

# A Resilient Domain Decomposition Polynomial Chaos Solver for Uncertain Elliptic PDEs

Paul Mycek<sup>a,\*</sup>, Andres Contreras<sup>b</sup>, Olivier Le Maître<sup>c</sup>, Khachik Sargsyan<sup>d</sup>, Francesco Rizzi<sup>d</sup>,  
Karla Morris<sup>d</sup>, Cosmin Safta<sup>d</sup>, Bert Debuschere<sup>d</sup>, Omar Knio<sup>a</sup>

<sup>a</sup>*Duke University, Dept. of Mechanical Engineering and Materials Science, 144 Hudson Hall, Box 90300,  
Durham, NC 27708.*

<sup>b</sup>*Duke University, Durham, NC.*

<sup>c</sup>*Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur, Orsay, France.*

<sup>d</sup>*Sandia National Laboratories, Livermore, CA.*

---

## Abstract

A resilient method is developed for the solution of uncertain elliptic PDEs on extreme scale platforms. The method is based on a hybrid domain decomposition, polynomial chaos (PC) framework that is designed to address soft faults. Specifically, parallel and independent solves of multiple deterministic local problems are used to define PC representations of local Dirichlet boundary-to-boundary maps that are used to reconstruct the global solution. A LAD-lasso type regression is developed for this purpose. The performance of the resulting algorithm is tested on an elliptic equation with an uncertain diffusivity field. Different test cases are considered in order to analyze the impacts of correlation structure of the uncertain diffusivity field, the stochastic resolution, as well as the probability of soft faults. In particular, the computations demonstrate that, provided sufficiently many samples are generated, the method effectively overcomes the occurrence of soft faults.

*Keywords:* Resilience, exascale computing, uncertainty quantification, polynomial chaos

*PACS:* 02

---

## 1. Introduction

As high performance computing (HPC) evolves towards exascale [1, 2], new scientific challenges need to be addressed to achieve reliable computations. One of the main obstacles is that systems 1,000 times more powerful, in terms of floating-point operations per second (flops), than today's leading petascale platforms are also expected, because of significantly higher error rates, to fail more frequently [3]. As a matter of fact, because clock rates are no longer increasing (or increasing very slowly), the increase in flops will mainly result from a significant increase in the number of processing units. Other challenges brought by extreme-scale computing include the need to

---

\*Corresponding author

*Email addresses:* paul.mycek@duke.edu (Paul Mycek), andres.contreras@duke.edu (Andres Contreras), olm@limsi.fr (Olivier Le Maître), ksargsy@sandia.gov (Khachik Sargsyan), fnrizzi@sandia.gov (Francesco Rizzi), knmorri@sandia.gov (Karla Morris), csafta@sandia.gov (Cosmin Safta), bjdebus@sandia.gov (Bert Debuschere), omar.knio@duke.edu (Omar Knio)

operate with relatively low memory per core, to cope with costly data movement, to make use of heterogeneous hardware, and to scale to very large numbers of cores [4].

There are two main ways of coping with system faults, namely fault-tolerance and resilience. Fault-tolerance techniques consist in detecting errors and recovering from them, while resilience techniques are designed to keep the application running to a correct solution in a timely and efficient manner despite system faults [1, 2]. One popular fault-tolerance technique for petascale systems is checkpoint-restart. It relies on periodic saves of the system state, which allows one to restore the system to a previous state whenever an error is detected. However, in exascale systems, the time needed for checkpointing may be close to the mean time between failures [5, 3], thus causing the system to spend most of its time checkpointing and restarting, rather than advancing towards the solution [2].

In the last few years, new approaches have been explored to deal with system faults. The Local Failure Local Recovery (LFLR) strategy, focusing on local checkpointing and recovery was proposed as an improvement of the original global checkpoint-restart [6]. Alternative approaches include algorithm-based fault tolerance (ABFT) [7, 8, 9, 10], effective use of state machine replication [11] or process-level redundancy [12], and algorithmic error correction code [13]. Many other approaches for resilience in extreme-scale computing have been developed (see, *e.g.*, [2]). Nevertheless, most of these new developments deal with fault-tolerance rather than resilience, as they rely on the detection of the faults in order to mitigate them.

Recently, our group has developed a soft fault resilient solver for elliptic partial differential equations (PDEs), see [14]. To deal with soft faults, the solver represents the solution as a state-of-knowledge, and updates this state in a resilient manner. In this framework, hard faults, such as a node crashing or a communication failing, are seamlessly treated as missing data and may thus be disregarded. One of the strengths of this approach lies in that it does not rely on fault detection, and therefore genuinely provides resilience, rather than fault-tolerance. This feature is particularly interesting to address silent faults, which, as their name suggests, are hard or impossible to detect. The solver is based on an overlapping domain decomposition method (see, *e.g.*, [15, 16]) and the resilient update involves the solving of local problems (independent from one subdomain to another), which is well suited for the parallel solving of large problems. Besides resilience, the solver also presents the advantage of requiring fewer communications, which benefits scalability.

The present work focuses on the development of a resilient elliptic solver for uncertainty quantification (UQ) in exascale computations, building on our previous effort [14]. Specifically, we address the situation where the model PDEs involve uncertain coefficients, and one wants to characterize the resulting uncertainty in the model solution. Our primary objective is to study the resilience of the proposed solver, and so we focus on linear elliptic problems in one spatial dimension. In addition, we restrict our attention to soft faults, namely faults that do not cause the program to terminate immediately but, rather, corrupt numbers and thus lead to erroneous computations [17]. Such faults will be modeled as random bit-flips in the numbers' binary representation, introduced with a prescribed probability in our simulations. Regarding uncertainty, a probabilistic approach is considered relying on stochastic spectral methods, specifically Wiener-type Polynomial Chaos (PC) approximations [18, 19]. The PC method assumes a representation of the uncertain PDE coefficients in terms of a (finite) number of independent random variables, and relies on a spectral expansion of the uncertain PDE solution on a suitable stochastic basis of multi-variate polynomials in these random variables. The efficiency of the PC methods for elliptic problems is due to the smoothness of the elliptic problem solution with respect to the random PDE coefficients, which provides spectral (exponential) convergence of the representation as the polynomial degree is in-

creased. This feature has motivated many works over the last 20 years, and several alternatives have been proposed to efficiently compute the PC expansion of the solution. Classically, see [19], PC approaches are separated into non-intrusive (NI) and stochastic Galerkin methods (*i.e.* methods of weighted residual). In NI methods, the PC expansion coefficients of the solution are estimated from a sample set of deterministic solves corresponding to (usually carefully) selected values of the PDE coefficients. The NI methods, such as the NI spectral projection (NISP, see *e.g.* [20, 21, 22, 23]) and collocation approaches (*e.g.* [24, 25]), only require the availability of a deterministic solver. In contrast, stochastic Galerkin methods involve a reformulation of the original problem, generally leading to a deterministic system of coupled PDEs for the PC expansion coefficients of the solution, whose efficient resolution requires dedicated strategies [26, 27, 28, 29, 30, 31, 32, 33]. As a result, making a stochastic Galerkin solver resilient to soft faults appears as quite a difficult task. On the contrary, making resilient NI methods is much easier as it suffices to rely on a resilient deterministic solver (for instance the solver in [14]) to compute the solution at sampled values of the PDE coefficients. In addition, these deterministic solutions are independent and can be computed in parallel similar to a Monte Carlo approach.

Our observation that the deterministic solver is made resilient by sampling the boundary values of the local problems (over the subdomains) suggests an alternative way to achieve resilience in the stochastic case: we propose here to extend the deterministic approach in [14] by sampling *jointly* the local boundary values *and* the PDE coefficients. In doing so, we expect to have more information available that can be used to recover from soft faults, and to be more effective in computing the whole solution (*i.e.* its PC expansion), compared to the case where one would proceed only locally in the random parameter space. In other words, we want to exploit the known smoothness of the solution with respect to the random parameters to effectively remedy soft faults. This strategy also keeps the amount of global communications to a minimum. The proposed resilient domain decomposition solver then consists in finding the PC expansion of the solution at the boundaries of the subdomains in order to satisfy a system of compatibility conditions obtained by stochastic Galerkin projection. The key point of the method is the construction of the Galerkin system expressing compatibility of the subdomains' boundary values. To make this construction resilient to soft faults, we associate the joint sampling approach to robust regression techniques that overcome the presence of soft faults (seen in this context as outliers). As a result, the proposed solver is hybrid in the sense that it mixes a (resilient) NI approach for the approximation of the stochastic compatibility conditions, with a Galerkin projection to determine the solution at the boundaries of the subdomains. The novelties of this work are many fold. First, the sampling of the PDE coefficients is performed at the subdomain PDE solve stage, as opposed to a fully NI approach consisting of an outer sampling of a deterministic resilient algorithm. Second, we develop a new LAD-lasso [34] type of regression, with properties similar to the elastic net [35], together with an efficient, resilient cross-validation procedure to find the optimal regularization parameter. A general algorithm to solve such regression problems is also developed. Third, the formulation of the stochastic domain decomposition approach is hybrid and mixes a NI approach for the construction of the compatibility relations and a Galerkin approach for its resolution.

The paper is organized as follows. In Section 2 we outline the domain decomposition framework and the deterministic algorithm for one-dimensional linear, elliptic PDEs. Section 3 discusses the extension to parameterized stochastic PDEs, introducing the PC discretization and describing the sampling approach. Section 4 is dedicated to the derivation of the robust regression techniques that provide resilience to soft faults, as well as stability for large problems. The resilience of the solver is studied in Section 5, for the test case of a diffusion equation with uncertain diffusivity. Finally,

general conclusions as well as a discussion of the proposed strategy, including ongoing work and potential improvements, are presented in Section 6.

## 2. Deterministic preliminaries

In this section we briefly recall the domain decomposition approach used in [14] for the resilient solution of deterministic problems. We start with the following 1D boundary value problem (BVP):

$$\begin{cases} \mathcal{L}u = g, & \text{in } \Omega = (0, 1) \\ u(0) = U_0, \\ u(1) = U_1, \end{cases} \quad (1)$$

where  $\mathcal{L}$  is a linear, second-order, elliptic operator. In addition, we assume that the problem is well-posed, *i.e.* it has a unique solution that continuously depends on the data, namely the boundary data,  $U_0$  and  $U_1$ , and the source field,  $g$ . Although this 1D problem is not rigorously speaking a *partial* differential equation (PDE), we shall nonetheless refer to it as such since our approach is meant to be generalized in 2D and 3D.

### 2.1. Domain decomposition and condensed problem

The domain  $\Omega$  is decomposed into  $N$  overlapping subdomains  $\Omega_d = (X_d^-, X_d^+)$ , with

$$X_1^- = 0, \quad X_N^+ = 1, \quad \text{and } X_d^- < X_d^+, \quad d = 1, \dots, N. \quad (2)$$

The subdomains are defined such that

$$\overline{\cup_{d=1}^N \Omega_d} = \bar{\Omega}, \quad \text{and } \Omega_d \cap \Omega_{d+1} \neq \emptyset \quad d = 1, \dots, N-1. \quad (3)$$

The decomposition of  $\Omega$  and the notations used later on are illustrated in Fig. 1.

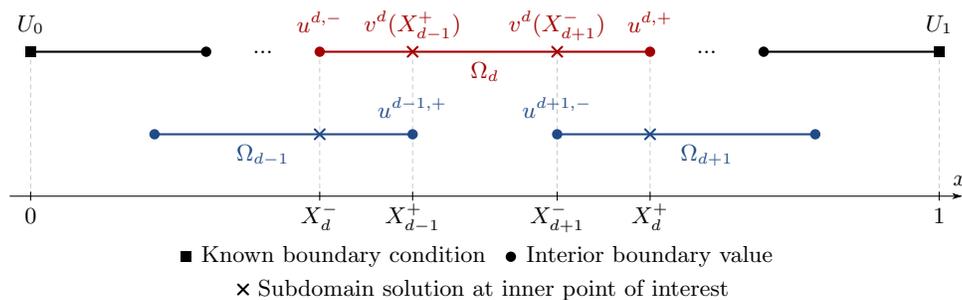


Figure 1: Illustration of the domain decomposition for  $\Omega = (0, 1)$ . The dependence of  $v_d$  on the boundary values has been dropped for simplicity.

Let us define the subproblem (4) associated with a subdomain  $\Omega_d$  as follows:

$$\begin{cases} \mathcal{L}v = g & \text{in } \Omega_d = (X_d^-, X_d^+) \\ v(X_d^-) = u^{d,-}, \\ v(X_d^+) = u^{d,+}, \end{cases} \quad (4)$$



the subproblems to obtain a (convergent) sequence of iterates for  $\mathbf{u}$ . In the context of exascale computing, where the solution of the subproblems is prone to soft faults, ensuring the global resilience of such an iterative approach is not straightforward. This observation has motivated the approach introduced in [14], which is based on an explicit construction of the operator  $\mathbf{T}$  and the system right-hand-side  $\mathbf{a}$ . From (9), we observe that the entries of the operator and right-hand-side involve the coefficients of the boundary-to-boundary maps  $f^{d,\pm}$ . Owing to the linearity of the maps, the coefficients  $a^{d,\pm}$ ,  $b^{d,\pm}$  and  $c^{d,\pm}$  can in principle be determined by solving only two independent subproblems over each subdomain [36]. In [14], a sampling of the subdomain boundary data was proposed to determine its associated maps' coefficients by solving (linear) regression problems. The approach requires more independent local problem solves but it was made resilient to soft faults by the introduction of a suitable error model in the regression problem. The estimated coefficients of the linear maps are subsequently assembled to form the boundary-to-boundary operator  $\mathbf{T}$ . In addition to its resilient character, one important virtue of the approach in [14] is that it requires few communications as the determination of the maps' coefficients is independent for different subdomains (see AppendixB, in particular Eq. (B.4)–(B.6)). Once the operator  $\mathbf{T}$  and right-hand-side  $\mathbf{a}$  have been determined, the system for the unknown boundary values can be solved using standard methods for non-symmetric systems (*e.g.* Krylov, Bi-CGSTAB and GMRES methods). The tridiagonal structure of  $\mathbf{T}$  for one-dimensional problems can also be exploited.

In the present work, we extend the approach of [14] in order to accommodate a stochastic variant of (1). As further discussed below, this extension introduces significant challenges, namely due to the presence of a random diffusivity field. These include the loss of linearity, owing to the product of the stochastic diffusivity field and the stochastic solution, and the potentially high dimensionality of the probability space needed to suitably represent the random inputs and the solution. In the following section, we develop means to address these challenges and for improving the resilience of the solver introduced in [14].

### 3. Resilient stochastic solver

#### 3.1. Stochastic problem

The deterministic problem is now extended to the stochastic case. We assume that the elliptic operator  $\mathcal{L}$  in (1) is parametrized using a vector of  $K$  real-valued independent random variables  $\boldsymbol{\xi}$ ,

$$\boldsymbol{\xi} = [\xi_1 \quad \cdots \quad \xi_K]^\top \in \mathbb{R}^K, \quad (10)$$

with known joint probability density  $p_{\boldsymbol{\xi}}$ . This leads to the following stochastic problem

$$\begin{cases} \mathcal{L}(\boldsymbol{\xi})u = g, & \text{in } \Omega = (0, 1) \\ u(x = 0) = U_0, \\ u(x = 1) = U_1, \end{cases} \quad (11)$$

whose solution  $u(x, \boldsymbol{\xi})$  is also random as it depends on the parameters. A classical example considered in the paper is the second-order elliptic partial differential equation with random coefficients. In this work (see the numerical illustration in Section 5), a Karhunen-Loève expansion will be used for the functional representation of these random coefficients (specifically the diffusivity field). The development below can be easily extended to the case where the source term  $g$  and boundary conditions are also random. We shall further assume that  $\mathcal{L}$  is almost surely elliptic and that  $u(x, \boldsymbol{\xi})$  has finite second order moments.

The Polynomial Chaos (PC) expansion of  $u(x, \boldsymbol{\xi})$  can be expressed as:

$$u(x, \boldsymbol{\xi}) = \sum_{\alpha} u_{\alpha}(x) \Psi_{\alpha}(\boldsymbol{\xi}), \quad (12)$$

where  $\alpha = (\alpha_1, \dots, \alpha_K) \in \mathbb{N}_0^K$  is a multi-index and  $\Psi_{\alpha}$  is a multivariate polynomial in  $\boldsymbol{\xi}$  consisting of the product of  $K$  univariate orthonormal polynomials, i.e.  $\Psi_{\alpha}(\boldsymbol{\xi}) = \psi_{\alpha_1}^1(\xi_1) \times \dots \times \psi_{\alpha_K}^K(\xi_K)$ , where  $\alpha_i$  refers to the polynomial degree. As a result, the  $\Psi_{\alpha}$  are orthonormal,

$$\langle \Psi_{\alpha}, \Psi_{\beta} \rangle_{\boldsymbol{\xi}} = \mathbb{E}_{\boldsymbol{\xi}} [\Psi_{\alpha} \Psi_{\beta}] = \int_{\mathbb{R}^K} \Psi_{\alpha}(\boldsymbol{\xi}) \Psi_{\beta}(\boldsymbol{\xi}) p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \begin{cases} 1 & \alpha = \beta, \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

and  $|\alpha| = \sum_{i=1}^K \alpha_i$  is the total degree of  $\Psi_{\alpha}$ . The set  $\{\Psi_{\alpha}, \alpha \in \mathbb{N}_0^K\}$  forms a complete orthonormal basis of  $L_2(p_{\boldsymbol{\xi}})$ , the space of square-integrable functionals with respect to the probability measure  $p_{\boldsymbol{\xi}}$ . In practice, the PC expansions are truncated retaining a subset  $\mathcal{A}$  of multi-indices. For instance, fixing the maximum total degree  $q$  of the expansion leads to

$$u(x, \boldsymbol{\xi}) \approx \sum_{\alpha \in \mathcal{A}} u_{\alpha}(x) \Psi_{\alpha}(\boldsymbol{\xi}), \quad \mathcal{A} = \{\alpha \in \mathbb{N}_0^K, |\alpha| \leq q\}. \quad (14)$$

In the following we denote by  $P = \text{card}(\mathcal{A})$  the cardinality of the truncated PC basis.

Different methods have been proposed for the determination of the expansion coefficients  $u_{\alpha}(x)$ , including stochastic Galerkin projection [26, 27, 28, 29, 30], non-intrusive spectral projection [20, 21, 22, 23], and collocation methods [24, 25]. The Galerkin projection is a weak formulation of (11) obtained by projecting the strong form on each polynomial  $\Psi_{\beta}$ ,  $\beta \in \mathcal{A}$ . This results in the following set of coupled linear problems for the expansion coefficients,

$$\forall \beta \in \mathcal{A}, \quad \begin{cases} \sum_{\alpha \in \mathcal{A}} \langle \mathcal{L}(\boldsymbol{\xi}) \Psi_{\alpha}, \Psi_{\beta} \rangle_{\boldsymbol{\xi}} u_{\alpha} = \langle g, \Psi_{\beta} \rangle_{\boldsymbol{\xi}} & \text{in } \Omega = (0, 1) \\ u_{\beta}(x=0) = \langle U_0, \Psi_{\beta} \rangle_{\boldsymbol{\xi}}, \\ u_{\beta}(x=1) = \langle U_1, \Psi_{\beta} \rangle_{\boldsymbol{\xi}}. \end{cases} \quad (15)$$

### 3.2. Stochastic affine maps

With a view to solving the stochastic problem (11) by means of a domain decomposition approach, we now extend the boundary-to-boundary maps associated with the subdomains, reusing the notations of section 2. Unlike the deterministic case, the unknown boundary data for  $\Omega_d$  that will satisfy the compatibility conditions are now random. This suggests extending the deterministic local problems (4) to stochastic ones accounting for the randomness of the elliptic operator  $\mathcal{L}(\boldsymbol{\xi})$ , and using random boundary data  $u^{d,\pm}(\boldsymbol{\xi})$ . For given stochastic boundary data, the boundary-to-boundary map values,  $f^{d,\pm}(u^{d,-}(\boldsymbol{\xi}), u^{d,+}(\boldsymbol{\xi}))$ , may be computed by solving the corresponding local stochastic problem. However, we can again exploit the linearity of the problem to recast the stochastic boundary-to-boundary relation as an affine mapping according to:

$$f^{d,\pm} : L_2(p_{\boldsymbol{\xi}})^2 \rightarrow L_2(p_{\boldsymbol{\xi}}), \quad f^{d,\pm}(u^{d,-}(\boldsymbol{\xi}), u^{d,+}(\boldsymbol{\xi})) = a^{d,\pm}(\boldsymbol{\xi}) + b^{d,\pm}(\boldsymbol{\xi}) u^{d,-}(\boldsymbol{\xi}) + c^{d,\pm}(\boldsymbol{\xi}) u^{d,+}(\boldsymbol{\xi}). \quad (16)$$

To stress the stochastic nature of the coefficients appearing in the affine maps, we shall write  $f^{d,\pm}(\cdot, \cdot) = f^{d,\pm}(\cdot, \cdot, \boldsymbol{\xi})$ . Indeed, even for deterministic subdomain boundary conditions, the maps are generally stochastic due to the randomness of the operator. The objective is then to determine

the random maps' coefficients  $a^{d,\pm}$ ,  $b^{d,\pm}$  and  $c^{d,\pm}$ . From the superposition principle and linearity arguments (see AppendixB), it can be further seen that, as in the deterministic case,  $c^{d,\pm} = 1 - b^{d,\pm}$ , reducing theoretically the determination of the maps coefficients to the solution of only two subproblems per subdomain. The generic form of the subproblems is

$$\begin{cases} \mathcal{L}(\boldsymbol{\xi})v^d = g, & \text{in } \Omega_d = (X_d^-, X_d^+) \\ v^d(X_d^-) = u^{d,-}, \\ v^d(X_d^+) = u^{d,+}, \end{cases} \quad (17)$$

where the boundary conditions can be selected as deterministic (*e.g.* the coefficients  $a^{d,\pm}(\boldsymbol{\xi})$  coincide with the solution at the inner boundary points  $X_{d-1}^+$  and  $X_{d+1}^-$  for homogeneous boundary conditions  $u^{d,\pm} = 0$ ). One could for instance rely on a PC expansion of the local solutions  $v^d(x, \boldsymbol{\xi})$  and a stochastic Galerkin projection of the local subproblem to compute the PC expansions of the coefficients. However, soft faults could considerably affect the solution of the Galerkin subproblems, with corrupted PC expansions for the maps coefficients as a result, and a resilient approach is again needed.

### 3.3. Sampling approach for the PC approximation of the maps

In this subsection, we focus on a sampling strategy for the resilient approximation of the stochastic maps  $f^{d,\pm}$  of a subdomain  $\Omega_d$ . The approach is the same for both the left and the right map,  $f^{d,-}$  and  $f^{d,+}$ , and for all subdomains. Therefore, we shall only describe the case of the left map, and drop superscripts to alleviate notations.

As discussed above, the stochastic map can be generically expressed as

$$f(u^-, u^+, \boldsymbol{\xi}) = a(\boldsymbol{\xi}) + b(\boldsymbol{\xi})u^- + c(\boldsymbol{\xi})u^+, \quad (18)$$

where  $u^-$  and  $u^+$  are the boundary data. Using  $c(\boldsymbol{\xi}) = 1 - b(\boldsymbol{\xi})$ , the coefficient  $c(\boldsymbol{\xi})$  can be eliminated to obtain

$$f(u^-, u^+, \boldsymbol{\xi}) - u^+ = a(\boldsymbol{\xi}) + b(\boldsymbol{\xi})(u^- - u^+). \quad (19)$$

The coefficients  $a(\boldsymbol{\xi})$  and  $b(\boldsymbol{\xi})$  are approximated using PC expansions,

$$a(\boldsymbol{\xi}) \approx \sum_{\alpha \in \mathcal{A}} a_\alpha \Psi_\alpha(\boldsymbol{\xi}), \quad b(\boldsymbol{\xi}) \approx \sum_{\alpha \in \mathcal{A}} b_\alpha \Psi_\alpha(\boldsymbol{\xi}). \quad (20)$$

The key idea to achieve resilience is to estimate the PC coefficients of  $a$  and  $b$  (and subsequently deduce  $c$ ) through a robust regression procedure. To perform the regression, we rely on a joint sampling of the boundary data  $u^-$ ,  $u^+$  and random parameters  $\boldsymbol{\xi}$ , generating a sample set of  $n$  independent triplets  $\{(u_i^-, u_i^+, \boldsymbol{\xi}_i), i = 1, \dots, n\}$ . The samples  $\boldsymbol{\xi}_i$  are drawn from the distribution  $p_{\boldsymbol{\xi}}$ , while the boundary values can be sampled freely over ranges for which the solution exists, *e.g.* uniformly from  $[0, 1]$ . With each triplet  $(u_i^-, u_i^+, \boldsymbol{\xi}_i)$  we associate the *deterministic* subproblem (4), with deterministic operator  $\mathcal{L} = \mathcal{L}(\boldsymbol{\xi}_i)$  and  $u_i^-$  (resp.  $u_i^+$ ) as left (resp. right) boundary value, and we denote by  $f_i$  the associated map value (solution at the inner point of interest). The regression aims at finding the PC expansion coefficients of  $a$  and  $b$  that minimize, in some sense, the sample set distance between the observed and approximated maps. For each triplet, the difference between the observed map value and its approximation is given by

$$r_i \equiv f_i - u_i^+ - \sum_{\alpha \in \mathcal{A}} a_\alpha \Psi_\alpha(\boldsymbol{\xi}_i) - (u_i^- - u_i^+) \sum_{\alpha \in \mathcal{A}} b_\alpha \Psi_\alpha(\boldsymbol{\xi}_i). \quad (21)$$

Denoting by  $\mathbf{y} \in \mathbb{R}^n$  the vector with components  $f_i - u_i^+$ ,  $\mathbf{a}$  (resp.  $\mathbf{b}$ ) the vector of  $\mathbb{R}^P$  containing the PC coefficients of  $a$  (resp.  $b$ ), the distances at the sample points can be expressed in vector form according to

$$\mathbf{r} = \mathbf{y} - \mathbf{X}_a \mathbf{a} - \mathbf{X}_b \mathbf{b}, \quad (22)$$

where the matrices  $\mathbf{X}_a$  and  $\mathbf{X}_b$  have elements

$$[\mathbf{X}_a]_{i,\alpha} = \Psi_\alpha(\xi_i), \quad [\mathbf{X}_b]_{i,\alpha} = (u_i^- - u_i^+) \Psi_\alpha(\xi_i), \quad i = 1, \dots, n \text{ and } \alpha \in \mathcal{A}. \quad (23)$$

Gathering the PC coefficients of  $a$  and  $b$  in a single vector  $\boldsymbol{\beta} \in \mathbb{R}^{2P}$ , we end up with

$$\mathbf{r} = \mathbf{y} - \mathbf{X} \boldsymbol{\beta}, \quad \mathbf{X} = [\mathbf{X}_a, \mathbf{X}_b], \quad (24)$$

where  $\mathbf{X}$  is called the design matrix. The question of finding  $\boldsymbol{\beta}$  that provides the best approximation of the exact map's coefficients is central to the present work. In particular, the direct minimization of  $\mathbf{r}$ , *e.g.* its  $\ell_2$ -norm is not an option here, because of potential soft faults. Specifically, some samples  $f_i$  are expected to be corrupted so the fit must be performed on noisy data. This calls for an appropriate regression method, which will be developed in Section 4 below.

*Remark.* The sampling approach described above considers a generic subdomain, while for the first and last ones a boundary value is actually known ( $U_0$  and  $U_1$  respectively). Different treatments of known boundary values can be envisioned. In particular, one could adapt the definition of the maps for the first and last subdomains, retaining a dependence and a sampling of the only remaining unknown boundary values. In that case, the global boundary conditions are implicitly accounted for and are not apparent in the compatibility conditions to be enforced. In the present work, we choose to keep the same treatment and map definition for all the subdomains, and deal with the known global boundary conditions when assembling the stochastic linear system expressing the compatibility conditions. The method then seamlessly applies to the case when the global boundary conditions are uncertain.

### 3.4. Condensed problem for the stochastic boundary values

Once the PC coefficients in Eq. (22) are estimated through regression on each subdomain and for each inner point of interest (see Sections 3.3 and 4), compatibility conditions can be derived using those coefficients. Specifically, the unknown boundary values must satisfy the following set of  $2(N-1)$  stochastic compatibility equations:

$$\begin{cases} f^{d,-}(u^{d,-}(\boldsymbol{\xi}), u^{d,+}(\boldsymbol{\xi}), \boldsymbol{\xi}) = u^{d-1,+}(\boldsymbol{\xi}), & d = 2, \dots, N, \\ f^{d,+}(u^{d,-}(\boldsymbol{\xi}), u^{d,+}(\boldsymbol{\xi}), \boldsymbol{\xi}) = u^{d+1,-}(\boldsymbol{\xi}), & d = 1, \dots, N-1, \end{cases} \quad (25)$$

where  $u^{1,-}(\boldsymbol{\xi}) = U_0$ ,  $u^{N,+}(\boldsymbol{\xi}) = U_1$ , and the maps are given by

$$f^{d,\pm}(u^{d,-}(\boldsymbol{\xi}), u^{d,+}(\boldsymbol{\xi}), \boldsymbol{\xi}) = a^{d,\pm}(\boldsymbol{\xi}) + b^{d,\pm}(\boldsymbol{\xi}) u^{d,-}(\boldsymbol{\xi}) + c^{d,\pm}(\boldsymbol{\xi}) u^{d,+}(\boldsymbol{\xi}). \quad (26)$$

To solve (25), we replace the coefficients and boundary data involved with their PC expansions,

$$a^{d,\pm}(\boldsymbol{\xi}) \approx \sum_{\alpha \in \mathcal{A}} a_\alpha^{d,\pm} \Psi_\alpha(\boldsymbol{\xi}), \quad b^{d,\pm}(\boldsymbol{\xi}) \approx \sum_{\alpha \in \mathcal{A}} b_\alpha^{d,\pm} \Psi_\alpha(\boldsymbol{\xi}),$$



---

**Algorithm 1:** Schematic steps of the resilient solver.

---

```
Partition the domain  $\Omega$  into subdomains ;
// Parallel loop
foreach subdomain  $\Omega_d$  do
    Randomly sample the boundary values  $(u^-, u^+)$  and the random vector  $\xi$  ;
    Solve the corresponding local PDE for each sample  $(u_i^-, u_i^+, \xi_i)$  ;
    Collect the solutions  $v_i^d$  ;
    getMapCoefficients  $(u_i^-, u_i^+, \xi_i, v_i^d)$ ;
end foreach
Assemble and solve the condensed system ; /* see equation (28) */

function getMapCoefficients  $(u_i^-, u_i^+, \xi_i, v_i^d)$ 
    Collect the solutions at the inner points,  $v_i^d(X_{d-1}^+)$  and  $v_i^d(X_{d+1}^-)$  into  $f_i^\pm$  ;
    Assemble the (left and right) regression problems ; /* see equations (23) and (24) */
    Solve the robust regression problems using IRT ; /* see section 4 and algorithm 2
    */
    return estimators of the maps' coefficients ;
end function
```

---

scalability. In that sense, the proposed strategy dramatically differs from that of sampling, *e.g.* wrapping a Monte-Carlo, or other non-intrusive sampler around the deterministic resilient algorithm presented in [14]. Such an alternative would require the assembly and solve of the condensed system for every sample. Consequently, this would result in a large amount of global communication, causing synchronization and data transfer latency that would degrade the parallel efficiency of the deterministic algorithm.

At this point, it is relevant to mention that the sampling strategy provides resilience at additional cost. As a matter of fact, more samples than what would be necessary in a fault-free environment need to be drawn, resulting in an increased number of local PDE solves. This additional cost is nonetheless affordable in the context of exascale computing, where it is considered that CPU time is cheap, as opposed to communication time. Indeed, the underlying domain decomposition can be made such that the local problems to be solved on the subdomains are orders of magnitude smaller than the global problem. Another source of overhead resides in the inference of the local maps, which is currently achieved through regression, as detailed further in section 4. The size of each regression problem is independent from the spatial discretization, but depends on the cardinality  $P$  of the PC basis, which suffers from the curse of dimensionality. We recall that the primary focus of the present work is the resilient aspect of the proposed approach, and further investigation of the regression overhead is not pursued here. Dimensionality reduction strategies, that would yield smaller regression problems, are currently being developed and will be reported on elsewhere.

Lastly, it should be stressed that some parts of the algorithm, in particular the regression stage and the condensed system assembly and solve, are not resilient, and thus need to be performed in a guaranteed fault-free environment. Provided that these problems are small enough, various techniques can be readily used to increase the reliability of these selected components.

#### 4. Robust regression

Following the sampling approach described in section 3.3, the determination of the PC coefficients of the map amounts to minimizing the residuals (see Eq. (24))

$$\mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}, \quad (30)$$

where the vector  $\mathbf{y}$  contains  $n$  observations and the vector  $\boldsymbol{\beta}$  contains the  $m$  unknown PC coefficients (recall here that  $m = 2P$ ). Each column of  $\mathbf{X}$  represents a predictor (or regressor) and contains  $n$  samples of this predictor (see Eq. (23)). Defining a regression problem thus amounts to defining the objective function  $J(\boldsymbol{\beta})$  of a minimization problem, namely

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^m} J(\boldsymbol{\beta}), \quad (31)$$

where  $\hat{\boldsymbol{\beta}}$  is called the estimator of  $\boldsymbol{\beta}$  for this particular minimization problem.

##### 4.1. Objective function

To define a suitable regression problem, it is first noted that the data or responses  $\mathbf{y}$  may be corrupted by bit-flips. In the context of regression, these corrupted values should be regarded as outliers. For this reason, any fitting based on least-squares (LS) type objective functions should be avoided, as it is known that the corresponding techniques are not robust to outliers (see, *e.g.*, [14] for an evidence in the context of bit-flips). Instead, we suggest to use a *least absolute deviations* (LAD) approach, which amounts to minimizing the  $L^1$  norm of the residual  $\mathbf{r}$ . LAD techniques have been used for a long time, even before LS techniques (see, *e.g.*, [37]), and are known to be robust to outliers.

Because the cardinality  $P$  of the polynomial basis, and so the number of coefficients in  $\boldsymbol{\beta}$ , increases dramatically with the number of stochastic dimensions  $K$  and the polynomial degree  $q$ , a stabilization of the LAD approach will also be needed. By stability, we mean here that the addition of new predictors (*i.e.* increasing the PC basis size) does not deteriorate the quality of the estimated map. The main source of instability is overfitting, which occurs when the number of samples is too small as compared to the complexity (*i.e.* the size  $m$ ) of the model sought. Overfitting may be avoided by increasing the number,  $n$ , of observations. However, this could result in a prohibitively large number of observations when considering large PC bases. Instead, we would like to use a reasonable and constant ratio  $\rho = n/m$  of observations (samples)  $n$  as compared to the number of unknown PC coefficients  $m$  defining the maps. To do so, we introduce regularization, appending the objective function with a penalty on the norm of the unknown coefficients vector  $\boldsymbol{\beta}$ . Specifically, for  $\lambda > 0$ , we consider objective functions of the form

$$J(\boldsymbol{\beta}) = \|\mathbf{r}\|_1 + \lambda \|\boldsymbol{\beta}\|_\gamma^\gamma = \sum_{i=1}^n |r_i| + \lambda \sum_{j=1}^m |\beta_j|^\gamma, \quad 1 \leq \gamma \leq 2. \quad (32)$$

The first term of the objective function corresponds to the (unpenalized) LAD problem, which ensures resilience to bit-flips. For  $\gamma = 2$ , the penalty term corresponds to the ridge penalty which is commonly used for the regularization of ill-conditioned regression problems. For  $\gamma = 1$ , the penalty term corresponds to a lasso penalty [38], resulting in a LAD-lasso regression problem [34]. The lasso penalty is known to promote the sparsity of  $\hat{\boldsymbol{\beta}}$ . When  $1 < \gamma < 2$ , the penalty term is a

compromise between lasso and ridge, similar to the elastic net penalty [35] which consists of a linear combination of both penalty terms. Because we are primarily concerned about overfitting,  $\gamma$  should be kept close to 1 in order to benefit from the lasso properties. On the other hand, by choosing  $\gamma$  slightly greater than 1, say  $\gamma = 1.3$ , we expect to get an effect similar to the elastic net, promoting both the grouping of highly correlated variables (as in ridge) and the sparsity of the solution (as in lasso).

*Remark.* It should be noted that stability analyses, carried out for unregularized least squares problems, indicate that choosing  $n \propto m^2$  is necessary to ensure stability [39, 40]. Although they lie outside the scope of these studies, unregularized LAD problems could be expected to have a similar behavior, especially when solved by means of an iteratively reweighted least squares algorithm (see next section). In fact, in our experiments, we observed that very large numbers of observations needed to be considered to ensure stability of unregularized LAD problems using large PC bases, consistent with the above mentioned LS results, which led us to introduce a sparsity-promoting penalty term.

#### 4.2. Iteratively reweighted least squares algorithm

For the minimization of the objective function (32) we rely on an *iteratively reweighted least squares* (IRLS) technique [41]. IRLS has been used independently to solve unpenalized LAD problems [42, 43, 44] and classical lasso problems [45, 46]. We propose here a natural extension of IRLS to solve regularized LAD problems. Consistently with the central idea of IRLS, let us define weight vectors  $\mathbf{w}^r$  and  $\mathbf{w}^\beta$  as follows:

$$w_i^r = \frac{1}{\max(\varepsilon, |r_i|)}, \quad \forall i = 1, \dots, n, \quad w_j^\beta = \frac{1}{\max(\varepsilon, |\beta_j|^{2-\gamma})}, \quad \forall j = 1, \dots, m, \quad 0 < \varepsilon \ll 1, \quad \gamma \in [1, 2], \quad (33)$$

where the role of  $\varepsilon$  is to prevent numerical overflows. Then, we consider the minimization of

$$J_\varepsilon(\boldsymbol{\beta}) = \sum_{i=1}^n w_i^r r_i^2 + \lambda \sum_{j=1}^m w_j^\beta \beta_j^2 \approx J(\boldsymbol{\beta}). \quad (34)$$

Note that for  $\gamma = 2$  and  $\varepsilon \ll 1$ , all entries of  $\mathbf{w}^\beta$  are equal to one, irrespective of  $\boldsymbol{\beta}$ . Equation (34) corresponds to the objective function of a weighted least squares problem with a diagonal Tikhonov (or weighted ridge) regularization, whose minimizer  $\hat{\boldsymbol{\beta}}$  is given by:

$$\hat{\boldsymbol{\beta}} = \mathbf{A}^{-1} \mathbf{X}^\top \mathbf{W}^r \mathbf{y}, \quad \text{with } \mathbf{A} = \mathbf{X}^\top \mathbf{W}^r \mathbf{X} + \lambda \mathbf{W}^\beta, \quad (35)$$

where  $\mathbf{W}^r = \text{diag}(\mathbf{w}^r)$  and  $\mathbf{W}^\beta = \text{diag}(\mathbf{w}^\beta)$  are diagonal weighting matrices. Equation (35) is usually referred to as the normal equation. Since the weight vectors depend on the solution  $\hat{\boldsymbol{\beta}}$ , the normal equation (35) is in fact nonlinear. An iterative strategy is employed to compute its solution. Starting from initial weights (*e.g.* unitary, or based on the solution at another  $\lambda$ ),  $\hat{\boldsymbol{\beta}}$  is computed solving the normal equation (35). With the new estimate of  $\hat{\boldsymbol{\beta}}$ , the weights are updated using Eq. (33), and the solution recomputed with the updated weights; this sequence of normal solves and weights updates is repeated until convergence. Clearly, the computationally intensive part of this iterative strategy is the solution of Eq.(35) given the current estimate of the weights. This step

can be conveniently recast as an ordinary least squares (OLS) problem such that dedicated solvers can be reused (*e.g.* LSQR [47, 48]). To this end, it suffices to notice that

$$J_\varepsilon(\boldsymbol{\beta}) = (\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta})^\top \tilde{\mathbf{W}}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta}), \quad (36)$$

with

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda} \mathbf{I}_m \end{bmatrix}, \quad \tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{w}^r \\ \mathbf{w}^\beta \end{bmatrix}, \quad \tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_m \end{bmatrix}, \quad (37)$$

and where  $\tilde{\mathbf{W}}$  is a diagonal weighting matrix whose diagonal is composed of the elements of  $\tilde{\mathbf{w}}$ . This defines a weighted least squares (WLS) problem, which can be recast into an OLS problem as follows:

$$J_\varepsilon(\boldsymbol{\beta}) = \left\| \tilde{\mathbf{y}}^* - \tilde{\mathbf{X}}^* \boldsymbol{\beta} \right\|_2^2, \quad (38)$$

with  $\tilde{\mathbf{y}}^* = \tilde{\mathbf{W}}^{1/2} \tilde{\mathbf{y}}$  and  $\tilde{\mathbf{X}}^* = \tilde{\mathbf{W}}^{1/2} \tilde{\mathbf{X}}$ . Algorithm 2 describes the iteratively reweighted Tikhonov (IRT) algorithm to solve (32) using the normal equation.

---

**Algorithm 2:** Iteratively reweighted Tikhonov (IRT) algorithm.

---

**Data:** Design matrix  $\mathbf{X}$  of size  $n$ -by- $m$ , response vector  $\mathbf{y}$  of size  $n$ .

**Input:** Initial diagonal weight matrices  $\mathbf{W}^r$  and  $\mathbf{W}^\beta$ , *e.g.*  $\mathbf{W}^r = \mathbf{I}_n$  and  $\mathbf{W}^\beta = \mathbf{I}_m$ , tuning parameter  $\lambda$  and norm  $\gamma$  for the regularization.

**Output:** Estimator vector  $\hat{\boldsymbol{\beta}}$  of size  $m$ .

**while** *convergence criterion not met* **do**

$\hat{\boldsymbol{\beta}} \leftarrow \arg \min_{\boldsymbol{\beta}} J_\varepsilon(\boldsymbol{\beta});$  /\* Solve the regression problem with current weights \*/

$\hat{\mathbf{r}} \leftarrow \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}};$  /\* Compute the current residual \*/

// Update weights, see eq. (33)

**for**  $i = 1$  **to**  $n$  **do**

$w_i^r \leftarrow 1 / \max(\varepsilon, |r_i|);$

**end for**

**for**  $j = 1$  **to**  $m$  **do**

$w_j^\beta \leftarrow 1 / \max(\varepsilon, |\beta_j|^{2-\gamma});$

**end for**

$\mathbf{W}^r \leftarrow \text{diag}(\mathbf{w}^r);$   $\mathbf{W}^\beta \leftarrow \text{diag}(\mathbf{w}^\beta);$  /\* Update  $\mathbf{W}^r$  and  $\mathbf{W}^\beta$  accordingly \*/

**end while**

---

### 4.3. Selection of the regularization parameter

The regularization parameter  $\lambda$  has to be selected to prevent overfitting while allowing for a model  $\boldsymbol{\beta}$  that adequately represents the observations. The selection of  $\lambda$  typically involves a statistical validation procedure, where different values for  $\lambda$  are tested to retain the one yielding the lowest prediction error. Since the prediction error is unknown, it has to be estimated from the set of observations. One of the most natural estimates for the prediction error is obtained by considering a separate validation set of observations which is compared to the predictions associated with  $\hat{\boldsymbol{\beta}}$ . A more elaborate idea of cross-validation, the so-called  $k$ -fold cross-validation [49, 50], consists in splitting the observation set into  $k$  subsets. The regression problem for  $\boldsymbol{\beta}$  is then performed

$k$  times, each time choosing one of the subsets to be the validation set and using the remaining observations as the training set. The  $k$  resulting estimates of the prediction error can be combined (*e.g.* averaged) to estimate the overall prediction error. The limiting case where  $k$  is equal to the number of observations  $n$  corresponds to the so-called *leave-one-out* procedure. The leave-one-out approach is widely used because it allows for analytic expressions through the introduction of rank-one updates of the regression problem. This is further discussed in the following.

When the solution  $\hat{\beta}$  has been computed using the IRT algorithm, the prediction vector  $\hat{\mathbf{y}}$  is given by:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^\top\mathbf{W}^r\mathbf{y} = \mathbf{H}\mathbf{y}, \quad (39)$$

where  $\mathbf{A}$  is given in Eq. (35) and  $\mathbf{H} \equiv \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^\top\mathbf{W}^r$  is often called the ‘‘hat’’ matrix. The prediction residual  $\hat{\mathbf{r}}$  is defined as the difference between  $\mathbf{y}$  and  $\hat{\mathbf{y}}$ :

$$\hat{\mathbf{r}} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\hat{\beta} = \mathbf{y} - \mathbf{H}\mathbf{y} = \mathbf{P}\mathbf{y}, \quad (40)$$

where  $\mathbf{P} = \mathbf{I} - \mathbf{H}$  is the projection matrix. We denote by  $\hat{\beta}_{(-i)}$  the estimator for the regression problem where the  $i$ -th observation is left out, that is using only the  $(n-1)$  remaining observations. Using consistent notations, we have

$$\hat{\beta}_{(-i)} = \mathbf{A}_{(-i)}^{-1}\mathbf{X}_{(-i)}^\top\mathbf{W}_{(-i)}^r\mathbf{y}_{(-i)}, \quad \text{with } \mathbf{A}_{(-i)} = \mathbf{X}_{(-i)}^\top\mathbf{W}_{(-i)}^r\mathbf{X}_{(-i)} + \mathbf{W}^\beta. \quad (41)$$

Assuming that dropping the  $i$ -th observation leaves the LAD and regularization weights unchanged (frozen state), one can derive the following equalities:

$$\mathbf{X}_{(-i)}^\top\mathbf{W}_{(-i)}^r\mathbf{X}_{(-i)} = \mathbf{X}^\top\mathbf{W}^r\mathbf{X} - w_i^r\mathbf{x}_i\mathbf{x}_i^\top, \quad \mathbf{X}_{(-i)}^\top\mathbf{W}_{(-i)}^r\mathbf{y}_{(-i)} = \mathbf{X}^\top\mathbf{W}^r\mathbf{y} - w_i^ry_i\mathbf{x}_i, \quad (42)$$

where  $\mathbf{x}_i = \mathbf{X}_i^\top$ . From the Sherman-Morrison identity for the rank-one update of  $\mathbf{A}_{(-i)}^{-1}$ , we have

$$\mathbf{A}_{(-i)}^{-1} = \mathbf{A}^{-1} + w_i^r\frac{\mathbf{A}^{-1}\mathbf{x}_i\mathbf{x}_i^\top\mathbf{A}^{-1}}{1 - h_i} \quad \text{and } \mathbf{A}_{(-i)}^{-1}\mathbf{x}_i = \frac{\mathbf{A}^{-1}\mathbf{x}_i}{1 - h_i}, \quad (43)$$

where  $h_i = \mathbf{H}_{ii} = w_i^r\mathbf{x}_i^\top\mathbf{A}^{-1}\mathbf{x}_i$  is the  $i$ -th diagonal element of  $\mathbf{H}$ . Combining (42) and (43), the prediction of the  $i$ -th observation from the estimator  $\hat{\beta}_{(-i)}$  is

$$\left[\hat{\mathbf{y}}_{(-i)}\right]_i = \mathbf{x}_i^\top\hat{\beta}_{(-i)} = \mathbf{x}_i^\top\mathbf{A}_{(-i)}^{-1}\mathbf{X}_{(-i)}^\top\mathbf{W}_{(-i)}^r\mathbf{y}_{(-i)} = \frac{\hat{y}_i - y_iw_i^r\mathbf{x}_i^\top\mathbf{A}^{-1}\mathbf{x}_i}{1 - h_i} = \frac{\hat{y}_i - y_ih_i}{1 - h_i}, \quad (44)$$

with corresponding leave-one-out residual

$$\left[\hat{\mathbf{r}}_{(-i)}\right]_i = y_i - \left[\hat{\mathbf{y}}_{(-i)}\right]_i = y_i - \frac{\hat{y}_i - y_ih_i}{1 - h_i} = \frac{y_i - \hat{y}_i}{1 - h_i} = \frac{\hat{r}_i}{1 - h_i}. \quad (45)$$

We stress that this expression of the leave-one-out residual is an approximation, since solving the regression problem for the  $(n-1)$  observations would actually affect the weights of the regression problem which were considered fixed in the derivation above. This approximation is valid only for situations where the estimator  $\hat{\beta}_{(-i)}$  (and so the predictor  $\hat{\mathbf{y}}_{(-i)}$ ) remains similar for all  $i$  or, in other words, if the regression solution is stable when leaving out one observation. In fact, this is the case for large  $\lambda$ . For small  $\lambda$ , the equalities in (42) and (43), as well as the update of  $\mathbf{A}^{-1}$ , are not correct anymore but remain suitable to detect the emergence of overfitting. Specifically, one selects

the value of  $\lambda$  leading to the minimal root-mean-square (RMS) value  $\Sigma_{\text{LOO}}$  of the leave-one-out residual, defined as

$$\Sigma_{\text{LOO}}^2 = \frac{1}{n} \sum_{i=1}^n [\hat{r}_{(-i)}]_i^2 = \frac{1}{n} \sum_{i=1}^n \frac{\hat{r}_i^2}{(1-h_i)^2} = \frac{1}{n} \mathbf{y}^\top \mathbf{P}(\text{diag } \mathbf{P})^{-2} \mathbf{P} \mathbf{y}. \quad (46)$$

However, it appears that even if the regression is robust to outliers, owing to the LAD part of the objective function, the RMS value of the leave-one-out residual is not resilient to the presence of observations corrupted by bit-flips. This is due to the fact that the regression is designed to disregard outliers, so a corrupted observation  $i$  is associated with a large residual value  $r_i$  when the bit-flips induce a large error on  $y_i$ . The estimate  $\Sigma_{\text{LOO}}^2$  is therefore essentially dominated by large bit-flips errors at the corrupted observations, making it difficult to measure small effects related to  $\lambda$  and the appearance of overfitting. This issue is also present for cross-validation estimates of the prediction error which are plagued by large bit-flips errors, since cross-validation observations can be corrupted as well. To overcome this issue, one could think of comparing the predictors  $\hat{\mathbf{y}}$  and  $\hat{\mathbf{y}}_{(-i)}$ , and select the value of  $\lambda$  such that the RMS value of  $\hat{y}_i - [\hat{\mathbf{y}}_{(-i)}]_i$  is minimal. However, this approach tends to favor high values of  $\lambda$  such that the model is over-stable and insensitive to removal of observations, but has poor predictive capabilities. A closer inspection of the LOO residuals  $[\hat{r}_{(-i)}]_i$  reveals that it is in fact well behaved for observations that are not corrupted by bit-flips. This observation leads us to consider the median value instead of the RMS residual value to design a criterion for selecting  $\lambda$ . Specifically, we define

$$m_{\text{LOO}} = \text{med} \{ |[\hat{r}_{(-i)}]_i|, i = 1, \dots, n \}, \quad (47)$$

where the symbol *med* denotes the median value. The main assumption supporting this approach is that fewer than half of the observations are corrupted, so  $m_{\text{LOO}}$  can be understood as an estimate of the predictive error based on the LOO residuals at uncorrupted observations only. The value of  $\lambda$  is then selected as to minimize  $m_{\text{LOO}}$ .

#### 4.4. Numerical examples

To conclude this section and validate the resilient LAD-lasso regression approach proposed above, we consider the approximation of a map for an elliptic test problem consisting of a one-dimensional diffusion equation in a unit domain with uncertain diffusivity field parameterized with  $K$  independent standard Gaussian random variables  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_K)$ . (See Section 5.1 for a complete description of the test problem.) To assess the effectiveness of the resilient LAD-lasso regression, depending on the dimension of the regression problem, two random parameterizations of the diffusivity field are considered. The first one uses  $K = 5$  random variables and variable PC order  $q$ . The second case uses  $q = 3$  and a variable number of random variables  $K$ . The map to be approximated,  $f(u^-, u^+, \boldsymbol{\xi})$ , corresponds to an inner point  $x_m \in (0, 1/2)$  (in this exercise, a single domain is used for the spatial discretization). The regression uses  $n = \rho m$  samples (observations) of the map, with a fixed value  $\rho = 3$  for all the tests of this section. Regarding the sampling, the Dirichlet boundary values of the map,  $u^-$  and  $u^+$ , are sampled uniformly in  $[0, 1]$ , together with  $\boldsymbol{\xi}$  according to its Gaussian distribution. For each sample, the corresponding deterministic elliptic problem is solved with a standard finite-element method to generate an observation  $f_i$  of the map, as discussed in section 3.3. To model soft-errors, the values  $f_i$  are subsequently corrupted as follows. For each  $f_i$ , we decide with a probability  $P_{\text{bf}}$  to alter its 64-bit binary representation (see the IEEE 754 Standard [51]), flipping at random one of its bits (*i.e.* a 0 becomes a 1, and vice

versa). As a result,  $P_{\text{bf}}$  controls the fraction of corrupted observations in the regression problem and can be interpreted as the probability for a PDE solve to be corrupted by a soft-fault. It should be noted that in the rare event of a bit-flip leading to the encoding of an infinite number (**Inf**) or a so-called *not-a-number* (**NaN**), the processing unit that is handling the corresponding sample may crash, potentially causing the entire computation to abort. To overcome the occurrence of such events, and more generally to tackle the issue of so-called hard faults [17], our approach was implemented in a server-client framework and resorted to the *User Level Failure Mitigation* prototype for MPI (ULFM-MPI) [52] to recover from the loss of individual processing units [53, 54]. Such an implementation is however not considered here and we instead focus on the treatment of silent data corruption (SDC). In any case, **Inf** and **NaN** encoded variables are easy to detect, and the corresponding samples may simply be dropped or recomputed. More generally, *a priori* bounds on the local PDE solutions can be used to detect faulty samples that lie outside these bounds, thus improving the overall resilience of the approach [55]. In the present numerical experiments, only the detection of **Inf** and **NaN** is considered. In order to investigate the most unfavorable scenario, instead of dropping or recomputing **Inf** and **NaN** encoded samples, we force the bit-flip to affect another bit, so that we still obtain a corrupted value.

The values of  $P_{\text{bf}}$  considered in this paper range between 0 and 10%. Such values may correspond to high probabilities of soft-faults, as compared to realistic exascale scenarios, but are motivated by the prospective nature of this study for which it is relevant to observe high failure rates to investigate resilience. To assess the error in the PC approximation  $\hat{f}(u^-, u^+, \boldsymbol{\xi})$  of the map, as obtained by solving the regression problem with potentially corrupted observations, we consider the following normalized map error

$$\tilde{\epsilon}_m = \frac{\|f - \hat{f}\|_*}{\|f\|_*}, \quad (48)$$

where the norm is over  $L_2(p_{\boldsymbol{\xi}}) \otimes L_2([0, 1]^2)$ :

$$\|g\|_*^2 = \int_0^1 du \int_0^1 dv \int_{\mathbb{R}^K} d\boldsymbol{\xi} g(u, v, \boldsymbol{\xi}) p_{\boldsymbol{\xi}}(\boldsymbol{\xi}). \quad (49)$$

The map error  $\tilde{\epsilon}_m$  is estimated by means of a Monte Carlo method from a large sample set of 100,000 realizations.

The plots of Fig. 2 report typical map errors  $\tilde{\epsilon}_m$ , at  $x_m = 0.1$ , for  $P_{\text{bf}} = 0$  and  $P_{\text{bf}} = 0.01$  (left and right column respectively), and variable PC basis size, by varying either  $q$  or  $K$  (top and bottom row respectively). For each case, the plots report the median values of the errors  $\tilde{\epsilon}_m$  estimated from 50 random samples sets (independent draws of sampling points and bit-flips); the plots also contrast four different choices of the objective function defining the regression approach: the case of least squares (labelled LS) and LAD without regularization ( $\lambda = 0$  in Eq. (32)) and their regularized versions (labelled lasso and LAD-lasso) with a selection of  $\lambda$  based on the minimization of  $m_{\text{LOO}}$  and using  $\gamma = 1.3$ . The plot in Fig. 2a demonstrates the importance of the regularization, even in the absence of bit-flips ( $P_{\text{bf}} = 0$ ): the map error for the non-regularized regressions (LS and LAD) increases after a certain value of  $q$ , thus illustrating the loss of stability for a number of samples scaling linearly with the PC basis dimension ( $\rho \equiv n/m = 3$ ). On the contrary, regularized regressions yield errors that remain almost constant even for large orders  $q$ . The slight increase in the lasso and LAD-lasso errors around  $q = 9$  is attributed to the procedure for the selection of  $\lambda$ , and corresponds to the price to pay for stability. When bit-flips are introduced, Fig. 2b shows that the least-squares based regressions (LS and lasso) become unstable for  $q \geq 4$ . In contrast, LAD and

LAD-lasso behave as in the case with no bit-flips, the latter remaining stable on the whole range of  $q$  reported. The plot then indicates that if LAD provides resilience, it still needs regularization to remain stable when  $q$  increases. Similarly, Fig. 2c and 2d confirm that the LAD-based regressions provide resilience, while the LS-based regressions are sensitive to bit-flips. However, these plots also indicate that for the polynomial degree considered ( $q = 3$ ), the regressions do not need to be regularized as the number of stochastic dimensions  $K$  (and the PC basis) increases: both LAD and LAD-lasso remain stable for the whole range of  $K$  tested even in the presence of bit-flips (see Fig. 2d).

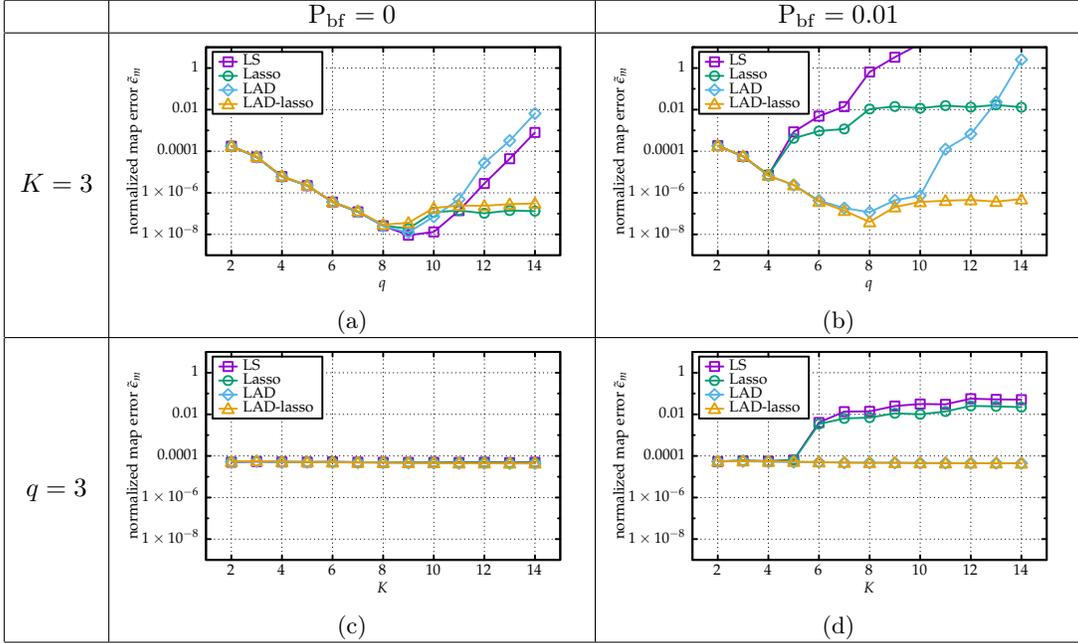


Figure 2: Median over 50 replicas of the normalized map error  $\tilde{\epsilon}_m$  for four regression approaches. Two probabilities of bit-flip  $P_{\text{bf}}$  are shown, (left)  $P_{\text{bf}} = 0$  and (right)  $P_{\text{bf}} = 0.01$ , as well as different PC bases with (top)  $K = 3$  and increasing  $q$ , and (bottom)  $q = 3$  and increasing  $K$ .

The results presented in Fig. 2 demonstrate the advantage of using LAD-lasso regression to ensure both resilience and stability in the map approximation. Obviously, these results depend to some extent on the considered map and the probability of bit-flips. To better appreciate the robustness and resilience of the LAD-lasso regression, we present in Fig. 3 the dependence of the median map error with respect to the location of the map point,  $x_m$ , for the case of  $K = q = 5$  ( $P = 252$ , see Eq. (57)). The map degenerates to  $f(u^-, u^+, \boldsymbol{\xi}) = u^-$  as  $x_m \rightarrow 0$ , and the plot of Fig. 3 indeed exhibits a decaying error with  $1/x_m$ : bit-flips are appropriately treated as outliers and are not confused with dependence on  $\boldsymbol{\xi}$ . In addition, the plot highlights the resilience of LAD-lasso, as the median error is essentially independent of  $P_{\text{bf}}$ .

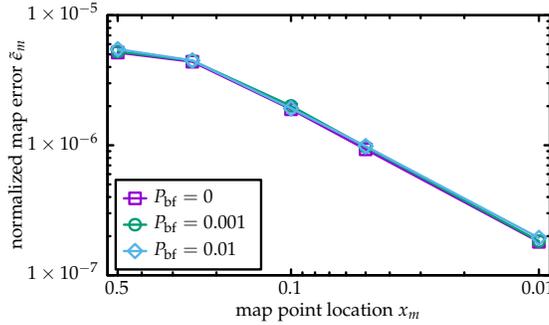


Figure 3: Median over 50 replicas of the normalized map error  $\bar{\epsilon}_m$  as a function of the map point location  $x_m$ , for three values of  $P_{bf}$ , and using the LAD-lasso. The regression is on a PC basis with  $K = 5$ ,  $q = 5$ .

## 5. Results

### 5.1. Test Problems and Preliminaries

We consider steady diffusion equations with a log-normal diffusivity field in order to investigate the resilience to soft fault errors of the proposed method. Specifically, the operator  $\mathcal{L}(\boldsymbol{\xi})$  in Eq. (11) is chosen as

$$\mathcal{L}(\boldsymbol{\xi})u = \frac{\partial}{\partial x} \left[ \kappa(x, \boldsymbol{\xi}) \frac{\partial}{\partial x} u(x, \boldsymbol{\xi}) \right], \quad (50)$$

where  $\kappa$  is a log-normal process:

$$\kappa(x, \boldsymbol{\xi}) = \exp [G(x, \boldsymbol{\xi})]. \quad (51)$$

In the previous equation,  $G$  is a Gaussian process with covariance function  $C$ . For simplicity, we shall assume  $G$  to be centered and stationary with covariance simply defined by

$$C(x, x') = \mathbb{E}[G(x, \cdot)G(x', \cdot)] = C(|x - x'|). \quad (52)$$

The process  $G$  has an infinite Karhunen-Loève (KL) expansion

$$G(x, \boldsymbol{\xi}) = \sum_{k=1}^{\infty} \sqrt{\lambda_k} \phi_k(x) \xi_k, \quad \text{with } \xi_k \sim \mathcal{N}(0, 1) \text{ i.i.d.}, \quad (53)$$

where  $\phi_k$  are the normalized eigenfunctions of  $C$  and  $\lambda_k \geq 0$  are the associated eigenvalues. For the numerical tests, we shall rely on stationary Gaussian processes  $G$ , having squared exponential covariance with correlation length  $L$  and variance  $\sigma_G^2$ :

$$C(x, x') = \sigma_G^2 \exp[-(x - x')^2 / (2L^2)]. \quad (54)$$

We shall consider two pairs of covariance parameters,  $(L = 1, \sigma_G = 0.5)$  and  $(L = 0.1, \sigma_G = 0.05)$ , to contrast the behavior of the method. Further, without loss of generality, we fix  $g = 0$  and the Dirichlet boundary conditions to  $U_0 = 0$  and  $U_1 = 1$ . For convenience, we use the correlation lengths  $L = 1$  and  $L = 0.1$  to identify the two covariance cases considered. As a closing note on the test problem in (50), we mention that the problem is well posed for coefficient  $\kappa$  bounded above and away from zero almost everywhere in  $\Omega$ . This is not the case for the log-normal field. However, the truncated problem for finite KL expansion and finite dimensional PC expansion, as considered below, are well posed. A complete theoretical analysis of the well-posedness of (50) for log-normal coefficients  $\kappa$  can be found in [56].

### 5.1.1. Truncating the Gaussian process

Figure 4a shows the eigenvalues, sorted in decreasing magnitude, of the covariance function (54) for the two correlation lengths,  $L = 1$  and  $L = 0.1$ . It highlights the slower rate of decay in the eigenvalues for the shorter correlation length. The decay in the eigenvalues allows for the truncation of the KL expansion of  $G$  to its, say,  $K$  dominant modes. We shall refer to  $K$  as the stochastic dimension in the following because it fixes the number of random variables in the parametrization. We denote  $G_K$  the truncated version of  $G$ ,  $\mathcal{L}_K$  the operator using  $G_K$  instead of  $G$  in the definition (51) of  $\kappa$ , and  $u_K$  the corresponding solution of Eq. (11). The spatially continuous solutions  $u$  and  $u_K$  are approximated in space with a standard piecewise linear finite-element method, on a uniform mesh fine enough to accommodate the solution features, where the numerical KL decomposition of  $G$  uses a piecewise-constant approximation of  $C$  and its eigenfunctions over the mesh elements. For clarity, we introduce a superscript  $h$  to denote the semi-discrete (in space) solution, *e.g.*  $u_K^h$ , and analyze the errors in semi-discrete solutions defined on same finite-element meshes. Figure 4b shows the decay with the stochastic dimension  $K$  of the normalized error  $u^h - u_K^h$ , in  $L_2$ -norm, resulting from the approximation of  $\mathcal{L}$  by  $\mathcal{L}_K$ . The normalized error norm is given by

$$\epsilon_{\text{KL}}^2(K) = \mathbb{E} \left[ \|u^h - u_K^h\|_{L_2(\Omega)}^2 \right] / \mathbb{E} \left[ \|u^h\|_{L_2(\Omega)}^2 \right]. \quad (55)$$

In practice, the error in (55) is evaluated from a large Monte Carlo (MC) sample set, consisting of 100,000 sample points, so that the MC sampling error is negligible.

As can be appreciated from Fig. 4b, the shorter correlation length implies that a higher stochastic dimension  $K$  is required to achieve a given error level. For  $L = 1$ , the error plateaus for  $K > 10$  because eigenvalues reach machine precision as can be seen in Fig. 4a.

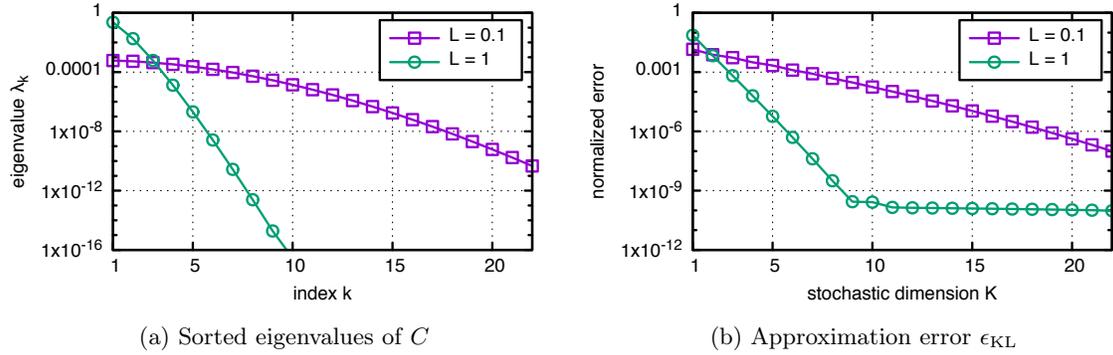


Figure 4: Effect of truncating the KL expansion of  $G$ . (a) sorted eigenvalues  $\lambda_k$  of the covariance function  $C$ ; (b) normalized error norm for different truncation index  $K$ . The solutions  $u^h$  and  $u_K^h$  are computed for each sample using a finite-element method, then the error is obtained from Eq. (55) through MC sampling.

### 5.1.2. PC expansion error

In addition to the spatial discretization and truncation of the Gaussian process with finite  $K$ , a stochastic discretization is introduced using PC expansions. This leads to another source of error. Following notations of Section 3, the fully discretized solution is expressed as

$$u_{K,q}^h(x, \xi) \equiv u_{\mathcal{A}}^h(x, \xi) = \sum_{\alpha \in \mathcal{A}} u_{\alpha}^h(x) \Psi_{\alpha}(\xi) \quad (56)$$

where the PC basis dimension  $P$  (the cardinality of  $\mathcal{A}$ ), the number of stochastic dimensions  $K$ , and the PC degree  $q$  are related by

$$P = \frac{(K+q)!}{K!q!}. \quad (57)$$

As discussed previously, the expansion coefficients  $u_\alpha^h(x)$  in (56) can be classically computed using different approaches. To serve as a reference, the Galerkin projection of the stochastic problem for the truncated operator  $\mathcal{L}_K$  is performed, and we denote  $\bar{u}_{K,q}^h(x, \boldsymbol{\xi})$  the corresponding discrete Galerkin solution using PC expansion of degree  $q$ . To assess the error of the Galerkin solution, we rely on two normalized error measures  $\epsilon_{\text{Gal}}(K, q)$  and  $\epsilon_{\text{PC}}(K, q)$  defined respectively as

$$\epsilon_{\text{Gal}}^2(K, q) = \frac{\mathbb{E} \left[ \|u^h - \bar{u}_{K,q}^h\|_{L_2(\Omega)}^2 \right]}{\mathbb{E} \left[ \|u^h\|_{L_2(\Omega)}^2 \right]}, \quad \text{and} \quad \epsilon_{\text{PC}}^2(K, q) = \frac{\mathbb{E} \left[ \|u_K^h - \bar{u}_{K,q}^h\|_{L_2(\Omega)}^2 \right]}{\mathbb{E} \left[ \|u_K^h\|_{L_2(\Omega)}^2 \right]}. \quad (58)$$

The first error measure  $\epsilon_{\text{Gal}}$  quantifies the total distance to the exact semi-discrete solution, whereas the second error measure  $\epsilon_{\text{PC}}$  quantifies only the effect of PC discretization on the approximation of  $u_K^h$ . Figure 5a depicts the dependence of  $\epsilon_{\text{Gal}}$  and  $\epsilon_{\text{PC}}$  on the polynomial degree  $q$ , for  $L = 1$  and a fixed value  $K = 5$ . We notice that for  $q \leq 4$  the two errors are similar denoting the predominance of the PC discretization error in the global error. However, for  $q > 4$ , the PC discretization error  $\epsilon_{\text{PC}}$  keeps decreasing with  $q$  while  $\epsilon_{\text{Gal}}$  plateaus, indicating that the dominant source of error becomes the truncation of  $G$ . In fact, it is seen that as  $q$  increases,  $\epsilon_{\text{Gal}}$  converges to the corresponding value of  $\epsilon_{\text{KL}}(K = 5)$  as one would have expected (see Fig. 4b). Similarly, Fig. 5b depicts the dependence of  $\epsilon_{\text{Gal}}$  on the stochastic dimension  $K$ , for  $L = 1$  and a fixed polynomial degree  $q = 2$ . We again observe a fast decay of the total error, followed by a plateau. In this case, the plateau arises from the stagnation of the PC approximation error  $\epsilon_{\text{PC}}$ , and a larger polynomial order is required to further reduce  $\epsilon_{\text{Gal}}$ . Based on these error measurements, unless stated otherwise below we shall use  $K = 5$  modes in the approximation of  $G$  when  $L = 1$ , and  $q = 2$  for the polynomial degree of the PC approximation when considering the case  $L = 0.1$ . Indeed, these values lead asymptotically to global errors less than  $10^{-5}$  as  $q$  and  $K$  respectively increase. For the shorter correlation length, the polynomial degree  $q$  is fixed rather than  $K$ , since significantly more modes need to be retained to decently approximate the log-normal process. The smaller variance  $\sigma_G^2$  of the process for  $L = 0.1$  explains the fact that only  $q = 2$  is needed to achieve an asymptotic error similar to that obtained with  $q = 5$  for  $L = 1$ .

### 5.1.3. Validation of the domain decomposition approach

Before considering soft fault effects, we first verify the proposed domain-decomposition approach. Specifically, we verify that the PC approximation of the boundary-to-boundary mappings, and the Galerkin interpretation of the compatibility conditions on the subdomain boundary values, do not lead to significant approximation error in the solution. To this end, in addition to the classical Galerkin solution  $\bar{u}_{K,q}^h$ , we denote  $\bar{\bar{u}}_{K,q}^h$  to be the solution of the domain decomposition (DD) approach. To alleviate notations, we drop the subscripts  $K, q$  relative to the PC discretization. For a decomposition of  $\Omega$  involving  $N$  subdomains, let  $N_\Gamma \equiv 2(N-1)$  and  $X_{1 \leq k \leq N_\Gamma}^\Gamma$  be the number and location of the inner boundary points, and consider the discrete norm

$$\|v\|_\Gamma^2 = \frac{1}{N_\Gamma} \mathbb{E} \left[ \sum_{k=1}^{N_\Gamma} |v(X_k^\Gamma, \boldsymbol{\xi})|^2 \right]. \quad (59)$$

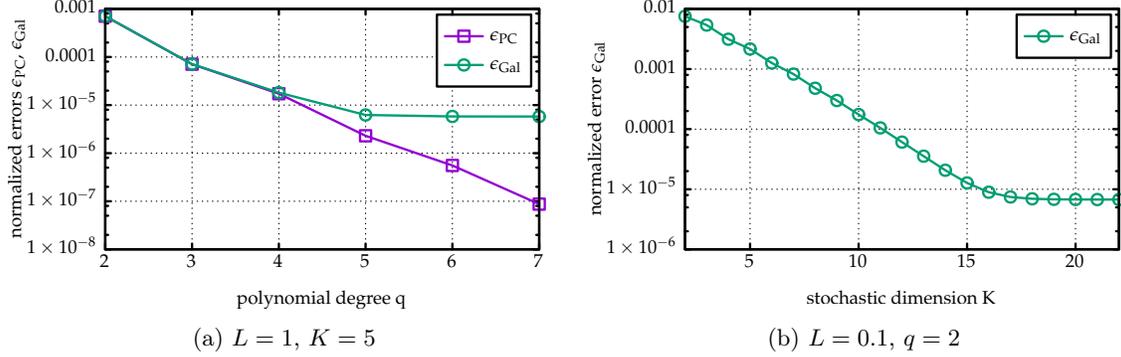


Figure 5: (a) normalized errors  $\epsilon_{Gal}$  and  $\epsilon_{PC}$  as a function of  $q$  for the case  $L = 1$  and using  $K = 5$ . (b) normalized error  $\epsilon_{Gal}$  as a function of the stochastic dimension  $K$  for the case  $L = 0.1$  and a fixed polynomial degree  $q = 2$ .

This norm is used as a measure of the normalized distance between the DD-solution or the classical Galerkin solution and the semi-discrete solution  $u^h$ , that is

$$\epsilon_{\Gamma}(v) \equiv \frac{\|v - u^h\|_{\Gamma}}{\|u^h\|_{\Gamma}}, \quad (60)$$

where  $v = \bar{u}^h$  or  $v = \bar{u}^h$ . Observe that this error measure depends on the PC discretization  $K$  and  $q$  as well as the parameters of the domain decomposition, namely  $N$  and the overlap  $\bar{h}$ . In all the experiments presented here, the overlap is set to be the same at each interface, that is  $\bar{h} \equiv X_d^+ - X_{d+1}^-$ , for  $d = 1, \dots, N - 1$ . Figure 6 reports the dependence of the normalized errors  $\epsilon_{\Gamma}$  for different discretization parameters and the two correlation functions. In these computations, no bit-flips are introduced so we have added the subscript  $\star$  to underline the absence of soft faults. Figure 6a, corresponding to the case  $L = 1$ , shows a decay of  $\epsilon_{\Gamma}$  with  $q$ , for both the Galerkin and the DD solutions, which is consistent with the results reported in Fig. 5a. Furthermore, the reported errors are essentially equal for the two approaches and the two values of  $N$  reported. Figure 6b demonstrates a similar behavior for the case  $L = 0.1$ , with a convergence of Galerkin and DD solutions as  $K$  is increased. A minor variability with  $N$  in  $\epsilon_{\Gamma}$  for the DD solution is also visible. This variability can be attributed to the metric definition and to the (weak) variability of the DD solution with respect to the sample sets involved in the approximation of the maps with finite  $\rho$  (here we used  $\rho = 3$ ). At any rate, the results reported demonstrate the validity and the accuracy of the proposed DD method in the absence of bit-flips. Below, we investigate the resilience of the DD methods when soft faults are introduced.

## 5.2. Analysis of resilience

We now proceed to analyze how the proposed DD solver performs under the presence of soft faults. Recall that the domain decomposition method proceeds from distributed construction of PC approximation for the local boundary-to-boundary maps from samples of  $\xi$  and boundary values  $u^{\pm}$ . As described in Section 4.4, soft faults are modeled by corrupting the computed samples of the map  $f^{d,\pm}$ : a sample  $f_i^{d,\pm}$  is corrupted with a probability  $P_{bf}$ , flipping at random one of the 64 bits in its binary representation. As a result,  $P_{bf}$  is (on average) the fraction of corrupted data used for the resilient regression on a subdomain. In the numerical experiments presented below,

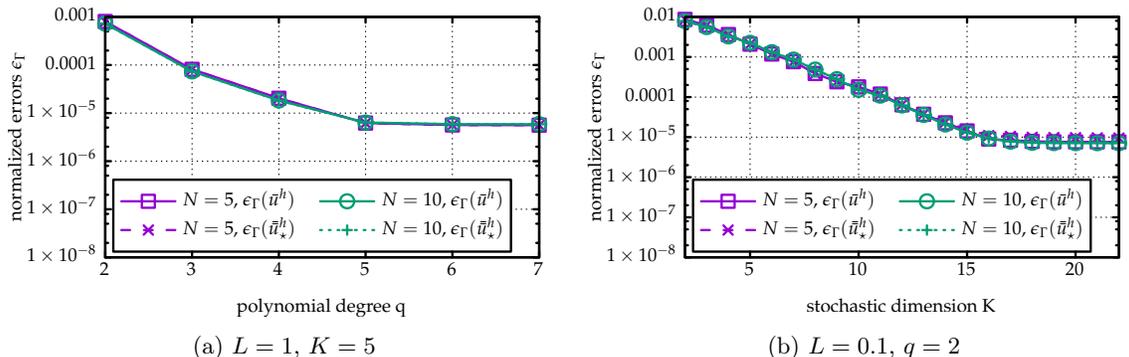


Figure 6: Error  $\epsilon_T$  between Galerkin ( $\bar{u}^h$ ) or fault-free DD ( $\bar{u}_*^h$ ) solutions and  $u^h$ , using two decompositions of  $\Omega$  in  $N = 5$  and  $N = 10$  subdomains. (a)  $L = 1$ ,  $K = 5$ , and variable  $q$ ; (b)  $L = 0.1$ ,  $q = 2$  and different  $K$ . Other parameters are  $\rho = 3$  and  $\bar{h} = 0.05$ .

each subdomain uses an independent random sampling of  $\xi$  and  $u^\pm$  and bit-flips are also drawn independently. To fairly assess the resilience of the DD approach, the solution  $\bar{u}^h$  is compared to the solution  $\bar{u}_*^h$  that would have been obtained for the sample set of  $\xi$  and  $u^\pm$ , but without any bit-flips of the map values, that is for  $P_{\text{bf}} = 0$ . The effect of bit-flips is then directly measured by considering the normalized distance

$$\epsilon = \frac{\|\bar{u}^h - \bar{u}_*^h\|_\Gamma}{\|\bar{u}_*^h\|_\Gamma}. \quad (61)$$

Thus,  $\epsilon$  measures the distance or error between the solutions of the DD method with and without bit-flips. It should be stressed that  $\epsilon$  is a random quantity because the two solutions  $\bar{u}^h$  and  $\bar{u}_*^h$  are constructed using random samples and the data of the resilient regression problem are randomly corrupted. As a consequence, we resort to statistical measures to report the behavior of  $\epsilon$  in our approach. More precisely, we focus on the quantiles of the  $\epsilon$  estimated from 1,000 independent replicas (runs) of the DD approach.

Figure 7 reports the statistics of  $\epsilon$  for  $L = 1$ . In these experiments the domain  $\Omega$  is discretized with 100 finite-elements and partitioned into  $N = 5$  subdomains with an overlap of 5 elements ( $\bar{h} = 0.05 = 5h$ ). The stochastic discretization uses  $K = 5$  and  $q = 5$ , so  $P = 252$ . Figures 7a to 7c show the quantiles of  $\epsilon$  as a function of the sample ratio  $\rho$  and for three bit-flip probabilities  $P_{\text{bf}} = 0.001$ , 0.01 and 0.1. The plots show several quantiles, including the the median value (bold line), as well as  $\epsilon$  for the 1,000 replicas (labelled realizations) to illustrate the dispersion.

Focusing first on the lowest bit-flip probability  $P_{\text{bf}} = 0.001$  depicted on Fig. 7a, we observe that the median value of  $\epsilon$  is small (about  $10^{-10}$ ) and independent of the sample ratio  $\rho$ . However, for  $\rho = 3$  a large dispersion of  $\epsilon$  is reported as reflected by the broad range of the quantiles. As the sample ratio  $\rho$  increases, the inter-quantile ranges shrink and become essentially constant for  $\rho > 7$ , with an estimated 99% quantile of  $\epsilon$  asymptotically below  $\epsilon = 10^{-8}$ . The fact that  $\epsilon$  does not converge to 0 as  $\rho$  increases is primarily due to the fact that the fraction of corrupted data in the iterative construction of the local maps is constant, so we can not expect to have  $\bar{u}^h \rightarrow \bar{u}_*^h$ . In addition, the IRLS algorithm is stopped when the weights are not evolving significantly from an iteration to another, leading to additional (small) differences between  $\bar{u}^h$  and  $\bar{u}_*^h$ . Note that if the 99% quantile of  $\epsilon$  is asymptotically low, replicas with significantly much larger  $\epsilon$  are still infrequently

reported. For instance, three replicas with  $\epsilon > 10^{-7}$  are reported for  $\rho = 7$ ; such events with large  $\epsilon$  become however more and more infrequent as  $\rho$  increases. This demonstrates that increasing the sample ratio  $\rho$  enhances the resilience of the computation.

Figures 7b and 7c show the same statistics, but for higher bit-flip probabilities  $P_{\text{bf}} = 0.01$  and  $0.1$ . The global behavior of  $\epsilon$  with  $\rho$  remains similar to the case  $P_{\text{bf}} = 0.001$  reported in Fig. 7a. Higher values of  $\rho$  are however necessary to achieve a given value of the quantiles of  $\epsilon$  when  $P_{\text{bf}}$  increases. Specifically, for  $P_{\text{bf}} = 0.01$  (Fig. 7b),  $\rho$  needs to be greater than 15 to obtain converged quantiles, whereas for  $P_{\text{bf}} = 0.1$  (Fig. 7c) the 99% quantile is still not converged for  $\rho = 20$  and a significant fraction of replicas have large  $\epsilon$ . This behavior is expected since  $P_{\text{bf}} = 0.1$  means that about 10% of the subdomain PDE solves are corrupted. Yet, the trend indicates that resilience can be improved by increasing  $\rho$  further. In practice, the fault probability is expected to be much smaller than the values considered in this study, and a reasonable value for  $\rho$ , for instance  $\rho = 3$ , is likely to be sufficient. Thus, the proposed approach would provide resilience to soft faults with negligible computational overhead (see the discussion in Section 4.4).

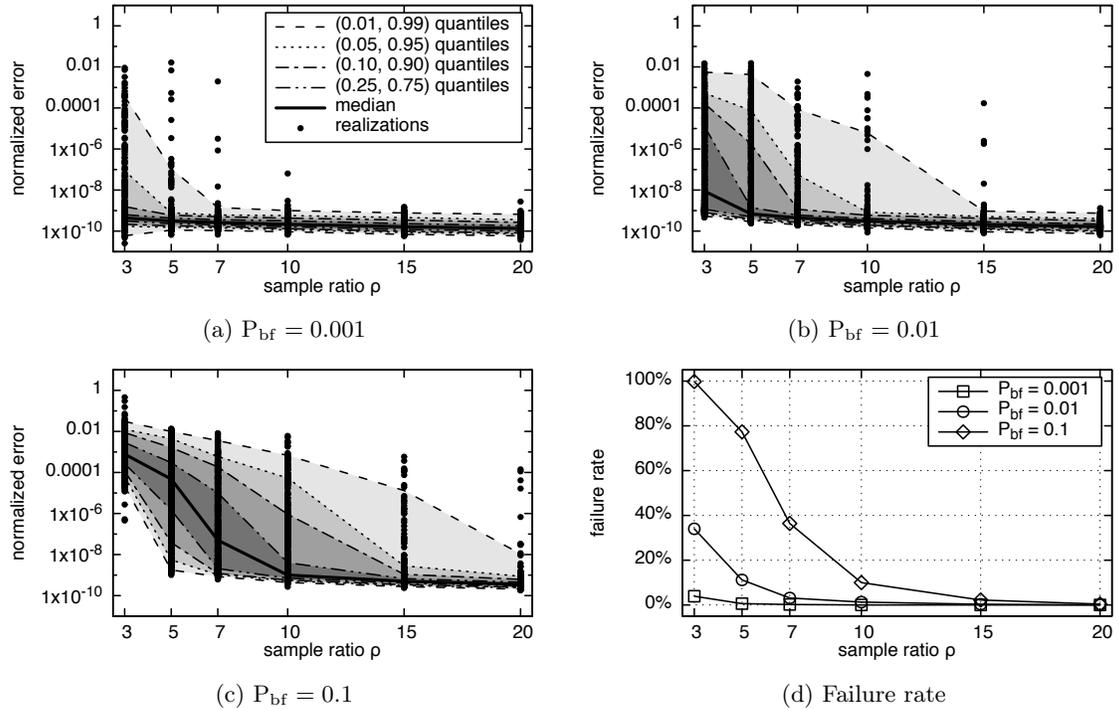


Figure 7: Analysis of the resilience for the covariance with  $L = 1$ . (a)–(c) quantiles of  $\epsilon$  (see (61)) as a function of the sample ratio  $\rho$ , and different bit-flip probabilities as indicated. Also shown are realizations of  $\epsilon$ . (d) failure rate of the resilient DD approach as a function of the the sample ratio  $\rho$  and for  $P_{\text{bf}} = 0.001, 0.01$  and  $0.1$ .

We can conclude from the previous numerical experiments that, given a fault probability, one can select an appropriate value for  $\rho$  in order to ensure resilience with prescribed confidence level. While quantiles are useful to characterize the expected range of  $\epsilon$ , a better probabilistic characterization of the resilience is needed. To this end, we consider that a particular computation is successful if

the solution  $\bar{u}^h$  satisfies

$$\|\bar{u}^h - \bar{u}_*^h\|_\Gamma \leq \|\bar{u}_*^h - \bar{u}^h\|_\Gamma, \quad (62)$$

where  $\bar{u}^h$  and  $\bar{u}_*^h$  use the same sample set of subdomain boundary values  $u^\pm$  and  $\xi$ . In words, a computation is deemed successful when the distance between  $\bar{u}^h$  and the corresponding fault-free solution,  $\bar{u}_*^h$ , is less or equal to the distance between  $\bar{u}_*^h$  and the classical Galerkin solution  $\bar{u}^h$ . When this criterion is not met, we say that the approach has failed. Due to the triangle inequality, this criterion ensures  $\|\bar{u}^h - \bar{u}^h\|_\Gamma \leq 2\|\bar{u}_*^h - \bar{u}^h\|_\Gamma$ , meaning that in a successful computation, the error with respect to the Galerkin solution is at most twice as large as the inherent error between the fault-free DD and Galerkin solutions. Using the criterion in Eq. (62), the failure rate of the resilient DD approach can be estimated using the 1,000 replicas by extracting the percentage of unsuccessful computations. Figure 7d shows the dependence of the failure rates on  $\rho$ , for the three bit-flip probabilities  $P_{\text{bf}}$  considered. As expected, regardless of the value of  $P_{\text{bf}}$ , the failure rate decreases as the sample ratio  $\rho$  increases. However, as the probability of bit-flips increases, larger values of  $\rho$  are needed to achieve a certain failure rate. For the smallest probability  $P_{\text{bf}} = 0.001$ , a failure rate of 0.6% is achieved for  $\rho = 5$ , whereas for  $\rho = 10$  and larger no failure is reported over the 1,000 replicas. For the intermediate case  $P_{\text{bf}} = 0.01$ , success for all replicas is obtained with  $\rho = 20$ , whereas for  $P_{\text{bf}} = 0.1$ , a 0.4% failure rate is estimated for  $\rho = 20$ .

Again, these results illustrate that resilience can be controlled by means of the sample ratio  $\rho$ . For more realistic values of  $P_{\text{bf}}$ , this ratio can be kept small enough so that resilience is obtained for a reasonable overhead. In addition, it should be stressed that in practice, mechanisms can be added to detect such failures (*e.g.* by looking at the PDE residual), and then mitigate them by adaptively generating additional samples.

Figure 8a shows the median and several quantiles of  $\epsilon$ , for  $L = 0.1$ ,  $K = 16$  and  $q = 2$  ( $P = 153$ ). We show only the intermediate bit-flip probability  $P_{\text{bf}} = 0.01$ , as results for  $P_{\text{bf}} = 0.001$  and  $P_{\text{bf}} = 0.1$  are qualitatively similar. A striking difference with the  $L = 1$  case (see Fig. 7b) is that realizations of  $\epsilon$  are much more clustered around their median value. In addition, as the sample ratio  $\rho$  increases, both the median and the quantiles of  $\epsilon$  decrease, but the inter-quantile range does not shrink much, in the log scale, compared to previous results. This indicates that, although the stochastic bases have sensibly the same dimension in the two cases, the solver is less sensitive here to bit-flips. In fact, further experiments (not shown) have highlighted that, on a given problem, the proposed approach generally remains resilient when the stochastic dimension  $K$  increases; on the contrary, a higher order PC expansion requires a higher sampling rate  $\rho$ . This trend can be explained by the need for a larger sample set to properly discriminate corrupted data when the PC order increases. Finally, the failure rate for the case  $L = 0.1$  is estimated (using 1,000 replicas) and contrasted to the case  $L = 1$  in Fig. 8b. The plot confirms the higher resilience for  $L = 0.1$ : no failure according to the criterion in (62) is reported over the 1,000 replicas and for all values of  $\rho$  tested. Again, the use of a lower PC order is mostly responsible for this behavior.

### 5.3. Domain decomposition parameters

To complete the study of the resilience, we investigate the influence of the domain decomposition parameters, namely the number of subdomains  $N$  (and inner interfaces) and their overlap  $\bar{h}$ . For simplicity, we use subdomains with equal size and the overlap is expressed as a percentage of subdomain length rather than an absolute value. Specifically, denoting by  $L_d = X_d^+ - X_d^-$  the length of a subdomain, we consider the overlap percentage  $\gamma \equiv 100 \times \bar{h}/L_d$ .

To investigate the influence of  $\gamma$  and  $N$  on the resilience, we consider two kinds of experiments inspired by scalability analyses in parallel computing:

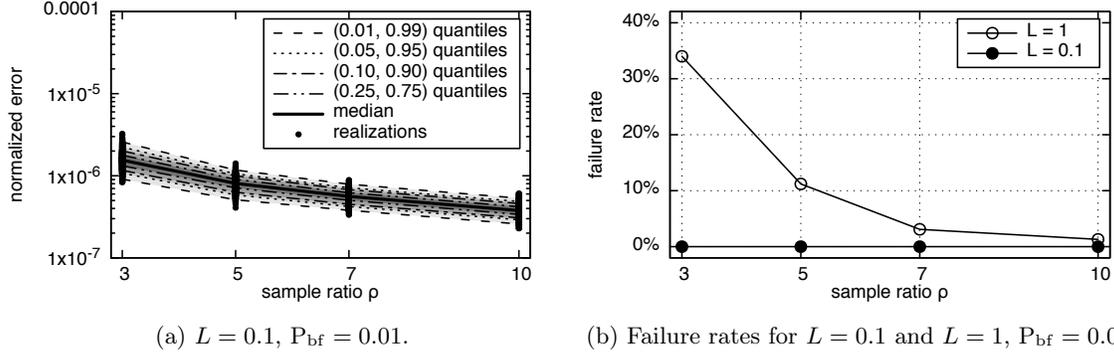


Figure 8: (a): quantiles and realization of  $\epsilon$  as a function of  $\rho$  for the case  $L = 0.1$  and  $P_{\text{bf}} = 0.01$ . (b): comparison of the failure rates of the proposed approach for  $L = 1$  ( $K = 5$  and  $q = 5$ ) and  $L = 0.1$  ( $K = 16$ ,  $q = 2$ ) with  $P_{\text{bf}} = 0.01$ . The stochastic discretization uses  $K = 16$  and  $q = 2$ . The rest of the numerical parameters are given in the text.

**Weak scaling** In these experiments, the number of subdomains is increased progressively keeping constant the number of finite-elements used for the discretization of the local problems, leading to a workload for each subdomain that is independent of  $N$ .

**Strong scaling** In these experiments, the total number of finite-elements  $N_e$  used for the global spatial discretization of  $\Omega$  is kept constant when  $N$  increases.

The comparison of the resilience as  $N$  varies requires an appropriate model for the soft faults, that is the definition of the dependence of the bit-flip probability on  $N$ . We shall assume that soft errors occur randomly with a given time-rate and independently from one subdomain to another. As such, the bit-flip probabilities should scale roughly with the computational load carried by each subdomain. For the present one-dimensional setting, the computational load (and time) for solving a local problem can be roughly estimated as linear in the number of elements used in the spatial discretization of the local problems (assembly and tridiagonal solve). Therefore, for the weak scaling experiments, the bit-flip probability is kept constant:  $P_{\text{bf}} = C_{\text{weak}}$ . For the strong scaling experiments, the bit-flip probability on the contrary scales with the inverse of the number of subdomains  $N$ ; we use  $P_{\text{bf}} = C_{\text{strong}}/N$ .

Figure 9 reports the evolution of the failure rate as the number of subdomain increases for the weak and strong scaling experiments. These computations are for the case of the covariance with  $L = 1$  with a fixed stochastic discretization  $K = 5$ ,  $q = 5$ , and constant sampling ratio  $\rho = 5$ . As previously, the failure rate is estimated from a set of 1,000 independent replicas using the criterion in (62) to decide the success of a computation. In the following experiments, high values of  $P_{\text{bf}}$  are considered without increasing the sampling rate  $\rho$ , so as to exaggerate failure effects and capture the trends.

Figure 9a shows the dependence on  $N$  of the failure rates in the weak scaling experiments and for 3 values of the overlapping ratio  $\gamma$ . In this setting, the domain was decomposed such that  $N_e/N \approx 21$  regardless of the overlap. It is seen that for every  $\gamma$  the failure rate increases with the number of subdomains  $N$ . In addition, as long as the failure rate is not too large, its dependence with  $N$  is essentially linear. This trend is expected because of the constant bit-flip probability set to  $P_{\text{bf}} = 0.01$ , in the weak scaling case, which implies that each map approximation involves the

same averaged fraction of corrupted samples irrespective of  $N$ . Therefore, the probability of having a map with significant errors increases with  $N$  as the number of maps to be reconstructed is affine in  $N$ . This in turn leads to a larger failure rate because the condensed system (see Section 3) is more likely to be affected by bit-flips for increasing  $N$ : the increasing probability of an erroneous map propagates to the solution of the condensed problem. This effect was also reported and further investigated in the deterministic case (see [14]).

A second important observation from the results reported in Fig. 9a concerns the higher failure rate for a lower overlap ratio. This trend seems to contradict the results reported previously for the error on the resilient map regression which was found to improve with decreasing overlapping (see Fig. 3). However, one needs to account for the propagation of these errors on the solution of the condensed system. In fact, it is well known that in the deterministic case the conditioning of the condensed system degrades with decaying overlap (see [14] for a detailed illustration in presence of bit-flips). This is also the case for the projected stochastic condensed system; it is consequently not a surprise that even though the errors on the map expansions improve, they affect more significantly the solution of the condensed system as  $\gamma \rightarrow 0$ . Also, the effect of  $\gamma$  is more pronounced as the number of subdomains increases.

Similar to the weak scaling, Fig. 9b reports, for different values of  $\gamma$ , the failure rates as function of  $N$  in the strong scaling case. In these computations we used bit-flip probabilities  $P_{\text{bf}} = 0.05/N$ , such that the weak and strong experiments coincide for  $N = 5$ . In addition, the total number of elements was  $N_e \approx 1,000$ . Contrary to the weak scaling case, it is seen that the number of subdomains does not influence the failure rate in the strong scaling experiments. Indeed, the probability of an erroneous map decreases as  $N$  increases, preventing error amplifications as the size of the condensed system increases (also the number of maps), with an essentially constant failure rate as a result. This claim is also supported by the much weaker effect of the overlap ratio on the failure rate, which is hardly noticeable in the results of Fig. 9b.

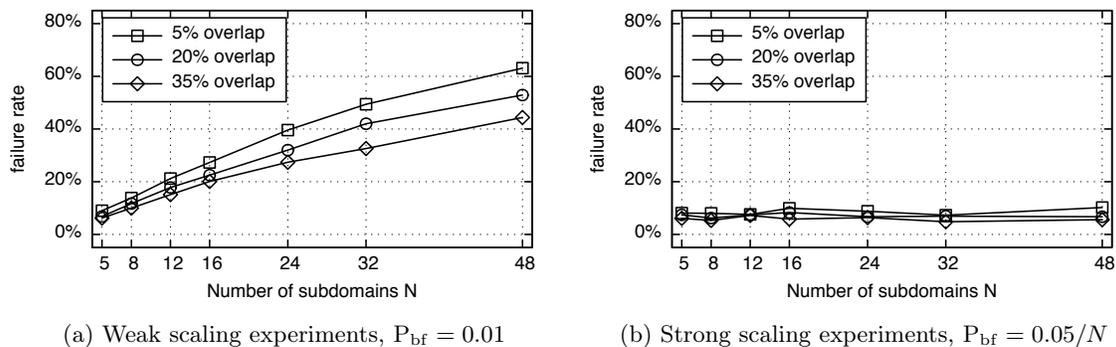


Figure 9: Dependency of the failure rate of the proposed method on the number of subdomains. (a): weak scaling experiments. (b): strong scaling experiments. The two cases coincide for  $N = 5$ . The failure rates were obtained over a total of 1,000 replicas. Details on the computational set-up are given in the text.

## 6. Discussion and conclusion

In this paper, we extended the work of [14] by constructing a resilient method for the solution of uncertain elliptic problems by means of a hybrid domain decomposition technique and Polynomial

Chaos (PC) approach. The method is designed for future exascale machines which are expected to be prone to soft faults. It relies on parallel and independent solves of multiple deterministic local problems, defined over subdomains, to reconstruct a PC representation of the local Dirichlet boundary-to-boundary maps. A LAD-lasso type regression has been proposed for the map reconstruction (see Section 4). Numerical experiments demonstrated that this regularized regression is able to properly overcome soft-fault-corrupted map values, provided that sufficiently many observations are available (even for a fixed ratio of corrupted observations). The local maps are then assembled to form the condensed linear system for the PC expansion of the unknown internal boundary values. This system is solved in a fault-free environment, and the PC expansions of internal boundary values can be subsequently used to recompute (independently) the stochastic solution over subdomains of interest.

The proposed method is particularly suited for exascale machines because its most computationally intensive parts, namely the generation of boundary-to-boundary map observations and the resilient PC regressions, are fully independent from a subdomain to another, enabling straightforward parallelism on distributed machines. This feature has to be contrasted with alternative approaches that would consist in using a resilient deterministic solver (*e.g.* as proposed in [14]) to sample the stochastic solution, and subsequently rely on a non-intrusive PC projection method (Non-Intrusive Spectral Projection, Sparse Grid Collocation method) to determine its PC expansion from the samples set. Such an approach would require significantly more communication and synchronization between computational nodes as global compatibility conditions need be enforced for each sample of the solution – obtaining each sample would, however, require less computational work. In our method, on the contrary, only the PC expansion coefficients of the maps need be transferred and only once. In addition, the projected condensed system may be assembled and solved in a distributed manner. The comparison between these approaches is part of ongoing work and will be reported on elsewhere.

In Section 5, the proposed method has been applied and validated on simple elliptic problems. We particularly focused on the analysis of its resilience to soft faults, which were modeled by introducing bit-flips randomly in the samples of the boundary-to-boundary maps. For the purpose of demonstrating resilience, unrealistically large rates of soft faults were considered, and computations showed that the method is able to ensure resilience in these extreme situations provided that the sampling rate is chosen large enough. In practice, future exascale machines are expected to exhibit finite but low rate of soft faults per computational node, such that the fraction of corrupted samples in a regression problem will be small and it can be reasonably claimed that the sampling rate will remain essentially the same as for a computation without fault. In other words, resilience will be achieved without significant computational overhead owing to the reformulation of the stochastic domain decomposition method based on PC approximation of the maps.

The robustness of the method with respect to the stochastic discretization and domain decomposition parameters was also numerically investigated. A key finding was that high-order PC computations appears more sensitive to soft faults, requiring a higher sampling rate to ensure resilience. This can be explained by the regression approach which easily disregards large bit-flips (owing to LAD) but can misinterpret small bit-flips as actual features of the maps if the number of observations is not large enough. The method, on the contrary, seems very robust in the case of a high number of random parameters and low PC expansion orders. The analysis also showed that, as expected, the resilience deteriorates when the number of subdomains increases while maintaining a constant computational load per subdomain (weak scaling case). However, assuming a fixed rate (per CPU time) of soft faults and a computational time for the solution of the local problems that

decays at least linearly with  $N$ , for a fixed global discretization (strong scaling case), resilience is not degraded by increasing the number of subdomains. The latter situation is more representative of exascale architectures where a large number of nodes will be used to solve small (independent) elementary problems.

As mentioned in the introduction, this work is of exploratory nature, and many developments and improvements can be explored to make the method effective on large-scale problems. An obvious point is the extension of the approach to two and three spatial dimensions. Although conceptually simple, extending the method to 2D and 3D will in practice raise several issues, in large part due to the increased dimensionality of the boundary-to-boundary maps, and accordingly the need for larger sample sets. However, the sampling rate  $\rho$  is not expected to change dramatically when considering higher spatial dimensions, even in presence of soft faults. Beside the need of a higher number of samples to construct the maps, we are currently considering alternative algorithms for the resolution of the LAD-lasso regression problem, including preconditioned iterative methods [47, 48] and reweighted coordinate descent [57, 58]. Similarly, improved solution methods for the linear system relating the PC expansions of the unknown boundary data would be necessary to apply the resilient approach on large scale problems and achieve scalability with the number of subdomains in addition to the scalability with respect to the samples distribution. Immediate efforts should consider the parallel iterative resolution of this system, with in particular the construction of dedicated preconditioners. Note that this system inheriting SPD properties of the original problem, the recent works in [31, 32, 33] could be adapted to our purpose. As a broader objective, we are also exploring various dimension-reduction approaches (see *e.g.* [59, 60, 61, 62, 63]) as well as the use of local uncertainty parameterizations [?] as potential routes to reduce both the complexity of the maps reconstruction and the dimensionality of condensed problem. Also in view of the implementation of the method on exascale prototypes, we are considering the treatment of hard faults in a server-client framework (see [53, 54]) as well as more elaborate soft fault models. Besides, we are exploring ways to detect failures (*i.e.* whenever our algorithm is not able to give an accurate answer). One way of doing so consists in looking at the PDE residual, which reflects the mismatch of the solution at the interfaces.

Finally, the extension of the method to stochastic non-linear problems can also be envisioned. In fact, the approach would already allow us to tackle the non-linear case by simply considering maps that are no longer linear in the local boundary data, that is constructing full expansions of  $f^{\pm,d}$  instead of imposing the affine forms in (16). Again, the proposed PC framework for the expansion of the maps can naturally accommodate for non-linear dependences with respect to the boundary data, but the main difficulty will be the definition of appropriate ranges and measures for the  $u^{\pm,d}$  values [14].

## Acknowledgments

This work was supported by the US Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under Award Number DE-SC0010540. The authors are also grateful to Dr. Habib Najm for helpful discussions.

## AppendixA. Compatibility conditions

Let  $u$  be the unique solution of (1) and let  $\mathbf{u}^\Gamma$  be the vector defined as:

$$\mathbf{u}^\Gamma = [u(X_1^+) \quad u(X_2^-) \quad u(X_2^+) \quad \cdots \quad u(X_{N-1}^-) \quad u(X_{N-1}^+) \quad u(X_N^-)]^\top. \quad (\text{A.1})$$

Let us consider the subdomain PDEs (4) with inner boundary data given by  $\mathbf{u}^\Gamma$  and with outer boundary conditions  $u^{1,-} = U_0$  and  $u^{N,+} = U_1$ . Clearly,  $\mathbf{u} = \mathbf{u}^\Gamma$  is solution of (9), or equivalently (5). In other words, given the global solution  $u$ , solving the subproblems using  $u$  at the interfaces as the inner boundary values yields solutions  $v^d$  that agree with  $u$  on each subdomain, and thus these inner boundary values naturally satisfy the compatibility equations. This shows that the compatibility equations admit at least one solution.

Let us now show that  $\mathbf{u}$  is the unique solution of the compatibility equations. In other words, let us show that imposing the compatibility conditions on the inner boundary values for the subproblems allows to uniquely recover the global solution  $u$ . Let

$$\hat{\mathbf{u}} = [\hat{u}^{1,+} \quad \hat{u}^{2,-} \quad \hat{u}^{2,+} \quad \dots \quad \hat{u}^{N-1,-} \quad \hat{u}^{N-1,+} \quad \hat{u}^{N,-}]^\top \quad (\text{A.2})$$

be a solution of the compatibility conditions (5) and let  $\hat{v}^d$  be the unique solution of

$$\begin{cases} \mathcal{L}v = g & \text{in } \Omega_d = (X_d^-, X_d^+) \\ v(X_d^-) = \hat{u}^{d,-}, \\ v(X_d^+) = \hat{u}^{d,+}, \end{cases} \quad (\text{A.3})$$

with outer boundary conditions  $\hat{u}^{1,-} = U_0$  and  $\hat{u}^{N,+} = U_1$ . Then,  $\hat{\mathbf{u}}$  being a solution of the compatibility conditions (5), it follows that the subdomain solutions  $\hat{v}^d$  solve the same PDE ( $\mathcal{L}v = g$  with same boundary conditions) in the overlapping regions, and consequently agree in these regions:

$$\hat{v}^d|_{[X_{d+1}^-, X_d^+]} = \hat{v}^{d+1}|_{[X_{d+1}^-, X_d^+]}, \quad \forall d = 1, \dots, N-1, \quad (\text{A.4})$$

so that we can define  $\hat{v}$  in  $\bar{\Omega}$  such that

$$\hat{v}|_{\bar{\Omega}_d} = \hat{v}^d|_{\bar{\Omega}_d}, \quad \forall d = 1, \dots, N. \quad (\text{A.5})$$

Clearly,  $\hat{v}$  is solution of (1). The solution being unique, we have  $\hat{v} = u$ . More particularly:

$$\begin{cases} \hat{v}(X_d^-) = \hat{u}^{d,-} = u(X_d^-), & \forall d = 2, \dots, N, \\ \hat{v}(X_d^+) = \hat{u}^{d,+} = u(X_d^+), & \forall d = 1, \dots, N-1, \end{cases} \quad (\text{A.6})$$

which shows that  $\hat{\mathbf{u}} = \mathbf{u}^\Gamma$  is the unique solution of (5). The extension to the stochastic case is straightforward.

## AppendixB. Exact linear maps

Let us define two auxiliary problems as follows:

$$\begin{cases} \mathcal{L}v = g & \text{in } \Omega_d \\ v(X_d^-) = 0, \\ v(X_d^+) = 0, \end{cases} \quad (\text{B.1}) \quad \begin{cases} \mathcal{L}v = 0 & \text{in } \Omega_d \\ v(X_d^-) = 1, \\ v(X_d^+) = 0. \end{cases} \quad (\text{B.2})$$

Due to the linearity of the operator  $\mathcal{L}$ , if  $v_{0,0}^d$  is solution of (B.1) and  $\bar{v}_{1,0}^d$  is solution of (B.2), then  $v_{0,0}^d + u^{d,-}\bar{v}_{1,0}^d + u^{d,+}(1 - \bar{v}_{1,0}^d)$  is solution of (4). This last expression actually shows that the function  $v^d : \Omega_d \times \mathbb{R}^2 \rightarrow \mathbb{R}$  defined as

$$v^d(x; u^{d,-}, u^{d,+}) = v_{0,0}^d(x) + u^{d,-} \cdot \bar{v}_{1,0}^d(x) + u^{d,+} \cdot [1 - \bar{v}_{1,0}^d(x)] \quad (\text{B.3})$$

linearly maps the boundary values  $u^{d,-}, u^{d,+}$  of the subproblem (4) to its solution at any  $x \in \bar{\Omega}_d$ . Using the following notations:

$$a^{d,-} = v_{0,0}^d(X_{d-1}^+), \quad a^{d,+} = v_{0,0}^d(X_{d+1}^-), \quad (\text{B.4})$$

$$b^{d,-} = \bar{v}_{1,0}^d(X_{d-1}^+), \quad b^{d,+} = \bar{v}_{1,0}^d(X_{d+1}^-), \quad (\text{B.5})$$

$$c^{d,-} = 1 - b^{d,-}, \quad c^{d,+} = 1 - b^{d,+}, \quad (\text{B.6})$$

leads to equation (7). Note that if there is no source term, *i.e.*  $g = 0$  in  $\Omega^d$ , then  $v_{0,0}^d = 0$  and thus  $a^{d,\pm} = 0$ . The extension to the stochastic case is straightforward.

- [1] F. Cappello, A. Geist, B. Gropp, L. Kale, B. Kramer, M. Snir, Toward exascale resilience, International Journal of High Performance Computing Applications.
- [2] F. Cappello, A. Geist, W. Gropp, S. Kale, B. Kramer, M. Snir, Toward exascale resilience: 2014 update, Supercomputing frontiers and innovations 1 (1).  
URL <http://superfri.org/superfri/article/view/14>
- [3] M. Snir, R. W. Wisniewski, J. A. Abraham, S. V. Adve, S. Bagchi, P. Balaji, J. Belak, P. Bose, F. Cappello, B. Carlson, et al., Addressing failures in exascale computing, International Journal of High Performance Computing Applications (2014) 1094342014522573.
- [4] J. Shalf, S. Dosanjh, J. Morrison, Exascale computing technology challenges, in: Proceedings of the 9th International Conference on High Performance Computing for Computational Science, VECPAR'10, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 1–25.  
URL <http://dl.acm.org/citation.cfm?id=1964238.1964240>
- [5] P. Laplante, Dictionary of Computer Science, Engineering and Technology, Taylor & Francis, 2000.  
URL <https://books.google.com.sa/books?id=U1M3c1UwCfEC>
- [6] K. Teranishi, M. A. Heroux, Toward local failure local recovery resilience model using mpi-ulfm, in: Proceedings of the 21st European MPI Users' Group Meeting, EuroMPI/ASIA '14, ACM, New York, NY, USA, 2014, pp. 51:51–51:56. doi:10.1145/2642769.2642774.  
URL <http://doi.acm.org/10.1145/2642769.2642774>
- [7] G. Bosilca, R. Delmas, J. Dongarra, J. Langou, Algorithm-based fault tolerance applied to high performance computing, Journal of Parallel and Distributed Computing 69 (4) (2009) 410–416. doi:10.1016/j.jpdc.2008.12.002.
- [8] C. Ding, C. Karlsson, H. Liu, T. Davies, Z. Chen, Matrix multiplication on gpus with on-line fault tolerance, in: Parallel and Distributed Processing with Applications (ISPA), 2011 IEEE 9th International Symposium on, 2011, pp. 311–317. doi:10.1109/ISPA.2011.50.

- [9] P. Du, A. Bouteiller, G. Bosilca, T. Herault, J. Dongarra, Algorithm-based fault tolerance for dense matrix factorizations, in: Proceedings of the 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming, PPOPP '12, ACM, New York, NY, USA, 2012, pp. 225–234. doi:10.1145/2145816.2145845.  
URL <http://doi.acm.org/10.1145/2145816.2145845>
- [10] Z. Chen, Algorithm-based recovery for iterative methods without checkpointing, in: Proceedings of the 20th international symposium on High performance distributed computing, HPDC '11, ACM, New York, NY, USA, 2011, pp. 73–84. doi:10.1145/1996130.1996142.  
URL <http://doi.acm.org/10.1145/1996130.1996142>
- [11] K. Ferreira, J. Stearley, J. H. Laros, III, R. Oldfield, K. Pedretti, R. Brightwell, R. Riesen, P. G. Bridges, D. Arnold, Evaluating the viability of process replication reliability for exascale systems, in: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11, ACM, New York, NY, USA, 2011, pp. 44:1–44:12. doi:10.1145/2063384.2063443.  
URL <http://doi.acm.org/10.1145/2063384.2063443>
- [12] A. Shye, T. Moseley, V. Reddi, J. Blomstedt, D. Connors, Using process-level redundancy to exploit multiple cores for transient fault tolerance, in: Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP International Conference on, 2007, pp. 297–306. doi:10.1109/DSN.2007.98.
- [13] K. Malkowski, P. Raghavan, M. Kandemir, Analyzing the soft error resilience of linear solvers on multicore multiprocessors, in: Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on, 2010, pp. 1–12. doi:10.1109/IPDPS.2010.5470411.
- [14] K. Sargsyan, F. Rizzi, P. Mycek, C. Safta, K. Morris, H. Najm, O. Le Maître, O. Knio, B. Debusschere, Fault resilient domain decomposition preconditioner for PDEs, SIAM Journal on Scientific Computing Accepted for publication.
- [15] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Numerical mathematics and scientific computation, Clarendon Press, 1999.  
URL <https://books.google.com/books?id=CswWZ9Ca9IIC>
- [16] A. Toselli, O. Widlund, Domain Decomposition Methods - Algorithms and Theory, Springer Series in Computational Mathematics, Springer, 2005. doi:10.1007/b137868.  
URL <http://link.springer.com/book/10.1007/b137868>
- [17] P. G. Bridges, K. B. Ferreira, M. A. Heroux, M. Hoemmen, Fault-tolerant linear solvers via selective reliability, arXiv.org.
- [18] R. G. Ghanem, P. D. Spanos, Stochastic finite elements: a spectral approach, Courier Corporation, 2003.
- [19] O. Le Maître, O. Knio, Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics, Scientific Computation, Springer Netherlands, 2010.  
URL <https://books.google.com/books?id=10FCquL6SxYC>

- [20] O. P. Le Maître, M. T. Reagan, H. N. Najm, R. G. Ghanem, O. M. Knio, A stochastic projection method for fluid flow: Ii. random process, *Journal of Computational Physics* 181 (1) (2002) 9 – 44. doi:<http://dx.doi.org/10.1006/jcph.2002.7104>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999102971044>
- [21] M. T. Reagan, H. N. Najm, R. G. Ghanem, O. M. Knio, Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection, *Combustion and Flame* 132 (3) (2003) 545–555.
- [22] P. G. Constantine, M. S. Eldred, E. T. Phipps, Sparse pseudospectral approximation method, *Computer Methods in Applied Mechanics and Engineering* 229–232 (2012) 1 – 12. doi:<http://dx.doi.org/10.1016/j.cma.2012.03.019>.  
URL <http://www.sciencedirect.com/science/article/pii/S0045782512000953>
- [23] P. R. Conrad, Y. M. Marzouk, Adaptive smolyak pseudospectral approximations, *SIAM Journal on Scientific Computing* 35 (6) (2013) A2643–A2670. arXiv:<http://dx.doi.org/10.1137/120890715>, doi:10.1137/120890715.  
URL <http://dx.doi.org/10.1137/120890715>
- [24] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, *SIAM Journal on Numerical Analysis* 45 (3) (2007) 1005–1034. arXiv:<http://dx.doi.org/10.1137/050645142>, doi:10.1137/050645142.  
URL <http://dx.doi.org/10.1137/050645142>
- [25] F. Nobile, R. Tempone, C. G. Webster, An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM Journal on Numerical Analysis* 46 (5) (2008) 2411–2442. arXiv:<http://dx.doi.org/10.1137/070680540>, doi:10.1137/070680540.  
URL <http://dx.doi.org/10.1137/070680540>
- [26] M. Pellissetti, R. Ghanem, Iterative solution of systems of linear equations arising in the context of stochastic finite elements, *Advances in engineering software* 31 (8) (2000) 607–616. doi:10.1016/S0965-9978(00)00034-X.
- [27] O. Le Maître, O. Knio, B. Debusschere, H. Najm, R. Ghanem, A multigrid solver for two-dimensional stochastic diffusion equations, *Computer Methods in Applied Mechanics and Engineering* 192 (41–42) (2003) 4723 – 4744. doi:[http://dx.doi.org/10.1016/S0045-7825\(03\)00457-2](http://dx.doi.org/10.1016/S0045-7825(03)00457-2).  
URL <http://www.sciencedirect.com/science/article/pii/S0045782503004572>
- [28] H. G. Matthies, A. Keese, Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, *Computer Methods in Applied Mechanics and Engineering* 194 (12–16) (2005) 1295 – 1331, special Issue on Computational Methods in Stochastic Mechanics and Reliability Analysis. doi:<http://dx.doi.org/10.1016/j.cma.2004.05.027>.  
URL <http://www.sciencedirect.com/science/article/pii/S0045782504003950>
- [29] O. Le Maître, A Newton method for the resolution of steady stochastic Navier–Stokes equations, *Computers & Fluids* 38 (8) (2009) 1566 – 1579. doi:<http://dx.doi.org/10.1016/j.compfluid.2009.01.001>.  
URL <http://www.sciencedirect.com/science/article/pii/S0045793009000036>

- [30] E. Rosseel, S. Vandewalle, Iterative solvers for the stochastic finite element method, *SIAM Journal on Scientific Computing* 32 (1) (2010) 372–397. doi:10.1137/080727026.
- [31] B. Sousedík, R. G. Ghanem, E. T. Phipps, Hierarchical schur complement preconditioner for the stochastic galerkin finite element methods, *Numerical Linear Algebra with Applications* 21 (1) (2014) 136–151. doi:10.1002/nla.1869.  
URL <http://dx.doi.org/10.1002/nla.1869>
- [32] E. Ullmann, C. E. Powell, Solving log-transformed random diffusion problems by stochastic galerkin mixed finite element methods, *SIAM/ASA Journal on Uncertainty Quantification* 3 (1) (2015) 509–534. arXiv:<http://dx.doi.org/10.1137/14100097X>, doi:10.1137/14100097X.  
URL <http://dx.doi.org/10.1137/14100097X>
- [33] W. Dahmen, R. DeVore, L. Grasedyck, E. Süli, Tensor-sparsity of solutions to high-dimensional elliptic partial differential equations, *Foundations of Computational Mathematics* 16 (4) (2016) 813–874. doi:10.1007/s10208-015-9265-9.  
URL <http://dx.doi.org/10.1007/s10208-015-9265-9>
- [34] H. Wang, G. Li, G. Jiang, Robust regression shrinkage and consistent variable selection through the lad-lasso, *Journal of Business & Economic Statistics* 25 (3) (2007) 347–355.
- [35] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2) (2005) 301–320.
- [36] M. Garbey, Acceleration of the schwarz method for elliptic problems, *SIAM Journal on Scientific Computing* 26 (6) (2005) 1871–1893.
- [37] S. Portnoy, R. Koenker, et al., The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators, *Statistical Science* 12 (4) (1997) 279–300.
- [38] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* (1996) 267–288.
- [39] A. Cohen, M. A. Davenport, D. Leviatan, On the stability and accuracy of least squares approximations, *Foundations of computational mathematics* 13 (5) (2013) 819–834.
- [40] G. Migliorati, F. Nobile, E. von Schwerin, R. Tempone, Approximation of quantities of interest in stochastic PDEs by the random discrete  $L^2$  projection on polynomial spaces, *SIAM Journal on Scientific Computing* 35 (3) (2013) A1440–A1460.
- [41] M. R. Osborne, *Finite Algorithms in Optimization and Data Analysis*, John Wiley & Sons, Inc., New York, NY, USA, 1985.
- [42] E. J. Schlossmacher, An iterative technique for absolute deviations curve fitting, *Journal of the American Statistical Association* 68 (344) (1973) pp. 857–859.  
URL <http://www.jstor.org/stable/2284512>
- [43] D. Coleman, P. Holland, N. Kaden, V. Klema, S. C. Peters, A system of subroutines for iteratively reweighted least squares computations, *ACM Trans. Math. Softw.* 6 (3) (1980) 327–336. doi:10.1145/355900.355905.  
URL <http://doi.acm.org/10.1145/355900.355905>

- [44] J. A. Scales, A. Gersztenkorn, Robust methods in inverse theory, *Inverse problems* 4 (4) (1988) 1071.
- [45] R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing, in: *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on, IEEE, 2008*, pp. 3869–3872.
- [46] I. Daubechies, R. DeVore, M. Fornasier, C. S. Güntürk, Iteratively reweighted least squares minimization for sparse recovery, *Communications on Pure and Applied Mathematics* 63 (1) (2010) 1–38.
- [47] C. C. Paige, M. A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Softw.* 8 (1) (1982) 43–71. doi:10.1145/355984.355989. URL <http://doi.acm.org/10.1145/355984.355989>
- [48] C. C. Paige, M. A. Saunders, Algorithm 583: LSQR: Sparse linear equations and least squares problems, *ACM Trans. Math. Softw.* 8 (2) (1982) 195–209. doi:10.1145/355993.356000. URL <http://doi.acm.org/10.1145/355993.356000>
- [49] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: *Ijcai*, Vol. 14, 1995, pp. 1137–1145.
- [50] S. Arlot, A. Celisse, et al., A survey of cross-validation procedures for model selection, *Statistics surveys* 4 (2010) 40–79.
- [51] IEEE Standard for Floating-Point Arithmetic, Tech. rep., Microprocessor Standards Committee of the IEEE Computer Society (2008). doi:10.1109/IEEESTD.2008.4610935.
- [52] W. Bland, A. Bouteiller, T. Herault, G. Bosilca, J. Dongarra, Post-failure recovery of mpi communication capability: Design and rationale, *International Journal of High Performance Computing Applications* 27 (3) (2013) 244–254. arXiv:<http://hpc.sagepub.com/content/27/3/244.full.pdf+html>, doi:10.1177/1094342013488238. URL <http://hpc.sagepub.com/content/27/3/244.abstract>
- [53] F. Rizzi, K. Morris, K. Sargsyan, P. Mycek, C. Safta, H. Najm, O. Le Maître, O. Knio, B. Debusschere, Partial differential equations preconditioner resilient to soft and hard faults, in: *FTS – IEEE Cluster*, 2015.
- [54] F. Rizzi, K. Morris, K. Sargsyan, P. Mycek, C. Safta, B. Debusschere, O. LeMaitre, O. Knio, ULFM-MPI implementation of a resilient task-based partial differential equations preconditioner, in: *Proceedings of the ACM Workshop on Fault-Tolerance for HPC at Extreme Scale, FTXS '16*, ACM, New York, NY, USA, 2016, pp. 19–26. doi:10.1145/2909428.2909429. URL <http://doi.acm.org/10.1145/2909428.2909429>
- [55] P. Mycek, F. Rizzi, O. Le Maître, K. Sargsyan, K. Morris, C. Safta, B. Debusschere, O. Knio, Discrete a priori bounds for the detection of corrupted PDE solutions in exascale computations, *SIAM Journal on Scientific Computing* (submitted).

- [56] J. Charrier, Strong and weak error estimates for elliptic partial differential equations with random coefficients, *SIAM Journal on Numerical Analysis* 50 (1) (2012) 216–246. arXiv:<http://dx.doi.org/10.1137/100800531>, doi:10.1137/100800531. URL <http://dx.doi.org/10.1137/100800531>
- [57] T. T. Wu, K. Lange, Coordinate descent algorithms for lasso penalized regression, *The Annals of Applied Statistics* 2 (1) (2008) 222–244.
- [58] P. Breheny, J. Huang, Coordinate descent algorithms for nonconvex penalized regression with applications to biological feature selection, *The Annals of Applied Statistics* 5 (1) (2011) 232–253.
- [59] M. Arnst, R. Ghanem, E. Phipps, J. Red-Horse, Dimension reduction in stochastic modeling of coupled problems, *International Journal for Numerical Methods in Engineering* 92 (11) (2012) 940–968.
- [60] M. Arnst, R. Ghanem, E. Phipps, J. Red-Horse, Measure transformation and efficient quadrature in reduced-dimensional stochastic modeling of coupled problems, *International Journal for Numerical Methods in Engineering* 92 (12) (2012) 1044–1080.
- [61] M. Arnst, R. Ghanem, E. Phipps, J. Red-Horse, Reduced chaos expansions with random coefficients in reduced-dimensional stochastic modeling of coupled problems, *International Journal for Numerical Methods in Engineering* 97 (5) (2014) 352–376.
- [62] Q. Liao, K. Willcox, A domain decomposition approach for uncertainty analysis, *SIAM Journal on Scientific Computing* 37 (1) (2015) A103–A133.
- [63] Y. Chen, J. Jakeman, C. Gittelson, D. Xiu, Local polynomial chaos expansion for linear differential equations with high dimensional random inputs, *SIAM Journal on Scientific Computing* 37 (1) (2015) A79–A102.