On the predictivity of pore-scale simulations: estimating uncertainties with multilevel Monte Carlo

Matteo Icardi[a,c,d,*], Gianluca Boccardo[b], Raúl Tempone[a]

[a] *CEMSE, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia*
[b] *DISAT, Politecnico di Torino, Torino, Italy*
[c] *ICES, The University of Texas at Austin, USA*
[d] *Mathematics Institute, University of Warwick, UK*

Abstract

A fast method with tunable accuracy is proposed to estimate errors and uncertainties in pore-scale and Digital Rock Physics (DRP) problems. The overall predictivity of these studies can be, in fact, hindered by many factors including sample heterogeneity, computational and imaging limitations, model inadequacy and not perfectly known physical parameters. The typical objective of pore-scale studies is the estimation of macroscopic effective parameters such as permeability, effective diffusivity and hydrodynamic dispersion. However, these are often non-deterministic quantities (i.e., results obtained for specific pore-scale sample and setup are not totally reproducible by another "equivalent" sample and setup). The stochastic nature can arise due to the multi-scale heterogeneity, the computational and experimental limitations in considering large samples, and the complexity of the physical models. These approximations, in fact, introduce an error that, being dependent on a large number of complex factors, can be modeled as random. We propose a general simulation tool, based on multilevel Monte Carlo, that can reduce drastically the computational cost needed for computing accurate statistics of effective parameters and other quantities of interest, under any of these random errors. This is, to our knowledge, the first attempt to include Uncertainty Quantification (UQ) in pore-scale physics and simulation. The method can also provide estimates of the discretization error and it is tested on three-dimensional transport problems in heterogeneous materials, where the sampling procedure is done by generation algorithms able to reproduce realistic consolidated and unconsolidated random sphere and ellipsoid packings and arrangements. A totally automatic workflow is developed in an open-source code [1], that include rigid body physics and random packing algorithms, unstructured mesh discretization, finite volume solvers, extrapolation and post-processing techniques. The proposed method can be efficiently used in many porous media applications for problems such as stochastic homogenization/upscaling, propagation of uncertainty from microscopic fluid and rock properties to macro-scale parameters, robust estimation of Representative Elementary Volume size for arbitrary physics.

[*] matteo.icardi@warwick.ac.uk

Keywords: pore-scale simulation, multilevel Monte Carlo, stochastic upscaling, uncertainty quantification
2010 MSC: 76S05, 68T37, 60H15, 65C05

1. Introduction

Simulations and experiments at the pore-scale (field now also known as Digital Rock Physics, DRP) have become an important tool to understand the complex physics involved in environmental and industrial processes such as Enhanced Oil Recovery, Carbon Dioxide Storage, transport of charges in batteries and fuel cells, filtration of colloidal particles in subsurface and biological flows. In these and other fields, a big effort has been recently spent in trying to prove the predictive ability of pore-scale simulations, as alternative or complementary to experiments. While these have been proven to be effective in understanding and giving deeper insight in complex mechanisms, such as capillary trapping or anomalous transport, when more precise quantitative information such as global effective parameters have to be extracted, they can still be affected by significant errors and uncertainties [2, 3, 4, 5, 6].

First of all, real porous media show heterogeneities on a wide (or continuum) range of scales. Therefore, even in the assumption that large-scale variations can be explicitly represented and solved (provided that enough information about this variability is available), the role of sample size and sample location (or realization in the stochastic terminology) deeply affects the meaning of computing a single effective parameter, and of using it for upscaled models. If we assume that the heterogeneity of the medium shows a clear separation of scales, one may be able to find a single deterministic effective parameter (e.g., permeability. See, for example, Mostaghimi et al. [7]), by taking larger and larger sample size[1]. However, practically, three situations may happen: (i) the scale separation hypothesis does not hold, (ii) results on sufficiently large samples are not available (for computational, imaging, or experimental limitations), or (iii) the quantity of interest cannot be homogenized (i.e., it may be, a local quantity or a quantity involving local derivatives). In all these situations, the resulting computation of effective parameters is clearly dependent on the sample chosen and therefore, can be conveniently represented as a stochastic process.

Another very important source of error, in the case of computer simulations of pore-scale images, is the lack of knowledge in the detailed micro-scale structures (e.g., shape and roughness of the grains). Accurate imaging techniques can partially overcome this problem but segmentation uncertainties and discretization procedures (e.g., voxels or mesh decompositions) often reintroduce a high degree of uncertainty. Some of these inaccuracies are controlled by arbitrary user-defined parameters that make most of the pore-scale studies unreliable or non-reproducible. A last source of error, often overlooked, is related to insufficient resolution of the numerical simulations and to other numerical artifacts due to the discretization of the Partial Differential Equations (PDEs). If some of these errors can be

[1] It has to be noted however that the classical concept of Representative Elementary Volume (REV) is not universal but strongly depends on the desired quantity of interest (or effective parameter) under study.

negligible for certain parameters (e.g., small roughness does not affect significantly global absolute permeability), the same can have dramatically effects on others (e.g. roughness, and hence surface area, becomes crucial for surface reactions).

In this work we propose a general and efficient way to quantify the effects of statistical, parametric, and numerical uncertainties, applicable to complex simulations such as the ones encountered in pore-scale physics. Our approach, based on multilevel Monte Carlo sampling, gives a full statistical description of the desired quantities of interest (e.g., effective parameters) at a computational cost that is of the order of a few runs with a finely discretized mesh whereas many more simulation runs (depending on the variability of the quantity under study) would be necessary with the classic Monte Carlo approach. Multilevel Monte Carlo (MLMC)[8] shares the same flexibility and robustness with the classic Monte Carlo sampling but it has a much wider applicability thanks to the drastically reduced computational cost. The key idea of the method is to take advantage of the numerical and statistical properties of under-resolved cheaper simulations that are used to "precondition" the fine-scale results. MLMC has been already proposed for stochastic homogenization in the context of elliptic partial differential equations (e.g. Darcy or heat equation) with random heterogeneous diffusion coefficients [9] (for a primer about stochastic homogenization we refer to the recent review of Alexanderian [10]). Other attempts to speed up computations in stochastic homogenization have been proposed by Blanc et al. [11], by using variance reduction techniques (the same techniques at the basis of MLMC). Here we apply some of these ideas in a different context (i.e., pore-scale simulations) and we propose the method as a general uncertainty quantification tool for different sources of errors and uncertainties. In this work, the statistical analysis of porous media, made of spherical and irregular grains, is performed adaptively and automatically, starting with given statistical description of the random geometry realizations (e.g., fixing the grain/pore size distribution and the packing algorithm), geometric properties, fluid or rock parameters. The method has been implemented in a modular way to make use of existing open-source software as black-box solvers and it has been made available online [1].

The aim of this work is to show that modern uncertainty quantification and error analysis techniques can be successfully applied in pore-scale physics, providing important insights on the reproducibility and representativity of the many recent results in the field (we refer to our recent work and other reviews [12, 13, 14, 15] and references therein for a list of the important advances and results in the field).The main novelties of this work are:

- Uncertainty quantification is for the first time applied to pore-scale simulations
- A description of the role of numerical and statistical errors associated with pore-scale simulations is considered and a mathod to quantify the role of each them is proposed
- A fully automated algorithm to generate, solve and analyze a large number of pore-scale problems is implemented

The manuscript is organized as follows: first the models of random porous media and random materials and generation algorithms are introduced. Then, in Section 3, the MLMC technique is introduced and proposed, for the first time, for studying the effect of random pore-scale geometries. The physical models and the respective transport and flow solvers used in this work are presented in Section 4. The last section is instead devoted to the presentation of the results obtained for three simplified test cases and a general discussion about their interpretation in the intent of addressing the general problem of the reliability and predictivity of the state-of-the-art DRP.

2. Random porous media

The choice of the geometric model to be used in the pore-scale simulations (and most importantly how to describe and represent it) is crucial to obtain realistic macro-scale correlations. In particular, when upscaled quantities and correlations are to be found, by introducing appropriate variations (randomness) in the geometry, more informative results can be obtained about the predictivity and uncertainty in the macro-scale models. While a general way to reproduce the complex variations present in natural rocks is not available, it is often possible to reproduce synthetic granular materials, given some global (mean) information and some physical constraints. Once a certain generation procedure is assumed to be representative of the real variations present in a porous medium, mean and variance of the observed effective parameters can play an important role in developing new more predictive (often stochastic) macro-scale models. It is important to notice here that, in the limit of large sample sizes, no variabilities is expected in averaged properties. However, often one is interested or limited in a finite-size sample analysis. This is the case, for example of micro-CT images or for limited computational resources. Furthermore, heterogeneities typically appear in a wide range of scales so that practically, the concept of Representative Elementary Volume does not apply. In this case, a statistical description (e.g., statistical moments)

In this work we will consider different types of porous media, differentiating mainly between random *arrangements* and random *packings*. The former describes a geometric model in which the different grains of the porous medium are arranged in a specific enclosing volume, and whose locations are usually obtained by means of a random placement procedure, avoiding or allowing a certain amount of overlap between them. These random arrangements can be created from simple purely algorithmic procedures (though sometimes they can be quite costly) and do not have a physical correspondence to reality, whereas porous media are usually the result of a sedimentation process, meaning each grain will obviously possess a set of contact points with other neighbouring grains. However, when many simulations have to be run, random arrangements allow for a much faster and extensive prototyping of different structures, and thus allow more detailed studies on a wide range of parameters (e.g.: porosity, tortuosity). In fact, with this same methodology, it is possible to simulate consolidated[2] porous media models, simply by creating very dense (read, low-porosity) arrangements and allowing a small amount of grain-grain overlap in the algorithm.

Random packings instead, as mentioned, are the result of a settling process and as such, a pseudo-physical algorithm is needed to recreate the settling/deposition process. As a matter of fact, a choice most frequently made is to use actual porous media models, obtained experimentally through a variety of imaging techniques such as X-ray micro-computer tomography or scanning electron microscope scans. Two issues are to be evidenced with this approach. While it is undoubtedly possible in principle to obtain very precise and realistic

[2] For clarity, *unconsolidated* porous media will have “clean” contact points between the grains, where *consolidated* porous media will have additional volume in the contact area between the grains, i.e.: they will be partly glued together. This happens in the case of cementification of the grains due to solid matter sedimentation, for example.

models, technological limitations on image discretization often introduce a relevant source of uncertainty. Moreover, many geometric properties which are of great influence in the subsequent fluid dynamic analysis [3] are difficult or generally cumbersome to extract. Secondly, these methods would be ill-suited to be paired with the uncertainty quantification study proposed in this work, which requires a way of producing and analyzing a very high number of realizations of porous media models in order to extract relevant statistics. A more aptly chosen technique is then indeed to create these random packings approximate simulation tools in order to create a fully *in-silico* simulation package which can take care of both the construction of the geometric model and its analysis.

In the next sections details on the generation algorithms, both for the case of random arrangements and random packings, are described. Both methods have been developed to use generic grain shapes, defined either by analytical equations (spheres, icosahedra, ellipsoids) or by irregular closed surfaces (described by watertight triangle meshes). The results presented in this work, however, will focus on simple examples with spheres and ellipsoids only, to demonstrate the applicability of the method.

2.1. Random arrangements

Random arrangements are created by randomly picking size and shape of the grains from pre-defined grain size and grain shape distributions. Each grain is then placed randomly in the domain. To generate non-overlapping random loose packings, the placements that lead to overlapping are withdrawn and the placement algorithm continues until all the desired grain are placed. This simple algorithm can work only for very high porosity and results in non realistic disconnected porous matrix. However the study of these geometries can be relevant for other problems such as solid heterogeneous materials or multiphase dispersed flows. To recreate more realistic granular and consolidated porous media, many algorithms have been proposed (see [18, 19] and references therein). In this work an extended version of the Jodrey-Tory algorithm [20] has been used, that consists in a post-processing iterative greedy-type moves to displace overlapping or detached grains. Though the property (e.g. "entropy") of the arrangement will depend on the details of the implementation (such as the displacement length), the result is a random arrangement with a desired minimum and maximum degree of overlapping between grains.

2.2. Random packings

As mentioned, random packings are created by reproducing the process of settling and deposition, that is, to simulate the effect of gravity on a collection of grains drawn from a certain grain size distribution. Also in this case the resulted geometry is "random", in the sense that is the result of complex interactions that depends on the initial conditions and the subsequent collisions. In general the randomness is introduced by the initial position of the grains, their initial velocity and by additional grain manipulation that can happen before or after the settling.

[3] To cite a few examples, in the case of packed bed catalytic reactors: radial porosity profiles, distribution of angles of the particles and the individuation of their contact points [16, 17].

The implementation of this packing algorithm has been done within the open-source software Blender [21]. More specifically, the functionality of the Bullet Physics Library (integrated into Blender) was exploited. The BPL is a large collection of codes used to simulate rigid body dynamics and more importantly, to calculate the outcome of the collisions between rigid bodies. A number of reasons brought to this choice. The first is a computational aspect, as the Blender rigid body simulation does not consider the interaction between the rigid bodies and the surrounding fluid, making the code particularly suitable for fast and real-time simulations. In order to correctly reproduce a random packing, in fact, this is not strictly needed[4] and would just add a heavy computational overhead to the simulation. Being able to omit this calculation makes for a great improvement over other methods such as those based on Discrete Element Methods (DEM), commonly used in the simulation of granular flows.

Secondly, while other packing codes (such as those based on molecular dynamics) only offer the possibility of considering spherical objects (i.e.: spherical interaction potentials) or convex items at most [22], with Blender it is possible to manage any arbitrarily defined particle shape, even complex non-convex ones, which are the ones of most difficult treatment in rigid body simulations. As already mentioned, in this work only results concerning packing of spheres (like the one represented in Figure 1) will be presented for the sake of brevity, and mostly to serve as a supporting proof of concept for the wide applicability of the general MLMC method. Nonetheless, this methodology has also been tested with the creation of packings of arbitrary grain shape. Both in the latter case and for regular spheres, an accurate validation procedure has been carried out to ensure that the resulting geometric structure shown the same features as the real media to be reproduced. This has been done through the comparison of bulk porosity and wall porosity (in the case of constrained packings) of the in-silico packings with extensive experimental data available in literature. Moreover, the fluid dynamic behaviour of the created packing was analyzed and compared with consolidated empirical laws for the systems considered (e.g.: comparison of fluid pressure drop obtained from pore-scale simulations with Ergun's law predictions). Much more information on the creation of this packings, and the validation procedure just described, can be found in our previous work [23].

Finally, the choice of Blender was also made due to its extensive scripting functionality. In order to connect the rigid body simulation to the code structure of the MLMC estimator, a comprehensive tool was written (in the language Python, version 3.0) allowing for an easy user specification of the test case and a fully automatic set up of the Blender simulation. Thus, the workflow of the code as it is now implemented [1] only requires the user to specify the details of the chosen grain size distribution, the number of grains to be drawn from this distribution, a container in which the solid grains will settle, and an arbitrary (convex or non-convex) grain shape (with eventually any random modifiers such as random roughness

[4] For example, in the creation of lab-scale column experiments, a container is filled with solid grains while there is just air inside, which does not influence the packing process in any way due to the great density difference between solid and fluid.

or random consolidation processes).

Both the generation algorithms (random packing and random arrangements) have been then integrated with OpenFOAM solvers and wrapped into a python automatic solver that is used by the multilevel Monte Carlo algorithm. More details about the code are presented in the appendices.

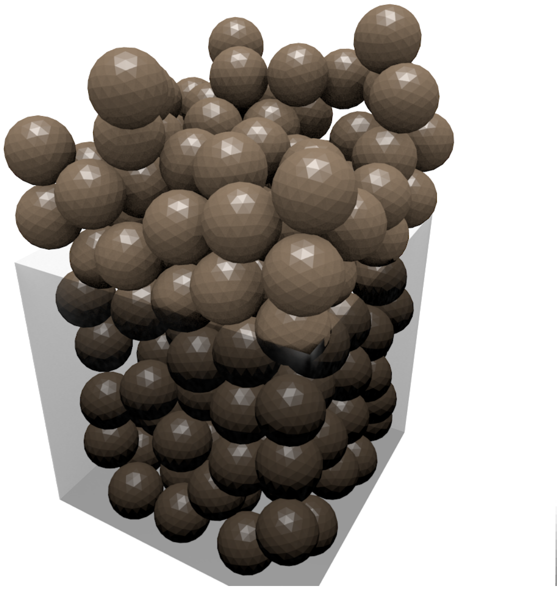

Figure 1: Example of a random packing of sphere generated by a virtual settling process of spheres in a cubical container. This type of packings are studied in Section 5.3

3. Multilevel Monte Carlo

Multi-level Monte Carlo (MLMC) is based on the very simple idea of decomposing the final quantity of interest Q into the sum of an initial approximation and a series of incremental corrections, i.e.

$$Q \approx Q_0 + \sum_{\ell=1}^{L} Q_\ell - Q_{\ell-1}$$

where the subscripts indicate successive numerical approximations (e.g., coming from simulations with increasing levels of mesh refinement) with a certain convergence property towards the exact solution. In the usual approach, only the last “level” L would be taken into account, while here having expressed it as a telescopic sum will play a crucial role in the statistical approximation. In fact, if we assume now that Q is a random variable $Q = Q(\omega)$ dependent on a random event ω (omitted from now on to simplify the notation), we can exploit the linearity of the statistical moments and write, for example, its expected value as

$$\mathrm{E}[Q] \approx \mathrm{E}[Q_L] = \mathrm{E}[Q_0] + \sum_{\ell=1}^{L} \mathrm{E}[Q_\ell - Q_{\ell-1}]$$

If we now replace each expectation with its respective MC estimate, we obtain the MLMC estimator

$$\mathcal{A}_{\mathcal{MLMC}} = \frac{1}{M_0} \sum_{m=1}^{M_0} Q_0 + \sum_{\ell=1}^{L} \frac{1}{M_\ell} \sum_{m=1}^{M_\ell} Q_\ell - Q_{\ell-1} \tag{1}$$

instead of the classical MC estimator (computed with a single level L correspondent to the finest solution and M samples):

$$\mathcal{A}_{\mathcal{MC}} = \frac{1}{M} \sum_{m=1}^{M} Q_L \ , \tag{2}$$

where M_0 and M_ℓ are the number of samples taken in the first and each subsequent level, respectively. MLMC can be used not only for computing the mean (expected) value but also higher order moments or full probability distribution. In our examples we limit ourselves to the estimation of mean and variance of the quantity of interests. In Appendix A.1 some details about the estimation of the variance and higher-order moments are reported.

The biggest advantage of this simple formulation is that, under very mild assumptions, it can drastically reduce the variance of the estimator (compared to the standard Monte Carlo done on the last level only) in a “control variate” spirit[5]. While we refer to the many theoretical works (e.g., [24, 8, 25]) for detailed proofs and derivation, here it is important to mention the asymptotical result for which MLMC, not only reduces the cost of the classical MC estimator by a constant factor, but it can change the overall computational complexity

[5] Readers may notice that, compared to the classical control variate technique, here there is no scaling constant appearing in the differences. This is important to preserve the telescopic property of the sum (e.g., cancellation of all terms but the last one). Furthermore, in MLMC, the control variate technique is applied recursively.

of the problem, canceling the effect of the computational cost of the single realizations. This means that the computational gain of MLMC compared to MC exponentially increases for increasing desired accuracies (and therefore increasing cost of the single realizations). Therefore MLMC is particularly suited for computationally expensive problems where common MC sampling can easily become unaffordable. This can be intuitively explained by the fact that (asymptotically) the cost of the estimator will be distributed mostly on the lowest levels that are computationally negligible, while the fine levels are solved only for a very small number of samples.

To achieve this optimality, the number of levels and the number of samples used have to be chosen (usually on-the-fly) according to given formula based on expected values, variances, and costs of each term in the expansion. Roughly speaking, the only assumptions requested are[6]

$$\alpha \geq \min(\beta/2, \gamma); \qquad \beta > \gamma$$

where α, β, γ are respectively:

- α is the weak convergence rate, e.g., the rate with which the expected value of the numerical solutions $\mathrm{E}[Q_\ell]$ converges to the exact solution $\mathrm{E}[Q]$ with respect to a certain discretization parameter (usually minimum grid size). This is, for most problems, simply the deterministic convergence rate imposed by the numerical discretization[7]. This property is important if we want to have a control on the numerical discretization error (that can be interpreted as statistical bias in the estimator 1). For example, if we assume a simple linear convergence rate $\alpha = 1$, if $Q_\ell - Q_{\ell-1} < C$, the remaining error $Q - Q_\ell$ will be also smaller than C.

- the multilevel variance rate β is related to a very important, but less intuitive, concept. For the MLMC estimator to work, the corrections $Q_\ell - Q_{\ell-1}$ (differences) on the different levels should have variance $\mathcal{V}\left(Q_\ell - Q_{\ell-1}\right)$ smaller than the original variance $\mathcal{V}\left(Q_0\right)$ and it should decay with a rate β with respect to the discretization parameter. This means that, the solution couples on two subsequent levels should be more and more statistically correlated, when increasing the levels. The same quantity $\mathcal{V}\left(Q_\ell - Q_{\ell-1}\right)$ is also needed to compute a sampling approximation the total variance of the estimator $\sum_{\ell=1}^{L} \mathcal{V}\left(Q_\ell - Q_{\ell-1}\right)$ and therefore its correspondent statistical error. It can be easily noticed that, since this variance is decreasing with the levels with a rate β, one may use large number of samples at the coarsest level to counterbalance the largest contribution to the total variance while much less samples are needed in the finer levels.

- γ is simply the growth rate of the cost of solving a single realization problem with respect to the discretization parameter.

[6] If the second condition does not hold, MLMC can still be used with very good improvements over standard MC but the computational complexity will not be independent from the cost of the single realization anymore.

[7] This is not true when the convergence properties of the numerical discretization are, in some sense, dependent on the random realization in a non linear way.

These three rates are commonly measured in a $\log_2$ basis since, when considering grid resolution studies, grid is commonly refined of a factor two in each levels. For practical and realistic problems however, these rates are not known exactly and sometimes are not constant. In this work we will monitor numerically the rate of these quantities (even when their rates are not constant) to build the optimal estimator (see [25] for details) and we finally evaluate the computational gain of MLMC compared to MC for each test-case. It is important to notice that MLMC can work (i.e., it can reduce the total cost of computing statistics of the desired QoI), though in a non-optimal way, even when the above relations between the rates are not satisfied.

3.1. Algorithm

1. Start assuming an initial guess for the number of levels L and the number of samples M_ℓ in each level $\ell = 0, \dots, L$
2. For each level ℓ and for each random realization $(1, \dots, M_\ell)$, generate the geometry and solve the PDE problem with two discretization levels ℓ and $\ell - 1$ to obtain Q_ℓ and $Q_{\ell-1}$
3. Compute the multilevel mean $\mathrm{E}[Q_\ell - Q_{\ell-1}]$ and variance $\mathcal{V}\left(Q_\ell - Q_{\ell-1}\right)$ at each level
4. Update the number of samples in each level to optimize the cost and reduce the statistical error below the desired tolerance (see, for example, [25]). Compute the additional samples needed in case M_ℓ has been increased for any level.
5. Compute the statistical error (e.g., with the Central Limit Theorem provided a confidence level ϵ) and the bias (numerical) error estimate
6. If the bias is above the prescribed tolerance, set $L = L + 1$ and a first guess for the number of samples in the new level M_L and go back to step 2
7. Otherwise assemble the estimator Eq. 1 and compute the desired statistics with the associated error estimates

In the above algorithm we mentioned that, together with the desired tolerance, a prescribed confidence should be provided to the algorithm, needed to compute the statistical error. In the following results the confidence ϵ is always set to 0.99 meaning that the tolerance is imposed with a probability 0.99.

3.2. Generalized notion of "level"

Two aspects of MLMC have important consequences on the implementation and the extension to general problems. Firstly, the notion of "levels" is totally arbitrary. One can easily develop a MLMC strategy with any kind of hierarchical representation/discretization of the problem. The choice of grid resolution is natural for certain PDE problems as it can be studied with standard numerical analysis tools. For more complex or non standard problems any other discretization or approximation parameter may be used to generate a hierarchy. The intuitive requirement is only an increasing cost and accuracy with the discretization parameter. Very interesting theoretical results and applications seem to arise when more than one discretization parameter is alternately varied in a multi-dimensional hierarchy, as proposed by Haji-Ali et al. [26]. This will be subject of future works. In the results presented here the grid resolution remains the main parameter used but it is always coupled to other parameters to further reduce the cost of coarse samples but keeping the correlation between levels controlled. In particular, when solving coarse levels, the whole accuracy and precision of the numerical representation is reduced by decreasing the number of linear and non-linear solver iterations, the tolerances of the pressure-velocity coupling in the Navier-Stokes solver, the geometrical tolerances to generate the geometry and the grid, the precision of the algorithm to generate random packings. It is important to notice that it has to be checked that, coarsening all these parameters contemporarily, a high correlation between the samples at two consecutive levels is preserved (through the analysis of the rate β).

This brings us to the second important aspect that relates to the physical meaning of the coarsening strategy. MLMC is built in a way (thanks to the telescopic sum) such that it is not required that the coarse representations have any physical meaning or numerical consistency. Even in the case where, at coarse levels, the solution looses physical meaning or consistency (due to, for example, missed physical constraint), it is enough to check that the finest level only is consistent and physically meaningful. In the context of UQ, all the other levels can be though as “proxies” or “surrogate” models (that still maintain the same stochastic nature, though). In the following examples, additional considerations about the hierarchy and the discretization parameters will be given for each specific case.

4. Numerical solvers

Among the many physical phenomena of interest at the pore-scale in subsurface and industrial flows, in this first work we focus on the influence effective diffusivity (or equivalently heat or electrical conductivity[8]) and permeability (or equivalent drag force or hydraulic conductivity). Future works will deal with the upscaling of multiphase and reactive flows.

The solvers for flow and transport used in this work are modified versions of the ones offered within the open-source OpenFOAM software. They are all based on a finite-volume discretization for use in unstructured meshes. We refer to Boccardo et al. [27], Icardi et al. [15] for more details about the grid generation and the solution of the Navier-Stokes equations. It is important to highlight here that all the pre- and post-processing steps (including geometry generation) are totally automatized to be able to be run multiple times in the MLMC estimators.

Let us consider a domain $\Omega = [0,1]^3 \setminus B$ where B is the volume occupied by solid grains[9]. To compute the effective diffusivity we solve in Ω the following elliptic PDE for a scalar concentration c

$$\nabla \cdot (D_0 \nabla c) = 0 \tag{3}$$

with $D_0 = 1$ the bulk diffusivity and $c = 1$ at the inlet, $c = 0$ at the outlet, $\mathbf{n} \cdot \nabla_c = 0$ on the surface of the grains, Γ, and on the lateral boundaries. Being the total volume $V = \int_\Omega dV = 1$, $\Delta x = 1$ and $\Delta c = 1$, the first column of effective diffusion coefficient can be computed as an integral over the all domain

$$\mathbb{D}_1 = \frac{D_0}{V} \frac{\Delta x}{\Delta c} \int_\Omega \phi \nabla c = \int_\Omega \phi \nabla c \tag{4}$$

with ϕ being the porosity. Being this quantity a vector (and the effective diffusion coefficient a tensor, when we apply a concentration gradient in the other directions) we only consider the longitudinal direction $D = \mathbb{D}_{11} = \int_\Omega \phi \frac{\partial c}{\partial x}$.

The effective permeability is instead computed by solving the incompressible Navier-Stokes equations in the void space with no-slip boundary conditions on the grains, fixed pressure drop between inlet and outlet faces, and symmetry (no flow) condition on the lateral boundaries. All the simulations have been performed in a dimensionless setup with unitary imposed pressure gradient ΔP, viscosity μ and density ρ. This is equivalent to a Reynolds number dependent purely on the length scale considered. Under the assumption of fluid in the Stokes regime (thus, Darcy's law being valid) and with the described setup the first column of the permeability can be simply computed as

$$\mathbb{K}_1 = \frac{\mu}{V} \frac{\Delta x}{\Delta P} \int_\Omega \phi \mathbf{u} = \int_\Omega \phi \mathbf{u} \tag{5}$$

[8] The ratio between electrical conductivity of the porous material and the bulk conductivity is also called the "formation factor"

[9] The operator \ indicates a complement operation between two sets. Thus Ω is the fluid zone.

with $\mathbf{u}$ being the fluid velocity. Again, we are only interested in the first component $K = \mathbb{K}_{11} = \int_\Omega \phi u_x$.

where u_x is the $x-$component of the fluid velocity. When studying random porous materials, the estimation of geometric parameters are often quantities of interest if they are not known a-priori or by a constraint in the random geometry generation. In particular, when a specific porosity is not imposed, it has to be considered as an output of the problem. Another interesting property is the specific surface area of the porous media defined as the ratio between the surface area, Γ, and the fluid volume, B. This is particularly important for the derivation of upscaled models that involves surface reactions.

4.1. Deterministic convergence

The solvers have been tested against semi-analytical and highly accurate values obtained for regular sphere arrangemets in Khirevich et al. [28], Venema et al. [29] and references therein. An effective diffusion coefficient and the effective permeability have been computed from the solution of pure diffusion and Navier-Stokes equations. Different types of grids and refinements have been used and compared. Figure 2 reports the normalized value and the relative error on the effective diffusion and drag coefficients, respectively in semi-log (left) and double logarithmic (right) scales. Even if no clear convergence rate can be extrapolated, the convergence towards the reference solution is verified. Significant differences can be noticed for different grid refinements around the spheres. In all the cases it is very interesting to notice that there is always a non-monotone convergence to the reference solution (i.e., the error changes sign) as also clearly shown by Khirevich et al. [28]. This seems to be an effect of the discretization of complex curved surfaces. The slowest convergence provided by the voxelized (stair-step) meshes is counterbalanced by a clear linear convergence for highly enough resolutions, while body-fitted meshes can show more unpredictable convergence rates.

It is assumed that the convergence properties of the solver for every realization of the random geometries follow a similar behaviour. This allows us extend these arguments to the statistical estimation of the mean (expected) value $\mathrm{E}[Q]$. Therefore we can estimate the numerical approximation error and stop the refinement process at a certain level L that satisfy certain criteria. In this work a simple (and restrictive) stopping criterion has been implemented that consists in checking that the last difference

$$\mathrm{E}[Q_L - Q_{L-1}] < \theta \ \mathrm{TOL}$$

is under the prescribed tolerance. θ is a constant parameter (between 0 and 1) that is used to split the bias and statistical error. This criterion however may fail when a non-monotone convergence is observed, with the approximated solution that cross the exact value at a certain refinement level and then converge again towards it with a different rate. This appears as a flat or positive (growth) rate in the decay of the differences $Q_\ell - Q_{\ell-1}$. This unfortunately happens in many practical problems when two sources of errors (e.g., bulk and boundary errors) are compensating each other. To avoid this false convergence, for every problem we obtain solutions with a very large number of levels for a single realization of the

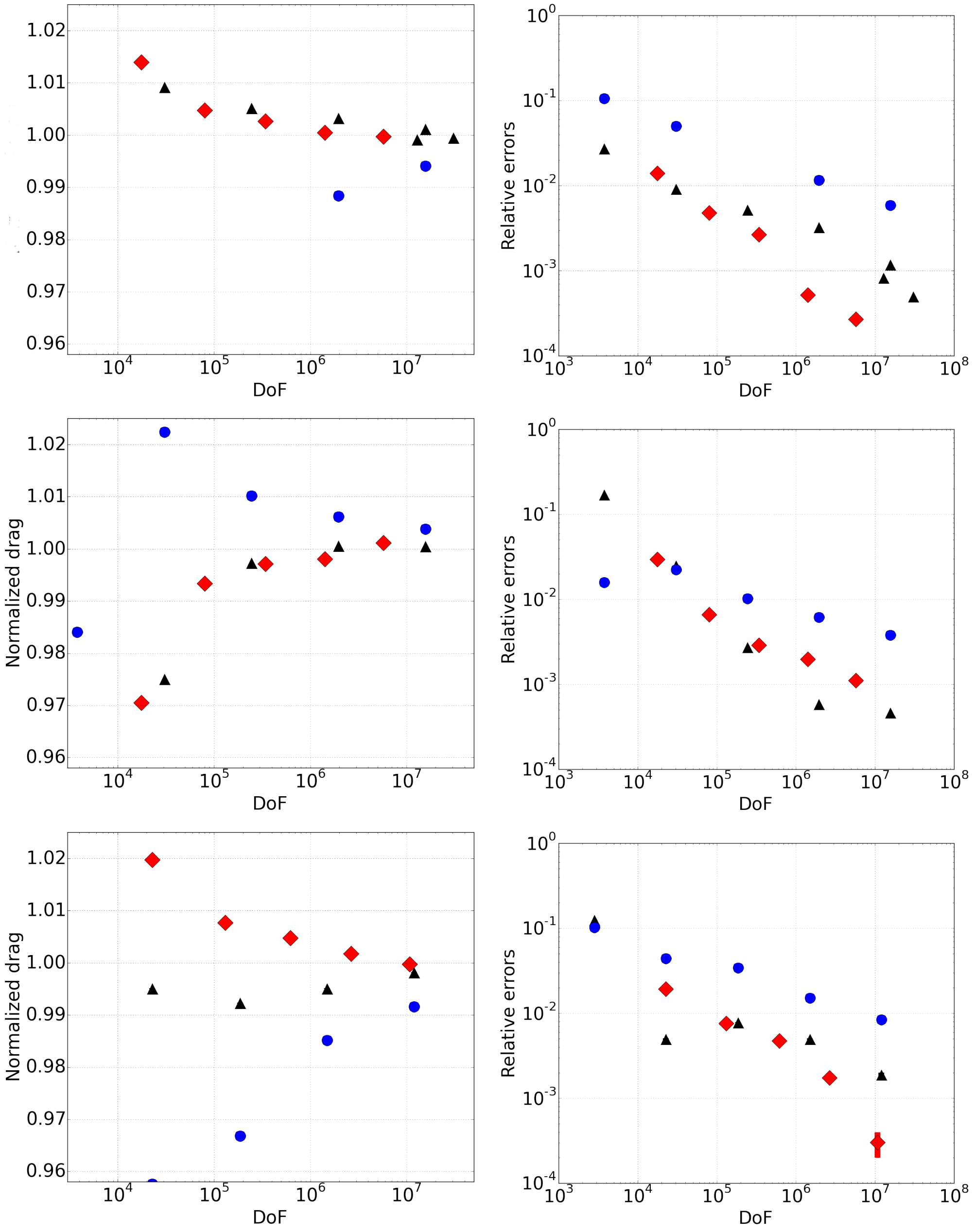

Figure 2: Convergence of the finite-volume discretization to the references values, reported as relative errors in linear (left) and logarithmic (right) scale, for effective diffusion (top) and mean flux (middle and bottom) on regular sphere arrangements. Values are reported against increasing degrees of freedom (i.e.: mesh cell numbers) on the $x-$axis. Black triangles and red diamonds represents respectively uniformly refined and locally refined body-fitted mesh, while blue circles represent stair-step voxelized meshes.

random geometry. Whenever we observe a crossing point (a resolution at which the errors are compensating each other producing a fictitious exact solution), we inform the MLMC algorithm to perform an extra check by using the differences in adjacent levels.

More sophisticated and accurate stopping criteria can be implemented based on a better and robust statistical estimation of the convergence rate [30] or on a-priori knowledge of the convergence rates. Future development will focus on the improvement of these methods in order to give better estimates of the approximation error.

4.2. Aitken's delta-squared method

In this work, we are focusing on stationary solutions and on specific quantities of interest rather than all scalar or vector fields for which the CFD code calculates a solution. This is what commonly happens in many applications where there is no need of complex data analysis or exploration and the quantities to be computed are identified a-priori. In this situation we can represent the non-linear iterations of the solver and the intermediate results as elements of converging series. Therefore we have implemented some series acceleration methods to extrapolate the limit of the series [31]. The most suitable for our purposes turned out to be the Aitken delta-squared method, that consists in the following non-linear transformation

$$A(x_n) = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n} \tag{6}$$

where, x_n is the converging sequence and Δ and Δ^2 are the first and second order differences. The formula however is not valid for constant series or constant increments where the second order difference is zero. However this is not the case for typical convergent series (in particular the one arising from the convergence of non-linear iterations).

This turns out to be particularly important for large simulations. In fact the finer the resolution is, the more iterations are needed to solve, for example, the pressure-velocity coupling in the Navier-Stokes equation. For most of the test problems presented in the following sections, the use of this transformation allowed us to stop the simulation at much larger residuals (e.g., 10^{-4} instead of 10^{-8}) and save considerable amount of computational time.

5. Examples

Three illustrative examples are shown in this section to cover some of the possible applications of the proposed method and codes. Particular attention has been put in the computational efficiencies that will allow more complex applications in future works.

- The first example is the estimation of the formation factor (effective conductivity or diffusion) of high-porosity media at fixed sample size. For this application an extensive study has been done by varying the input tolerance of the algorithm.
- In the second example the permeability of consolidated sphere arrangements have been studied at different sample sizes with a fixed tolerance
- The third example makes use of more realistic unconsolidated sphere packings and their permeability and porosity

Together with the transport properties, also geometrical quantities (such as porosity and surface area) are considered as random quantities of interest. We remind that all the problems have been solved in a completely dimensionless framework. Therefore effective diffusivity and permeability will have no physical units. All the simulations have been run on a 20-cores Intel Xeon E5-2680 v2 (2.80GHz) workstation. A large amount of memory (192 GB) is required to run 40 parallel processes, each one solving a CFD problem. As explained in the appendix, further optimization has been implemented for hybrid parallelization on supercomputers and memory savings.

5.1. Effective diffusivity of heterogeneous materials

In this example we solve the pure diffusion problem of Eq. 3 with Neumann homogeneous boundary conditions at the grain walls and lateral boundaries and Dirichlet boundary conditions at the inlet (as detailed in the previous section), with an unitary diffusion coefficient in an unit box domain with random non-overlapping ellipsoidal inclusions. The ellipsoids are placed in the central part of the domain and oriented along the cartesian axes and their axes lengths are distributed along a lognormal distribution of mean $\mu = 0.1$ and $\sigma = 0.4\mu$. The placement algorithm stops when the total volume of placed ellipsoids exceeds an average density (in the packed region inside the box) equal to 0.3. Figure 3 shows two realizations of the geometry with the solution field computed on the surfaces. In this example the ellipsoid are always oriented in the axis directions but a general random transformation matrix (through random orthogonal matrices) have been also implemented to deal with arbitrary orientations. As it can be seen, a 'voxelized', staircase discretization has been used to speed up the meshing time.

The effective diffusion coefficient of the material can be computed from the concentration field c with Eq. 4, as well as other discrete geometrical quantities can be computed after the meshing step. The statistical estimation of the surface area, porosity and effective diffusion is therefore performed by the MLMC algorithm. Four different tolerances for the effective diffusivity (from 1 to 10%) has been imposed to the algorithm that is run three times for each

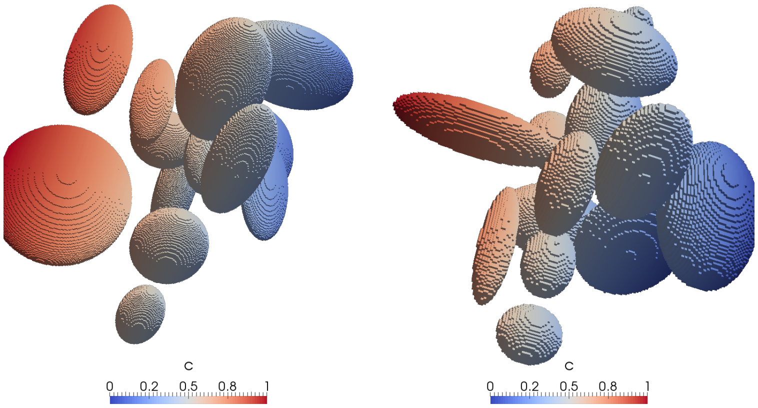

Figure 3: Two realizations of the random problem described in 5.1 with refinement levels $\ell = 4$ (left) and $\ell = 3$(right). The grains, colored by the concentration field at their surface, are visible. The domain Ω is their complement.

of them, for total of 12 runs of the MLMC algorithm. The number of samples and number of refinement levels is adaptively chosen to satisfy the imposed tolerance. The concept of "level" in this testcase is defined by the grid resolution (using a uniform grid). However, when coarse grids are used, also lower solver tolerances and less solver iterations (not to be confused with the overall MLMC tolerance, these tolerances are related only to the finite volume solver) are used. The optimal tuning of how these parameters should be relaxed for coarse levels is a non trivial optimization problem. In our examples, preliminary studies have been run to roughly determine how the cost and the correlation between samples at different levels depend on each parameter, and choose the solver tolerances at each level as a power law on the level. As explained in Section 3.2, this helps in reducing the cost of coarse levels, keeping sufficiently high the correlation between successive levels. The idea is to balance all sources of errors so that, at each level, they are all approximately of the same magnitude. This is in practice very hard to optimize for a generic problem where different discretization parameters act in a combined way, and no theoretical convergence results can be applied.

When the algorithm is run with the largest tolerance (10%), it ends up choosing only three levels with about 20 realizations at the coarsest level and two in the finest one. For the finest tolerance (1%) five levels are needed with about 700 coarse solutions and only one fine solution. However, to compute good statistics the number of realizations per level is forced to be always larger than four. This compromises the optimality condition and the total cost but ensures robustness. Due to this constraint the overall computational savings compared to Monte Carlo at the same tolerance is in all the cases only between 2× and 50×, where better savings are obtained for smaller tolerances. The comparison with standard Monte Carlo is performed considering a sampling in the finest level with a number of samples determined to have the same statistical error of the MLMC estimator. However this comparison is strongly biased in favor of MC because, for simplicity, it is assumed that (i) the variance of the QoI is the same at each level so that we can use the variance estimate provided by the coarsest level of MLMC that has a large number of samples; (ii) no robustness constraint (forcing the number of samples to be larger than a minimum) is implemented while the cost of MLMC includes this extra cost.

The typical analysis to be done on each MLMC run is presented in Figure 4. As explained in Section 3, MLMC is based on the estimation of differences for the same random realizations on two different levels. The convergence of the mean (solid black line) and variance (dashed-dotted blue line) of these differences[10] for higher levels is verified as well as the computational cost (dotted red line) for each level that, as expected, is growing exponentially. A fourth dashed green curve representing the distribution of the statistical errors among the different levels (their sum is equal to the final statistical error of the estimator), is also shown. This clearly shows the main idea behind MLMC: balancing costly simulations that have small variance and are sampled with few samples, with many samples of cheap simulations to "pre-condition" the problem so that the total error will be split approximately equally on the different levels. The final result of the estimator is given by the cumulative sum (integral) of the curve $\mathrm{E}[Q_\ell - Q_{\ell-1}]$.

This analysis is reported for each of the three quantities of interest, verifying the correct implementation of the MLMC estimator and confirms, as expected, the applicability of the method (i.e., the blue dotted line representing the multilevel variance is generally decaying very fast). Being each point a statistical estimation of the mean and variances of differences between two levels, it can be provided also with error bars. Compared to the typical problems studied in MLMC (such as the diffusion equation with random coefficients) the random geometry, the possible non-linearity of the quantities of interest and the complexity of the discretization method can cause the mean, variance and cost rates to be non-constant. However it can be recognized, in agreement with the deterministic convergence properties shown in Section 4, a linear convergence rate for the mean diffusion coefficient and a second-order convergence for the porosity. These rates are visible also in the behavior of the variance. While it seems more complex the behavior of the specific surface area estimation.

As it can be seen in Figure 5, considering the different errors bars available in each estimation, all the runs agree on a mean porosity (considering also the empty region close to the walls of the box) of around 0.924, surface area per unit volume of fluid of 3.6, and effective diffusion of 0.876 [11]. As it can be noticed, all the points fall always inside the error bars of the other estimators with the same and different tolerance. This is a first consistency check that can be done when no exact solutions are available. The total CPU time needed for a MLMC estimation with 1% tolerance was about 1h with 40 parallel processes while the cost of a single fine realization requires slightly less than 1h on a single core.

[10] The value correspondent to level zero is not a difference but simply the mean and variance at the coarsest level

[11] We remind the reader that these are all normalized and dimensionless quantities

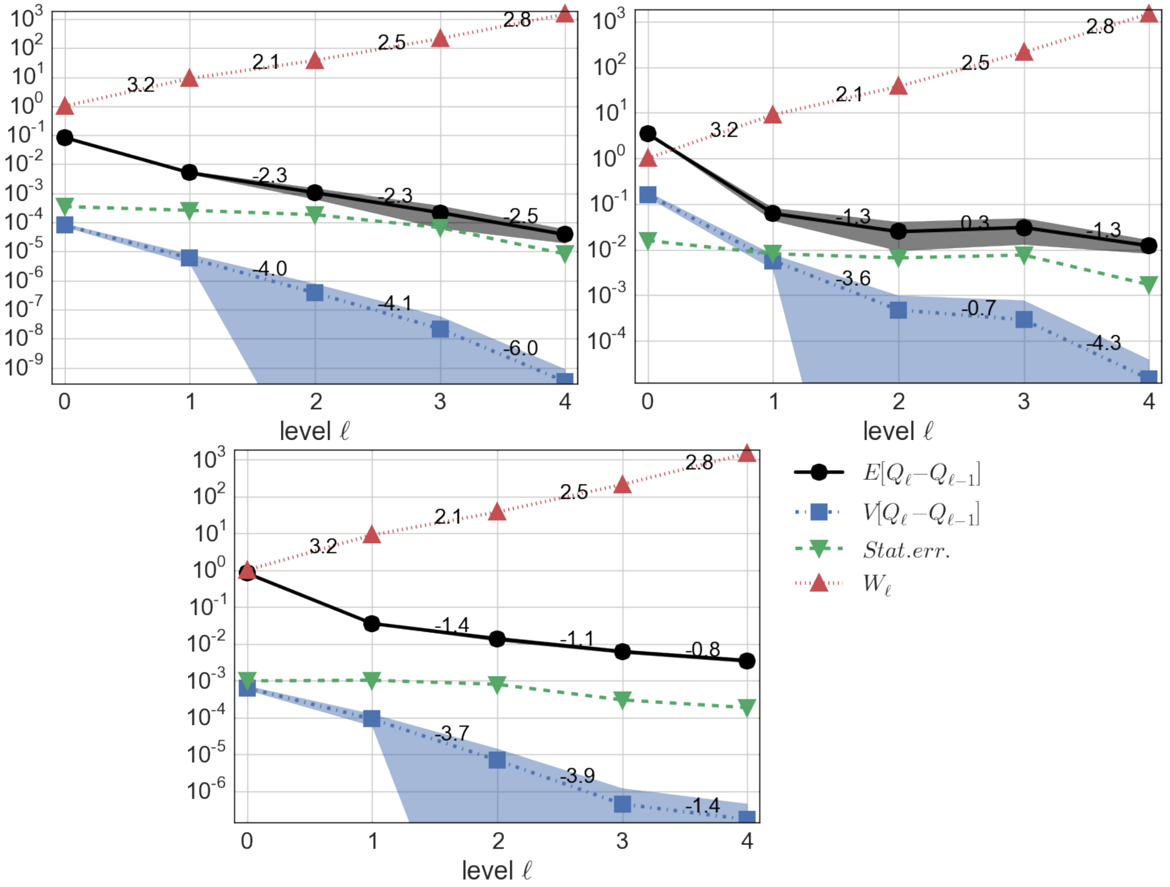

Figure 4: MLMC convergence rates for (i) porosity, (ii) surface area, and (iii) diffusivity estimators for the problem of Section 5.1. Computational cost (dotted red line with upper triangles), mean (solid black line with circles) and variance (dashed-dotted line with squares) of the differences, distribution of the statistical error among the different levels (dashed green with downward triangles). The black numbers between two points represent the local rates α, β, γ respectively for work, mean and variance.

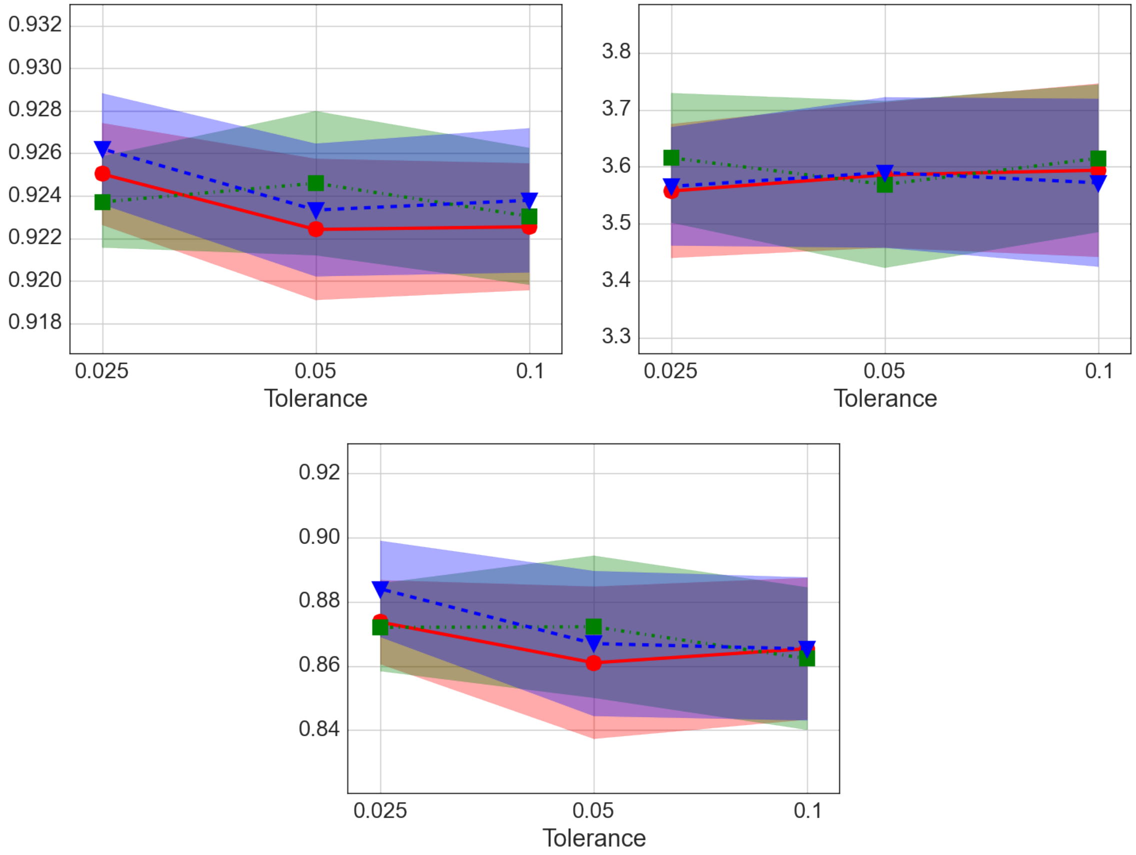

Figure 5: Three runs of MLMC estimators (represented with different colors) for three different tolerances. The mean quantities are shown with error bars given by bias and statistical error estimates.

5.2. Permeability of consolidated sphere arrangements

This examples deals with the simulation of incompressible fluid flow in consolidated sphere arrangement. Here we simply assume a random placement of spheres followed by a Jodrey-Tory algorithm to ensure that all the spheres don't overlap for more than 80% of their radius. As before, the sphere size is drawn from a lognormal distribution with mean $\mu = 0.1$ and $\sigma = 0.4\mu$. The packing is stopped when the density exceeds 0.5. Two random realizations with their discretization on different levels are shown in Fig. 6

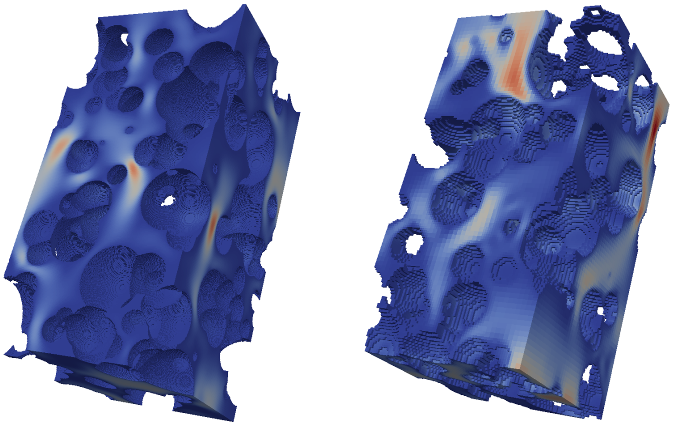

Figure 6: Two realizations of the random problem described in 5.2 with refinement levels $\ell = 4$ (left) and $\ell = 3$(right). The fluid domain is here represented, colored by the velocity magnitude, with holes representing the grains.

First a study for a fixed domain 2 × 1 × 1 is performed to test the MLMC algorithm than a study on different sizes (starting from the domain 2 × 1 × 1, keeping the aspect ratio and doubling the total volume in each step) is performed computing the sample mean and the sample variance of the effective permeability[12]. Also in this case the discretization parameters involved are not only related to the grid resolution (chosen again to be uniform) but also to the Navier-Stokes solver tolerances and the geometric tolerances to generate the mesh.

Figure 7 shows, for the case with fixed domain size, the typical behavior of MLMC with a second order convergence for the mean (this can be explained by the contemporary grid refinement and tolerance decrease) and a faster convergence for the variance (approximately with a rate twice as faster as the mean, as expected) and an increasing cost. In this problem a 1% tolerance has been set on the mean permeability and the MLMC algorithm stopped

[12] To avoid confusion, it is important to notice that explicitly inserting the permeability variance in the list of QoI (i.e., defining a prescribed tolerance) has nothing to do with multilevel variances that are always computed inside the MLMC algorithm. Computing the permeability (as well as any variance) in the MLMC estimator implicitly involves fourth order moments (the variances of the variance estimator). The different approaches to compute the permeability variance are reported in the Appendix.

after computing about 12 thousands realizations at level zero, and respectively 721, 63, 7, and 2 realizations at the subsequent levels. As before the last level has been increased to 5 to increase robustness of the mean and variance estimates. However it is important to highlight that only 2 fine realizations would have been enough to compute the mean permeability with 1% accuracy. The total cost of the MLMC estimator is of 60 hours (splitter over 40 processes) and the cost of a single fine realization is of 10 hours. The overall computational savings is in this case over 400× compared to MC.

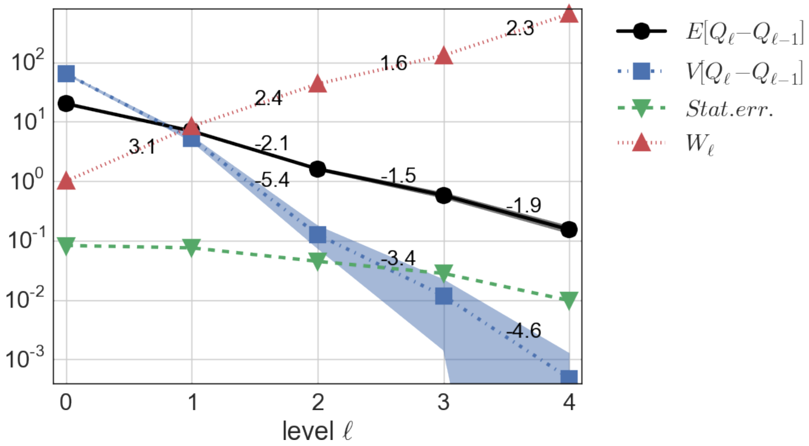

Figure 7: MLMC convergence rates for the estimation of effective permeability in random consolidated sphere arrangements of Section 5.2. Computational cost (dotted red line with upper triangles), mean (solid black line with circles) and variance (dashed-dotted line with squares) of the differences, distribution of the statistical error among the different levels (dashed green with downward triangles).The black numbers between two points represent the local rates α, β, γ respectively for work, mean and variance.

What can be expected from the second setup with increasing domains sizes, is a decrease of the permeability variance with increasing volume size and a contemporary convergence of the mean permeability towards of constant value. This means that, for the quantity of interest under study (permeability in this case) there exist an homogenization limit. This is the standard study commonly performed to determine the REV size for upscaling. It is important to notice that this is not true in general for other quantities of interest of the flow. In this first work only this property is studied, but major advantages are expected for our MLMC method in the case of non-homogenizing quantities (where therefore large variations can be observed). Furthermore, all the variability observed here is purely due (like in the previous cases) to a finite-size sample and would tend to zero in the limit of large sample. This allows us to verify the consistency of our approach and illustrate a possible use of the method to control the finite-sample errors and optimize resources. However, the method might give more interesting insights when heterogeneities at larger scales are introduced in the geometry generation. This would require a well-defined description of a specific material that is beyond the scope of this work.

Figure 8 confirms this expected behavior. Two MLMC realizations (one red solid line with circles and the other dotted green line with squares) have been performed for each of the four different domain sizes. For this problem a 5% tolerance has been used for the mean permeability while a much larger tolerance (30%) has to be used for the variance. In fact, the computation of the variance is, in general, much harder than the mean. Generally three levels are enough to compute permeability mean and variance for smaller domains. In that case, in fact, being the tolerance relative, a larger error, particularly in the variance, can be admitted. When larger domain sizes are considered, the computation of the mean becomes relatively easy because there is less variation. However, computing the variance with the same relative tolerance is harder and may require more samples and levels.

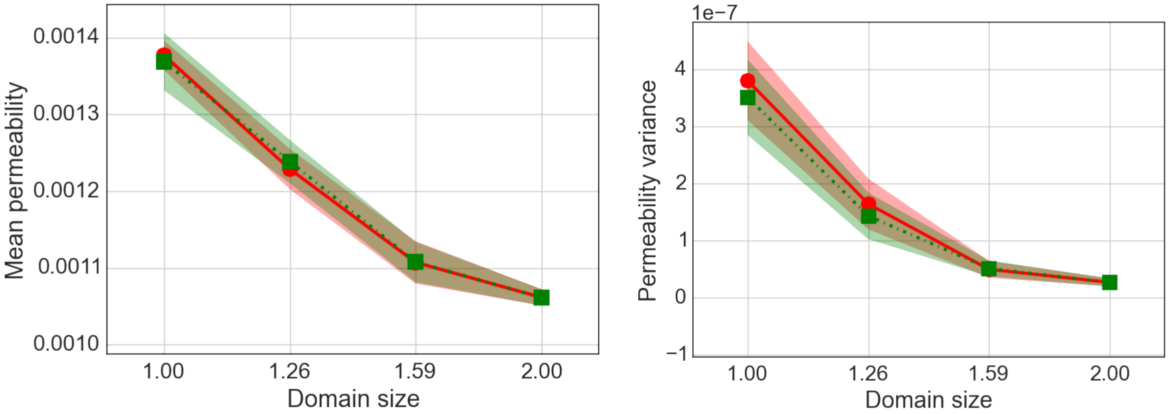

Figure 8: Two runs of MLMC estimators (represented with different colors) for four different domain sizes. The mean permeability (left) and permeability variance (right) are shown with error bars given by bias and statistical error estimates.

The overall computational savings of these estimators is between 6× and 90× compared to a single-level MC estimator at the finest level, depending on the realization and on the domain size. As it has been previously noticed, this comparison is very conservative. The total cost of this variance study is about 2 days with 38 parallel processes.

5.3. Permeability of realistic sphere packing

The third and last example is related to more physically realistic (though still simplified) materials, given by the packing algorithm described in Section 2. We started by considering the experiments performed by Moroni and Cushman [32], where a cube filled with spherical particles is used to study dispersion properties. For these types of experiments where there is no direct control on the geometry, it is important to assess the reproducibiity of the results, by studying the possible variations (randomness) introduced by the random filling of the container. Here our objective is therefore to compute mean porosity, mean permeability and their variances, mimicking the same type of packing of this experiment, but reproduced arbitrary number of times in a random way. However it turned out that even a single accurate simulations of this packing would require a relatively high number of parallel processes. Furthermore, as it is expected, the variation in permeability for such homogeneous sphere packing is very small when the size of the box is large enough (here it is 16× the particle diameter). This means that the we are in a situation where only (relatively) few samples are needed but with a very high computational cost. Though this is a common situation that the proposed method is able to address (provided that supercomputing resources are available), this turned out not to be a good illustrative example for the method. Therefore, to increase the variations of the observed permeability we reduced the domain size to 5× the particle diameter. The statistical analysis of dispersion and other multiphase transport properties on the original geometries [32] will be presented in our future works. An example of the reduced geometry studied here is instead represented in Figure 1.

In this case, a non uniform refinement has been used to reduce the cost and potentially make use of a larger number of levels. Voxelized meshes are again used to reduce the meshing cost. In this case, in fact, the pre-processing (geometry generation and meshing) take a fixed and significant amount of time, no matter the discretization level used in the flow simulation. This will generate a fixed overhead cost that is generally not taken into account in the classical MLMC theory. Trying to apply the multilevel idea also on the packing algorithm seems natural but its application is cumbersome. In fact, the packing algorithm (as well as any type of molecular dynamics) can be considered as “chaotic”, in the sense that the final results can drastically change on small perturbations of the initial, operating or discretization conditions. A requisite of MLMC is that the same realization has to be computed on two levels and a natural concept of levels for the random packing algorithm is the surface resolution of the grains. However a different representation of the grains lead to different collision patterns and finally to completely different packings (due to the chaotic property of the dynamics). In the MLMC framework this means that the solutions (whether permeability or other quantities) on the two levels are not correlated and the variance of the difference does not decay. Selecting the right discretization parameters that might keep a correlation between the samples is a complicated task (it has to consider both the system dynamics and the specific quantity of interest) and will be subject of further studies. In this work a multilevel discretization of the packing algorithm has been applied only for computing geometrical quantities after the settling (such as porosity, surface area, etc). However the multilevel implementation of the flow solver, that is the most expensive part, can still give a good accuracy to MLMC studies. A rescaling of the grains (by a factor 0.9) has also been applied to simplify the solution on coarse grids and avoid too many blocked pores caused by under resolution. This rescaling is set in an adaptive way depending on the level (e.g., $1 - 3^{-\ell}(1 - \eta)$) so that the converged solution will tend to the unscaled one.

The simulation has been run with a tolerance on 3% on the permeability and no tolerance on porosity and on the variances. Due to the effects of the wall the resulting mean porosity is about 54% while the internal porosity is about 40%. The mean permeability is found to be about $4 \cdot 10^{-4}$. While no significant variations of porosity are observed among the samples (less than 1%), the variations on the permeability are of the order of 10%[13]. This is one more simple demonstration that REV (and upscaling sample size) are to be defined depending on the quantity of interest under study. The results of the MLMC estimators for mean and variance of permeability, porosity and surface area are reported in Figure 9. The permeability has been rescaled by a factor 1000 to avoid numerical cancellation effects in its variance. Four levels have been found sufficient to reach the desired tolerance with about 1000 samples on the coarse level and 9 in the finest one. The total computational cost is of about 4 hours with 40 parallel processes and the overall computational savings compared to MC is, in this case, about 10× but, as before, a much larger savings are gained when lower tolerances are used or when a larger variability is present in the random geometry.

[13] Since only a tolerance for the mean permeability has been set, the variances (and therefore the percentage standard deviation reported) have a large error bar of about 150% that is however not so significant when considering a variance. This means that the actual variations of porosity might be between 0.66% and 1.5% and the permeability fluctuations between 6% and 15%.

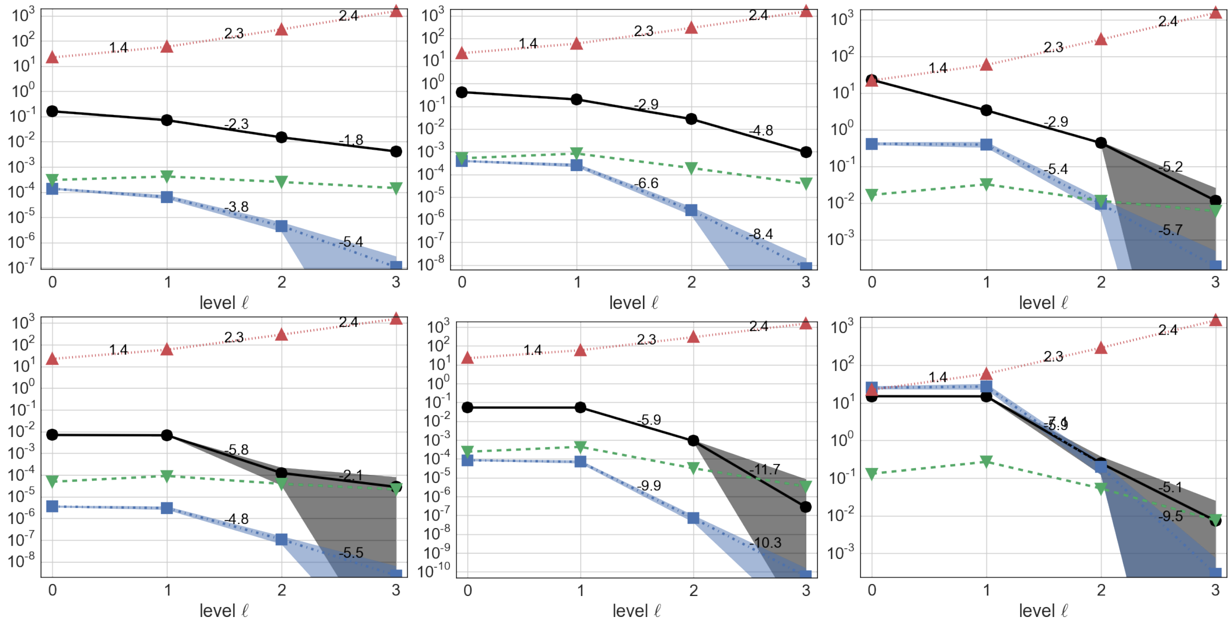

Figure 9: MLMC convergence rates for the estimation of effective permeability (left), porosity (center) and surface area (right) in random sphere packings of Section 5.3. Computation of mean (top) and variance (bottom). Computational cost (dotted red line with upper triangles), mean (solid black line with circles) and variance (dashed-dotted line with squares) of the differences, distribution of the statistical error among the different levels (dashed green with downward triangles).The black numbers between two points represent the local rates α, β, γ respectively for work, mean and variance.

6. Conclusions

We have proposed a general method to estimate numerical, parametric and sampling errors in pore-scale simulations, based on multilevel Monte Carlo (MLMC). This is achieved by a tolerance-based dynamic estimation of statistics (e.g., mean and variance) of effective parameters of porous media samples. MLMC can explicitly control that the bias (i.e., discretization error) and the statistical error (i.e., the error due to the randomness of the samples) fall below a certain user-defined tolerance.

The method has been tested for the analysis of the variabilities in averaged geometric properties and effective permeability and diffusivity with finite sample size but it is applicable to a wide range of problems, as far as a parametrization (explicit or implicit) of the input uncertainty is available. This includes, for example, pre-defined parameter probability distributions (e.g., for fluid or surface properties such as surface tension and contact angle in multiphase flows), or pre-defined procedures to generate random materials that are comparable to the real material under study. The verification of this requirement (that the random parameters or geometry is an actual realizable event in the real system under study) is crucial to the final interpretation and meaning of the statistical estimation performed by our method. For example, to study the effect of large-scale heterogeneity, detailed information about statistical distribution and correlation function of the input random parameters are neeeded.This will be addressed in future works from an inverse problem perspective. This can make use of the same multilevel structure and Monte Carlo estimation, with the recently proposed Markov Chain Multilevel Monte Carlo [33]. In this context, for example, pore-scale geometrical properties could be inferred from available experimental data.

Considering that the scalability of MLMC (similarly to classical Monte Carlo) comes at almost no cost (by parallelizing on the numbers of fine and coarse realizations, while the scalability of single pore-scale flow simulations is often limited to hundreds of cores), this demonstrates the high potential and feasibility of this uncertainty studies. The parallelization can also be hybridized with fine realizations solved on a number of cores dependent on the refinement levels and multiple realizations solved simultaneously. As such, with this method is possible to take full advantage of the ever growing availability of HPC resources.

A possible limitation of the method is due to the fact that MLMC (as well as other methods used here, like the Aitken extrapolation) is practically applicable only for studying scalar or low-dimensional vector quantities. This limitation is the price to be paid for having detailed statistical information and it is a general aspect of many UQ studies that, by definition, focus on quantitative study of specific quantities of interest (QoI) and need a preliminary well-defined parametrization and definition of the problem. After the numerous works in recent literature in pore-scale physics devoted to the description and definition of computational (or experimental) procedure to compute specific quantities (e.g., effective parameters or more complex quantities such as capillary pressure curves or description of other non-linear effects), it is our opinion that these standard upscaling procedures (to pass from an infinite dimensional representation to a few upscaled representative quantities) have to be benchmarked, validated and applied in a systematic way for a wide range of pore geometries and physical regimes to understand their robustness, predictivity, and general applicability.

Within this objective, the proposed approach is, to our knowledge, the first practically applicable method for general Uncertainty Quantification (UQ) studies of problems arising in Digital Rock Physics, where the cost of a naïve sensitivity analysis is still prohibitive. To some extent and with a few precautions, the method can also be applied for a faster and cheaper estimation of parameters on experimental images, when a high number of samples can be extracted at different resolutions but only a limited amount of them can be performed in high-resolution.

The method has been implemented [1] on top of existing open-source software and is to be released open-source together with the full setup of the simulations presented in this work. Future efforts will be directed towards the computation of the full Probability Density Functions (PDF) of effective parameters, following the recent work of Giles et al. [34], and the coupling with other solvers to include more physics (e.g. multiphase and reactive flows) and more type of random materials (fractures and explicit heterogeneity definitions), extending the range of possible analysis to more complex and non-linear properties such as relative permeabilities, effective reaction rates, filtration efficiencies, just to name a few. One of the most interesting and straightforward extension that will be considered is the estimation of hydrodynamical dispersion parameter. Though in principle equivalent to the effective parameters studied here, some theoretical and implementation issues have to be addressed. For example, this requires the solution of two separate equations (flow and transport) and each of them will have different convergence properties, making the optimization of the method more challenging. A unified error analysis (such as the one proposed by Charrier [35]) should be first considered. Furthermore, the study of physically more important (and complex) parameters would require an adequate characterization of the input variabilities and uncertainties to be able to obtain physically meaningful results (e.g. the random geometry generation process should be tuned to reproduce specific porous media).

7. Acknowledgments

Research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST). MI and RT were supported by the Academic Excellency Alliance (AEA) UT Austin-KAUST project "Uncertainty quantification for predictive modeling of the dissolution of porous and fractured media" and by the KAUST SRI Center for Uncertainty Quantification in Computational Science and Engineering. Computational resources for the simulations presented in this publication have been made available by the Texas Advanced Computing Center, KAUST Research Computing and KAUST SuperComputing Lab. MI thanks Siarhei Khirevich for providing the reference results for regular sphere arrangements and for his comments and discussions. GB thanks Rajandrea Sethi and Daniele Marchisio for their support.

Appendix A. Multilevel Monte Carlo details

Appendix A.1.

Three possible way of estimating variances, standard deviations and central higher order moments have been implemented:

- *Single-pass estimation.* For simulations where many quantities of interests and a very large number of realization are needed (or for communication limitations due to parallelization), one may want to estimate both the mean and the variance at one time. In this case, the variance $\mathrm{Var}(Q)$ can be rewritten as $\mathrm{E}[Q^2] - \mathrm{E}[Q]^2$ and the two moments can be estimated separately (by defining a QoI vector (Q, Q^2). In this case, the error estimation performed on the QoI is no more valid for the variance. However it is easy to proof a new error bound for the variance

  $$e_{VAR} \leq e_2 + 2e_1(\mathrm{E}[Q] + e_1)$$

  where e_1 represents the error in the mean, e_2 the error in the second order moment and the terms where the error appears in power larger than one can be neglected. If one is interested in the standard deviation, the Taylor expansion (valid for small error e) $\sqrt{\mathrm{Var}(Q) + e_{VAR}} \approx \sqrt{\mathrm{Var}(Q)} + \frac{e_{VAR}}{2\sqrt{\mathrm{Var}(qoi)+e_{VAR}}}$ can be performed to link the error in the variance to the error in the standard deviation. Or a more rigorous and stringent bound can be derived as

  $$e_{STD} \leq \frac{e_{VAR}}{\widetilde{\mathrm{Var}(Q)}}$$

  where the tilde operator denotes the estimated variance, provided that it preserves positivity of the square root. It is not possible however to implement directly this method in the MLMC adaptive algorithm that requires error estimates at each level.

- *Two-pass estimation.* A very similar approach is to first compute an estimation of the first order moment (mean) and then modify all the realizations by shifting them against the computed mean and computing the second order (central) moments. In this case, assuming that the mean value estimator is independent from the single realization (true only for large number of samples), the standard MLMC adaptive algorithm can be used to adaptively choose number of levels and samples for a certain tolerance. A drawback is that the algorithm must store all the realizations and this may create memory storage issues. Furthermore, since the mean used to shift the data has a certain error, an additional bias term appears. This is however practically very small in practical situations where one is interested in having a very precise mean and a rough estimate of the variance, setting therefore a very low tolerance for the mean and a much larger one for the variance.

- The most appropriate way to compute the variance, as shown by Bierig and Chernov [36] is instead to compute, at each level, the sample variance and compute the differences between the two estimators at the different levels. Compared to the previous method, instead of using a single mean (given by the MLMC estimator), a mean at each level is computed and subtracted to each result at that level. The memory requirement are similar (but only 2 levels at a time have to be stored) but no bias is introduced and the same adaptive algorithm can be used.

Appendix B. Flowchart, implementation and parallelization details

Schematic charts of the code [1] are presented in Figures B.10-B.11. In the first figure, the overall MLMC code is explained. As it can be seen the code is split in three main parts: the statistical estimation, the geometry creation and the solver. Each of them can refer to different sub-modules to be able to approach different problems effectively. For example different PDEs can be solved by finite volumes with OpenFOAM or by finite elements with Fenics or GetDP. Geometries can be built either with random arrangements (*randomgeo*) or with random packing (*bsand*). Finally statistical study can be done with MLMC, with standard MC or with deterministic ad-hoc sampling strategies. The process of solving a single realization is instead schematized in Figure B.11. This includes three main steps (preprocessing, packing algorithm, solution), performed respectively by a Python code (called BSand) and its wrappers and by OpenFOAM. Each step is characterized by various possible randomness or uncertainties. The purely deterministic steps are written in italic letters and are only related to the analysis and the finite volume solver while many parameters can possibly be set as random or uncertain in all the other steps.

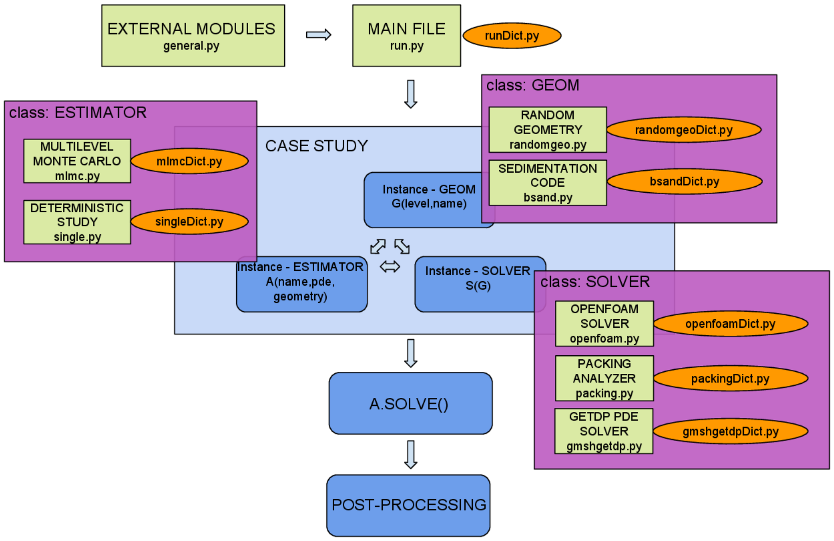
EXTERNAL MODULES
general.py
MAIN FILE
run.py
runDict.py
class: GEOM
RANDOM GEOMETRY randomgeo.py
randomgeoDict.py
SEDIMENTATION CODE bsand.py
bsandDict.py
class: ESTIMATOR
MULTILEVEL MONTE CARLO mlmc.py
mlmcDict.py
DETERMINISTIC STUDY single.py
singleDict.py
CASE STUDY
Instance - GEOM G(level,name)
Instance - ESTIMATOR A(name,pde, geometry)
Instance - SOLVER S(G)
class: SOLVER
OPENFOAM SOLVER openfoam.py
openfoamDict.py
PACKING ANALYZER packing.py
packingDict.py
GETDP PDE SOLVER gmshgetdp.py
gmshgetdpDict.py
A.SOLVE()
POST-PROCESSING

Figure B.10: Structure of the code.

Since three-dimensional flow problems can easily take a few days of CPU time at a reasonably good resolution, an appropriate parallelization strategy is needed to address complex applications. Monte Carlo simulations are inherently embarrassingly parallel since each sample is independent from the other. However two difficulties appears. First the multilevel estimator adaptively compute the multilevel variances and mean so that a certain communication between the processors must be preserved, and the number of samples to run cannot be known in advance. For this reason the code has been first parallelized with

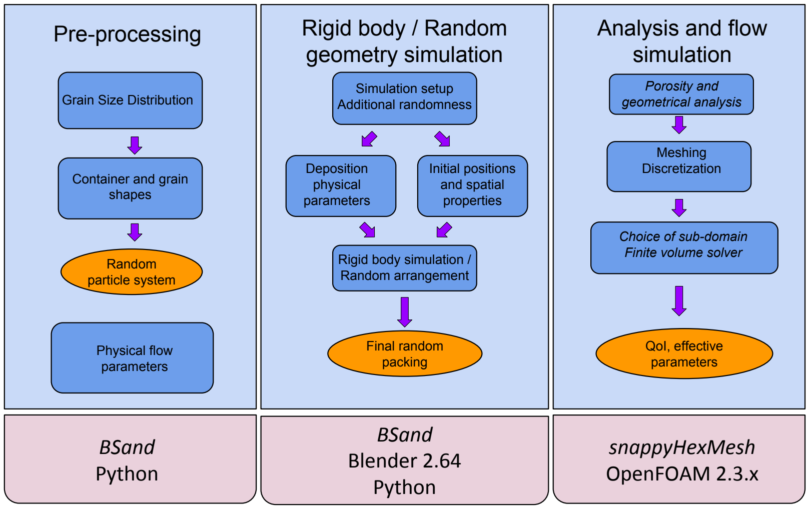
Pre-processing
Grain Size Distribution
Container and grain shapes
Random particle system
Physical flow parameters
Rigid body / Random geometry simulation
Simulation setup Additional randomness
Deposition physical parameters
Initial positions and spatial properties
Rigid body simulation / Random arrangement
Final random packing
Analysis and flow simulation
Porosity and geometrical analysis
Meshing Discretization
Choice of sub-domain Finite volume solver
QoI, effective parameters
BSand
Python
BSand
Blender 2.64
Python
snappyHexMesh
OpenFOAM 2.3.x

Figure B.11: Structure of the code.

dynamic parallel data structures (*Queues*) using the module *rpyc* to handle the communication between nodes and between processes. A second difficulty is that finer simulations need to be parallelized themselves. This second level of parallelism has been implemented using the intrinsic capabilities of the forward solvers (either OpenFOAM or other solvers), with standard *OpenMPI* and domain decomposition techniques.

References

[1] 2015;URL: `https://bitbucket.org/micardi/porescalemc`.

[2] Hesse F, Radu F, Thullner M, Attinger S. Upscaling of the advection-diffusion-reaction equation with monod reaction. Advances in Water Resources 2009;32(8):1336 –51.

[3] Neuman SP, Tartakovsky DM. Perspective on theories of non-fickian transport in heterogeneous media. Advances in Water Resources 2009;32(5):670 –80. Dispersion in Porous Media.

[4] Battiato I, Tartakovsky D. Applicability regimes for macroscopic models of reactive transport in porous media. Journal of Contaminant Hydrology 2011;120–121(0):18 –26. Reactive Transport in the Subsurface: Mixing, Spreading and Reaction in Heterogeneous Media.

[5] Battiato I, Tartakovsky DM, Tartakovsky AM, Scheibe T. Hybrid models of reactive transport in porous and fractured media. Advances in Water Resources 2011;34(9):1140 –50. New Computational Methods and Software Tools.

[6] Horgue P, Augier F, Duru P, Prat M, Quintard M. Experimental and numerical study of two-phase flows in arrays of cylinders. Chemical Engineering Science 2013;102(0):335 –45.

[7] Mostaghimi P, Blunt MJ, Bijeljic B. Computations of absolute permeability on micro-ct images. Mathematical Geosciences 2013;45(1):103–25.

[8] Giles MB. Multilevel monte carlo path simulation. Operations Research 2008;56(3):607–17.

[9] Efendiev Y, Kronsbein C, Legoll F. Multi-level monte carlo approaches for numerical homogenization. arXiv preprint arXiv:13012798 2013;.

[10] Alexanderian A. A primer on homogenization of elliptic pdes with stationary and ergodic random coefficient functions. arXiv preprint arXiv:14085827 2014;.

[11] Blanc X, Bris CL, Legoll F. Some variance reduction methods for numerical stochastic homogenization. arXiv preprint arXiv:150902389 2015;.

[12] Andra H, Combaret N, Dvorkin J, Glatt E, Han J, Kabel M, et al. Digital rock physics benchmarks part i: Imaging and segmentation. Computers & Geosciences 2013;50:25–32.

[13] Andra H, Combaret N, Dvorkin J, Glatt E, Han J, Kabel M, et al. Digital rock physics benchmarks part ii: Computing effective properties. Computers & Geosciences 2013;50:33–43.

[14] Blunt MJ, Bijeljic B, Dong H, Gharbi O, Iglauer S, Mostaghimi P, et al. Pore-scale imaging and modelling. Advances in Water Resources 2013;51:197–216.

[15] Icardi M, Boccardo G, Marchisio D, Tosco TAE, Sethi R. Pore-scale simulation of fuid flow and solute dispersion in three-dimensional porous media. Physical review E 2014;.

[16] Augier F, Idoux F, Delenne J. Numerical simulations of transfer and transport properties inside packed beds of spherical particles. Chemical Engineering Science 2010;65(3):1055 –64.

[17] Caulkin R, Jia X, Xu C, Fairweather M, Williams RA, Stitt H, et al. Simulations of structures in packed columns and validation by x-ray tomography. Industrial & Engineering Chemistry Research 2009;48(1):202–13.

[18] Baranau V, Tallarek U. Random-close packing limits for monodisperse and polydisperse hard spheres. Soft matter 2014;10(21):3826–41.

[19] Torquato S. Random heterogeneous materials: microstructure and macroscopic properties; vol. 16. Springer; 2002.

[20] Jodrey W, Tory E. Computer simulation of isotropic, homogeneous, dense random packing of equal spheres. Powder Technology 1981;30(2):111–8.

[21] Blender - a 3D modelling and rendering package. Blender Foundation; Blender Institute, Amsterdam; 2015. URL: `http://www.blender.org`.

[22] Donev A, Torquato S, Stillinger FH. Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles. i. algorithmic details. J Comput Phys 2005;202(2):737–64.

[23] Boccardo G, Augier F, Haroun Y, Ferre D, Marchisio DL. Validation of a novel open-source work-flow for the simulation of packed-bed reactors. Chemical Engineering Journal 2015;.

[24] Cliffe K, Giles M, Scheichl R, Teckentrup AL. Multilevel monte carlo methods and applications to elliptic pdes with random coefficients. Computing and Visualization in Science 2011;14(1):3–15.

[25] Haji-Ali AL, Nobile F, von Schwerin E, Tempone R. Optimization of mesh hierarchies in multilevel monte carlo samplers. arXiv preprint arXiv:14032480 2014;.

[26] Haji-Ali AL, Nobile F, Tempone R. Multi index monte carlo: When sparsity meets sampling. arXiv preprint arXiv:14053757 2014;.

[27] Boccardo G, Del Plato L, Marchisio D, Augier F, Haroun Y, Ferre D, et al. Pore-scale simulation of fluid flow in packed-bed reactors via rigid-body simulations and cfd. In: Conference proceedings. SINTEF-NTNU; 2014, p. xx–.

[28] Khirevich S, Ginzburg I, Tallarek U. Coarse-and fine-grid numerical behavior of mrt/trt lattice-boltzmann schemes in regular and random sphere packings. Journal of Computational Physics 2015;281:708–42.

[29] Venema P, Struis R, Leyte J, Bedeaux D. The effective self-diffusion coefficient of solvent molecules in colloidal crystals. Journal of colloid and interface science 1991;141(2):360–73.

[30] Collier N, Haji-Ali AL, Nobile F, von Schwerin E, Tempone R. A continuation multilevel monte carlo algorithm. BIT Numerical Mathematics 2014;:1–34.

[31] Brezinski C. Extrapolation algorithms and padé approximations: a historical survey. Applied numerical mathematics 1996;20(3):299–318.

[32] Moroni M, Cushman JH. Statistical mechanics with three-dimensional particle tracking velocimetry experiments in the study of anomalous dispersion. ii. experiments. Physics of Fluids (1994-present) 2001;13(1):81–91.

[33] Ketelsen C, Scheichl R, Teckentrup A. A hierarchical multilevel markov chain monte carlo algorithm with applications to uncertainty quantification in subsurface flow. arXiv preprint arXiv:13037343 2013;.

[34] Giles M, Nagapetyan T, Ritter K. Multi-level monte carlo approximation of distribution functions and densities. Preprint 2014;157.

[35] Charrier J. Numerical analysis of the advection-diffusion of a solute in porous media with uncertainty. SIAM/ASA Journal on Uncertainty Quantification 2015;3(1):650–85.

[36] Bierig C, Chernov A. Convergence analysis of multilevel monte carlo variance estimators and application for random obstacle problems. Numerische Mathematik 2014;:1–35.