptive strategies are obtained by stopping the algorithm as soon as at least $W_{max}(w)$ points have been added to the sparse grid (including the points needed for the exploration of the reduced margin), with $W_{max}(w) = \{1, 20, 50, 100, 250, 500, 1000, 2000, 4000\}$, for $w = 0, \dots, 8$. As for the quasi-optimal sparse grids, the sets $\mathbf{I}(w)$ are defined as

$$\mathbf{I}(w) = \left\{\mathbf{i} \in \mathbb{N}_+^N : P(\mathbf{i}) \geq e^{-w}\right\} \tag{27}$$

with $w = 0, 1, \dots, 5$, the reference solution being obtained with $w = 6$. We recall that the profits $P(\mathbf{i})$ are defined as the ratios between the error and work contributions, $P(\mathbf{i}) = \Delta E(\mathbf{i})/\Delta W(\mathbf{i})$, where $\Delta E(\mathbf{i})$ are estimated combining equations (19) and (9), and $\Delta W(\mathbf{i})$ are defined either as (10) or (13).

The computational cost associated to each sparse grid is expressed in terms of number of linear system solves. For the adaptive sparse grids, this count also includes the cost of the exploration of the reduced margin. Moreover, when using non-nested points we also take into account the system solves related to the points that have been included and then excluded from the sparse grid, cf. equation (12). As for the quasi-optimal sparse grids, their construction requires some additional solves to estimate the parameters $g_1, \dots, g_N$ in (19), cf. [25, 4]. More precisely, the n-th rate is estimated by fixing all variables but y_n to their expected value, computing the value of the QoI increasing the number of collocation points along y_n and then fitting the resulting interpolation error: in practice, this amounts to solving 25 linear systems per random variable, which are included in the work count.

We start our discussion from Figure 2, where we show the convergence results obtained with the dimension-adaptive Algorithm 2 varying the choice of profit indicators (cf. Algorithm 1) and the choice of interpolation points, i.e. Genz–Keister versus Gauss–Hermite points, the latter denoted by a suffix NN in the plot, as per “non nested” (cf. Table 1); in this test, we have set the buffer size to $N_b = 10$. More specifically, we used the “deltaint-based” profit indicators in Figure 2-(a) (D and D/NP in the plots, where NP stands for “divided by number of points”) and “weighted L^∞-based” profit indicators in Figure 2-(b) (WLinf and WLinf/NP in the plots). In both cases we observe that there is not much difference between the profit indicators that take into account the number of points and the ones which do not; also the choice of nested or non nested nodes does not seems to affect the convergence.

The numbers next to each point give information about the shape of the multi-index sets $\mathbf{I}(w)$ generated by the adaptive algorithm, and consequently on the distribution of the sparse grid points on the $\widetilde{N}$-dimensional parameter space. The first number (out of the brackets) indicates the number of active directions, while the second number (in the brackets) denotes the highest dimensionality of the tensor grids composing the sparse grid, cf. equation (4). Here and in the following, green labels refer to grids with nested points, while red labels to grids with non-nested points: we show only two series of labels per plot, due to the fact that accounting for the work contributions in the profit definition does not seem to play a role in this test and, consequently, the sequences of sets $\mathbf{I}(w)$ generated by AD-D and AD-D/NP are essentially identical (and the same for ADNN-D and ADNN-D/NP). Observe that after ≈ 20 problems solves the algorithm has activated "only" 1 variable due to the fact the at the beginning of the algorithm N_b variables must be explored, requiring $1+2N_b = 21$ solver calls, in order to decide which variable should be activated as second; moreover the number of "active" variables is always smaller than $N = 45$, which is the number of "active" variables for the reference solution. In Figure 2-(a) and 2-(b) we have also added the convergence curve for a plain MC approximation. This has been generated as $\sqrt{\mathbb{V}\mathrm{ar}(u)/M(w)}$, with $\mathbb{V}\mathrm{ar}(u)$ estimated as $\mathbb{V}\mathrm{ar}(u) \approx \mathcal{Q}^m_{\mathbf{I}(w_{ref})}[u^2] - (\mathcal{Q}^m_{\mathbf{I}(w_{ref})}[u])^2$.

In Figure 3-(a) we show instead the errors obtained by using quasi-optimal sparse grid approximations of the QoI built on Genz–Keister and Gauss–Hermite knots (labeled OPT and OPT NN respectively). Observe that since we build the sets $\mathbf{I}(w)$ in (27) again with a "buffered" procedure analogous to the one described in Section 4.1, the rate g_n is computed only at the level w for which y_n enters the buffer of random variables, and such work is thus accounted for at level w; this explains the initial plateau that can be seen in the convergence. Again, the labels next to each point represent the number of active variables (outside the brackets) and the number of variables activated at the same time (in the brackets). These numbers suggest that, for the same work, the adaptive sparse grids seem to activate a slightly smaller number of variables than the quasi-optimal ones, while the tensor grids dimensionality seems to be comparable. Also for the quasi-optimal sparse grids the number of "active" variables is always smaller than $N = 45$.

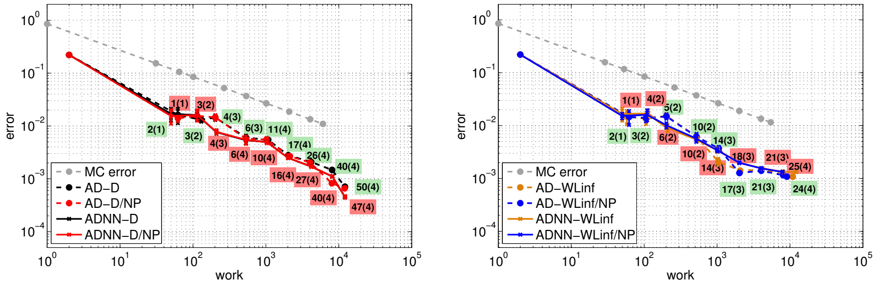
Finally, Figure 3-(b) shows a comparison between the quasi-optimal and the adaptive schemes; among the adaptive schemes presented we take into account for this comparison the profit indicators deltaint and Weighted Linf / new points. We can observe that, except for small values of work for which the cost needed to compute the parameters g_i in eq. (19) largely dominates the cost needed to actually compute the quasi-optimal sparse grid approximation, the quasi-optimal and the adaptive schemes behave similarly.

Rough case: $\nu = 0.5$

In this case we deal with a rough input random field a that has realizations which are not even differentiable: thus, the slow decay of the eigenvalues of the Karhunen–Loève expansion may render unfavorable a sparse grid approach, even considering advanced techniques like the adaptive or the quasi-optimal schemes. Therefore we now solve the problem by the MCCV approach introduced in the Subsection 4.2, using $\varepsilon = 2^{-5}$ as smoothing parameter.

At each sparse grid approximation level w we use $M(w) = W_{\mathbf{I}(w),m}$ samples in the MCCV estimator, i.e. we balance the work of the sparse grid and that of the MC sampling so that the total work is $2W_{\mathbf{I}(w),m}$; other work splitting, e.g. balancing the two error contributions of the method detailed in Proposition 1, could be considered as well.

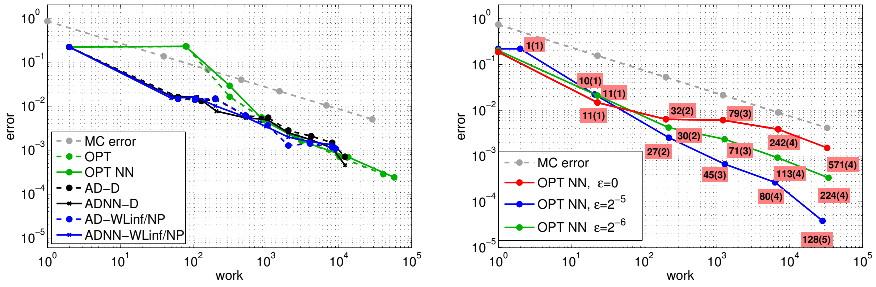
In practice, we will approximate the sparse grid component of the error by considering a reference solution obtained with a quasi-optimal sparse grid built with approximately 86500 nodes based on Gauss–Hermite abscissas with $N = 163$ active random



(a) Profit indicators: deltaint (AD-D and ADNN-D), deltaint / new points (AD-D/NP and ADNN-D/NP).

(b) Profit indicator: Weighted Linf (AD-WLinf and ADNN-WLinf), Weighted Linf / new points (AD-WLinf/NP and ADNN-WLinf/NP).

Fig. 4 Case $\nu = 0.5$, MCCV-adaptive sparse grid mean error. Bars represent 3 standard deviation of the sampling error.



(a) Comparison MCCV-adaptive / MCCV-quasi-optimal sparse grid mean error.

(b) Quasi-optimal sparse grid error for different values of ε.

Fig. 5 Case $\nu = 0.5$. Left: comparison between MCCV-adaptive and MCCV-quasi-optimal schemes Right: Quasi-optimal sparse grid error for different values of ε.

variables, that takes into account 99.99% of the total variability of the smoothed field, and the sampling component by estimating $\mathbb{V}\mathrm{ar}(\tilde{u}^{CV}) \approx \widehat{u^2}^{MCCV}_{M(w)} - (\hat{u}^{MCCV}_{M(w)})^2$; as mentioned in Section 4.2, the sampling component is based on a Fourier expansion of the non-smoothed field γ, that has been truncated after $129 \times 129 = 16641$ random variables. To summarize, we have

$$err(w) \approx \sqrt{\frac{\widehat{u^2}^{MCCV}_{M(w)} - (\hat{u}^{MCCV}_{M(w)})^2}{M(w)}} + \left| \mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon] - \mathcal{Q}^m_{\mathbf{I}(w_{ref})}[u^\varepsilon] \right|.$$

In Figure 4 we show the performance of the MCCV algorithm with adaptive sparse grids. Since we are running a sampling method, we also add to the plot error bars indicating the interval spanning ±3 standard deviations of the error from its average value, assessed over 4 runs of the method. The considerations that can be made by looking at these plots are similar to the ones we did in the case $\nu = 2.5$, i.e. there is basically no difference between the profit indicators that take into account the number of points and those which do not; also changing the family of nodes does not seem to have a substantial impact on the quality of the approximation. Observe that since we are balancing the works of the Monte Carlo sampling and of the sparse grid, the observed convergence rate is larger than the MC rate 1/2 for little values of work, where the sparse grid error dominates the sampling error and converges with a faster rate than 1/2 (remember that the sparse grid is applied to a smoothed problem). For large w, the sampling error dominates the sparse grid error and one essentially recovers the MC rate 1/2, however with a much smaller constant than MC due to the presence of the control variate.

Figure 5-(a) shows the convergence of the MCCV method combined with quasi-optimal sparse grids and compares the results obtained with the adaptive sparse grids procedures. Again, among the adaptive schemes presented we consider the profit indicators deltaint and Weighted Linf / new points. For both quasi-optimal and adaptive schemes we only plot the average error of the Quantity of Interest over four runs. As in the smooth case, for sufficiently large values of work, all the schemes perform similarly. Note that in the quasi-optimal case, the quantities g_n are actually computed for the first 50 random variables only, after which we instead set $g_n = \sqrt{\lambda_n}$, cf. equation (18), i.e. we approximate g_n with the value of the corresponding coefficient of the KL expansion. Indeed, these random variables have a moderate impact on the solution and numerical cancellations effects may significantly affect the results of the fitting procedure.

Finally, 5-(b) shows the convergence of the quasi-optimal sparse grid error $|\mathcal{Q}^m_{\mathbf{I}(w)}[u^\varepsilon] - \mathcal{Q}^m_{\mathbf{I}(w_{ref})}[u^\varepsilon]|$ for different values of ε. It is clearly visible that the convergence rate deteriorates as ε decreases, thus motivating the introduction of the MCCV approach. Observe that for the sake of comparison, in this plot the work needed to determine the rates (that would cause an initial plateau in the convergence plot) has been neglected. We also observe that in this case there is a significant difference between the number of random variables activated by the quasi-optimal and adaptive sparse grid schemes. In fact, the latter tends to activate less variables than the former, adding conversely more points on the activated ones.

6 Conclusions

In this work we have proposed an improved version of the classical adaptive sparse grid algorithm, that can handle non-nested collocation points and unbounded domains, and can be used for an arbitrary large number of random variables, assuming that a "rough ordering" of the variables according to their importance is available. We have also implemented several indicators to drive the adaptation process.

We have then used this algorithm to solve a Darcy equation with random lognormal permeability, and compared the results obtained by changing collocation points and adaptivity indicators against those obtained by the quasi-optimal sparse grids algorithm. The computational analysis has been performed first on a case with smooth permeability realizations, and then in the case of rough realizations: in the latter case, we have actually considered the sparse grid in a Monte Carlo Control Variate approach, in which the sparse grids are applied to a smoothed problem and the results serve as control variate for a Monte Carlo sampling for the rough problem. The numerical results seem to suggest that

1. using non-nested points in an adaptive sparse grid framework yields results that are comparable to those obtained by nested points, at least in the log-normal context;
2. changing the indicator driving the adaptivity process does not have a dramatic impact on the quality of the solution; this however may be due to the specific choice of the QoI considered here, and more testing should be performed;
3. the adaptive and the quasi-optimal sparse grids perform similarly on lognormal problems, in agreement with our previous findings on uniform random variables;
4. in the case of smooth log-permeability fields the adaptive and the quasi-optimal sparse grids give quite satisfactory results;
5. in the case of rough fields the adaptive / quasi-optimal sparse grids *alone* have a performance asymptotically similar to a standard MC (with just a slight improvement on the constant) and we do not advocate their use in such a case; on the other hand the results are satisfactory if the sparse grids are used as control variate in a MC sampling.

Acknowledgements F. Nobile, F. Tesei and L. Tamellini have received support from the Center for ADvanced MOdeling Science (CADMOS) and partial support by the Swiss National Science Foundation under the Project No. 140574 “Efficient numerical methods for flow and transport phenomena in heterogeneous random porous media”. R. Tempone is a member of the KAUST SRI Center for Uncertainty Quantification in Computational Science and Engineering.

References

1. I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Review*, 52(2):317–355, June 2010.
2. J. Bäck, F. Nobile, L. Tamellini, and R. Tempone. Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison. In J. Hesthaven and E. Ronquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 43–62. Springer, 2011. Selected papers from the ICOSAHOM ’09 conference, June 22-26, Trondheim, Norway.
3. V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.*, 12(4):273–288, 2000.
4. J. Beck, F. Nobile, L. Tamellini, and R. Tempone. On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods. *Mathematical Models and Methods in Applied Sciences*, 22(09), 2012.
5. J. Beck, F. Nobile, L. Tamellini, and R. Tempone. A Quasi-optimal Sparse Grids Procedure for Groundwater Flows. In M. Azaïez, H. El Fekih, and J. S. Hesthaven, editors, *Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM 2012*, volume 95 of *Lecture*

Notes in Computational Science and Engineering, pages 1–16. Springer, 2014. Selected papers from the ICOSAHOM '12 conference.

6. H. Bungartz and M. Griebel. Sparse grids. *Acta Numer.*, 13:147–269, 2004.
7. J. Charrier. Strong and weak error estimates for elliptic partial differential equations with random coefficients. *SIAM J. Numer. Anal.*, 50(1), 2012.
8. A. Chkifa, A. Cohen, and C. Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Foundations of Computational Mathematics*, pages 1–33, 2013.
9. K. Cliffe, M. Giles, R. Scheichl, and A. Teckentrup. Multilevel monte carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3–15, 2011.
10. B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, New York, second edition, 2002.
11. P. Diggle and P. J. Ribeiro. *Model-based geostatistics*. Springer, 2007.
12. M. S. Eldred and J. Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. American Institute of Aeronautics and Astronautics Paper 2009-0976.
13. H. C. Elman, C. W. Miller, E. T. Phipps, and R. S. Tuminaro. Assessment of Collocation and Galerkin approaches to linear diffusion equations with random data. *International Journal for Uncertainty Quantification*, 1(1):19–33, 2011.
14. O. Ernst and B. Sprungk. Stochastic collocation for elliptic PDEs with random data: The lognormal case. In J. Garcke and D. Pflüger, editors, *Sparse Grids and Applications - Munich 2012*, volume 97 of *Lecture Notes in Computational Science and Engineering*, pages 29–53. Springer International Publishing, 2014.
15. J. Foo, X. Wan, and G. Karniadakis. The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications. *Journal of Computational Physics*, 227(22):9572–9595, 2008.
16. A. Genz and B. D. Keister. Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight. *J. Comput. Appl. Math.*, 71(2):299–309, 1996.
17. T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71(1):65–87, 2003.
18. C. J. Gittelson. Stochastic Galerkin discretization of the log-normal isotropic diffusion problem. *Math. Models Methods Appl. Sci.*, 20(2):237–263, 2010.
19. H. Harbrecht, M. Peters, and M. Siebenmorgen. Multilevel accelerated quadrature for PDEs with log-normal distributed random coefficient. preprint 2013-18, Universität Basel, 2013.
20. J. D. Jakeman, R. Archibald, and D. Xiu. Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. *J. Comput. Phys.*, 230(10), 2011.
21. A. Klimke. *Uncertainty modeling using fuzzy arithmetic and sparse grids*. PhD thesis, Universität Stuttgart, Shaker Verlag, Aachen, 2006.
22. S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. Wiley-Interscience series in discrete mathematics and optimization. J. Wiley & Sons, 1990.
23. A. Narayan and J. Jakeman. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. arXiv arXiv:1404.5663, e-print, 2014.
24. F. Nobile, L. Tamellini, and R. Tempone. Comparison of Clenshaw–Curtis and Leja quasi-optimal sparse grids for the approximation of random PDEs. MATHICSE report 41/2014, EPFL, 2014. Submitted.
25. F. Nobile, L. Tamellini, and R. Tempone. Convergence of quasi-optimal sparse grids approximation of Hilbert-valued functions: application to random elliptic PDEs. Mathicse report 12/2014, EPFL, 2014. Submitted.
26. F. Nobile, R. Tempone, and C. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.
27. C. Schillings and C. Schwab. Sparse, adaptive Smolyak quadratures for Bayesian inverse problems. *Inverse Problems*, 29(6), 2013.

28. S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 4:240–243, 1963.
29. F. Tesei and F. Nobile. A Multi Level Monte Carlo Method with Control Variate for elliptic PDEs with log-normal coefficients. Mathicse report 49/2019, EPFL, 2014.
30. G. Wasilkowski and H. Wozniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity*, 11(1):1 – 56, 1995.
31. G. Zhang, D. Lu, M. Ye, M. Gunzburger, and C. Webster. An adaptive sparse-grid high-order stochastic collocation method for bayesian inference in groundwater reactive transport modeling. *Water Resources Research*, 49(10):6871–6892, 2013.

Recent publications:

MATHEMATICS INSTITUTE OF COMPUTATIONAL SCIENCE AND ENGINEERING
Section of Mathematics
Ecole Polytechnique Fédérale
CH-1015 Lausanne

40.2014 Diane Guignard, Fabio Nobile, Marco Picasso:
A posteriori error estimations for elliptic partial differential equations with small uncertainties

41.2014 Fabio Nobile, Lorenzo Tamellini, Raúl Tempone:
Comparison of Clenshaw-Curtis and Leja quasi-optimal sparse grids for the approximation of random PDEs

42.2014 Assyr Abdulle, Patrick Henning:
A reduced basis localized orthogonal decomposition

43.2014 Paola F. Antonietti, Ilario Mazzieri, Alfio Quarteroni:
Improving seismic risk protection through mathemactical modeling

44.2014 Luca Dedè, Alfio Quarteroni, Sehngfeng Zhu:
Isogeometric analysis and proper orthogonal decomposition for parabolic problems

45.2014 Zvonimir Bujanovic, Daniel Kressner:
A block algorithm for computing antitriangular factorizations of symmetric matrices

46.2014 Assyr Abdulle:
The role of numerical integration in numerical in numerical homogenization

47.2014 Andrea Manzoni, Stefano Pagani, Toni Lassila:
Accurate solution of Bayesian inverse uncertainty quantification problems using model and error reduction methods

48.2014 Marco Picasso:
From the free surface flow of a viscoelastic fluid towards the elastic deformation of a solid

49.2014 Fabio Nobile, Francesco Tesei:
A multi level Monte Carlo method with control variate for elliptic PDEs with log-normal coefficients

***

01.2015 Peng Chen, Alfio Quarteroni, Gianluigi Rozza:
Reduced order methods for uncertainty quantification problems

02.2015 Federico Negri, Andrea Manzoni, David Amsallem:
Efficient model reduction of parametrized systems by matrix discrete empirical interpolation

03.2015 Giovanni Migliorati, Fabio Nobile, Raúl Tempone:
Convergence estimate in probability and in expectation for discrete least squares with noisy evaluations at random points

04.2015 Fabio Nobile, Lorenzo Tamellini, Francesco Tesei, Raúl Tempone:
An adaptive sparse grid alorithm for elliptic PDEs with lognormal diffusion coefficent