On the Implementation of a Heterogeneous Multi-scale Finite Element Method for Nonlinear Elliptic Problems

Patrick Henning and Mario Ohlberger

Abstract. In this contribution, we formulate a heterogeneous multiscale finite element method (HMM) for monotone elliptic operators. This is done in the general concept of HMM, which was initially introduced by E and Engquist [E, Engquist, *The heterogeneous multiscale methods*, Commun. Math. Sci., 1(1):87–132, 2003]. Since the straightforward formulation is not suitable for a direct implementation in the nonlinear setting, we present a corresponding algorithm, which involves the computation of additional cell problems. The algorithm is validated by numerical experiments and can be used to effectively determine homogenized solutions.

1 Introduction

In this work, we are concerned with monotone elliptic multiscale problems of the following structure:

$$\begin{aligned} -\nabla \cdot A^{\varepsilon}(x, \nabla u^{\varepsilon}) &= f \text{ in } \Omega, \\ u^{\varepsilon} &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{1}$$

Here, A^{ε} describes a rapidly oscillating, nonlinear diffusion operator, where the parameter ε represents the size of the microscale. For instance, if the oscillations are periodic, then ε denotes the period. These types of equations describe a large variety of engineering problems, such as transport of pollutants in groundwater or the conductivity of compositional materials. Resolving the entire microstructure, in order to perform a highly expensive fine-scale computation, typically results in an intractable

Patrick Henning · Mario Ohlberger
Institut für Numerische und Angewandte Mathematik, Einsteinstraße 62,
48149 Münster, Germany
e-mail: patrick.henning@math.uni-muenster.de,
mario.ohlberger@uni-muenster.de

computational demand. To reduce the complexity of the problem significantly, equations such as (1) are often treated by so called multiscale methods, which include a large range of different approaches. In this work, we restrict ourselves to a realization of the *heterogeneous multiscale finite element method* (HMM). The concept of HMM was initially introduced by E and Engquist in 2003 [5] and extended by several contributions. A good overview on this can be found in the work of Abdulle [1]. A HMM realization with near optimal computational complexity is given by Abdulle and Engquist [2], a version for the treatment of advection-diffusion problems with large drift by Henning and Ohlberger [9]. A-posteriori error estimates are derived in the work of Henning and Ohlberger [11, 7, 8]. In the framework of Multiscale finite element methods (MsFEM) Efendiev, Hou and Ginting [6] introduce a multiscale finite element method for nonlinear problems. The HMM for monotone operators, that we present in this contribution, is formulated in analogy to the linear case as presented in [5]. However, if we are interested in implementing the method, we face the problem, that the formulation is not yet suitable for programming purposes since it is not clear how to efficiently compute the Jacobian matrix in a Newton scheme. In the last part of Section 3 we discuss this issue in detail. In fact, we figure out that the usage of an accessory Newton method results in the assembling of additional cell problems. Considering these new problems, we present a general algorithm for solving the discrete HMM problem for monotone operators. This algorithm is implemented using the software toolbox DUNE, which is for instance introduced in [3]. In particular, we employ the DUNE module DUNE-FEM (see [4]).

2 Analytical and Discrete Setting

Let $\Omega \subset \mathbb{R}^N$ denote a polygonal bounded domain with dimension $N \leq 3$. By $\mathring{H}^1(\Omega)$ we define the space of $H^1(\Omega)$-functions with compact support in Ω. In the following, we are concerned with solving the following elliptic multiscale problem:

Problem 0.1 (Nonlinear elliptic equation with fast oscillations). Find $u^\varepsilon \in \mathring{H}^1(\Omega)$:

$$\int_\Omega A^\varepsilon(x, \nabla u^\varepsilon) \cdot \nabla \Phi(x)\, dx = \int_\Omega f(x)\Phi(x)\, dx \qquad \forall \Phi \in \mathring{H}^1(\Omega). \tag{2}$$

Here, $A^\varepsilon : \Omega \times (L^2(\Omega))^N \to (L^2(\Omega))^N$ denotes a nonlinear diffusion operator, which meets the subsequent monotonicity and continuity conditions: there exist two constants $0 < \alpha \leq \beta < \infty$ such that for all $x \in \Omega$ and all $\xi_1, \xi_2 \in \mathbb{R}^N$:

$$\begin{aligned} < A^\varepsilon(x,\xi_1) - A^\varepsilon(x,\xi_2), \xi_1 - \xi_2 > &\geq \alpha|\xi_1 - \xi_2|^2, \\ |A^\varepsilon(x,\xi_1) - A^\varepsilon(x,\xi_2)| &\leq \beta|\xi_1 - \xi_2| \text{ and} \\ A^\varepsilon(x,0) &= 0. \end{aligned}$$

With these assumptions, we have a unique solution of problem (2) (see e.g. [10]).

In order to create a suitable discrete setting, in which we can formulate the heterogeneous multiscale method, we denote by $\mathscr{T}_H$ a regular simplicial partition of Ω with elements T. x_T is the barycenter of $T \in \mathscr{T}_H$. By $Y = (-\frac{1}{2}, \frac{1}{2})^N$ we define the 0-centered unit cube. Scaling Y with $\kappa \in \mathbb{R}_{>0}$ and shifting it by $x_T \in T$, we introduce the notation $Y_{T,\kappa} := \{x + \kappa y|\ y \in Y\}$. By $\mathscr{T}_h$, we denote a regular, periodic partition of Y. y_K is the barycenter of $K \in \mathscr{T}_h$. The mapping $x_T^\kappa : Y \to Y_{T,\kappa}$ is given by $x_T^\kappa(y) := x_T + \kappa y$. Moreover, we introduce the following discrete spaces, where $\mathbb{P}^1$ is the space of polynomials of degree 1:

$$\begin{aligned}
V_H(\Omega) &:= \{\Phi_H \in \mathring{H}^1(\Omega) \cap C^0(\Omega)\ \ |\Phi_{H_{|T}} \in \mathbb{P}^1(T)\ \ \forall T \in \mathscr{T}_H\};\\
W_h(Y) &:= \{\phi_h \in C^0(Y)|\ \phi_h \text{ is } Y\text{-periodic, } \int_Y \phi_h = 0 \text{ and } \phi_{h|K} \in \mathbb{P}^1(K)\ \forall K \in \mathscr{T}_h\};\\
W_h(Y_{T,\kappa}) &:= \{\phi_h \in C^0(Y_{T,\kappa})|\ (\phi_h \circ x_T^\kappa) \in W_h(Y)\}.
\end{aligned}$$

Definition 0.1 (Reconstruction operator). In order to locally reconstruct information of the fine-scale behavior of u^ε, we introduce the *local reconstruction operator* $R_h^T : V_H(\Omega) \to V_H(\Omega) + W_h(Y_{T,\delta})$. For $\delta > 0$ and $\Phi_H \in V_H(\Omega)$, the corresponding reconstruction $R_h^T(\Phi_H) \in \Phi_H + W_h(Y_{T,\delta})$ is defined as the solution of

$$\int_{Y_{T,\delta}} A^\varepsilon\left(x, \nabla_x R_h^T(\Phi_H)(x)\right) \cdot \nabla_x \phi_h(x)\ \ dx = 0 \qquad \forall \phi_h \in W_h(Y_{T,\delta}). \tag{3}$$

Note that R_h^T is well defined since (3) always has a unique solution, if $Y_{T,\delta} \subset T$. We also point out that (3) is the discrete analog to the well-posed so-called cell problems known from standard homogenization theory (cf. [12]).

3 The Heterogeneous Multiscale Finite Element Method for Monotone Operators

The general idea behind the concept of HMM is based on a scale separation of the solution of problem (2). If we assume, that u^ε can be separated into its coarse- and fine-scale part, we can artificially separate the test functions in the same way and postulate $\Phi = \Phi^{coarse} + \phi^{fine}$, where ϕ^{fine} remains very small in the L^2-norm, but produces large gradients due to fast oscillations. Together with (2) and choosing $\phi^{fine} = 0$ we obtain

$$\int_\Omega A^\varepsilon(\cdot, \nabla u^\varepsilon) \cdot \nabla \Phi^{coarse} = \int_\Omega f \Phi^{coarse}. \tag{4}$$

On the other hand, with $\Phi^{coarse} = 0$ we get:

$$\int_\Omega A^\varepsilon(\cdot, \nabla u^\varepsilon) \cdot \nabla \phi^{fine} = \int_\Omega f \phi^{fine} \approx 0. \tag{5}$$

Discretizing (4) and (5), restricting the fine-scale computations on representative cells $x_T + \delta Y$ and reconstructing the macroscopic test function in order to guarantee uniqueness and existence, gives us the canonical formulation of HMM, as initially suggested by E and Engquist [5] for linear elliptic equations. In analogy, we can state the heterogeneous multiscale method for monotone operators:

Definition 0.2 (Heterogeneous Multiscale Method for monotone operators). We define the HMM approximation u_H of u^ε by: $u_H \in V_H(\Omega)$ solves

$$(f_H, \Phi_H)_{L^2(\Omega)} = A_H(u_H, \Phi_H) \qquad \forall \Phi_H \in V_H(\Omega), \tag{6}$$

with

$$A_H(u_H, \Phi_H) := \sum_{T \in \mathscr{T}_H} |T| \fint_{Y_{T,\varepsilon}} A^\varepsilon \left(x, \nabla_x R_h^T(u_H)(x)\right) \cdot \nabla_x R_h^T(\Phi_H)(x) \;\, dx. \tag{7}$$

Here, R_h^T denotes the local reconstruction operator, as it has been is defined in (3). For the oversampling parameter δ, we assume $\delta \geq \varepsilon$. An expedient choice for the periodic case could be $\delta = \varepsilon$, for the non-periodic case $\delta = 2\varepsilon$.

Note, for every $\delta \geq \varepsilon$, the HMM produces a unique solution $u_H \in V_H(\Omega)$. This is a direct effect of the monotonicity of $A^\varepsilon(x, \cdot)$. For the case that $A^\varepsilon(x, \cdot)$ is a linear operator, (7) results in a linear system of equations, which is symmetric and therefore cheap to solve.

If we assume, that the nonlinear diffusion operator A^ε only contains periodic oscillations, the homogenization of problem (2) is well known. However, even in this scenario, it is not immediately clear, how to carry over the analytical results to a suitable discretization. That is why the heterogeneous multiscale finite element method above becomes an effective scheme to determine the homogenized solution of elliptic multiscale problems with a periodically oscillating, monotone diffusion operator. In fact, with slight modifications, we can show, that the HMM is equivalent to a straightforward discretization of the standard homogenized equation of (2).

Problems Concerning the Implementation

In order to determine the effect of the reconstruction operator R_h^T on a function Φ_H, we need to solve the problems given by (3). Each of these so-called cell problems is a standard nonlinear elliptic problem, which can be solved easily by common methods. However, the situation is completely different for the HMM macro problem. In order to solve (6) numerically, we might face difficulties, depending on the structure of the nonlinearity. This is due to the fact that we do not only have one nonlinearity produced by A^ε, but an additional nonlinearity within, produced by the reconstruction operator R_h^T. In contrast to A^ε, an evaluation of R_h^T is not cheap, which is why we require a special treatment for this equation. This claim will be emphasized, if we have a closer look at (6). Let $\Phi_1, ..., \Phi_M$ denote the Lagrange base of V_H. Defining $G: \mathbb{R}^M \to \mathbb{R}^M$ by

$$
\begin{aligned}
(G(\alpha))_i := \sum_{T\in\mathscr{T}_H} |T| \fint_{Y_{T,\varepsilon}} & A^{\varepsilon}(x, \nabla_x R_h^T(\sum_j^M \alpha_j \Phi_j)(x)) \cdot \nabla_x R_h^T(\Phi_i)(x)\ \ dx \\
& - \int_{\Omega} f(x)\Phi_i(x)\, dx,
\end{aligned}
$$

we are looking for $\bar{\alpha}$ with $G(\bar{\alpha}) = 0$. Solving this numerically (with an iterative scheme) requires either the computation of the Jacobian matrix $D_\alpha G$ or a corresponding approximation. There are two possibilities:

1. If we want to make direct use of $D_\alpha G$, we require the effect of the Fréchet derivative of the reconstruction operator, i.e. DR_h^T, which we can not determine explicitly.
2. If we approximate $D_\alpha G$ by difference quotients $\frac{(G(\alpha+he_j)-G(\alpha))_i}{h}$, we do not only need to determine $R_h^T(\sum_j^M \alpha_j \Phi_j)$, but also $R_h^T(\sum_j^M(\alpha_j + \sigma_{ji}h)\Phi_j)$ for any relevant i. Each of this involves solving a nonlinear elliptic problem for every $T \in \mathscr{T}_H$. Moreover this has to be done for every iteration step carried out for finding the solution of $G(\bar{\alpha}) = 0$. Therefore, a very high computational demand is implied.

In any case, we are dealing with accessory problems. In the next section we present a corresponding solution.

4 Implementation

In this section we want to state an algorithm in order to compute the solution of the HMM macro problem given by equation (6). This algorithm is a combination of HMM and Newton Method, which in particular can be used for solving the nonlinear elliptic multiscale problem (2). It is derived and stated in Subsection 4.1. In Subsection 4.2 we comment on its implementation in the DUNE framework.

4.1 HM Newton Method

We now consider a general treatment of the HMM macro problem, given by equation (6). First of all we emphasize that this treatment strongly depends on the structure of A^ε and should be modified according to problem specific characteristics. For instance, if we are dealing with a linear diffusion operator, i.e. $A^\varepsilon(x,\xi) = \bar{A}^\varepsilon(x)\xi$ with a matrix $\bar{A}^\varepsilon$, the implementation of (6) is straightforward, since (6) directly yields a linear system of equations. For the case that the nonlinearity is of the form $A^\varepsilon(x,\xi) = \bar{A}^\varepsilon(x,\xi)\xi$, with uniformly coercive $\bar{A}^\varepsilon$, the exact problem (2) can be linearized by means of the iteration $\int_\Omega \bar{A}^\varepsilon(x,\nabla u^{n-1})\nabla u^n \nabla \Phi = \int_\Omega f\Phi$, where u^n converges to u^ε under appropriate assumptions. Again, solving the HMM macro problem is uncomplicated, since we can use the linearized form above.

In the situation of a very general nonlinearity, we are facing problems which we described at the end of Section 3. Therefore, it is necessary to derive a method, which explicitly incorporates a general way for solving the macro problem (6). In the following we present an approach, where we combine the HMM with a Newton method. This procedure results in additional cell problems to solve.

First of all, we want to introduce a new representation of equation (6), which is more convenient with regard to applying the Newton method. For this purpose, we define the local correction operator Q_h^T.

Definition 0.3 (Correction operator). With the local reconstruction operator R_h^T defined in (3), we introduce a corresponding correction operator Q_h^T, which only yields (scaled) microscopic contributions. The operator $Q_h^T : V_H(\Omega) \to W_h(Y)$ is defined by $Q_h^T(\Phi_H)(y) := \frac{1}{\delta}\left(R_h^T(\Phi_H) - \Phi_H\right) \circ x_T^{\delta}(y)$ for $\Phi_H \in V_H(\Omega)$.

With this definition, we obtain the subsequent representation of equation (6):

Lemma 0.1 (Reformulation of the HMM operators). *For $T \in \mathscr{T}_H$, $\Phi_H \in V_H(\Omega)$ and Q_h^T given by Definition 0.3, we have that $Q_h^T(\Phi_H)$ is the solution of the following cell problem:*

$$\int_Y A^{\varepsilon}\left(x_T + \delta y, \nabla_x \Phi_H(x_T) + \nabla_y Q_h^T(\Phi_H)(y)\right) \cdot \nabla_y \phi_h(y)\, dy = 0 \quad \forall \phi_h \in W_h(Y). \tag{8}$$

Moreover, for $\Psi_H, \Phi_H \in V_H(\Omega)$, we can rewrite A_H (given by (7)) as:

$$\begin{aligned} A_H(\Psi_H, \Phi_H) = \sum_{T \in \mathscr{T}_H} |T| &\fint_{Y_{\frac{\varepsilon}{\delta}}} A^{\varepsilon}(x_T + \delta y, \Psi_H(x_T) + \nabla_y Q_h^T(\Psi_H)(y)) \\ &\cdot \left(\nabla_x \Phi_H(x_T) + \nabla_y Q_h^T(\Phi_H)(y)\right)\; dy. \end{aligned} \tag{9}$$

Proof. The results are directly obtained by using the transformation formula and the following equation, which holds for every $\Phi_H \in V_H$:

$$\begin{aligned} \nabla_y Q_h^T(\Phi_H)(y) &= \frac{1}{\delta}\nabla_y \left(R_h^T(\Phi_H)(x_T + \delta y) - \Phi_H(x_T + \delta y)\right) \\ &= \left(\nabla R_h^T(\Phi_H)\right)(x_T + \delta y) - \nabla \Phi_H(x_T). \end{aligned}$$ □

Now, we are prepared to derive the HM Newton algorithm. By $\{\Phi_i |\ 1 \leq i \leq M\}$ we denote the Lagrange base of the discrete space $V_H(\Omega)$. In particular M defines the number of macroscopic base functions. If we want to solve the HMM macro problem, i.e. find $u_H \in V_H$ with

$$A_H(u_H, \Phi_i) - (f, \Phi_i)_{L^2(\Omega)} = 0 \qquad \forall \Phi_i,\ 1 \leq i \leq M,$$

we can use the reformulation (9), so that we can equivalently look for $\bar{\alpha} \in \mathbb{R}^M$ with $G(\bar{\alpha}) = 0$. Here, $G : \mathbb{R}^M \to \mathbb{R}^M$ is defined by

$$
\begin{aligned}
(G(\alpha))_i := \sum_{T\in\mathscr{T}_H} |T| \fint_{Y_{\frac{\varepsilon}{\delta}}} & A^{\varepsilon}(x_T+\delta y, \sum_{j=1}^{M}\alpha_j\nabla_x\Phi_j(x_T)+\nabla_y Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)(y)) \\
& \cdot \big(\nabla_x\Phi_i(x_T)+\nabla_y Q_h^T(\Phi_i)(y)\big) \;\; dy - \int_{\Omega} f(x)\Phi_i(x)\, dx.
\end{aligned}
$$

If $\bar{\alpha}$ is computed, so is u_H, where we have the relation $u_H(x)=\sum_{j=1}^{N}\bar{\alpha}_j\Phi_j(x)$.

In order to solve the nonlinear algebraic equation $G(\bar{\alpha})=0$, we want to use the Newton method, with which we get the following iteration scheme:

$$
\alpha^{(n+1)} = \alpha^{(n)} - \Big((D_\alpha G)(\alpha^{(n)})\Big)^{-1} G(\alpha^{(n)}).
$$

Here, $D_\alpha G$ is the Jacobian matrix of G. With $\triangle\alpha^{(n)} := \alpha^{(n+1)}-\alpha^{(n)}$ we obtain:

$$
D_\alpha G(\alpha^{(n)})\triangle\alpha^{(n)} = -G(\alpha^{(n)}).
$$

This implies, that we need to compute the components of $D_\alpha G$. To determine $\frac{\mathrm{d}}{\mathrm{d}\alpha_k}(G(\alpha))_i$ for any $\alpha\in\mathbb{R}^M$, we calculate:

$$
\begin{aligned}
& D_{\alpha_k} A^{\varepsilon}(x_T+\delta y, \sum_{j=1}^{M}\alpha_j\nabla_x\Phi_j(x_T)+\nabla_y Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)(y)) \\
&= D_\xi A^{\varepsilon}(x_T+\delta y, \sum_{j=1}^{M}\alpha_j\nabla_x\Phi_j(x_T)+\nabla_y Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)(y)) \\
&\qquad (\nabla_x\Phi_k(x_T)+\nabla_y(D_{\alpha_k}(Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)))(y))
\end{aligned}
$$

It remains to determine $(D_{\alpha_k}(Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)))(y)$, which depends on the effect of the Fréchet derivative of the correction operator. Here, we need to use the reformulated cell problem (8) to obtain

$$
\begin{aligned}
0 &= D_{\alpha_k}\int_Y A^{\varepsilon}(x_T+\delta y, \sum_{j=1}^{M}\alpha_j\nabla_x\Phi_j(x_T)+\nabla_y Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)(y))\cdot\nabla_y\phi_h(y)\, dy \\
&= \int_Y (D_\xi A^{\varepsilon})(x_T+\delta y, \sum_{j=1}^{M}\alpha_j\nabla\Phi_j(x_T)+\nabla_y Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)(y)) \\
&\qquad (\nabla_x\Phi_k(x_T)+\nabla_y(D_{\alpha_k}(Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j)))(y))\cdot\nabla_y\phi_h(y)\, dy.
\end{aligned}
$$

Since this equation holds for every $\phi_h\in W_h(Y)$, we have a characterization of $D_{\alpha_k}(Q_h^T(\sum_{j=1}^{M}\alpha_j\Phi_j))$. Combining the results, we can compute $D_\alpha G(\alpha^{(n)})$ by first solving (8) to obtain $Q_h^T(u_H^{(n)})$ and using this to determine $D_{\alpha_k}(Q_h^T(u_H^{(n)}))$ afterwards.

The remaining strategy is straightforward: we assemble $G(\alpha^{(n)})$ and solve the linear system $D_\alpha G(\alpha^{(n)}) \triangle \alpha^{(n)} = -G(\alpha^{(n)})$. The subsequent HM Newton algorithm is a detailed summary of the whole procedure above.

Let $u_H^{(0)} \in V_H(\Omega)$ denote a suitable initial value for the HM Newton iterations. From now on, we use the following discretization $A_h^\varepsilon : \bigcup_{T \in \mathscr{T}_H} Y_{T,\delta} \times \mathbb{R}^N \to \mathbb{R}^N$ of the monotone elliptic diffusion operator A^ε. For $T \in \mathscr{T}_H,\ K \in \mathscr{T}_h$ and $\xi \in \mathbb{R}^N$ we define:

$$\begin{aligned} A_h^\varepsilon(\cdot,\xi)_{|x_T^\delta(K)} &:= A^\varepsilon(x_T^\delta(y_K),\xi) \text{ and} \\ (D_\xi A^\varepsilon)_h(\cdot,\xi)_{|x_T^\delta(K)} &:= (D_\xi A^\varepsilon)(x_T^\delta(y_K),\xi), \end{aligned}$$

where $D_\xi A^\varepsilon$ denotes Jacobian matrix of A^ε with respect to the second variable. Now, for a given tolerance $TOL > 0$, we present our algorithm for solving the nonlinear elliptic HMM macro problem (6). It is in Petrov Galerkin formulation, which means that we do not reconstruct the test functions to save computational demand.

Algorithm [HM Newton Method]
Let $u_H^{(n)} \in V_H(\Omega)$ name the solution of the last HM Newton iteration step.

1. For $T \in \mathscr{T}_H$, determine the local corrector $Q_h^T(u_H^{(n)}) \in W_h(Y)$ defined by

$$\int_Y A_h^\varepsilon\left(x_T + \delta y, \nabla_x u_H^{(n)}(x_T) + \nabla_y Q_h^T(u_H^{(n)})(y)\right) \cdot \nabla_y \phi_h(y)\, dy = 0,$$

for all $\phi_h \in W_h(Y)$. Determine $Q_h^T(u_H^{(n)})$ for any $T \in \mathscr{T}_H$.

2. For any $T \in \mathscr{T}_H$ and for any macroscopic base function Φ_i $(1 \leq i \leq M)$ with $(\text{supp}\, \Phi_i) \cap T \neq \emptyset$, determine $D_{Q_h^T}(\Phi_i, u_H^{(n)}) \in W_h(Y)$ defined by:

$$\begin{aligned} \int_Y\ & (D_\xi A^\varepsilon)_h \left(x_T + \delta y, \nabla_x u_H^{(n)}(x_T) + \nabla_y Q_h^T(u_H^{(n)})(x_T)\right) \\ & \left(\nabla_x \Phi_i(x_T) + \nabla_y D_{Q_h^T}(\Phi_i, u_H^{(n)})(y)\right) \cdot \nabla_y \phi_h(y)\, dy = 0, \end{aligned}$$

for all $\phi_h \in W_h(Y)$.

3. Define the entries of the HMM stiffness matrix $M^{(n)}$ by:

$$\begin{aligned} M_{ij}^{(n)} := \sum_{T \in \mathscr{T}_H} |T| \fint_{\frac{\varepsilon}{\delta} Y} & \Big((D_\xi A^\varepsilon)_h(x_T + \delta y, \nabla_x u_H^{(n)}(x_T) + \nabla_y Q_h^T(u_H^{(n)})(y)) \\ & (\nabla_x \Phi_j(x_T) + \nabla_y D_{Q_h^T}(\Phi_j, u_H^{(n)})(y))\Big) \cdot \nabla_x \Phi_i(x_T)\, dy \end{aligned}$$

and the entries of the right hand side by:

$$F_i^{(n)} := \int_\Omega f(x)\Phi_i(x)\,dx$$
$$- \sum_{T\in\mathscr{T}_H} |T| \fint_{\frac{\varepsilon}{\delta}Y} A_h^\varepsilon(x_T+\delta y, \nabla_x u_H^{(n)}(x_T) + \nabla_y Q_h^T(u_H^{(n)})(y)) \cdot \nabla_x \Phi_i(x_T)\,dy.$$

Now, find $(\triangle\alpha)^{(n+1)} \in \mathbb{R}^M$, with

$$M^{(n)}(\triangle\alpha)^{(n+1)} = F^{(n)}.$$

4. Define

$$u_H^{(n+1)} := u_H^{(n)} + \sum_{k=1}^M (\triangle\alpha)_k^{(n+1)} \Phi_k.$$

If ($\|(\triangle\alpha)^{(n+1)}\| <$ TOL) stop the algorithm ($u_H \approx u_H^{(n+1)}$),
else start again with step **1**, where $n \mapsto n+1$.

This algorithm is capable of treating any nonlinearity of the diffusion operator, which meets the assumption, stated in Section 2. In that way, we can say that it is an algorithm for the worst case scenario. Note, since we are in the setting of strictly monotone operators, we have convergence for any initial value. In our computations we chose $u_H^{(0)} = 0$.

4.2 *Realization Using* DUNE-FEM

Having a look at HM Newton algorithm, we see that it purely consists of solving a large number of linear elliptic problems. Due to limiting memory capacities, these data is stored in files for later use. Since DUNE-FEM (see [4]) provides us with a lot of tools to handle standard linear elliptic equations, the implementation becomes rather easy. We need two different discrete function space objects of the `LagrangeDiscreteFunctionSpace`-type, with a template argument describing the specific grid partition. For the macroscopic space, we use the `AdaptiveLeafGridPart`-object and for the microscopic space with periodic boundary condition, we use the `PeriodicLeafGridPart`-object. The stiffness matrix in our linear system of equations is of the `SparseRowMatrixOperator`-type. In order to assemble the stiffness matrix, we implemented and constructed a `DiscreteEllipticOperator`-object, which fits into the general concept of DUNE-FEM and that simply adds values to the entries of an initially empty sparse matrix. The `DiscreteEllipticOperator`-class requires a template argument, which describes the realization of the diffusion operator. Here we used a `DiffusionOperator`-object inheriting from the existing `FunctionSpace`-class. The `DiffusionOperator` that we implemented provides two methods: the `diffusiveFlux` evaluates the diffusion in a certain point x and in a certain direction ξ and returns the current flux afterwards (i.e. $flux = A^\varepsilon(x,\xi)$).

On the other hand, the `jacobianDiffusiveFlux`-method evaluates the Jacobian matrix of the diffusion operator with respect to the second variable (i.e. return $flux = D_{\xi} A^{\varepsilon}(x,\xi)\, direction$). By means of these methods we assemble the stiffness matrix and the right hand side in our non-symmetric algebraic system of equations. For solving it, we make use of the `OEMBICGSQOp`-class, which provides us with a Bi-CG squared method. The `OEMBICGSQOp`-class takes two template arguments: our discrete function space class and our realization of the `SparseRowMatrixOperator`-class. An overview on the essential classes for solving a cell problem might look like this:

```
using namespace Dune;
typedef FunctionSpace< double , double , DIM , POLORDER > FuncSpace;
typedef LagrangeDiscreteFunctionSpace< FuncSpace, PeriodicGridPart, 1 > DiscFuncSpace;
typedef AdaptiveDiscreteFunction< DiscFuncSpace > DiscFunction;
typedef SparseRowMatrixOperator< DiscFunction, DiscFunction, MatrixTraits > FEMMatrix;
typedef OEMBICGSQOp< DiscFunction, FEMMatrix > InverseFEMMatrix;
typedef DiffusionOperator< FuncSpace > Diffusion;
typedef DiscreteEllipticOperator< DiscFunction, Diffusion > DiscreteEllipticOperator;
```

Finally, we assemble and solve all the required cell problems in the order in which they are stated in the algorithm. For instance:

```
DiscFuncSpace discFuncSpace( periodicGridPart );
Diffusion A;
DiscreteEllipticOperator discreteEllipticOp( discFuncSpace, A );
DiscFunction solutionCellProblem("solution␣cell␣problem", discFuncSpace );
DiscFunction rhs("right␣hand␣side␣cell␣problem", discFuncSpace );
FEMMatrix stiffnessMatrix("FEM␣stiffness␣matrix", discFuncSpace, discFuncSpace );
discreteEllipticOp.assembleMatrix( stiffnessMatrix );
InverseFEMMatrix biCGStab( stiffnessMatrix, 1e-8, 1e-8, 20000, VERBOSE );
rhsassembler.assemble < QUADORDER >( G , rhs);
biCGStab( rhs, solutionCellProblem );
```

Saving the results to a file and using them to assemble the final macroscopic system of equations ends the required code.

5 Numerical Experiment

In the following we present the results of a numerical experiment, which we used to validate our HM Newton algorithm and the corresponding implementation. We are dealing with a nonlinear model problem, where the diffusion operator A^{ε} is periodically oscillating with period $\varepsilon = 0.05$. In this scenario we are able to determine the corresponding homogenized solution u_0, as well as we can perform an expensive fine scale computation to determine the exact solution u^{ε}. The equation is:

Problem 0.2 (Nonlinear elliptic model equation). Find $u^{\varepsilon} \in \mathring{H}^1([0,2]^2)$ with:

$$\begin{aligned} -\nabla \cdot A^{\varepsilon}(x, \nabla u^{\varepsilon}) &= 1 \;\text{ in }\; [0,2]^2, \\ u^{\varepsilon} &= 0 \;\text{ on }\; \partial[0,2]^2. \end{aligned}$$

The nonlinear diffusion operator A^{ε} is given by:

$$A^{\varepsilon}(x,\xi) = \begin{pmatrix} (0.1 + \cos(2\pi \frac{x_1}{\varepsilon})^2) \cdot (\xi_1 + \frac{1}{3}\xi_1^3) \\ (0.101 + (0.1\sin(2\pi \frac{x_2}{\varepsilon}))) \cdot (\xi_2 + \frac{1}{3}\xi_2^3) \end{pmatrix}.$$

Table 1 Evaluation of the error between HMM-approximation u_H and homogenized solution u_0. We can see a second order convergence of u_H to u_0 in the L^2-norm.

H	h	$\|u_H - u_0\|_{L^2(\Omega)}$
2^{-1}	2^{-2}	$2.62 \cdot 10^{-1}$
2^{-2}	2^{-3}	$8.06 \cdot 10^{-2}$
2^{-3}	2^{-4}	$2.34 \cdot 10^{-2}$
2^{-4}	2^{-5}	$5.47 \cdot 10^{-3}$
2^{-5}	2^{-6}	$9.91 \cdot 10^{-4}$

$(H,h) \to (\frac{H}{2}, \frac{h}{2})$	EOC(e_H)
$(2^{-1}, 2^{-2}) \to (2^{-2}, 2^{-3})$	1.7018
$(2^{-2}, 2^{-3}) \to (2^{-3}, 2^{-4})$	1.7864
$(2^{-3}, 2^{-4}) \to (2^{-4}, 2^{-5})$	2.0941
$(2^{-4}, 2^{-5}) \to (2^{-5}, 2^{-6})$	2.4645

Table 2 Evaluation of the error between HMM-approximation u_H and exact solution u^ε. We observe convergence, but only up to an accuracy of order ε, which is due to $\|u_0 - u^\varepsilon\|_{L^2(\Omega)} = \mathrm{O}(\varepsilon)$.

H	h	$\|u_H - u^\varepsilon\|_{L^2(\Omega)}$
2^{-1}	2^{-2}	$2.62 \cdot 10^{-1}$
2^{-2}	2^{-3}	$8.14 \cdot 10^{-2}$
2^{-3}	2^{-4}	$2.74 \cdot 10^{-2}$
2^{-4}	2^{-5}	$1.61 \cdot 10^{-2}$
2^{-5}	2^{-6}	$1.55 \cdot 10^{-2}$

$(H,h) \to (\frac{H}{2}, \frac{h}{2})$	EOC(e_H^ε)
$(2^{-1}, 2^{-2}) \to (2^{-2}, 2^{-3})$	1.6868
$(2^{-2}, 2^{-3}) \to (2^{-3}, 2^{-4})$	1.5697
$(2^{-3}, 2^{-4}) \to (2^{-4}, 2^{-5})$	0.7625
$(2^{-4}, 2^{-5}) \to (2^{-5}, 2^{-6})$	0.0645

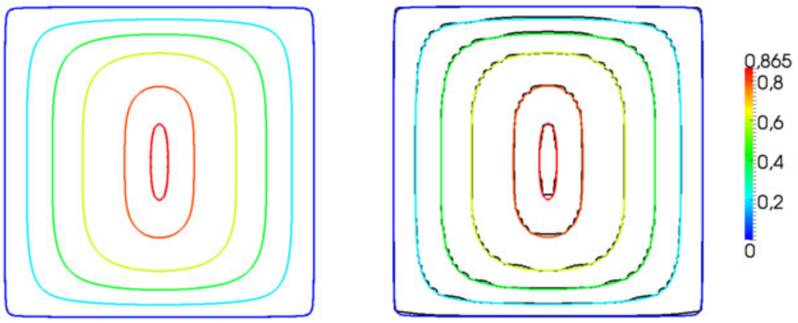

Fig. 1 Left figure: comparison between the isolines of HMM approximation u_H and homogenized solution u_0. There is no difference perceptible. Right figure: comparison between the isolines of u_H and solution u^ε. We can see only small difference, resulting from the fine scale oscillations of u^ε.

Below, H denotes the macro mesh size, given by $H :=\sup\{\operatorname{diam}(T)|T \in \mathscr{T}_H\}$ and h denotes the micro mesh size, defined by $h :=\sup\{\operatorname{diam}(K)|K \in \mathscr{T}_h\}$. Moreover, the experimental order of convergence (EOC) for two errors e_H and $e_{\frac{H}{2}}$ (i.e. for $(H,h) \rightarrow (\frac{H}{2},\frac{h}{2})$) is defined by the ratio

$$\frac{\log\left(\frac{\|e_H\|_{L^2(\Omega)}}{\|e_{\frac{H}{2}}\|_{L^2(\Omega)}}\right)}{\log(2)}.$$

Since heterogeneous multiscale finite elements methods are constructed to approximate the coarse scale (or homogenized) part u_0 of the exact solution u^ε, we start with evaluating the error $e_H := \|u_0 - u_H\|_{L^2(\Omega)}$. The results are shown in Table 1, where we can see that u_H converges to u_0 with second order. This is just the behavior that we expected. If we have a look at the error $e^\varepsilon_H := \|u^\varepsilon - u_H\|_{L^2(\Omega)}$ we only expect convergence up to a certain accuracy. This is reasonable, if we consider that u^ε contains fine scale oscillations, which can not be captured by the coarse function u_H. Analytically, it is well known that $\|u^\varepsilon - u_0\|_{L^2(\Omega)} = \mathrm{O}(\varepsilon)$. Again, having a look at Table 2, this prediction can be confirmed. However, since ε is very small, u_H can be used as an accurate approximation of u^ε itself. That is also what we see, by looking at Figure 1. The isolines of HMM approximation and homogenized solution are almost identical, whereas the isolines of u_H and u^ε only differ in microscale contributions. We conclude that the HM Newton algorithm is an accurate and effective tool to determine the HMM approximation u_H, given by equation (6).

Acknowledgements. This work was supported by the Deutsche Forschungsgemeinschaft under the contract number OH 98/4-1.

References

1. Abdulle, A.: The finite element heterogeneous multiscale method: a computational strategy for multiscale PDEs. In: Multiple Scales Problems in Biomathematics, Mechanics, Physics and Numerics. GAKUTO Internat. Ser. Math. Sci. Appl., Tokyo, vol. 31, pp. 133–181 (2009)
2. Abdulle, A., Engquist, B.: Finite element heterogeneous multiscale methods with near optimal computational complexity. Multiscale Model. Simul. 6(4), 1059–1084 (2007)
3. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Kornhuber, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and tests in DUNE. Computing 82, 121–138 (2008)
4. Dedner, A., Klöfkorn, R., Nolte, M., Ohlberger, M.: A generic interface for parallel and adaptive discretization schemes: abstraction principles and the DUNE-FEM module. In: Computing, pp. 1–32. Springer Wien (2010)
5. Weinan, E., Engquist, B.: The heterogeneous multiscale methods. Commun. Math. Sci. 1(1), 87–132 (2003)
6. Efendiev, Y., Hou, T., Ginting, V.: Multiscale finite element methods for nonlinear problems and their applications. Commun. Math. Sci. 2(4), 553–589 (2004)

7. Henning, P., Ohlberger, M.: The heterogeneous multiscale finite element method for elliptic homogenization problems in perforated domains. Numer. Math. 113(4), 601–629 (2009)
8. Henning, P., Ohlberger, M.: A-posteriori error estimate for a heterogeneous multiscale finite element method for advection-diffusion problems with rapidly oscillating coefficients and large expected drift. Preprint Uni. Münster N09/09
9. Henning, P., Ohlberger, M.: The heterogeneous multiscale finite element method for advection-diffusion problems with rapidly oscillating coefficients and large expected drift. Netw. Heterog. Media 5(4) (2010)
10. Ladyzhenskaya, O.A., Uraltseva, N.N.: Linear and quasilinear elliptic equations. Academic Press, New York (1968)
11. Ohlberger, M.: A posteriori error estimates for the heterogeneous multiscale finite element method for elliptic homoenization problems. Multiscale Model. Simul. 4(1), 88–114 (2005)
12. Wall, P.: Some homogenization and corrector results for nonlinear monotone operators. J. Nonlinear Math. Phys. 5(3), 331–348 (1998)