The Multi-index Stochastic Collocation method for uncertainty quantification of partial differential equations with random parameters

J. Beck[1], L. Tamellini[2], R. Tempone[1,3]
[1] *King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia*
[2] *Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes" (CNR-IMATI), Pavia, Italy*
[3] *Alexander von Humboldt Prof. in Mathematics of Uncertainty Quantification, RWTH Aachen Univ., Germany*

***Abstract*—In many real-life applications, one needs to solve partial differential equations (PDEs) to predict the behavior of a system, most often by numerical methods. This goal is often hampered by the fact that the parameters of the equations might be not known exactly, and modeled as random variables; one therefore would like to assess how this uncertainty propagates to the solution of the PDE. To this end, in this contribution we discuss the Multi-Index Stochastic Collocation (MISC) method, and show its effectiveness with on a numerical test.**

***Keywords*—Stochastic collocation, combination technique, multi-level methods, PDE with random coefficients**

I. Introduction

In many applications, scientists and engineers need to solve ordinary / partial differential equations (ODEs / PDEs respectively) to predict the behavior of a system, either analytically or more commonly by computer simulations (in the remainder of this contribution we focus only on PDEs). While the mathematical structure of these PDEs is often well-understood (at least for classical applications, such as solid mechanics, fluid dynamics, heat exchange, electromagnetism, etc) and numerous well-studied numerical methods exist to approximate their solutions, the predictive power of these simulations is often hampered by the fact that the parameters of the equations might be not known exactly, and they might be considered e.g. as random variables (or random fields). "Parameters" here is used in a broad sense, and includes physical/chemical coefficients (e.g., density, viscosity, thermal conductivity, permeability), shape of the domain, initial conditions, boundary conditions, which can be thoughts as the "inputs" or "data" of the PDE, as opposed to the solution of the PDE, which can be thought as the "output" of the PDE. This uncertainty can be due to multiple reasons, such as limited experimental measurements, or intrinsic randomness of the quantity (e.g. rainfall, earthquakes).

It is therefore crucial to assess how the uncertainty on the data/inputs of the ODEs/PDEs propagates to the outputs of these equations, i.e., to compute mean, variance and higher moments of the solution, and ideally its probability density function. This is the goal of the so-called forward Uncertainty Quantification (UQ) analyses. Other related tasks are inverse UQ and Optimization Under Uncertainty): they can be tackled with similar approaches, but we do not discuss them here. In this context, it is useful to think of the solution of the PDE as a random function, that associates to each realization of the random parameters the corresponding solution of the PDE. In the following, assuming that the PDE depends on N random parameters: we denote by $\boldsymbol{y} \in \mathbb{R}^N$ the vector containing the realizations of the random parameters, by $\Gamma \subset \mathbb{R}^N$ the set in which $\boldsymbol{y}$ can take values (the so-called "parameter space"), and by $u(\boldsymbol{x}, \boldsymbol{y})$ the solution of the PDE, where we have also highlighted the dependence of u on the space/time variables $\boldsymbol{x}$.

Most methods for forward UQ rely on solving the PDE for multiple realizations of the random parameters (i.e., sampling $u(\boldsymbol{y})$ over Γ) and then post-processing the corresponding solutions to obtain the desidered statistical information on the solution. The most trivial of these methods is of course the Monte Carlo method, whereby the statistical information for the solution is obtained by simply averaging the samples of $u(\boldsymbol{y})$ obtained. It is of course possible to replace Monte-Carlo with more effective sampling methods, such as Quasi Monte Carlo or Latin Hypercube Sampling, see e.g. [7]. These methods however do not take full advantage of the possible smoothness of the function to be sampled, i.e. the fact that the map $\boldsymbol{y} \to u(\boldsymbol{x}, \boldsymbol{y})$ could be not only a continuous function but actually a function whose derivatives up to a certain degree might be continuous or square-integrable – even an analytic function at times. These properties derive from the structure of the PDE at hand. This fact can instead been exploited by methods that are more traditional of the numerical analysis background, i.e., numerical quadrature and interpolation methods (stemming from the fact that computing an expected value / higher moment is nothing but a weighted integral over the parameter space Γ). On the other hand, classical numerical quadrature/interpolation methods scale poorly with the dimension N of the parameter space – in the worst case, the number of required samples grows exponentially with N ("curse of dimensionality"). In addition to this, the function $u(\boldsymbol{y})$ to be sampled is typically expensive to evaluate (since it requires solving numerically a PDE), so that naive approaches become unbearably expensive. A very popular and quite effective approach to reduce the "curse of dimensionality" effect are the so-called "sparse-grids" schemes, i.e., quadrature/interpolation schemes carefully designed to deal with high-dimensional, possibly smooth functions, see e.g. [9], [10].

Another very popular approach to reduce the computational complexity is the so-called multi-level approach, first proposed in the context of numerical finance and then applied to engineering applications [2], [3]. In this approach, a first sampling of the parameters space is performed by solving the required PDEs with a coarse computational mesh (hence, with

a cheap method); then, a refinement of the mesh is introduced and a few additional PDEs are solved, to correct the previous estimate. The procedure can be iterated on a hierarchy of increasingly refined meshes where at each level less and less PDEs are solved. In this way, the computational cost of the procedure is substantially reduced without compromising the accuracy of the prediction. In this version (and in the several improvements proposed in literature, see e.g. [5], [8]) the sampling strategy at each level is still a Monte-Carlo / Quasi-Monte-Carlo strategy, i.e., strategies that again do not fully exploit the regularity of the solution u. This can be reached instead combining the multi-level strategy with the sparse-grids schemes mentioned above, [1], [4], [11]. In this contribution we focus on one specific example of such method, the Multi-Index Stochastic Collocation method. In particular, our exposition follows closely [1].

II. Multi-index Stochastic Collocation (MISC)

To fix the ideas, let's consider a specific PDE with random coefficients for which we want to perform a forward UQ analysis. For instance, let's consider the heat equation to compute the pointwise temperature of a metal bar being heated, whose heat-conduction coefficient is uncertain. In this scenario, the parameter $\boldsymbol{y}$ represents the heat conduction coefficient: one parameter $N = 1$ is enough if the bar is homogeneous, while if the bar is composite with pieces with different material, then we will need $N > 1$. Each $y_n, n = 1, \dots N$ can be thought as a uniform random variable over a certain range $y_n \sim \mathcal{U}(a_n, b_n)$, and we can assume y_n to be independent. The parameter space Γ is the hypercube obtained by taking Cartesian products of each $[a_n, b_n]$, and the probability density function of $\boldsymbol{y}$ is simply $\rho(\boldsymbol{y}) = \prod_{n=1}^{N} \frac{1}{b_n - a_n}$. The solution u of the PDE is the pointwise temperature in the bar, and is a function of the spatial coordinate $\boldsymbol{x}$ as well as of the random heat-conductivities $\boldsymbol{y}$, i.e. $u = u(\boldsymbol{x}, \boldsymbol{y})$.

Let us now consider an exahedral mesh to approximate u for a given value of the heat coefficient, by solving the PDE. For simplicity, let us assume that all the elements have the same size and are allowed to be non-cubic, i.e., their edges have size $h_1 = c_1 2^{-\alpha_1}$, $h_2 = c_2 2^{-\alpha_2}$, $h_3 = c_3 2^{-\alpha_3}$, for some constants c_1, c_2, c_3 and user-defined integer values $\alpha_1, \alpha_2, \alpha_3$. We collect the three values of α_i in a multi-index $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]$; prescribing the multi-index $\boldsymbol{\alpha}$ thus prescribes the computational mesh to be generated. If this flexibility is not allowed by the mesh-generator (or by the problem itself), one can set $\alpha_1 = \alpha_2 = \alpha_3 = \alpha$, i.e., controlling the mesh-generation by a single integer value α. Let us denote by $G_{\boldsymbol{\alpha}}$ the quantity of interest of our UQ analysis computed over the mesh specified by $\boldsymbol{\alpha}$; this could be for instance the value of the temperature in a specific point of the bar being heated. Thus, the final goal of the UQ analysis is to compute an approximation of e.g. $\mathbb{E}[G_{\boldsymbol{\alpha}}]$, i.e., of the expected value of $G_{\boldsymbol{\alpha}}$.

The MISC method can be used for this goal. It is based on selecting the values $\boldsymbol{y}_j$ as points of a Cartesian grid obtained by tensorization of univariate quadrature rules (which should be chosen according to $\rho(\boldsymbol{y})$ for computational efficiency). In this work, we use as univariate quadrature rule the Clenshaw–Curtis (CC) univariate quadrature, which is optimal when $y_1, y_2, \dots, y_N$ are uniform and independent random variables as in our case. Then, for a multi-index $\boldsymbol{\beta} \in \mathbb{N}^N$ and given the function $m(i)$ with $m(0) = 0$, $m(1) = 1$, $m(i) = 2^{i-1} + 1$ for $i \geq 2$, $m(\beta_1)$ CC values are generated for y_1, $m(\beta_2)$ CC values for y_2 etc, and then consider the grid obtained by taking the Cartesian product of the N sets of points thus generated. The quadrature weight of each point of the Cartesian grid is immediately obtained by taking the product of the corresponding univariate weights. The approximation of $\mathbb{E}[G_{\boldsymbol{\alpha}}]$ computed over this grid is denoted as $\mathcal{Q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$. Clearly, one would like to have both multi-indices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with large components, say $\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$, i.e., to average the values of many PDE solutions over a refined computational mesh. However, as we already mentioned this is typically unfeasible due to computational costs.

Instead, the idea of MISC resorts to the previously mentioned multi-level approach, i.e. the single, highly refined approximation $\mathcal{Q}_{\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star}$ is replaced by a linear combination of many coarser $\mathcal{Q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$, where whenever one refines the spatial discretization $\boldsymbol{\alpha}$, the quadrature level $\boldsymbol{\beta}$ is kept to a minimum and vice versa (of course, the combined cost of computing the set of coarse discretizations should be smaller than the cost of the highly refined one). In formula,

$$\mathbb{E}[G_{\boldsymbol{\alpha}^\star}] \approx \mathcal{Q}_{\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star} \approx \sum_{[\boldsymbol{\alpha},\boldsymbol{\beta}] \in \mathcal{I}} c_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} \mathcal{Q}_{\boldsymbol{\alpha},\boldsymbol{\beta}} \tag{1}$$

where $c_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ is a real number and $\mathcal{I}$ is a collection of feasible discretizations, designed with the purpose just explained. For instance,

$$\mathcal{I} = \{\boldsymbol{\alpha} \in \mathbb{R}^3, \boldsymbol{\beta} \in \mathbb{R}^N : |\boldsymbol{\alpha}| + |\boldsymbol{\beta}| \leq L\}$$

for some integer value L. A suitable set $\mathcal{I}$ can be designed can either be designed a-priori, by a careful analysis of the PDE at hand, see e.g. [1], or on-the-run by adaptive algorithms, see e.g. [6]; in this contribution the focus is on the former option.

III. Numerical results

To show the effectiveness of MISC, we briefly comment in this section on a set of results originally reported in [1], to which we refer the interested reader for details. The metal bar is shown in Figure 1 top-left. The pointwise heat-conductivity is modeled as a random field, parametrizes with $N = 3$ i.i.d. uniform random variables, $y_i \sim \mathcal{U}(-1, 1)$, i.e., $\Gamma = [-1, 1]^3$; one possible realization of this random field is shown in Figure 1 top-right. The quantity of interest (the quantity $G_{\boldsymbol{\alpha}}$) is the integral of the temperature over the metal bar, and we aim at computing its expected value, $\mathbb{E}[G_{\boldsymbol{\alpha}}]$.

Figure 1 bottom-left reports the growth of the computational time required to approximate $\mathbb{E}[G_{\boldsymbol{\alpha}}]$ as we require a smaller and smaller tolerance. The methods considered are MISC, the standard Multi-Level Monte-Carlo method (MLMC) and a certain variant of MLMC, called Multi-Index Monte Carlo (MIMC). We can immediately see that the computational time required by MISC grows significantly slower than for MLMC and MIMC, especially for small tolerances. We also report in dotted lines the theoretical growth of the computational times for all these methods (as well as standard Monte-Carlo), which show good agreement between the theory and the actual computational times.

Finally, Figure 1 bottom-right shows for the number of PDEs solved on each mesh, for some of the tolerances

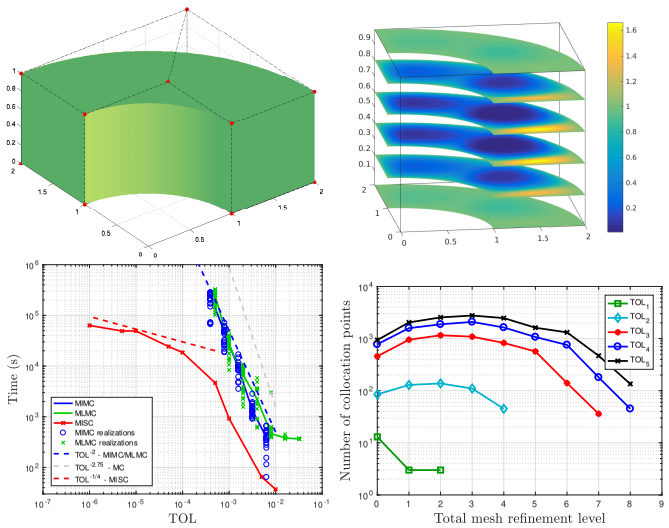

Fig. 1. Numerical test (plots already appeared in [1]). Metal bar (top-left); one realizations of the random heat-capacity (top-right); Tolerance-vs-time plot for MISC and other methods (bottom-left); number of PDEs solved on each grid for each tolerance (bottom-right).

required to obtain the tolerance-vs-time plot. The horizontal axis reports the "total mesh refinement", i.e. the sum $\alpha_1 + \alpha_2 + \alpha_3$: the larger this sum, the more refined the computational mesh, and the more expensive solving the PDE over it. Tolerances are named $TOL_1, TOL_2, \ldots, TOL_5$, with TOL_1 being the largest and TOL_5 the smallest. As expected, if a large tolerance is required, only a few PDE solves are needed, on a few meshes, none of them very refined. As the required tolerance gets smaller, the number of meshes required increases, and more refined meshes are introduced; at the same time, more and more PDEs are being solved on each mesh. Crucially, however, the number of PDEs that need to be solved on the refined meshes is always significantly smaller than those to be solved on coarse meshes: e.g., for TOL_5 we are solving a few thousand PDEs on meshes with total refinement 3 and slightly more than a hundred on the meshes with total refinement 8, which makes MISC a quite effective method for forward UQ purposes.

References

[1] J. Beck, L. Tamellini, R. Tempone, "IGA-based Multi-Index Stochastic Collocation for random PDEs on arbitrary domains," *Computer Methods in Applied Mechanics and Engineering*, vol. 351, pp. 330 – 350, 2019.

[2] K. Cliffe, M. Giles, R. Scheichl, A. Teckentrup, "Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients." *Computing and Visualization in Science* vol 14(1), pp. 3–15, 2011.

[3] M. Giles, "Multilevel monte carlo path simulation", *Operations Research* vol. 56(3), pp. 607–617, 2008.

[4] A.-L. Haji-Ali, F. Nobile, L. Tamellini, R. Tempone, "Multi-index stochastic collocation for random PDEs," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 95 – 122, 2016.

[5] A.-L. Haji-Ali, F. Nobile, R. Tempone, "Multi-index Monte Carlo: when sparsity meets sampling," *Numerische Mathematik*, pp. 1–40, 2015.

[6] J.D. Jakeman, M. Eldred, G. Geraci, A. Gorodetsky, "Adaptive multi-index collocation for uncertainty quantification and sensitivity analysis," *International Journal for Numerical Methods in Engineering*, 2019.

[7] F. Kuo, D. Nuyens, "Application of quasi-monte carlo methods to elliptic pdes with random diffusion coefficients: A survey of analysis and implementation," *Foundations of Computational Mathematics*, vol. 16(6), pp. 1631–1696, 2016.

[8] F. Kuo, C. Schwab, I. Sloan, "Multi-level Quasi-Monte Carlo Finite Element Methods for a Class of Elliptic PDEs with Random Coefficients," *Foundations of Computational Mathematics*, vol.15(2), pp. 411–449, 2015.

[9] F. Nobile, L. Tamellini, R. Tempone, "Convergence of quasi-optimal sparse-grid approximation of Hilbert-space-valued functions: application to random elliptic PDEs," *Numerische Mathematik* vol. 134(2), pp. 343–388, 2016.

[10] C. Schillings, C. Schwab, "Sparse, adaptive Smolyak quadratures for Bayesian inverse problems," *Inverse Problems* vol. 29(6), 2013.

[11] A. Teckentrup, P. Jantsch, C. Webster, M. Gunzburger, "A Multilevel Stochastic Collocation Method for Partial Differential Equations with Random Input Data," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 3(1), pp. 1046–1074, 2015.