Comput. Methods Appl. Mech. Engrg. 198 (2009) 2839–2859

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A short and versatile finite element multiscale code for homogenization problems

Assyr Abdulle *, Achim Nonnenmacher

Mathematics Section, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 26 August 2008
Received in revised form 12 January 2009
Accepted 6 March 2009
Available online 7 May 2009

Keywords:
Multiscale method
Macro-to-micro modeling
Microstructure
FEM implementation
Composite material

ABSTRACT

We describe a multiscale finite element (FE) solver for elliptic or parabolic problems with highly oscillating coefficients. Based on recent developments of the so-called heterogeneous multiscale method (HMM), the algorithm relies on coupled macro- and microsolvers. The framework of the HMM allows to design a code whose structure follows the classical finite elements implementation at the macro level. To account for the fine scales of the problem, elementwise numerical integration is replaced by micro FE methods on sampling domains. We discuss a short and flexible FE implementation of the multiscale algorithm, which can accommodate simplicial or quadrilateral FE and various coupling conditions for the constrained micro simulations. Extensive numerical examples including three dimensional and time dependent problems are presented illustrating the efficiency and the versatility of the computational strategy.

1. Introduction

The development of numerical methods for multiscale partial differential equations (PDEs) with highly oscillating coefficients is an increasingly active field of research. Indeed, classical numerical approaches often lead to overwhelming computational complexity for such problems. Despite the flourishing activity on computational homogenization problems or related multiscale problems [8,17,18,22–30] to mention only a few references in the literature, very few open-box computer codes are currently available in this area. In turn, numerical comparisons between different strategies have been rare (we mention [28] for a discussion involving the method discussed in this paper and some other numerical homogenization algorithms). The goal of this paper is to present a short finite element (FE) implementation for multiscale elliptic or parabolic problems. For convenience, we propose a MATLAB implementation, which allows for a concise coding. A FORTRAN or C/C++ version could be easily obtained following the structure of the code given in this paper.

Our code and algorithm, called FE-HMM, is based on the heterogeneous multiscale method (HMM), a framework proposed by E and Engquist [19] for general multiscale problems with scale separation (see also [21] for a recent review). A class of problems for which the HMM has been extensively studied are multiscale PDEs (see [8] for a review). The algorithm described in this paper is based on simultaneous coupling or "on the fly" computations, extracting coarse dynamics from microproblems. In a FEM context, the physical problem is solved directly by a macroscopic solver with input data computed by solving constrained microscopic problems on sampling domains (microscopic solvers).

The algorithm proposed in this paper is short (less than 200 lines) and flexible, easy to use and to modify. At the same time, it is capable to handle challenging 3D problems as can be seen in Section 4, where we describe the heat conduction in a microprocessor package made of composite materials. Generalization for elastic problems can be done following the implementation described here. We believe that such a code can be useful for many practical applications since it can easily be modified by users to account for specific features of their problems. Our multiscale code can handle

- simplicial and quadrilateral FE (at the macroscopic level, i.e., for meshing the physical domain),
- various coupling conditions between macroscopic and microscopic solvers,
- time dependent (parabolic) and 3D problems.

As we will see, a parallelization of the algorithm is straightforward. We propose an implementation for problems with two scales. This could however be generalized to problems with several separated scales. For simplicity, we also limit ourselves to piecewise (bi)linear FE approximations. The code proposed can easily be extended to higher order FE approximation and the algorithm discussed in the first part of the paper is written in this general

* Corresponding author.
E-mail addresses: assyr.abdulle@epfl.ch (A. Abdulle), achim.nonnenmacher@epfl.ch (A. Nonnenmacher).

0045-7825/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2009.03.019

setting. The code and implementation discussed here will be available at http://iacs.epfl.ch/anmc/.

We note that a variety of micro–macro approaches have been proposed in the literature, usually for (nonlinear) elasticity (see [32,17,27] and the references therein). The methods presented in the aforementioned papers are based on the concept of a representative volume element (RVE). Detailed modeling of the RVE is used to determine the unknown parameters of the macroscopic constitutive equations whose solutions give the homogenized material properties. In this perspective, the algorithm proposed in this paper shares some similarities with these strategies. But there are also differences. First, let us note that nonlinear elastic problems are considered in the aforementioned cited papers while we present here a code for linear elliptic/parabolic problems. But the algorithm itself has been analyzed for elastic problems [4] and for nonlinear problems [20] and our code could be extended for these cases. Second, while the latter strategies rely on serial coupling between macro- and microproblems, the present strategy is based on a simultaneous coupling. By serial coupling we mean, given a macroscopic state, to

1. determine appropriate microproblems *beforehand* and compute microsolutions on sampling domains (or RVE) constrained by the *known* macrostate,
2. update the (a priori unknown) macroscopic parameters (e.g. macro stiffness tensor),
3. compute a new macroscopic solution and go back to 1.

By simultaneous coupling we mean to

1. discretize the macroproblem (with a priori unknown parameters),
2. compute (when macro parameters need to be evaluated) microproblems on sampling domains (or RVE), based on the original problem constrained by the *unknown* macrostate[1],
3. compute the macrosolution.

For linear problems there is no iteration in the latter algorithm while for nonlinear problems the iteration is at the macrolevel, similarly as for the FE solution of single scale nonlinear problems. From the implementation point of view, the latter algorithm allows to borrow the coding structure from standard FEM (macrolevel), while the fine scale problem is accounted for in the elements' stiffness matrix calculation (instead of numerical integration, microproblems are solved on sampling domains). This allows to take advantage, at the macrolevel, of the huge variety of FEM developed for single scale problems [8]. This algorithm also allows a priori error estimates of the discretization error coming from macro- and micromeshes and in turn, an indication of the size of the macro- and micromeshes in order to control the global discretization error.

The outline of the paper is as follows: In Section 2, we describe the multiscale elliptic problems we want to solve and recall the algorithm upon which the FE-HMM is built. The FE implementation is detailed in Section 3. Several numerical examples in two and three dimensions, including parabolic problems, with various types of elements and oscillating coefficients (deterministic or random) are given in Section 4 to illustrate the use of our code. We close Section 4 by presenting an application of the code to the computation of the heat distribution in a heatsink and a microprocessor, assumed to be made of composite materials.

Notations. In what follows, $C>0$ denotes a generic constant, independent of ε, whose value can change at any occurrence but depends only on the quantities which are indicated explicitly. For $r=(r_1,\dots,r_d)\in\mathbb{N}^d$, we denote $|r|=r_1+\cdots+r_d, D^r=\partial_1^{r_1}\cdots\partial_d^{r_d}$. We will consider the usual Sobolev space $H^1(\Omega)=\{u\in L^2(\Omega); D^r u\in L^2(\Omega), |r|\leqslant 1\}$, with norm $\|u\|_{H^1(\Omega)}=\left(\sum_{|r|\leqslant 1}\|D^r u\|^2_{L^2(\Omega)}\right)^{1/2}$. We will also consider $H^1_0(\Omega)$ the closure of $C_0^\infty(\Omega)$ for the $\|\cdot\|_{H^1(\Omega)}$ norm and the spaces $W^{l,\infty}(\Omega)=\{u\in L^\infty(\Omega); D^r u\in L^\infty(\Omega), |r|\leqslant l\}$. Finally, for the unit cube $Y=(0,1)^d$, we will consider $W^1_{\text{per}}(Y)=\{v\in H^1_{\text{per}}(Y); \int_Y v\,dx=0\}$, where $H^1_{\text{per}}(Y)$ is defined as the closure of $\mathscr{C}^\infty_{\text{per}}(Y)$ (the subset of $\mathscr{C}^\infty(\mathbb{R}^d)$ of periodic functions in Y).

2. Model problem

The class of problems we want to solve are elliptic problems with coefficients originating from some fine scale structure. We consider a domain $\Omega\in\mathbb{R}^d, d=1,2,3$ with a Lipschitz continuous boundary $\partial\Omega=\partial\Omega_D\cup\partial\Omega_N$ where Dirichlet conditions are imposed on $\partial\Omega_D$ and Neumann conditions on $\partial\Omega_N$. We assume that $\partial\Omega_D$ has positive measure. Given $f\in L^2(\Omega), g_D\in H^{1/2}(\partial\Omega), g_N\in H^{-1/2}(\partial\Omega_N)$ we consider the second-order elliptic equation

$$\begin{aligned}&-\nabla\cdot(a^\varepsilon\nabla u^\varepsilon)=f \quad \text{in } \Omega,\\ &u^\varepsilon=g_D \quad \text{on } \partial\Omega_D,\\ &n\cdot(a^\varepsilon\nabla u^\varepsilon)=g_N \quad \text{on } \partial\Omega_N,\end{aligned} \tag{1}$$

where a^ε is symmetric, satisfies $a^\varepsilon(x)\in(L^\infty(\Omega))^{d\times d}$ and is uniformly elliptic and bounded, i.e.,

$$\exists\lambda,\Lambda>0 \quad \text{such that } \lambda|\xi|^2\leqslant a^\varepsilon(x)\xi\cdot\xi\leqslant\Lambda|\xi|^2, \quad \forall\xi \in\mathbb{R}^d \quad \text{and } \forall\varepsilon, \tag{2}$$

where ε represents a small scale in the problem that characterizes the multiscale nature of the tensor $a^\varepsilon(x)$. An application of Lax–Milgram theorem gives us a family of solutions which are bounded in $H^1_0(\Omega)$ independently of ε and the variational form of (1) is to find $u^\varepsilon\in H^1_D(\Omega)$ such that

$$\begin{aligned}B^\varepsilon(u^\varepsilon,v)&:=\int_\Omega a^\varepsilon\cdot\nabla u^\varepsilon\nabla v\,dx\\ &=\int_\Omega f v\,dx+\int_{\partial\Omega_N} g_N v\,dx-\int_\Omega a^\varepsilon\cdot\nabla g_D\nabla v\,dx=:l^\varepsilon(v)\end{aligned} \tag{3}$$

for all $v\in H^1_D(\Omega)$, where $H^1_D(\Omega):=\{v\in H^1(\Omega); v=0 \text{ on } \partial\Omega_D\}$. A finite element discretization of this variational problem is standard. To avoid technicality with curved boundary and regularity issues with re-entrant corner, we assume in what follows that Ω is a convex polygonal domain in $\mathbb{R}^d$. Let $\mathscr{T}_h$ be a partition of Ω in simplicial or quadrilateral elements K of diameter h_K and denote $h=\max_{K\in\mathscr{T}_h}h_K$. For this partition we define a finite dimensional subspace of $H^1_D(\Omega)$ by

$$V^p_D(\Omega,\mathscr{T}_h)=\{v^h\in H^1_D(\Omega); u^h|_K\in\mathscr{R}^p(K),\ \forall K\in\mathscr{T}_h\}, \tag{4}$$

where $\mathscr{R}^p(K)$ is a suitable space of polynomials with highest degree p (see Section 2.2) and where the partition $\mathscr{T}_h$ is assumed to be regular (see [13] for details). The solution of the discretized problem reads: find $u^h\in V^p_D(\Omega,\mathscr{T}_h)$ such that

$$B^\varepsilon(u^h,v^h)=l^\varepsilon(v^h) \quad \forall v^h\in V^p_D(\Omega,\mathscr{T}_h). \tag{5}$$

There is however a major issue: using a standard FEM for (1) requires to resolve the smallest scale present in the problem, denoted here by ε. This means that the mesh size should satisfy $h<\varepsilon$. If ε is small, the cost associated with the FEM (5) will be prohibitive. This fundamental problem associated with the complexity of discretizing multiscale problems such as (1) has been among the motivation

[1] Note that the unknown macrostate is expanded in a macro basis; while the coefficients of this expansion are unknown (solution of the macroproblem), the basis functions themselves are known and used for the constraints.

to develop homogenization theory and is briefly described in the next section.

2.1. Homogenized problem

Homogenization theory has been an active field of research for the past 30 years. Among the huge literature we mention four books [10,31,16,14] where the interested reader can find more material on the subject including details of the brief discussion which follows.

It is known from homogenization theory that the solution u^ε of (1) converges (usually in a weak sense) to a homogenized solution u^0, solving an elliptic problem

$$\begin{aligned} &-\nabla\cdot(a^0\nabla u^0)=f \quad \text{in } \Omega,\\ &u^0=g_D \quad \text{on } \partial\Omega_D,\\ &n\cdot(a^0\nabla u^0)=g_N \quad \text{on } \partial\Omega_N, \end{aligned} \tag{6}$$

where the so-called homogenized tensor $a^0(x)$ again satisfies $\lambda|\xi|^2\leqslant a^0(x)\xi\cdot\xi\leqslant\Lambda|\xi|^2, \forall\xi\in\mathbb{R}^d$. Under additional assumptions on the small scale such as periodicity, $a^\varepsilon=a(x,x/\varepsilon)=a(x,y)$ is $Y-$periodic in y, where $Y=(0,1)^d$, explicit equations (but in general no analytic solutions) are available to compute the homogenized tensor $a^0(x)$. At each point $x\in\Omega, d$ solutions $\chi^j, j=1,\ldots,d$ of so-called cell problems must be computed and $a^0(x)$ is given by a suitable average (involving integrals) of these χ^j solutions. Thus, even in this fortunate case, $a^0(x)$ must usually be computed numerically, and can therefore only be obtained at discrete values of the domain Ω.

2.2. The finite element heterogeneous multiscale method (FE-HMM)

The so-called finite element heterogeneous multiscale method (FE-HMM) aims at capturing the homogenized (coarse) solution u^0 of (6) without computing $a^0(x)$ explicitly and by relying only on input data given by (1). Furthermore, the knowledge of the oscillatory data of (1) are only needed on sampling domains. This is important since in practice, for many applications, the fine scale structure of the problem can be obtained only on part of the computational domain. We mention problems in material sciences, where the fine scale structure may only be accessible locally by modern scanning and microscopy techniques or in geosciences where required knowledge of a landscape may only be sparsely available.

2.2.1. FE-HMM: the algorithm

The basic FE-HMM can be described as follows.

Macro finite element space. We consider

$$V_D^p(\Omega,\mathcal{T}_H)=\{v^H\in H_D^1(\Omega); u^H|_K\in\mathcal{R}^p(K),\ \forall K\in\mathcal{T}_H\} \tag{7}$$

with macroelements $K\in\mathcal{T}_H$. Here H, the size of the triangulation, is allowed to be much larger than ε. As mentioned in the introduction, we will discuss an implementation for piecewise (bi)linear polynomials, i.e. $p=1$. In this case

- $\mathcal{R}$ is the space of linear polynomials on the element K, if K is a simplicial element (triangle if $d=2$, tetrahedron if $d=3$),
- $\mathcal{R}$ is the space of bilinear polynomials on the element K, if K is a quadrilateral element (quadrilateral if $d=2$, hexahedron if $d=3$).

Quadrature formulas. Within each macroelement $K\in\mathcal{T}_H$ we consider for $\ell=1,\ldots,\mathcal{L}$, where $\mathcal{L}$ is the number of quadrature points

- integration points $x_{K_{\delta_\ell}}\in K$,
- sampling domains $K_{\delta_\ell}=x_{K_{\delta_\ell}}+\delta I$, where $I=(-1/2,1/2)^d$ and $\delta\geqslant\varepsilon$,
- quadrature weights ω_{K_ℓ}.

The set $\{x_{K_{\delta_\ell}},\omega_{K_\ell}\}_{\ell=1}^{\mathcal{L}}$ is a quadrature formula on K chosen such that

$$\int_K q(x)\,dx=\sum_{\ell\in\mathcal{L}}\omega_{K_\ell}q(x_{K_{\delta_\ell}})\,dx \quad \forall q(x)\in\mathcal{R}. \tag{8}$$

For piecewise linear elements $\mathcal{L}=1, \omega_{K_\ell}=|K|$ and $x_{K_{\delta_\ell}}$ is chosen to be located at the barycenter of the simplicial K. For bilinear elements, $\mathcal{L}=4$ and $\{\omega_{K_\ell},x_{K_{\delta_\ell}}\}_{\ell=1}^4$ is the two points Gauss quadrature rule given by $\omega_{K_\ell}=|K|/4, x_{K_{\delta_\ell}}=F_K(1/2\pm\sqrt{3}/6, 1/2\pm\sqrt{3}/6)$, where F_K is the affine mapping such that $F_K(\widehat{K})=K$ and $\widehat{K}=(0,1)^2$.

Remark 1. For higher order FE-HMM, the quadrature formula must be chosen such that standard FEM with numerical integration converges with optimal rate (see [13, Chapter 4.1]). For details about the choice of the quadrature formulas for the FE-HMM for arbitrary orders, we refer to [8].

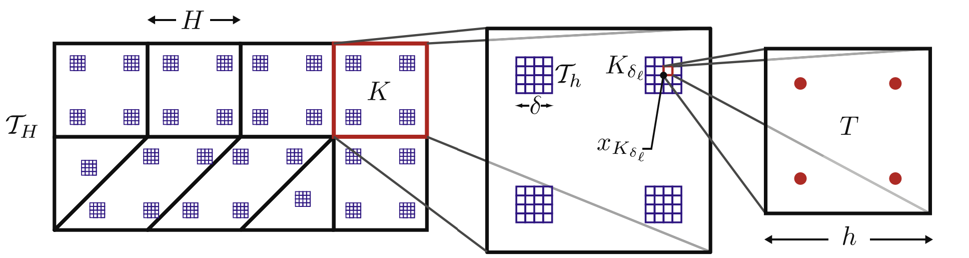
Fig. 2.1 illustrates the quadrature formulas and the coupling between the macro- and microproblem which we describe in the following.

Macro bilinear form. For a discretization in the coarse FE space (7) we need to modify the bilinear form (5). For $v^H, w^H\in V_D^p(\Omega,\mathcal{T}_H)$ we define

$$B(v^H,w^H)=\sum_{K\in\mathcal{T}_H}\sum_{\ell=1}^{\mathcal{L}}\frac{\omega_{K_\ell}}{|K_{\delta_{K_\ell}}|}\int_{K_{\delta_\ell}}a^\varepsilon(x)\nabla v_{K_\ell}^h\cdot\nabla w_{K_\ell}^h\,dx, \tag{9}$$

where $v_{K_\ell}^h, w_{K_\ell}^h$ are appropriate microfunctions defined on sampling domains K_{δ_ℓ} (see below) and the factor $|K_{\delta_\ell}|$ gives the appropriate weight for the contribution of the integrals defined on K_{δ_ℓ} instead of K.

Microsolver. For every macroelement K we compute the additive contribution to the macro stiffness matrix by computing microfunctions $v_{K_\ell}^h$ (and $w_{K_\ell}^h$) obtained by solving microfunctions



$\leftarrow H \rightarrow$
$\mathcal{T}_H$
K
$\mathcal{T}_h$
$\leftarrow\delta\rightarrow$
K_{δ_ℓ}
$x_{K_{\delta_\ell}}$
T
$\leftarrow h \rightarrow$

(a) macro discretization with sampling domains (micro problems)

(b) microproblems

(c) example of quadrature for a microproblem

Fig. 2.1. In every element of the FE-HMM, the contribution to the stiffness matrix of the macroelements (a) is given by the solutions of the microproblems (b), which are computed using numerical quadrature on every microelement (c).

on sampling domains $K_{\delta_\ell}, \ell = 1, \dots, \mathscr{L}$. The microproblems read as follows: find $v^h_{K_\ell}$ such that $(v^h_{K_\ell} - v^H_{lin,K_\ell}) \in S^q(K_{\delta_\ell}, \mathscr{T}_h)$ and

$$\int_{K_{\delta_\ell}} a^\varepsilon(x)\nabla v^h_{K_\ell} \cdot \nabla z^h \, dx = 0 \quad \forall z^h \in S^q(K_{\delta_\ell}, \mathscr{T}_h), \tag{10}$$

where

$$v^H_{lin,K_\ell}(x) = v(x_{K_{\delta_\ell}})^H + (x - x_{K_{\delta_\ell}}) \cdot \nabla v^H(x_{K_{\delta_\ell}}) \tag{11}$$

is a linearization of the macrofunction v^H at the integration point $x_{K_{\delta_\ell}}$ (see [20,8] for details) and

$$S^q(K_{\delta_\ell}, \mathscr{T}_h) = \{z^h \in W(K_{\delta_\ell}); z^h|_T, \in \mathscr{R}^q(T), T \in \mathscr{T}_h\}, \tag{12}$$

where $W(K_{\delta_\ell})$ determines the coupling condition or boundary conditions used for computing the microfunctions $v^h_{K_\ell}$ (or $w^h_{K_\ell}$). We will again set $q = 1$ and since the sampling domain has a simple geometry, we will use quadrilateral FE. Notice that for piecewise linear functions $v^H_{lin,K_\ell} = v^H$. Several choices are possible for the coupling condition we will consider

- $W(K_{\delta_{K_\ell}}) = W^1_{\text{per}}(K_{\delta_\ell})$ and we denote $S^1(K_{\delta_\ell}, \mathscr{T}_h)$ by $S^1_P(K_{\delta_\ell}, \mathscr{T}_h)$; this coupling condition will be referred in what follows as (P),
- $W(K_{\delta_\ell}) = H^1_0(K_{\delta_\ell})$ and we denote $S^1(K_{\delta_\ell}, \mathscr{T}_h)$ by $S^1_D(K_{\delta_\ell}, \mathscr{T}_h)$; this coupling condition will be referred in what follows as (D),

where $W^1_{\text{per}}(K_{\delta_\ell}) = \{v \in H^1_{\text{per}}(K_{\delta_\ell}); \int_{K_{\delta_\ell}} v\,dx = 0\}$, and $H^1_{\text{per}}(K_{\delta_\ell})$ is defined as the closure of $\mathscr{C}^\infty_{\text{per}}(K_{\delta_\ell})$ (the subset of $\mathscr{C}^\infty(\mathbb{R}^n)$ of periodic functions in K_{δ_ℓ}).

Remark 2. For a tensor $a^\varepsilon(x) = a(x, x/\varepsilon)$ with explicit scale separation, it is preferable to collocate the slow variable at the integration points $a(x_{K_{\delta_\ell}}, x/\varepsilon)$ in both the macro and micro bilinear forms (9) and (10). In the periodic case, choosing δ as an integer multiple of ε gives robust, i.e. independent of ε, convergence results (see [2, App. A]).

Variational problem. The macrosolution of the FE-HMM is defined by the following variational problem: find $u^H \in V^p_D(\Omega, \mathscr{T}_H)$ such that

$$B(u^H, v^H) = \int_\Omega f v^H \, dx + \int_{\partial\Omega_N} g_N v^H \, dx - B(g_D, v^H) \quad \forall v^H \in V^p_D(\Omega, \mathscr{T}_H). \tag{13}$$

The primary goal of this method is to capture the effective solution u_0 of (6), i.e., u^H converges to u^0 as H goes to zero. Remember that the method also depends on a micromesh, thus h going to zero is also necessary for the above convergence (see Section 2.2.2).

Post-processing procedure. While u^H converges to u^ε in the L^2 norm, a convergence of u^H to u^ε in the energy (H^1) norm is not guaranteed in general. Indeed, the oscillations of u^ε introduce $\mathscr{O}(1)$ perturbations in the gradient which are not captured by u^H. An approximation of u^ε in the energy norm can nevertheless be obtained from the macrosolution u^H by using a post-processing procedure. The known small scale solution in the sampling domain (10) computed during the assembly of the FE-HMM can be extended locally on the macroelement K and added to u^H. This is done as follows (for simplicial elements): define $u^{H,\varepsilon} : \Omega \to \mathbb{R}$ given by its restriction on each macroelement K by

$$u^{H,\varepsilon}(x)|_K = u^H(x) + (u^h - u^H)(x - [x]_{K_{\delta_\ell}}) \quad \text{for } x \in K \in \mathscr{T}_H, \tag{14}$$

where for $x \in \mathbb{R}^d$, $[x]_{K_{\delta_\ell}}$ denotes the unique combination $\delta\sum_{i=1}^d b_i e_i$, where $b_i \in \mathbb{Z}$ and $(e_i)_{i=1}^d$ is the canonical basis of $\mathbb{R}^d$, such that $(x - [x]_{K_{\delta_\ell}}) \in K_{\delta_\ell}$, see Fig. 2.2 for an illustration. Notice that $u^{H,\varepsilon}(x)$ can be discontinuous across the macroelements K.

2.2.2. FE-HMM: a priori estimates

First one shows that there exists a unique solution of the problem (13). This can be done without specific assumptions on a^ε (of course we assume (2)). Second, to derive a-priori estimates, some knowledge of the homogenized problem is needed. Under periodicity assumptions for a^ε the following can be shown

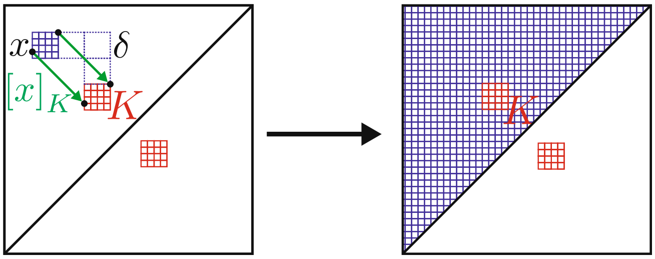


Fig. 2.2. Post-processing procedure: the small scale solution of the sampling domain is extended locally on the macroelements K as given in Eq. (14).

$$\|u^0 - u^H\| \leqslant e_{\text{MAC}} + e_{\text{MOD}} + e_{\text{MIC}},$$

where $e_{\text{MAC}}, e_{\text{MOD}}, e_{\text{MIC}}$ stand for macro error, modeling error and micro error. The various components of the error can be estimated as follows (written here for piecewise (bi)linear macro and micro FE).

- e_{MAC} is the error of a standard FE approximation in the space $V^p_D(\Omega, \mathscr{T}_H)$. For example for $p = 1$ we have $e_{\text{MAC}} \leqslant C \cdot H$ in the H^1 norm and $e_{\text{MAC}} \leqslant C \cdot H^2$ in the L^2 norm,
- e_{MIC} is the error of a FE approximation in the space $S^q(K_{\delta_j}, \mathscr{T}_h)$ transmitted through the macro bilinear form. For example for $q = 1$ we have $e_{\text{MIC}} \leqslant C\left(\frac{h}{\varepsilon}\right)^2$ for *both* the H^1 and the L^2 norm,
- e_{MOD} depends on the coupling of the macro and micro FE spaces and the size of the sampling domain K_{δ_ℓ}, but it is independent of the macro and micro mesh sizes. For a periodic problem with $W(K_{\delta_\ell}) = W^1_{\text{per}}(K_{\delta_\ell})$ (see (12)) and $\delta = \varepsilon$, we have $e_{\text{MOD}} = \varepsilon$.[2] For a periodic problem with $W(K_{\delta_\ell}) = H^1_0(K_{\delta\ell})$ and $\delta > \varepsilon, e_{\text{MOD}} = \delta + \varepsilon/\delta$.

Higher order macro- and microsolvers are discussed in [6,8,20]. For the post-processed solution, using $W(K_{\delta_\ell}) = W^1_{\text{per}}(K_{\delta_\ell})$ for the FE-HMM, we obtain

$$\|u^\varepsilon - u^{H,\varepsilon}\| \leqslant C\left(H + \sqrt{\varepsilon} + \frac{h}{\varepsilon}\right),$$

where $\|\cdot\|$ denotes here the broken norm $\left(\sum_{K\in\mathscr{T}_H}\|\nabla u\|^2_{L^2(K)}\right)^{1/2}$. Details and proof of the above estimates can be found in [2–5,8,20].We emphasize on three aspects of the FE-HMM.

- First and most important for the computational aspects, the fully discrete a priori estimates described above and first obtained in [3] show that refining the macro- and the micromeshes simultaneously is necessary for convergence. These estimates also give important information about the rate at which macro- and micromeshes have to be refined and about the complexity of the FE-HMM. We also notice that mesh refinement has no influence on the modeling error e_{MOD}.
- Second, the algorithm presented above is not restricted to continuous FE for the macro- and microsolvers and we can accommodate other types of macro- and microsolvers. Such an extension was proposed in [6], where a multiscale method with microsolvers based on spectral methods has been investigated. A fully discrete analysis reveals exponential convergence of the microsolver (provided sufficient regularity in the microscales) and in turn a near optimal computational complexity (linear complexity in the macro degrees of freedom). Another

[2] For a tensor $a^\varepsilon(x) = a(x, x/\varepsilon)$ with explicit scale separation by collocating the slow variable at integration points and choosing δ as an integer multiple of ε one has $e_{\text{MOD}} = 0$ (see Remark 2).

extension has also been given in [7], where a multiscale method with a macrosolver based on a discontinuous Galerkin FE space has been developed.

- Third, the FE-HMM described below is well-posed and the method is applicable also in the non-periodic case and convergence rates describing the macro and micro error can be given also for this case. However, scale separation is needed for the macro to micro strategy to make sense and the third component of the error, the so-called modeling error, has so far only be characterized in the (non-uniformly) periodic and random cases [20,8].

3. Implementation

For simplicity of notations, we describe the implementation for two-dimensional problems, and comment on the subtleties which occur for three-dimensional problems when appropriate. We emphasize that our code can handle both two and three-dimensional problems. We also restrict ourselves to a piecewise linear or bilinear macro FE approximation and emphasize again that modification of our solver to accommodate higher order FEM can easily be obtained.

Data representation of the macro triangulation. We recall that a data-representation of a triangulation $\mathcal{T}_h$ of a domain G requires the following input data

- `Coordinates.dat` containing in line i: x-coordinate, y-coordinate of node number i,
- `Elements3.dat` and `Elements4.dat` containing in line i the node numbers of the ith triangular or quadrilateral elements, node1 node2 node3 (for a triangle) and node1 node2 node3 node4 (for a quadrilateral), respectively.
- `Dirichlet.dat` and `Neumann.dat` containing the node numbers of boundary nodes.

If $G = \Omega$ we will use the above notation, if $G = K_\delta$ (sampling domain) we will use the prefix `Micro`.... The data representation used is standard unless explicitly mentioned and we refer for example to [9] for details.

Macro and micro basis functions. In what follows we denote by

- $\{\varphi_m^H\}_{m=1}^{M_{\text{mac}}}$ the basis of the macro FE space $V_D^1(\Omega, \mathcal{T}_H)$ defined in (7), where M_{mac} is the number of discretization points in the macrodomain; the elements (simplicial or quadrilateral) of the macro triangulation $\mathcal{T}_H$ are denoted by K,
- $\{\psi_{m,K_\ell}^h\}_{m=1}^{M_{\text{mic}}}$ the basis of the micro FE space $S^1(K_{\delta_\ell}, \mathcal{T}_h)$ defined in (12) with $q = 1$, where $M_{\text{mic}} = (N_{\text{mic}})^d$ $(d = 2, 3)$ and N_{mic} is the number of discretization points in each direction of the d-dimensional micro sampling domain; the elements (simplicial or quadrilateral) of the micro triangulation $\mathcal{T}_h$ are denoted by T.

3.1. Core structure: assembling the macroproblem

Following a standard assembly process for the macro bilinear form (9), we compute the contribution A_K to the stiffness matrix associated with the macroelement K. We identify the macro basis functions with non-zero support in K denoted by $\varphi_i^H, i = 1, \dots, \mu_K$ and determine A_K based on contributions computed on sampling domains $K_{\delta_\ell} \subset K$. The process reads as follows:

$$A_K := B(\varphi_i^H, \varphi_j^H)_{i,j=1}^{\mu_K} = \left(\sum_{\ell=1}^{\mathcal{L}} \frac{\omega_{K_\ell}}{|K_{\delta_\ell}|} \int_{K_{\delta_\ell}} a^\varepsilon(x) \nabla \varphi_{K_\ell,i}^h \cdot \nabla \varphi_{K_\ell,j}^h \, dx \right)_{i,j=1}^{\mu_K} \quad (15)$$

$$= \underbrace{\sum_{\ell=1}^{\mathcal{L}} \frac{\omega_{K_\ell}}{|K_{\delta_\ell}|} (\mathcal{A}_{K_\ell}^T A_{mic,K_\ell} \mathcal{A}_{K_\ell})}_{\texttt{hmm\_stima\_type\_of\_K}}, \quad (16)$$

where `type_of_K` refers to the type of the macroelement K, i.e., `type_of_K=tri` for triangular elements or `type_of_K=quad` for quadrilateral elements. Notice that A_{mic,K_ℓ} is a $M_{\text{mic}} \times M_{\text{mic}}$ matrix and $\mathcal{A}_{K_\ell}$ a $M_{\text{mic}} \times \mu_K$ matrix. Furthermore, $\varphi_{i,K_{\delta_\ell}}^h$ is the microfunction solution of (10) such that $(\varphi_{i,K_{\delta_\ell}}^h - \varphi_{lin,i,K_{\delta_\ell}}^H) \in S^1(K_{\delta_\ell}, \mathcal{T}_h)$ and $\varphi_{lin,i,K_{\delta_\ell}}^H$ is the linearization (11) of the basis function φ_i^H.

We then compute the contribution to the right-hand side for the triangle K as

$$b_K = \int_K f \varphi_i^H \, dx \simeq f(x_{K_b}) \int_K \varphi_i^H \, dx, \quad (17)$$

where $x_{K_b} \in K$ is an integration point located at the barycenter of K and $\int_K \varphi_i^H \, dx$ can be computed explicitly (its value is for example $|K|/6$ if K is a triangle or $|K|/4$ if K is a quadrilateral). Upon a suitable numbering of the macro nodes, the system corresponding to the macrosolution can be written as

$$\underbrace{\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}}_{A} \begin{pmatrix} U \\ U_D \end{pmatrix} = \underbrace{\begin{pmatrix} b_I \\ b_D \end{pmatrix}}_{b}, \quad (18)$$

where U are the free macro nodes to be computed and U_D the values (known a priori) of the nodes at the portion of the boundary enforced with Dirichlet conditions. The first block of the above system of equations gives the desired macro solution

$$A_{11} U = b_I - A_{12} U_D.$$

The above procedure as well as the incorporation of the boundary conditions for the macro problem are standard and we follow the structure of the MATLAB implementation described in [9]. A flow chart of the algorithm is shown in Fig. 3.1.

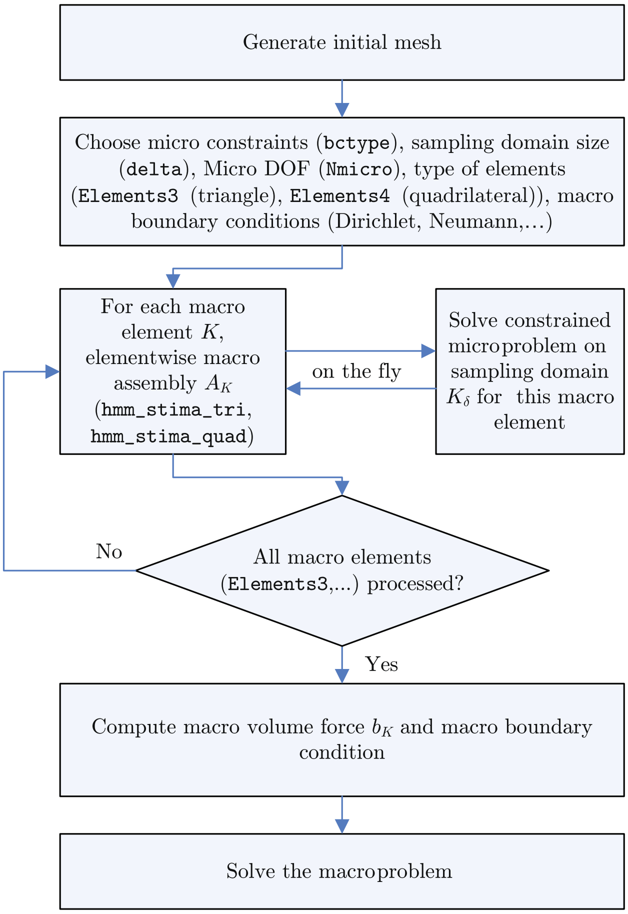

Generate initial mesh
Choose micro constraints (bctype), sampling domain size (delta), Micro DOF (Nmicro), type of elements (Elements3 (triangle), Elements4 (quadrilateral)), macro boundary conditions (Dirichlet, Neumann,...)
For each macro element K, elementwise macro assembly A_K (hmm_stima_tri, hmm_stima_quad)
on the fly
Solve constrained microproblem on sampling domain K_δ for this macro element
No
All macro elements (Elements3,...) processed?
Yes
Compute macro volume force b_K and macro boundary condition
Solve the macroproblem

Fig. 3.1. Flow chart for FE-HMM algorithm.

While for classical FEM, `stima_tri` or `stima_quad` are given by a suitable elementwise integration, for the FE-HMM, `stima_tri` or `stima_quad` are given by a suitable contribution of micro functions computed on sampling domains, constrained by the basis functions associated with the element K. This is explained in detail in the next section.

We note that the solution of the constrained microproblem on one macroelement is independent of the constrained microproblem on a different macroelement. The macro assembly is thus well-suited for parallel implementation.

3.2. Local assembly and microsolutions on sampling domains

In order to compute the matrix A of (18) we have to compute for each macroelement K the local macro contribution A_K (see (16)) to the stiffness matrix. This assembly is based upon suitable solutions of microproblems on sampling domains $K_{\delta_\ell} \in K$.

Data representation of the micro triangulation. Since the sampling domains K_{δ_ℓ} have a simple shape (square or cube) we choose a quadrilateral mesh $\mathcal{T}_h$ with $M_{\text{mic}} = (N_{\text{mic}})^d (d = 2, 3)$ grid points. The `MicroCoordinates` must be computed around the integration points of each macroelement and is implemented in `micromesh_coords`.

The `MicroElements` do not depend on the `MicroCoordinates` and are computed once at the initialisation of the code, which is done in `micromesh_elements`.

Structure of the micro assembly process. To have the flexibility to incorporate various micro boundary conditions we will use Lagrange multipliers to compute φ^h. Given a sampling domain $K_{\delta_\ell} \subset K$ and an associated linearized macro basis function $\varphi^H_{lin,i,K_{\delta_\ell}}$ we consider instead of (10) the minimization problem

$$\varphi^h_{i,K_{\delta_\ell}} = \operatorname{argmin} \int_{K_{\delta_\ell}} a\left(x_{K_{\delta_\ell}}, \frac{x}{\varepsilon}\right) \nabla w^h \cdot \nabla w^h \, dx \tag{19}$$

over all function $w^h \in S^1(K_{\delta_\ell}, \mathcal{T}_h) = \{z^h \in H^1(K_{\delta_\ell}); z^h|_T \in \mathcal{R}(T), T \in \mathcal{T}_h\}$, such that

- $w^h - \varphi^H_{lin,i,K_{\delta_\ell}} \in S^1_P(K_{\delta_\ell}, \mathcal{T}_h)$ (periodic coupling) or
- $w^h - \varphi^H_{lin,i,K_{\delta_\ell}} \in S^1_D(K_{\delta_\ell}, \mathcal{T}_h)$ (Dirichlet coupling).

Remark 3. If we do not have explicit scale separation, i.e. $a^\varepsilon(x)$ instead of $a\left(x_{K_{\delta_\ell}}, \frac{x}{\varepsilon}\right)$, we can insert a dummy parameter for $x_{K_{\delta_\ell}}$, such as $x_{K_{\delta_\ell}} = 0$ and only use the second parameter $\frac{x}{\varepsilon}$ in the tensor function of the MATLAB code.

To compute (19) we expand $\varphi^H_{lin,i,K_{\delta_\ell}}$ in the basis of $S^1(K_{\delta_\ell}, \mathcal{T}_h)$

$$\varphi^H_{lin,i,K_{\delta_\ell}} = \sum_{m=1}^{M_{\text{mic}}} \beta_{i,m} \psi^h_{m,K_{\delta_\ell}}(x) \tag{20}$$

and the above minimization problem leads, by introducing Lagrange multipliers, to a saddle point problem given in linear form by (see [2, Section 5])

$$\begin{aligned} &A_{mic,K_{\delta_\ell}} \boldsymbol{\alpha}_{i,K_{\delta_\ell}} + D^T \lambda = 0, \\ &D(\boldsymbol{\alpha}_{i,K_{\delta_\ell}} - \boldsymbol{\beta}_{i,K_{\delta_\ell}}) = 0, \end{aligned} \tag{21}$$

where $\boldsymbol{\alpha}_{i,K_{\delta_\ell}} = (\alpha_{i,1}, \ldots, \alpha_{i,M_{\text{mic}}})^T, \boldsymbol{\beta}_{i,K_{\delta_\ell}} = (\beta_{i,1}, \ldots, \beta_{i,M_{\text{mic}}})^T$, and λ is the Lagrange multiplier. We then have

$$\varphi^h_{i,K_{\delta_\ell}} = \sum_{m=1}^{M_{\text{mic}}} \alpha_m \psi^h_{m,K_{\delta_\ell}}. \tag{22}$$

$A_{mic,K_{\delta_\ell}}$ is the micro stiffness matrix with entries given by

$$(A_{mic,K_{\delta_\ell}})_{mn} = \int_{K_\delta} a\left(x_{K_{\delta_\ell}}, \frac{x}{\varepsilon}\right) \nabla \psi^h_{m,K_{\delta_\ell}} \cdot \nabla \psi^h_{n,K_{\delta_\ell}} \, dx, \tag{23}$$

and D is the matrix of the constraints detailed below.

Assemble and solve the constrained microproblem. The problem (21) is solved for each basis function $\varphi^H_{lin,i,K_{\delta_\ell}}, i = 1, \ldots, \mu_K$ and we define the $M_{\text{mic}} \times \mu_K$ matrix $\mathcal{A}_{K_\ell}$ with columns given by $\boldsymbol{\alpha}_{i,K_{\delta_\ell}}$, i.e., $\mathcal{A}_{K_\ell} = (\boldsymbol{\alpha}_{1,K_{\delta_\ell}}, \ldots, \boldsymbol{\alpha}_{\mu_K,K_{\delta_\ell}})$. This is done for each quadrature node of the macroelement and we obtain

$$A_K = \underbrace{\sum_{\ell=1}^{\mathcal{L}} \frac{\omega_{K_\ell}}{|K_{\delta_\ell}|} (\mathcal{A}^T_{K_\ell} A_{mic,K_\ell} \mathcal{A}_{K_\ell})}_{\texttt{hmm\_stima\_type\_of\_K}}$$

as defined in (16). The corresponding MATLAB code is given here for quadrilateral macroelement (hence the factor 0.25), i.e., `type_of_K=quad`,

```
for node_no=1:4
    % select current quadrature nodes

  ...

    % Assemble matrix AConstr for constrained system and corresponding rhs
    AConstr=[A ConstraintMat';...
        ConstraintMat sparse(size(ConstraintMat,1),size(ConstraintMat,1))];
    RhsConstr=[zeros(NoOfNodes,4);ConstraintRhs];

    % solve constrained microproblem
    x=AConstr\RhsConstr;
    alpha=x(1:size(MicroCoordinates,1),:);

  ...

    %The contribution to the macro stiffness matrix
    A_macro=A_macro+ .25 * K_macro/K_micro *alpha'*A*alpha;
end
```

where `A_macro` and `A` in the above code are denoted A_K and A_{mic}, respectively, in the above discussion.

The main steps of `hmm_stima_type_of_K` are

- to assemble the micro stiffness matrix (see Section 3.2.1),
- to incorporate the micro coupling constraints (P) or (D) (see Section 3.2.2).

These steps are explained in the two following sections. We will mainly concentrate on a given sampling domain and skip the index corresponding to the particular macro integration point. We will thus write $K_\delta, \varphi_i^h, \varphi_{lin,i}^H, \psi_m^h, x_k$ instead of $K_{\delta_l}, \varphi^h_{i,K_{\delta_\ell}}, \varphi^H_{lin,i,K_{\delta_\ell}}, \psi^h_{m,K_{\delta_\ell}}, x_{K_{\delta_\ell}}$. When no confusion can occur, we will also skip the underscript index for the microfunctions $\varphi_i^h, i = 1,\dots,\mu_K$ and simply write φ^h for a given microfunction and φ^H for its corresponding macrofunction.

3.2.1. Assembly of the micro stiffness matrix

Gauss quadrature rules are used to evaluate the integral (23) numerically. This is done in a standard way in `hmm_micro_stima_quad`

$$(A_{\text{mic}})_{\ell m} = \sum_{T\in\mathcal{T}_h}\int_T a\left(x_k,\frac{x}{\varepsilon}\right)\nabla\psi_\ell^h\cdot\nabla\psi_m^h\,dx.$$

For each microelement T the contribution to the micro stiffness matrix is evaluated via

$$\sum_{i=1}^{4}\frac{1}{4}\int_{\widehat{T}} a\left(x_k,\frac{F_T(\xi)}{\varepsilon}\right)(D_{F,T}^{-T}\,\nabla\hat{\psi}_\ell^h(\xi))\cdot(D_{F,T}^{-T}\,\nabla\hat{\psi}_m^h(\xi))|\det D_{F,T}|d(\xi),$$

where $F_T(\xi) = x_0 + D_{F,T}\xi$ is the affine mapping (here we only allow parallelograms for simplicity) from the reference element $\widehat{T} = [0,1]^2$ onto $T\in\mathcal{T}_h$ with $D_{F,T}$ given by `Map`, ξ_i are determined by the two points Gauss quadrature rule on the reference element defined in `quadnodes`, and $\hat{\psi}_l, \hat{\psi}_m$ are the usual bilinear shape functions on the reference square $\widehat{T}$ with derivatives computed in `phi_prime_hat`. We note that the slow variable x_k in $a(x_k,\frac{x}{\varepsilon})$ is kept fixed in the microproblem. The tensor $a(x_k,\frac{x_i}{\varepsilon})$ is evaluated in the function `tensor_a.m`. The MATLAB code for the micro stiffness matrix assembly is given by

3.2.2. Micro boundary conditions and coupling constraints

The suitable constraints between macro- and microsolvers (or boundary conditions for the microproblem) are encoded in the matrix D (see (21)). In our MATLAB implementation, we provide both Dirichlet or periodic boundary conditions for the microproblem. Other boundary conditions could easily be incorporated using the structure explained below. They can be selected by setting the variable `bctype` in the main file `hmmfem2d.m` to either `dirichlet` or `periodic`. The chosen boundary conditions are then selected in `hmm_stima_type_of_K`.

Periodic micro constraints. If $(\varphi^h - \varphi^H_{lin}) \in S_P(K_\delta,\mathcal{T}_h)$ (see (12)) we must have

(i) $\int_{K_\delta}(\varphi^h - \varphi^H_{lin})\,dx = 0,$
(ii) $(\varphi^h - \varphi^H_{lin})(p) = (\varphi^h - \varphi^H_{lin})(p')$, for all L couples (p,p') of boundary nodes on opposite edges avoiding redundant couples at the corner.

In this situation, the matrix D has the form

$$D = \begin{pmatrix} b_1 & \dots & b_{M_{\text{mic}}} \\ & \widetilde{D} & \end{pmatrix}, \tag{24}$$

where the first row corresponds to the condition (i) and the $L\times M_{\text{mic}}$ matrix $\widetilde{D}$ corresponds to the condition (ii). The first condition simply fixes the constant of the periodic boundary value problem and can be chosen arbitrarily. A convenient choice is $\int_{K_\delta}\varphi^h\,dx = 0$, i.e.,

$$\sum_{m=1}^{M_{\text{mic}}}\alpha_m\int_{K_\delta}\psi_m^h\,dx = \sum_{T\in\mathcal{T}_h}\sum_{m=1}^{M_{\text{mic}}}\alpha_m\underbrace{\int_T\psi_m^h\,dx}_{b_m} = \sum_{T\in\mathcal{T}_h} b^T\alpha = 0. \tag{25}$$

```
function M = hmm_microstima_quad(vertices,...
    MacroQuadNode, epsilon)

...

% map from reference element
Map=[vertices(2,:)-vertices(1,:);vertices(4,:)-vertices(1,:)]';

% gauss quadrature at the 4 quadrature nodes, reference element and mapped
% element
quadnodes_ref=[.5-sqrt(3)/6, .5+sqrt(3)/6, .5-sqrt(3)/6, .5+sqrt(3)/6;...
               .5-sqrt(3)/6, .5-sqrt(3)/6, .5+sqrt(3)/6, .5+sqrt(3)/6];
quadnodes= Map*quadnodes_ref+repmat(vertices(1,:)',1,4);

M=zeros(4,4);
for node=1:4
    quadnode=quadnodes(:,node);
    x=quadnodes_ref(1,node);
    y=quadnodes_ref(2,node);

    phi_prime_hat=[y-1, x-1; 1-y, -x; y, x; -y, 1-x];
    phi_prime_invD=(phi_prime_hat/Map);

    % evaluate coefficiency tensor at the specific nodes
    a=tensor_a(MacroQuadNode, quadnode/epsilon);
    M=M+(det(Map))*phi_prime_invD*a*phi_prime_invD';
end
M=M/4;

end
```

This is implemented in the MATLAB function `micro_constraints_periodic.m`:

```
NoOfNodes = size(MicroCoordinates,1);
b=sparse(NoOfNodes,1);

for j=1:size(MicroElements,1)
    b(MicroElements(j,:))=b(MicroElements(j,:))+...
        det([1,1,1; MicroCoordinates(MicroElements(j,1:3),:)'])/4;
end
```

To treat the periodicity conditions we select the L non-redundant couples at opposite sides of K_δ as $(p,p')=(p_{\mu(\ell)},p_{\nu(\ell)})$, $\ell=1,\dots,L$, where $\mu(\ell)$ and $\nu(\ell)$ are the node numbers of the marked couples. Using (22) we can express the periodicity condition for the ℓth constraint as

$$\varphi^h(p_{\mu(\ell)})-\varphi^h(p_{\nu(\ell)})=\sum_{m=1}^{M_{\mathrm{mic}}}\widetilde{D}_{\ell m}\alpha_m=\varphi^H_{\mathrm{lin}}(p_{\mu(\ell)})-\varphi^H_{\mathrm{lin}}(p_{\nu(\ell)}),\tag{26}$$

where $\varphi^h(p_{\mu(j)})=\alpha_j$, and the matrix $\widetilde{D}\in\mathbb{R}^{L\times M_{\mathrm{mic}}}$ has entry $\widetilde{D}_{\ell j}=1$ if $j=\mu(\ell),\widetilde{D}_{\ell j}=-1$ if $j=\nu(\ell)$ and 0 otherwise. For the example sketched in Fig. 3.2 (left picture) we get

$$\widetilde{D}=\begin{pmatrix}1&0&0&0&0&0&-1&0&0\\0&1&0&0&0&0&0&-1&0\\0&0&1&0&0&0&0&0&-1\\1&0&-1&0&0&0&0&0&0\\0&0&0&1&0&-1&0&0&0\end{pmatrix}.$$

Notice that the nodes can be marked once at the beginning of the code. This is done in `make_constraints.m`

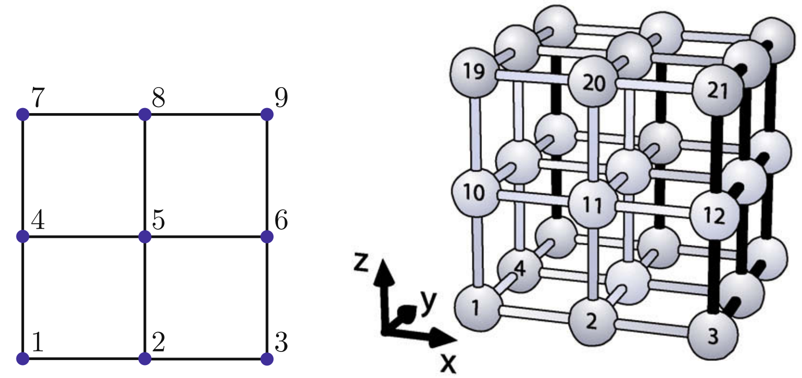
```
Constraints(:,1)=[1:N 1:N:N*N-N];
Constraints(:,2)=[(N-1)*N+1:N*N N:N:N*N-N];
```

and the computation of the matrix $\widetilde{D}$ is implemented in `micro_constraints_periodic.m` as follows:

```
NoOfPerNodes=size(Constraints,1);
ConstraintMat=sparse(repmat(1:NoOfPerNodes,1,2), ...
    [Constraints(:,1) Constraints(:,2)], ...
    [ones(NoOfPerNodes,1); -ones(NoOfPerNodes,1)]);
```

where we took advantage of the MATLAB function `repmat` which creates a large $m\times n$ tiled matrix of which each tile is the replicated original matrix. Thus here, with $m=1$ and $n=2$ we get a vector $(1,2,\dots,\mathtt{NoOfPerNodes},1,2,\dots,\mathtt{NoOfPerNodes})^T$.

We calculate the right-hand side of (26) for all macro basis functions φ^H_j at the same time via $\varphi^H_j(x)=\hat{\varphi}^H_j(F_K^{-1}(x))$ where $F_T(\xi)=x_0+D_{F,T}\,\xi$ is the affine mapping from the reference triangle $\widehat{K}$ onto $K\in\mathscr{T}_h$, where $D_{F,T}$ is given by `MacroMap` and $\hat{\varphi}^H_j$ is the macro reference-element shape function. The corresponding MATLAB code is



(a) Example of micro mesh. Marked couples of nodes are (1,7), (2,8), (3,9), (1,3), (4,6). Periodicity constraints for the couple (7,9) is redundant as it is implicitly enforced by (1,7), (1,3), (3,9).

(b) Example of a three dimensional micro mesh, black lines correspond to boundary node identifications that are redundant. For example periodicity for the couple (3,21) is implicitly enforced by (1,3), (1,19) and (19,21).

Fig. 3.2. Microdomain and periodic boundary conditions in 2d and 3d.

```
% coordinates of periodic boundary conditions
% x_node1 and x_node2 are corresponding periodic nodes
x_node1= MicroCoordinates(Constraints(:,1), :);
x_node2= MicroCoordinates(Constraints(:,2), :);

% coordinates in the reference quadrilateral corresponding to the nodes with
% periodic boundary conditions
x_node_ref1 = (MacroMap\(x_node1-repmat(MacroVertices(1,:), NoOfPerNodes,1))')';
x_node_ref2 = (MacroMap\(x_node2-repmat(MacroVertices(1,:), NoOfPerNodes,1))')';

% corresponding rhs
if (MacroNodes==3) % we have macro triangles
    ConstraintRhs=shapefunction_tri(x_node_ref1)-...
        shapefunction_tri(x_node_ref2);
end
if (MacroNodes==4) % we have macro quadrilaterals
    ConstraintRhs=shapefunction_quad_lin(x_node_ref1, refquadnode)-...
        shapefunction_quad_lin(x_node_ref2, refquadnode);
end
```

The macro basis functions $\hat{\varphi}^H_j$ are evaluated in `shapefunction_tri` and `shapefunction_quad_lin` for triangles and quadrilaterals, respectively. Finally, we merge the constraint matrix b from (25) and $\widetilde{D}$ from (26) to assemble the matrix D given in (24) as well as the right-hand side.

```
% merge both to one unified matrix / vector
ConstraintMat=[b';ConstraintMat];
ConstraintRhs=[zeros(1,MacroNodes);ConstraintRhs];
```

Dirichlet micro constraints. Other constraints for the microsolver, such as constraints through Dirichlet boundary conditions, can be implemented similarly as explained before. In our code we can switch from periodic to Dirichlet boundary constraints. For Dirichlet boundary constraints we have $(\varphi^h - \varphi^H_{lin}) \in S_D(K_\delta, \mathcal{T}_h)$ (see (12)), and thus

$$(\varphi^h - \varphi^H_{lin})(p_{\mu(\ell)}) = 0 \tag{27}$$

for all boundary nodes $p_{\mu(1)}, \dots, p_{\mu(L)}$. We get for the ℓth constraint

$$\sum_{m=1}^{M} D_{\ell m}\alpha_m = \varphi^H_{lin}(p_{\mu(\ell)}), \tag{28}$$

where the entries of $D \in \mathbb{R}^{L\times M_{\text{mic}}}$ are given by $D_{\ell m} = 1$ if $m = \mu(\ell)$ and 0 otherwise. The micro boundary nodes $p_{\mu(\ell)}$ are determined in `make_constraints.m` at the beginning of the code

```
        Constraints=[1:N N*N-N+1:N*N 1+N:N:N*N-2*N+1 2*N:N:N*N-N];
```

and the computation of the matrix D and the right-hand side of (28) is performed similarly as in the periodic case in `micro_constraints_dirichlet.m`.

Remark 4. While we chose to present and discuss periodic and Dirichlet micro constraints, other constraints through Neumann or Robin boundary conditions could easily be implemented, following the procedure described for periodic or Dirichlet constraints.

3.3. Parabolic problems

As mentioned in the introduction, our code can handle parabolic problems without difficulties. Consider for example

$$\frac{\partial u^\varepsilon}{\partial t} = \nabla\cdot(a^\varepsilon \nabla u^\varepsilon) + f \quad \text{in } \Omega\times[t_0, T] \tag{29}$$

with initial conditions given by $u(x,t_0) = u_0(x)$ and mixed Dirichlet and Neumann boundary conditions as for the elliptic problem (1). Using the simple backward Euler method in time leads to the following weak form for the HMM

$$\int_\Omega u^H_n v^H\,dx + \Delta t B(u^H_n, v^H) = \int_\Omega u^H_{n-1} v^H\,dx + \Delta t\left(\int_\Omega f v^H\,dx + \int_{\partial\Omega_N} g_N v^H\,dx - B(g_D, v^H)\right) \tag{30}$$

$\forall v^H \in V^p_D(\Omega, \mathcal{T}_H)$ and $t \in (t_0, T)$, where u^H_n is an approximation of $u(x,t_n), t_n = t_0 + n\Delta t$ and $B(\cdot,\cdot)$ is defined in (9). This leads to the linear system

$$(M + \Delta t A)u^H_n = b + Mu^H_{n-1}, \tag{31}$$

where A and b are as in (18) and obtained as described in Section 2 and the mass matrix M has entries given by $M_{ij} = \int_\Omega \varphi^H_i \varphi^H_j\,dx$. We described here the simplest time integration procedure. Of course more sophisticated solvers could be implemented. Notice that by using discontinuous Galerkin methods for the HMM [7], the mass matrix M becomes block diagonal and an explicit stabilized scheme as proposed in [1] could be applied for the time integration, avoiding the use of linear solver for the whole approximation procedure of parabolic multiscale problems. This may be an appealing strategy when the spatially discretized problem is of high dimension and contains complicated nonlinear terms. In this case, the implementation of iterative solvers requires some care (suitable preconditioners, etc.) while the above strategy is very easy to implement while being efficient.

3.4. Three dimensional problems

With only a few modifications we can extend the 2d code presented above to three dimensional problems. For the macroproblem, it is convenient to use tetrahedral elements. In three dimension, the sampling domains are chosen to be cubes of size δ and parallelepipeds meshes are suitable for the microproblems. At the macrolevel, the core structure using 3d shape functions is built upon standard procedure [9]. Macro assembly using microsolvers on sampling domains follows the procedure explained in Section 2 with obvious changes. Let us notice that some care is needed to implement the micro boundary conditions. For example, for periodic boundary conditions, a moments reflection is needed to find all couples of boundary nodes on opposite sides while avoiding redundant couples. Fig. 3.2b gives an example of such a node selection.

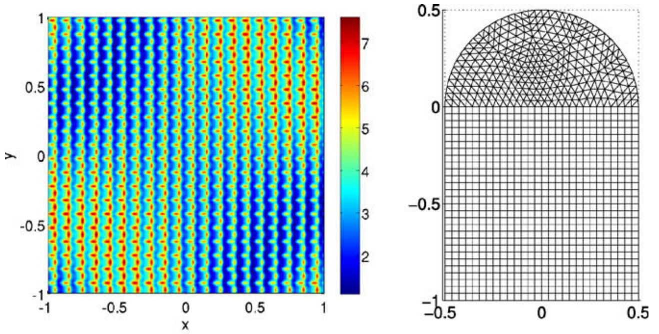
4. Numerical examples

In what follows, we present several numerical examples to illustrate the use, the versatility, the capabilities and the performance of our code.

In a first set of examples, we present an elliptic problem with non-uniformly periodic coefficients (Section 4.1), an elliptic problem with a random tensor (Section 4.2) and a time dependent multiscale (parabolic) problem (Section 4.3). We notice that problems with random tensors, similar as chosen here, are widely used in the modeling of porous media flows. In a second set of examples, we investigate three dimensional simulations based on simplified real-world engineering problems: the steady state heat distribution in a heatsink (Section 4.4) and in a microprocessor (Section 4.5). Both the processor and the heatsink are assumed to be made of composite materials.

4.1. Problems with non-uniformly periodic coefficients

For the first example, we perform numerous numerical experiments to illustrate the convergence behavior of the FE-HMM in various norms, to show the impact of the size of the sampling domains on the numerical solution and to indicate how macro- and micromeshes have to be refined in order to minimize the computational complexity with optimal convergence rates. For the other examples, we do not perform a rigorous numerical convergence study. Such studies have been reported elsewhere for the FE-HMM [2,3,5,6,8,28]. We nevertheless occasionally complement our plots by giving energy norms of the exact, homogenized or FE-HMM solutions to illustrate some (convergence) aspects of the proposed algorithm.



(a) Conductivity tensor for $\varepsilon = 0.1$ (b) Grid

Fig. 4.1. Snapshot of the conductivity tensor (left picture) and sketch of the computational mesh (right picture) for the elliptic problem 4.1.

We consider the two scale problem taken from [6]

$$-\nabla\cdot\left(a\left(x,\frac{x}{\varepsilon}\right)\nabla u^\varepsilon\right) = f(x) \quad \text{in } \Omega, \tag{32}$$

$$u^\varepsilon = 0 \quad \text{on } \partial\Omega_D, \tag{33}$$

$$-n\cdot\left(a\left(x,\frac{x}{\varepsilon}\right)\nabla u^\varepsilon\right) = 0 \quad \text{on } \partial\Omega_N, \tag{34}$$

where the conductivity tensor a^ε is given by

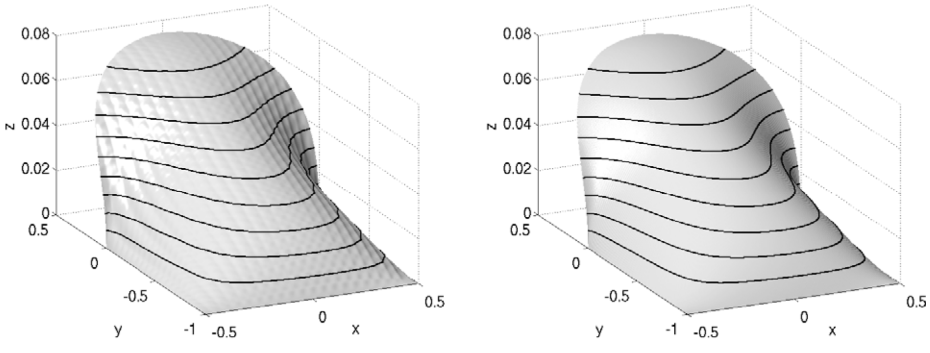
$$a\left(x,\frac{x}{\varepsilon}\right) = \frac{1.5\sin(2\pi x_1/\varepsilon)}{1.5\sin(2\pi x_2/\varepsilon)} + \frac{1.5\sin(2\pi x_2/\varepsilon)}{1.5\cos(2\pi x_1/\varepsilon)} + \sin(4x_1x_2) + 1, \tag{35}$$

with $x = (x_1, x_2)$. We choose $f = 1$. A snapshot of the conductivity tensor is shown in Fig. 4.1a.

The domain Ω consists of a semi circle and a rectangle, meshed with 1137 nodes using 576 triangles and 784 quadrilaterals, respectively (see Fig. 4.1b). This problem does not have an explicit analytical solution and we thus compute a finescale solution using a standard FEM with a fine mesh with about 10^6 degrees of freedom (DOF) in order to resolve the microscale. Neumann boundary conditions (34) are given on the boundary of the semi circle Ω_N, whereas the remaining part has Dirichlet boundary conditions (33). While the problem itself is not periodic, the fast scale is, and we thus choose $\delta = \varepsilon$ for the size of the sampling domains and periodic micro boundary conditions. Fig. 4.3 shows the solution generated by the FE-HMM.

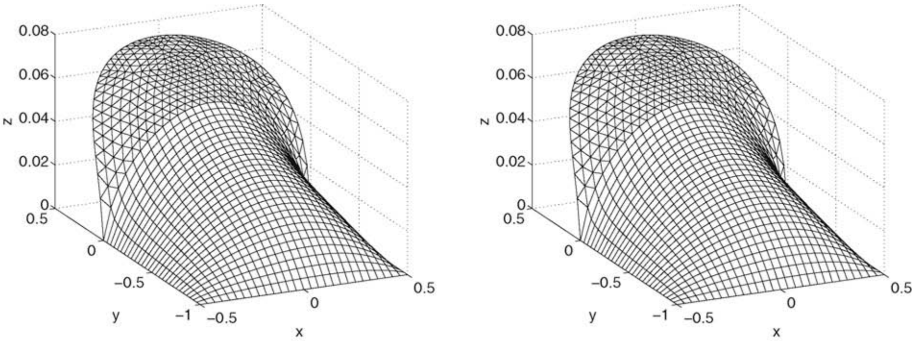
In Figs. 4.2 and 4.4 we compare the FE-HMM solution with the fine scale solutions obtained by choosing $\varepsilon = 0.05$ and $\varepsilon = 0.005$. Fine scale solutions with smaller ε quickly get impossible to compute due to the increasing complexity. The plots of Fig. 4.2 show the behavior of the fine scale solution as ε decreases. For $\varepsilon = 0.005$ the oscillations are barely visible (remember that in the limit $\varepsilon \to 0$ one gets the homogenized solution u^0 of (6) as discussed in Section 2.1).

The FE-HMM captures the effective solution and is thus independent of ε. This fact is consistent with the numerical results provided in Table 4.1, where we compare the energy norm $\|u\|_A := \sqrt{\int_\Omega a\cdot\nabla u\nabla u\,dx}$ and maximum norm $\|u\|_\infty := \sup_{x\in\Omega}|u(x)|$ for different values of ε and N_{mic}. Here, N_{mic} refers to the number of discretization points in each direction of the d-dimensional sampling domains of the microsolver (10) of the FE-HMM, i.e. we have a total of N_{mic}^d DOF in each sampling domain. In Fig. 4.5, we show the influence of the micro discretization in the macrosolution of



(a) Fine scale results for $\varepsilon = 5\cdot10^{-2}$ (b) Fine scale results for $\varepsilon = 5\cdot10^{-3}$

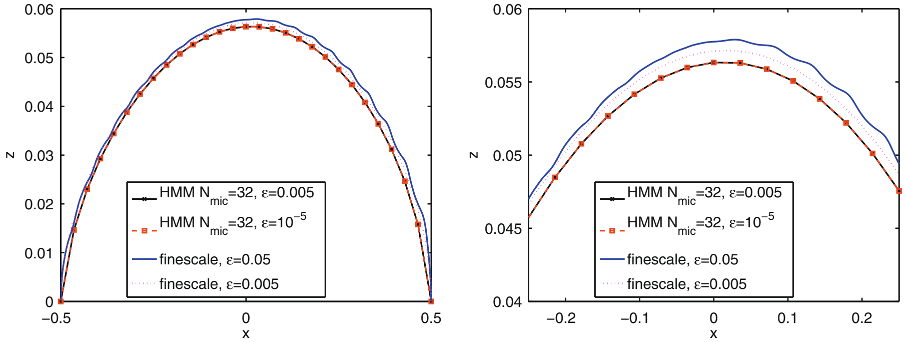
Fig. 4.2. Fine scale solutions of the problem described in Section 4.1 computed on a mesh with 10^6 degrees of freedom for two different ε.



(a) $\varepsilon = 5 \cdot 10^{-3}, N_{Mic} = 32$

(b) $\varepsilon = 10^{-5}, N_{Mic} = 32$

Fig. 4.3. FE-HMM solutions of the elliptic problem described in Section 4.1 for two different ε. The FE-HMM captures the effective solution and is thus independent of ε.



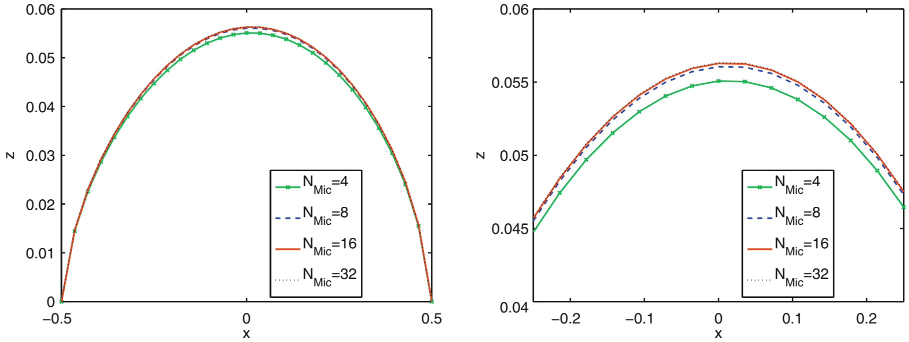
(a) full cross-section

(b) zoom into cross-section

Fig. 4.4. FE-HMM solution (with 1100 macro DOF) and fine scale solutions (with 10^6 DOF) of problem 4.1 for various ε (cross-section of the solution of the problem described in Section 4.1 through $y = 0$).

Table 4.1
Energy and maximum norm for the solution of the problem described in Section 4.1.

		$N_{\text{mic}} = 4$	$N_{\text{mic}} = 8$	$N_{\text{mic}} = 16$	$N_{\text{mic}} = 32$	Finescale
$\varepsilon = 0.005$	$\|u\|_A$	0.2105	0.2124	0.2128	0.2129	0.2146
	$\|u\|_\infty$	0.0713249	0.072289	0.0725326	0.0725888	0.0736806
$\varepsilon = 10^{-5}$	$\|u\|_A$	0.2101	0.2124	0.2128	0.2129	–
	$\|u\|_\infty$	0.0710427	0.0722795	0.0725326	0.0725888	–



(a) full cross-section

(b) zoom into cross section

Fig. 4.5. FE-HMM solutions of the problem described in Section 4.1 for $\varepsilon = 0.005$ and $N_{\text{mic}} = 4, 8, 16, 32$ (cross-section of the solution of the problem described in Section 4.1 through $y = 0$).

the FE-HMM by varying N_{mic}. The results presented in Section 2.2.2 allow to choose the appropriate micro mesh size according to the selected macro discretization. This is further discussed in Section 4.1.1.

Sampling domain size. We next report numerical results in the situation when δ, the size of the sampling domain, is not an integer multiple of ε, i.e. when δ does not comprise an integer number of ε micro oscillations. This situation arises even for periodic problems, if one does not know the exact size of the period. In Table 4.2, we study the influence of such “modeling error” for the HMM. We choose $\delta = \frac{5}{3}\varepsilon$, and compare the HMM solution obtained with either periodic or Dirichlet boundary conditions for the microproblem. For each type of boundary conditions we also refine the discretization of the microsolver by increasing N_{mic}.

As can be seen, we get better solutions with periodic boundary conditions. The better behavior of periodic boundary conditions even for such non-matching sampling domains is well-known but yet not fully understood [34]. Cross-section plots for a fixed N_{mic} are presented in Fig. 4.6. Further numerical results for the FE-HMM can be found in [3,5,28].

Remark 5. We emphasize that while it is convenient to compare the HMM solution u^H with a resolved fine scale solution, the former does usually not converges to the latter in the H^1 or in the energy norm as $\varepsilon \to 0$ (remember that u^H converges to u^0 as macro- and micromeshes tend to zero). In the L^2 norm, the following convergence rate can be obtained $\|u^\varepsilon - u^0\|_{L^2(\Omega)} \leqslant C\varepsilon$, and by the triangle inequality and by using results of Section 2.2.2, convergence rate for u^H (to the fine scale solution in the L^2 norm) can also be obtained (see [3] for details).

4.1.1. Performance of the method

Finally, we want to investigate the dependence of L^2 and H^1 errors on the choice of the macro- and micromeshes and give an indication of CPU-times for the FE-HMM.

In order to easily compare L^2 and H^1 errors for different mesh sizes, we consider the following quasi-1d-problem on a square domain for which the exact solution and the homogenized tensor can be computed analytically (see [3])

$$-\nabla \cdot \left(a\left(\frac{x}{\varepsilon}\right)\nabla u^\varepsilon\right) = f(x) \quad \text{in } \Omega = (0,1)^2, \tag{36}$$

$$u^\varepsilon|_{\Gamma_D} = 0 \quad \text{on } \Gamma_D := \{x_1 = 0\} \cup \{x_1 = 1\}, \tag{37}$$

$$n \cdot \left(a\left(\frac{x}{\varepsilon}\right)\nabla u^\varepsilon\right)\Big|\Gamma_D = 0 \quad \text{on } \Gamma_N := \partial\Omega \setminus \Gamma_N. \tag{38}$$

We choose $f(x) \equiv 1$ and set $a(\frac{x}{\varepsilon}) = \cos(2\pi\frac{x_1}{\varepsilon}) + 2$. For the FE-HMM, we choose quadrilaterals for the macro- and micro-discretization.

Optimal refinement strategy for the macro- and micromeshes. In what follows, N_{mac} refers to the number of discretization points in each direction of the domain $\Omega = (0,1)^2$, while we recall that N_{mic} refers to the number of discretization points in each direction of the d-dimensional sampling domains (here $d = 2$). Thus, the macromesh H is given by $H = 1/N_{\text{mac}}$ and the micromesh by $h = \delta/N_{\text{mic}}$ (in the following computations we choose $\delta = \varepsilon$). As the sampling domain is of size δ, we will also consider the scaled micromesh given by $h_s = h/\delta = 1/N_{\text{mic}}$. The macro and micro DOF are then given by $\mathcal{O}(M_{\text{mac}})$ and $\mathcal{O}(M_{\text{mic}})$, respectively, where $M_{\text{mac}} = N^d_{\text{mac}}$ and $M_{\text{mic}} = N^d_{\text{mic}}$ Following the a-priori estimates stated in Section 2.2.2 we see that

$$M_{\text{mic}} \propto M_{\text{mac}} \quad (L^2\text{norm}), \quad M_{\text{mic}} \propto \sqrt{M_{\text{mac}}} \quad (H^1\text{norm}),$$

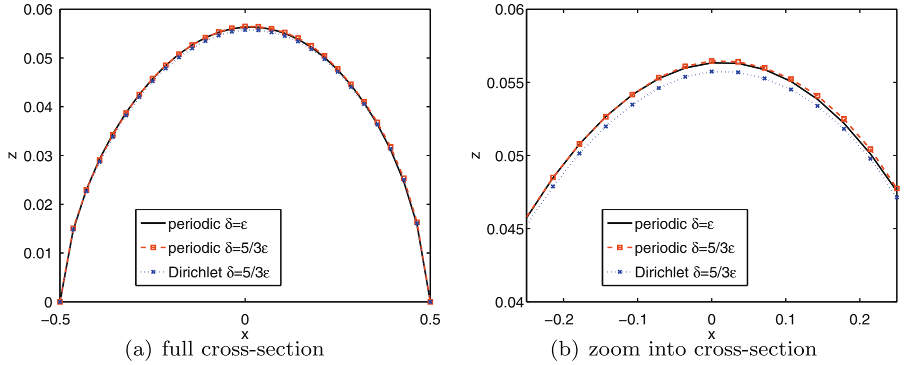
i.e., $h_s \propto H$ or $N_{\text{mic}} \propto N_{\text{mac}}$ (for the L^2 norm) and $h_s \propto \sqrt{H}$ or $N_{\text{mic}} \propto \sqrt{N_{\text{mac}}}$ (for the H^1 norm) are the best refinement strategies for optimal convergence rates with minimal computational cost. We thus obtain a complexity of $\mathcal{O}(M_{\text{mac}} \cdot M_{\text{mic}}) = \mathcal{O}(M^{3/2}_{\text{mac}})$ floating point operations for a linear (macro) convergence rate in the H^1 norm and $\mathcal{O}(M_{\text{mac}} \cdot M_{\text{mic}}) = \mathcal{O}(M^2_{\text{mac}})$ floating point operations for a quadratic convergence rate in the L^2 norm. Here we assume that the cost (floating point operations) of the method is proportional to the total DOF (which is the case for example when using multigrid linear solver). For the above problem, we select a sequence of macromeshes $H = 1/16, 1/32, 1/64, 1/128$ and adapt the micromesh h_s according to the above criteria. This gives (selecting $h_s = H$ for the L^2 error and $h_s = \sqrt{H}$ for the H^1 error) the following macro- and micromeshes written here in term of N_{mic} and $N_{\text{mac}}(H = 1/N_{\text{mac}}, h_s = 1/N_{\text{mic}})$:

N_{mac}	16	32	64	128
N_{mic} (L^2-micro refinement strategy)	16	32	64	128
N_{mic} (H^1-micro refinement strategy)	4	6	8	11

The results presented in Figs. 4.7 and 4.8 show the errors and CPU-times corresponding to the aforementioned refinement-strategies (CPU-times measured on an AMD Opteron 2214 HE).

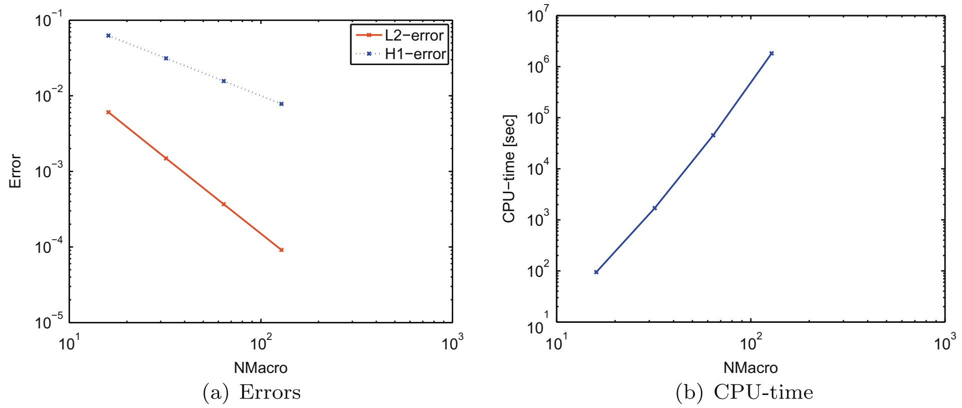
Table 4.2
Energy and maximum norm for the FE-HMM and fine-scale solution of the elliptic problem described in Section 4.1 with a non-integer number of periods within each sampling domain ($\delta = \frac{5}{3}\varepsilon, \varepsilon = 5 \cdot 10^{-3}$).

		$N_{\text{mic}} = 4$	$N_{\text{mic}} = 8$	$N_{\text{mic}} = 16$	$N_{\text{mic}} = 32$	Finescale
Periodic	$\|u\|_A$	0.2085	0.2119	0.2130	0.2133	0.2146
	$\|u\|_\infty$	0.0702	0.0720	0.0726	0.0728	0.0737
Dirichlet	$\|u\|_A$	0.2080	0.2107	0.2117	0.2119	0.2146
	$\|u\|_\infty$	0.0699	0.0714	0.0719	0.0720	0.0737



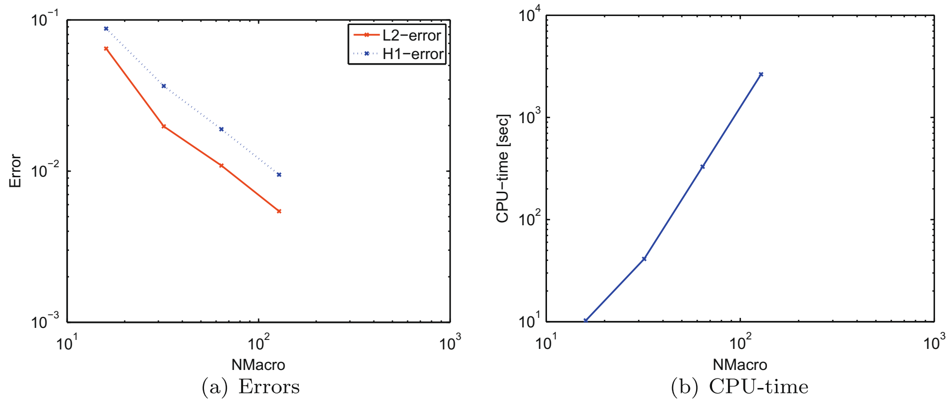
(a) full cross-section (b) zoom into cross-section

Fig. 4.6. FE-HMM solutions of the problem described in Section 4.1 with a non-integer number of periods within the sampling domain with $\delta = \frac{5}{3}\varepsilon$, $\varepsilon = 0.05$, $N_{\text{mic}} = 32$ (cross-section through $y = 0$).



(a) Errors (b) CPU-time

Fig. 4.7. Dependence of errors and CPU-time on the macro DOF for the test-case problem described in Section 4.1.1 (L^2-micro refinement strategy).



(a) Errors (b) CPU-time

Fig. 4.8. Dependence of errors and CPU-time on the macro DOF for the test-case problem described in Section 4.1.1 (H^1-micro refinement strategy).

Remark 6. Because of the constant homogenized tensor we could solve the microproblem just once. However, to assess realistic computing times for two-scale problems, we provide the CPU-times that would be needed to solve the full problem.

As predicted by the a-priori-estimates, the chosen refinement strategies achieve quadratic and linear convergence rates for the error in the L^2 and H^1 norms, respectively. We list in Table 4.3 the detailed errors.

We see in Table 4.3 that an appropriate refinement strategy is crucial to achieve the optimal quadratic convergence rate in the L^2 norm. We also see (in accordance with the theoretical estimates) that we can use a much larger micromesh for H^1 approximations than for L^2 approximations. We emphasize that a correct refinement strategy is crucial for efficient computations. Finally, we observe as expected that the convergence rates are independent of ε.

Remark 7. A comparison with a fine-scale solution is not useful in the present context of performance-evaluation. Indeed, as the performance of the FE-HMM is independent of the fine scale ε, this parameter can be tuned arbitrarily for the fine scale problem, leading to arbitrary (high) CPU-time for the corresponding fine scale solution.

Parallel implementation of the FE-HMM. We remark that an implementation on a parallel computer is easily possible and leads – due to the independence of the different microproblems – to a near-optimal speedup. A parallel version of the FE-HMM (on eight nodes) has been used for the 3d-problems in Sections 4.4 and 4.5.

4.2. Problems with random tensor

We again consider problem (32), but this time with a random tensor $a^\varepsilon(x)$. The domain Ω is the same as in the previous example of Section 4.1 except that zero Dirichlet boundary conditions are chosen on the entire boundary. Again, triangles are used in the circular part of the domain while quadrilaterals are used in the remaining computational domain. We notice that random models for the fine scale are often used for problems involving the pressure equation in porous media flow [33].

This example shows how to address problems where the tensor is not given by an explicit function, but only as a random field.

Table 4.3
Comparison of the L^2 and H^1 errors for the test-case described in Section 4.1.1 using different refinement strategies.

H		1/16	1/32	1/64	1/128
L^2-approximation	$\|u^0 - u^H\|_{L^2(\Omega)}$	6.06e−03	1.48e−03	3.67e−04	9.13e−05
	$\|u^0 - u^H\|_{H^1(\Omega)}$	6.25e−02	3.13e−02	1.56e−02	7.80e−03
H^1-approximation	$\|u^0 - u^H\|_{L^2(\Omega)}$	6.46e−02	1.98e−02	1.09e−02	5.43e−03
	$\|u^0 - u^H\|_{H^1(\Omega)}$	8.76e−02	3.65e−02	1.89e−02	9.48e−03

While the stochastic field for the considered problem is computer-generated, our code could be used with a real-life tensor obtained from imaging techniques (for example through scanning or microscopy).

The realization of the log-normal stochastic field with mean zero and variance $\sigma = 1$ is generated by the moving ellipse average method Section 4.1 [33]. We set the correlation lengths of the stochastic field to be $\varepsilon_x = 0.01$ and $\varepsilon_y = 0.02$. A snapshot of this tensor is shown in Fig. 4.9a.

We compute a reference solution on a fine grid of about 10^6 degrees of freedom (DOF), see Fig. 4.9b, and compare the solution with the FE-HMM on the macro coarse grid with 1100 DOF. For the FE-HMM we present results for various sizes of the sampling domain. We first choose $\delta = 0.02$ and then $\delta = 0.06$. We clearly see in Fig. 4.9c and d that the profile of the solution is closer to the reference solution as the sampling domain contains more correlation length of the random field. This observation can also be seen in Table 4.4 when comparing the energy norm of the various solutions obtained with the FE-HMM to the energy norm of the reference solution.

4.3. Parabolic problems

As mentioned in Section 3.4, our code can handle parabolic problems without difficulties. As an example, we consider

$$\frac{\partial u}{\partial t} = \nabla \cdot (a^{\varepsilon}(x) u^{\varepsilon}) + f \quad \text{in } \Omega \times [0, T]. \tag{39}$$

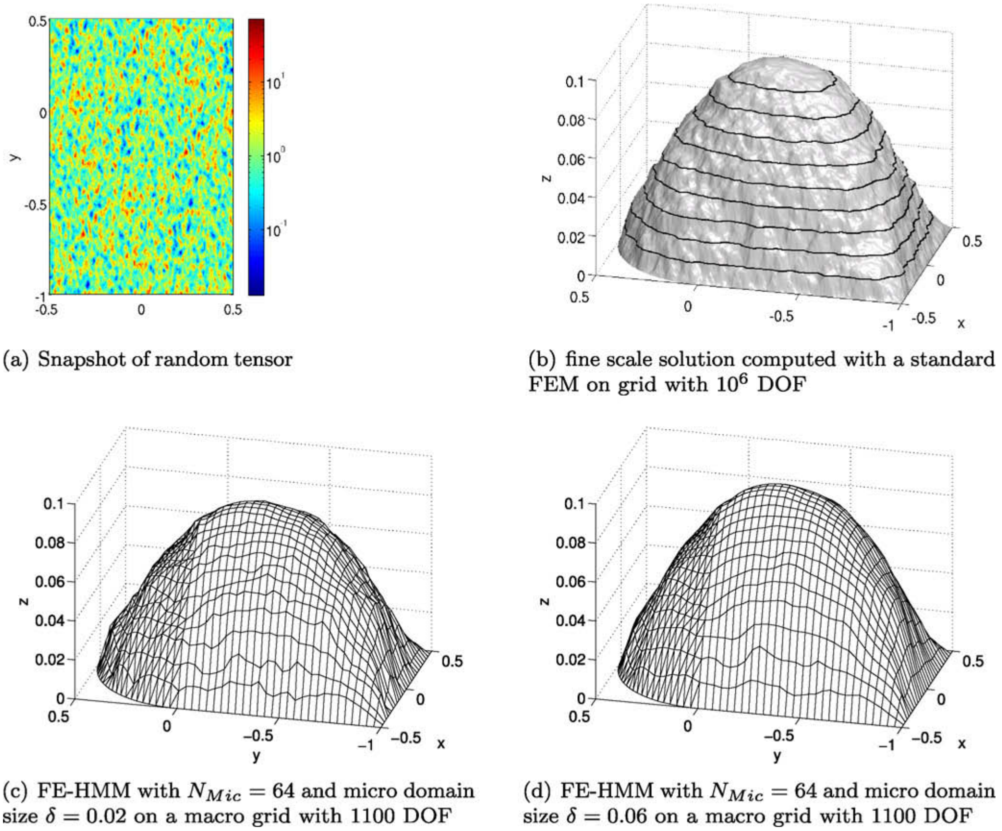
For simplicity, we consider an implicit Euler scheme for time integration. We choose $T = 1$ and a time step of $\Delta t = 0.1$.

We keep the same domain and boundary conditions as in problem Section 4.1, set $f = 0$ and consider the initial condition $u_0 = -10(x - 0.5) \cdot (x + 0.5) \cdot (y + 1)$. The tensor used is a scaled version of the tensor (35), given by

$$a\left(x, \frac{x}{\varepsilon}\right) = 0.1 \cdot \left(\frac{1.5\sin(2\pi x_1/\varepsilon)}{1.5\sin(2\pi x_2/\varepsilon)} + \frac{1.5\sin(2\pi x_2/\varepsilon)}{1.5\cos(2\pi x_1/\varepsilon)} + \sin(4x_1x_2) + 1\right).$$

Since we consider a time-independent tensor a^{ε} here, we only have to solve the microproblems once and can use the same stiffness matrix whenever we solve the linear equation arising from the implicit Euler scheme. The code corresponding to this problem is given in `fe_hmm2d_para.m`. More general problems with tensors of the form $a(x, \frac{x}{\varepsilon}, t)$ can be computed in a similar manner, but in this case the stiffness matrix has to be updated at each time step.

Results for this parabolic problem solved with the FE-HMM code are sketched in Fig. 4.10. The results at $t = 1$ are extremely close to each other and are indistinguishable to the eye. A comparison of the energy and infinity norms of various numerical



(a) Snapshot of random tensor

(b) fine scale solution computed with a standard FEM on grid with 10^6 DOF

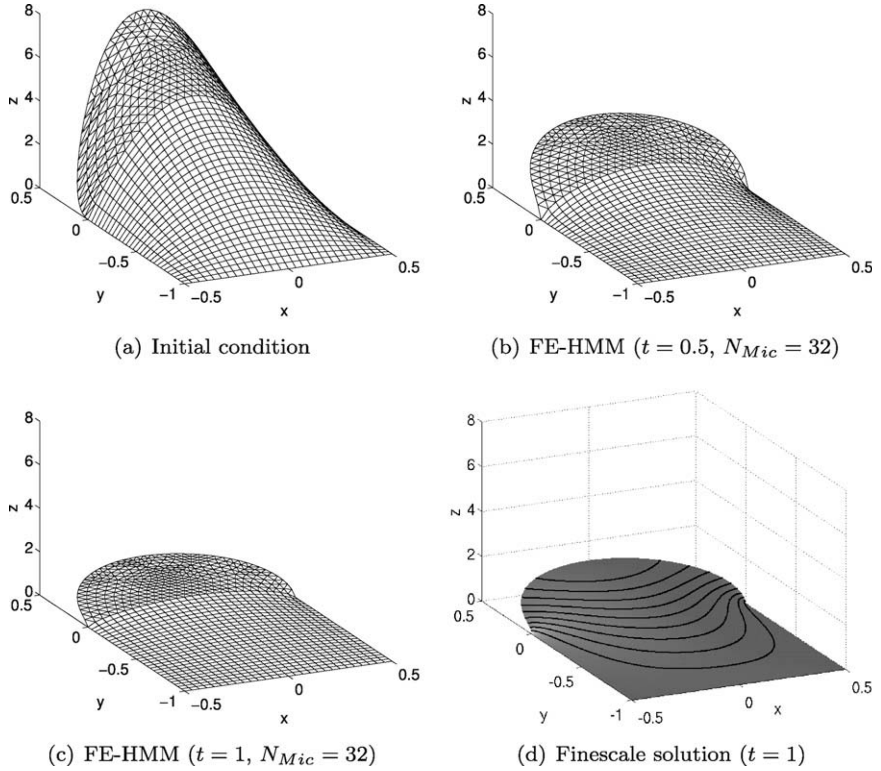
(c) FE-HMM with $N_{Mic} = 64$ and micro domain size $\delta = 0.02$ on a macro grid with 1100 DOF

(d) FE-HMM with $N_{Mic} = 64$ and micro domain size $\delta = 0.06$ on a macro grid with 1100 DOF

Fig. 4.9. FE-HMM and fine scale solution of the random problem described in Section 4.2. Snapshot of the tensor and comparison of the fine-scale solution with FE-HMM solutions with differently sized micro-domains.

Table 4.4
Energy norms of the FE-HMM and fine-scale solutions of the problem described in Section 4.2 with a random tensor. FE-HMM results are given for various micro sampling-domain sizes $\delta \times \delta$ and DOF N_{mic}^2.

	$N_{\text{mic}} = 4$	$N_{\text{mic}} = 8$	$N_{\text{mic}} = 16$	$N_{\text{mic}} = 32$	$N_{\text{mic}} = 64$	Finescale
$\delta = 0.02$	0.2352	0.2415	0.2439	0.2449	0.2454	0.2583
$\delta = 0.06$	0.2313	0.2454	0.2520	0.2551	0.2567	0.2583



(a) Initial condition

(b) FE-HMM ($t = 0.5$, $N_{Mic} = 32$)

(c) FE-HMM ($t = 1$, $N_{Mic} = 32$)

(d) Finescale solution ($t = 1$)

Fig. 4.10. FE-HMM (1100 macro DOF) and fine scale (10^6 DOF) solution of the parabolic problem described in Section 4.3, obtained with the described MATLAB code ($\varepsilon = 0.005$).

solutions is given in Table 4.5. Again, we clearly see that the FE-HMM is independent of ε.

4.4. Three dimensional problem: steady state heat distribution in a heatsink

We now present a 3d multiscale problem, the steady state heat conduction of a heatsink made of a layered material. Crucial for the efficiency of today's microprocessors is a sophisticated cooling process which is usually achieved with the help of a heatsink, where advanced composite materials are used in order to obtain the most efficient cooling procedure. These heatsinks are typically mounted on top of a microprocessor to dissipate heat away from it, often with the help of a fan built on top it. However, since we want to avoid discussing modelling issues, we neglect the effects originating from such fans and more refined models.

Equations. Heat is transferred away in three ways: conduction, convection and radiation, the latter of which we ignore. Heat conduction within the heatsink and its different components can be described in terms of Fourier's law

$$-\nabla \cdot (a\nabla u) = f, \tag{40}$$

where a is the thermal conductivity tensor, u is the temperature of the body and f is a heat source. Convective heat transfer with the surrounding air is expressed by the Robin boundary condition

$$n \cdot (a\nabla u) = q_0 + \alpha(u_{\text{amb}} - u),$$

where α is the heat transfer coefficient, u_{amb} is the ambient temperature and q_0 represents the heat flux entering the domain.

Using the above heat transfer models leads to the following problem describing the steady-state temperature distribution of the heatsink

$$-\nabla \cdot (a^\varepsilon \nabla u^\varepsilon) = f \quad \text{in } \Omega, \tag{41}$$

$$n \cdot (a^\varepsilon \nabla u^\varepsilon) + \alpha u^\varepsilon = g_R \quad \text{on } \partial\Omega_R, \tag{42}$$

$$n \cdot (a^\varepsilon \nabla u^\varepsilon) = g_N \quad \text{on } \partial\Omega_N, \tag{43}$$

where Ω is the whole domain and $\partial\Omega_R$ and $\partial\Omega_N$ are the surfaces of the heatsink with Robin and Neumann boundary conditions, respectively. The right-hand side of the Robin boundary conditions is given by $g_R = q_0 + \alpha u_{\text{amb}}$. Notice that we add (as usual) the superscript ε to emphasize on the multiscale (composite) nature of the material.

The Robin boundary conditions (42) involve extra terms $\int_{\partial\Omega_R} g_R v\, ds$ and $-\int_{\partial\Omega_R} \alpha u v\, ds$ to be incorporated into the weak form (3). For the FE-HMM, this results in an extra contribution R to the stiffness matrix A, given by

$$R_{ij} = \int_{\partial\Omega_R} \alpha \varphi_i^H \varphi_j^H\, ds, \tag{44}$$

and an extra contribution to the load vector l given by

Table 4.5
Energy and maximum errors for the FE-HMM and the fine-scale solutions of the parabolic problem described in Section 4.3 at $t = 1$. The computation of the fine-scale solution with $\varepsilon = 10^{-5}$ would involve around 10^{14} DOF and has therefore not been performed.

		$N_{\text{mic}} = 4$	$N_{\text{mic}} = 8$	$N_{\text{mic}} = 16$	$N_{\text{mic}} = 32$	Finescale
$\varepsilon = 0.005$	$\lVert u\rVert_A$	0.8354	0.8491	0.8525	0.8533	0.8697
	$\lVert u\rVert_\infty$	1.0839	1.1060	1.1117	1.1130	1.1379
$\varepsilon = 10^{-5}$	$\lVert u\rVert_A$	0.8313	0.8490	0.8525	0.8533	–
	$\lVert u\rVert_\infty$	1.0773	1.1058	1.1117	1.1130	–

$$r_j = \int_{\partial\Omega_R} g_R \varphi_j^H \, ds. \tag{45}$$

While we consider here two-scale tensors with periodicity in the microscale, i.e. non-uniformly periodic tensors, other tensors for realistic materials obtained via imaging techniques and given as data points could be easily taken into account.

4.4.1. Setup

The side of the heatsink mounted on top of the processor has an incoming heat flux that we model by Neumann boundary conditions

$$n \cdot (a\nabla u) = g_N \quad \text{on } \partial\Omega_N,$$

see Fig. 4.11(b). The incoming heat flux entering $\partial\Omega_N$ is given by $g_N(x) = 3493\, e^{-1000(x^2+y^2)}\ \frac{\mathrm{W}}{\mathrm{m}^2}$, where the numerical values are chosen such that we have a total incoming power of $P = \int_{\partial\Omega_N} g_N\, dx = 10\,\mathrm{W}$.

In our computations, we will assume that the heatsink is made of a layered material. The advantage of choosing such a material is that an analytic formula for the homogenized material is available. This allows to compare the solution of the FE-HMM with an exact solution. Needless to say that for most of materials used in cooling procedures (e.g. [11,35]), analytical formulas are not available and the use of computational procedures as described in this article are unavoidable. We note that recent studies suggest the use of carbon nanotubes to act as a heatsink [11].

Remark 8. We emphasize that for the chosen tensor, there is no advantage in using the FE-HMM since the homogenized tensor is constant and can be analytically computed. Of course for such problems, we need only to call the microsolver once when using the FE-HMM, and no knowledge about the correct averaging procedure is needed (see [8, Section 3.3.2]). The purpose of using such a simple tensor as considered here is that we can make precise comparisons with various averaging procedures. A more general tensor (for which no explicit homogenized tensor is available) is used in Section 4.5, where we study the heat distribution in a microprocessor.

We set the multiscale tensor to be

$$a^\varepsilon(x) = \begin{pmatrix} a_{11}^\varepsilon(x) & 35 & 0 \\ 35 & a_{22}^\varepsilon(x) & 0 \\ 0 & 0 & 200 \end{pmatrix}, \tag{46}$$

where

$$a_{11}^\varepsilon(x) = [500/(5 + 3.5 \cdot \sin(2\pi x_1/\varepsilon))] \cdot e^{10 \cdot x_3},$$
$$a_{22}^\varepsilon(x) = [500/(5 + 3.5 \cdot \cos(2\pi x_1/\varepsilon))] \cdot e^{10 \cdot x_3}.$$

For such tensors it is well-known that explicit analytic formulas are available for the homogenized tensor (see for example [14, Chapter 5.4]). For the tensor (46) we obtain

$$a^0(x) \approx \begin{pmatrix} 100.0 \cdot e^{10 \cdot x_3} & 35 & 0 \\ 35 & 140.0 \cdot e^{10 \cdot x_3} & 0 \\ 0 & 0 & 200 \end{pmatrix} \tag{47}$$

for the homogenized tensor. In our numerical experiments, the heatsink has dimensions $87.5 \times 91.9 \times 50\,\mathrm{mm}^3$. The macromesh was generated by CUBIT [15] and consists of 70000 tetrahedra with 17000 grid points and the average tetrahedra volume is $2.30\,\mathrm{mm}^3$.

Arithmetic and harmonic mean. We also compare the results of the FE-HMM with numerical results obtained from tensor averaged using more naive averaging procedures as the arithmetic or harmonic means. To compute these averaged tensors, we keep the macro variables fixed and average the tensor elementwise over the microscale only.

$$a_{ij}^{\text{arithmetic}}(x_k) = \int_Y a_{ij}(x_k, y)\, dy, \tag{48}$$

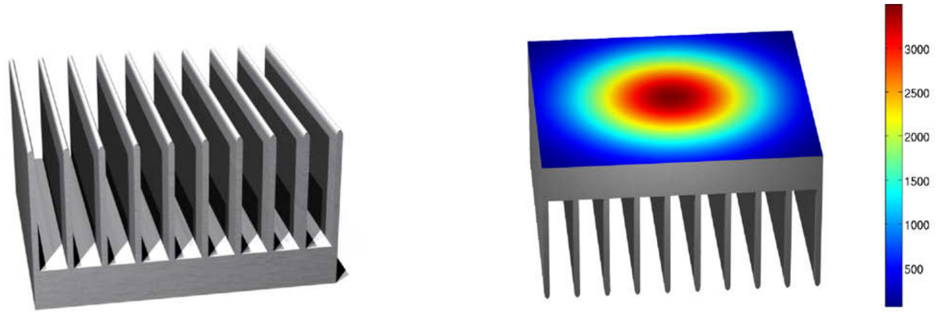
$$a_{ij}^{\text{geometric}}(x_k) = \left(\int_Y (a_{ij}(x_k, y))^{-1}\, dy \right)^{-1}, \tag{49}$$

where Y is the $3d$ unit cube. This leads to the following tensors

$$a^{\text{arithmetic}}(x) = \begin{pmatrix} 140.0 \cdot e^{10x_3} & 35 & 0 \\ 35 & 140.0 \cdot e^{10x_3} & 0 \\ 0 & 0 & 200 \end{pmatrix}, \tag{50}$$

$$a^{\text{harmonic}}(x) = \begin{pmatrix} 100.0 \cdot e^{10x_3} & 35 & 0 \\ 35 & 100.0 \cdot e^{10x_3} & 0.0 \\ 0 & 0 & 200 \end{pmatrix}. \tag{51}$$

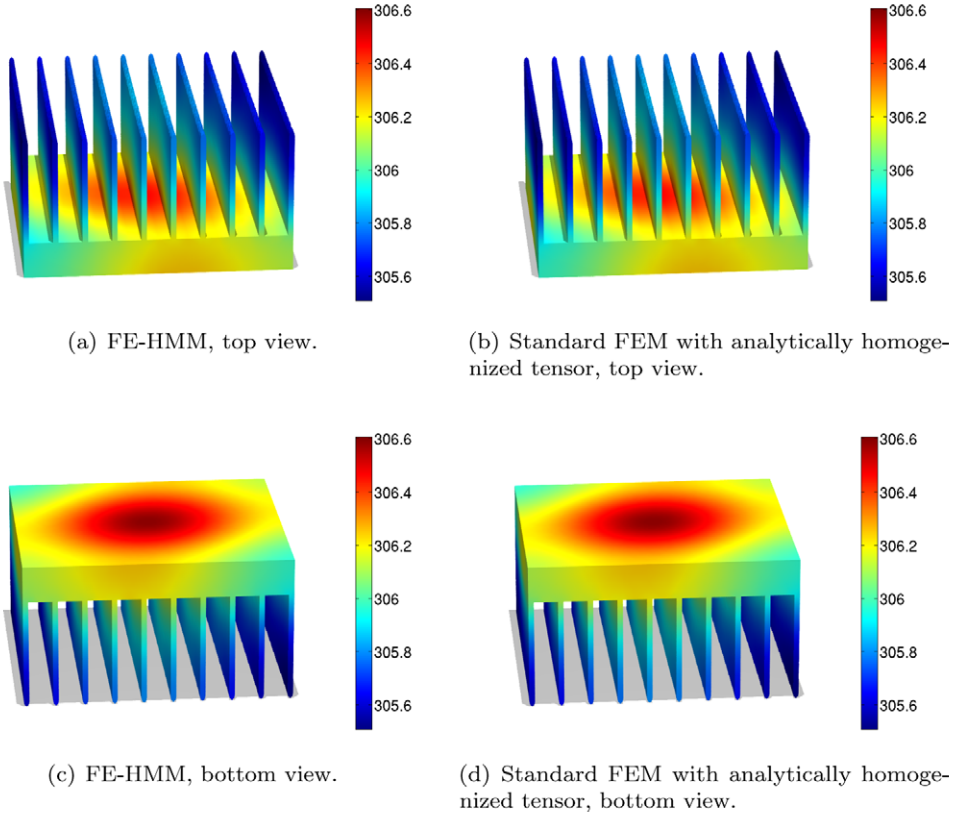
Compared to the homogenized tensor, we clearly see that the tensor obtained through arithmetic average overestimates the conductivity of the material while the tensor obtained through harmonic average underestimates the conductivity of the material.



(a) Heatsink, which can be mounted on a microprocessor to dissipate heat.

(b) Boundary conditions. The side shown in color is connected to the processor package and represent incoming heat flux. Neumann boundary conditions (43) are assigned to this face. The color bar refers to the incoming heat flux from the processor (measured in $\frac{\mathrm{W}}{\mathrm{m}^2}$), reflecting the fact the major part of the incoming flux is located close to the processor core. The remaining (grey) part is modeled with Robin boundary conditions (42).

Fig. 4.11. Model of the heatsink simulated in Section 4.4.



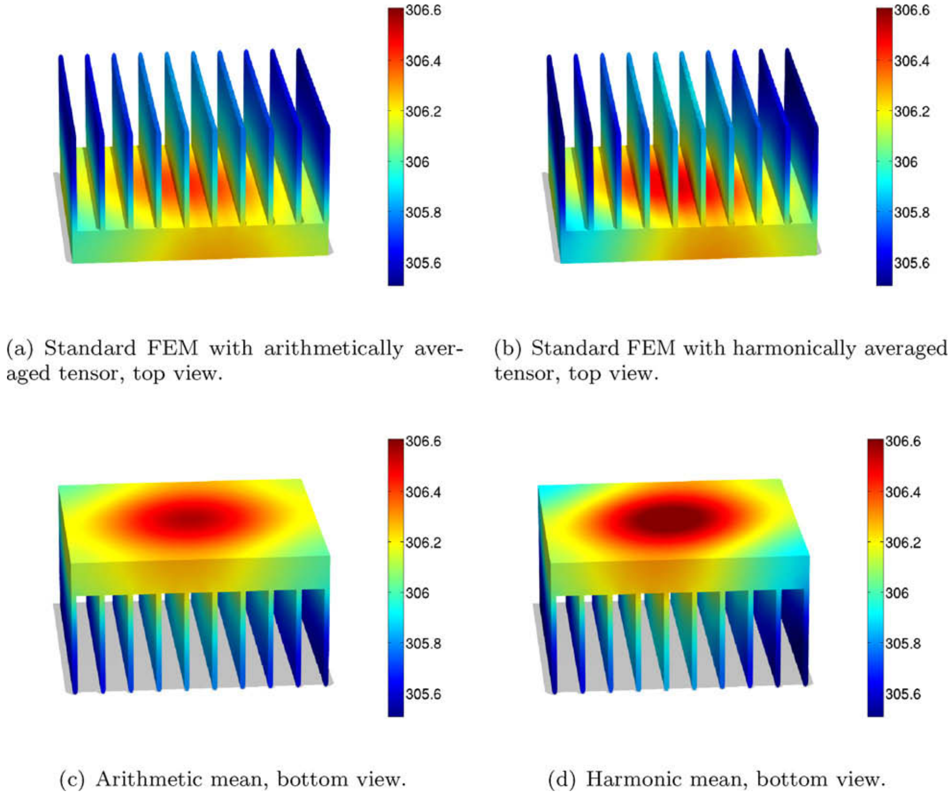
(a) FE-HMM, top view.

(b) Standard FEM with analytically homogenized tensor, top view.

(c) FE-HMM, bottom view.

(d) Standard FEM with analytically homogenized tensor, bottom view.

Fig. 4.12. Steady-state heat distribution in a heatsink (problem described in Section 4.4). Comparison of the results obtained with the FE-HMM and a standard FEM solved with the analytically tensor (47). The macromesh used in both computations has 17000 DOF. For the FE-HMM, we choose $\varepsilon = 10^{-5}$ and $N_{\text{mic}} = 16$ (as the FE-HMM is independent of ε (in the periodic case) any other value of ε gives the same results). The color bar represents the temperature in K.



(a) Standard FEM with arithmetically averaged tensor, top view.

(b) Standard FEM with harmonically averaged tensor, top view.

(c) Arithmetic mean, bottom view.

(d) Harmonic mean, bottom view.

Fig. 4.13. Steady-state heat distribution in a heatsink (problem described in Section 4.4). Comparison between the results obtained with an arithmetic and harmonic average of the multiscale tensor (46). We can see a significant difference with the results obtained using the homogenized tensor presented in Fig. 4.12. The macromesh used in both computations has 17000 DOF. The color bar represents the temperature in K.

Table 4.6
Energy norm, minimal value and infinity norm for the solution of the problem described in Section 4.4. Comparison of the solutions obtained by the FE-HMM, a standard FEM with homogenized tensor (47), a standard FEM with tensor obtained by arithmetic or harmonic averages. The macromesh used in all computations has 17000 DOF (u represents the temperature in K).

		$N_{\text{mic}} = 4$	$N_{\text{mic}} = 8$	$N_{\text{mic}} = 16$
FE-HMM	$\|u\|_A$	2.1668	2.1671	2.1671
	$\inf_{x\in\Omega}\|u(x)\|$	305.527	305.527	305.527
	$\|u\|_\infty$	306.579	306.58	306.58
		Homogenized	Arithmetic mean	Harmonic mean
Standard FEM	$\|u\|_A$	2.1874	2.1586	2.2516
	$\inf_{x\in\Omega}\|u(x)\|$	305.509	305.54	305.43
	$\|u\|_\infty$	306.607	306.57	306.68

4.4.2. Results

We compare in Fig. 4.12 the results obtained with the FE-HMM to the solution of the FEM with the homogenized tensor. We see a very good qualitative agreement between the two solutions. Both solutions show an interesting effect, that is, the heatsink is more conductive in one diagonal direction. This is due to the anisotropy of the (homogenized) conductivity tensor.

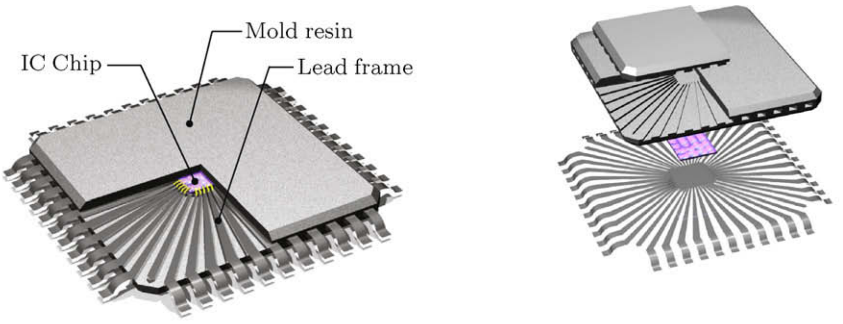
Fig. 4.13 shows the solutions obtained using the arithmetic and harmonic means. As expected by the over- and under-estimation of these conductivity tensors, a comparison to the homogenized solution presented in Fig. 4.12 shows a significant discrepancy.

Table 4.6 provides more insights in the quality of the solution of the FE-HMM compared to the homogenized solution, by providing the values of $\|u\|_\infty$, $\inf_{x\in\Omega}|u(x)|$ and the energy norm $\|u\|_a$ for various numerical experiments.

4.5. Three dimensional problem: heat distribution in a microprocessor

To demonstrate the versatility of our code we finally present a real-world application, namely the steady-state temperature distribution in an embedded microprocessor as used e.g. in cellphones or automotive electronics. The increasing packaging density in microprocessors and the resulting growth in temperature requires new cooling methods, which are crucial for the efficiency and long term reliability of micro chips. The use of composite materials for leadframes and packaging material (see Fig. 4.14) is crucial for the efficiency of the microprocessor. Understanding the effective properties of such devices is therefore of prime importance.

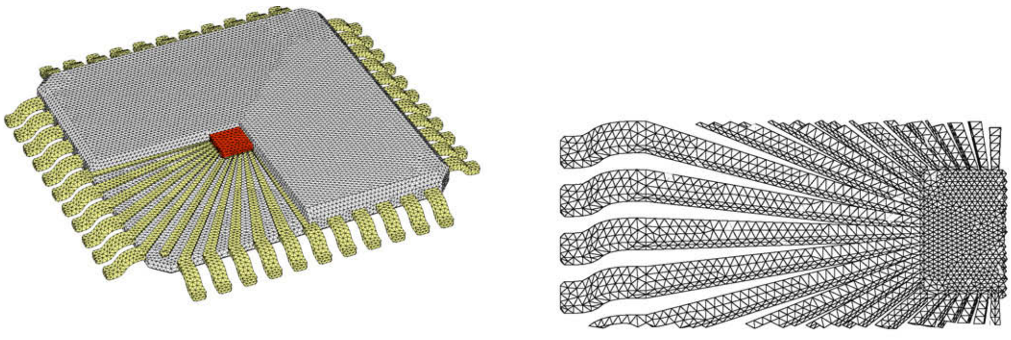
For example, various components of different materials (as metal and silicon) in a microprocessor must have matching coefficients of thermal expansion (CTE), otherwise thermal stresses



(a) Model of a microprocessor.

(b) Components of the simulated microprocessor.

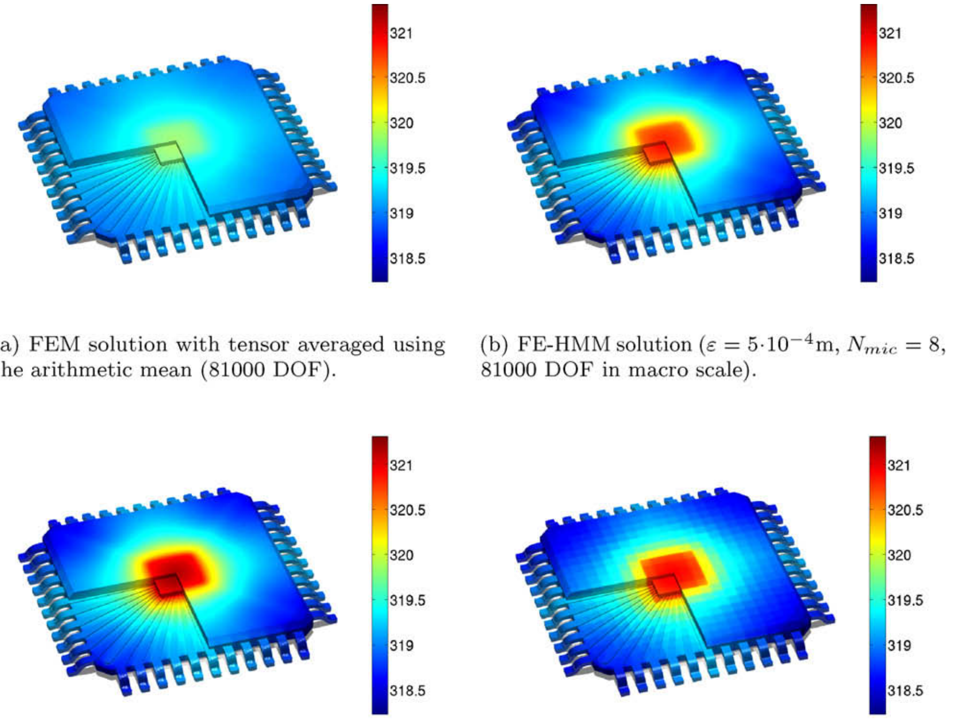
Fig. 4.14. Model of the microprocessor used in Section 4.5.



(a) Mesh of different components of the full processor, one corner is removed here to allow for inside view.

(b) Detailed view of the mesh used for the leadframe.

Fig. 4.15. Coarse finite element mesh used in the 3D microprocessor problem described in Section 4.5 with a total of 81000 grid points and 430000 tetrahedra.



(a) FEM solution with tensor averaged using the arithmetic mean (81000 DOF).

(b) FE-HMM solution ($\varepsilon = 5 \cdot 10^{-4}$m, $N_{mic} = 8$, 81000 DOF in macro scale).

(c) FE-HMM solution ($\varepsilon = 5 \cdot 10^{-4}$m, $N_{mic} = 16$, 81000 DOF in macro scale).

(d) Finescale solution. Due to the large ε, the oscillations in the micro scale lead to the checkered pattern ($\varepsilon = 5 \cdot 10^{-4}$m, $3.9 \cdot 10^{6}$ DOF).

Fig. 4.16. Steady state heat distribution in the microprocessor as sketched in Fig. 4.14 for the problem described in Section 4.5. Comparison of the solutions obtained by a FEM with arithmetic averaging of the tensor, the FE-HMM and a finescale FEM. The color bar represents the temperature in K.

might cause component failure. Given that the IC chip is made of silicon, it is important to use a material for the leadframe which has a compatible CTE. Unfortunately, the obvious candidates for constructing the leadframe (such as copper or aluminium) have a CTE much higher than silicon. The way out of this problem is to use composite materials. For example nickel-iron alloys are widely used for leadframes. A second problem is to find composite materials (with matching CTEs) featuring a high thermal conductivity at the same time. We refer to [35] for details and an overview of modern composite materials with large thermal conductivities.

We will see on the example below, that our MATLAB code, although simple and short, is nevertheless capable of handling such challenging problems.

We consider a slightly simplified model of a modern embedded microprocessor which consists of the silicon IC chip, a leadframe, which in our example acts as a heat spreader, and a mold resin encapsulation (packaging), which covers the IC chip for protection (see Fig. 4.14). While the simplified model does not reflect effects such as air flow over the package or heat conduction through the board on which the chip is mounted, these effects could be simulated without further difficulties with the proposed code. However, since we again want to avoid discussing modelling issues, we omit such effects originating from a more refined model.

4.5.1. Setup

As in Section 4.4, we use Fourier's law of cooling for the heat conduction within the microprocessor. Convection on the surface is again described by Robin boundary conditions. While conductivity tensors for realistic materials obtained via imaging techniques could be taken into account without further difficulties, we use simplified tensors here. To simulate the various composite materials, we use different conductivity tensors for each material. The diagonal entries of these tensors are given by

$$a^{\varepsilon}_{ii,\text{leadframe}}(x) = 400\cos\left(2\pi\frac{x_i}{\varepsilon}\right) + 400\exp\left(20\sqrt{x_1^2+x_2^2}\right),$$

$$a^{\varepsilon}_{ii,\text{resin}}(x) = \begin{cases} 1.9\cos\left(2\pi\frac{x_i}{\varepsilon}\right) + 2 & i = 1,2, \\ 3.8\cos\left(2\pi\frac{x_i}{\varepsilon}\right) + 4 & i = 3, \end{cases}$$

$$a^{\varepsilon}_{ii,\text{chip}}(x) = 140.$$

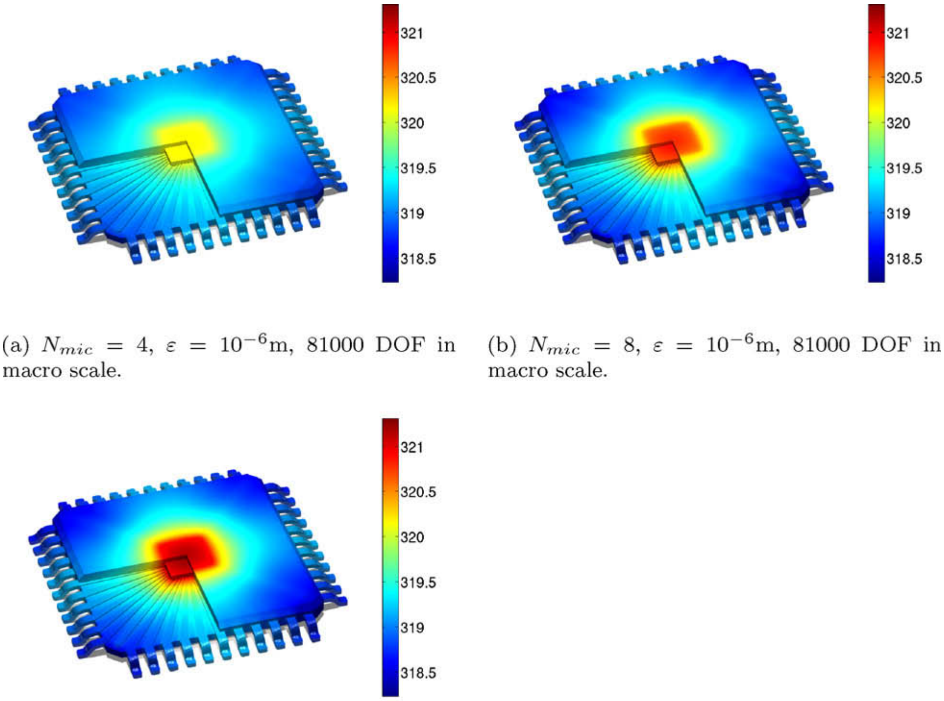
Notice that we center the chip at the origin $x = (0,0,0)$ of the coordinate system. The tensor corresponding to the leadframe has a non-periodic slow variation, which models a change in the material structure from the center of the leadframe to the periphery. The tensor corresponding to the resin is chosen to be oscillating and anisotropic with a larger conductivity in the z-direction. Finally, the conductivity for the chip is assumed to be constant. There is no explicit analytical solution to this homogenization problem.

We assume a power of the chip of $P_{\text{chip}} = 0.125$ W and consider a chip size of $V = 2 \times 2 \times 0.2$ mm^3, which leads to an external heat flux f in (41) of $f = \frac{P}{V} = 1.875 \cdot 10^8 \ \frac{\text{W}}{\text{m}^3}$. We set $q_0 = 0$.

The ambient temperature is set to $u_{\text{amb}} = 293.15$ K[3] and we fix the heat transfer coefficient $\alpha = 20 \ \frac{\text{W}}{\text{m}^2\text{K}}$, being a rough estimate for air cooling.

The macromesh was again generated by CUBIT and consists of 81000 grid points with a maximum tetrahedra volume of $1.4 \cdot 10^{-3}$ mm^3 (see Fig. 4.15). The size of the full processor model is $12.2 \times 12.2 \times 1$ mm^3. In many realistic materials the heterogeneities have a size of order 10^{-6} m, so we set $\varepsilon = 10^{-5}$ m and

[3] The heat in the immediate vicinity of the device can be much higher than room temperature. This should be reflected in a more realistic simulation.



(a) $N_{mic} = 4$, $\varepsilon = 10^{-6}$m, 81000 DOF in macro scale.

(b) $N_{mic} = 8$, $\varepsilon = 10^{-6}$m, 81000 DOF in macro scale.

(c) $N_{mic} = 16$, $\varepsilon = 10^{-6}$m, 81000 DOF in macro scale.

Fig. 4.17. Steady state heat distribution in the microprocessor as sketched in Fig. 4.14 for the problem described in Section 4.5. Comparison of FE-HMM solutions with different resolutions of the microproblems. The color bar represents the temperature in K.

Table 4.7
Energy norm, minimal and maximum value for the solution of the 3D problem described in Section 4.5. Comparison of the solutions obtained by a FEM with arithmetic averaging of the tensor, the FE-HMM and a finescale FEM. Values are temperatures in K.

		$N_{\text{mic}} = 4$	$N_{\text{mic}} = 8$	$N_{\text{mic}} = 16$	Finescale	Average
$\varepsilon = 5 \cdot 10^{-4}$	$\|u\|_A$	0.3906	0.4961	0.5514	0.5189	0.3122
	$\inf_{x\in\Omega}\|u(x)\|$	318.688	318.456	318.350	318.23	318.843
	$\|u\|_\infty$	320.241	320.9	321.309	320.97	319.851
$\varepsilon = 10^{-5}$	$\|u\|_A$	0.3894	0.4963	0.5512	–	0.3122
	$\inf_{x\in\Omega}\|u(x)\|$	318.688	318.454	318.3506	–	318.843
	$\|u\|_\infty$	320.234	320.901	321.3081	–	319.851

choose $\delta = 10^{-5}$ m. With this δ and about 10 grid points (in each direction) per oscillation length to resolve the microscopic composite heterogeneities, a full fine scale computation would lead to more than 10^{11} grid points and would thus be computationally out of reach. For comparison purpose, we choose a relatively large $\varepsilon = 5 \cdot 10^{-4}$ m to generate a reference fine scale solution on a mesh consisting of 3.9 million grid points and about 22 million tetrahedra. As we will see below, computations with realistic values of ε can be done without difficulties with the FE-HMM.

4.5.2. Results

We compare in Fig. 4.16 a solution obtained by taking the arithmetic mean of the fast scale of the conductivity tensors, the solution of the FE-HMM and a fine-scale solution. We also compute a solution with the FE-HMM with $\varepsilon = 10^{-6}$ m. We notice, as expected, that a naive average (such as the arithmetic average) gives a completely false result by overestimating the conductivity of the microprocessor.

We observe in Fig. 4.16 a very good qualitative agreement between the FE-HMM and the fine-scale solution. The results of the FE-HMM are independent of ε, as can be seen by comparing the results with $\varepsilon = 5 \cdot 10^{-4}$ m and the results for small (realistic) heterogenities (of order 10^{-6} m, thus $\varepsilon = 10^{-5}$ m). This is again expected as the FE-HMM captures the effective (homogenized) solution of the multiscale problem (41).

Finally, we present in Fig. 4.17 further results for the FE-HMM (with $\varepsilon = 10^{-5}$ m), where we keep the macro DOF fixed (81000 grid points), while varying the micro degrees of freedom N_{mic}. We observe that the influence of the micro error at the macroscale is not negligible. If the small scales are not resolved on the sampling domain, a significant error occurs for the macro FE-HMM solution.

To get a rough estimate of the quality of the various experiments performed in this section, we provide in Table 4.7 the values of $\min_{x\in\Omega}|u(x)|, \|u\|_\infty$ and the energy norm $\|u\|_a$ for the various numerical solutions.

5. Conclusion

In this paper we have presented a short and versatile multiscale FE solver for PDEs (elliptic, parabolic) with highly oscillating coefficients. While many multiscale strategies have been developed in the past few years for multiscale PDEs very few detailed numerical implementations were available so far. The algorithm is based on the heterogeneous multiscale method, which gives a methodology to couple efficiently macro- and microsolvers. We have discussed in detail the implementation of the multiscale FE solver and have shown that the multiscale strategy can be built on the core structure of a standard FE code. The code also allows also for different triangulations (simplicial, quadrilateral) and can trivially be parallelized. Implementation of time-dependent problems has also been discussed.

To illustrate the performance of our code, we have presented a number of numerical examples for both two and three dimensional problems with periodic, non-uniformly periodic and random tensors. Various boundary conditions and coupling conditions between macro and micro FE solvers have been discussed. Time-dependent problems have also been presented. These results show that, although simple, our code can nevertheless handle challenging problems. We believe that the simplicity and the versatility of the proposed code could be useful for further developments in computational methods for multiscale PDEs, and that our code or algorithm could easily be integrated as a subroutine for more general multiscale computational problems (e.g. involving optimization procedures, inverse or stochastic problems).

Acknowledgement

The work of A. Abdulle was partially supported by an EPSRC Advanced Fellowship EP/E05207X/1.

References

[1] A. Abdulle, Fourth order Chebyshev methods with recurrence relation, SIAM J. Sci. Comput. 23 (2002) 2042–2055.
[2] A. Abdulle, C. Schwab, Heterogeneous multiscale FEM for diffusion problem on rough surfaces, SIAM Multiscale Model. Simul. 3 (1) (2005) 195–220.
[3] A. Abdulle, On a-priori error analysis of fully discrete heterogeneous multiscale FEM, SIAM Multiscale Model. Simul. 4 (2) (2005) 447–459.
[4] A. Abdulle, Analysis of a heterogeneous multiscale FEM for problems in elasticity, Math. Mod. Meth. Appl. Sci. (M3AS) 16 (2) (2006) 615–635.
[5] A. Abdulle, Heterogeneous Multiscale Methods with Quadrilateral Finite Elements, in: Numerical Mathematics and Advanced Applications, Springer, Berlin, 2006, pp. 743–751.
[6] A. Abdulle, B. Engquist, Finite element heterogeneous multiscale methods with near optimal computational complexity, SIAM Multiscale Model. Simul. 6 (2007) 1059–1084.
[7] A. Abdulle, Multiscale method based on discontinuous finite element methods for homogenization problems, C.R. Acad. Sci. Paris, Ser. I 346 (2007) 97–102.
[8] A. Abdulle, The finite element heterogeneous multiscale method: a computational strategy for multiscale PDEs, GAKUTO Int. Ser. Math. Sci. Appl. 31 (2009) 135–184.
[9] J. Alberty, C. Carstensen, S.A. Funken, Remarks around 50 lines of Matlab: short finite element implementation, Numer. Algorithms 20 (2–3) (1999) 117–137.
[10] A. Bensoussan, J.-L. Lions, G. Papanicolaou, Asymptotic Analysis for Periodic Structures, North-Holland, Amsterdam, 1978.
[11] J. Biercuk et al., Carbon nanotube composites for thermal management, Appl. Phys. Lett., AIP 80 (15) (2002) 2767.
[13] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland, 1978.
[14] D. Cioranescu, P. Donato, An Introduction to Homogenization, Oxford University Press, 1999.
[15] <http://cubit.sandia.gov>.
[16] V.V. Jikov, S.M. Kozlov, O.A. Oleinik, Homogenization of Differential Operators and Integral Functionals, Springer-Verlag, Berlin, Heidelberg, 1994.
[17] V. Kouznetsova, W.A.M. Brekelmans, F.P.T. Baaijens, An approach to micro–macro modeling of heterogeneous materials, Comput. Mech. 27 (2001) 37–48.
[18] V. Kouznetsova, M. Geers, W. Brekelmans, Multi-scale second-order computational homogenization of multi-phase materials: a nested finite element solution strategy, Comput. Methods Appl. Mech. Engrg. 193 (2004) 5525–5550.
[19] B. Engquist, The heterogeneous multi-scale methods, Commun. Math. Sci. 1 (1) (2003) 87–132.
[20] P. Ming, P. Zhang, Analysis of the heterogeneous multiscale method for elliptic homogenization problems, J. AMS 18 (1) (2004) 121–156.
[21] B. Engquist, X. Li, W. Ren, E. Vanden-Eijnden, The heterogeneous multiscale method: a review, Commun. Comput. Phys. 2 (2007) 367–450.
[22] J. Fish, V. Belsky, Multigrid method for periodic heterogeneous media. I. Convergence studies for one-dimensional case, Comput. Methods Appl. Mech. Engrg. 126 (1995) 1–16.
[23] J. Fish, V. Belsky, Multigrid method for periodic heterogeneous media. II. Multiscale modeling and quality control in multidimensional case, Comput. Methods Appl. Mech. Engrg. 126 (1995) 17–38.
[24] Viet Ha Hoang, Christoph Schwab, High-dimensional finite elements for elliptic problems with multiple scales, SIAM Multiscale Model. Simulat. 3 (1) (2005) 195–220.
[25] T. Hou, X. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, J. Comput Phys. 134 (1997) 169–189.
[26] V. Kippe, J.E. Aarnes, K.-A. Lie, A comparison of multiscale methods for elliptic problems in porous media flow, Comput. Geosci., 2007, (Special issue on multiscale methods).
[27] C. Miehe, Computational micro-to-macro transitions for discretized micro-structures of heterogeneous materials at finite strains based on the minimization of averaged incremental energy, Comput. Methods Appl. Mech. Engrg. 192 (2003) 559–591.
[28] P. Ming, Y. Yue, Numerical methods for multiscale elliptic problems, J. Comput. Phys. 214 (1) (2006) 421–445.
[29] J.T. Oden, S. Prudhomme, A. Romkes, P. Bauman, Multiscale modeling of physical phenomena: adaptive control of models, SIAM J. Sci. Comput. 28 (2006) 2359–2389.
[30] A.M. Matache, C. Schwab, Two-scale FEM for homogenization problems, R.A.I.R.O. Anal. Numer. 36 (2002) 537–572.
[31] E. Sanchez-Palencia, Homogenization Techniques for Composite Media, Springer-Verlag, New York, 1987.
[32] K. Terada, N. Kikuchi, A class of general algorithm for multi-scale analyses of heterogeneous media, Comput. Methods Appl. Mech. Engrg. 190 (2001) 5427–5464.
[33] T.C. Wallstrom, S. Hou, M.A. Christie, L.J. Durlofsky, D.H. Sharp, Accurate scale up of two phase flow using renormalization and nonuniform coarsening, Comput. Geosci. 3 (1) (1999) 67–87.
[34] Y. Yue, W. E, The local microscale problem in the multiscale modeling of strongly heterogeneous media: effects of boundary conditions and cell size, J. Comput. Phys. 222 (2) (2006) 556–572.
[35] C. Zweben, Advances in composite materials for thermal management in electronic packaging, in: JOM Journal of the Minerals, Metals and Materials Society, vol. 50, 1998, Springer, pp. 47–51.