Cement and Concrete Research 41 (2011) 1320–1329

Contents lists available at ScienceDirect

Cement and Concrete Research

journal homepage: http://ees.elsevier.com/CEMCON/default.asp

Numerical homogenization of concrete microstructures without explicit meshes

Julien Sanahuja *, Charles Toulemonde

EDF R&D, Site des Renardières, Route de Sens, Ecuelles, 77250 Moret sur Loing, France

ARTICLE INFO

Article history:
Received 3 December 2010
Accepted 29 March 2011

Keywords:
Microstructure (B)
Creep (C)
Diffusion (C)
Elastic moduli (C)
Micromechanics (C)

ABSTRACT

Life management of electric hydro or nuclear power plants requires to estimate long-term concrete properties on facilities, for obvious safety and serviceability reasons. Decades-old structures are foreseen to be operational for several more decades. As a large number of different concrete formulations are found in EDF facilities, empirical models based on many experiments cannot be an option for a large fleet of power plant buildings. To build predictive models, homogenization techniques offer an appealing alternative. To properly upscale creep, especially at long term, a rather precise description of the microstructure is required. However, the complexity of the morphology of concrete poses several challenges. In particular, concrete is formulated to maximize the packing density of the granular skeleton, leading to aggregates spanning several length scales with small inter particle spacings. Thus, explicit meshing of realistic concrete microstructures is either out of reach of current meshing algorithms or would produce a number of degrees of freedom far higher than the current generic FEM codes capabilities.

This paper proposes a method to deal with complex matrix-inclusions microstructures such as the ones encountered at the mortar or concrete scales, without explicitly meshing them. The microstructure is superimposed to an independent mesh, which is a regular Cartesian grid. This inevitably yields so called "gray elements", spanning across multiple phases. As the reliability of the estimate of the effective properties highly depends on the behavior affected to these gray elements, special attention is paid to them. As far as the question of the solvers is concerned, generic FEM codes are found to lack efficiency: they cannot reach high enough levels of discretization with classical free meshes, and they do not take advantage of the regular structure of the mesh. Thus, a specific finite differences/finite volumes solver has been developed. At first, generic off-the-shelf linear system solvers were used. To further improve the efficiency in terms of memory requirements, specific variants of the preconditioned conjugate gradient were implemented. This allowed to homogenize the conductivity of a concrete-like microstructure using more than 10^9 degrees of freedom on a rather common hardware for 2010 (a PC embedding 48 GB of RAM). Taking benefit of the properties of the regular Cartesian grid we have also investigated a multi-level method to improve the CPU efficiency of the code.

1. Effective creep estimation: does stiffness contrast matter?

To estimate the effective creep function of a composite made up of several non aging linear visco-elastic phases, the correspondence principle is classically used [11]. Indeed, the Laplace–Carson transform changes the structure of a visco-elastic behavior into an elastic one. This allows to take advantage of homogenization models developed in the framework of linear elasticity. The effective creep function is thus obtained in the Laplace domain. The last step is to revert into the physical time domain, inverting the Laplace transform. Unfortunately, except for very simple elementary visco-elastic behaviors and elastic homogenization models, this inversion is not analytically tractable. A numerical inversion procedure is thus required, such as for example [10,19] the truncated series expansion:

$$f(t) \approx \frac{\ln 2}{t} \sum_{k=1}^{2M} \xi_k \mathcal{L}_f\left(\frac{k \ln 2}{t}\right) \tag{1}$$

$\mathcal{L}_f$ denoting the known Laplace transform of f, M being a sufficiently large integer (typically $M = 10$ is found to be enough), and ξ_k being appropriate weights [10,19].

For pedagogic purposes, let us consider the upscalling of mortar creep to estimate concrete creep. To keep things analytically tractable, concrete morphology is modeled resorting to a Mori–Tanaka scheme [16]. The mortar behavior is represented by an isotropic Burgers model (Fig. 1), with different stiffness and viscous properties for spherical and deviatoric loadings, and the aggregates are assumed to be elastic. The mechanical characteristics of mortar and aggregates, come from [15] and are reported into Table 1. The aggregates volume fraction

* Corresponding author.
E-mail addresses: julien.sanahuja@edf.fr (J. Sanahuja), charles.toulemonde@edf.fr (C. Toulemonde).

0008-8846/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cemconres.2011.03.023

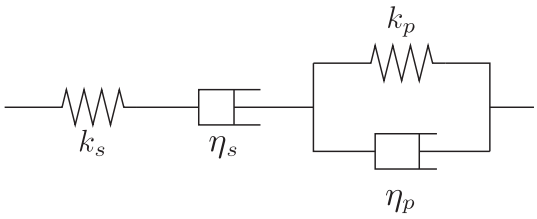

Fig. 1. Burgers rheological model.

Table 1
Mechanical properties of mortar (Burgers model, see Fig. 1) and aggregates [15].

		Spherical component ($*3\mathbb{J}$)	Deviatoric component ($*2\mathbb{K}$)
Mortar visco-elasticity	k_s (GPa)	12.8	7.76
	k_p (GPa)	4.49	5.13
	η_s (GPa·d)	5600	5470
	η_p (GPa·d)	123	115
Aggregates elasticity	(GPa)	52.2	28.3

is 0.50 [15]. The creep functions of mortar, aggregates and concrete are plotted on the left part of Fig. 2. The behavior of concrete is estimated through both an analytical and the numerical Laplace transform inversion (Eq. (1)), with $M=10$. The analytical and numerical results are virtually superimposed.

Let us now focus on the numerical inversion of the Laplace transform. As readily seen from Eq. (1), $2M$ calls to the elastic homogenization model are required to get the value of the effective creep function at one given physical time t. The elastic homogenization model being used in the Laplace–Carson domain, the stiffness of mortar appearing in these computations is not the physical elastic stiffness. Rather, this fictitious stiffness depends on the visco-elastic behavior of mortar and of the Laplace variable $p=k(\ln 2)/t$. Thus, for each computed time t, $2M$ elastic homogenizations are performed, using various ratios (so-called contrasts) between the Young's modulus (E_i) of inclusions (aggregates) and the fictitious Young's modulus (E_m) affected to the matrix phase (mortar). The minimum and maximum of these contrasts are reported on Table 2 for various times between 1 min and 100 years. The higher the time t, the higher the contrast. In particular, a contrast of 100 is not unrealistic: it is required to estimate creep of decades-old structures.

From the elastic homogenization point of view, higher contrasts require a more refined description of the microstructure [14]. This fact is basically illustrated on the right part of Fig. 2: the effective Young's modulus is plotted as a function of the contrast between inclusions and matrix, reusing the same volume fraction of aggregates as in the previous creep computation. Two different Eshelby-based [9] homogenization schemes are used to investigate radically different microstructures: the Mori–Tanaka scheme [16] (inclusions embedded into a matrix) and the self-consistent scheme [13] (polycrystal-like morphology). As expected, the discrepancy between the two estimates increases as contrast increases.

Table 2
Minimum and maximum contrasts encountered in the underlying $2M$ elastic homogenization computations required to estimate the effective creep function at time t.

Time after loading (t)	E_i/E_m min	E_i/E_m max
1 min	3.73	3.73
1 day	3.75	4.11
1 year	7.29	13.0
10 years	11.2	40.8
100 years	25.5	317.

To properly estimate long term creep (that is at large times after loading), the elastic homogenization model used in the Laplace domain must integrate a rather detailed morphological model. As semi-analytical homogenization schemes may not be able to reach this level of detail, it is necessary to resort to numerical homogenization. Thus, the remaining part of this paper only focuses on linear behaviors (elasticity/conductivity), investigating various contrasts between the properties (elastic moduli/conductivities) of inclusions and matrix. More precisely, contrasts of 3 (which roughly corresponds to the typical ratio of elastic moduli of aggregates and mortar), 10^2 (having creep models in mind) and 10^{-8} (to also explore low contrasts, for completeness) are considered. Creep modeling will be the subject of another paper.

2. Homogenization of concrete-like microstructures: a numerical approach

As shown in the previous section in the case of creep, accurate quantitative homogenization of concrete behavior requires to take into account a rather precise description of the morphology. To the authors' knowledge, only numerical homogenization can reach this goal. A first attempt has been done, resorting to a generic FEM code. Unfortunately, sufficient discretization levels were not reachable. Then finite differences and finite volumes on regular Cartesian grids, avoiding the storage and the computation of a supporting mesh, were investigated.

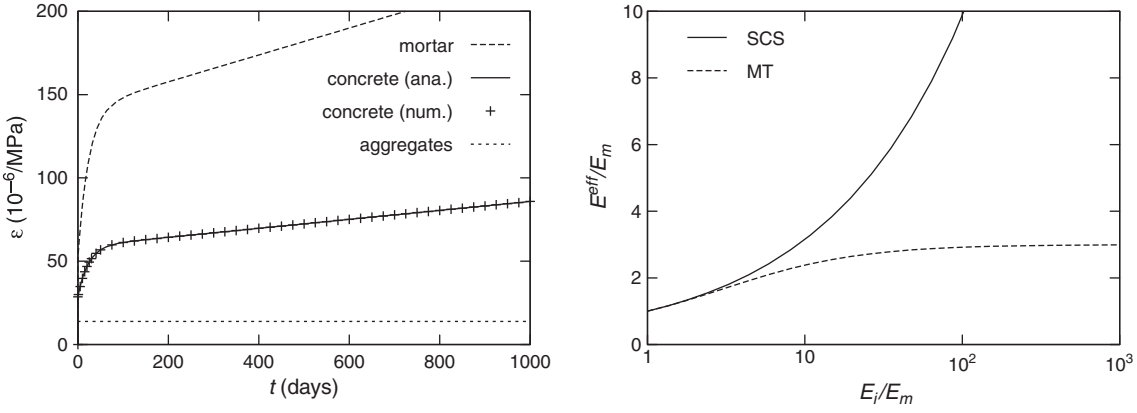

Fig. 2. Left: uniaxial creep functions of aggregates (elastic), mortar (visco-elastic, Burgers model) and concrete (visco-elastic, estimated by homogenization). Right: influence of contrast on the effective Young's modulus of a two phase material, considering two homogenization schemes, Mori–Tanaka (MT) and self-consistent (SCS).

2.1. A first attempt using FEM

Due to its highly compact granular skeleton with a wide particle size distribution, the microstructure of concrete is inherently difficult to mesh. Even if meshing would be possible, the number of degrees of freedom (DOFs) would be too large for current FEM codes on current computers. Alternative approaches are thus required. A method has been proposed [20], which still takes advantage of existing FEM codes, but relaxes the demand on the meshing procedure. More precisely, the aggregates are superimposed over an independent mesh of the representative volume element (RVE) domain (this mesh can, for example, be a regular grid of hexahedra subdivided into tetrahedra). For each finite element, a geometrical computation estimates the local volume fraction of aggregates. As far as simple aggregate shape models (sphere and polyhedron) are concerned, this estimation can be performed computing exactly the intersection volume between the finite element and the aggregate. However, this can be tough even for simple shapes, and this can become intractable for more complex and realistic shapes. We therefore resort to a subsampling process to estimate the intersection volume: a given number of sample points are distributed into the finite element, and the number of points lying inside the aggregate is counted. This only requires to be able to test whether a point is inside or outside the aggregate. Thus this process is applicable to any aggregate shape, provided that the geometry can be mathematically described using for example a level-set function (field >0 inside the aggregate, $=0$ on the boundary and <0 outside). The number of sample points needs to be sufficient enough to provide an accurate estimation of the local volume fraction of aggregates.

Table 3
Estimates of the "physical" effective modulus from numerical homogenization computations (see Fig. 3).

E_i/E_m	$C^{eff}_{1111}(h\to 0)/E_m$	Scheme chosen to perform the extrapolation
10^{-8}	0.45	Voigt
3	1.8	Either Voigt or Reuss
10^{2}	4.6	Reuss

The next step is to associate to each finite element a proper elastic stiffness. While there is no ambiguity for the elements sitting in either pure matrix or pure inclusion; the case of so-called "gray" elements, intersecting both matrix and inclusion(s), has to be carefully considered. We have investigated several functions (so-called "schemes" by reference to semi-analytical homogenization schemes, even if scale separation is far from being reached here) providing the stiffness tensor of the element as a function of the elementary stiffness tensors of the different phases and of the local volume fraction of aggregates:

- minimum of stiffness tensors in the sense of quadratic forms (Hill inf),
- maximum of stiffness tensors in the sense of quadratic forms (Hill sup),
- weighted average of stiffness tensors (Voigt),
- inverse of weighted average of inverse of stiffness tensors (Reuss).

A classical FEM computation, using Code_Aster [1], is then performed on the representative volume element, and an estimate of the effective stiffness is derived from the average strain and stress fields in the domain. In a continuum mechanics framework (that is, neglecting

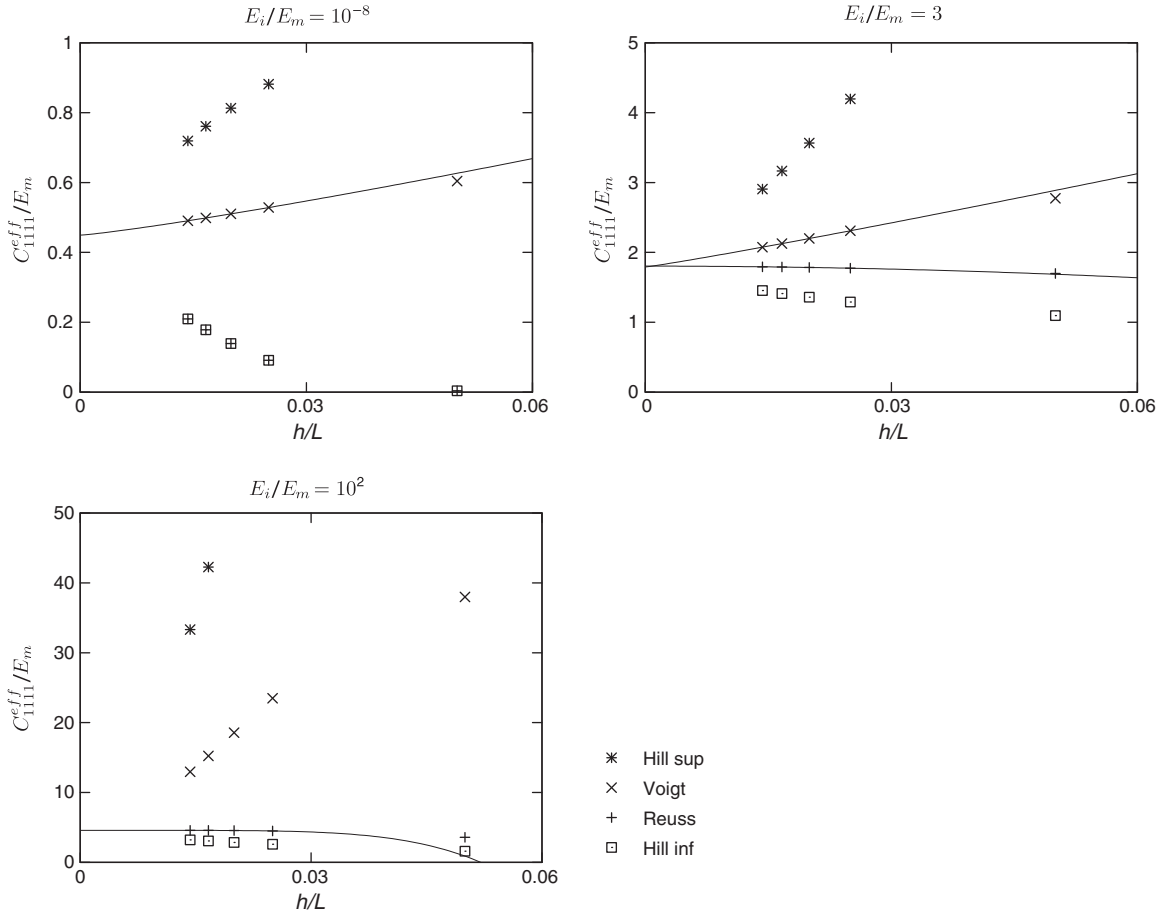

Fig. 3. Component 1,1,1,1 of the estimated effective stiffness tensor of a microstructure made up of 2024 spheres packed into a cube of length L (see Fig. 8 right), as a function of the mesh element size (h). Four schemes (Hill inf, Hill sup, Voigt, and Reuss) have been investigated. Three different contrasts E_i/E_m (ratio of the Young's moduli of the inclusion and the matrix) have been considered. The curves correspond to power–law fits on the results coming from the 4 densest meshes (only for Voigt and Reuss schemes).

Table 4
Reduction of memory consumption reached by PCG variants investigated on a 256^3 grid.

PCG variant	RAM reduction wrt. classical PCG	Time increase wrt. classical PCG
Classical	0%	0%
Matrix-free	55%	8.6%
Matrix-free low cost	66%	62%

Table 5
Influence of the pre-conditioner in the case of a sphere into a cube (contrast 100), on a 256^3 grid, the criterion on the residual relative to the right hand side being 10^{-6}.

Pre-conditioner	Number of iterations	Computing time (s)
None	4281	4040
Jacobi	471	506
ICC	321	496
Multi-level	251	640

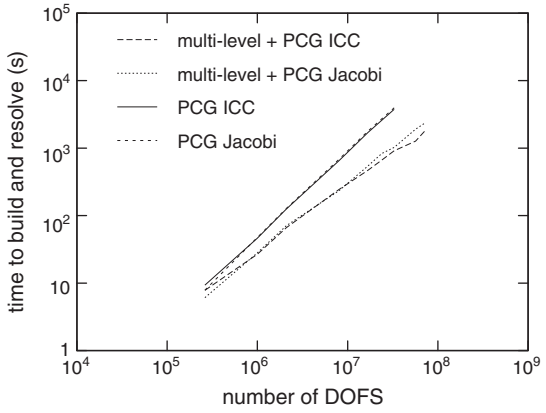

Fig. 4. Effects of multi-level initialization on PCG CPU time in the case of a complex micro-structure: the higher the discretization, the better is the effect.

the errors induced by the FEM discretization), note that the Hill inf and Hill sup schemes respectively yield lower and upper bounds on the effective moduli [12].

As expected, for given microstructure and matrix/inclusions moduli, the effective stiffness estimate mainly depends on two numerical "parameters": the mesh element size and the scheme used to associate a stiffness to the gray elements. The estimated effective stiffness is expected to tend towards the "physical" (as opposed to the estimates coming from a discretization of the microstructure) effective stiffness of the modeled material, when the mesh element size tends towards 0, regardless of the scheme chosen (see Fig. 3). Proposing an empirical power–law fit ($C^{eff}_{1111}/E_m = C^{eff}_{1111}(h\to 0)/E_m + a(h/L)^p$), some estimates of the effective stiffness of the material can be derived (see Table 3). Depending on the contrast, some schemes are found to perform better than others. At low contrast, the Voigt scheme has been used to estimate the asymptotic effective stiffness, whereas at high contrast the Reuss scheme has been used. A more precise analysis of the results can give some understanding of the dependence to mesh element size. The sensibility is smaller and the extrapolation is better when gray element behavior is closer to the average property: in the case of contrasts larger than 1, gray elements are closer to the average behavior when they are computed with Reuss scheme which provides rather small moduli values that are close to the average modulus. Interestingly, at moderate contrast ($E_i/E_m = 3$), both the Voigt and the Reuss schemes provide a very similar estimate.

Clearly enough, as far as "realistic" microstructures are concerned, the extrapolation of the effective stiffness is delicate at extreme contrasts (both low and high, with respect to 1). This fact has even been experienced by [18] on elasticity (that is at "natural" contrasts): numerical modeling of elasticity of a cement paste using FEM yields results within a dispersion of about 25% depending on the type of the finite element considered, and about 40% depending of the amount of mesh refinement. Computations using rather small mesh sizes are therefore required to accurately extrapolate. Unfortunately, generic FEM codes seem to be too RAM consuming. For example, on a computer embedding 8 GB of RAM, we only managed to reach $h/L = 0.014$, which, on the considered microstructure corresponds to a volume fraction of gray elements of 0.40 which is unfortunately still rather large (23%). Alternative solvers are required.

2.2. An alternative and more efficient approach: finite differences and finite volumes on regular grids

The idea is to take advantage of the structure of the regular grid, in particular avoiding the computation and the storage of a mesh, to reduce both RAM consumption and computing time, and investigating higher discretization levels in order to reach better accuracy. As a first attempt, we restrict our investigations to the framework of linear conductivity (to model heat transfer for example). The behavior of the phases is supposed to be isotropic.

The discretization of conductivity problems on regular grids using either finite differences or finite volumes being classical, the latter is not recalled; only the structure of the obtained matrix is described. We rather focus on the resolution of the linear system arising from the discretization.

2.2.1. Numerical modeling

The problem to be solved to homogenize conductivity comes from two field equations:

- the heat conservation equation: $div\left(\underline{j}\right) = 0$,
- the Fourier (or Fick)'s law: $\underline{j} = -\lambda \cdot \underline{grad}(T)$,

(T being the temperature, $\underline{j}$ the heat flux and λ the local conductivity field), and boundary conditions which can be either of the homogeneous temperature gradient type (Dirichlet) or of the homogeneous heat flux type (Neumann). Resorting to order 1 finite differences to approximate 1st order derivatives, the discretization of this problem in the framework of either finite differences or finite volumes yields a linear system whose matrix is symmetric positive definite. The latter has a band structure, with a total of 7 bands of width 1, including the main diagonal. Taking the symmetry into account, only 4 bands are stored. From a programming point of view, the matrix is stored as 4 vectors, to save memory. The solving algorithm is then common to both finite differences and finite volumes methods.

2.2.2. Linear system solvers

As a first attempt, the direct solver MUMPS [2] has been tried. Unfortunately and as expected, direct solvers require too much RAM to be applicable to the present case. Iterative solvers are much more appealing. The PETSc library [3–5], implementing a large set of Krylov subspace iterative solvers and pre-conditioners, has been tested. This allowed to reach much finer grids. But the generic matrix structure proposed by PETSc did not allow to take fully advantage of the structure of the matrix of the problem at hand. We ended up implementing a preconditioned conjugate gradient (PCG) solver [6]. Apart from the classical version, two refined variants have been implemented to further reduce RAM requirements and thus reach finer grids on a given hardware. The first one avoids the matrix storage, computing on the fly matrix–vector products. The second one (so-called "low cost" variant) goes further slightly modifying the conjugate gradient implementation to save one temporary vector (thus using a workspace made up of 3 vectors instead of 4). These variants are, as expected, more time

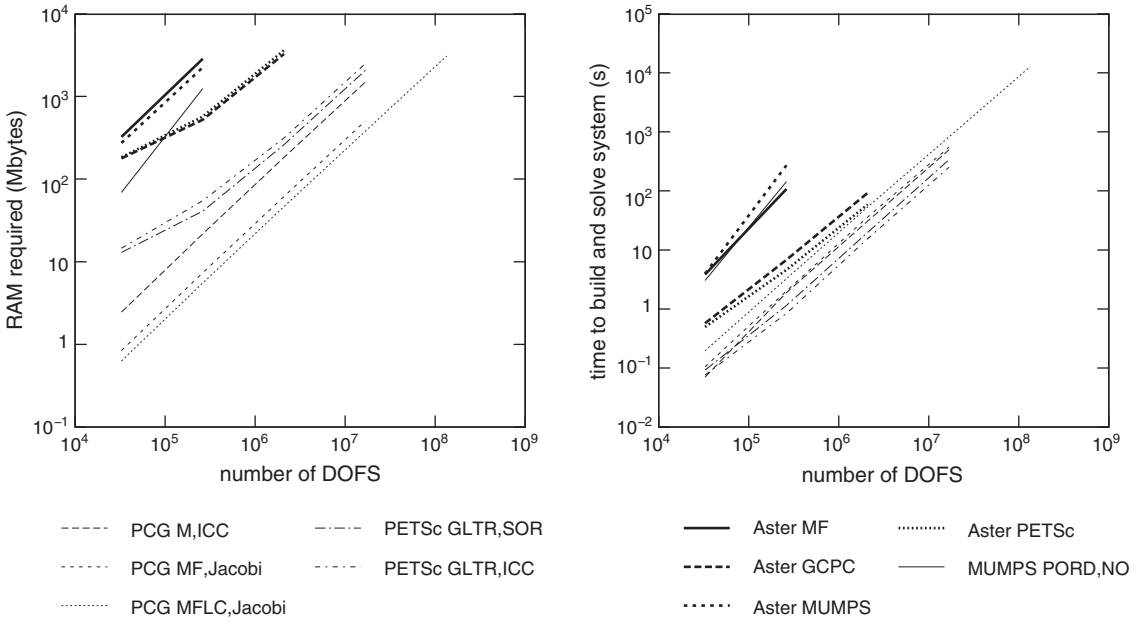

Fig. 5. RAM consumed and time required to build and solve the linear system, as a function of the number of degrees of freedom. As far as PETSc is concerned, the GLTR solver associated to the SOR pre-conditioner minimizes the RAM consumption, and associated to the ICC pre-conditioner, it reduces the CPU time. As far as PCG is concerned, "M,ICC" corresponds to the classical variant, with an ICC pre-conditioner; "MF,Jacobi" corresponds to the matrix-free variant, with a Jacobi pre-conditioner; "MFLC,Jacobi" corresponds to the matrix-free low cost version, with a Jacobi pre-conditioner.

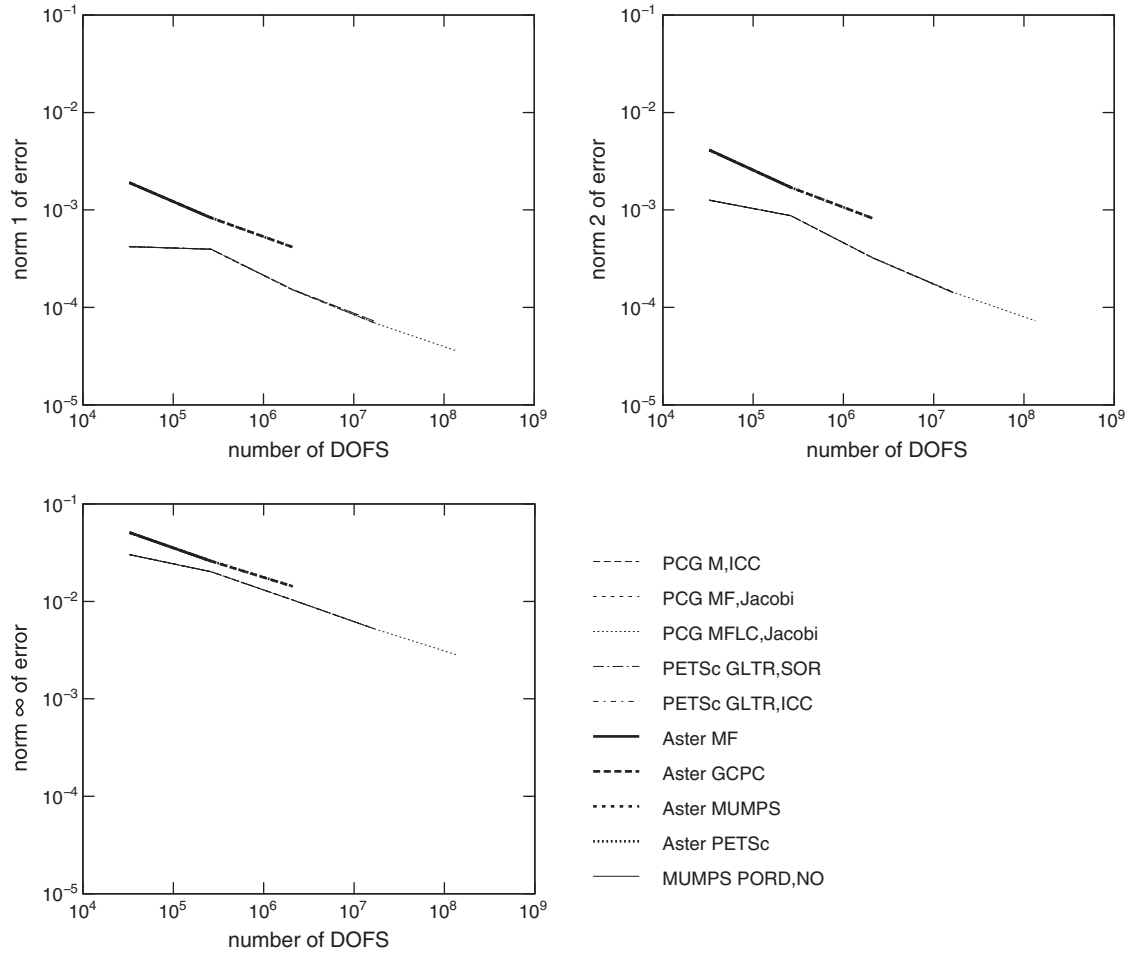

Fig. 6. Error (1, 2, ∞ norms) between the finite differences solution and a reference solution obtained by sampling of a solution computed by FEM on an explicit quadratic mesh of 31.2 Mnodes, as a function of the number of DOFs.

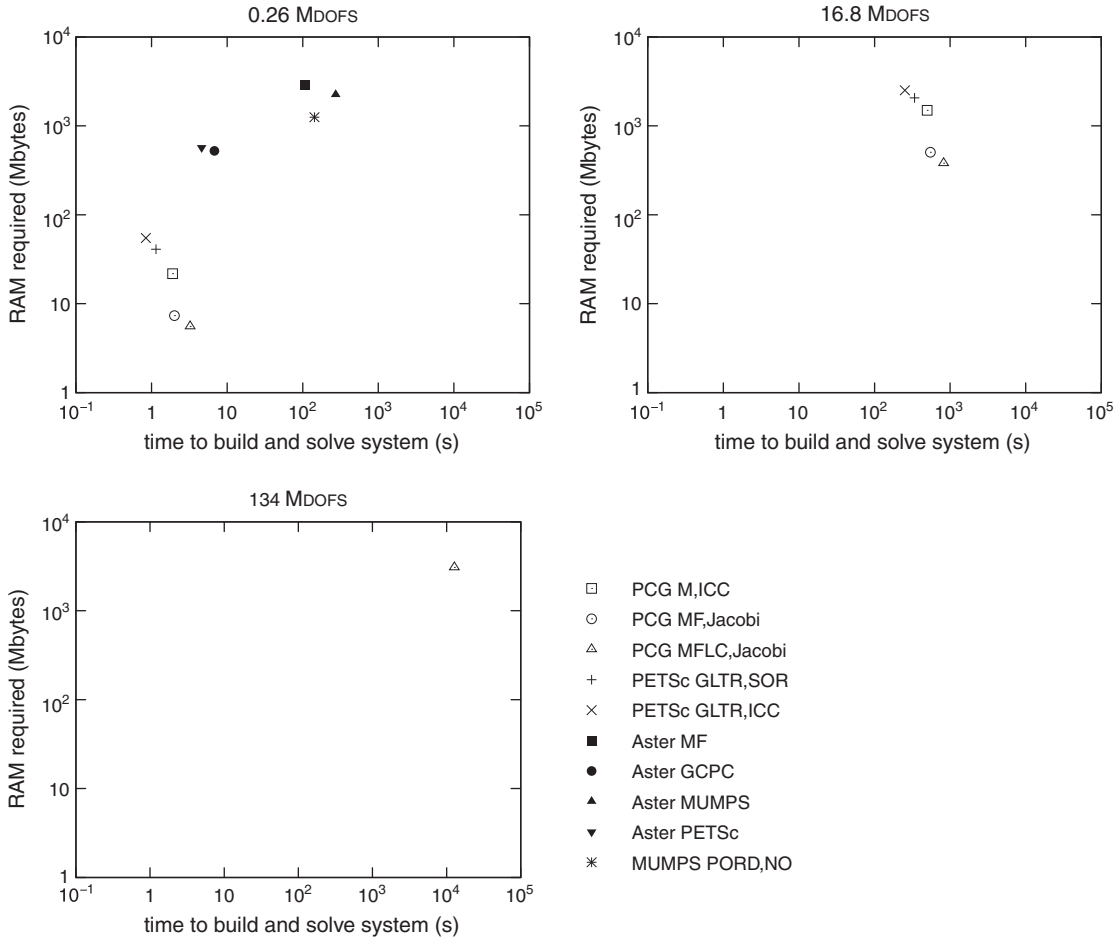

Fig. 7. RAM and time required, at fixed numbers of DOFs.

consuming (see Table 4). However, as far as the matrix-free variant is concerned, the extra time required is not so high compared to the gain on memory requirements. The "low cost" version should be reserved to "extreme cases" where a refined discretization is required but hardware is severely limited, although the extra time is much less important than what can be expected from a classical conjugate gradient algorithm starting to use the swap space on hard-disk.

Pre-conditioners are also common to both approaches (finite differences and finite volumes). Two pre-conditioners have been investigated:

- Jacobi [6] (1 band: inverse of the diagonal),
- incomplete Cholesky (ICC) (5 bands in this case: provides a better estimate of the inverse).

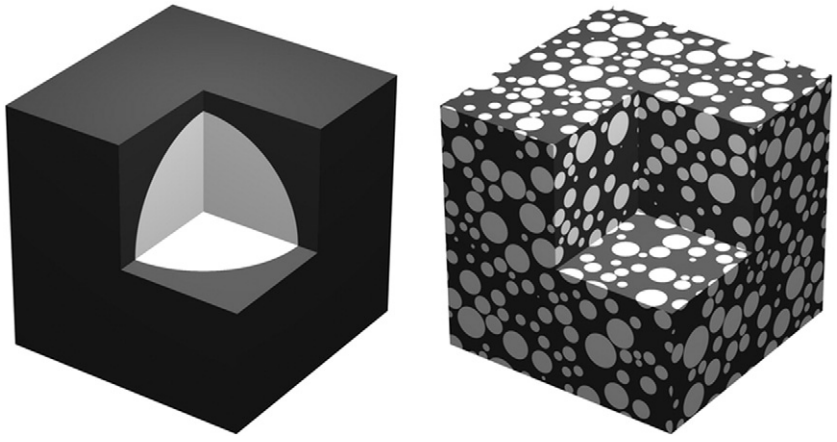

Fig. 8. Two microstructures considered: sphere centered into a cube, and 2024 spheres packed into a cube.

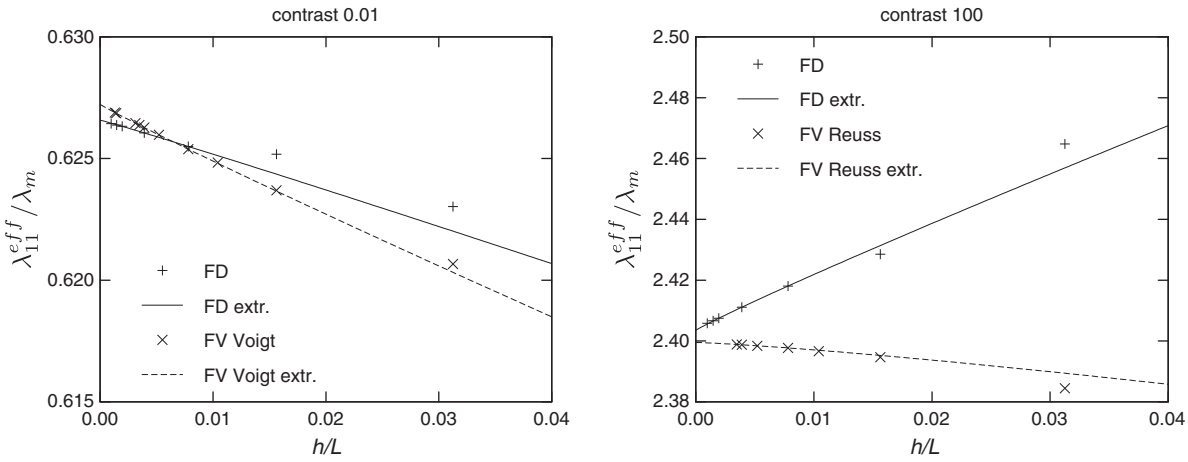

Fig. 9. Results on one sphere: points from computations and extrapolation by a power law (on grids having a number of nodes along one edge higher than 128). FD means finite differences, FV finite volumes, resorting to either a Voigt or a Reuss scheme to estimate the conductivity of gray voxels (as in Section 2.1).

Table 6
Relative error between the effective conductivity numerically computed by finite differences and either the extrapolation from a power–law fit (on results at N = 128, 256, 512), or the extrapolation from an affine law fit (on results at N = 256, 512).

N	Power–law extrapolation		Affine law extrapolation	
	Contrast 0.01	Contrast 100	Contrast 0.01	Contrast 100
680	$1.6*10^{-5}$	$2.8*10^{-5}$	$1.8*10^{-5}$	$1.8*10^{-5}$
1024	$3.5*10^{-5}$	$1.0*10^{-4}$	$4.0*10^{-5}$	$6.9*10^{-5}$

Table 7
Relative error between the effective conductivity numerically computed by finite differences and either the extrapolation from the power–law fit on results at $N = 128$, 256, 512, or the extrapolation from the affine law fit on results at $N = 256$, 512.

N	Power–law extrapolation		Affine law extrapolation	
	Contrast 0.01	Contrast 100	Contrast 0.01	Contrast 100
680	$7.9*10^{-5}$	$1.2*10^{-4}$	$6.4*10^{-5}$	$2.1*10^{-3}$
1024	$2.3*10^{-4}$	$4.4*10^{-4}$	$1.6*10^{-4}$	$5.1*10^{-3}$

As expected, the pre-conditioner improve the convergence of iterative solvers, reducing in a significant amount the number of iterations required by the conjugate gradient algorithm (see example on Table 5). Computing time can also be reduced, if the complexity of the pre-conditioner application does not counteract the gain obtained by the decrease of the number of iterations (compare ICC with Jacobi computing times).

In order to accelerate the convergence of the code we have experimented a multi-level method. A better initialization of the PCG can be extrapolated from the solution computed at a coarser level. For example, in the case of 2^n edge discretization that principle could be applied many times. The initialization of the 256^3 voxels system can be given by a 128^3 voxels one, and so on. That step corresponds to an injection step, it is then followed by cycles made of the residual. As a consequence, that initialization reduces the number of PCG iterations to reach a given level of convergence. In more complex cases (cf. Section 4.2), this method runs faster than the classical ones (cf. Fig. 4). ICC and Jacobi pre-conditioner have been tested with this multi-level initialization. In our study ICC runs faster than Jacobi for the larger f problems. For this reason, it has been chosen as the default pre-conditioner of the multi-level method for the largest computations presented in the section dedicated to applications (cf. Section 4).

2.2.3. Post-processing

Once the temperature field computed on the grid, it is possible to evaluate the temperature gradient and heat flux fields. The effective conductivity estimate λ^{eff} is the linear operator relating the spatial averages of theses fields. To each grid size corresponds an effective conductivity estimate. The "physical" effective conductivity, which can be expected from zero-sized grid cells, is called "physical conductivity". The latter is eventually estimated by extrapolation to a zero-sized grid ($h \to 0$).

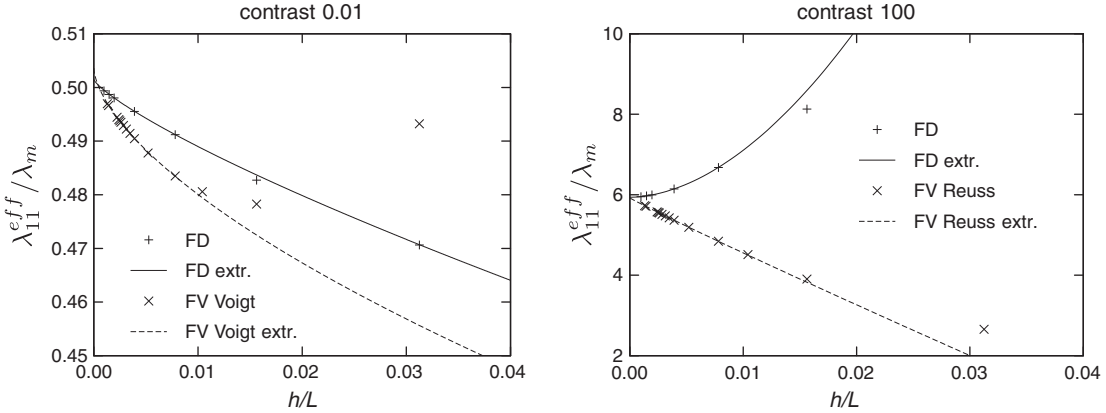

Fig. 10. Results on the 2024 spheres microstructure: points from computations and extrapolation by a power–law (on discretizations higher or equal to 128). FD means finite differences, FV finite volumes, resorting to either a Voigt or a Reuss scheme to estimate the conductivity of gray voxels (as in Section 2.1).

Table 8
Results for the 1 sphere and the 2024 spheres morphologies (the extrapolations from finite volumes computations have been performed using a Voigt scheme at low contrast and using a Reuss scheme at high contrast). Five digits are given on the effective conductivity to highlight the fact that extrapolations from finite differences and finite volumes are very close.

Morphology	1 sph.	1 sph.	1 sph.	1 sph.	2024 sph.	2024 sph.	2024 sph.	2024 sph.
Contrast	100	100	0.01	0.01	100	100	0.01	0.01
Boundary conditions	Gradient	Flux	Gradient	Flux	Gradient	Flux	Gradient	Flux
FV extrapolation	2.3995	2.0935	0.6272	0.5978	5.9222	3.1158	0.5036	0.3316
FD extrapolation	2.4035		0.6266		5.9380		0.5015	
FV exponent p	1.25		0.95		0.96		0.62	
FD exponent p	0.94		1.04		1.84		0.79	

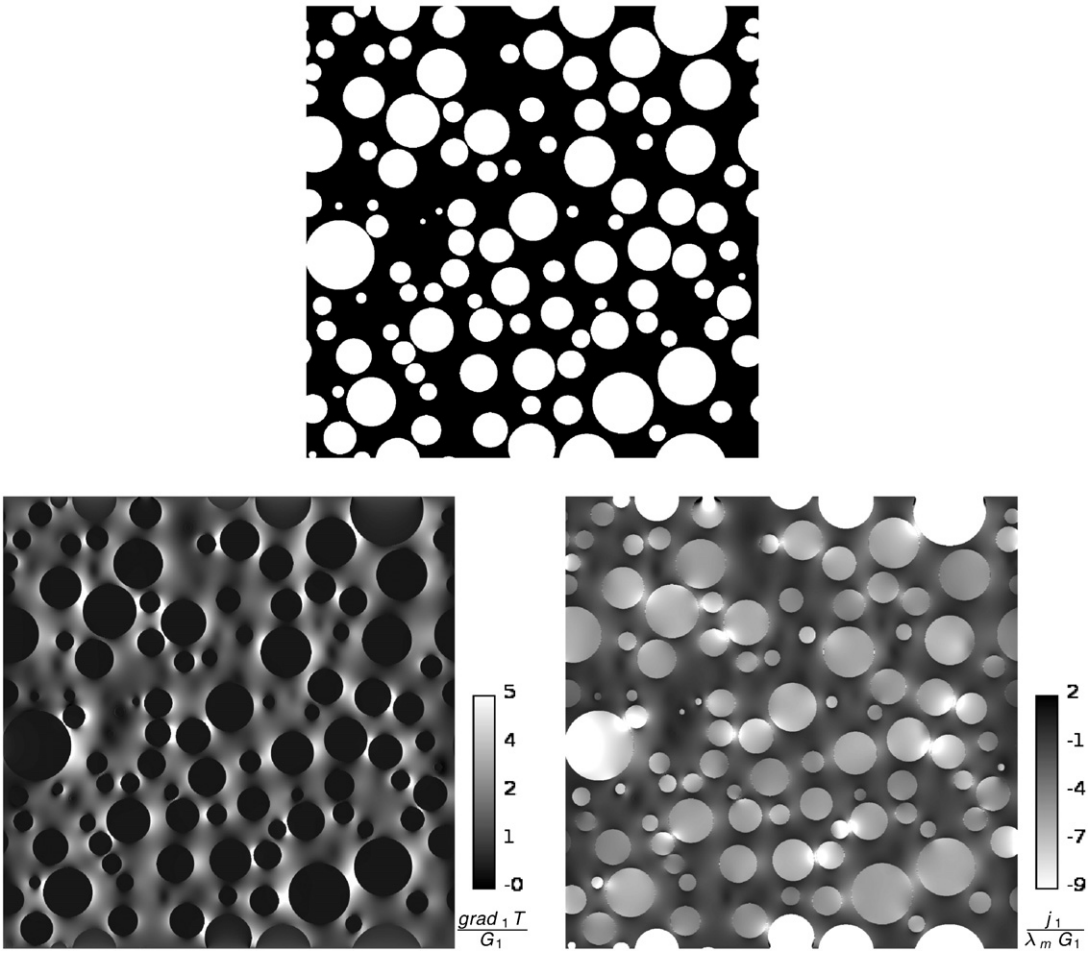

Fig. 11. Microstructure and results (components along $\underline{e}_1$ of the temperature gradient and of the heat flux) on the 2024 spheres microstructure, with a contrast of 100, solved by finite differences on 1024^3 nodes: 2D cut by a plane normal to $\underline{e}_3$.

3. Benchmark: a huge performance scattering

A benchmark has been performed to:

- compare to existing tools the grid solver proposed here,
- in the framework of the grid solver, compare the performances of linear system solvers.

The microstructure is made up of a sphere centered into a cube.[1] The computations have been performed on a Compaq 8710w computer embedding a Core 2 duo T7500 @ 2.2 GHz and 4 GB of RAM. In every case, only one CPU core has been used.

The following tools and solvers have been investigated:

- Code_Aster 10 [1] on a regular mesh (as described in Section 2.1) with the internal solvers MULT_FRONT (denoted MF), GCPC and the external solvers MUMPS, PETSc;
- finite differences on grid and the MUMPS 4.9 [2] library (PORD renumbering allows to minimize both CPU time and RAM consumption);
- finite differences on grid and the PETSc 2.3.3 library [3–5] (among every available couples (solver and pre-conditioner), we chose the ones minimizing either CPU time (GLTR and ICC) or RAM consumption (GLTR and SOR));

[1] Albeit simplistic, this microstructure allows to perform a "reference" computation using FEM on a highly refined explicit mesh.

- finite differences on grid and our own implementation of the PCG (three variants: classical (M), matrix-free (MF) and matrix-free low cost (MFLC)).

As far as Code_Aster is concerned, to remain in the same conditions as finite differences on grid (in terms of DOFs positions), the mesh is a regular Cartesian grid, made up of 8 nodes hexahedra.

The left part of Fig. 5 represents as a function of the number of degrees of freedom the amount of RAM required, for the 10 methods investigated (key is common to both plots of Fig. 5). Two sets of solvers can clearly be outlined. Code_Aster (for every solver) and MUMPS are the most memory demanding. PETSc and PCG are the most lightweight in terms of memory requirements. Note that even the classical "M" variant of PCG requires less memory than PETSc. This is mainly due to the fact that PCG takes fully advantage of the dedicated structure of the matrix, whereas the interface with PETSc is done resorting to generic sparse matrices.

The right part of Fig. 5 represents as a function of the number of degrees of freedom the time required to build and solve the linear system of equations. The fast computations (until a dozen of seconds) have been performed several times and an average has been done, in order to improve the reliability of the time measurement. Two solvers sets can again be defined, but this time with a different arrangement. The direct solvers (Code_Aster + MF, Code_Aster + MUMPS, and MUMPS) are the slowest. The iterative solvers are the fastest. In this latter set, from the slowest to the quickest, we find Code_Aster + PETSc, PCG then PETSc.[2] PETSc performs here slightly better than PCG. Indeed, it surely embeds more efficient and more optimized pre-conditioners and solvers than those implemented into our PCG solver.

Apart from the CPU and RAM performance, it is of paramount importance to check, as the number of DOFs increases, the proper convergence of the numerical solution towards the physically correct temperature field. To check this, a so-called reference result has been computed using FEM on an explicit mesh of the microstructure. Taking symmetries into account, it is possible to mesh only one eighth of the cube. The mesh contains $31.2*10^6$ nodes and $23.6*10^6$ quadratic tetrahedra. The temperature field from FEM has then been sampled on the nodes of the grids used by finite differences.

The norms (1, 2 and ∞) of the difference between the finite difference result and the reference are plotted on Fig. 6 as a function of the number of DOFs. As expected, every norm decreases when the grid is refined. The results are arranged into two groups of nearly superimposed curves, each group corresponding to one method:

- finite elements (regular grid of hexahedron) with Code_Aster,
- finite differences with the tool described in Section 2.1.

To take a broader view on the results of this performance benchmark, we represent on each subgraph of Fig. 7, at fixed number of DOFs, each solver by a point in the plane (time required and memory consumed). The most attractive solvers from the engineer viewpoint are those minimizing both CPU time and RAM.

Let us start by studying the 0.26 MDOFs case, which albeit very modest does not eliminate any of the 10 solvers investigated here. As expected, Code_Aster and the direct solver (MUMPS) are not able to compete with PETSc and PCG. PETSc minimizes the CPU time whereas PCG minimizes the RAM consumed. Note that a huge scattering of performance can be found: a ratio of more than 300 has been found between the slowest and the fastest, and a ratio of more than 500 has been found between the largest and the lowest amount of RAM required. The increase of the number of DOFs does not modify the ordering of the surviving solvers (the limit being here the 4 GB of RAM of the test hardware). At 134 MDOFs (corresponding to a grid of 512^3 nodes), only the lightest method, the low cost variant of PCG, is able to solve the problem.

To conclude, the most efficient (in terms of RAM) is not a generic off-the-shelf solver, but our dedicated PCG.[3] The most efficient in terms of CPU time is PETSc (for small problems because afterwards the available RAM becomes a limiting factor). To highlight the interest of PCG, note that it allowed to perform a computation on a 512^3 grid, with double precision floating point arithmetic, requiring only 3.0 GB of RAM.

4. Applications

The finite differences/volumes code being validated with respect to "reference" results and its performance being evaluated, it is now time to apply it to estimate effective properties of composite materials. Two microstructures are considered (Fig. 8):

- a simple case: a sphere centered into a cube,
- a more realistic case (with respect to the modeling of mortar and concrete-like materials): 2024 spheres packed into a cube and intersecting its faces.

The aim is to investigate whether the asymptotic effective conductivity can be reliably estimated or not.

4.1. A simple case: sphere centered into a cube

The simple case of one sphere (Fig. 8 left) is considered. The estimate of the effective conductivity highly depends on the fineness of the grid. The finer the grid, the closer the estimate is to the extrapolation. See Fig. 9 presenting the estimate of the component 1,1 of the effective conductivity as a function of the relative voxel length $h/L = 1/N$, N being the number of voxels along an edge of the cube, whose length is L. On this figure, points correspond to computations and curves correspond to an empirical interpolation resorting to a power–law, performed on points at $N \geq 128$ (and excluding the two finest discretizations for finite differences to check extrapolability, see Table 6 further on):

$$\lambda_{11}^{eff}(h/L) = \lambda_{11}^{phys} + a(h/L)^p \tag{2}$$

λ_{11}^{phys} corresponds to the physical estimation of the effective conductivity (results are gathered on Table 8). Note that the fitted exponents p are quite close to 1 in every case: see Table 8.

To check the quality (in terms of extrapolability towards small h) of the power law fit for finite differences, we excluded the results obtained on the largest grids ($N = 680$ and $N = 1024$). The relative error between the results from computations and extrapolations are gathered in Table 6. Although increasing with the fineness of discretization, these errors remain low. The errors with respect to an affine extrapolation (from results at $N = 256$ and 512) are also reported. If the interest of the power law fit here seems questionable, it will become much more important on the more realistic microstructure (see next section).

4.2. A more realistic microstructure: 2024 spheres packed into a cube

We then investigated a more realistic microstructure, closer to either mortar or concrete, made up of 2024 spheres packed into a cube (Fig. 8 right). The minimal distance between two inclusions is, normalized by the edge length of the cube, equal to $1.14*10^{-3}$, thus inclusions can come very close one to another. This motivates computations on fine grids.

[2] This benchmark does not take into account the benefit of the multi-level approach.

[3] Indeed, a more generic tool can be expected as being less efficient than a dedicated one. The finite differences/volume code implemented here is obviously limited to regular grids.

The estimation of the component 1,1 of the effective conductivity is represented on Fig. 10 as a function of the relative voxel size $h/L = 1/N$. The power law extrapolation is done on results from $N \geq 128$ (and excluding the two highest discretizations for finite differences to check extrapolability, see Table 7). Indeed, the microstructure being rather dense, we consider that the coarser discretizations are not able to describe accurately enough the microstructure (artificial percolation is found between inclusions): points corresponding to $N < 128$ lie rather far from the curves. The extrapolated estimate of the effective conductivity is reported on Table 8. Note that the fit exponents p are farther from 1 compared to the sphere case (see Table 8).

Considering the lack of reference result, the quality of the power–law fit is checked by comparison between the numerical computation and the extrapolation, at the highest discretizations (Table 7). At highest contrasts, the extrapolation resorting to a simple affine law is clearly not accurate enough.

Plane cuts of results obtained from finite differences on the 1024^3 discretizations of the 2024 spheres microstructure, at a contrast equal to 100, are represented on Fig. 11. Boundary conditions are of the homogeneous temperature gradient type, with $(G_1, G_2, G_3) = (1,0,0)$. As expected, the temperature gradient is rather uniform in inclusions. It would however be interesting to quantify its distribution. The inclusions being 100 times more conducting, the heat flux is more important (in a negative way) in inclusions than in matrix. We also observe that the boundary conditions highly disturb the gradient and flux fields along the boundary of the REV, in a region whose thickness is approximately the size of the largest aggregate. As expected, these perturbations yield differences between the estimates from homogeneous gradient and homogeneous flux boundary conditions.

5. Conclusion and prospects

Quantitative estimation of the effective creep of concrete requires numerical upscalling techniques incorporating a detailed description of the microstructure. This requirement seems to be only fulfilled by numerical homogenization approaches. As the microstructure of concrete is genuinely complex, standard approaches such as meshing explicitly the aggregates are not tractable. An alternative approach consists in superimposing the aggregates onto an independent mesh, thus defining gray elements, whose behavior has to be carefully determined. Unfortunately, the FEM approach is then found to be too RAM consuming to allow a precise extrapolation of the effective stiffness, especially at extreme contrasts. Clearly enough, alternative solvers are due.

Finite differences and volumes on a regular grid were then investigated. We ended up implementing our own variants of the classical PCG algorithm. Our code appears as less demanding in terms of RAM than other approaches tested in Section **3**. This means that for a given hardware, more refined discretizations, thus closer to industrial needs, become accessible. The extrapolations of the effective conductivity of a given microstructure, obtained from two different discretization approaches (namely finite differences and finite volumes) lie remarkably close one to another, for each contrast and microstructure considered (see Table 8). This provides some confidence on the reliability of the proposed numerical homogenization procedure.

However, as far as computing time is concerned, the prototype described here remains largely improvable: on the 2024 spheres microstructure with contrast 100 (the most difficult case), computations on 512^3, 680^3 and 1024^3 DOFs respectively lasted almost 8, 33 and 64 h with PCG. Before going into mechanical modeling, the computing time thus needs to be improved. Several options can be investigated:

- following our first experiment with a multi-level approach, implement the multigrid algorithm [7], either as a solver on its own or as a conjugate gradient pre-conditioner,
- investigate alternative matrix storage strategies and DOFs numberings to improve cache efficiency,
- take advantage of parallelization.

The FFT method [17] also appears as a truly appealing solver candidate. A more extensive comparison of numerical homogenization methods than what is proposed here would be desirable. In this respect, we organized an informal benchmark embracing several teams, approaches, microstructures and types of effective properties, whose results can be found in [8].

Once an efficient and reliable solver for conductivity and elasticity is implemented, it is possible to investigate creep.

References

[1] http://www.code-aster.org/.
[2] http://mumps.enseeiht.fr/.
[3] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Users Manual. ANL-95/11 — Revision 3.0.0, Argonne National Laboratory, 2008.
[4] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Web page, http://www.mcs.anl.gov/petsc, 2009.
[5] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.
[6] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J.M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, Templates for the solution of linear systems: building blocks for iterative methods, Society for Industrial and Applied Mathematics, 1994.
[7] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, SIAM, Philadelphia, 2000.
[8] C. Dunant, A. Giorla, C. Péniguel, J. Sanahuja, C. Toulemonde, A.-B. Tran, F. Willot. J. Yvonnet Estimation of effective behaviour of cementitious materials through numerical homogenization: multi-approach benchmark , submitted for publication.
[9] J.D. Eshelby, The determination of the elastic field of an ellipsoidal inclusion, and related problems. Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences 241 (1226) (1957) 376–396.
[10] D.P. Gaver, Observing stochastic processes and approximate transform inversion, Operations Research 14 (3) (1966) 444–459.
[11] Z. Hashin, Viscoelastic behavior of heterogeneous media, Journal of Applied Mechanics 32 (1965) 630–636.
[12] R. Hill, Elastic properties of reinforced solids: some theoretical principles, Journal of the Mechanics and Physics of Solids 11 (1963) 357–372.
[13] R. Hill, A self-consistent mechanics of composite materials, Journal of the Mechanics and Physics of Solids 13 (4) (1965) 213–222.
[14] T. Kanit, S. Forest, I. Galliet, V. Mounoury, D. Jeulin, Determination of the size of the representative volume element for random composites: statistical and numerical approach, International Journal of Solids and Structures 40 (2003) 3647–3679.
[15] Q. Le, Modélisation multi-échelle des matériaux viscoélastiques hétérogènes; application à l'identification et à l'estimation du fluage propre de béton d'enceintes de centrales nucléaires, Université Paris-Est, Champs-sur-Marne, France, 2008.
[16] T. Mori, K. Tanaka, Average stress in matrix and average elastic energy of materials with misfitting inclusions, Acta Metallurgica 21 (5) (1973) 1605–1609.
[17] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, Computer Methods in Applied Mechanics and Engineering 157 (1998) 69–94.
[18] L. Stefan, F. Benboudjema, J.-M. Torrenti, B. Bissonnette, Prediction of elastic properties of cement pastes at early ages, Computational Materials Science 47 (2010) 775–784.
[19] H. Stehfest, Algorithm 368: numerical inversion of Laplace transforms, Communications of the Association for Computing Machinery 13 (1) (1970) 47–49.
[20] C. Toulemonde, R. Masson, J. El Gharib, Modeling the effective elastic behavior of composites: a mixed finite element and homogenisation approach, Comptes Rendus Mécanique 336 (2008) 275–282.