



HAL
open science

Méropé: A Microstructure Generator for Simulation of Heterogeneous Materials

Marc Josien

► **To cite this version:**

Marc Josien. Méropé: A Microstructure Generator for Simulation of Heterogeneous Materials. 2024.
hal-04486524

HAL Id: hal-04486524

<https://hal.science/hal-04486524>

Preprint submitted on 2 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Méropé: A Microstructure Generator for Simulation of Heterogeneous Materials

Marc Josien^a

^a*CEA, DES, IRESNE, DEC, SESC, LMCP, Cadarache F-13108 Saint-Paul-Lez-Durance, France,*

Abstract

Méropé is a software devoted to the geometrical design and the discretization of microstructures of random heterogeneous materials. Méropé aims at building large samples of microstructured materials, called Representative Volume Elements, in order to derive their effective physical behaviors. Various microstructures are supported: spherical, polyhedral or spheropolyhedral inclusions, polycrystals, Gaussian fields and boolean combinations of these. Discretization takes two forms: either regular Cartesian grids of (composite) voxels for computations with FFT-based solvers, or tetrahedral meshes for computations with Finite Element solvers. A special emphasis on the code has been put on performance, which will be further improved in the future.

This article aims at introducing the main features of the software as well as exemplifying its use.

Keywords: Microstructure, Sphere Packing, Voxelation, Mesh, Material science, Homogenization

1. Introduction

Méropé is a software devoted to the geometrical design and the discretization of microstructures of random heterogeneous materials. It aims at building large samples of these, called Representative Volume Elements, in order to derive their effective physical behaviors on large scales.

Méropé is a stand-alone component of the CEA software platform PLEIADES [31] dedicated to the simulation of nuclear fuels. It includes a C++ library wrapped into a Python interface, and can be used in combination with Finite Element (FE) solvers and solvers based on the Fast Fourier Transform (FFT). Méropé is distributed under a licence for research and education only on a github repository [24], which contains a fairly comprehensive documentation, on which this article is based.

1.1. The RVE method

Modern simulation of complex materials, such as nuclear fuels [31], concrete, wood, electromagnetic metamaterials, particle-based composites [39, 9], ... generally relies upon a multi-scale approach. Indeed, the materials under concern are typically heterogeneous at the microscopic scale: they feature a microstructure which dictates their effective behaviors at larger scales. In the sequel, the microstructure is supposed to be random, but stationary ergodic – roughly speaking, the statistics of the microstructure is invariant by translation, see [23] for more details.

There exist analytical approaches to derive the effective behavior of very specific heterogeneous materials [41] ; however, in general, one can but resort to numerical simulations. The strategy is the following: the heterogeneous material is sampled on a volume, which is sufficiently large with respect to the characteristic length of the microstructure: the Representative Volume Element (RVE). Then a physical simulation is performed, which is used in turn to derive the relevant parameters modeling the effective behavior on larger scales.

The RVE method takes root in the field of mathematical homogenization [3, 23]. In the past decades, understanding and improving this procedure motivated many researches, from theoretical and applied points of view. The interested reader may consult this non-exhaustive list of references: [7] for the well-posedness of the strategy, [26, 39] for numerical investigations about the boundary conditions, [18, 11] for a fine analysis of the approximation error with medium-size RVEs, and [5] for an overview of virtual microstructure generation.

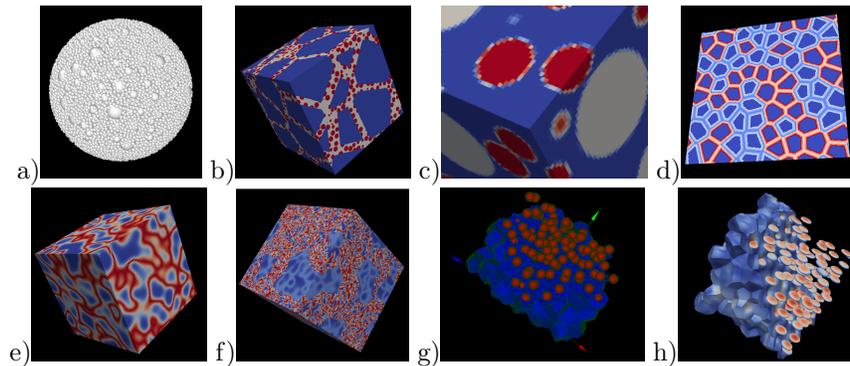


Figure 1: Examples of microstructures created with Mérope. From left to right: a) Polydisperse sphere packing inside a sphere, b) Polycrystal coated with a material with bubbles, c) Spherical inclusions (detail with composite voxels), d) 2D polycrystal with intergranular layers e) Gaussian field, f) Combination of Gaussian fields and a polycrystal, g) Periodic mesh of spherical inclusions inside a matrix (the matrix is cut in order to visualize the spheres), h) Deformation of g) under mechanical loading in linear elasticity (colors indicate the amplitude of deformation, the matrix is cut in order to visualize the spheres).

1.2. Functionalities of Mérope

The numerical implementation of the RVE method involves two main components: a microstructure generator, for sampling an RVE, and a solver, for simulating the physics under concern. Mérope focuses on the first task, which is subdivided into two steps: define an abstract microstructure, and then, discretize it as a mesh.

Mérope models 2D and 3D materials that belong to 3 main classes: inclusions inside a matrix, polycrystals, and anamorphosed Gaussian random fields. It allows for arbitrary combination of these by means of masks and Boolean operators. See Figure 1 b)–g) for illustrative examples.

Then, two different types of meshes can be produced: either a voxelation –that is, a regular Cartesian mesh– or an unstructured tetraedric mesh (in the 3D case). Each type of discretization is intended for a specific class of solvers: either FFT-based solvers, or usual FE solvers.

Since this is only a high-level introductory article, we will not write code neither point to specific scripts there. Nevertheless, impatient reader can consult the gallery here <https://github.com/MarcJos/Merope/blob/main/doc/Gallery.md>, where he may find a large variety of microstructures, as well as Python scripts for producing them.

1.3. Microstructure generators

There exist various softwares, either specific- or general-purpose, dedicated to microstructure generation. Here, we focus on classical ones, and disregard the promising ones based on machine-learning approaches, see [27] for example. In particular, we would like to emphasize on the following ones: voro++ [36], which is a compact library purely dedicated to building Laguerre tessellation (a crucial model for polycrystals); Neper [34], which is a medium-size code providing various functionalities for building and analyzing polycrystals; Dream 3D [19], which is general-purpose industrial code for microstructure, with a particular emphasis on voxelized microstructures.

In this landscape, Mérope features as a medium-size general-purpose code, with specific features related to homogenization, among which: the use of a periodic geometry, not only for voxelations but also for tetrahedral meshes, with adapted periodic surfaces, the use of composite voxels for FFT solvers. The variety of microstructures than can be obtained is broad (in particular, broader than Neper and Voro++), and the possibility to combine them paves the way to modeling realistic microstructures. Mérope is used through a user-friendly Python interface, and is based on a C++17 implementation, and proves to be competitive in terms of speed and robustness.

1.4. Performance

Notice that, from an operational point of view, microstructure generation is rarely the performance bottleneck in the RVE simulation when computing an

effective physical law; the major part of the computational time is spent inside the solver. Nevertheless, this is not the case anymore when the user attempts to optimize the microstructure parameters [33]: then, the speed of the microstructure generator becomes crucial. Thus, special attention has been paid to the performance of Mérope. Some important parts have been parallelized for multi-core shared memory architectures. Furthermore, we performed benchmarks in [24] that indicate for example that Mérope challenges successfully Neper [34] for polycrystal generation and voxelation. Mérope can generate a 3D polycrystal of 10^5 crystallites and voxelize it over a grid of 10^8 voxels within 15 seconds.

1.5. Outline

Sections 2 and 3 explain in more details the available microstructure geometries and how they are discretized, respectively. Section 4 exemplifies the use of Mérope on a (virtual) porous microstructure. We hope that it may serve as a didactic basis for numerical experimentalists. Section 5 briefly comments on the architecture of Mérope and its performances. Last, Section 6 concludes the article and proposes some perspectives.

2. Geometry of the microstructure

2.1. Periodicity

Many sampling strategies and Partial Differential Equation (PDE) boundary conditions can be used in the RVE method in order to obtain the effective behavior of the material. In particular, it is common to appeal to Dirichlet, Neumann or periodic boundary conditions, for a RVE sampling geometry which may be periodic or not (see [26]).

Nevertheless, it has been established empirically and mathematically in [26, 17, 11, 39] that periodic RVEs with periodic boundary conditions are superior: Given an RVE size, they yield more accurate effective behaviors than other reasonable strategies. Very roughly, the advantage of this periodic framework is that it amounts to considering the physics under concern on a periodic cuboid, for which there is intrinsically no boundary. Thus, the quality of the computed effective behavior is not tarnished by boundary-layer effects.

Hence, Mérope exclusively produces geometries on periodic cuboids – the only exception to this rule concerns sphere packings, where other options are available, see Section 2.2. In particular, this means that, when representing the geometry inside a cuboid of the Euclidian space, one sees that the microstructure crosses the boundary in a periodic way: for example, in Figure 2, the “cut” part of the red disk on the right-hand border is retrieved on the left-hand border.

When solving a PDE such as mechanical equilibrium, periodic boundary conditions should be employed as well. This is naturally tackled by FFT-based solvers. Concerning unstructured tetraedric meshes, it should be handled specifically, see Section 3.2.

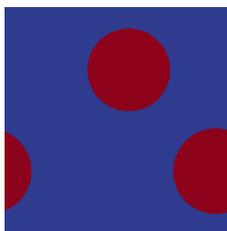


Figure 2: Representation of two disks in a periodic square

2.2. Sphere packing

2.2.1. Principle

The first building block for polycrystals and inclusion-made microstructures in Mérope are non-intersecting polydisperse sphere packings (polydispersity means that the spheres may have different radii). These can be generated by two different procedures:

- the Random Sequential Addition algorithm (RSA) [41], which iteratively places non-intersecting spheres, the centers of which are drawn randomly by a Poisson point process of uniform intensity (we discuss it in more details in Section 2.2.2),
- the mechanical contraction algorithm of Williams and Philipse (WP) [43], which starts from a diluted RSA output, and squeezes it.

Moreover, it is possible to generate as well random intersecting balls, with a Poisson Point Process of uniform intensity [40] (also called “Boolean algorithm”).

Spheres are placed inside a shape, in 2D or 3D, that can be either a periodic cuboid, a standard cuboid, a sphere, or a cylinder. See Figure 3 for examples.

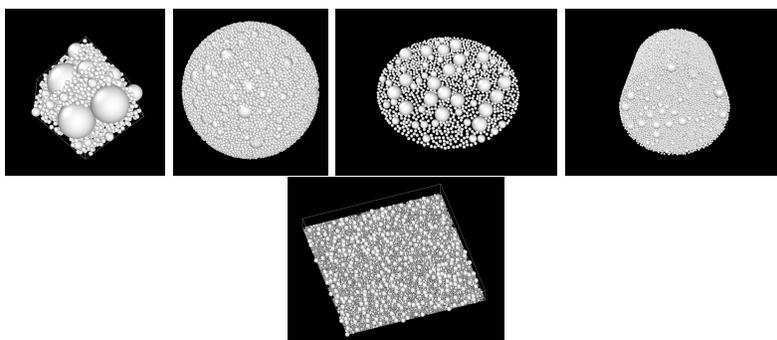


Figure 3: Various non-intersecting sphere packings in 2D and 3D, inside various shapes (from left to right: 3D inside periodic cube, 3D inside a sphere, 2D inside a disk, 3D inside a cylinder, 2D inside a periodic square)

The volume fraction, that is the total volume of spheres over the volume of the

	2D	3D
RSA	$\sim 55\%$	$\sim 38\%$
WP	$\sim 84\%$	$\sim 61\%$

Table 1: Asymptotic maximal volume fraction of monodisperse spheres packing ; see [42, Table 1] for the RSA values ; WP values are empirical values related to the Mérope implementation (for large RVEs with more than 10^4 spheres).

cuboid, is an important characteristic of sphere packings. For monodisperse spheres (*i.e.* spheres with equal radii), each non-intersecting sphere packing features an asymptotic maximal volume fraction when the volume of the cuboid goes to infinity. This volume fraction is described in Table 1. As expected by construction, the RSA procedure reaches relatively low volume fractions, whereas the WP procedure allows for higher volume fractions. Nevertheless, the WP algorithm depends on implementation parameters, and produces more heterogeneous configurations.

2.2.2. Algorithmic aspects of the RSA implementation

Here, we give more details on our implementation of the RSA algorithm [41]. The classical algorithm proceeds iteratively with these two steps for placing a new sphere a radius r :

- draw a sphere center c inside the box according to a Poisson point process of uniform intensity,
- verify if the new sphere of center c and radius r intersects already validated spheres. If it does, reject the new sphere else, validate the new sphere.

The algorithm reaches the *maximal configuration* when it is impossible to place a new sphere of desired radius r . Such a configuration is virtually out of reach for naive implementations of the RSA: Indeed, the more already placed spheres there are, the harder it becomes to place a new sphere.

Nevertheless, our implementation of the RSA algorithm is based on [13], that we generalize to polydisperse spheres packing. This algorithm resorts to an additional octree structure paving the cuboid, which keeps track of the zones where new spheres can be placed –these zones are small cuboids. New candidate spheres are drawn only in these zones. At some steps, each zone is further subdivided in new zones, which are removed if it is guaranteed that a new sphere cannot be placed inside. Such an algorithm is provably unbiased, and yields maximal configurations.

Empirically, its speed (measured as the number of new spheres placed in a second) is roughly independent of the size of the domain and the already achieved volume fraction. In this regards, Mérope’s performances are comparable to the original article [13], *e.g.* placing about 7.000 spheres by second when performing the RSA algorithm on a single CPU. Depending on the available main memory, between 10^5 and 10^7 spheres can be placed. We are currently improving

the current implementation by resorting to MPI and OpenMP to accelerate it further, and reach higher numbers of spheres [25].

2.3. Inclusion-made structures and polycrystals

Mérope emulates microstructures as a collection of simple inclusions (spheres, polyhedrons or spheropolyhedrons, possibly with inner layers), embedded into a matrix. The point of view is the following: each point of the microstructure is described by a single phase, which basically indicates the type of material, into which the point is. For Mérope inclusion-made structures, either the point is inside an inclusion of a given material, or it is inside the matrix. Such models are common in material science [5]; they are rich enough to represent porous media, metals, ceramics [33]...

Each inclusion is accessible to the user, who can change its phase or insert additional layers of uniform width inside it. (See for example the polycrystal in Figure 1 d), where each crystallite features 3 different layers.)

Polycrystal are considered as a subclass of inclusion-made structures, in which the inclusions actually cover all the volume (there is no matrix phase). In Mérope, random polycrystals are built from sphere packing, by means of Laguerre or Voronoi tessellations, where the center of each tessell is the center of an underlying sphere. For Laguerre tessellations, the radius of the sphere is used as weight (this weight is 0 for Voronoi tessellations). Namely, a point x of the RVE belongs to the crystallite i if it minimizes the quantity $d_i(x) := \text{dist}(x, c_i)^2 - r_i^2$, where c_i and r_i are the center and radius of the i th sphere, respectively, and where the function “dist” is the periodic euclidean distance. Anisotropy of the crystal (aspect ratio) can be imposed by applying a linear transform to an isotropic crystal.

It may also be useful to prescribe a priori the volume of each crystallite –for example based on experimental observations. (See Figure 4 for examples). Such a requirement is made possible by selecting the appropriate radius for each crystallite of the Laguerre tessellation (without changing its center). To this purpose, Mérope implements the state-of-the-art accelerated gradient-descent algorithm proposed in [29].

Last but no least, Mérope employs the library voro++ [36] to compute the Laguerre and Voronoi tessellations.

2.4. Random fields

Materials displaying variable densities, such as MOX fuels [14], can be modeled by random fields.

Mérope implements a subclass of these fields: stationary Gaussian scalar fields of zero mean, see *e.g.* Figure 1 e). We refer to [30] for mathematical details. We just emphasize that such fields g are entirely characterized by their covariance function

$$c(x, y) = c(x - y) = \langle g(x)g(y) \rangle,$$

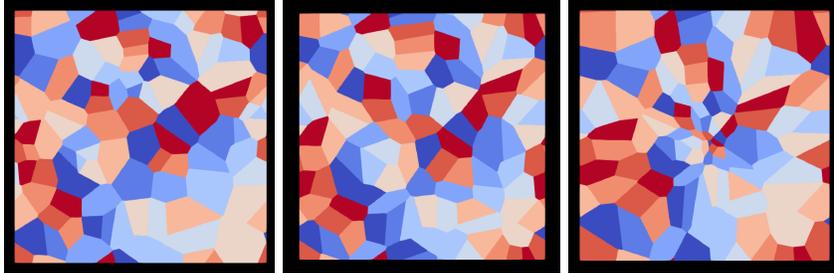


Figure 4: From left to right : 2D Original Laguerre tessellation; 2D Optimized Laguerre tessellation such that all crystallites share the same volume; 2D Optimized Laguerre tessellation such that the volume of each crystallite of center c is proportionnal to $|c|^2$.

where x and y are points of the RVE, and $\langle X \rangle$, the expectation of a random variable X . Gaussian fields g are then submitted to a nonlinear transform (or anamorphosed). This transformation is convenient to obtain finally a random field the range of which is physically relevant; for example a density in $[0, 1]$, a thermal conductivity larger than 0, etc. Both the covariance function and the nonlinear transform can be optimized to mimic properties measured on experimental samples (see [14]).

2.5. Combining microstructures

Méropé microstructures of all types can be combined recursively by means of masks and Boolean operations, giving rise to more complex materials. For example, the microstructure of Figure 1 f) is obtained by combining two different Gaussian fields, with a polycrystal as a mask. Notice however that, if all geometries can be voxelized, the meshing procedure with tetrahedra is restricted to simple microstructures.

3. Discretization of the microstructure

3.1. Voxelation

3.1.1. Principle

Voxelations are natural inputs for FFT-based solvers. This class of solvers, often used in thermomechanics, is quite recent (see the seminal article [32]), and takes advantage of the efficiency of the modern HPC implementation of the FFT-based routine, to achieve micromechanics computation on “gigantic” RVE with an important speed-up, compared with FE solvers used on unstructured mesh. (We refer to [38] for a comprehensive review on these methods.)

Méropé can be used in combination with the FFT-based solver AMITEX-FFTP [15], and offers “composite” voxels functionalities [9] for thermal conductivity problems. Composite voxels allow for mitigating the discretization error, by identifying all the phases present in each voxel, and averaging their physical properties in a suitable way. As shown in [38, 9], they reduce drastically the

discretization error, when the contrast of physical properties is high. As shown in Section 4 on a specific case, using composite voxels for high-contrast materials may reduce the error on the effective conductivity from 100% to 1%.

3.1.2. Algorithmic aspects and composite voxels

A special effort has been dedicated to gain efficiency in voxelation. The principle of the algorithm is the following: each inclusion is responsible for its voxelation, which it performs by computing on a subgrid its own voxelation. Then, on each subgrid, we use a strategy for fast coloring the pure voxels (*i.e.* voxels that only contain a single phase); the larger the inclusion sizes are compared to the voxel dimensions, the more efficient is this strategy.

On the illustrative basis of Figure 5, we detail how this subgrid of voxels is computed. We take advantage of the data alignment in the last coordinates and consider “columns” of voxels. Then, this column is subdivided into 3 zones (each zone is possibly not connected), depending on the 3 possible interaction between the layer and the voxel: either the latter is fully inside the layer (pure inner voxel), or fully out of the layer (pure outer voxel), or intersects the boundary of the layer (composite voxel). Only the limit of the zones are computed first. Hence, pure outer voxels are discarded, whereas pure inner voxels can be efficiently colored in a simple loop. The composite voxels, which are a minority, are further investigated. For each of them, the volume fraction of the considered layer is computed and put inside the voxel. If the surface of the layer intersected by the voxel is planar this computation is analytical [10], else, the surface is approximated by a planar surface.

When all the inclusions has filled the large common grid, each voxel is either pure, with a single phase (possibly the one of the matrix), or a mixture of phases i with volume fractions ϕ_i . These volume fractions are used in turn to compute averaged physical laws. Thus, it is possible to average the thermal conductivity λ of a mixture of materials by Reuss, Voigt, Smallest or Largest estimates, namely:

$$\begin{cases} \lambda_{\text{Reuss}} = \frac{1}{\sum_i \phi_i \frac{1}{\lambda_i}}, & \lambda_{\text{Voigt}} = \sum_i \phi_i \lambda_i, \\ \lambda_{\text{Smallest}} = \min_i \lambda_i, & \lambda_{\text{Largest}} = \max_i \lambda_i. \end{cases} \quad (1)$$

It is intended to make use of general laws for laminates in the future, in coordination with the solver AMITEX-FFTP. We refer to Section 4 for a study case using composite voxels, exemplifying that they yield more accurate results.

For assessing Mérope’s performances, we performed benchmarks in [24], comparing Mérope with various other microstructure generator. For example, on a single CPU, Mérope challenges successfully Neper [34] for polycrystal generation and voxelation, being faster by a factor between 2 and 20 depending on the type

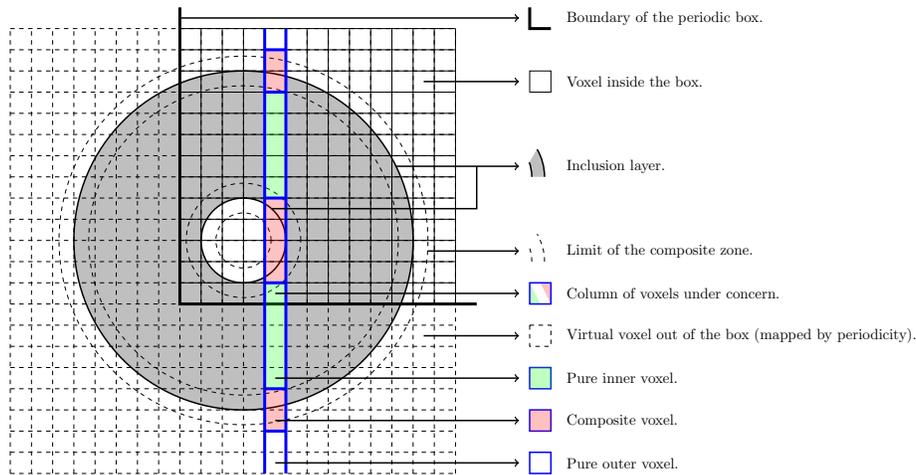


Figure 5: Voxelation principle. In gray, the layer of the (circular) inclusion that shall be voxelized. The thick line materializes the “boundary” of the periodic cuboid ; the small squares with continuous boundaries are truly inside the domain, whereas the small squares with dashed boundaries shall be copied by periodicity. The blue lines delimit the “column” of voxels that are currently voxelized. Inside this zone, pure inner voxels (green), pure outer voxels (white), and composite voxels (red) are identified.

of microstructure.¹ For example, Mérope can generate a 3D polycrystal of 10^5 crystallites and voxelize it over a grid of 10^8 voxels within 15 seconds. Thanks to its performance, it has also been used for optimizing microstructure parameters in [33], where, although a full 3D microstructure has to be generated, it was only required to voxelize slices of it².

3.2. Mesh

Tetraedric meshes are inputs for usual FE solvers, such as MFEM³ [4], Manta [22] or Cast3M [1]. The specificity of Mérope is that it produces periodic meshes, the periodicity surfaces of which are not flat, but adapted to the microstructure (see Figure 1 g) and h) and Figure 6, where it appears clearly on a 2D slice). Thus, it avoids numerical artifacts due to the usual flat periodicity surfaces (see *e.g.* [37]), which are caused by inclusions touching these surfaces and may result in mesh singularities.

Mérope defines the periodicity surface as follows:

¹See <https://github.com/MarcJos/Merope/blob/main/doc/Performances.md>. Computations have been performed on a CPU Intel Xeon Platinum 8268, comparing Neper 4.8.1-1 and Mérope 1.1.2, both compiled with g++ 12.1.0.

²Voxellizing first in 3D and then extracting a slice was not an option! Instead, Mérope proposes partial voxelation facilities.

³We refer the interested reader to examples in <https://github.com/thelfer/mfem-mgis>.

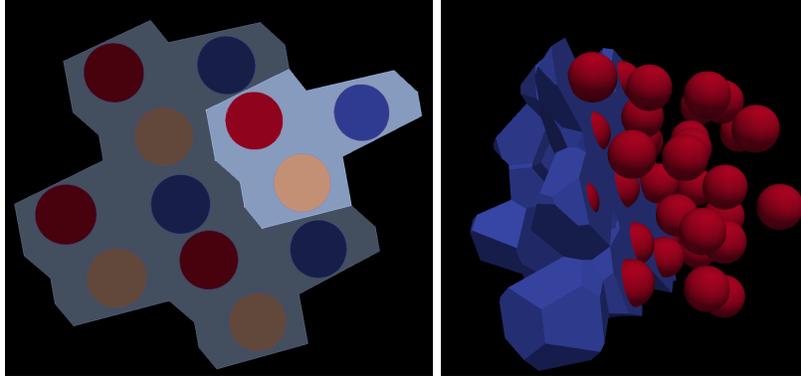


Figure 6: Left: Periodic mesh represented as follows: The unitary periodic mesh is in bright colors, and three periodic copies of the mesh are in darkened colors. (Edges are not shown.) Right: For comparison, a more complicated 3D geometry of periodic mesh, where the upper part of the matrix (blue) is removed in order to see the inner inclusions (red).

- For polycrystals, it is naturally the envelope of the polycrystal, when embedded in \mathbb{R}^3 , see *e.g.* [34].
- For well-separated spherical inclusions, the associated Laguerre tessellation is built (see Section 2.3). Then, the outer surface of this tessellation embedded in \mathbb{R}^3 is employed as the periodicity surfaces. By construction, this surface is as far as possible from the spheres, and thus, no mesh singularity may happen. This idea was present in [12], and will be described in more details in the upcoming [6].

Notice that, due to the technicality involved in building tetraedric meshes, the user may only mesh simple structures, namely: spherical inclusions and polycrystals. Also, strictly speaking, Mérope does *not* produce a mesh itself, but rather an input file in the .geo format for the mesh generator gmsh [16], which is an open-source software.

4. Simple and illustrative case study

4.1. Description of the study

As an illustrative case study for Mérope, we propose a totally virtual study of the effective thermal conductivity $\bar{\lambda}$ of an imaginary microstructured material. Although the results are not interesting *per se*, we would like to underline the methodology, which is general.

The material under concern is made of clay filled with lead bullets coated by a thin silver layer (see Figure 7). First, we make precise our microstructure definition. Then, we study the discretization error, and choose a homogenization rule for the composite voxels. (This will illustrate the efficiency of composite

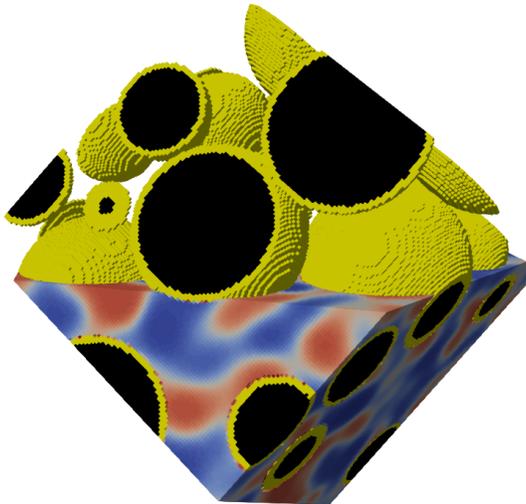


Figure 7: A representation of the microstructure. In black, the lead, in yellow, the silver, and in varying colors, the value of the thermal conductivity of the clay. A part of the clay matrix has been removed in order to see the bullets.

voxels.) Next, we investigate the influence of the size of the RVE. As a conclusion, we propose a choice of coherent parameters in terms of discretization step, RVE size, and sample number, in order to have the best precision with limited computational cost (here, we strive for an error of order 3%).

4.2. Description of the material microstructure

The clay will be modeled as a heterogeneous material, the conductivity of which is described by an anamorphosed Gaussian field, that is

$$\lambda_{\text{clay}}(x) = \lambda_{0,\text{clay}}(2 + \tanh(g(x))),$$

where x is the point coordinate, the function g is a stationary Gaussian field [30] of zero mean and covariance function

$$c(x - y) = \mathbb{E}[g(x)g(y)] = \exp(-(x - y)^2/2r_0^2).$$

The bullets are of two distinct classes, the first ones of radii r_1 and volume fraction ϕ_1 , and the second ones of radii r_2 and volume fraction ϕ_2 . We furthermore assume that the bullets distribution can be modelled by the mechanical contraction algorithm of Williams and Philipse. All the bullets are made of a core of lead coated by a thin silver layer of width l . The lead and silver materials are supposed to be of homogeneous thermal conductivities, λ_{lead} and λ_{silver} , respectively. In Table 2, we display the numerical values of each parameter.

	r_0	r_1	r_2	l	ϕ_1	ϕ_2	$\lambda_{0,\text{clay}}$	λ_{lead}	λ_{silver}
Value	1.0	2.0	3.5	0.25	0.2	0.3	0.6	35.3	429
Unit	mm	mm	mm	mm	-	-	W/m/K	W/m/K	W/m/K

Table 2: Geometric and physical parameters of the microstructure. See [2] for conductivities of silver and lead.

Although this framework is not related to any realistic application, it embodies main features of actual problems, when attempting to compute the thermal conductivity of a microstructure representative of nuclear materials [33, 8], which may involve high contrasts and thin layers.

4.3. Discretization error

The first step is an empirical study of the discretization error. We take a small cube, with edge length 15mm, on which we investigate the error, using different types of pure voxels, the conductivity of which is equal to the one its center, and composite voxels, namely according the rules Voigt, Reuss, Smallest, Largest (see Section 3.1). One difficulty of the structure that we defined is that the Gaussian fields cannot be easily refined.⁴ However, since the major source of error is due to the coating, due to its fineness and its high conductivity, w.r.t. to the other material. Hence, we only study this numerical error, by averaging the conductivity of the clay as being equal to $\lambda_{0,\text{clay}}$. The results are displayed in Figure 8.

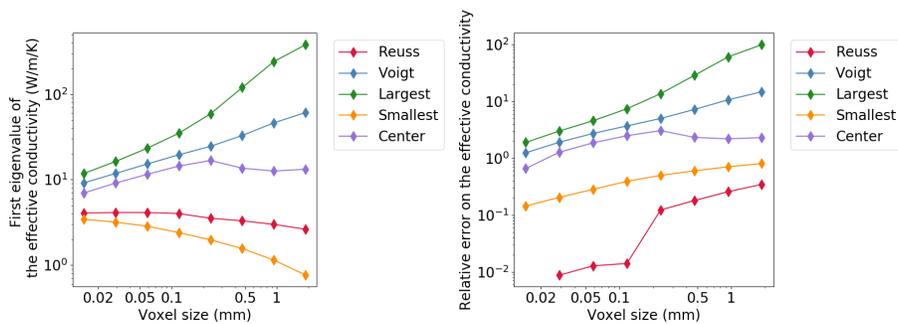


Figure 8: Left: First eigenvalue of the effective conductivity, in W/m/K, in function of the discretization step in mm (length of the edge of a voxel), for various composite voxels. Right: Relative error on the effective conductivity in function of the discretization step, in mm, for various composite voxels, using the Euclidean norm on the matrix $\|M\| = \sqrt{\sum_{i,j=1}^3 (M)_{ij}^2}$.

The effective conductivity is matrix-valued. To visualize them more easily, we

⁴Picking another Gaussian field on a smaller scale is different from refining a given Gaussian field, which would necessitate when refining to interpolate and then pick new random variables on the new points.

plot the largest eigenvalue of this matrix on the left of Figure 8. First, as expected, for fixed voxel size, these eigenvalues are ordered as follows:

$$\bar{\lambda}_{\text{Smallest}} \leq \bar{\lambda}_{\text{Reuss}} \leq \bar{\lambda}_{\text{Voigt}} \leq \bar{\lambda}_{\text{Largest}}.$$

This inequality is expected from the local inequalities obtained from (1). Nevertheless, we observe that the range of values is broad for moderate voxel sizes. For example, the eigenvalues range from $3.9Wm-1K^{-1}$ to $34Wm-1K^{-1}$ for a voxel size of $0.12mm$, which is smaller than the smallest characteristic size (namely, the width of the silver coating). Hence, the choice of a suitable discretization rule for composite voxels is of paramount importance, otherwise, it may lead to a relative error of more than 1000%.

Then, we observe that the error increment of the Reuss composite voxels is far smaller than for the other voxel rules. Thus, we use as a reference value the effective thermal conductivity for the Reuss rule computed with the smallest possible voxel edge length, namely $0.015mm$. This value relates to a voxelation of around 10^9 voxels, which is already costly in terms of memory (just storing the coefficient field in double precision takes 8GB of memory; then using a FFT solver such as AMITEX-FFTP may take more than 400GB RAM), and takes time (around 5h for evaluating the homogenized matrix on 18 CPUs Intel Xeon Platinum 8268 with the software AMITEX-FFTP [15]). On the right, we show the relative error on the homogenized matrix, by comparing the outputs of our computation to this reference value.

This establishes the better accuracy of the Reuss method, followed by the Smallest method, and then the Voigt, Center and Largest methods. Notice that, even for moderate size of voxels of $0.12mm$, the error for the Reuss method is of order 1%, whereas it is higher higher than 100% for all other methods but for the Smallest method, where it is of order 10%. The superiority of the Reuss method is not unexpected. Indeed, it corresponds to a homogenization rule in a laminate where the thermal flux is assumed to be orthogonal to the layers [3]. In the precise case that we are studying, since the silver is of high conductivity with respect to the other materials, the thermal flux is easily driven by the silver layer, and then effective conductivity is highly dependent on the thermal barrier caused by the touching zones between two different bullets, where the thermal flux is approximately orthogonal to the surfaces of the spheres.

4.4. Homogenization error

The last step is the study of the homogenization error or, in other words, the representativeness of a cube of edge length L . As in [18], we decompose the error on the effective conductivity $\bar{\lambda}$, estimated by the value $\bar{\lambda}_L$ in the cube of edge length L , into two parts :

$$\langle \|\bar{\lambda} - \bar{\lambda}_L\|^2 \rangle = \|\bar{\lambda} - \langle \bar{\lambda}_L \rangle\|^2 + \langle \|\bar{\lambda}_L - \langle \bar{\lambda}_L \rangle\|^2 \rangle. \quad (2)$$

The right-hand side term of (2) is the square of the bias, and the second one is the square of the standard deviation. Here, the average $\langle \cdot \rangle$ denotes the expectation (recall that our microstructure is assumed to be random). Notice that, in

practice, the empirical apparent thermal conductivity $\tilde{\lambda}_L$ is evaluated by averaging over N microstructures independently throw. This results in decreasing the standard deviation as

$$\langle \|\bar{\lambda} - \tilde{\lambda}_L\|^2 \rangle = \|\bar{\lambda} - \langle \bar{\lambda}_L \rangle\|^2 + \frac{1}{N} \langle \|\bar{\lambda}_L - \langle \bar{\lambda}_L \rangle\|^2 \rangle.$$

Following [28, 11], when $L \uparrow \infty$, we expect the scaling laws

$$\|\bar{\lambda} - \langle \bar{\lambda}_L \rangle\| \simeq CL^{-3}, \quad \text{and} \quad \sqrt{\langle \|\bar{\lambda}_L - \langle \bar{\lambda}_L \rangle\|^2 \rangle} \simeq CL^{-\frac{3}{2}}.$$

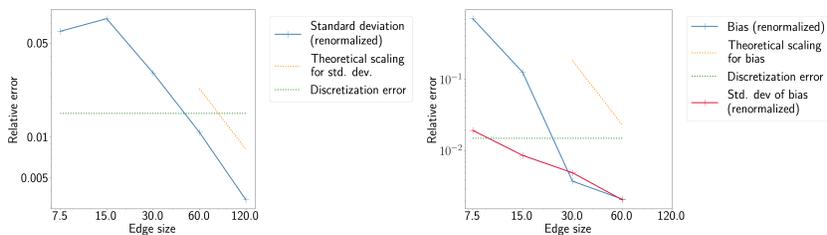


Figure 9: Left: Relative standard deviation of the apparent thermal conductivity as a function of the edge length L (with theoretical scaling in L^{-3}). Right: Relative empirical bias of the apparent thermal conductivity as a function of the edge length L , and standard deviation thereof. For both graphs, the discretization error is displayed, as well as theoretical scalings.

Here comes the numerical investigations. First, according to Section 4.3, we choose a voxel edge length being equal to $15/128 \simeq 0.12mm$, with Reuss composite voxels (leading to approximately a relative error of 1.5% due to discretization). Then, for each edge length L in the range 7.5 to 120 by dyadic steps, we draw independently random microstructures for which we compute the apparent thermal conductivity. This leads us to evaluate empirically the standard deviation of the apparent thermal conductivity, as well as an empirical bias, which we compute as $|\tilde{\lambda}_L - \tilde{\lambda}_{L_{\max}}|$, where $L_{\max} = 120$. For convenience, we renormalize our results by $|\tilde{\lambda}_{L=120}|$.

Our results are plotted in Figure 9. First, we remark that we get the correct scaling of the standard deviation. The latter even seems to survive independently of the discretization error: this is not surprising in the sense that it comes from the law of large numbers, which is somehow independent to the discretization error. Then, as already remarked in [39], we do not recover the expected scaling of the bias. Here, it may be explained easily: the standard deviation of the bias becomes larger than the value of the bias itself (hence, the precise value of the bias becomes meaningless).

4.5. Conclusions of the numerical study

From an applied perspective, a numerical experimentalist attempting to compute the effective thermal coefficient of the given material may draw the following conclusions:

- A discretization step $h = 0.12mm$ is sufficient to get a discretization error of about 1%.
- Then, the size of the volume should be chosen as $L = 30$. Indeed, for higher L the bias then becomes smaller than the discretization error.
- At $L = 30$, the relative standard deviation is of $s = 3\%$. Hence, for having a relative error of $e := 1\%$ due to randomness within a 95% confident interval, N independent computation of the apparent thermal conductivity should be performed, with $N = (2 * s/e)^2 = 36$.

Last, putting together all the sources of error, for the chosen parameter, we end up with an error of order 3% as desired (the total error being the sum of the discretization error, the bias and the standard deviation).

Such a preliminary study is crucial. If the numerical experimentalist is not only interested in a single class a microstructures as in Section 4.1, but rather with a family of such structures with varying parameters, he may transpose its conclusions about discretization error and representativeness to the whole family. (Thus, he may save time and computational resources.)

Evaluating each source of error in order to get the best precision can easily decrease by a factor 10 – 100 the total CPU hours cost of the whole study. In practice, since a single FFT (or FE) computation with a large number of voxels is costly (it can take hours on a dozen of processors, *cf.* Section 4.3), such a method can save thousands of CPU hours, which is far from being negligible when using a laptop or a small cluster.

5. Prerequisites, architecture, code quality and performance

5.1. Prerequisites and synergy with other applications

For compiling the Mérope code, the user shall have as a prerequisite:

- a recent Linux distribution (Debian 10+, Ubuntu 22+),
- cmake 3.16+, g++ 10.2+, and Python 3.5,
- the Math Kernel Library (used for implementation of the FFT),
- Pybind11 [21], which is used to build the Python user interface from the C++ Mérope code,
- voro++ [36] to compute the Laguerre and Voronoi tessellations.

Moreover, Mérope can be (and has been) used in synergy with the following applications:

- the mesh generator gmsh [16],
- the FFT-based solver AMITEX-FFTP [15],
- the code generator for physical laws MFront [20],

- the FE solvers Cast3M [1] and MFEM [4],
- the Discrete Element Method simulator Rockable [35], which may create an input geometry which is voxelized through Mérope.

5.2. Architecture

The main part of Mérope is a collection of C++ libraries, implemented with standard C++17. Two main modules are distinguished:

- The module AlgoPacking, which is dedicated to the generation of sphere packing, and which collects basic geometry functions.
- The module merope_core, which contains the microstructure abstractions, and the discretization procedures.

The user accesses these libraries through a Python interface, which maps one-to-one the C++ functions, thanks to the pybind11 library [21].

Some Python scripts are proposed to the user. These generate microstructures and perform some RVE experiments, such as computing the effective thermal coefficients of a polycrystalline material.

5.3. Code quality

Mérope attempts to propose a state-of-the-art code to build virtual microstructures, in terms of user-friendliness, flexibility, and robustness. In this regards, the github repository [24] proposes a fairly comprehensive documentation. Concerning the code quality, non-regression is ensured by continuous integration tests performed on some CEA computers (but not directly through continuous integration platform provided by github). These tests have shown that the coverage rate for functions is about 75%, which is moderately satisfactory, and will be improved in the future.

6. Conclusion and perspectives

Mérope is a numerical tool for generating samples of random microstructures in order to compute their effective properties with the RVE approach in 2D and 3D. It proposes a rich varieties of microstructures: inclusions with layers, polycrystals, random fields, and arbitrary combinations of the latter. Two types of discretization are implemented: voxelation, for FFT solvers, and tetraedric mesh, for FE solvers. The voxelation procedures are more advanced and feature state-of-the-art functionalities, whereas the meshing procedure is limited to simple structures, although it proposes a quite innovative implementation of periodic surfaces [12]. Mérope has been used in the research on nuclear fuels [33, 6], but its functionalities may reach a broader audience, interested in RVE numerical computations -especially in micromechanics.

Mérope is extensively documented and submitted to regression tests. Its performances have been assessed by benchmarks, leading to the conclusion that it enjoys state-of-the art performances on a single CPU.

Méropé is currently still under development. The main perspectives concern the following functionalities:

- HPC implementation (in MPI), in particular for sphere packing –which is of independent interest. (We refer to the upcoming [25].)
- Generalization of the use of composite voxels for arbitrary physics, in tight coupling with the solver AMITEX-FFTP.
- Extension of the meshing procedure to more general microstructures.
- Post-processing facilities for extracting statistics of microstructures.

7. Acknowledgment and licence

Méropé is a stand-alone component of the CEA software platform PLEIADES [31] dedicated to the simulation of nuclear fuels. As such, the Méropé project has been funded by the CEA and its industrial partners EDF and Framatome. We thank all of them for their support. In this regards, we emphasize that the Méropé software is the property of the CEA and is distributed under a licence for research and education only (see [24] for the detailed licence).

Last, but not least, we warmly thank all the contributors to Méropé: Louis Belgrand, Victor Blanc, Etienne Castelier, Cedric Cocquebert, Thomas Helfer, Marcelle Ibrahim, Guillaume Latu, Renaud Masson, Léo Moutin, Raphaël Prat, as well as Matti Schneider, Lionel Gélébart, Christophe Bourcier and Isabelle Ramière for fruitful discussions.

This text has been exclusively written by a human, and not by any IA tool.

References

- [1] Cast3m. <http://www-cast3m.cea.fr/>.
- [2] *CRC Handbook of Chemistry and Physics*. CRC press, 104 edition, 2023.
- [3] G. Allaire. *Shape optimization by the homogenization method*, volume 146 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2002.
- [4] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cervený, V. Dobrev, Y. Dudouit, A. Fisher, Tz. Kolev, W. Pazner, M. Stowell, V. Tomov, I. Akkerman, J. Dahm, D. Medina, and S. Zampini. MFEM: A modular finite element methods library. *Computers & Mathematics with Applications*, 81:42–74, 2021.
- [5] S. Bargmann, B. Klusemann, J. Markmann, J. E. Schnabel, K. Schneider, C. Soyarslan, and J. Wilmers. Generation of 3d representative volume elements for heterogeneous materials: A review. *Progress in Materials Science*, 96:322–384, 2018.

- [6] L. Belgrand, I. Ramière, M. Josien, and F. Lebon. Improved accuracy of dirichlet-like microscopic solutions on representative volume elements. in preparation.
- [7] A. Bourgeat and A. Piatnitski. Approximations of effective coefficients in stochastic homogenization. In *Annales de l'IHP Probabilités et statistiques*, volume 40, pages 153–165, 2004.
- [8] T. Calvet, J.-M. Vanson, and R. Masson. A DEM/FFT approach to simulate the effective thermal conductivity of granular media. *International Journal of Thermal Sciences*, 172:107339, 2022.
- [9] R. Charière, A. Marano, and L. Gélébart. Use of composite voxels in FFT based elastic simulations of hollow glass microspheres/polypropylene composites. *International Journal of Solids and Structures*, 182:1–14, 2020.
- [10] Y. Cho and E. Cho. The volume of simplices clipped by a half space. *Applied mathematics letters*, 14(6):731–735, 2001.
- [11] N. Clozeau, M. Josien, F. Otto, and Q. Xu. Bias in the representative volume element method: periodize the ensemble instead of its realizations. *Foundations of Computational Mathematics*, pages 1–83, 2023.
- [12] M. Danielsson, D. Parks, and M. Boyce. Micromechanics, macromechanics and constitutive modeling of the elasto-viscoplastic deformation of rubber-toughened glassy polymers. *Journal of the Mechanics and Physics of Solids*, 55(3):533–561, 2007.
- [13] M. Ebeida, S. Mitchell, A. Patney, A. Davidson, and J. Owens. A simple algorithm for maximal poisson-disk sampling in high dimensions. In *Computer Graphics Forum*, volume 31, pages 785–794. Wiley Online Library, 2012.
- [14] A. El Abdi. *Génération 3D aléatoire de microstructures de combustibles nucléaires MOX et homogénéisation mécanique*. PhD thesis, 2021. hal-01901247v1.
- [15] L. Gélébart and R. Mondon-Cancel. Non-linear extension of FFT-based methods accelerated by conjugate gradients to evaluate the mechanical behavior of composite materials. *Computational Materials Science*, 77:430–439, 2013.
- [16] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.
- [17] A. Gloria. Reduction of the resonance error—part 1: Approximation of homogenized coefficients. *Mathematical Models and Methods in Applied Sciences*, 21(08):1601–1630, 2011.

- [18] A. Gloria, S. Neukamm, and F. Otto. Quantification of ergodicity in stochastic homogenization: optimal bounds via spectral gap on Glauber dynamics. *Invent. Math.*, 199(2):455–515, 2015.
- [19] M. Groeber and M. Jackson. DREAM. 3D: a digital representation environment for the analysis of microstructure in 3d. *Integrating materials and manufacturing innovation*, 3:56–72, 2014.
- [20] T. Helfer, B. Michel, J.-M. Proix, M. Salvo, J. Sercombe, and M. Casella. Introducing the open-source mfront code generator: Application to mechanical behaviours and material knowledge management within the PLEIADES fuel element modelling platform. *Computers & Mathematics with Applications*, 70(5):994–1023, 2015.
- [21] W. Jakob. Pybind11. <https://github.com/pybind/pybind11>.
- [22] O. Jamond, N. Lelong, A. Fourmont, J. Bluthé, M. Breuze, P. Bouda, G. Brooking, F. Drui, A. Epalle, O. Fandeur, et al. Manta: un code HPC généraliste pour la simulation de problèmes complexes en mécanique. In *CSMA 2022 15ème Colloque National en Calcul des Structures*, 2022.
- [23] V. Jikov, S. Kozlov, and O. Oleĭnik. *Homogenization of differential operators and integral functionals*. Springer-Verlag, Berlin, 1994.
- [24] M. Josien. <https://github.com/MarcJos/Merope>.
- [25] M. Josien and R. Prat. Parallel and bias-free RSA algorithm for maximal Poisson-sphere sampling in dimensions $d \geq 2$. 2024. In preparation.
- [26] T. Kanit, S. Forest, I. Galliet, V. Mounoury, and D. Jeulin. Determination of the size of the representative volume element for random composites: statistical and numerical approach. *Journal of the Mechanics and Physics of Solids*, 40(13–14):3647–3679, 2003.
- [27] S. Kench and S. Cooper. Generating three-dimensional structures from a two-dimensional slice with generative adversarial network-based dimensionality expansion. *Nature Machine Intelligence*, 3(4):299–305, 2021.
- [28] V. Khoromskaia, B. Khoromskij, and F. Otto. Numerical study in stochastic homogenization for elliptic partial differential equations: Convergence rate in the size of representative volume elements. *Numerical Linear Algebra with Applications*, 27(3):e2296, 2020.
- [29] J. Kuhn, M. Schneider, P. Sonnweber-Ribic, and T. Böhlke. Fast methods for computing centroidal laguerre tessellations for prescribed volume fractions with applications to microstructure generation of polycrystalline materials. *Computer Methods in Applied Mechanics and Engineering*, 369:113175, 2020.

- [30] M. A. Lifshits. *Gaussian random functions*, volume 322 of *Mathematics and its Applications*. Kluwer Academic Publishers, Dordrecht, 1995.
- [31] B. Michel, I. Ramière, I. Viillard, C. Introini, M. Lainet, N. Chauvin, V. Marelle, A. Bouloire, T. Helfer, R. Masson, J. Sercombe, J.C. Dumas, L. Noirot, and S. Bernaud. Chapter 9 - two fuel performance codes of the PLEIADES platform: ALCYONE and GERMINAL. In *Nuclear Power Plant Design and Analysis Codes*, Series in Energy, pages 207–233. Woodhead Publishing, 2021.
- [32] H. Moulinec and P. Suquet. A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *Comptes rendus de l'Académie des sciences. Série II. Mécanique, physique, chimie, astronomie.*, 1994.
- [33] L. Moutin, J. Meynard, M. Josien, M. Bornert, C. Duguay, F. Adenot, V. Bouineau, L. Fayette, and R. Masson. Realistic morphological models of weakly to strongly branched pore networks for the computation of effective properties. *International Journal of Solids and Structures*, page 112249, 2023.
- [34] R. Quey, P. Dawson, and F. Barbe. Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, 200(17-20):1729–1745, 2011.
- [35] V. Richefeu. Rockable. <https://richefeu.github.io/homepage-vrichefeu/tools/0000/01/02/rockable/>.
- [36] C. Rycroft. Voro++: A three-dimensional voronoi cell library in c++. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2009.
- [37] K. Schneider, B. Klusemann, and S. Bargmann. Automatic three-dimensional geometry and mesh generation of periodic representative volume elements for matrix-inclusion composites. *Advances in Engineering Software*, 99:177–188, 2016.
- [38] M. Schneider. A review of nonlinear FFT-based computational homogenization methods. *Acta Mechanica*, 232(6):2051–2100, 2021.
- [39] M. Schneider, M. Josien, and F. Otto. Representative volume elements for matrix-inclusion composites - a computational study on the effects of an improper treatment of particles intersecting the boundary and the benefits of periodizing the ensemble. *Journal of the Mechanics and Physics of Solids*, 158:104652, 2022.
- [40] R. Streit. *Poisson point processes*. Springer, New York, 2010. Imaging, tracking, and sensing.

- [41] S. Torquato. Random heterogeneous materials: microstructure and macroscopic properties. *Appl. Mech. Rev.*, 55(4), 2002.
- [42] S. Torquato, O. U. Uche, and F. H. Stillinger. Random sequential addition of hard spheres in high euclidean dimensions. *Phys. Rev. E*, 74:061308, Dec 2006.
- [43] S. Williams and A. Philipse. Random packings of spheres and spherocylinders simulated by mechanical contraction. *Physical Review E*, 67(5):051301, 2003.