

The heterogeneous multiscale method*

Assyr Abdulle

*ANMC, Mathematics Section,
École Polytechnique Fédérale de Lausanne,
Lausanne, Switzerland
E-mail: assyr.abdulle@epfl.ch*

Weinan E

*Beijing International Center for Mathematical Research,
Peking University,
Beijing, China
and
Department of Mathematics and PACM,
Princeton University,
Princeton, USA
E-mail: weinan@math.princeton.edu*

Björn Engquist

*Department of Mathematics,
University of Texas,
Austin, USA
E-mail: engquist@ices.utexas.edu*

Eric Vanden-Eijnden

*Courant Institute of Mathematical Sciences,
New York University,
New York, USA
E-mail: eve2@cims.nyu.edu*

The heterogeneous multiscale method (HMM), a general framework for designing multiscale algorithms, is reviewed. Emphasis is given to the error analysis that comes naturally with the framework. Examples of finite element and finite difference HMM are presented. Applications to dynamical systems and stochastic simulation algorithms with multiple time scales, spall fracture and heat conduction in microprocessors are discussed.

* Colour online for monochrome figures available at journals.cambridge.org/anu.

CONTENTS

1	Introduction	2
2	The HMM framework	4
3	ODEs and dynamical systems	16
4	Finite element HMM	37
5	Finite volume methods	67
6	Conclusion	78
	References	80

1. Introduction

The heterogeneous multiscale method (HMM) is a general framework for designing multiscale algorithms for a wide variety of applications (E and Engquist 2002, E and Engquist 2003, E *et al.* 2007*a*). The word ‘heterogeneous’ was used in order to emphasize the fact that the algorithm may involve macro and micro models of very different natures: for example, the micro model may come from molecular dynamics and the macro model may come from continuum theory. In fact, at a very rough level, HMM can be thought of as a way of blending together models of very heterogeneous types.

Most problems that we encounter in nature have a multiscale character. The multiscale character can occur in a variety of ways. Take, for example, problems from materials science, where many properties, such as conductivity, have a multiscale nature. This is the case for composites. It could also be that the material can be viewed at different levels of detail: as a continuous medium, in which case one naturally applies the principles of continuum mechanics, or at the atomic scale, in which case one naturally applies various atomistic models of molecular dynamics or quantum mechanics. Each viewpoint has its merits and drawbacks. Continuum models are quite efficient but sometimes their accuracy is inadequate, particularly when defects are involved. Atomic models are typically more accurate, but much less efficient. This situation is not limited to materials science but is quite common in most areas of science and engineering. One of the main motivations for multiscale modelling is to develop models that have accuracy close to that of microscopic models and efficiency close to that of macroscopic models.

From the viewpoint of numerical algorithms, we are interested in extracting useful information from the microscopic model, which in principle has the required accuracy. If we use the traditional viewpoint, then we would have to solve the microscopic model in full detail, which is practically impossible for engineering applications. In terms of computational complexity, the best one can do with such an approach is to have linear scaling algorithms: the complexity scales as the number of microscopic degrees of freedom. However, in many cases, we are not interested in the full microscopic solution or

we cannot afford the cost of computing it. Instead, we are only interested in the behaviour of some macroscopic variables, or the microscopic behaviour in very small parts of the system, for example near defects. The question is: Can we develop much more efficient algorithms, such as *sublinear* scaling algorithms, that would give us such information? To develop these new types of algorithms, we have to compress not only the operators, as has been done in multigrid methods, but also the variables. We have to be content with getting information about only a subset of the system variables. These types of algorithms cannot be completely general: one has to explore special features of the problem in order to construct such algorithms.

From the viewpoint of analysis, many analytical techniques have been developed in order to derive simplified models. Examples include averaging methods, homogenization methods, matched asymptotics, WKB methods, geometric optics approximations, and renormalization group methods (E 2011). The principles of such techniques are quite general, but in practice they only give us explicit analytical models in very limited situations. In other situations, it is natural to ask whether one can devise efficient computational techniques based on these principles. This is the case that we are interested in, and it was one of the main motivations for developing HMM.

This was the background against which HMM was proposed. Of course, prior to HMM, there were already many techniques of a similar spirit in many different fields. Well-known examples include:

- Car–Parrinello molecular dynamics, in which electronic structure models are used together with molecular dynamics to predict the dynamics of nuclei (Car and Parrinello 1985),
- the quasicontinuum method, in which atomistic models are used to analyse the mechanical deformation of crystalline solids (Tadmor, Ortiz and Phillips 1996),
- superparametrization models, in which cloud-resolving models are used to capture large-scale tropical dynamics of the atmosphere (Grabowski 2001, Xing, Grabowski and Majda 2009).

HMM was proposed as a general framework that can be used for a variety of problems. It was not the only attempt. Other notable examples include the extended multigrid method and the equation-free approach (Brandt 2002, Kevrekidis *et al.* 2003). A common theme of these approaches is that the microscopic models are used throughout the computational process. These should be contrasted with techniques such as model reduction methods, wavelet-based homogenization and variational multiscale methods, in which the microscale model is only used at the beginning of the computation to obtain compressed effective operators.

In spite of competing efforts, HMM was the only general attempt based on a top-down philosophy, which at the time was not the most popular

viewpoint. In fact, in the early days of multiscale modelling, most efforts were devoted to a bottom-up approach, seeking strategies that would give us the information needed by working only with the microscale model, without any prior information about the system at the macroscale. This certainly sounds very attractive, and may at first sight seem the only correct approach. In one way, a key insight of HMM was the recognition that the bottom-up approach is bound to have technical difficulties, and will for some time be limited to rather simple applications. One can appreciate such difficulties by noticing the fact that, even if the effective macroscale model is explicitly available, designing stable and accurate numerical algorithms for such macroscale models is still a non-trivial task. Important constraints, such as conservation properties or upwinding, have to be implemented in order to guarantee that the algorithms give rise to the correct numerical solutions. Implementing such constraints at the level of microscale models, in the absence of any explicit knowledge about the macroscale model, seems to be next to impossible. Therefore compromises are necessary: for many problems we have to guess the form of the macroscale model to start from. Fortunately, in many cases we do have some prior knowledge of the macroscale behaviour of the system under consideration, and this knowledge is often sufficient for us to make an adequate guess.

Since multiscale modelling is a vast subject, touching almost all aspects of modelling, we will not be able to do justice to all the work that has been done on this subject. Instead we will focus on HMM. For a general introduction to multiscale modelling, we refer to the monograph by E (2011).

2. The HMM framework

2.1. The main components of HMM

We will use U to describe the set of macroscopic variables, and u the set of microscopic variables. They are related by

$$U = Q(u), \quad (2.1)$$

where Q is called the compression operator. Any operator that reconstructs u from U is called a reconstruction operator:

$$u = R(U). \quad (2.2)$$

For consistency, Q and R should satisfy the relation

$$Q(R(U)) = U. \quad (2.3)$$

In HMM, we assume that we have an incomplete macroscale model to begin with:

$$F(U, D) = 0. \quad (2.4)$$

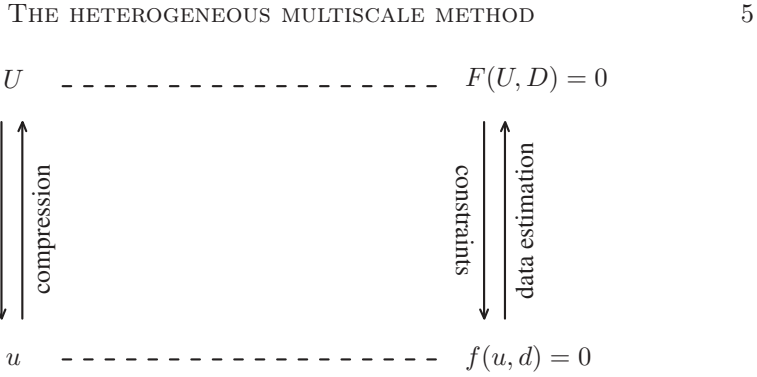


Figure 2.1. Schematics of the HMM framework.

Here D represents the missing part of the model. For example, if this is a model in continuum mechanics, then D might be the constitutive relation for the stress. If it is a model for molecular dynamics, then D might be the inter-atomic forces. If it is a model for heat conduction in composite materials, then D might be the macroscale effective conductivity tensor.

HMM proceeds by estimating the missing data on the fly using the microscale model, at each location where some missing data is needed. To do this, the microscale model has to be constrained so that its macrostate is the same as the macrostate we are interested in, that is,

$$f(u, d(U)) = 0. \tag{2.5}$$

Here $d(U)$ represents the constraint for the microscale model. For example, if the microscale model is the canonical ensemble of molecular dynamics, d might be the average density, momentum and energy.

If we use H and h to denote the macro and micro numerical parameters, such as mesh size, one can write HMM abstractly in the following form:

$$\begin{aligned} F_H(U_H, D_H(u_h)) &= 0, \\ f_h(u_h, d_h(U_H)) &= 0. \end{aligned} \tag{2.6}$$

In practical terms, the basic components of HMM are as follows.

- 1 *A macroscopic solver.* Based on knowledge of the macroscale behaviour of the system, we make an assumption about the form of the macroscale model, for which we select a suitable macroscale solver. For example, if we are dealing with a variational problem, we may use a finite element method as the macroscale solver.
- 2 *A procedure for estimating the missing macroscale data D using the microscale model.* This is typically done in two steps.
 - (a) *Constrained microscale simulation.* At each point where macroscale data are needed, perform a series of microscopic simulations which are constrained so that they are consistent with the local value of the macro variable.

6 A. ABDULLE, WEINAN E, B. ENGQUIST AND E. VANDEN-EIJNDEN

- (b) *Data processing.* Use the results from the microscopic simulations to extract the macroscale data needed in the macroscale solver.

For dynamical problems, we can state the HMM procedure formally as follows. At each macro time step:

- 1 Given the current state of the macro variables U^n , re-initialize the micro variables:

$$u^{n,0} = RU^n. \quad (2.7)$$

- 2 Evolve the micro variables for some micro time steps:

$$u^{n,m+1} = \mathcal{S}_{\delta t}(u^{n,m}; U^n), \quad m = 0, \dots, M-1. \quad (2.8)$$

- 3 Estimate D :

$$D^n = \mathcal{D}_M(u^{n,0}, u^{n,1}, \dots, u^{n,M}). \quad (2.9)$$

- 4 Evolve the macro variables for one macro time step using the macro solver:

$$U^{n+1} = S_{\Delta t}(U^n; D^n). \quad (2.10)$$

Here R is some reconstruction operator which plays the same role as the interpolation or prolongation operators in the multigrid method; $\mathcal{S}_{\delta t}$ is the micro solver, which also depends on U^n through the constraints, as indicated; and \mathcal{D}_M is some data processing operator, which in general involves spatial/temporal/ensemble averaging. This is sometimes referred to as the data estimator. Finally, $S_{\Delta t}$ is the macro solver.

For static problems, the procedure is very similar, particularly in the context of iterative algorithms: we simply replace macro time step by macro iteration step.

For dynamic problems, there are two important time scales that we need to consider. The first, denoted by t_M , is the time scale for the dynamics of the macro variables. The second, denoted by τ_ε , is the relaxation time for the microscopic model. We will need to distinguish between two different cases. The first is when the two time scales are comparable, that is, $\tau_\varepsilon \sim t_M$. In this case, from the viewpoint of numerical efficiency, there is not much room to play with as far as time scales are concerned. We just have to evolve the microscale model along with the macroscale model. The second case is when $\tau_\varepsilon \ll t_M$. This is the case we will focus on. The general guideline in this case is as follows.

- 1 Choose Δt to accurately resolve the t_M time scale.
- 2 Choose M so that $M\delta t$ covers the τ_ε time scale sufficiently to allow equilibration to take place in the micro model.

2.2. Simple examples

Examples of this set-up include the following.

- ODEs with disparate time scales, where U is a complete set of slow variables and u is the full set of variables. In this case, the macroscale model could just be an ODE system for U , and the missing data could just be the force. If we know that the macroscale system has additional structure, then we can take that into account when selecting the macro solver. For example, if the macroscale system is a Hamiltonian system, then we can use a symplectic integrator as the macro solver.
- Elliptic equations with multiscale coefficients, such as those arising in the modelling of the behaviour of composite materials, where U is the averaged displacement field and u is the microscale displacement field. In this case, the macroscale model is still an elliptic equation, and the missing data could be the coefficients in the macroscale model. If we use the finite element method as the macro solver, then the missing data could just be the stiffness matrix, which has to be evaluated with the help of the full microscale model.
- Molecular dynamics models of complex fluids such as polymer fluids. Here U is the set of hydrodynamic variables, which in the simplest case would be the field of mass, momentum and energy densities, and u is the set of microscopic variables, that is, the position and momentum of all of the participating particles in a molecular dynamics model. The macroscale model might be the set of local conservation laws for U . The missing data might be the fluxes in these conservation laws.
- Microscopic pore-scale models of the pressure distribution in a porous medium. Here U is the macroscale pressure distribution and u is the set of variables in the pore-scale model, which could be a network-based model. The macroscale model is of Darcy law type. The missing data are the coefficients in this model.
- Microscale models of the moving contact line. Here U is the set of macroscopic variables (velocity and pressure fields, position of the interface between the fluid phases), and u is the set of variables in the microscopic description, say using molecular dynamics, of the contact line region. The macroscale model might just be the standard model in two-phase flows. The missing data might be the boundary conditions at the contact line.

Example 1: Stiff ODEs. Consider

$$\begin{aligned}\frac{dx}{dt} &= f(x, y), \\ \frac{dy}{dt} &= -\frac{1}{\varepsilon}(y - \varphi(x)).\end{aligned}\tag{2.11}$$

Here $U = x, u = (x, y)$. The macroscale should of course be an ODE, which can be written as

$$\frac{dx}{dt} = F(x) = f(x, \varphi(x)). \tag{2.12}$$

The missing data are implicitly given by F .

Let us choose the simplest solver for (2.12), the forward Euler method. HMM would then proceed as follows.

- 1 Initialize the micro solver, *e.g.*, $y^{n,0} = y^{n-1,M}$.
- 2 Apply the micro solver for M micro steps:

$$y^{n,m+1} = y^{n,m} - \frac{\delta t}{\varepsilon}(y^{n,m} - \varphi(x^n)), \tag{2.13}$$

for $m = 0, 1, \dots, M - 1$.

- 3 Estimate $F(x)$:

$$F^n = f(x^n, y^{n,m}). \tag{2.14}$$

- 4 Apply the macro solver:

$$x^{n+1} = x^n + \Delta t F^n. \tag{2.15}$$

Example 2: Stiff stochastic ODEs. Consider the stochastic ODE

$$\begin{aligned} \frac{dx}{dt} &= f(x, y), \\ \frac{dy}{dt} &= -\frac{1}{\varepsilon}(y - \varphi(x)) + \sqrt{\frac{2}{\varepsilon}}\dot{w}, \end{aligned} \tag{2.16}$$

where $\dot{w}(t)$ is standard white noise. Averaging theorems suggest that the effective macroscale equation should again be an ODE:

$$\frac{dx}{dt} = F(x). \tag{2.17}$$

HMM with forward Euler as the macro solver proceeds as follows.

- 1 Initialize the micro solver, *e.g.*, $y^{n,0} = y^{n-1,M}$.
- 2 Apply the micro solver for M micro steps:

$$y^{n,m+1} = y^{n,m} - \frac{\delta t}{\varepsilon}(y^{n,m} - \varphi(x^n)) + \sqrt{\frac{2\delta t}{\varepsilon}}\xi^{n,m}, \tag{2.18}$$

for $m = 0, 1, \dots, M - 1$. Here the $\{\xi^{n,m}\}$ are independent normal random variables with mean 0 and variance 1.

- 3 Estimate $F(x)$:

$$F^n = \frac{1}{M} \sum_{m=1}^M f(x^n, y^{n,m}). \tag{2.19}$$

4 Apply the macro solver:

$$x^{n+1} = x^n + \Delta t F^n. \quad (2.20)$$

Example 3: Elliptic PDEs with multiscale coefficients. Consider

$$-\nabla \cdot (a^\varepsilon(x) \nabla) u^\varepsilon(x) = f(x). \quad (2.21)$$

Abstract homogenization theory tells us that the macroscale model should be of the form

$$-\nabla \cdot (a^0(x) \nabla) U(x) = f(x). \quad (2.22)$$

The missing data are the coefficients $a^0(x)$. Naturally, for the macroscale solver we choose standard finite element methods, for example the piecewise linear finite element method over a coarse mesh. The data that need to be estimated form the stiffness matrix for the finite element method. If $a^0 = a^0(x)$ were known, we would simply follow standard practice and use numerical quadrature to compute the elements in the stiffness matrix. Since a^0 is not known, we set up a microscale simulation around each quadrature point in order to estimate the function value needed at that quadrature point. The details of this procedure will be discussed later.

Example 4: The parabolic homogenization problem. Consider

$$\partial_t u^\varepsilon = \partial_x \cdot \left(a \left(x, \frac{x}{\varepsilon}, t \right) \partial_x u^\varepsilon \right), \quad (2.23)$$

where $a(x, y, t)$ is a smooth function and is periodic in y , say with period 1. The macroscale model is of the form

$$\partial_t U = \partial_x \cdot D, \quad (2.24)$$

$$D = \left\langle a \left(x, \frac{x}{\varepsilon}, t \right) \partial_x u^\varepsilon \right\rangle, \quad (2.25)$$

where $\langle \cdot \rangle$ means taking spatial averages.

We will choose a finite volume method as the macro solver. Then D needs to be evaluated at the cell boundaries (Abdulle and E 2003). We will make the assumption that the flux D depends on the local values of U and $\partial_x U$ only. Consequently, for the micro model, we will impose the boundary condition that $u^\varepsilon(x, t) - Ax$ is periodic where $A = \partial_x U$ is evaluated at the location of interest.

Denote the micro solver by

$$u^{n+1} = \mathcal{S}_{\delta\tau, \delta x}(u^n; A). \quad (2.26)$$

Assuming that we have the numerical approximation $\{U_j^n\}$ (where $t^n = n\Delta t, U_j^n \sim U(n\Delta t, j\Delta x)$) at the n th macro time step, we obtain the numerical approximation at the next macro time step via the following procedure.

10 A. ABDULLE, WEINAN E, B. ENGQUIST AND E. VANDEN-EIJNDEN

- 1 For each j , let $A_j^n = (U_j^n - U_{j-1}^n)/\Delta x$.
- 2 Re-initialize the micro solver, so $u_j^0(x) - A_j^n x$ is periodic for each j .
- 3 Apply the micro solver M steps:

$$u_j^{n,m+1} = \mathcal{S}_{\delta\tau, \delta x}(u_j^{n,m}; A_j^n),$$

with $m = 0, 1, \dots, M - 1$.

- 4 Compute

$$D_{j-1/2}^{n+1} = \left\langle a\left(x, \frac{x}{\varepsilon}, t^n\right) \partial_x u_j^{n,M} \right\rangle. \quad (2.27)$$

- 5 Evolve the macrostate variables using

$$U_j^{n+1} = U_j^n + \Delta t \frac{D_{j+1/2}^{n+1} - D_{j-1/2}^{n+1}}{\Delta x}. \quad (2.28)$$

Example 5: Incompressible polymeric fluid flow. Let U be the macroscale velocity field. The macroscale model should be of the form

$$\begin{aligned} \rho_0(\partial_t U + (U \cdot \nabla)U) &= \nabla \cdot \sigma, \\ \nabla \cdot U &= 0. \end{aligned}$$

These are simply statements of the conservation of momentum and mass, for a fluid of constant density ρ_0 . The missing data comprise the stress field σ : $D = \sigma$.

Let us assume that the micro model is a molecular dynamics model for the particles that make up the fluid:

$$m_j \frac{d^2 y_j}{dt^2} = f_j, \quad j = 1, 2, \dots, N. \quad (2.29)$$

Here m_j, y_j are, respectively, the mass and position of the j th particle and f_j is the force acting on the j th particle, and u is the set of variables in this model.

Given that the macroscale model is in the form of an incompressible flow equation, it is natural to select the projection method as the macro solver (Chorin 1967). In the implementation of the projection method, we will need the values of σ at appropriate grid points. These are the data that need to be estimated. At this point, we have to make an assumption on the functional dependence of σ ; this enters in the constraints applied to the microscale model. Let us assume that

$$\sigma = \sigma(\nabla U). \quad (2.30)$$

We will constrain the molecular dynamics in such a way that the average strain rate is given by the value of ∇U at the relevant grid point. In general,