Acta Numerica (2012), pp. 1–87 © Cambridge University Press, 2012
doi:10.1017/S09624929XXXXXXXX Printed in the United Kingdom

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

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

$$U = Q(u), \tag{2.1}$$

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

$$u = R(U). \tag{2.2}$$

For consistency, Q and R should satisfy the relation

$$Q(R(U)) = U. \tag{2.3}$$

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

$$F(U, D) = 0. \tag{2.4}$$

$$\begin{tikzcd} U \arrow[rrr, dash, dashed] \arrow[d, shift right, "\text{reconstruction}"'] & & & F(U,D)=0 \arrow[d, shift right, "\text{constraints}"'] \\ u \arrow[rrr, dash, dashed] \arrow[u, shift right, "\text{compression}"'] & & & f(u,d)=0 \arrow[u, shift right, "\text{data estimation}"'] \end{tikzcd}$$

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.

(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:

$$u^{n,0} = RU^n. \tag{2.7}$$

2 Evolve the micro variables for some micro time steps:

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

3 Estimate D:

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

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

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

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

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= f(x,y),\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= -\frac{1}{\varepsilon}(y-\varphi(x)).\end{aligned} \tag{2.11}$$

please check (2.12)

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

$$\frac{\mathrm{d}x}{\mathrm{d}t} = F(x) = f(x, \varphi(x)). \tag{2.12}$$

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 $m = 0, 1, \ldots, M - 1$.

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

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

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

$$\frac{\mathrm{d}x}{\mathrm{d}t} = F(x). \tag{2.17}$$

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 $m = 0, 1, \ldots, M-1$. 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:

$$x^{n+1} = x^n + \Delta t\, F^n. \tag{2.20}$$

Example 3: Elliptic PDEs with multiscale coefficients. Consider

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

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

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

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

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

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

$$\partial_t U = \partial_x \cdot D, \tag{2.24}$$

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

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

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

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 nth macro time step, we obtain the numerical approximation at the next macro time step via the following procedure.

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 $m = 0, 1, \ldots, M-1$.

4 Compute

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

5 Evolve the macrostate variables using

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

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:

$$m_j \frac{\mathrm{d}^2 y_j}{\mathrm{d}t^2} = f_j, \quad j = 1, 2, \ldots, N. \tag{2.29}$$

Here m_j, y_j are, respectively, the mass and position of the jth particle and f_j is the force acting on the jth 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

$$\sigma = \sigma(\nabla U). \tag{2.30}$$

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, implementing such constraints is the most difficult step in HMM. For the present example, one possible strategy is discussed in Ren and E (2005).

From the results of molecular dynamics, we need to extract the values of the required components of the stress. For this purpose, we need a formula that expresses stress in terms of the output of the molecular dynamics. This can be obtained by modifying the Irving–Kirkwood formula (Irving and Kirkwood 1950). These details will be explained in Section 5. We refer to Ren and E (2005) for some numerical results obtained using this strategy.

2.3. Error analysis

The basic idea, as explained by E and Engquist (2003), is to compare the HMM solution with the solution of the selected macroscale solver for the effective macroscale model. Their discrepancy is caused by an additional error in the HMM solution, due to the error in the data estimation process. This new error term is called the HMM error, denoted by $e(\mathrm{HMM})$. We will assume that both the HMM and the macro solver for the effective macroscale model can be expressed in the form

$$U^{n+1}_{\mathrm{HMM}} = U^n_{\mathrm{HMM}} + \Delta t F^\varepsilon(U^n_{\mathrm{HMM}}, U^{n-1}_{\mathrm{HMM}}, \ldots), \tag{2.31}$$

$$\bar{U}^{n+1}_H = \bar{U}^n_H + \Delta t \bar{F}(\bar{U}^n_H, \bar{U}^{n-1}_H, \ldots). \tag{2.32}$$

Note that

$$\|Qu^\varepsilon - U_{\mathrm{HMM}}\| \le \|Qu^\varepsilon - \bar{U}\| + \|\bar{U}_H - \bar{U}\| + \|U_{\mathrm{HMM}} - \bar{U}_H\|, \tag{2.33}$$

where $\bar{U}$ is the solution of the macroscale model, $\bar{U}_H$ is the numerical solution to the effective macroscale model computed using (2.31), and U_{HMM} is the HMM solution. The first term on the right-hand side of (2.33) is due to the error of the effective model; the second term is due to the error in the macroscale solver; the third term is the HMM error, due to the error in the estimated data. Normally we expect that estimates of the following type hold:

$$\|Qu^\varepsilon - \bar{U}\| \le C\varepsilon^\alpha, \tag{2.34}$$

$$\|\bar{U} - \bar{U}_H\| \le C(\Delta t)^k, \tag{2.35}$$

where k is the order of accuracy of the macro solver. In addition, define

$$e(\mathrm{HMM}) = \max_U \|\bar{F}(U^n, U^{n-1}, \ldots) - F^\varepsilon(U^n, U^{n-1}, \ldots)\|. \tag{2.36}$$

Then, under general stability conditions, one can show that (E and Engquist 2003):

$$\|U_{\mathrm{HMM}} - \bar{U}_H\| \le Ce(\mathrm{HMM}) \tag{2.37}$$

for some constant C. Therefore, we have

$$\|Qu^\varepsilon - U_{\mathrm{HMM}}\| \le C(\varepsilon^\alpha + (\Delta t)^k + e(\mathrm{HMM})). \tag{2.38}$$

'in' > 'to'

The key to getting concrete error estimates, and thereby giving guidelines to designing multiscale methods, lies in the estimation of e(HMM). This is specific to each problem. But typically, e(HMM) contains several contributions:

- the relaxation error, due to the influence of the transients,
- the sampling error, due to empirical averages being used to estimate the ensemble average,
- the size effect, due to a computational domain of much smaller size being used to estimate quantities on domains of much larger size.

Examples of these error components will be discussed in subsequent sections.

2.4. Difficulties with HMM

The most significant shortcoming of HMM is that it is based on a preconceived macroscale model. If the form of the macroscale model is chosen incorrectly, one cannot expect the resulting HMM procedure to produce accurate results. For example, if the effective macroscale model should be a stochastic ODE but one makes the assumption that it is a deterministic ODE, then the stochastic component of the macroscale solution will not be captured correctly by an HMM based on such an assumption.

There is a important reason for starting with the macro solver: even for problems for which we do have a sufficiently accurate macroscale model, finding an effective numerical algorithm for that macroscale model may still be a significant task. Indeed, this has been the focus of the computational mathematics community for more than 50 years. One example is Euler's equation in gas dynamics, whose solutions typically contain shocks, *i.e.*, discontinuities (LeVeque 1990). In this case, the numerical algorithms have to satisfy certain constraints in order to be able to solve Euler's equation accurately. Obviously this should also be a concern for multiscale methods.

For practical problems of interest, we have often accumulated some knowledge of the macroscale model. Such information can be used when making assumptions about the macroscale model used in HMM. In cases when one makes a wrong assumption, one can still argue that HMM produces an 'optimal approximation' for the macroscale behaviour of the solution in the class of the models considered. In this sense, HMM is a way of addressing the following question: What is the best one can do given the knowledge we have about the problem at all scales?

Another difficulty is that HMM, as presented, is not very 'seamless', for two reasons.

- It requires knowledge of a complete set of macroscale variables to begin with.
- It requires conversion back and forth between the macro- and microstates of the system. This can become rather difficult in actual implementations, particularly when constructing discrete microstates (*e.g.* in molecular dynamics) from continuous macroscale variables.

The seamless strategy proposed in E, Ren and Vanden-Eijnden (2009) is intended to overcome the second difficulty. This will be explained next. The first difficulty is more significant. It can be dealt with in various situations with different levels of satisfaction, as we explain in Section 3.1.2.

2.5. Seamless HMM

To motivate the seamless algorithm, let us consider the stiff ODE example (2.11). If we want an efficient algorithm for capturing the behaviour of x without resolving the detailed behaviour of y, we can simply change the small parameter ε to a larger value ε', the size of which is determined by the accuracy requirement

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= f(x,y), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= -\frac{1}{\varepsilon'}(y-\varphi(x)). \end{aligned} \tag{2.39}$$

This is then solved using standard ODE solvers. This idea of *boosting* the parameters is very simple, but it can also be quite useful, as in Car–Parrinello molecular dynamics (Car and Parrinello 1985).

We can look at this differently. Instead of changing the value of ε, we may change the clock for the microscale model, that is, if we use $\tau = t\varepsilon/\varepsilon'$ in the second equation in (2.39), then (2.39) can be written as

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= f(x,y), \\ \frac{\mathrm{d}y}{\mathrm{d}\tau} &= -\frac{1}{\varepsilon}(y-\varphi(x)). \end{aligned} \tag{2.40}$$

If we discretize this equation using standard ODE solvers but with different time step sizes for the first and second equations in (2.40), we obtain the following algorithm:

$$y^{n+1} = y^n - \frac{\delta\tau}{\varepsilon}(y^n - \varphi(x^n)), \tag{2.41}$$

$$D^{n+1} = y^{n+1}, \tag{2.42}$$

$$x^{n+1} = x^n + \Delta' t f(x^n, D^{n+1}). \tag{2.43}$$

Here $y^n \sim y(n\delta\tau)$ and $x^n \sim x(n\Delta' t)$. The value of $\delta\tau$ is the time step size we

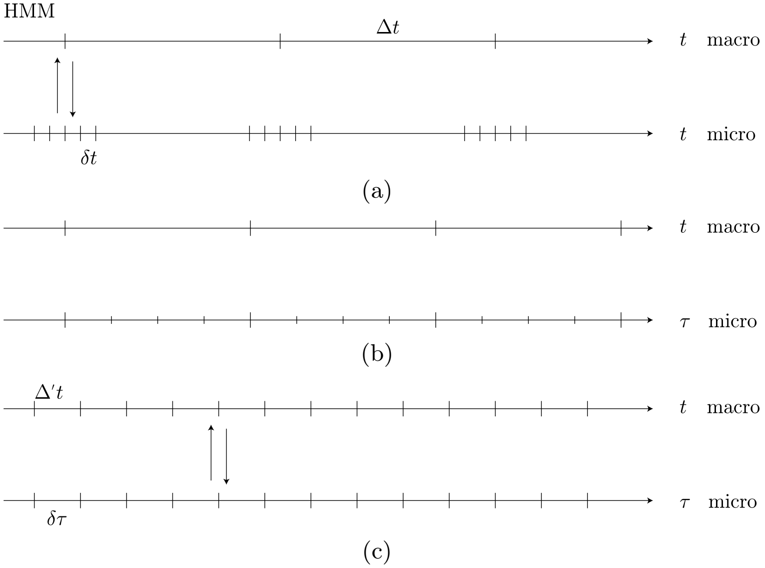


Figure 2.2. (a) Illustration of HMM, and (c) the seamless algorithm. (b) Rescaling the micro time scale.

would use if we were to attempt to solve (2.40) accurately. If (2.40) were the molecular dynamics equations, then $\delta\tau$ would be the standard femtosecond time step size. $\Delta' t$ is the time step one would use for (2.39). It satisfies

$$\frac{\Delta' t}{\varepsilon'} = \frac{\delta\tau}{\varepsilon}. \tag{2.44}$$

In general, $\Delta' t$ should be chosen such that one not only resolves the macro time scale, but also allows the microstate to relax sufficiently, that is, to adjust to the changing macroscale environment. For example, if Δt is the time step size required for accurate resolution of the macroscale dynamics, and if τ_ε is the relaxation time of the microscopic model, then we should choose $\Delta' t = \Delta t/M$ where $M \gg \tau_\varepsilon/\delta\tau$.

The advantage of this second viewpoint is that it is quite general, and it does not require tuning parameters in the microscopic model. In a nutshell, the basic idea is as follows.

1 Run the (constrained) micro solver using its own time step $\delta\tau$.

2 Run the macro solver at a pace that is slower than a standard macro model: $\Delta' t = \Delta t/M$.

3 Exchange data between the micro and macro solvers at every step.

'is' > 'are'

Because data are exchanged at every time step, there is no need to reconstruct new microscale states to initialize the microscale simulation. Intuitively, what one does is to force the microscale model to accommodate the changes in the macroscale environment (here the change in x) at a much faster pace. This is illustrated in Figure 2.2.

The general form of the seamless algorithm can be written as follows.

1 Given the current state of the micro variables $u(\tau)$ and the macro variables $U(t)$, evolve the micro variables for one time step:

$$u(\tau + \delta\tau) = \mathcal{S}_{\delta\tau}(u(\tau); U(t)). \tag{2.45}$$

2 Estimate D:

$$D = \mathcal{D}(u(\tau + \delta\tau)). \tag{2.46}$$

3 Evolve the macro variables:

$$U(t + \Delta' t) = S_{\Delta' t}(U(t); D). \tag{2.47}$$

In this algorithm, we alternate between the macro and micro solvers, each running with its own time step (therefore the micro and macro solvers use different clocks). At every step, the required macroscale data are estimated from the results of the micro model (at that step) and are supplied to the macro solver. The new values of the macrostate variables are then used to constrain the micro solver.

From the consideration of time scales alone, the computational savings in the seamless algorithm come from the fact that effectively the system evolves on the time step $\Delta' t$. In the case when the time scales are disparate, $\Delta' t$ can be much larger than $\delta\tau$. Therefore one can define the savings factor:

$$C_S = \frac{\Delta' t}{\delta\tau} = \frac{\Delta t}{M\delta\tau}. \tag{2.48}$$

For example, assume that the microscopic model is molecular dynamics, and the time step size is femtoseconds ($\delta\tau = 10^{-15}$ seconds): to simulate one second of physical time we need to run the simulation for 10^{15} steps. On the other hand, assume that the relaxation time is of the order of picoseconds (10^{-12} seconds), that is, $O(10^3)$ micro time steps: then $M = 10^5$ is a reasonable choice. Simulating one second of physical time using the seamless algorithm requires 10^5 steps. This is a saving by a factor of 10^{10}. The price to be paid is that we no longer obtain accurate dynamic information at the level of microscopic detail: we can only hope to get the distribution of the fast variables accurately.

A slightly different interpretation of the seamless algorithms will be given in the next subsection.

Example: SDEs with multiple time scales. Going back to the SDE (2.16), the seamless algorithm with forward Euler scheme is simply

$$y^{n+1} = y^n - \frac{\delta\tau}{\varepsilon}(y^n - \phi(x^n)) + \sqrt{\frac{2\delta\tau}{\varepsilon}}\xi^n, \tag{2.49}$$

$$x^{n+1} = x^n + \Delta' t\, f(x^n, y^{n+1}), \tag{2.50}$$

where the $\{\xi^n\}$ are independent normal random variables with mean 0 and variance 1. Note that for HMM we have $x^n \sim x(n\Delta t)$, but for the seamless algorithm we have $x^n \sim x(n\Delta' t) = x(n\Delta t/M)$.

2.6. HMM for type A problems

There are two major types of multiscale problems: type A problems, for which multiscale modelling is needed to resolve local defects or singularities, and type B problems, for which multiscale modelling is needed to supply constitutive relations at the macroscale (E and Engquist 2003). HMM is most easily formulated for type B problems, and so far the formulations we have discussed are mostly for these problems. But HMM can also be used for type A problems, particularly in situations for which the relaxation time scale for the local structure near defects is much faster than the macroscopic dynamic time scale. In that case, the effect of the microscale structure near defects can be viewed as providing effective boundary conditions for the macroscale model at the defects.

One example is the moving contact line problem. There the macroscale model is simply the continuum two-phase flow model. The missing data are the boundary conditions at the contact line, which is then extracted from a molecular dynamics simulation of the local structure around the contact line.

3. ODEs and dynamical systems

3.1. General considerations

3.1.1. Standard HMM scheme

It is useful to revisit the ideas discussed in the previous section in the context of more general stiff ODEs. Consider a generic multiscale system of the type

$$\begin{aligned} \frac{\mathrm{d}x^\epsilon}{\mathrm{d}t} &= f(x^\epsilon, y^\epsilon), \\ \frac{\mathrm{d}y^\epsilon}{\mathrm{d}t} &= \frac{1}{\epsilon} g(x^\epsilon, y^\epsilon). \end{aligned} \tag{3.1}$$

Let us assume that, in the limit as $\epsilon \to 0$, the dynamics of this system can be described by the parametric distribution

$$\mu_{X(t)}(\mathrm{d}y), \tag{3.2}$$

where $\mu_x(\mathrm{d}y)$ is the invariant distribution of

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = g(x, Y), \tag{3.3}$$

with x viewed as a fixed parameter, and $X(t)$ in (3.2) solves

$$\frac{\mathrm{d}X}{\mathrm{d}t} = F(X(t)), \tag{3.4}$$

where

$$F(x) = \int f(x, y)\mu_x(\mathrm{d}y). \tag{3.5}$$

In other words, we assume that, as $\epsilon \to 0$, x^ϵ converges pathwise to X, while y^ϵ converges in law to Y, in a fashion that is captured by the distribution (3.2). This way of phrasing the limit theorem encompasses all the situations of interest to HMM. For example, (3.1) might be a set of stiff ODEs with a slow manifold structure, in which case $\mu_x(y)$ is atomic on this manifold (*i.e.*, for any x, $\mu_x(\mathrm{d}y) = \delta_{y(x)}(\mathrm{d}x)$ for some $y(x)$). Alternatively, (3.1) might be a set of SDEs, in which case $\mu_x(y)$ is typically broad (E 2011).

The standard HMM scheme starts with an integrator for (3.4). We will denote the numerical map associated with this integrator by η_t^F. Here the superscript F makes explicit the dependence of this map on the function F defined in (3.5), which we do not know explicitly: the aim of HMM is to approximate it on the fly. Note that η_t^F could depend on the derivatives of F itself, in which case the HMM scheme will have to estimate these as well as F: the F-information needed in the map η_t^F will be denoted collectively by $(F(x), DF(x), \ldots)$. In order to obtain this input, let ζ_t^x denote the numerical map associated with (3.3), where x is now a parameter. Finally, given the output $(\zeta_{\delta t}^x(y_0), (\zeta_{\delta t}^x)^{o2}(y_0), \ldots, (\zeta_{\delta t}^x)^{oN}(y_0))$ of this map for N micro steps of size δt starting from the initial condition $Y(0) = y_0$, let

$$(\tilde{F}(x), D\tilde{F}(x), \ldots) = \mathcal{F}\big(\zeta_{\delta t}^x(y_0), (\zeta_{\delta t}^x)^{o2}(y_0), \ldots, (\zeta_{\delta t}^x)^{oN}(y_0)\big) \tag{3.6}$$

denote the estimator for $(F(x), DF(x), \ldots)$. The HMM scheme then iterates using the following procedure.

1 *Microscale solver.* Given $X_{m\Delta t}$, generate

$$(\zeta_{\delta t}^{X_{m\Delta t}}(y_0), (\zeta_{\delta t}^{X_{m\Delta t}})^{o2}(y_0), \ldots, (\zeta_{\delta t}^{X_{m\Delta t}})^{oN}(y_0)).$$

2 *Estimator.* Use these data to estimate

$$(\tilde{F}(X_{m\Delta t}), D\tilde{F}(X_{m\Delta t}), \ldots) = \\ (\zeta_{\delta t}^{X_{m\Delta t}}(y_0), (\zeta_{\delta t}^{X_{m\Delta t}})^{o2}(y_0), \ldots, (\zeta_{\delta t}^{X_{m\Delta t}})^{oN}(y_0)).$$

3 *Macroscale solver.* Set

$$X_{(m+1)\Delta t} = \eta_{\Delta t}^{\tilde{F}}(X_{m\Delta t}).$$

Note that in the micro solver step, an initial y_0 is needed. A natural choice is to use the last update of the previous iteration, that is, at iteration $m+1$ take $y_0^{m+1} = (\zeta_{\delta t}^{X_{m\Delta t}})^{oN}(y_0^m)$.

3.1.2. Seamless HMM schemes

Let us now revisit the question of developing seamless HMM schemes, already discussed in Section 2.5. To this end, consider again the generic multiscale system (3.1), let ϕ_t denote a map that approximates numerically the solution of

$$\begin{aligned} \frac{\mathrm{d}X}{\mathrm{d}t} &= f(X,Y), \\ \frac{\mathrm{d}Y}{\mathrm{d}t} &= 0, \end{aligned} \tag{3.7}$$

and let ψ_t denote a map that approximates numerically the solution of

$$\begin{aligned} \frac{\mathrm{d}X}{\mathrm{d}t} &= 0, \\ \frac{\mathrm{d}Y}{\mathrm{d}t} &= g(X,Y). \end{aligned} \tag{3.8}$$

Note that the map ψ_t associated with (3.8) leaves the slow variables x^ϵ unchanged and could, for example, be used in the micro solver of HMM: in the notation of the last section $\psi_t(x,y) = (x, \xi_t^x(y))$. In terms of these maps, a vanilla splitting scheme for (3.1) with time step δt would, for example, iterate

$$\phi_{\delta t} \circ \psi_{\delta t/\epsilon}. \tag{3.9}$$

Alternatively, we could take advantage of the fact that the map associated with (3.7) probably remains stable with larger time steps than that associated with (3.8), and use

$$\phi_{M\delta t} \circ \psi_{\delta t/\epsilon}^{\circ M}, \tag{3.10}$$

instead of M iterates of (3.9), to advance time by $M\delta t$.

It is interesting to revisit the HMM using this notation. Suppose that as the estimator in the HMM we simply pick the last value of the fast variables after M steps of size δt, then take a macro time step of size Δt. The resulting scheme can be written compactly as

$$\phi_{\Delta t} \circ \psi_{\delta t/\epsilon}^{\circ M}. \tag{3.11}$$

This scheme has exactly the same cost as (3.10), but it advances the variables by Δt instead of $M\delta t$, thereby resulting in an efficiency gain if

$$\Delta t > M\delta t. \tag{3.12}$$

If (3.1) were a stiff system of ODEs, the estimator chosen to derive (3.11) is the standard one, and so we already know why HMM works. Remarkably, it works in the general case. To see why, note that (3.11) can alternatively be interpreted as a time-splitting scheme for (*cf.* (3.1))

$$\begin{aligned} \frac{\mathrm{d}x^{\epsilon'}}{\mathrm{d}t} &= f(x^{\epsilon'}, y^{\epsilon'}), \\ \frac{\mathrm{d}y^{\epsilon'}}{\mathrm{d}t} &= \frac{1}{\epsilon'} g(x^{\epsilon'}, y^{\epsilon'}), \end{aligned} \tag{3.13}$$

where $\epsilon' = \epsilon\Delta t/M\delta t$. Indeed, (3.11) is consistent with (3.13) in the limit

$$\Delta t \to 0, \quad \delta t \to 0 \quad \text{with} \quad \Delta t/M\delta t \to \alpha \in (1, \infty), \tag{3.14}$$

where we used the inequality (3.12) to restrict the range of α. Thus (3.13) is a version of the original system (3.1) in which ϵ has been boosted to the higher value $\epsilon' = \epsilon\alpha > \epsilon$, thereby making it less stiff. The key observation that this re-interpretation permits is this: as long as

$$\epsilon\Delta t \ll M\delta t \quad \Leftrightarrow \quad \epsilon' = \epsilon\alpha \ll 1, \tag{3.15}$$

the limit theorem we used to justify the HMM scheme guarantees that the solution to (3.13) remains close to that of the original system (3.1), in the sense that $x^{\epsilon'} \approx x^{\epsilon}$ pathwise, and the stationary distribution of $y^{\epsilon'}$ conditional on the current value of $x^{\epsilon'}$ approximates that of y^{ϵ} conditional on the current value of x^{ϵ}. This also implies that the numerical solution generated by (3.11) gives an approximation to the solution of the original system (3.1), which is accurate in the same sense. Thus, inequalities (3.12) and (3.15) give the conditions necessary for the scheme to provide an efficiency gain and remain accurate, respectively.

To make the link with the discussion in Section 2.5, note that (3.13) can also be interpreted as a system in which ϵ remains the same but the clock of the fast variables has been changed to be consistent with the time-rescaling $\tau = t\epsilon/\epsilon'$.

In practice, the values of Δt for which (3.15) remain valid may be too large for the scheme (3.11) to be stable, which seems to put an unnecessary cap on the efficiency gain achievable by the scheme. In these cases, letting $\Delta' t = \Delta t/M$, we can use the following scheme, whose cost is close to that of (3.11) and also advances the variables by Δt at every iterate (*i.e.*, provides the same efficiency boost) without suffering from the same stability limitations:

$$(\phi_{\Delta' t} \circ \psi_{\delta t/\epsilon})^{\circ M}. \tag{3.16}$$

In terms of $\Delta' t$, (3.15) becomes

$$\Delta' t \ll \frac{\delta t}{\epsilon}. \tag{3.17}$$

If the numerical error of the vanilla time-splitting scheme (3.9) is $C(\delta t/\epsilon)^a$

for some $C > 0$, then that of the scheme (3.16) is

$$C(\Delta' t/\epsilon')^a + \bar{C}\epsilon'^b = C(\delta t/\epsilon)^a + \bar{C}(\epsilon\Delta' t/\delta t)^b, \tag{3.18}$$

where $\bar{C} > 0$ and the exponent b depends on the rate of convergence of the solution of (3.1) towards its limit as $\epsilon \to 0$. Note that in order to be useful, the constant C in the error estimate (3.18) needs to be independent of ϵ. This property is not standard for time-dependent problems, but is expected to hold if the map ψ_t for (3.8) is chosen appropriately, and takes advantage of the fact that the fast variables reach a stable (statistical) steady state conditional on the slow variables being fixed. The scheme (3.16) was first introduced by Fatkullin and Vanden-Eijnden (2004) and further developed by E *et al.* (2009). It is a seamless version of HMM in which the slow and fast variables are evolved concurrently, and no re-initialization of the fast variables is required. This last feature is especially handy in situations where the values taken by the fast variables are constrained by the slow variables, which arises in molecular dynamics simulations, for example (for applications in this context see, *e.g.*, Maragliano and Vanden-Eijnden (2006, 2007), Abrams and Vanden-Eijnden (2010)).

The scheme (3.16) is so simple, both in its implementation and interpretation, that it may seem like a trivialization of HMM: just boost the value of ϵ in the original system (3.1), adjust the time step accordingly, and we are done! Yet it opens the door to more sophisticated strategies. Indeed, since $\bar{C}(\epsilon\Delta' t/\delta t)^b$ is the only extra term appearing in the error estimate (3.18), and this term is controlled by the rate of convergence of (3.1) towards its limit as $\epsilon \to 0$, this suggests modifying (3.1) in such a way that the limit is unchanged but convergence is accelerated. How to actually achieve this should be examined on a case-by-case basis.

The seamless version of HMM also opens the door to schemes in which the slow variables do not need to be identified beforehand, as was first noted by Vanden-Eijnden (2007). To see why, suppose that (3.1) is replaced by

$$\frac{\mathrm{d}z^\epsilon}{\mathrm{d}t} = F(z^\epsilon) + \frac{1}{\epsilon}G(z^\epsilon), \tag{3.19}$$

and let us assume that there exists some diffeomorphism that maps z^ϵ onto the variables (x^ϵ, y^ϵ) satisfying (3.1). More specifically, we require that $x^\epsilon = \theta(z^\epsilon)$ for some θ such that

$$D\theta(z)G(z) = 0, \tag{3.20}$$

so that

$$\frac{\mathrm{d}x^\epsilon}{\mathrm{d}t} = D\theta(z^\epsilon)F(z^\epsilon), \tag{3.21}$$

that is, the variables $x^\epsilon = \theta(z^\epsilon)$ are indeed slow. We also require that no other (hidden) slow variable exists, that is, $x^\epsilon = \theta(z^\epsilon)$ are those entering the limit theorem on which the HMM scheme is built. Even if we do not know the explicit form of θ we can make use of its existence. To see how, let Φ_t denote the map that approximates the solution to

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = F(Z), \tag{3.22}$$

and let Ψ_t denote the map that approximates the solution to

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = G(Z). \tag{3.23}$$

In terms of these maps, a vanilla time-splitting integrator for (3.19) would, for example, read (*cf.* (3.9))

$$\Phi_{\delta t} \circ \Psi_{\delta t/\epsilon} \tag{3.24}$$

or (*cf.* (3.10))

$$\Phi_{M\delta t} \circ \Psi^{\circ M}_{\delta t/\epsilon}. \tag{3.25}$$

This last scheme updates the variables by $M\delta t$ at every iterate, and the discussion above suggests replacing it instead by (*cf.* (3.11))

$$\Phi_{\Delta t} \circ \Psi^{\circ M}_{\delta t/\epsilon} \tag{3.26}$$

or

$$(\Phi_{\Delta' t} \circ \Psi_{\delta t/\epsilon})^{\circ M}, \tag{3.27}$$

where again $\Delta' t = \Delta t/M$. In the limit (3.14), both (3.26) and (3.27) are consistent with a version of (3.19) where ϵ has been boosted to a higher value $\epsilon' = \epsilon\alpha = \epsilon\Delta t/M\delta t$, that is,

$$\frac{\mathrm{d}z^{\epsilon'}}{\mathrm{d}t} = F(z^{\epsilon'}) + \frac{1}{\epsilon'}G(z^{\epsilon'}), \tag{3.28}$$

which can again be used to explain why the scheme is accurate: essentially (3.15) needs to hold, and if the numerical error of the vanilla time-splitting scheme (3.24) is $C(\delta t/\epsilon)^a$, for some $C > 0$, then that of the scheme (3.27) is again given by (3.18).

The scheme (3.27) was first proposed by Vanden-Eijnden (2007). Later a very close variant of this scheme was proposed by Tao, Owhadi and Marsden (2010) under the name 'FLAVORS'. Specifically, if we let $\chi_t^{1/\epsilon}$ denote the map that approximates numerically the solution of (3.19), and assume that we can control the size of ϵ in this map, then the scheme proposed by Tao *et al.* (2010) is (in our notation)

$$\left(\chi^0_{\Delta' t - \delta t} \circ \chi^{1/\epsilon}_{\delta t}\right)^{\circ M}. \tag{3.29}$$

As can be seen, compared to (3.27), this scheme amounts to a slight modification of the way the time-splitting is implemented: (3.27) evolves the variables by first using the term G/ϵ for a δt step, and then the term F for a $\Delta' t$ step, whereas (3.29) evolves the variables by first using the full $F + G/\epsilon$ for a δt step, and then the term F alone for a $\Delta' t - \delta t$ step. In terms of accuracy, this modification is inconsequential: like (3.27), (3.29) is consistent with (3.28) in the limit (3.15). It also leads to no significant change in cost.

3.2. ODEs with oscillatory solutions

One advantage of the HMM framework is that sophisticated data estimation techniques can be used in the estimator step in order to improve its performance. This is most easily seen in the context of ODEs with highly oscillatory solutions. The examples discussed here also illustrate the subtleties involved in knowing or not knowing the slow variables. To be consistent with the existing literature, we will also use slightly different notation here.

Ordinary differential equations (ODEs) with highly oscillatory solutions are naturally challenging for numerical simulation. Several different numerical approaches have been suggested, each appropriate to some class of ODEs. For harmonic oscillatory problems, traditional numerical approaches attempt to either filter out or rapidly fit ϵ-scale oscillations to some known functions in order to reduce complexity (*e.g.*, Gautschi 1961, Kreiss 1992, Scheid 1983), or use some form of Poincaré map to determine slow changes in the orbital structure (Gear and Gallivan 1982, Petzold, Laurent and Jeng 1997). A general class of approaches aiming at Hamiltonian systems are geometric integration schemes that preserve a discrete version of certain invariance. We refer readers to Hairer, Lubich and Wanner (2002) and Leimkuhler and Reich (2004) for a more extensive list of the literature. These types of algorithms require $O(\epsilon^{-1})$ time steps. In certain applications, special considerations are given to the high cost of evaluating non-local potential in large systems: see, for instance, the impulse method and its derivatives (Leimkuhler and Reich 2004). For a recent review of numerical methods for highly oscillatory systems, see Cohen, Jahnke, Lorenz and Lubich (2006).

In recent years, substantial progress has been made in HMM-style integration; see, for example, E and Lu (2007), E *et al.* (2009), Calvo and Sanz-Serna (2009, 2010), Chartier, Murua and Sanz-Serna (2010), Sanz-Serna (2009), Ariel, Sanz-Serna, and Tsai (2012) and Tao *et al.* (2010).

3.2.1. A kernel averaging theorem

In the present context, the appropriate HMM is best illustrated by Figure 3.1: the upper axis represents the grid used for U imposed by the macro solver, and the lower axis represents the finer grids on which solutions of the microscopic equation are constructed by the micro solver, with initial conditions determined from the values of U. The downward-pointing arrows

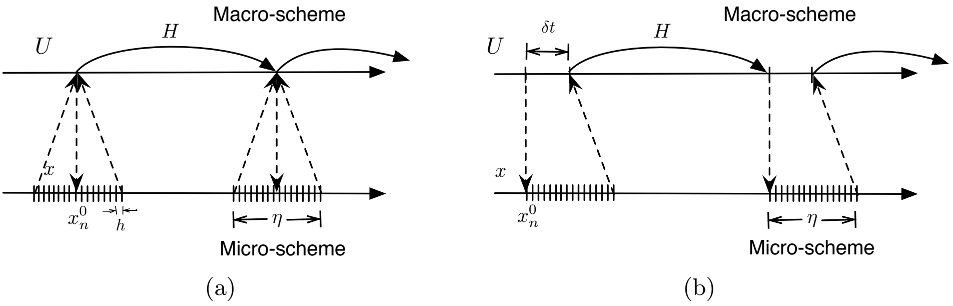


Figure 3.1. Two typical structures of the proposed multiscale algorithm. Figure (a) is for time-reversible problems.

symbolize the determination of a consistent initial condition for the microscale equation from U at t_n: this process is referred to as reconstruction. The upward-pointing arrows relate the evaluation of the effective equation from the time history of microscale variables computed in a short time interval η: this process is referred to as data estimation. Data estimation includes an evaluation of $\mathrm{d}U/\mathrm{d}t$ as well as the values of U at different times. The process of evaluating a consistent value of Uat time t^* from the time history of the microscopic variables is referred to as compression. Typically, this evaluation is accomplished by filtering with a chosen compactly supported kernel K.

Hence we may present the above procedures algorithmically as follows.

1 *Force estimation.*

(a) Reconstruction: at $T = t_n$, $U^n \mapsto x_n^0 = RU^n$.

(b) Solve

$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = f_\epsilon(x_n, t), \quad x(t_n) = x_n^0,$$

for $t \in [t_n, t_n + \eta]$.

(c) Compression: $U^* = Q[x_n(\cdot)]$.

(d) Estimate force: $\bar{f}(t_n + \delta t_*) \sim \tilde{f}(t_n + \delta t_*) = K * f_\epsilon(x_n)$.

(e) If $n = 0$, prepare (reset) the initial data by $\tilde{K}$: $U^0 = \tilde{K} * x_n$.

2 Evolve the macro variables:

$$\{U^j\}_{j=0}^n, U^*, \quad \bar{f}(t_n + \delta t_*) \longrightarrow U^{n+1}, \quad T = t_{n+1}.$$

3 Repeat.

In this subsection, the reconstruction operator R will be taken to be the identity operator, that is, $RU^n = U^n$, and the compression $Q[x_n(\cdot)] = x_n(t_n + \delta t_*)$.

The two diagrams in Figure 3.1 therefore differentiate the general structures of the HMM ODE solvers. The top axis represents the macro grid on which we solve for the values of U (step 2), and under each grid node on the top axis is a corresponding micro grid over which step 1(b) takes place. The arrow pointing from each macro grid point t_n down to a micro grid denotes the action taken in step 1(a), while the arrows pointing from each micro grid up to the macro axis represent steps 1(c) and 1(d).

The schemes depicted in Figure 3.1 allow $\bar{f}$ and U_* (see steps 1 and 2) to be evaluated at the time location $t_n + \delta t_*$. The structure depicted in Figure 3.1(a) requires $\delta t_* = 0$, and is thus suitable for implementing a linear multi-step method as macro solver. In this type of scheme, the effective force is usually evaluated at the left end of each fine-scale evolution, and therefore a non-symmetric kernel is used. In comparison, in the structure in Figure 3.1(b), $\delta t_* = \eta/2$, and a one-step method such as Runge–Kutta can be used as a macro solver. Due to this particular structure, a symmetric kernel can be used. In what follows, we use H and h for the discrete time steps used in the macro and micro grid respectively.

Force estimation by specialized kernels. In our formulation, we need to estimate the effective force locally at a point using the microscale data (steps 1(b)–1(d)). Motivated by the analytic averaging techniques (see, *e.g.*, E (2011)), we hypothesize that the effective force of a system of interest can be defined by

$$\bar{f}(t) = \lim_{\delta \longrightarrow 0} \left[\lim_{\epsilon \longrightarrow 0} \frac{1}{\delta} \int_t^{t+\delta} f_\epsilon(\tau) \, \mathrm{d}\tau \right].$$

We assume that $\bar{f}$ is slowly varying in the sense that

$$\left| \frac{\mathrm{d}^p}{\mathrm{d}t^p} \bar{f}(t) \right| \leq C \;\text{ for } 0 \leq p \leq s$$

for some constant C independent of ϵ. Our goal in this section is to show that time filtering using a kernel $K_\eta^{p,q}$ converges to $\bar{f}$, defined below, with $\eta = \eta(\epsilon) \longrightarrow 0$ as $\epsilon \longrightarrow 0$:

$$K_\eta^{p,q} * f_\epsilon = K_\eta^{p,q} * \big(\bar{f} + g_\epsilon(t)\big) \longrightarrow \bar{f} \quad \text{as } \epsilon \longrightarrow 0. \tag{3.30}$$

In many situations, f_ϵ or g_ϵ assumes special forms such as $f_\epsilon(t) = f_\epsilon(t, t/\epsilon)$, which are periodic in the second variable. For example,

$$\frac{\mathrm{d}u}{\mathrm{d}t} = f_\epsilon(u,t) = \frac{\mathrm{i}}{\epsilon}\lambda u + \phi(t)$$

has solution

$$u(t) = \mathrm{e}^{\mathrm{i}\epsilon^{-1}\lambda t}\left(u_0 + \int_0^t \mathrm{e}^{-\mathrm{i}\epsilon^{-1}\lambda s} \phi(s) \, \mathrm{d}s \right).$$

The force $f_\epsilon(t) = f_\epsilon(u,t) = \frac{\mathrm{i}}{\epsilon}\lambda u + h(t)$ is of the form $f_\epsilon(t, t/\epsilon)$. In this case, we define

$$\bar{f}(t) = \int_0^1 f(t,s)\,\mathrm{d}s,$$

and

$$g_\epsilon(t) = g\left(t, \frac{t}{\epsilon}\right) = f_\epsilon - \bar{f}(t).$$

We will use $\mathbb{K}^{p,q}$ to denote the kernel space discussed here. $K \in \mathbb{K}^{p,q}(I)$ if $K \in C_c^q(\mathbb{R})$ with $\operatorname{supp}(K) = I$, and

$$\int_{\mathbb{R}} K(t)t^r\,\mathrm{d}t = \begin{cases} 1 & r = 0, \\ 0 & 1 \le r \le p. \end{cases}$$

Furthermore, we will use $K_\eta(t)$ to denote the scaling of K:

$$K_\eta(t) := \frac{1}{\eta}K\left(\frac{t}{\eta}\right).$$

For convenience, we will also use $K^{p,q}$ to denote a function in $\mathbb{K}^{p,q}(I)$. We use $K \in \mathbb{K}^{p,q}([-1,0])$ if $\delta t^* = 0$ in step 1 of the algorithm described above, and $K \in \mathbb{K}^{p,q}([-1,1])$ if $\delta t^* = \eta/2$.

Theorem 3.1. Let $f_\epsilon(t) = f(t, t/\epsilon)$, where $f(t,s)$ is 1-periodic in the second variable and $\partial^r f(t,s)/\partial t^r$ is continuous for $r = 0, \dots, p-1$. For any $K \in \mathbb{K}^{p,q}$, there exist constants $C_{\bar{f},g_\epsilon}$and C_{g_ϵ}, independent of ϵ and η, such that

$$E = |K_\eta * f_\epsilon(t) - \bar{f}(t)| \le C_{\bar{f},g_\epsilon}\eta^p + C_{g_\epsilon}\left(\frac{\epsilon}{\eta}\right)^q.$$

Furthermore, the error is minimized if η is chosen to scale with $\epsilon^{q/(p+q)}$.

3.2.2. Problems with hidden slow variables

We start by considering a class of systems that has an explicit slow–fast grouping in the solution's components:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \frac{1}{\epsilon}f(x,y,t) + g(x,y,t), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= h(x,y,t). \end{aligned} \tag{3.31}$$

Here the x components stay bounded but are highly oscillatory, and the y components are called the slow variables of the system, since their time derivatives are formally bounded. This would be the case for Hamiltonian systems written in the action and angle coordinates (E 2011). If $x(t)$ yields an invariant measure on some fixed manifold $\mathcal{M}$, then $y(t)$ can be consistently approximated in any constant time interval by the averaged equation

$$\frac{\mathrm{d}\bar{y}}{\mathrm{d}t} = \bar{h}(y,t) := \int_{\mathcal{M}} h(x,y,t)\,\mathrm{d}\mu(x). \tag{3.32}$$

Such systems are widely studied in the construction of multiscale methods; see, for instance, Vanden-Eijnden (2003) or E (2003). In this case, it is reasonable to use y as the macroscopic variable, that is, $U = \bar{y} \simeq y$, and $\mathcal{R}(U, D_R) = (x^*, y)$, where D_R gives the value $x^* \in \mathcal{M}$. For example, x^* may be taken from the x values in the previous microscale simulation. The compression $\mathcal{Q}$ may simply be $\mathcal{Q}(x, y) = y$.

However, if there are resonances among the oscillations, $x(t)$ is unlikely to remain on a fixed invariant manifold (Kreiss 1992), and more sophistication in the algorithm is needed. One can see the essence of this problem from the simple example

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = \frac{\mathrm{i}}{\epsilon}x + g(x), \ \ x(0) = 1, \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = h(x), \ \ y(0) = y_0. \end{cases} \Longrightarrow \begin{cases} \dfrac{\mathrm{d}w}{\mathrm{d}t} = \mathrm{e}^{-\frac{\mathrm{i}}{\epsilon}t} g(\mathrm{e}^{\frac{\mathrm{i}}{\epsilon}t}w), \ \ x(t) = \mathrm{e}^{\frac{\mathrm{i}}{\epsilon}t}w(t), \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = h(\mathrm{e}^{\frac{\mathrm{i}}{\epsilon}t}w), \ \ y(0) = y_0. \end{cases}$$

Let us formally decompose $\mathrm{e}^{-\frac{\mathrm{i}}{\epsilon}t} g(\mathrm{e}^{\frac{\mathrm{i}}{\epsilon}t}w) = \bar{g}(w) + \alpha(\mathrm{e}^{\frac{\mathrm{i}}{\epsilon}t}, w)$, where $\bar{g}$ does not depend on any fast oscillations. If $\bar{g} \equiv 0$, then $w(t)$ stays close to 1, due to strong self-averaging in α. Thus (3.32) corresponds to averaging h over the unit circle, and $\mathrm{d}\mu$ is the arc-length element. Resonance in this system corresponds to the case where $\bar{g}$ is non-zero. Consequently, the averaging has to be performed with the correct measure:

$$\frac{\mathrm{d}\bar{y}}{\mathrm{d}t} = \bar{h}(y, t) := \int_{\mathcal{M}(t)} h(x, y, t)\,\mathrm{d}\mu(x; t).$$

For example, if $g(x) = x$, then $\bar{g}(w) = w$, and $\alpha \equiv 0$. Consequently, $\mathcal{M}(t)$ is a circle with radius equal to $w(t) = \exp(t)$. Without knowledge of $w(t)$, it is impossible to define a consistent reconstruction operator $\mathcal{R}$, and consequently it is impossible to build a convergent multiscale algorithm. In some of the literature, the issue caused by resonance is referred to as the system having hidden slow variables (Fatkullin and Vanden-Eijnden 2004, Engquist and Tsai 2005, Ariel, Engquist, Kreiss and Tsai 2009*c*, Tao *et al.* 2010). It is essential that a multiscale method should compute the effect of the hidden slow variables accurately.

It is possible to design multiscale algorithms that compute the effective behaviour of highly oscillatory dynamical systems by using slow variables (Ariel, Engquist and Tsai 2009*a*). We continue our discussion using the previous example, but instead we rewrite the equation for x as a system in $\mathbb{R}^2$:

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}t} &= \frac{1}{\epsilon}x_2 + x_1, \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} &= -\frac{1}{\epsilon}x_1 + x_2, \end{aligned}$$

with initial conditions $(x_1(0), x_2(0)) = (0, 1)$. Thus

$$(x_1(t), x_2(t)) = (\mathrm{e}^t \sin \epsilon^{-1} t,\ \mathrm{e}^t \cos \epsilon^{-1} t).$$

Taking $I = x_1^2 + x_2^2$, we notice that I has a bounded derivative along the trajectory of the solution, that is, $I' := (\mathrm{d}/\mathrm{d}t) I(x_1(t), x_2(t)) = 2I$ is independent of ϵ. For this particular example one can easily solve for I, $I(t) = I(0)\,\mathrm{e}^{2t}$. In fact, the uniform bound on I' indicates the *slow* nature of $I(x_1(t), x_2(t))$ when compared with the fast oscillations in $(x_1(t), x_2(t))$. This type of characterization of the effective dynamics of highly oscillation systems is commonly used in the literature. In this example it was easy to find the slow quantity I. In general it is difficult, and the proposed research aims to avoid this difficulty. In classical mechanics, I is referred to as the action variable, and there is a corresponding angle variable whose gradient is τ.

We say that $\xi : \mathcal{D}_0 \subset \mathbb{R}^d \mapsto \mathbb{R}$ has a bounded derivative along $x^\epsilon(t)$ if, for $0 < \epsilon < \epsilon_0$,

$$\sup_{x^\epsilon \in \mathcal{D}_0, \epsilon \in (0, \epsilon_0)} |\nabla \xi(x^\epsilon) \cdot \dot{x}^\epsilon| \leq C. \tag{3.33}$$

Such functions are commonly referred to as slow variables of the system; see, for example, Kreiss (1979, 1992), Kreiss and Lorenz (1993), Fatkullin and Vanden-Eijnden (2004), Gear and Kevrekidis (2005) and Ariel *et al.* (2009*a*, 2009*b*). Other approaches to finding slow variables include, for example, those of Artstein, Kevrekidis, Slemrod and Titi (2007*a*) and Artstein, Linshiz and Titi (2007*b*).

Typically, one may expect the trace of a slow variable along the dynamical system's solutions, $\xi \circ x^\epsilon$, to converge to a smooth curve as $\epsilon \to 0$. We shall denote this limit by $\bar{\xi}(t; x_0)$. In designing multiscale algorithms for this type of problem, it is often convenient to aim at constructing accurate approximation of $\bar{\xi}$ by suitable filtering of the oscillations in $x^\epsilon(t)$.

3.2.3. Algorithms with hidden slow variables

Consider an ODE system of the general form

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{1}{\epsilon} f(x) + g(x), \quad x(0) = x_0, \tag{3.34}$$

where $0 < \epsilon \leq \epsilon_0$ is a small parameter that characterizes the separation of time scales in the problem. Let $\zeta(t)$ denote the solution of

$$\frac{\mathrm{d}\zeta}{\mathrm{d}t} = \bar{F}(\zeta) = \int_{S^1} F(\zeta, \sigma, \gamma = 0)\, \mathrm{d}\sigma, \quad \zeta(0) = \xi(0). \tag{3.35}$$

The first step in our algorithm is to identify the slow variables $\xi(x)$. The slow variables can be determined analytically or numerically by identifying coefficients in a predetermined form. For details, see Ariel *et al.* (2009*a*).

Then the ODE (3.34) is integrated using a two-level algorithm: each level corresponds to the integration of (3.34) in a different time scale. The first is a macro solver, which integrates the averaged equation (3.35). The second level is a micro solver that is invoked whenever the macro solver calls for it. Each time the micro solver is invoked, it computes a short time solution of (3.34) using suitable initial data.

1 *Construction of slow variables.* Find functions $\xi_1(x), \dots, \xi_r(x)$ such that $|\nabla_x \xi \dot{x}| \leq C_0$ and $\operatorname{rank}(\partial\xi/\partial x) = r$. (See Ariel *et al.* (2009*a*) for details.)

2 *Multiscale evolution.*

   (a) Initial conditions: $x(0) = x_0$ and $n = 0$.

   (b) *Force estimation.*

      (i) Microscale solver: solve (3.34) in $t \in [t_n, t_n + 2\eta]$ with initial conditions x_n.

      (ii) Averaging: approximate $\dot{\xi}(t_n + \eta)$ by

$$\langle \dot{\xi} \rangle_\eta (t_n + \eta) = (K_\eta * \dot{\xi})(t_n + \eta) = (-\dot{K}_\eta * \xi)(t_n + \eta).$$

   (c) Macroscale solver (forward Euler example): $x_{n+1} = x_n^{M/2} + H\delta x$, where δx is the least-squares solution to the linear system

$$\delta x \cdot \nabla \xi_i = \bar{F}_i(\xi) = \langle \dot{\xi}_i \rangle_\eta,$$

   for all $i = 1, \dots, r$.

   (d) $n = n + 1$. Repeat steps (b) and (c) until time T.

Note that there is no need to actually change the original ODE (3.34) to a form with explicit use of the slow variables.

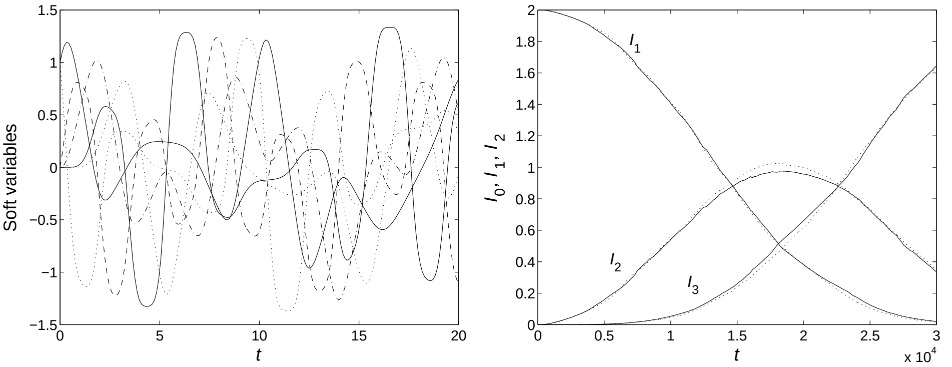
3.2.4. A Fermi–Pasta–Ulam model

The Fermi–Pasta–Ulam model (Fermi, Pasta and Ulam 1955) is a simple system of unit mass particles connected by springs. The springs alternate between stiff linear and soft non-linear ones. Recently, this model was considered by Hairer *et al.* (2002) as a benchmark problem for studying the long-time properties of numerical solutions to stiff ODEs using geometric integrators. The model is derived from the Hamiltonian

$$H(p,q) = \frac{1}{2}\sum_{i=1}^{2k} p_i^2 + \frac{1}{4}\epsilon^{-2}\sum_{i=1}^{k}(q_{2i} - q_{2i-1})^2 + \sum_{i=0}^{k}(q_{2i+1} - q_{2i})^4. \tag{3.36}$$

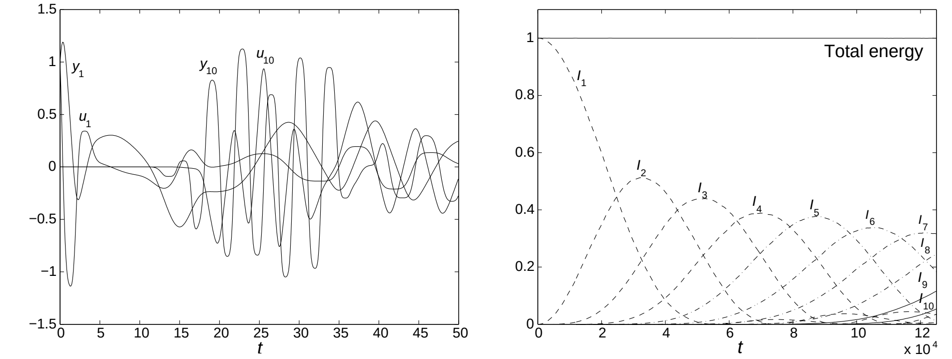
The following linear change of variables is convenient, since it separates the elongations of the stiff springs and associated momentum,

$$x_i = \epsilon^{-1}(q_{2i-1} - q_{2i})/\sqrt{2}, \quad v_i = (p_{2i-1} - p_{2i})/\sqrt{2}, \tag{3.37}$$



(a) (b)

Figure 3.2. Comparison of the HMM approximation for the solution of the Fermi–Pasta–Ulam equations of motion (3.39) with three stiff springs, $k = 3$, with that obtained using the Verlet method with step size of the order of ϵ. (a) Soft variables on an $O(1)$ time scale, and (b) I_1, I_2 and I_3 on an $O(\epsilon^{-1})$ scale. With the above parameters the HMM algorithm runs an order of magnitude faster than the Verlet one. The ratio between running times increases with smaller ϵ.



(a) (b)

Figure 3.3. The HMM approximation for the solution of the Fermi–Pasta–Ulam equations of motion (3.39) with ten stiff springs, $k = 10$. (a) y_1, u_1, y_{10} and u_{10} on an $O(1)$ time scale, and (b) $I_1, \dots, I_{10}$ on an $O(\epsilon^{-1})$ scale. The Verlet method takes too long to integrate.

and a second set of variables associated with the soft springs:

$$y_i = (q_{2i-1} + q_{2i})/\sqrt{2}, \quad u_i = (p_{2i-1} + p_{2i})/\sqrt{2}, \tag{3.38}$$

Defining $y_0 = x_0 = y_{2k+1} = x_{2k+1} = 0$, the equations of motion become

$$\begin{aligned}
\dot{y}_i &= u_i, \\
\dot{x}_i &= \epsilon^{-1} v_i, \\
\dot{u}_i &= -(y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + (y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3, \\
\dot{v}_i &= -\epsilon^{-1} x_i + (y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + (y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3.
\end{aligned} \tag{3.39}$$

Typical initial conditions are $x_1 = y_1 = v_1 = u_1 = 1$ and zero otherwise, which means that initially $k-1$ of the stiff springs are at rest. The system admits $4k-1$ slow variables: first, all the degrees of freedom which are related to the soft springs, y_i and u_i, $i = 1, \ldots, k$; second, the total energy (kinetic + potential) of the stiff springs, $I_i = x_i^2 + v_i^2$; and finally, the relative phases between the different stiff springs, $\phi_k = x_1 x_i + v_1 v_i$, $i = 1, \ldots, k-1$.

On the $O(1)$ time scale the system can be evolved using the HMM algorithm described above. Figure 3.2(a) depicts our results for systems with three stiff springs, $k = 3$, and Figure 3.3(a) shows the results for ten springs, $k = 10$. Simulation parameters for $k = 3$ are $\epsilon = 10^{-4}$, $h = \epsilon/15$, and $H = 0.02$ and $\eta = 15.4\epsilon$. For $k = 10$ we used $\epsilon = 10^{-4}$, $h = \epsilon/15$, and $H = 0.02$ and $\eta = 20.4\epsilon$. On the $O(\epsilon^{-1})$ time scale the dynamics become more interesting as the energies I_i begin to change (Fermi *et al.* 1955, Hairer *et al.* 2002).

3.3. Stochastic simulation algorithms

Chemical kinetics can be described using either the rate equations or stochastic simulation algorithms. The former is suited to situations when the volume (or number) of the participating species is large. The latter is necessary for situations when stochastic and/or discrete effects are important. This is particularly the case in biological applications. Since the rate of reaction often depends exponentially on physical parameters such as temperature, it is quite common for these rates to be very different. Hence the rate equations are typically very stiff and the stochastic simulation algorithms often have a pronounced multiscale nature.

In this subsection we discuss how HMM can be used to overcome the difficulties associated with the disparity of the rates in stochastic simulation algorithms. We begin with the general set-up.

Let us assume that we have a total of N species of molecules, denoted by $S_1, \ldots, S_N$. The number of molecules of species S_k is denoted by x_k, The state vector is then given by $x = (x_1, \ldots, x_N)$, with corresponding state space denoted by $\mathcal{X}$. Assume that there are M reaction channels, each described by its reaction rate and stoichiometric vector:

$$R_j = (a_j, \nu_j), \quad R = \{R_1, \dots, R_M\}. \tag{3.40}$$

Given the state x, the occurrence of the reactions on an infinitesimal time interval $\mathrm{d}t$ is independent for different reactions, and the probability of the reaction R_j happening during this time interval is given by $a_j(x)\,\mathrm{d}t$. After reaction R_j, the state of the system changes to $x + \nu_j$. In the chemistry and biology literature, this is often called the stochastic simulation algorithm (SSA), or Gillespie algorithm, named after an algorithm that realizes this process exactly (see Gillespie (1976)).

Let $X(t)$ be the state variable at time t, and let $\mathbb{E}_x$ denote the expectation conditional on $X(0) = x$. Then the observable $u(x,t) = \mathbb{E}_x f(X(t))$ satisfies the following backward Kolmogorov equation:

$$\frac{\partial u(x,t)}{\partial t} = \sum_j a_j(x)\big(u(x+\nu_j, t) - u(x,t)\big) = (Lu)(x,t). \tag{3.41}$$

The operator L is the infinitesimal generator of the Markov process associated with the chemical kinetic system we are considering.

The SSA proceeds as follows. Let

$$a(x) = \sum_{j=1}^{M_R} a_j(x). \tag{3.42}$$

Assume that the current time is t_n, and the system is at state X_n. We perform the following steps.

1 Generate independent random numbers r_1 and r_2 with uniform distribution on the unit interval $(0,1]$. Let

$$\delta t_{n+1} = -\frac{\ln r_1}{a(X_n)}, \tag{3.43}$$

and let k_{n+1} be the natural number such that

$$\frac{1}{a(X_n)} \sum_{j=0}^{k_{n+1}-1} a_j(X_n) < r_2 \le \frac{1}{a(X_n)} \sum_{j=0}^{k_{n+1}} a_j(X_n), \tag{3.44}$$

where $a_0 = 0$ by convention.

2 Update the time and the state of the system by

$$t_{n+1} = t_n + \delta t_{n+1}, \quad X_{n+1} = X_n + \nu_{k_{n+1}}. \tag{3.45}$$

Then repeat.

In this algorithm, r_1 is used to update the clock and r_2 is used to select the particular reaction to be executed.

Now assume the rate functions have the form

$$a(x) = \left(a^s(x), \frac{1}{\varepsilon}a^f(x)\right), \tag{3.46}$$

where $\varepsilon \ll 1$ represents the ratio of the fast and slow time scales of the system. The corresponding reactions and the associated stoichiometric vectors can be grouped accordingly:

$$R^s = \{(a^s, \nu^s)\}, \quad R^f = \left\{\left(\frac{1}{\varepsilon}a^f, \nu^f\right)\right\}. \tag{3.47}$$

We call R^s the slow reactions and R^f the fast reactions. We have made the simplifying assumption that the rates are divided into two groups. Clearly the algorithm can be easily extended to situations when there are multiple groups. The more serious assumption here is that the groups of slow and fast reactions do not change in time. Some adaptive strategies have to be introduced in order to remove this restriction.

Our interest is in the dynamics of the slow processes, not the detailed dynamics of the fast processes. To this end, an effective system can be derived using standard averaging methods (Kurtz 1973, Papanicolaou 1977). For this purpose it is helpful to introduce an auxiliary process, called the *virtual fast process* (Cao, Gillespie and Petzold 2005). This auxiliary process retains the fast reactions only: all slow reactions are turned off. Intuitively, each realization of the SSA consists of a sequence of realizations of the virtual fast process, punctuated by occasional firing of the slow reactions. Due to the time scale separation, with high probability, the virtual fast process has enough time to relax to equilibrium before another slow reaction takes place. Therefore the effective slow rates for the slow dynamics should simply be the original slow rates averaged over the equilibrium distributions associated with the virtual fast process.

Let μ_x denote the equilibrium distribution of the virtual fast process when it is initialized at x, and let

$$\tilde{a}_i^s(x) = \sum_{z\in\mathcal{X}} a_i^s(z)\mu_x(z). \tag{3.48}$$

It can be shown that the effective dynamics is governed by the set of reactions:

$$\bar{R} = (\bar{a}^s(x), \bar{\nu}^s). \tag{3.49}$$

The effective master equation is given by

$$\frac{\partial u}{\partial t}(x,t) = \sum_{i=1}^{M_s} \tilde{a}_i^s(x)\big(u(x+\nu_i^s, t) - u(x,t)\big). \tag{3.50}$$

We will discuss an algorithm proposed by E, Liu and Vanden-Eijnden (2005*a*), which is a simple modification of the original SSA by adding a nested structure according to the time scale of the rates. The process at each level of the time scale is simulated with an SSA with some effective rates. Results from simulations on fast time scales are used to compute the rates for the SSA at slower time scales.

Let t_n, X_n be the current time and state of the system respectively. The two-level nested SSA proceeds as follows.

1 *Microscale solver: the inner SSA.* Pick an integer N. Run N independent replicas of SSA with the fast reactions $R^f = \{(\varepsilon^{-1}a^f, \nu^f\})\}$ only, for a time interval of $T_0 + T_f$. During this calculation, compute the modified slow rates for $j = 1, \dots, M_s$,

$$\tilde{a}_j^s = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{T_f}\int_{T_0}^{T_f+T_0} a_j^s(X_\tau^k)\,\mathrm{d}\tau, \tag{3.51}$$

where X_τ^k is the result of the kth replica of this auxiliary virtual fast process at virtual time τ whose initial value is $X_{t=0}^k = X_n$, and T_0 is a parameter we choose in order to minimize the effect of the transients to the equilibrium in the virtual fast process.

2 *Macroscale solver: the outer SSA.* Run one step of SSA for the modified slow reactions $\tilde{R}^s = (\tilde{a}^s, \nu^s)$ to generate (t_{n+1}, X_{n+1}) from (t_n, X_n).

Then repeat.

This algorithm is called the nested stochastic simulation algorithm (nested SSA). Unlike standard examples of HMM, here we do not need to know what the slow and fast variables are in order to carry out the computation. The algorithm is formulated in terms of the original variables.

Convergence and efficiency of the nested SSA. The original SSA is an exact realization of the stochastic chemical kinetic system. The nested SSA, on the other hand, is an approximation. The errors in the nested SSA can be analysed using the general strategy for analysing HMM. The details can be found in E, Liu and Vanden-Eijnden (2007*b*).

Let f be a smooth function. Let $\tilde{X}_t$ denote the solution of the nested SSA. Consider the observable $v(x,t) = \mathbb{E}_x f(X_t)$, where the expectation is taken with respect to the randomness in the outer SSA only. Let $u(x,t)$ be the solution of the effective equation (3.50) with $u(x,0) = f(x)$. The following result is proved in E *et al.* (2005*a*, 2007*b*).

Theorem 3.2. For any $T > 0$, there exist constants C and α independent of (N, T_0, T_f) such that

$$\sup_{0\le t\le T, x\in\mathcal{X}} \mathbb{E}|v(x,t) - u(x,t)| \le C\left(\varepsilon + \frac{\mathrm{e}^{-\alpha T_0/\varepsilon}}{1+T_f/\varepsilon} + \frac{1}{\sqrt{N(1+T_f/\varepsilon)}}\right). \quad (3.52)$$

This result can be used to analyse the efficiency of the nested SSA. Given a chemical kinetic system with $R = \{(a_j, \nu_j)\}$, we assume that the total rate $a(x) = \sum a_j(x)$ does not fluctuate greatly in time: $a(x) \sim O(\varepsilon^{-1})$. Given an error tolerance λ, we choose the parameters in the nested SSA so that each term in (3.52) is less than $O(\lambda)$. One possible choice of the parameters is

$$T_0 = 0, \quad N = 1 + \varepsilon^{-1} T_f = \frac{1}{\lambda}. \quad (3.53)$$

The total cost for the nested SSA over a time interval of $O(1)$ is

$$\text{cost} = O(N(1 + T_0/\varepsilon + T_f/\varepsilon)) = O\left(\frac{1}{\lambda^2}\right). \quad (3.54)$$

In comparison, the cost for the direct SSA is

$$\text{cost} = O\left(\frac{1}{\varepsilon}\right). \quad (3.55)$$

since the time step size is of order ε. When $\varepsilon \ll \lambda^2$, the nested SSA is much more efficient than the direct SSA.

Next we discuss the influence of the other numerical parameters on the efficiency. The parameter T_0, which plays the role of numerical relaxation time, has little influence on the efficiency. Given the same error tolerance λ, for the last term in the error estimate (3.52) to be less than $O(\lambda)$, we need to have

$$N\left(1 + \frac{\varepsilon}{T_f}\right) \ge O\left(\frac{1}{\lambda^2}\right). \quad (3.56)$$

Therefore

$$\text{cost} \ge O\left(N\left(1 + \frac{\varepsilon^{-1}}{T_f}\right)\right) = O\left(\frac{1}{\lambda^2}\right), \quad (3.57)$$

which is the same as (3.55) regardless of the value of T_0. The above argument also implies that the optimal cost for the nested SSA is $O\big(\frac{1}{\lambda^2}\big)$ to achieve an error tolerance of λ.

Turning now to the effect of parameter N, the number of realizations for inner SSA, let us see what happens when we take $N = 1$. For the error estimate (3.52) to satisfy the same error tolerance λ, we have to choose

$$1 + \frac{\varepsilon}{T_f} = \frac{1}{\lambda^2}. \quad (3.58)$$

The cost of the nested SSA is given by

$$\text{cost} = O\biggl(N\biggl(1+\frac{\varepsilon}{T_f}\biggr)\biggr) = O\biggl(\frac{1}{\lambda^2}\biggr), \tag{3.59}$$

which is the same as the cost if we use multiple realizations. This means that using multiple realizations in the inner SSA does not increase the efficiency of the overall scheme either. Obviously, using multiple realizations is advantageous for implementation on parallel computers.

Other versions of the nested SSA are discussed by Samant and Vlachos (2005) and Salis and Kaznessis (2005). Although they appear to be quite different, it can be shown that they are essentially the same as the nested SSA discussed here.

A numerical example: a virus infection model. As a concrete example, we discuss a virus infection model studied by E *et al.* (2005*a*) and Haseltine and Rawlings (2002). The model was originally proposed by Srivastava, You, Summers and Yin (2002) as an example of the failure of modelling reacting networks with deterministic dynamics. The reactions considered in this model are listed in Table 3.1 with $M_R = 6$. The species that need to be modelled are *genome*, *struct*, *template* and *virus* ($N_s = 4$). *Genome* is the vehicle of the viral genetic information which can take the form of DNA, positive-strand RNA, negative-strand RNA, or some other variants. *Struct* represents the structural proteins that make up the virus. *Template* refers to the form of the nucleic acid that is transcribed and involved in catalytically synthesizing every viral component. The nucleotides and amino acids are assumed to be available at constant concentrations.

When *template* > 0, the production and degradation of *struct*, which are the third and fifth reactions, marked with $*$ in Table 3.1, are faster than the others. From the reaction rates, we can see that the ratio of time scales is about $\varepsilon = 10^{-3}$. In a system that consists of only the fast reactions,

Table 3.1. Reaction channels of the virus infection model.

nucleotides	$\xrightarrow{a_1=1.\times \mathit{template}}$	*genome*
nucleotides + *genome*	$\xrightarrow{a_2=.025\times \mathit{genome}}$	*template*
nucleotides + *amino acids*	$\xrightarrow{a_3=1000\times \mathit{template}}$	*struct* *
template	$\xrightarrow{a_4=.25\times \mathit{template}}$	*degraded*
struct	$\xrightarrow{a_5=1.9985\times \mathit{struct}}$	*degraded/secreted* *
genome + *struct*	$\xrightarrow{a_6=7.5d-6\times \mathit{genome}\times \mathit{struct}}$	*virus*

Table 3.2. The reduced virus infection model.

$\textit{nucleotides}$	$\xrightarrow{a_1=1.\times \textit{template}}$	$\textit{genome}$
$\textit{nucleotides} + \textit{genome}$	$\xrightarrow{a_2=.025\times \textit{genome}}$	$\textit{template}$
$\textit{template}$	$\xrightarrow{a_4=.25\times \textit{template}}$	$\textit{degraded}$
$\textit{genome} + \textit{struct}$	$\xrightarrow{a_6=3.75d-3\times \textit{genome}^2\times \textit{struct}}$	$\textit{virus}$

Table 3.3. Efficiency of the nested SSA for the virus infection model.

T_f/ε	1	4	16	64	'Exact'
CPU	154.8	461.3	2068.2	9190.9	34806.4
$\overline{\textit{template}}$	4.027	3.947	3.796	3.757	3.717 ± 0.005
var(*template*)	5.401	5.254	5.007	4.882	4.978 ± 0.005

struct has an equilibrium distribution which is Poisson, with parameter $\lambda = 500 \times \textit{template}$, such that

$$\mathbb{P}_{\textit{template}}(\textit{struct} = n) = \frac{(500 \times \textit{template})^n}{n!} \exp(-500 \times \textit{template}). \qquad (3.60)$$

Note that *struct* only shows up in the last slow reaction. The reduced dynamics in the form of the slow reactions ($a_{1,2,4,6}$), with the rates averaged with respect to the quasi-equilibrium of the fast reactions ($a_{3,5}$), can be given as a system with four reactions, shown in Table 3.2. The initial condition is chosen to be

$$(\textit{struct},\ \textit{genome},\ \textit{template},\ \textit{virus}) = (0, 0, 10, 0). \qquad (3.61)$$

The mean value and the variance of *template* at time $T = 20$ are used as a benchmark. A computation of these values by a direct SSA using $N_0 = 10^6$ realizations leads to

$$\overline{\textit{template}} = 3.7170 \pm 0.005, \quad \mathrm{var}(\textit{template}) = 4.9777 \pm 0.005. \qquad (3.62)$$

For the nested SSA, we make a series of simulations in which we choose the size of the ensemble and the simulation time of the inner SSA according to

$$(N, T_0, T/\varepsilon) = (1, 0, 2^{2k}), \qquad (3.63)$$

for different values of $k = 0, 1, 2, 3, \ldots$. The error estimate in (3.52) then implies that the error δ should decay with the rate

$$\delta = O(2^{-k}), \qquad (3.64)$$

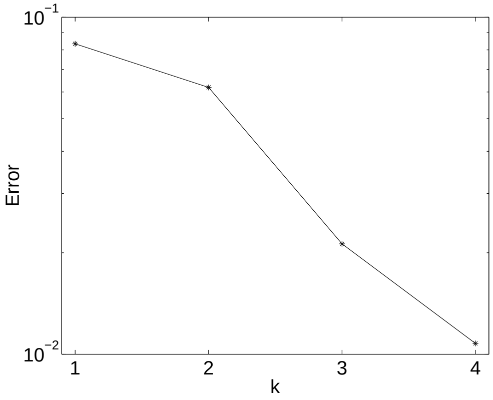


Figure 3.4. Relative errors of $\overline{template}$ using the nested SSA for the virus infection model (courtesy of Di Liu).

which is consistent with the results in Figure 3.4. Table 3.3 gives the total CPU time and the obtained values of $\overline{template}$ and $\mathrm{var}(template)$, with the parameters of inner SSA chosen according to (3.63) and using $N_0 = 10^6$ realizations of the outer SSA (the same as in the direct SSA).

4. Finite element HMM

Finite element heterogeneous multiscale methods (FE-HMM) for homogenization problems, first proposed by E and Engquist (2003), have been developed for elliptic problems (Abdulle 2005*b*, E, Ming and Zhang 2005*b*, Abdulle 2006, Abdulle and Schwab 2005), elastic problems (Abdulle 2006), parabolic problems (Ming and Zhang 2007, Abdulle and Vilmart 2012*a*), wave problems (Abdulle and Grote 2011), and advection–diffusion problems (Abdulle 2005*a*, Henning and Ohlberger 2010). Further developments for elliptic problems include coupling of finite element macroscopic methods with spectral methods (Abdulle and Engquist 2007), with reduced basis methods (Abdulle and Bai 2012), FE-HMM with special quadrature formulas (Du and Ming 2010), and discontinuous Galerkin FE-HMM (Abdulle 2008, Abdulle 2012).

Various numerical methods have been derived in the past few years for multiscale partial differential equations, mostly for elliptic problems. We do not attempt to review the literature on the subject, and just give a few references. Numerical approaches for homogenization problems were pioneered by Babuška (1976) for static problems and Engquist (1987) for dynamic problems. For multiscale elliptic problems, an important early development is the generalized finite element method proposed by Babuška and Osborn (1983), who developed the idea of adapting the finite element space to the particular fine-scale features of the problem. Dorobantu and Engquist

(1998) and Engquist and Runborg (2002) proposed a method based on multiresolution analysis, Neuss, Jäger and Wittum (2001) combined the multigrid method with homogenization in the coarsening process, Hou, Wu and Cai (1999) proposed the multiscale finite element method (MsFEM) based on modified basis functions obtained from the fine-scale equations (see Efendiev and Hou (2009) for a review), Matache, Babuška and Schwab (2000) and Matache and Schwab (2002) developed the two-scale finite element method, and Viet Ha Hoang and Schwab (2005) proposed the high-dimensional finite element method. We also mention the huge literature in the structural mechanics and engineering communities concerned with micro–macro methods based on representative volume elements (RVEs). Such methods have been proposed for various types of problems, but often without convergence analysis. We mention the methods of Terada and Kikuchi (2001), Miehe, Schröder and Bayreuther (2002) and Geers, Kouznetsova and Brekelmans (2010).

4.1. General methodology

We start by explaining the methodology of the FE-HMM. We consider a general multiscale problem of the form

$$L^\varepsilon(u^\varepsilon, a^\varepsilon) = f \quad \text{in } \Omega,$$

where L^ε is a differential operator, a^ε denotes the data of the problem, Ω is an open bounded subset of $\mathbb{R}^d$, $f : \Omega \to \mathbb{R}$ is given and $u^\varepsilon : \bar{\Omega} \to \mathbb{R}$ is the solution of the above problem for which appropriate boundary conditions are specified. To emphasize the multiscale nature of the data of the above problem, we put a superscript ε (representing the typical size of a small scale in the considered problem) on a, L and u. Here, for simplicity, in view of numerical discretization, we assume that $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ is a polygonal

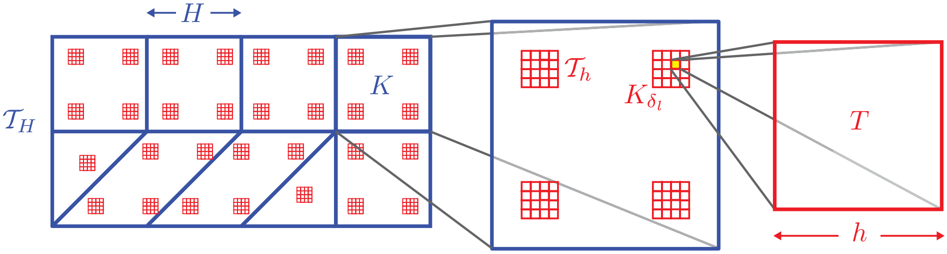


Figure 4.1. Sketch of the FE-HMM algorithm: macro triangulation $\mathcal{T}_H$ (quadrilateral and simplicial elements) with sampling domains within the macro elements. A close-up of the sampling domains shows the micro triangulation $\mathcal{T}_h$, chosen here to be quadrilaterals.

domain. The effective problem, assumed to exist, is of the form

$$L^0(u^0, a^0) = f \text{ in } \Omega,$$

where $u^0 : \bar{\Omega} \to \mathbb{R}$ is the effective solution. One can think of u^0 as being the limit (in an appropriate sense) of the solution u^ε as $\varepsilon \to 0$. The weak solutions of any of the above problems are assumed to belong to an appropriate Sobolev space, denoted by $H(\Omega)$. The FE-HMM can be summarized by the following steps.

- *Macroscale solver.*
  - *Macro triangulation.* Define a macroscopic partition $\Omega = \bigcup_{K \in \mathcal{T}_H} K$ and a macroscopic finite-dimensional subspace $S(\Omega, \mathcal{T}_H)$ of $H(\Omega)$.
  - *Macro method.* Let $u_H \in S(\Omega, \mathcal{T}_H)$ be the solution of $L_H(u_H) = f$, where L_H is an unknown approximation of the effective differential operator L^0.
- *Microscale solver.*
  - *Constrained micro simulations.* For every $K \in \mathcal{T}_H$, consider a suitable *quadrature formula* $\{x_{K_j}, \omega_{K_j}\}_{j=1}^J$, and *sampling domains* $K_{\delta_j} = x_{K_j} + \delta I$, $I = (-1/2, 1/2)^d$ $(\delta \geq \varepsilon)$. Define a micro triangulation $\bigcup_{T \in \mathcal{T}_h} T = K_{\delta_j}$ and a micro finite-dimensional space $S(K_{\delta_j}, \mathcal{T}_h)$. Compute micro solutions $u_h \in S(K_{\delta_j}, \mathcal{T}_h)$ constrained by the macro variable using the original fine-scale operator.
  - *Data processing.* Recover $L_H(u_H)|_K$ locally by a suitable average of u_h in K_{δ_j}.

We notice that the method described here could be used to solve problems with a right-hand side depending on ε (*e.g.*, f^ε) with minor changes (see Abdulle and Schwab (2005)).

4.2. Model problem

To describe the method, we consider $L^\varepsilon(u^\varepsilon, a^\varepsilon) = -\nabla \cdot (a^\varepsilon \nabla u^\varepsilon)$, that is, the linear elliptic homogenization problem seen in Section 2,

$$-\nabla \cdot (a^\varepsilon \nabla u^\varepsilon) = f \text{ in } \Omega, \quad u^\varepsilon = 0 \text{ on } \partial\Omega, \tag{4.1}$$

where the family of tensors $a^\varepsilon(x) \in (L^\infty(\Omega))^{d \times d}$ (indexed by ε) is assumed to be uniformly elliptic and bounded, that is,

$$\exists \lambda, \Lambda > 0 \quad \text{such that} \quad \lambda|\xi|^2 \leq a^\varepsilon(x)\xi \cdot \xi \quad \text{and} \quad |a^\varepsilon(x)\xi| \leq \Lambda|\xi|, \tag{4.2}$$
$$\forall \xi \in \mathbb{R}^d, \quad \forall \varepsilon, \quad \text{a.e. } x \in \Omega,$$

where ε is a microscopic scale that characterizes the multiscale nature of the tensor $a^\varepsilon(x)$. Here we take zero Dirichlet boundary conditions for simplicity.[1] We also assume that $f \in L^2(\Omega)$ ($f \in H^{-1}(\Omega)$ would also be possible). By the Lax–Milgram theorem, owing to the uniform ellipticity and boundedness of the tensor a^ε, the weak form of (4.1) possesses for each ε a unique solution, and we can thus consider a family of solutions $\{u^\varepsilon\}$ (indexed by ε) which are bounded in $H_0^1(\Omega)$ by the same constant.

Homogenization results. The goal of homogenization theory is to find an 'averaged equation' corresponding to (4.1). Using H-convergence theory (Murat and Tartar 1997) – or G-convergence theory in the symmetric case (De Giorgi and Spagnolo 1973) – and without further assumptions on the heterogeneities of the tensor $a^\varepsilon(x)$, it is possible to show that there exists a tensor $a^0(x)$ (again uniformly elliptic and bounded) and a subsequence of $\{u^\varepsilon\}$ which weakly converges in $H_0^1(\Omega)$ to a function $u^0 \in H_0^1(\Omega)$, such that

$$-\nabla \cdot (a^0 \nabla u^0) = f \quad \text{in } \Omega, \qquad u^0 = 0 \quad \text{on } \partial\Omega. \tag{4.3}$$

new display

Returning to the general description of Section 4.1, we see that

$$L^0(u^0, a^0) = -\nabla \cdot (a^0 \nabla u^0).$$

If $a^\varepsilon(x)$ has a more specific structure, for example if $a^\varepsilon(x) = a(x, x/\varepsilon)$ and is periodic in its second argument, then classical results show that the whole sequence $\{u^\varepsilon\}$ weakly converges to $u^0 \in H_0^1(\Omega)$, and the homogenized tensor $a^0(x)$ at $x \in \Omega$ can be characterized as the average of the solutions of 'cell problems' (*i.e.*, d boundary value problems on a domain of size ε^d involving the fine-scale tensor $a^\varepsilon(x) = a(x, x/\varepsilon)$). For details we refer, for example, to Bensoussan, Lions and Papanicolaou (1978), Jikov, Kozlov and Oleinik (1994) and Cioranescu and Donato (1999). Notice that even in this fortunate case, we generally still have an infinite number of cell problems to solve, due to the slow variation in the tensor. Classical numerical approaches consist of pre-computing an approximation of the effective tensor $a^0(x_i)$ at predefined sampling points $x_i \in \Omega$, $i = 1, \ldots, p$, and using a standard finite element method based on these pre-computed data. Although straightforward, this sequential strategy does not allow for a natural way to derive *a priori* or *a posteriori* error control of the overall discretization (as it depends on the accuracy of the numerically pre-computed homogenized tensors). Further, this procedure does not give a straightforward discretization for non-periodic or non-linear problems, and makes it difficult to switch locally to a fine-scale approximation (desirable in some applications).

[1] The algorithm and results stated in this section are valid for other boundary conditions (non-zero Dirichlet, Neumann, mixed, *etc.*) with obvious changes.

4.3. FE-HMM: algorithm

In this section we give a detailed description of the FE-HMM. We restrict ourselves to the model problem (4.1). In Section 4.3.3, generalizations to more involved problems (*e.g.*, non-linear problems) will be described.

4.3.1. Macroscale solver

Let $\mathcal{T}_H$ denote a family of (macro) partitions[2] of Ω in simplicial or rectangular elements K. The diameter of an element $K \in \mathcal{T}_H$ is denoted by H_K and we set $H = \max_{K\in\mathcal{T}_H} H_K$. We then consider a macro finite element space

$$S_0^\ell(\Omega, \mathcal{T}_H) = \{v_H \in H_0^1(\Omega) : v_H|_K \in \mathcal{R}^\ell(K),\ \forall K \in \mathcal{T}_H\}, \tag{4.4}$$

where $\mathcal{R}^\ell(K)$ is the space $\mathcal{P}^\ell(K)$ of polynomials on K of total degree at most ℓ, if K is a simplicial finite element, or the space $\mathcal{Q}^\ell(K)$ of polynomials on K of degree at most ℓ in each variable, if K is a rectangular finite element. For every K we consider the quadrature formula $\{x_{K_j}, \omega_{K_j}\}_{j=1}^J$ (described further in Section 4.3.2), and sampling domains K_{δ_j}, defined as $K_{\delta_j} = x_{K_j} + \delta I,\ I = (-1/2, 1/2)^d\ \ (\delta \geq \varepsilon)$.

Multiscale method. Find $u_H \in S_0^\ell(\Omega, \mathcal{T}_H)$ such that

$$B_H(u_H, v_H) = \int_\Omega f v_H \,\mathrm{d}x, \quad \forall v_H \in S_0^\ell(\Omega, \mathcal{T}_H), \tag{4.5}$$

where for arbitrary $v_H, w_H \in S_0^\ell(\Omega, \mathcal{T}_H)$ the bilinear form $B_H(\cdot,\cdot)$ is defined by

$$B_H(v_H, w_H) := \sum_{K\in\mathcal{T}_H} \sum_{j=1}^J \frac{\omega_{K_j}}{|K_{\delta_j}|} \int_{K_{\delta_j}} a^\varepsilon(x) \nabla v_{h,K_j} \cdot \nabla w_{h,K_j} \,\mathrm{d}x, \tag{4.6}$$

where $v_{h,K_j} = R_{K_j}(v_H)$, $w_{h,K_j} = R_{K_j}(w_H)$ and $R_{K_j} = R|_{K_j}$ is the reconstruction operator (in the terminology of Section 2) restricted to the sampling domain K_{δ_j}. In the notation of Section 4.1, we have

$$L_H :\ S_0^\ell(\Omega, \mathcal{T}_H) \longrightarrow L^2(\Omega),$$

where

$$(L_H(v_H), w_H) = B_H(v_H, w_H),$$

and $(\cdot,\cdot)$ is the L^2 inner product. In order to assemble the bilinear form $B_H(v_H, w_H)$ we need to compute the functions $R_{K_j}(v_H), R_{K_j}(w_H)$, as described in the next subsection.

[2] By macro partition we mean that $H \gg \varepsilon$ is allowed.

4.3.2. Microscale solver

The location of the sampling domains K_{δ_j} and the choice of the weights ω_{K_j} $(j = 1, \ldots, J)$ in the definition of $B_H(\cdot, \cdot)$ rely on the definition of appropriate quadrature formulas. We consider $\hat{K}$ to be the reference element, and for every element K of the triangulation we let F_K denote the $\mathcal{C}^1$-diffeomorphism such that $K = F_K(\hat{K})$. For every K we consider the quadrature formula $x_{K_j} = F_K(\hat{x}_j)$, $\omega_{K_j} = \hat{\omega}_j|\det(\partial F_K)|$, $j = 1, \ldots, J$, where $\{\hat{x}_j, \hat{\omega}_j\}_{j=1}^J$ is a quadrature formula on $\hat{K}$. We then consider a (micro) partition $\mathcal{T}_h$ of each sampling domain K_{δ_j} in simplicial or rectangular elements T and define a micro finite element space

$$S^q(K_{\delta_j}, \mathcal{T}_h) = \{z_h \in W(K_{\delta_j}) : z_h|_T \in \mathcal{R}^q(T),\ \forall T \in \mathcal{T}_h\}, \tag{4.7}$$

where $h = \max_{T \in \mathcal{T}_h} h_T$ (h_T is the diameter of the element T) and $W(K_{\delta_j})$ is a given Sobolev space.

Coupling. Various spaces $W(K_{\delta_j})$ can be chosen for the micro numerical method, for example

$$W(K_{\delta_j}) = W^1_{\text{per}}(K_{\delta_j}) = \left\{ z \in H^1_{\text{per}}(K_{\delta_j}) : \int_{K_{\delta_j}} z \,\mathrm{d}x = 0 \right\}, \tag{4.8}$$

for a periodic coupling, or

$$W(K_{\delta_j}) = H^1_0(K_{\delta_j}) \tag{4.9}$$

for a coupling via Dirichlet boundary conditions. Other coupling conditions, constraining the averaged gradient of a microscale solution, can be used. We refer to Yue and E (2007) for further discussion.

Micro method. For each $v_H \in S^\ell_0(\Omega, \mathcal{T}_H)$ we define a micro function v_{h,K_j}, satisfying $(v_{h,K_j} - v_{H,\text{lin}}) \in S^q(K_{\delta_j}, \mathcal{T}_h)$, which is a solution of

$$\int_{K_{\delta_j}} a^\varepsilon(x) \nabla v_{h,K_j} \cdot \nabla z_h \,\mathrm{d}x = 0, \quad \forall z_h \in S^q(K_{\delta_j}, \mathcal{T}_h), \tag{4.10}$$

where $v_{H,\text{lin}}|_{K_{\delta_j}} = v_H(x_{K_j}) + (x - x_{K_j}) \cdot \nabla v_H(x_{K_j})$ is the linearization of v_H at the quadrature point x_{K_j}. As explained in Section 4.4.1, the bilinear form (4.6) can be rewritten as

$$B_H(v_H, w_H) = \sum_{K \in \mathcal{T}_H} \sum_{j=1}^J \omega_{K_j} a^0_K(x_{K_j}) \nabla v_H(x_{K_j}) \cdot \nabla w_H(x_{K_j}),$$

where $a^0_K(x_{K_j})$ is the macroscale data (effective tensor) at the quadrature point x_{K_j} recovered by the microscale simulations. This is the data processing step mentioned in Section 4.1, which is implicitly computed by defining the bilinear form via the quantities

$$\frac{1}{|K_{\delta_j}|}\int_{K_{\delta_j}} a^{\varepsilon}(x)\nabla v_{h,K_j}\cdot\nabla w_{h,K_j}\,\mathrm{d}x.$$

Remark 4.1. Conditions on the quadrature formula are needed to ensure that the optimal convergence rates for elliptic finite element methods are obtained when using numerical quadrature. We make the following classical assumptions on the quadrature formula $\{\hat{x}_j,\hat{\omega}_j\}_{j=1}^J$ on the reference element $\hat{K}$ (see Ciarlet and Raviart (1972)):

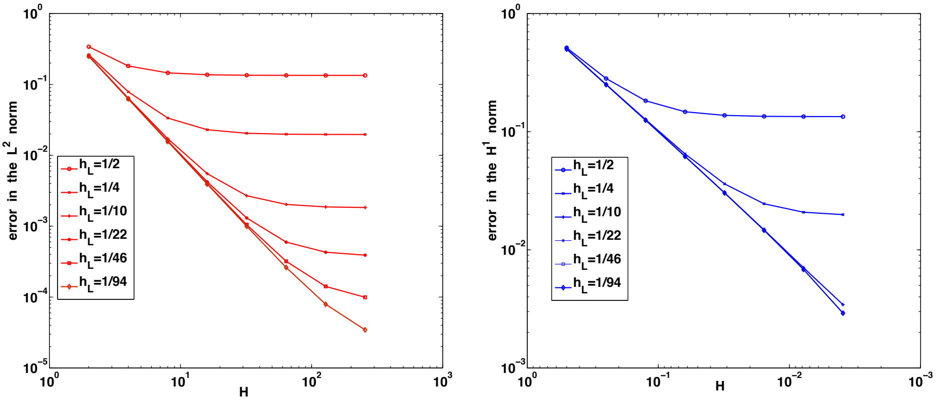
(**Q1**) $\hat{\omega}_j>0,\ j=1,\dots,J,\ \sum_{j=1}^J\hat{\omega}_j|\nabla\hat{p}(\hat{x}_j)|^2\geq\hat{\lambda}\|\nabla\hat{p}\|^2_{L^2(\hat{K})}$, for all $\hat{p}(\hat{x})\in\mathcal{R}^{\ell}(\hat{K})$, $\hat{\lambda}>0$,

(**Q2**) $\int_{\hat{K}}\hat{p}(\hat{x})\,\mathrm{d}\hat{x}=\sum_{j=1}^J\hat{\omega}_j\hat{p}(\hat{x}_j)$, for all $\hat{p}(\hat{x})\in\mathcal{R}^{\sigma}(\hat{K})$, for which $\sigma=\max(2\ell-2,\ell)$ if $\hat{K}$ is a simplicial finite element, or $\sigma=\max(2\ell-1,\ell+1)$ if $\hat{K}$ is a rectangular finite element.

4.3.3. Implementation and numerical illustration

The macro–micro methodology described in Section 4.3 allows easy implementation and design of a code whose structure follows the classical finite element implementation at the macro level. In particular, elementwise assembly for each macro element K can be computed to find the additive contribution from (4.6) to the macroscopic stiffness matrix. Abdulle and Nonnenmacher (2009*b*) propose a short (less than 200 lines) and flexible MATLAB implementation, capable of handling two- and three-dimensional elliptic and parabolic problems. All the numerical experiments presented in these subsections have been made with the code presented by Abdulle and Nonnenmacher (2009*b*) or a variation of it (*e.g.*, Abdulle and Vilmart (2011*a*)). These public-domain codes are available at http://anmc.epfl.ch. Additional numerical experiments can be found in Abdulle (2009) and Ming and Yue (2006).

Macro–micro mesh refinement. Some care is needed in the choice of the macroscopic mesh $\mathcal{T}_H$ (a triangulation of the physical domain Ω) and the microscopic mesh $\mathcal{T}_h$ (a triangulation of the sampling domain K_{δ_j}). Taking N_{mic} elements in each space dimension for the discretization of the sampling domain K_{δ_j}, we have $h=\delta/N_{\mathrm{mic}}$ and thus $\hat{h}:=h/\varepsilon=(\delta/\varepsilon)\cdot(1/N_{\mathrm{mic}})$. Since δ scales with ε, typically $\delta=C\varepsilon$ (where C is a constant of moderate size), we have $\hat{h}=C/N_{\mathrm{mic}}$. We let $M_{\mathrm{mic}}=O(\hat{h}^{-d})$ denote the number of degrees of freedom for the micro finite element method and let M_{mac} denote the number of degrees of freedom for the macro finite element method. For quasi-uniform macro meshes, the macro mesh size H (sometimes denoted



(a) (b)

Figure 4.2. Error with respect to the homogenized solution u^0 for the FE-HMM applied to a two-dimensional periodic problem: (a) L^2 error, (b) H^1 error. The horizontal axis represents the macro mesh size (decreasing from left to right). Each curve represents the error obtained by macro mesh refinement for fixed micro meshes $h/\varepsilon = 1/2, 1/4, 1/10, 1/22, 1/46, 1/94$. Horizontal curves indicate a dominant micro error.

$H = 1/N_{\text{mac}}$) and the micro mesh size $\hat{h}$ are related to M_{mac} and M_{mic} by

$$H = O(M_{\text{mac}}^{-1/d}), \quad \hat{h} = O(M_{\text{mic}}^{-1/d}).$$

A priori error analysis for the fully discrete method (4.5), first given in Abdulle (2005*b*), provides an insight into this meshing problem, and reveals that the optimal refinement of the micro mesh is given by

$$\hat{h} := \frac{h}{\varepsilon} \simeq H^{\frac{\ell}{2q}} \quad (H^1 \text{ norm}), \quad \hat{h} := \frac{h}{\varepsilon} \simeq H^{\frac{\ell+1}{2q}} \quad (L^2 \text{ norm}).$$

The corresponding complexity, in terms of macro degrees of freedom, satisfies

$$\underbrace{H^{-d}}_{M_{\text{mac}}} \cdot \underbrace{H^{\frac{-d\ell}{2q}}}_{M_{\text{mic}}} \cdot n_s = (M_{\text{mac}})^{1+\frac{\ell}{2q}} \cdot n_s \qquad \text{for the } H^1 \text{ norm},$$

$$\underbrace{H^{-d}}_{M_{\text{mac}}} \cdot \underbrace{H^{\frac{-d(\ell+1)}{2q}}}_{M_{\text{mic}}} \cdot n_s = (M_{\text{mac}})^{1+\frac{\ell+1}{2q}} \cdot n_s \quad \text{for the } L^2 \text{ norm},$$

where n_s denotes the number of sampling domains per macro element $K \in \mathcal{T}_H$.[3] For example, using piecewise linear polynomials on simplicial finite elements, assuming quasi-uniform macro and micro meshes, and that the

[3] Notice that as the micro problems are solved independently, the method is well suited to parallel implementation, which can reduce significantly the complexity of the FE-HMM.

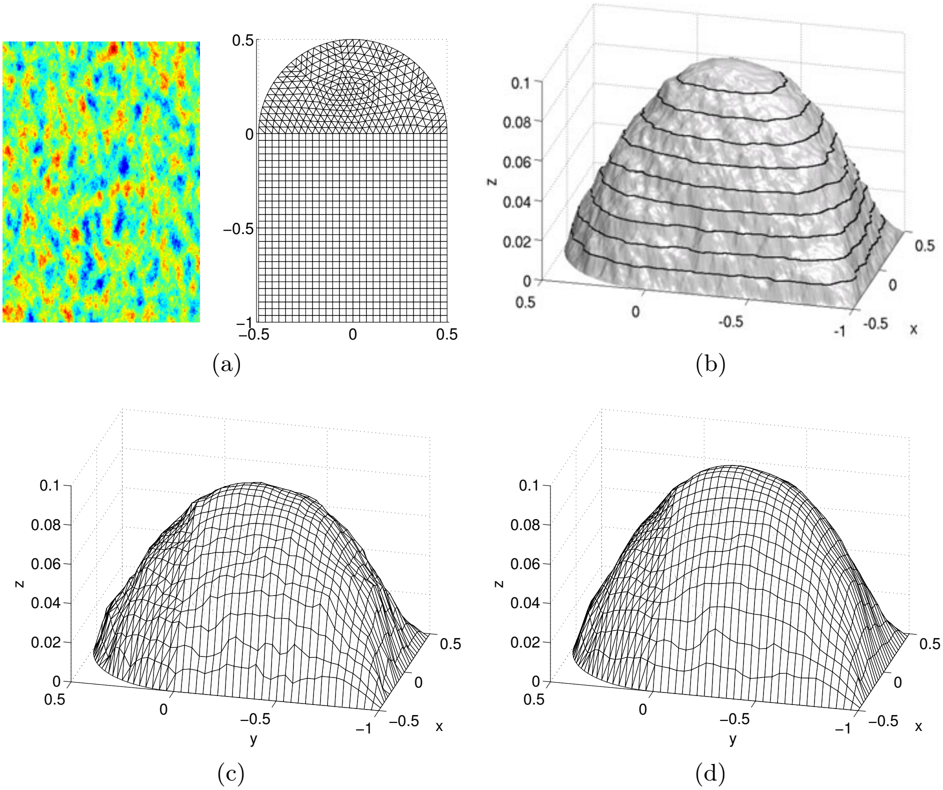
complexity is proportional to the total degrees of freedom, we obtain a cost of $O(M_{\mathrm{mac}}^{3/2})$ (H^1 norm) and $O(M_{\mathrm{mac}}^2)$ (L^2 norm). These convergence rates are illustrated in Figure 4.2 on a very simple multiscale problem, for which reference solutions can easily be computed to high precision (see Abdulle (2005*b*)). In contrast, the memory demand is proportional to only $M_{\mathrm{mac}} + M_{\mathrm{mic}}$, as the micro problems, being independent, can be solved one at a time. Finally, we note that by using spectral methods or high-order finite element methods for the micro solvers, it is possible to reduce the total cost of the FE-HMM to log-linear complexity in the macro degrees of freedom. This was investigated by Abdulle and Engquist (2007). Such an approach, however, requires high regularity of the oscillating tensor a^ε, which may not hold for some applications, for example in materials science.

Example 1: Homogenization problem with non-periodic tensor. We consider problem (4.1), with a log-normal conductivity tensor a^ε (taken from Abdulle and Nonnenmacher (2009*b*)). The domain Ω consists of a semicircle and a rectangle, meshed with 1137 nodes using 576 triangles and 784 quadrilaterals, respectively: see Figure 4.3(a). We use the moving ellipse average method (Wallstrom *et al.* 1999, Section 4.1) to generate the realization of the log-normal stochastic field with mean zero and variance $\sigma = 1$. The correlation lengths of the stochastic field are set to be $\varepsilon_1 = 0.01$ and $\varepsilon_2 = 0.02$ in the x, y direction, respectively. A snapshot of this tensor is shown in Figure 4.3(a).

As this problem does not have an explicit analytical solution, we compute a fine-scale solution using a standard finite element method with a fine mesh of about 10^6 degrees of freedom in order to resolve the microscale for a given realization: see Figure 4.3(b). We compare this reference solution with the FE-HMM on the coarse macro grid with about 1100 degrees of freedom for the same realization. For the FE-HMM we present results for various sizes of the sampling domains ($\delta = 0.02$ and $\delta = 0.06$). We can see in Figures 4.3(c) and 4.3(d) that the FE-HMM solution is closer to the reference solution, as the sampling domain contains more correlation lengths of the random field.

Table 4.1. Energy norms of the FE-HMM and fine-scale solutions of the problem described in Example 1 with a random tensor. FE-HMM results are given for various micro sampling-domain sizes $\delta \times \delta$ and degrees of freedom N_{mic}^2 of the micro problems ($N_{\mathrm{mic}} = 1/\hat{h}$).

N_{mic}	4	8	16	32	64	Fine-scale
$\delta = 0.02$	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) (b)

(c) (d)

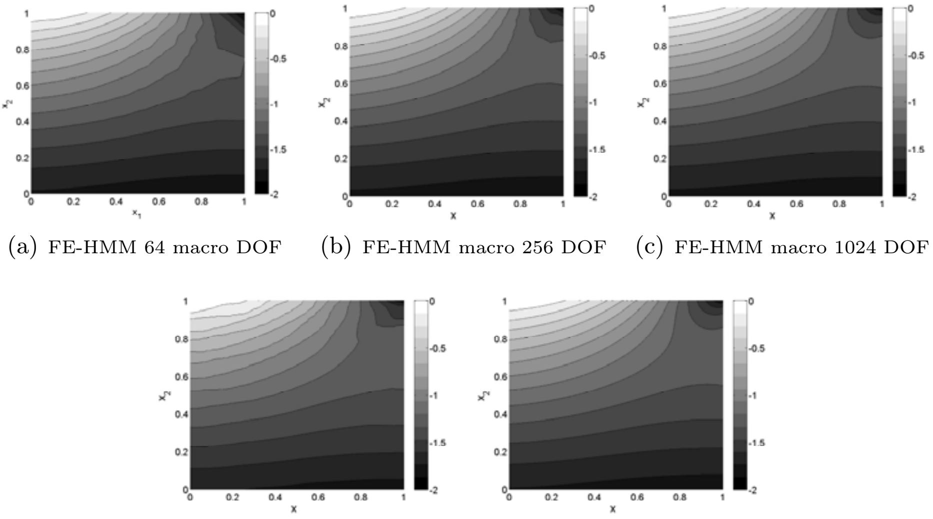
Figure 4.3. FE-HMM and fine-scale solution of the random problem described in Example 1. (a) Snapshot of the random tensor and computational domain. (b) Fine-scale solution computed with a standard finite element method on grid with 10^6 degrees of freedom. (c) FE-HMM with $N_{\text{mic}} = 64$ and micro domain size $\delta = 0.02$ on a macro grid with 1100 degrees of freedom. (d) FE-HMM with $N_{\text{mic}} = 64$ and micro domain size $\delta = 0.06$ on a macro grid with 1100 degrees of freedom.

This observation can also be seen in Table 4.1 when comparing the energy norm of the various solutions obtained with the FE-HMM to the energy norm of the reference solution.

Example 2: A non-linear problem, the Richards equation. We consider the steady-state Richards equation, which describes the fluid pressure $u(x,t)$ in an unsaturated porous medium,

$$-\nabla \cdot \big(a^\varepsilon(u^\varepsilon(x))\nabla(u^\varepsilon(x) - x_2)\big) = f(x) \quad \text{in } \Omega = (0,1)^2, \tag{4.11}$$

where x_2 is the vertical coordinate, and f corresponds to possible sources or sinks. Here a^ε is a multiscale permeability tensor that depends on the pressure u^ε, hence equation (4.11) is a non-linear non-monotone problem.



(a) FE-HMM 64 macro DOF (b) FE-HMM macro 256 DOF (c) FE-HMM macro 1024 DOF

(d) FEM 1024 DOF (e) resolved FEM 1.5×10^6 DOF

Figure 4.4. The Richards problem (4.11) with mixed boundary conditions (Dirichlet and Neumann). (a–c) Level curves of the FE-HMM solutions with $N_{\text{mic}} = N_{\text{mac}}$. The macro degrees of freedom are indicated. (d, e) Level curves of a standard finite element method with respectively unresolved (1024 degrees of freedom) and resolved (1.5×10^6 degrees of freedom) meshes for the given microscale.

The application of the FE-HMM to such problems was first considered by E *et al.* (2005*b*). A complete analysis has recently been given by Abdulle and Vilmart (2011*b*, 2011*a*), and is summarized in Section 4.4.2. The FE-HMM for this problem reads as follows. Find $u_H \in S_0^\ell(\Omega, \mathcal{T}_H)$ such that

$$B_H(u_H; u_H, w_H) = F_H(w_H), \ \forall w_H \in S_0^\ell(\Omega, \mathcal{T}_H), \tag{4.12}$$

where

$$\begin{aligned} &B_H(u_H; v_H, w_H) \qquad\qquad (4.13)\\ &\quad := \sum_{K \in \mathcal{T}_H} \sum_{j=1}^{J} \frac{\omega_{K_j}}{|K_{\delta_j}|} \int_{K_{\delta_j}} a^\varepsilon(x, u_H(x_{K_j})) \nabla v_{h,K_j}^{u_H(x_{K_j})}(x) \cdot \nabla w_{h,K_j}^{u_H(x_{K_j})}(x) \,\mathrm{d}x, \end{aligned}$$

and $v_{h,K_j}^{u_H(x_{K_j})}$ (and similarly $w_{h,K_j}^{u_H(x_{K_j})}$) is a solution of a linear micro problem similar to (4.10). For the parameter $s = u_H(x_{K_j})$ it reads: find v_{h,K_j}^s such that $(v_{h,K_j}^s - v_{H,\text{lin}}) \in S^q(K_{\delta_j}, \mathcal{T}_h)$ and

$$\int_{K_{\delta_j}} a^\varepsilon(x, s) \nabla v_{h,K_j}^s(x) \cdot \nabla z_h(x) \,\mathrm{d}x = 0, \quad \forall z_h \in S^q(K_{\delta_j}, \mathcal{T}_h). \tag{4.14}$$

We observe that while the macro problem (4.12) is non-linear, the micro problems (4.14) are linear. Indeed, in the implementation $s = u_H(x_{K_j})$ is given by the current (available) state of the macro solution (*e.g.*, using a Newton method to solve the macro problem). For the numerical simulation of (4.11) we set $f(x) \equiv 0$ for simplicity, and consider the boundary conditions

$$\begin{aligned} u^\varepsilon(x) = -1.9x_1^2 \quad &\text{on } \partial\Omega_D = [0,1] \times \{1\}, \\ n \cdot (a^\varepsilon(u^\varepsilon(x))\nabla(u^\varepsilon(x) - x_2)) = 0 \quad &\text{on } \partial\Omega_N = \{0,1\} \times [0,1] \cup [0,1] \times \{0\}, \end{aligned}$$

and an exponential model for the permeability tensor a^ε, similar to that in Chen, Deng and Ye (2005*b*, Section 5.1),

$$a^\varepsilon(x,s) = \alpha^\varepsilon(x)\,\mathrm{e}^{\alpha^\varepsilon(x)s} \quad \text{where } \alpha^\varepsilon(x) = \frac{1/117.4}{2 + 1.8\sin(2\pi(2x_2/\varepsilon - x_1/\varepsilon))}. \tag{4.15}$$

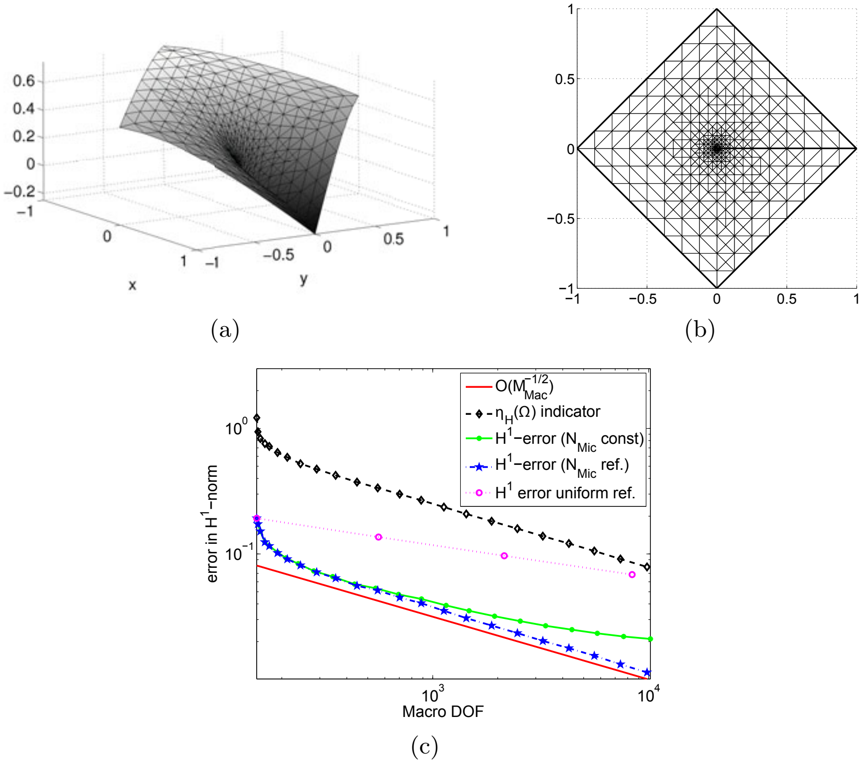
Numerical solutions are shown in Figure 4.4. Figures 4.4(a)–4.4(c) are solutions of the FE-HMM with $\mathcal{P}^1$-triangular finite elements and a decreasing macro mesh size (uniform macro meshes with $H = 1/8, 1/16, 1/32$). The meshes of the micro solver are refined according to the 'optimal L^2 refinement strategy' $h/\varepsilon \simeq H$. The numerical results obtained with the FE-HMM are compared to a reference solution of the problem (4.11) (obtained by a resolved standard finite element method) plotted in Figure 4.4(e) (for $\varepsilon = 10^{-2}$, $\sim 10^6$ degrees of freedom are used). For comparison, we also plot in Figure 4.4(d) the result obtained by a standard finite element method on a coarse 32×32 mesh that does not resolve the fine oscillations. We observe that this unresolved finite element method does not give a correct qualitative result. In contrast, the FE-HMM captures the correct behaviour of the problem at a much lower computational cost.

Example 3: A crack problem and adaptive FE-HMM. In our next experiment we consider a crack problem in a heterogeneous medium, characterized by a highly oscillating conductivity tensor,

$$-\nabla \cdot (a^\varepsilon(x)\nabla u^\varepsilon) = 1 \quad \text{in } \Omega, \tag{4.16}$$

$$u^\varepsilon = g_D \quad \text{on } \Gamma_D = \partial\Omega, \tag{4.17}$$

on a domain $\Omega = \{|x| + |y| < 1\}\backslash\{0 \le x \le 1,\ y = 0\}$ with a crack along the positive x-axis (see Figure 4.5). As the solution of the homogenized problem is not in $H^2(\Omega)$, it is well known that a standard piecewise linear finite element method for the effective problem will give sub-optimal convergence rates. In this situation, adaptive mesh refinement is needed. By using local error indicators, one identifies, marks and refines those elements that contribute the most to the global error in order to better equidistribute the error in the finite element mesh. Overall, the procedure consists of the



(a) (b)

(c)

Figure 4.5. (a, b) FE-HMM solution and mesh after 10 iterations for the crack problem described in Example 3, using highly oscillating locally periodic coefficients. (c) Rate of decay for various errors and for the error indicator in the H^1 norm.

following cycle:

$$\text{SOLVE} \longrightarrow \text{ESTIMATE} \longrightarrow \text{MARK} \longrightarrow \text{REFINE}.$$

An energy-based *a posteriori* error analysis for a standard finite element method usually relies on local error indicators η_h comprising two components: the jump residual, which measures the discontinuity of the normal flux across elements interfaces, and the element residual, which measures the accuracy of the solution inside each element. Both quantities depend on the (conductivity) tensor of the elliptic problem. Residual-based adaptive FE-HMMs for the energy norm were proposed and analysed by Abdulle and Nonnenmacher (2009*a*, 2011*a*) and also by Ohlberger (2005), who derived an *a posteriori* error estimate in a 'two-scale norm'. For the FE-HMM, data involved in the usual error indicator are not available beforehand but computed during the integration process. An important ingredient for adaptive FE-HMM is an effective jump based on 'multiscale fluxes', first defined by

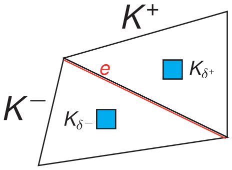


Figure 4.6. Sketch of the ingredients for the multiscale flux jump defined in (4.18).

Abdulle (2008, 2012). Let $\mathcal{T}_H$ denote a conformal mesh and let $\mathcal{E}_H$ be the set of interfaces. Two elements sharing an interface $e \in \mathcal{E}_H$ are labelled K^+ and K^-. Consider the micro functions u_{h,K^+} and u_{h,K^-} satisfying (4.10) in the two sampling domains K_δ^+ and K_δ^- of the elements K^+ and K^-, respectively, constrained by the macro solution u_H of (4.5). The multiscale flux jump is then defined by

$$\llbracket \overline{a^\varepsilon(x)\nabla u_h} \rrbracket_e := \left(\frac{1}{|K_\delta^+|} \int_{K_\delta^+} a^\varepsilon(x)\nabla u_{h,K^+}\,\mathrm{d}x - \frac{1}{|K_\delta^-|} \int_{K_\delta^-} a^\varepsilon(x)\nabla u_{h,K^-}\,\mathrm{d}x \right) \cdot n_e, \tag{4.18}$$

where the unit outward normal n_e is chosen to be $n_e = n^+$ (see Figure 4.6). We omit the index K_δ for the micro solutions u_h in $\llbracket \overline{a^\varepsilon(x)\nabla u_h} \rrbracket_e$, as the jump over e involves two sampling domains in adjacent elements. We also assume for simplicity that (4.5) is solved using a piecewise linear macro finite element space (see Abdulle and Nonnenmacher (2011*b*) and Nonnenmacher (2011) for a generalization of the 'multiscale flux jump' and the *a posteriori* error analysis for higher-order macro finite elements). Then, the local error indicator $\eta_H(K)$ on an element K is defined by (Abdulle and Nonnenmacher 2009*a*, 2011*a*)

$$\eta_H(K)^2 := H_K^2 \|f_H\|^2_{L^2(K)} + \frac{1}{2} \sum_{e \subset \partial K} H_e \| \llbracket \overline{a^\varepsilon \nabla u_h} \rrbracket_e \|^2_{L^2(e)}, \tag{4.19}$$

where f_H is a piecewise constant approximation of f. The quantity $\eta_H(K)^2$ is a measure of the local error $\|u^0 - u_H\|^2_{H^1(K)}$, and is used to identify the elements that contribute most to the error. These elements are then marked for refinement. In order to prove the reliability and the efficiency of an *a posteriori* error estimate, Abdulle and Nonnenmacher (2009*a*, 2011*a*) have derived rigorous upper and lower bounds of the error in terms of the residual. For such bounds, one needs to introduce the data approximation error $\xi_H(K)$ on an element K by

$$\xi_H(K)^2 := H_K^2 \|f_H - f\|^2_{L^2(K)} + \|(a_K^0 - a^0(x))\nabla u_H\|^2_{L^2(K)}, \tag{4.20}$$

where $a^0(x)$ is the unknown homogenized tensor of problem (4.3) and a_K^0

Table 4.2. Number of micro problems with various $\hat{h} = h/\varepsilon$ resolutions to obtain an error of $\|e^H\|_{H^1(\Omega)} \leq 0.07$ for the crack problem described in Example 3 (computation with a periodic tensor).

$\hat{h}$	1/8	1/16	1/24	1/32	1/40	1/48
adaptive FE-HMM, 10th iteration	278	218	60	24	28	40
uniform FE-HMM, 4th iteration	–	–	–	16384	–	–

is the effective macro tensor (in the macro element K) that is recovered by the FE-HMM (see Lemma 4.3). In order to control the data approximation error, the quantity $\|(a^0_K - a^0(x))\nabla u_H\|^2_{L^2(K)}$ needs to be quantified. Under suitable assumptions (Abdulle and Nonnenmacher 2009*a*, 2011*a*) (*e.g.*, (**H1**), (**H2**) in Section 4.4), one can prove that

$$\sup_{x\in K} \|(a^0_K - a^0(x))\nabla u_H\|_{L^2(K)} \leq C\biggl(H_K + \left(\frac{h}{\varepsilon}\right)^2\biggr) + e_{\mathrm{MOD}}, \tag{4.21}$$

where e_{MOD} is a quantity independent of H, h. This shows that the micro–macro mesh refinement described at the beginning of Section 4.3.3 should now be performed locally. In turn, this local refinement has an important consequence on the complexity of the FE-HMM. For uniform refinement one needs to refine every sampling domain of each macro element, but in an adaptive mesh refinement strategy the micro mesh will be refined only in the sampling domains of a macro element marked for refinement (at the above rate $\frac{h}{\varepsilon} = \sqrt{H_K}$). All other computations in the sampling domains, that is, all the micro functions computed in a previous iteration, can be re-used in the next iteration in unrefined macro elements. The savings in computational complexity are illustrated in Table 4.2, where we observe that only a small fraction of the micro problems computed via uniform refinement needs to be computed using adaptive and local refinement as described above.

We present in Figure 4.5 the decay of the error after several iterations of the adaptive FE-HMM applied to the crack problem of Example 3. In order to have an exact homogenized solution for comparison purposes, we choose a periodic tensor in our computation (details can be found in Abdulle and Nonnenmacher (2011*a*, Section 6.2)). An example with random tensors can also be found in Abdulle and Nonnenmacher (2011*a*) and Nonnenmacher (2011). The rate of convergence of the error and the error indicator, reported in Figure 4.5, confirm the theoretical rate of $O(M_{\mathrm{mac}}^{-1/d})$, where we recall that M_{mac} denotes the macro degrees of freedom. We also see that the local error indicator η_H decays with the right slope and the effectivity index $\eta_H(\Omega)/\|u^0 - u_H\|_{H^1(\Omega)}$ is comparable to the effectivity index for

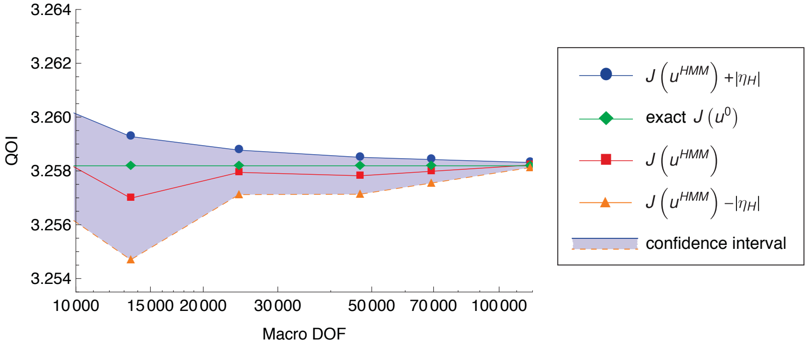


Figure 4.7. Illustration of the goal-oriented FE-HMM, which allows us to specify a confidence interval, the shaded area, where the exact quantity of interest (here a regularized pointwise derivative) is expected to be.

a single-scale residual-based adaptive finite element method (Eriksson, Estep, Hansbo and Johnson 1995). We also illustrate in Figure 4.5 the use of uniform refinement and adaptive FE-HMM without micro refinement. Both strategies give the wrong convergence rate for this problem.

Finally, we briefly mention another type of adaptivity, often more interesting for engineers, namely adaptivity in quantities of interest. Suppose we are interested in a quantity of interest of the exact solution $J(u^0)$, where J denotes a linear bounded functional and u^0 is the homogenized solution of a given multiscale problem (*e.g.*, (4.1)). The question now becomes: Is it possible to refine the numerical solution of a multiscale method, say the FE-HMM, to have an approximation of the quantity of interest $J(u_H) \simeq J(u^0)$?

For single-scale problems, such 'goal-oriented adaptivity' has been studied in Prudhomme and Oden (1999), Oden and Prudhomme (2001), Becker and Rannacher (2001), Nochetto, Veeser and Verani (2009) and Ainsworth and Rankin (2012). We also mention the related work of Oden, Prudhomme, Romkes and Bauman (2006) on adaptive control of the model. In the context of numerical homogenization, the numerical analysis literature on the topic seems rather scarce. Abdulle and Nonnenmacher (2011*b*) discuss *a posteriori* error estimates in quantities of interest for the FE-HMM. The quantity $J(u^0-u_H)$ is shown to have an exact representation in terms of the local error estimator and the data approximation error. Provided that certain higher-order approximation terms can be neglected (see Nochetto *et al.* (2009) or Ainsworth and Rankin (2012) for a discussion of this issue), it is possible to find a 'confidence interval' for the estimation of the error in the given quantity of interest. This is illustrated in Figure 4.7, where the quantity of interest is a regularized pointwise derivative of an effective solution corresponding to a multiscale elliptic problem. The solution u_H is obtained from the FE-HMM with goal-oriented adaptive refinement. We refer to Abdulle and Nonnenmacher (2011*b*) for details.

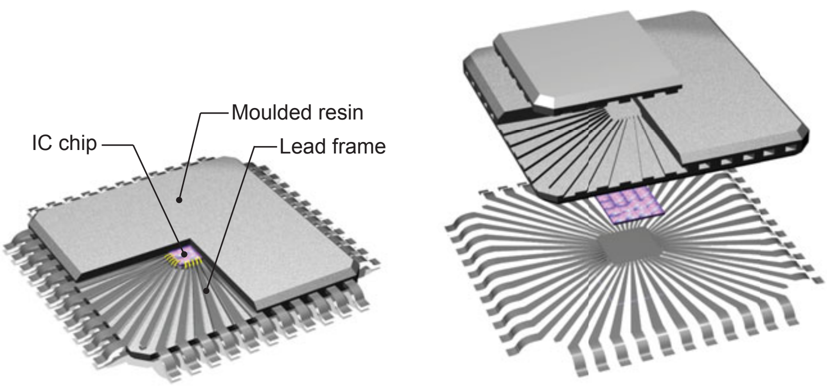


Figure 4.8. Microprocessor model used for the numerical computation of Example 4.

Example 4: Three-dimensional problems. In our last example, we consider the heat distribution in a microprocessor. The model considered in our computations, depicted in Figure 4.8, consists of a silicon IC chip, a lead frame, which acts as a heat spreader, and a moulded resin encapsulation (the packaging covering the IC chip for protection). Due to the increasing packaging density in such devices and the resulting temperature growth, the use of composite materials with high thermal conductivity is crucial. The micro structure of the composite materials, whose representative size is denoted by ε, is responsible for the multiscale behaviour of the associated heat conductivity problem. The application of the FE-HMM to this example was first considered by Abdulle and Nonnenmacher (2009*b*), who also discuss additional three-dimensional examples (see also Nonnenmacher (2011)).

Geometry and mesh. The volume of the full processor model is $12.2 \times 12.2 \times 1$ mm^3. We use a macro mesh generated by CUBIT Sandia National Laboratories (1997-2010) that consists of 81 000 grid points with a maximum tetrahedron volume of 1.4×10^{-3} mm^3 (see Figure 4.9). Scale resolution for a composite with $\varepsilon = 10^{-6}$ m as used below, with about 10 grid points per oscillation (in each spatial direction), would result in a mesh with 10^{12} grid points and is thus computationally unfeasible with a standard finite element method. In contrast, numerical experiments with realistic values of ε can be computed without difficulty with the FE-HMM. For numerical comparisons, we also present numerical experiments with a relatively large value of ε, $\varepsilon = 5 \times 10^{-4}$ m. For this value we compute a reference fine-scale solution that involves 3.9 million grid points and about 22 million tetrahedra. The heat transfer by conduction is modelled by (4.1). Convective heat transfer with the surrounding air is modelled by the Robin and Neumann boundary

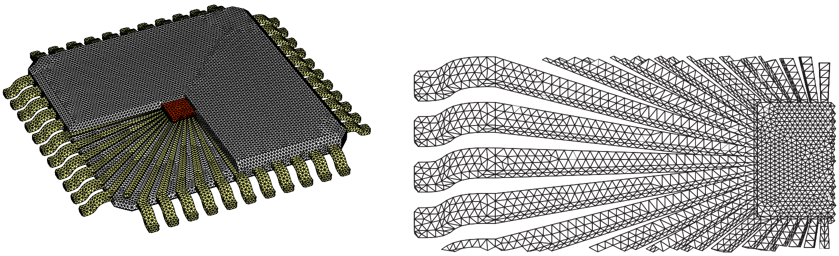


Figure 4.9. (a) Macro mesh of the three-dimensional microprocessor problem of Example 4: 81 000 grid points and 430 000 tetrahedra. (b) Close-up of the macro mesh used for the lead frame.

conditions:

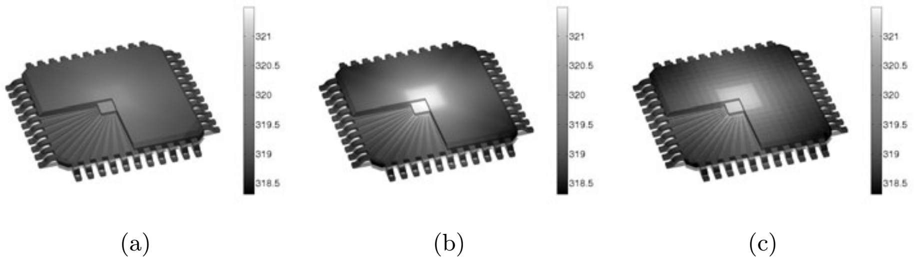
$$n \cdot (a^{\varepsilon} \nabla u^{\varepsilon}) + \alpha u^{\varepsilon} = g_R \quad \text{on } \partial\Omega_R, \tag{4.22}$$

$$n \cdot (a^{\varepsilon} \nabla u^{\varepsilon}) = g_N \quad \text{on } \partial\Omega_N, \tag{4.23}$$

where Ω is the domain of the object considered, and $\partial\Omega_R$ and $\partial\Omega_N$ are the surfaces of the three-dimensional object with Robin and Neumann boundary conditions, respectively. The right-hand side of the Robin boundary condition is given by $g_R = q_0 + \alpha u_{\text{amb}}$.

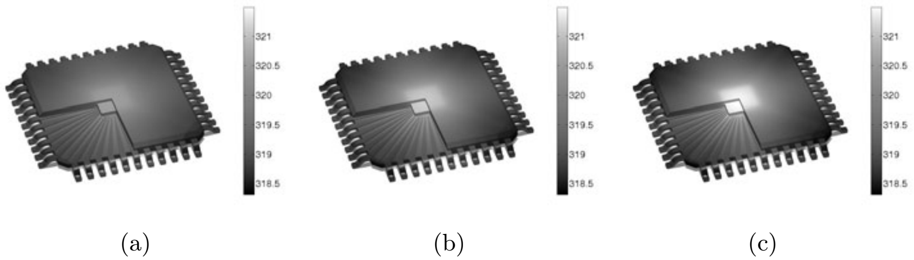
In practice, conductivity tensors for realistic materials could be obtained via imaging techniques (our computational strategy could accommodate such data). Here we use simplified tensors (one for each material). The tensor corresponding to the lead frame $a^{\varepsilon}_{\text{leadframe}}$ is chosen to have a non-periodic slow variation, to mimic a change in the material structure from the centre of the lead frame to the periphery. For the resin, we chose $a^{\varepsilon}_{\text{resin}}$ to be oscillating and anisotropic with a larger conductivity in the z-direction, while the tensor corresponding to the chip $a^{\varepsilon}_{\text{chip}}$ is assumed to be constant (see Abdulle and Nonnenmacher (2009*b*) for details). Other data are defined as follows: the power of the chip is $P_{\text{chip}} = 0.125$ W and the chip size is chosen to be $V = 2 \times 2 \times 0.2$ mm^3. This yields an external heat flux f in (4.1) of $f = P/V = 1.875 \times 10^8$ W m^{-3}. The room temperature is $u_{\text{amb}} = 293.15$ K and the heat transfer coefficient is $\alpha = 20$ W m^{-2} K^{-1}, a rough estimate for the effect of air cooling.

In Figure 4.10 we compare (for $\varepsilon = 5 \times 10^{-4}$ m) the results obtained by the FE-HMM with a fine-scale (resolved) solution. For the FE-HMM, two different resolutions of the micro finite element method are used. Good qualitative agreement between the FE-HMM and the fine-scale solutions is observed. For comparison we also plot a solution obtained on the same macro grid as the FE-HMM, but with a standard finite element method



(a) (b) (c)

Figure 4.10. Steady-state heat distribution in the three-dimensional microprocessor problem of Example 4, with $\varepsilon = 5 \times 10^{-4}$ m, 81 000 degrees of freedom in the macro mesh. Comparison of various solutions. (a) Averaged tensor (arithmetic mean), 81 000 degrees of freedom, (b) FE-HMM solution, 81 000 macro degrees of freedom, (c) fine-scale solution, 3.9×10^6 degrees of freedom. The bar indicates the temperature in Kelvin (K).



(a) (b) (c)

Figure 4.11. Steady-state heat distribution in the three-dimensional microprocessor problem of Example 4, with $\varepsilon = 10^{-6}$ m, 81 000 degrees of freedom in the macro mesh. Comparison of FE-HMM solutions with different resolutions of the micro problems: (a) $N_{\mathrm{mic}} = 4$, (b) $N_{\mathrm{mic}} = 8$, (c) $N_{\mathrm{mic}} = 16$. The bar indicates the temperature in Kelvin (K).

using a naive averaging procedure for the micro structure (here the arithmetic average). We observe that the conductivity is overestimated in this situation, leading to incorrect qualitative behaviour for the heat distribution in the microprocessor. To get a rough estimate of the quality of the various experiments performed in this section, we provide the value of the energy norm of the FE-HMM calculation in Table 4.3, for various resolutions of the micro structure, and the energy norm of the resolved solution.

Finally, in Figure 4.11 we present computations with the realistic value $\varepsilon = 10^{-6}$ m. We notice that a computation with this ε is no more expensive than previous computations with a coarser value of ε (recall that the FE-HMM captures the effective solution). The computational cost, however, depends on the resolution of the micro structure in the sampling domains,

Table 4.3. Energy norm of various solutions of the heat transfer problem in a microprocessor (FE-HMM solution, resolved solution, solution obtained using arithmetic average for the various tensors). Here micro degrees of freedom on the sampling domains are given by N_{mic}^d.

		$N_{\mathrm{mic}} = 4$	$N_{\mathrm{mic}} = 8$	$N_{\mathrm{mic}} = 16$	Fine-scale	Average
$\varepsilon = 5 \times 10^{-4}$	$\Vert u \Vert_A$	0.3906	0.4961	0.5514	0.5189	0.3122
$\varepsilon = 10^{-6}$	$\Vert u \Vert_A$	0.3894	0.4963	0.5512	–	0.3122

that is, the number of points per wavelength. Indeed, as can be seen in Figure 4.11, the effect of the micro error at the macroscale is not negligible (see again Abdulle (2005*b*) and Section 4.4 for an analysis of this behaviour).

4.3.4. More sophisticated coupling

In all of the above examples, we used a standard finite element method at the micro and macro levels. The structure of the FE-HMM algorithm, which couples the macro and micro solvers only via constraints, allows us to use other types of method.

For many problems, local conservation properties in the numerical approximation and flexibility in meshing (*e.g.*, hanging nodes, local refinements) are desirable. For single-scale problems, these requirements have triggered the development of discontinuous Galerkin (DG) finite element methods. Such methods have been extensively studied for hyperbolic problems, advection–diffusion and diffusion problems (see Arnold, Brezzi, Cockburn and Marini (2001/2002) for a review). While a large body of literature is available for DG methods applied to single-scale problems, the construction and analysis of multiscale DG methods have rarely been addressed. In the context of the HMM, a DG method for hyperbolic and parabolic one-dimensional scalar problems was proposed in Chen, E and Shu (2005*a*). For multiscale elliptic problems, a DG method based on HMM was proposed in Abdulle (2008, 2012).

Another issue, as seen in Section 4.3.3, is the cost of the repeated micro solutions in sampling domains in the FE-HMM. If the micro structure is regular enough, one can take advantage of fast micro solvers (based on spectral methods, for example) to speed up the computation time considerably. The simultaneous refinement of macro and micro meshes can be avoided in this situation (see Abdulle and Engquist (2007)).

If, in contrast, there is only low regularity of the micro solutions, one can try to avoid repeated micro problems by suitable interpolation of the micro solutions in a few representative sampling domains. Provided that there is some smoothness in the micro solution with respect to the macro variables, this strategy can be successful. The use of reduced basis techniques for numerical homogenization was first explored by Boyaval (2008) and recently integrated into the HMM methodology of Abdulle and Bai (2012), who have proposed and analysed a reduced basis FE-HMM.

4.4. Error estimates

In this subsection we discuss *a priori* and *a posteriori* error analysis for the FE-HMM. We present the main steps of the *a priori* error analysis for linear elliptic problems in Section 4.4.1. We only discuss the analysis for the FE-HMM based on macro and micro finite element methods, and we refer to papers by Abdulle and Engquist (2007) for an analysis using micro pseudo-spectral methods, Abdulle (2012) for an analysis using discontinuous Galerkin macro methods, and Abdulle and Bai (2012) for an analysis incorporating reduced basis methods at the micro level. In Section 4.4.2 we briefly mention how the results obtained in the linear case can be generalized to a class of non-linear problems. For details on the analysis of such problems, we refer to Abdulle and Vilmart (2011*b*, 2011*a*). Finally we also discuss *a posteriori* error analysis for the FE-HMM in Section 4.4.3, following Abdulle and Nonnenmacher (2009*a*, 2011*a*). Unless otherwise specified, we shall assume that the macroscopic triangulation $\mathcal{T}_H$ is conformal and shape-regular.

4.4.1. A priori error analysis

We collect here a few preliminary results that are useful for the analysis of the FE-HMM.

Energy equivalence and coercivity (Abdulle 2005*b*, E *et al.* 2005*b*, Abdulle 2009). As an application of the following lemma, we deduce the coercivity of the bilinear form (4.6), hence the existence and uniqueness of problem (4.5).

Lemma 4.2. Let $v_H \in S_0^\ell(\Omega, \mathcal{T}_H)$, and let v_{h,K_j} be the corresponding solution of (4.10) in $S^q(K_{\delta_j}, \mathcal{T}_h)$, with boundary conditions given by (4.8) or (4.9). Assume that (4.2) holds. Then we have

$$\|\nabla v_{H,\mathrm{lin}}\|_{L^2(K_{\delta_j})} \leq \|\nabla v_{h,K_j}\|_{L^2(K_{\delta_j})} \leq C\|\nabla v_{H,\mathrm{lin}}\|_{L^2(K_{\delta_j})}, \tag{4.24}$$

where $v_{H,lin}$ is defined in Section 4.3.2.

Reformulation of the FE-HMM (Abdulle 2009, Abdulle 2012, E *et al.* 2005*b*). The bilinear form (4.6) has the reformulation

$$B_H(v_H, w_H) = \sum_{K\in\mathcal{T}_H} \sum_{j=1}^{J} \omega_{K_j} a_K^0(x_{K_j}) \nabla v_H(x_{K_j}) \cdot \nabla w_H(x_{K_j}). \tag{4.25}$$

In order for the expression (4.6) to be equal to the expression (4.25), we see that the following identity must hold:

$$\frac{1}{|K_{\delta_j}|}\int_{K_{\delta_j}} a^{\varepsilon}(x)\nabla v_{h,K_j}\cdot\nabla w_{h,K_j}\,\mathrm{d}x = \frac{1}{|K|}\int_K a_K^0(x_{K_j})\nabla v_{H,\mathrm{lin}}\cdot\nabla w_{H,\mathrm{lin}}\,\mathrm{d}x, \tag{4.26}$$

for all $v_H, w_H \in S_0^{\ell}(\Omega,\mathcal{T}_H)$ and for all v_{h,K_j} (resp. w_{h,K_j}) solutions of problem (4.10). The following lemma gives an appropriate definition for $a_K^0(x_{K_j})$, so that the above equation is valid (see Abdulle (2012)).

Lemma 4.3. For e_i, $i = 1,\ldots,d$ (the canonical basis of $\mathbb{R}^d$) consider the following problem. Find $\psi^i_{h,K_j}\in S^q(K_{\delta_j},\mathcal{T}_h)$ such that

$$\int_{K_{\delta_j}} a^{\varepsilon}(x)\nabla\psi^i_{h,K_j}\cdot\nabla z_h\,\mathrm{d}x = -\int_{K_{\delta_j}} a^{\varepsilon}(x)e_i\cdot\nabla z_h\,\mathrm{d}x,\ \forall z_h\in S^q(K_{\delta_j},\mathcal{T}_h), \tag{4.27}$$

where $S^q(K_{\delta_j},\mathcal{T}_h)$ is defined in (4.7) with either periodic or Dirichlet boundary conditions. If $a_K^0(x_{K_j})$ is defined by

$$a_K^0(x_{K_j}) = \frac{1}{|K_{\delta_j}|}\int_{K_{\delta_j}} a^{\varepsilon}(x)\big(I + J^T_{\psi_{h,K_j}}(x)\big)\,\mathrm{d}x, \tag{4.28}$$

where $J_{\psi_{h,K_j}}(x)$ is the $d\times d$ matrix with entries

$$\big(J_{\psi_{h,K_j}}(x)\big)_{i\ell} = \frac{\partial\psi^i_{h,K_j}}{\partial x_\ell}(x),$$

then (4.26) holds.

Remark 4.4. We observe that a standard finite element method with numerical quadrature for the effective problem (4.3) reads as follows (see Ciarlet and Raviart (1972)). Find $u_{0,H}\in S_0^{\ell}(\Omega,\mathcal{T}_H)$ such that

$$B_{0,H}(u_{0,H},v_H) = \int_\Omega f v_H\,\mathrm{d}x,\quad \forall v_H\in S_0^{\ell}(\Omega,\mathcal{T}_H), \tag{4.29}$$

where

$$B_{0,H}(v_H,w_H) = \sum_{K\in\mathcal{T}_H}\sum_{j=1}^{J}\omega_{j,K}a^0(x_{K_j})\nabla v_H(x_{K_j})\cdot\nabla w_H(x_{K_j}). \tag{4.30}$$

Thus, (4.6) can be seen as a finite element method with numerical quadrature for a modified effective problem. We expect (4.25) to be close to (4.30) and hence u_H to be close to $u_{0,H}$, if the procedure to recover the macro data from the microscale computation is accurate enough. This will be seen in the error analysis presented below.

The first theorem gives the macroscopic error of the FE-HMM (Abdulle 2005*a*, Abdulle 2011, E *et al.* 2005*b*). It does not rely on structure assumption (*e.g.*, periodicity) of the multiscale problem (4.1), but requires sufficient smoothness of the data and the solution of the effective problem (4.3). For a given natural number ℓ and $\mu = 0$ or 1, we assume that the solution u^0 and the tensor a^0 of (4.3) satisfy:

(**H1**) $u^0 \in H^{\ell+1}(\Omega)$, $a^0_{ij} \in W^{\ell+\mu,\infty}(\Omega)$ for $i, j = 1, \ldots, d$.

We notice that hypothesis (**H1**) is the usual assumption for a finite element method with numerical quadrature to converge with optimal rate H^ℓ and $H^{\ell+1}$ in the H^1 and L^2 norms, respectively.

Theorem 4.5. (Macro error) Let u^0, u_H be the solutions of problems (4.3) and (4.5), respectively. Suppose that (4.2), (**Q1**), (**Q2**) and (**H1**) hold.[4] Then we have the following estimates:

$$\|u^0 - u_H\|_{H^1(\Omega)} \leq C\big(H^l + e(\text{HMM})\big), \tag{4.31}$$

$$\|u^0 - u_H\|_{L^2(\Omega)} \leq C\big(H^{l+1} + e(\text{HMM})\big), \tag{4.32}$$

where C is independent of H and h and

$$e(\text{HMM}) = \sup_{K\in\mathcal{T}_H, x_{K_j}\in K} \|a^0(x_{K_j}) - a^0_K(x_{K_j})\|_F, \tag{4.33}$$

where $a^0_K(x_{K_j})$ is defined in (4.28) and $\|\cdot\|_F$ is the Frobenius norm.

Proof. Let $u_{0,H}$ be the solution of (4.29). It is proved in Ciarlet and Raviart (1972) that provided (4.2), (**Q1**), (**Q2**) and (**H1**) hold, we have

$$\|u^0 - u_{0,H}\|_{H^1(\Omega)} \leq CH^l, \tag{4.34}$$

$$\|u^0 - u_{0,H}\|_{L^2(\Omega)} \leq CH^{l+1}, \tag{4.35}$$

where C is independent of H. Using the coercivity of the bilinear form (4.30) (which follows from (**Q1**)), we obtain

$$\|u_{0,H} - u_H\|_{H^1(\Omega)} \leq C \sup_{w_H\in S^\ell_0(\Omega,\mathcal{T}_H)} \frac{|B_{0,H}(u_H, w_H) - B_H(u_H, w_H)|}{\|w_H\|_{H^1(\Omega)}}.$$

In view of (4.30), (4.25) and (**Q2**), using the Cauchy–Schwarz inequality we see that

$$\begin{aligned}&|B_{0,H}(u_H, w_H) - B_H(u_H, w_H)| \\ &\qquad \leq C \sup_{K\in\mathcal{T}_H, x_{K_j}\in K} \|a^0(x_{K_j}) - a^0_K(x_{K_j})\|_F \|u_H\|_{H^1(\Omega)} \|w_H\|_{H^1(\Omega)}.\end{aligned}$$

The above estimates together with (4.34) and (4.35) give the claimed result. □

[4] In hypothesis (**H1**), $\mu = 0$ for the estimate (4.31) and $\mu = 1$ for the estimate (4.32).

The next task is to estimate the quantity $e(\mathrm{HMM})$. It is convenient to decompose this quantity further. We therefore consider, for each macro element $K \in \mathcal{T}_H$ and each sampling domain $K_{\delta_j} \subset K$,

$$\bar{a}_K^0(x_{j,K}) = \frac{1}{|K_{\delta_j}|} \int_{K_{\delta_j}} a^\varepsilon(x)\big(I + J^T_{\psi^i_{K_j}}(x)\big)\,\mathrm{d}x, \tag{4.36}$$

a tensor defined as in (4.28), but here

$$\big(J_{\psi^i_{K_j}}(x)\big)_{i\ell} = \frac{\partial \psi^i_{K_j}}{\partial x_\ell}(x),$$

and $\psi^i_{K_j}$ is the (exact) solution of (4.27) in (4.8) or (4.9). Then, obviously,

$$e(\mathrm{HMM}) \le e_{\mathrm{MIC}} + e_{\mathrm{MOD}},$$

where

$$e_{\mathrm{MIC}} := \sup_{K \in \mathcal{T}_H, x_{K_j} \in K} \|\bar{a}_K^0(x_{K_j}) - a_K^0(x_{K_j})\|_F, \tag{4.37}$$

$$e_{\mathrm{MOD}} := \sup_{K \in \mathcal{T}_H, x_{K_j} \in K} \|a^0(x_{K_j}) - \bar{a}_K^0(x_{K_j})\|_F. \tag{4.38}$$

For symmetric tensors, the quantity e_{MIC} was first analysed in Abdulle (2005*a*) and the quantity e_{MOD} in E *et al.* (2005*b*). To analyse e_{MIC}, appropriate regularity of the oscillating tensor is required. As we use standard *a priori* results of finite element methods in the sampling domain, we also need appropriate regularity of $\psi^i_{K_j}$. However, the coefficient a^ε is allowed to be discontinuous (at the macroscopic level) through smooth interfaces. We therefore make the following assumption:

(**H2**) for $q \in \mathbb{N}$ we assume that $|\psi^i_{K_j}|_{H^{q+1}(K_{\delta_j})} \le C\ \varepsilon^{-q}\sqrt{|K_{\delta_j}|}$, with C independent of ε, of the quadrature points x_{K_j} and the domain K_{δ_j}.

Remark 4.6. If one assumes $a^\varepsilon \in W^{1,\infty}(\Omega)$ and that $|a^\varepsilon_{ij}|_{W^{1,\infty}(\Omega)} \le C\varepsilon^{-1}$ for $i, j = 1, \ldots, d$, then **(H2)** can be proved for $q = 1$ and $W(K_{\delta_j}) = H^1_0(K_{\delta_j})$, using classical H^2 regularity results (Ladyzhenskaya 1985, Chapter 2.6) (in fact we only need local regularity for a^ε_{ij}). If $a^\varepsilon = a(x, x/\varepsilon) = a(x, y)$, which is Y-periodic in y, if $\delta/\varepsilon \in \mathbb{N}$, and $W(K_{\delta_j}) = W^1_{\mathrm{per}}(K_{\delta_j})$, then (**H2**) can be established for a given q, provided that the tensor a^ε is sufficiently smooth (this follows classical regularity results for solutions of periodic boundary value problems: see, *e.g.*, Bers, John and Schechter (1957, Chapter 3)).

The following theorem was first obtained in Abdulle (2005*a*) for elliptic problems and in Abdulle (2006) for problems in elasticity (see also Abdulle (2009, 2012) for generalizations). Du and Ming (2010) extended the estimate to non-symmetric tensors (see also Abdulle and Vilmart (2011*a*, Lemma 4.6) for a short alternative proof).

Theorem 4.7. (Micro error) Assume that (4.2) and (**H2**) hold. Then

$$e_{\mathrm{MIC}} \le C\left(\frac{h}{\varepsilon}\right)^{2q}, \tag{4.39}$$

where C is independent of H, h and ε.

As first observed in Abdulle (2005*a*, 2009), we see that a fixed number of grid points per wavelength of the smallest oscillation of the problem does not ensure robust convergence of the FE-HMM. Indeed, the above theorem shows that for fixed micro meshes, the errors due to the micro solver are dominant for sufficiently fine macro meshes. Comparing the rate of convergence of the macro solver (Theorem 4.5) with the rate of convergence of the micro solver (see Theorem 4.7) gives a criterion to obtain the optimal (macro) convergence rate with minimal computational cost. Indeed, if the macro problem (4.5) is solved in $S_0^\ell(\Omega, \mathcal{T}_H)$, with micro problems (4.10) solved in $S^q(K_{\delta_j}, \mathcal{T}_h)$, then, assuming uniform macro and micro meshes, we have

$$\hat{h} \simeq H^{\frac{\ell}{2q}} \quad \text{(optimal refinement in } H^1 \text{ norm)}, \tag{4.40}$$

$$\hat{h} \simeq H^{\frac{\ell+1}{2q}} \quad \text{(optimal refinement in } L^2 \text{ norm)}, \tag{4.41}$$

and we recall that $\hat{h} = h/\varepsilon$ is independent of ε (see Section 4.3.3).

The last task is to estimate the modelling error. Here we require structure assumptions on the tensor such as local periodicity or random stationarity. For deterministic tensors we make the following assumption:

(**H3**) $a^\varepsilon(x) = a(x, x/\varepsilon) = a(x,y)$ is Y-periodic in y, and

$$a_{ij}(x,y) \in \mathcal{C}\big(\bar{\Omega}; W^{1,\infty}_{\mathrm{per}}(Y)\big),$$

for all $i, j = 1, \dots, d$,

where we set $Y = (0,1)^d$ for simplicity. For such a tensor the variables x and y are usually referred to as slow and fast variables, respectively. First we note the following result, obtained by Abdulle and Schwab (2005).

Lemma 4.8. Assume (**H3**) and that the micro problems (4.10) are solved in $S^q(K_{\delta_j}, \mathcal{T}_h) \subset W^1_{\mathrm{per}}(K_{\delta_j})$, with $\delta/\varepsilon \in \mathbb{N}$. Assume further that the slow variable tensor $a(x, x/\varepsilon)$ is collocated at the quadrature points x_{K_j}, that is, $a(x_{K_j}, x/\varepsilon)$ is used in the problem (4.10) and in the bilinear form (4.6). Then $e_{\mathrm{MOD}} = 0$.

A consequence of this result is that the FE-HMM converges with a robust rate (*i.e.*, independent of ε) to the homogenized solution. In a more general situation, for example when the size of the period is unknown, a modelling error arises due to a mismatch of the sampling domain size and ε (*e.g.*, $\delta/\varepsilon \notin \mathbb{N}$) and to artificial boundary conditions. This error is often called cell resonance in the literature (see, *e.g.*, Hou *et al.* (1999)). The following result was obtained by E *et al.* (2005*b*) (see also Yue and E (2007)).

Theorem 4.9. (Modelling error) Assume (**H3**) and that the micro problems (4.10) are solved in $S^q(K_{\delta_j}, \mathcal{T}_h) \subset H_0^1(K_{\delta_j})$, with $\delta > \varepsilon$. Then

$$e_{\mathrm{MOD}} \leq C\biggl(\frac{\varepsilon}{\delta} + \delta\biggr),$$

where C is independent of H, h and ε.

We conclude this section by mentioning that a new approach to cell resonance has recently been proposed by Gloria (2011). By modifying the cell problem (4.10) and adding a zeroth-order term, we obtain a micro problem with better error decay due to artificial boundary conditions, because of the faster decay of the associated Green's function. The bias introduced by modifying the cell problem can be controlled by tuning the constant associated with the zeroth-order term. Gloria (2011) reports a rate of $(\varepsilon/\delta)^p$, $p < 4$. Numerical experiments, however, show that this asymptotic rate is only obtained for $\delta \gg \varepsilon$. Nevertheless, this constitutes a promising approach for the issue of cell resonance error.

Elliptic problems with random coefficients. The FE-HMM for elliptic problems with random conductivity tensors was investigated by E *et al.* (2005*b*), who analysed the modelling error e_{MOD} under a stationarity assumption. Convergence rates have been derived for dimensions $d = 1$ and $d = 3$ under the additional assumption that the random tensors satisfy a uniform mixing condition (Yurinskiĭ 1986).

4.4.2. Example of a priori analysis for non-linear problems

We consider a class of non-linear non-monotone multiscale problems described in Example 2:

$$-\nabla \cdot (a^\varepsilon(x, u^\varepsilon(x)))\nabla u^\varepsilon(x)) = f(x) \text{ in } \Omega, \quad u^\varepsilon(x) = 0 \text{ on } \partial\Omega, \tag{4.42}$$

where $a^\varepsilon(x,s)$ is a $d \times d$ tensor satisfying (4.2), with entries $a^\varepsilon_{ij}(x,s)$ which are continuous functions on $\overline{\Omega} \times \mathbb{R}$ and uniformly Lipschitz-continuous with respect to s and ε, that is,

$$\begin{aligned} \exists \Lambda_1 > 0, \quad \text{such that} \quad & |a^\varepsilon_{ij}(x,s_1) - a^\varepsilon_{ij}(x,s_2)| \leq \Lambda_1 |s_1 - s_2|, \\ & \forall \varepsilon,\ \forall x \in \overline{\Omega},\ \forall s_1, s_2 \in \mathbb{R},\ \forall 1 \leq i,j \leq d. \end{aligned} \tag{4.43}$$

For such problems, it is known (see, *e.g.*, Boccardo and Murat (1981)) that there exists a subsequence of $\{a^\varepsilon(\cdot,s)\}$ (again indexed by ε) such that the corresponding sequence of solutions $\{u^\varepsilon\}$ converges weakly to u^0 in $H^1(\Omega)$, where u^0 is the solution of the so-called homogenized problem

$$-\nabla \cdot (a^0(x, u^0(x))\nabla u^0(x)) = f(x) \ \text{ in } \Omega, \quad u^0(x) = 0 \ \text{ on } \partial\Omega, \tag{4.44}$$

with a homogenized tensor $a^0(x,s)$, which can be shown to have properties similar to those assumed for $a^\varepsilon(x,s)$. The numerical method for computing an effective solution of (4.42) was described in (4.12). The coerciveness of the bilinear form $B_H(z^H;\cdot,\cdot)$ can be established similarly as in the linear case. Then the existence of a solution of problem (4.12) relies on the Brouwer fixed-point theorem applied to the non-linear map $S_H : S_0^\ell(\Omega, \mathcal{T}_H) \to S_0^\ell(\Omega, \mathcal{T}_H)$, where, for $z_H \in S_0^\ell(\Omega, \mathcal{T}_H)$, $S_H z_H$ is defined to be the solution of the linear problem

$$B_H(z_H; S_H z_H, w_H) = \int_\Omega f w_H \,\mathrm{d}x, \ \forall w_H \in S_0^\ell(\Omega, \mathcal{T}_H). \tag{4.45}$$

For details we refer, for example, to Douglas and Dupont (1975). In contrast, the uniqueness is much more involved (see below). For deriving *a priori* error estimates and rates of convergence for the error $\|u^0 - u_H\|_{H^1(\Omega)}$, we follow the methodology described in the linear case. There is, however, one main difference: *a priori* error analysis for finite element methods with numerical quadrature such as the results in Ciarlet and Raviart (1972), used in the linear case, need to be derived. Such results, which are quite technical in the non-linear non-monotone case, were derived by Abdulle and Vilmart (2011*b*, 2012*b*). Using their results, it is then possible to show the following (see Abdulle and Vilmart (2012*b*) for a proof).

Theorem 4.10. Let u^0, u_H be the solutions of problems (4.44) and (4.12), respectively. Let $\ell \geq 1$. Let $\mu = 0$ or 1. Assume (**Q1**), (**Q2**) and (**H1**) hold.[5] In addition, assume that $u^0 \in W^{1,\infty}(\Omega)$, $\partial_u a^0_{mn} \in W^{1,\infty}(\Omega \times \mathbb{R})$, and that the coefficients $a^0_{mn}(x,s)$ are twice differentiable with respect to s, with the first- and second-order derivatives continuous and bounded on $\overline{\Omega} \times \mathbb{R}$, for all $m, n = 1, \ldots, d$.

Then there exist $r_0 > 0$ and $H_0 > 0$ such that, provided

$$H \leq H_0 \quad \text{and} \quad e(\text{HMM}) \leq r_0, \tag{4.46}$$

any solution u_H of (4.12) satisfies

$$\|u^0 - u_H\|_{H^1(\Omega)} \leq C(H^\ell + e(\text{HMM})) \qquad \text{if } \mu = 0, 1, \tag{4.47}$$

$$\|u^0 - u_H\|_{L^2(\Omega)} \leq C(H^{\ell+1} + e(\text{HMM})) \ \text{ if } \mu = 1, \tag{4.48}$$

where $e(\text{HMM})$ is given in (4.49). Here, the constants C are independent of H and $e(\text{HMM})$.

[5] Here $a^0_{ij} \in W^{\ell+\mu,\infty}(\Omega \times \mathbb{R})$ for $i, j = 1, \ldots, d$.

Remark 4.11. Note that the macroscopic family of triangulations $\{\mathcal{T}_H\}$ satisfies the inverse assumption $H/H_K \leq C$ for all $K \in \mathcal{T}_H$ and all $\mathcal{T}_H$ in Theorem 4.10; such a condition is often used for the finite element analysis of non-linear problems (Xu 1996). We also assume, for simplicity, the convexity of the domain Ω for the L^2 bound. Such a condition on the domain allows us to infer suitable regularity to use an Aubin–Nitsche-type duality argument. A weaker condition on the adjoint of the linearized problem associated to (4.44) (see Abdulle and Vilmart (2011*b*, equation (5))) is sufficient. This condition is usually assumed for L^2 estimates of linear finite element methods with numerical quadrature (see Raviart (1973, equation (2.11))).

The proof of Theorem 4.10 is more involved than the similar theorem in the linear case. We note that the additional assumptions compared to the linear case (Theorem 4.5), that is, the $W^{1,\infty}$ assumption on u^0 and the smoothness of $s \mapsto a(x,s)$, are already used for the analysis of finite element methods for single-scale non-linear problems of the type (4.44); see for example Douglas and Dupont (1975). The inverse assumption is also used in Douglas and Dupont (1975) to prove the uniqueness of a numerical solution. The quantity $e(\text{HMM})$ is defined by

$$e(\text{HMM}) := \sup_{K \in \mathcal{T}_H, x_{K_j} \in K, s \in \mathbb{R}} \|a^0(x_{K_j}, s) - a^0_{K_j}(s)\|_F, \tag{4.49}$$

where a^0 is the homogenized tensor of (4.44) and $a^0_{K_j}$ is the tensor defined similarly to (4.28) by replacing $a^\varepsilon(x)$ with the non-linear tensor $a^\varepsilon(x,s)$ (where s plays the role of a parameter), ψ^i_{h,K_j} with $\psi^{i,s}_{h,K_j}$ and $J_{\psi_{h,K_j}}(x)$ with $J_{\psi^s_{h,K_j}}(x)$. A nice feature of the analysis in Abdulle and Vilmart (2011*a*) is that micro and macro modelling errors can be analysed, thanks to Theorem 4.10, following the analysis obtained for the linear case (estimates with a similar rate of convergence can be derived: see Abdulle and Vilmart (2011*a*, Theorems 3.6, 3.7)). The proof of the uniqueness of a numerical solution of problem (4.12) is quite involved, but can be derived without any structure assumption on the oscillation provided that

$$\begin{aligned} &s \in \mathbb{R} \mapsto a^\varepsilon(\cdot, s) \in (W^{1,\infty}(\Omega))^{d\times d} \text{ is of class } C^2 \\ &\text{and} \quad |\partial^k_u a^\varepsilon(\cdot, s)|_{W^{1,\infty}(\Omega)} \leq C\varepsilon^{-1}, \ \ k \leq 2, \end{aligned} \tag{4.50}$$

where C is independent of s and ε (see Abdulle and Vilmart (2011*a*, Theorem 3.3)). Here we present a simpler uniqueness result, which can be stated solely in terms of the size of the macro and micro meshes (see Abdulle and Vilmart (2011*a*, Corollary 3.4)).

Theorem 4.12. In addition to the hypothesis of Theorem 4.10, assume (**H2**) and (**H3′**), and (4.50). Assume $W(K_{\delta_j}) = W^1_{\text{per}}(K_{\delta_j})$ (periodic coupling conditions), $\delta/\varepsilon \in \mathbb{N}$ and that the slow variable of the tensor $a(x, x/\varepsilon, s)$

is collocated at the quadrature points x_{K_j} in the problem (4.14) and in the form (4.13). Then there exists a positive constant H_0 such that, for all

$$(h/\varepsilon)^{2q} \leq H \leq H_0,$$

the solution u_H of (4.12) is unique.

Here (**H3′**) is defined similarly to (**H3**) for the non-linear tensor $a^{\varepsilon}_{ij}(x,s) = a_{ij}(x, x/\varepsilon, s)$, where $a_{ij}(x,y,s)$ is Y-periodic in y, and the map $(x,s) \mapsto a_{ij}(x,\cdot,s)$ is Lipschitz-continuous and bounded from $\overline{\Omega} \times \mathbb{R}$ into $W^{1,\infty}_{\text{per}}(Y)$.

4.4.3. A posteriori error analysis

The goal of the residual-based *a posteriori* analysis is to give upper and lower bounds for the error $e_H := u^0 - u_H$ in the H^1 norm for a given solution u_H of the problem (4.5) in terms of residual and data approximations, $\eta_H(\Omega), \xi_H(\Omega)$ defined in (4.19) and (4.20), respectively. As mentioned in Section 4.3.3 (see Example 3), we will concentrate here on piecewise linear simplicial macro finite elements. Compared to classical residual-based *a posteriori* error analysis, we have two additional difficulties:

- the data of the effective problem (4.3) (used in the residual η_H) are not available beforehand,
- 'variational crimes' (see Braess (2007)) are committed by replacing the true bilinear form with the FE-HMM bilinear form.

The first issue can be dealt with by introducing the multiscale flux jump for the modelling of (4.19). For the second issue, the following representation formula derived in Abdulle and Nonnenmacher (2009*a*, 2011*a*) is crucial. Let $B_0(\cdot,\cdot)$ be the bilinear form corresponding to the variational formulation of (4.3). Then we can relate the bilinear form $B_0(\cdot,\cdot)$ evaluated at e_H to the actual numerical solution u_H, the multiscale flux and data approximation as follows. Here for simplicity we consider piecewise linear macroscopic finite element functions.

Lemma 4.13. (Error representation formula) For all $v \in H^1_0(\Omega)$, we have

$$\begin{aligned} B_0(e_H, v) = \int_\Omega f v \,\mathrm{d}x - \sum_{e \in \mathcal{E}_H} \int_e \overline{[\![a^\varepsilon(x)\nabla u_h]\!]}_e v \,\mathrm{d}s \\ + \sum_{K \in \mathcal{T}_H} \int_K (a^0_K(x_K) - a^0(x))\nabla u_H \cdot \nabla v \,\mathrm{d}x, \end{aligned} \tag{4.51}$$

where $\mathcal{E}_H$ denotes the set of interfaces of the conformal mesh $\mathcal{T}_H$, u_H is the solution of (4.5) and where the multiscale flux jump $\overline{[\![a^\varepsilon(x)\nabla u_h]\!]}_e$ is defined in (4.18).

Using the above formula, we obtain for $v = u^0 - u_H$

$$\begin{aligned}B_0(e_H, e_H) = &\int_\Omega f_H(e_H - I_H e_H)\,\mathrm{d}x + \int_\Omega (f - f_H)(e_H - I_H e_H)\,\mathrm{d}x \\ &- \sum_{e \in \mathcal{E}_H} \int_e \overline{[\![a^\varepsilon(x)\nabla u_h]\!]}_e (e_H - I_H e_H)\,\mathrm{d}s \\ &+ \sum_{K \in \mathcal{T}_H} \int_K (a_K^0(x_K) - a^0(x))\nabla u_H \cdot \nabla e_H\,\mathrm{d}x,\end{aligned}$$

where $a_K^0(x_K)$ is defined in (4.28), and I_H denotes the Clément interpolation operator (see Clément (1975)). It is a linear operator $I_H : H^1(\Omega) \to S(\Omega, \mathcal{T}_H)$ having the property that, for all $v \in H^1(\Omega)$ and $K \in \mathcal{T}_H$,

$$\|v - I_H v\|_{L^2(K)} \le C H_K \|\nabla v\|_{L^2(N(K))} \tag{4.52}$$

and

$$\|\nabla(v - I_H v)\|_{L^2(K)} \le C\|\nabla v\|_{L^2(N(K))}, \tag{4.53}$$

where $N(K)$ is the neighbourhood of K that consists of all elements of $\mathcal{T}_H$ which have a non-empty intersection with K. Using the properties of the Clément interpolation operator, Cauchy–Schwarz and Poincaré inequalities, the coercivity of $B_0(\cdot,\cdot)$ and the finite overlapping property of the neighbourhoods $N(K)$, we can derive the following result (see Abdulle and Nonnenmacher (2011*a*) for a complete proof).

Theorem 4.14. (Upper bound) Let u^0, u_H be the solutions of problems (4.3) and (4.5), respectively. Then

$$\|u^0 - u_H\|_{H^1(\Omega)}^2 \le C\big(\eta_H(\Omega)^2 + \xi_H(\Omega)^2\big),$$

where C depends only on the shape-regularity constant, the coercivity and continuity bounds (4.2), the dimension d and the constant of the Poincaré inequality.

To derive a lower bound, bubble functions can be used, as in the classical theory (Verfürth 1996). We recall the construction of such functions. Let $\widetilde{\mathcal{T}}_H$ be a refinement of $\mathcal{T}_H$ such that every $K \in \mathcal{T}_H$ has an interior node $\tilde{x}_K \in K\backslash\partial K$ in $\widetilde{\mathcal{T}}_H$, and every edge e of $\mathcal{T}_H$ not on the boundary $\partial\Omega$ has an interior node $\tilde{x}_e \in e\backslash\partial e$ in $\widetilde{\mathcal{T}}_H$. Based on the triangulation $\widetilde{\mathcal{T}}_H$, we define a piecewise linear finite element space $S_0^1(\Omega, \widetilde{\mathcal{T}}_H) \supset S_0^1(\Omega, \mathcal{T}_H)$.

We start by estimating the so-called interior residual. For any $K \in \mathcal{T}_H$, consider an interior bubble function, that is, a function $\psi_K \in S_0^1(\Omega, \widetilde{\mathcal{T}}_H)$ such that $0 \le \psi_K \le 1$, $\psi_K(\tilde{x}_K) = 1$ and $\psi_K \equiv 0$ on $\Omega\backslash K$. Using the representation formula (4.51) with a test function given by $v := \psi_K f_H \in$

$H_0^1(\Omega)$, one can estimate the so-called interior residual

$$H_K^2\|f_H\|^2_{L^2(K)} \leq C\Big(\|\nabla e_H\|^2_{L^2(K)} + H_K^2\|f - f_H\|^2_{L^2(K)} + \|(a_K^0(x_K) - a^0(x))\nabla u_H\|^2_{L^2(K)}\Big). \tag{4.54}$$

Next, we estimate the so-called jump residual. Let $e \in \mathcal{E}_H$ be an interior interface and let $K_1 \in \mathcal{T}_H$ and $K_2 \in \mathcal{T}_H$ be such that $K_1 \cap K_2 = e$. Furthermore, let $\tilde{x}_e \in e$ be an interior node and let $\psi_e \in S_0^1(\Omega, \widetilde{\mathcal{T}}_H)$ be a bubble function such that $\psi_e(\tilde{x}_e) = 1$ and $\psi_e \equiv 0$ on $\Omega\backslash(K_1 \cup K_2)$. Using the representation formula (4.51) with $v := \psi_e$, we find

$$H_e\|[\![\overline{a^\varepsilon(x)\nabla u_h}]\!]_e\|^2_{L^2(e)} \leq C\Big(\|\nabla e_H\|^2_{L^2(\omega_e)} + H^2_{\omega_e}\|f - f_H\|^2_{L^2(\omega_e)} + \|(a_K^0(x_K) - a^0(x))\nabla u_H\|^2_{L^2(\omega_e)}\Big), \tag{4.55}$$

where $H_{\omega_e} = \max_{i=1,2} H_i$, $\omega_e = K_1 \cup K_2$. Using the estimates (4.54) and (4.55) on the interior and jump residuals, respectively, we derive the following lower bound (see Abdulle and Nonnenmacher (2011*a*) for a complete proof).

Theorem 4.15. (Lower bound) Let u^0, u_H be the solutions of problems (4.3) and (4.5), respectively. Denote by ω_K the domain which consists of all elements sharing at least one side with K. Then

$$\eta_H(K)^2 \leq C\big(\|u^0 - u_H\|^2_{H^1(\omega_K)} + \xi_H(\omega_K)^2\big),$$

where C depends only on the shape-regularity constant, the coercivity and continuity bounds (4.2), the dimension d and the constant of the Poincaré inequality.

5. Finite volume methods

5.1. The algorithm

In this section we discuss the application of HMM in the setting of finite volume methods. Finite volume methods are generally preferred when the macroscopic models are in the form of conservation laws. In fluid and solid mechanics, the macroscale models are often derived from the conservation laws of mass, momentum and energy:

$$\begin{aligned} \partial_t\rho + \nabla\cdot q &= 0, \\ \partial_t q + \nabla\cdot\tau &= 0, \\ \partial_t E + \nabla\cdot J &= 0. \end{aligned} \tag{5.1}$$

Here ρ is the mass density of the system, $q = \rho v$ is the momentum density, with v being the velocity, and E is the total energy density; τ and J are the momentum and energy fluxes, respectively. In conventional continuum models, empirical relations are sought to express these fluxes in terms of the conserved densities. In HMM, the fluxes are computed directly from some underlying microscopic models.

An example of the microscopic model that one can consider is that of molecular dynamics, that is, Newton's equations of motion for the constituting atoms:

$$\begin{aligned} \dot{x}_i &= v_i, \\ m_i \dot{v}_i &= \sum_{j \neq i} f\big(x_i(t) - x_j(t)\big). \end{aligned} \tag{5.2}$$

Here m_i is the mass of the ith particle and $f(r) = -\nabla\phi(r)$ is the force exerted on the particles with interatomic potential $\phi(r)$. Here we have assumed that the particles interact via a pairwise potential, even though the algorithms discussed here can be easily adapted to the situation when general interatomic potentials are used.

To see how the microscopic model can be linked with the macroscopic model, observe that for (5.2) we can also write down the analogues of the equations (5.1), by defining

$$\begin{aligned} \widetilde{\rho}(x,t) &= \sum_i m_i \delta\big(x - x_i(t)\big), \\ \widetilde{q}(x,t) &= \sum_i m_i v_i(t) \delta\big(x - x_i(t)\big), \\ \widetilde{E}(x,t) &= \frac{1}{2}\sum_i \left[m_i |v_i|^2 + \sum_{j \neq i} \phi\big(x_i(t) - x_j(t)\big)\right] \delta\big(x - x_i(t)\big), \end{aligned} \tag{5.3}$$

and

$$\begin{aligned} \widetilde{\tau}(x,t) = \sum_i & m_i v_i \otimes v_i \delta\big(x - x_i(t)\big) \\ & + \frac{1}{2}\sum_i \sum_{j \neq i} f\big(x_i - x_j\big) \otimes (x_i - x_j) \\ & \times \int_0^1 \delta\big(x - \big(x_j + \lambda(x_i - x_j)\big)\big)\, \mathrm{d}\lambda, \end{aligned} \tag{5.4a}$$

$$\begin{aligned} \widetilde{J}(x,t) = \sum_i & v_i \left[\frac{1}{2} m_i |v_i|^2 + \frac{1}{2}\sum_{j \neq i} \phi\big(x_i(t) - x_j(t)\big)\right] \delta\big(x - x_i(t)\big) \\ & + \frac{1}{4}\sum_{j \neq i} \big(v_j(t) + v_i(t)\big) \cdot f\big(x_j(t) - x_i(t)\big)(x_i - x_j) \\ & \times \int_0^1 \delta\big(x - \big(x_j + \lambda(x_i - x_j)\big)\big)\, \mathrm{d}\lambda. \end{aligned} \tag{5.4b}$$

One can easily verify that from (5.2) we have

$$\begin{aligned} \partial_t\widetilde{\rho} + \nabla_x \cdot \widetilde{q} &= 0, \\ \partial_t\widetilde{q} + \nabla_x \cdot \widetilde{\tau} &= 0, \\ \partial_t\widetilde{E} + \nabla_x \cdot \widetilde{J} &= 0. \end{aligned} \tag{5.5}$$

An important factor to be considered is the scale separation: microscopic processes usually take place at a length scale of nanometres (10^{-9} m) and a time scale of femto- or picoseconds, so one can neglect the variation of the macroscale variables. Locally, one can then think of the atomistic system as being constrained by the local macroscopic quantities, namely the local mass, momentum and energy densities. In addition, the atomistic system will stay close to local equilibrium since there is sufficient time for relaxation to take place. Consequently, one may view the models (5.1) as being the ensemble averages of (5.2). This is the ideal situation for HMM.

An HMM strategy for this setting has been developed by Li and E (2005) and Ren and E (2005). Our presentation follows that of Li and E.

Macroscale solver. Since the macroscopic model (5.1) is in the form of conservation laws, it is natural to choose as the macroscale solver a finite volume method. Although there are many different finite volume methods that are available for conservation laws (*e.g.*, LeVeque (1990) and Godlewski and Raviart (1996)), many involve the computation of the Jacobian for the flux functions. These are less suited to the present problem since the flux function is not explicitly given to us. An exception is the Lax–Friedrichs central scheme (*e.g.*, Nessyahu and Tadmor (1990)), which is formulated over a staggered grid. As it turns out, this method can easily be coupled with molecular dynamics.

To be more specific, we first rewrite the conservation laws in a generic form,

$$\partial_t u + \partial_x f = 0. \tag{5.6}$$

We will confine our discussion to macroscopically one-dimensional problems since the extension to higher dimensions is straightforward. The first-order central scheme represents the solutions by piecewise constants, which are the average values over each cell:

$$u_k^n = \frac{1}{\Delta x}\int_{x_{k-1/2}}^{x_{k+1/2}} u(x,t^n)\,\mathrm{d}x.$$

Time integration over $[x_j,\ x_{j+1}] \times [t^n,\ t^{n+1})$ leads to the scheme

$$u_{k+1/2}^{n+1} = \frac{u_k^n + u_{k+1}^n}{2} - \frac{\Delta t}{\Delta x}\big(f_{k+1}^n - f_k^n\big), \tag{5.7}$$

with numerical flux

$$f_k^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(x_k, t)\,\mathrm{d}t,$$

which is taken to be $f_k^n = f(x_k, t^n)$ in the central scheme.

Now (5.1) is incomplete since we still do not know the fluxes. Next we describe how this information can be extracted from molecular dynamics (MD) or other atomistic models.

Microscale solver. We first discuss the case when the microscopic model is MD. At each point where numerical fluxes are needed, we perform a local MD simulation to estimate the fluxes. The atomistic simulation will be constrained by the local macroscopic variables, which are the local density, momentum and energy. To initialize the MD simulation, we first arrange a trial configuration of the atoms according to the local density. From the local total energy, momentum and potential energy V of the trial configuration, we can compute the thermal energy K_θ by

$$K_\theta = E - V - \frac{1}{2}\rho v^2.$$

If K_θ is negative, the trial configuration is rejected and another trial configuration with the same number of atoms is generated. Otherwise it is accepted, and a local initial temperature is computed from K_θ. The velocities of the atoms are then given by

$$v_i = v + v_i',$$

where v_i' is thermal velocity given by the Maxwell–Boltzmann distribution with the given local temperature.

The set of ODEs (5.2) can be solved by a standard finite difference scheme such as Verlet's method. For the present analysis, however, we will assume that the ODEs are solved exactly to avoid unnecessary complication. After the MD system equilibrates, we obtain the required quantities by averaging. Specifically, let $\tilde{j}$ be the spatial average over the simulation box of a local flux whose expression was given in (5.4):

$$\tilde{J} = \tilde{J}(X), \quad X = (x_1, x_2, \dots, x_N, v_1, v_2, \dots, v_N). \tag{5.8}$$

Then we obtain the corresponding macroscale quantity by time-averaging,

$$J = \frac{1}{T} \int_{\tau}^{\tau+T} \tilde{J}\big(X(t)\big)\,\mathrm{d}t, \tag{5.9}$$

where τ is the starting point when the time-averaging begins, and T is the duration of the time-averaging. An additional ensemble averaging can also be used if desired. For detailed discussion of the set-up of the MD as well as boundary conditions, see Li and E (2005).

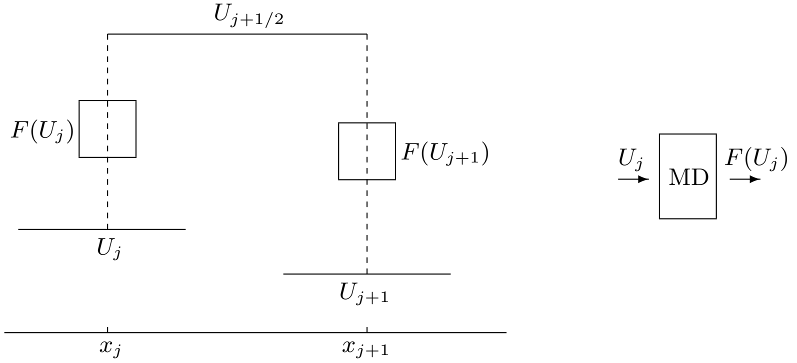


Figure 5.1. A schematic illustration of the numerical procedure. Starting from piecewise constant solutions u_k^n, one integrates (5.6) in time and in the cell x_k, x_{k+1}. The time step Δt is chosen in such a way that the waves coming from $x_{k+1/2}$ will not reach x_k, and thus for $t \in [t^n, t^{n+1}]$, $u(x_k, t) = u_k^n$. If $f(u)$ at x_k is found to be unknown, we perform an MD simulation using u_k^n to invoke and restrict the microscopic process. The required flux is then extracted from the simulation and the integration is completed. Analogously one can embed the MD simulation in higher-order macro schemes or higher dimensions.

The overall numerical procedure is shown schematically in Figure 5.1. At each time step, the scheme (5.7) requires as input the fluxes at grid point x_j to complete the time integration. These flux values are obtained by performing local MD simulations that are consistent with the local macroscale state $(\rho,\ q,\ E)$. Once these values are computed, one can advance to the next macro time step using (5.7).

One result from such a method is shown in Figure 5.2. Here the set-up for the macroscale model is a Riemann problem for one-dimensional wave propagation in solids. The result of HMM is compared with that of a direct MD simulation. The microscale model is MD with Lennard–Jones potential.

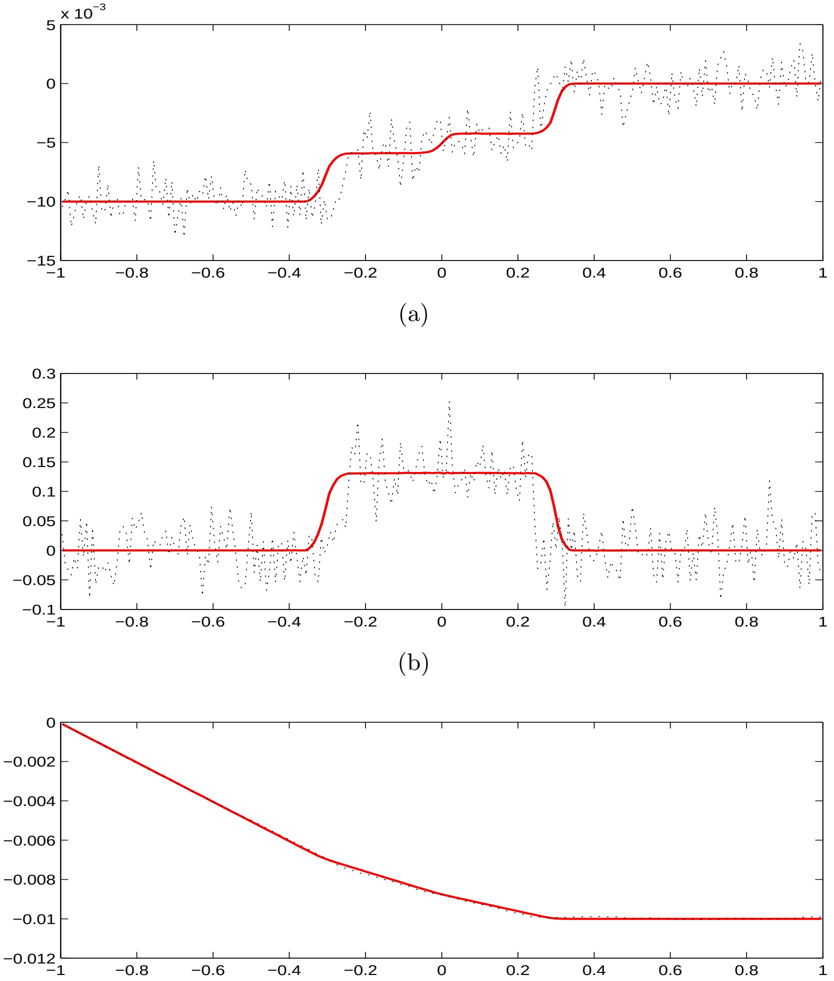
5.2. Error analysis

As we have seen before, error estimates for HMM proceed in two steps. The first step is to estimate the ‘HMM error’, that is, the error in the estimated data. The second step is to obtain the overall error estimate.

For the exact solution we will take the solution for the case when the fluxes are given by the averaged quantities for the equilibrium distribution, with parameters given by the local values of the macroscale variables (density, momentum, energy).

First, let us look at the HMM error. Let

$$e(\mathrm{HMM}) = E[|J - \bar{J}|], \tag{5.10}$$



(c)

Figure 5.2. Numerical test on shock formation and propagation. 200 macro grid points are used, and each local MD simulation consists of 40×10 atoms and 10^4 steps of time integration. The solution is displayed after 40 steps of integration over macro time steps. Solid line, computed solution; dashed line, full atom simulation (one realization). (a) Strain, (b) velocity, (c) displacement.

where J denotes the value of the flux computed by HMM, and $\bar{J}$ is the exact value for the flux:

$$\bar{J} = \int \tilde{J}(X)\,\mathrm{d}\mu_\infty, \tag{5.11}$$

with μ_∞ being the local equilibrium distribution.

Let L be the size of the microscale simulation box and let d be the dimension of the problem. Let μ_L^t denote the particle distribution for the MD system at time t, and let μ_L be the equilibrium particle distribution for a system of size L. In addition, define two quantities $\alpha(L)$ and $\tau_0(L)$ by the following:

$$\left|\int f(X)\,\mathrm{d}(\mu_L - \mu_L^t)\right| \le C\,\mathrm{e}^{-\frac{t}{\tau_0(L)}}\|f\|_{w,\infty}, \tag{A_1}$$

and

$$\left|\int f(X)\,\mathrm{d}(\mu_L - \mu_\infty)\right| \le C\alpha(L)\|f\|_{w,\infty}, \tag{A_2}$$

for any continuous function f that satisfies

$$\|f\|_{w,\infty} \le \infty,$$

where

$$\|f\|_{w,\infty} = \sup_X \big(1+|X|^2\big)^{-p/2}|f(X)|,$$

for $p \ge 2$. Here $\tau_0(L)$ can be regarded as the relaxation time for a system of size L, and $\alpha(L)$ measures the error due to the finite size effect.

Now let

$$J_L^t = \int \tilde{J}(X)\,\mathrm{d}\mu_L^t, \tag{5.12}$$

$$A(t,s) = E\big[\big(\tilde{J}(t) - J_L^t\big)\big(\tilde{J}(s) - J_L^s\big)\big], \tag{5.13}$$

where the expectation is with respect to μ_L^0, the initial distribution. We further assume that

$$\int_0^{+\infty} A(t,s)\,\mathrm{d}s \le C_0,\ \forall t \ge 0. \tag{A_3}$$

This amounts to assuming that the correlation has sufficiently fast time decay.

With (A_1), (A_2), we have

$$\left|\bar{J} - \frac{1}{T}\int_\tau^{\tau+T} J_L^t\,\mathrm{d}t\right| \le C(\alpha(L) + T^{-1}\,\mathrm{e}^{-\frac{t}{\tau_0(L)}}). \tag{5.14}$$

Combined with (A_3), we have

$$\begin{aligned}
&E\biggl[\biggl(J - \frac{1}{T}\int_{\tau}^{\tau+T} J_L^t \,\mathrm{d}t\biggr)^2\biggr] \\
&\quad = \frac{1}{T^2}E\biggl[\int_{\tau}^{\tau+T}\int_{\tau}^{\tau+T} (\tilde{J}(t) - J_L^t)(\tilde{J}(s) - J_L^s)\,\mathrm{d}t\,\mathrm{d}s\biggr], \\
&\quad \le \frac{C_0}{T}.
\end{aligned}$$

Hence, we have

$$e(\mathrm{HMM}) \le C\biggl(\alpha(L) + T^{-1}\,\mathrm{e}^{-\frac{\tau}{\tau_0(L)}} + \frac{1}{\sqrt{T}}\biggr). \tag{5.15}$$

If ensemble averaging is also used, for example with M independent copies, the above estimate becomes

$$e(\mathrm{HMM}) \le C\biggl(\alpha(L) + T^{-1}\,\mathrm{e}^{-\frac{\tau}{\tau_0(L)}} + \frac{1}{\sqrt{TM}}\biggr). \tag{5.16}$$

We clearly see that $e(\mathrm{HMM})$ is controlled by the behaviour of $\alpha(L)$ and $\tau_0(L)$. In general one expects that

$$\alpha(L) \sim L^{-d/2}, \tag{5.17}$$

from the Central Limit Theorem, and

$$\tau_0(L) \sim L^r, \tag{5.18}$$

with $r > 0$.

From (5.17) and (5.18), we see the conflicting effects of choosing the size of the MD system. When L is increased, the finite size error is reduced while the relaxation time τ_0 is increased. Similarly, when L is decreased, τ_0 decreases but $\alpha(L)$ will increase.

The macroscopic models considered in this paper have the following properties. They are hyperbolic and they have a strictly convex entropy function. To avoid irrelevant complications, we will further assume that the numerical solutions at the macroscale level are bounded and the exact solution is piecewise Lipschitz-continuous.

Assume that (5.6) is solved by the three-point conservative scheme

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\bigl(f_{j+1/2}^n - f_{j-1/2}^n\bigr), \tag{5.19}$$

where $f_{j+1/2} = F(u_j, u_{j+1})$, and we have omitted the superscript. For Lax–Friedrichs-type schemes, which are the ones considered in the following analysis, the numerical flux $f_{j+1/2}$ is given by

$$F(u_j, u_{j+1}) = \frac{1}{2}\big(f(u_j) + f(u_{j+1})\big) - \frac{a}{2}\big(u_{j+1} - u_j\big). \tag{5.20}$$

The constant a is chosen to be larger than the eigenvalues of ∇f,

$$a \geq \max |\lambda(\nabla f)|,$$

to ensure stability. Since the time step and the grid size are always proportional in the scheme, we will use Δx to indicate the discretization error.

The HMM solution can be written as

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\big(g_{j+1/2} - g_{j-1/2}\big), \tag{5.21}$$

$$g_{j+1/2} = G(u_j, u_{j+1}), \tag{5.22}$$

with

$$G(u_j, u_{j+1}) = \frac{1}{2}\big(g(u_j) + g(u_{j+1})\big) - \frac{a}{2}\big(u_{j+1} - u_j\big). \tag{5.23}$$

Define the intermediate variable

$$w_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\big(\widehat{\mathcal{B}}_{j+1/2} - \widehat{\mathcal{B}}_{j-1/2}\big), \tag{5.24}$$

$$\widehat{\mathcal{B}}_{j+1/2} = F(u_j, u_{j+1}), \tag{5.25}$$

and then

$$\|w^{n+1} - u^{n+1}\| \leq Ce(\text{HMM}), \tag{5.26}$$

with some constant C. The distance between $\{w_j^{n+1}\}$ and $\{v_j^n\}$, however, is directly related to the stability of the numerical methods.

Theorem 5.1. Let v be the exact solution to the PDEs (5.6) and $u_{\text{HMM}} = \{u_j^n\}$. Then we have

$$\big\|v - u_{\text{HMM}}\big\|_{L_2} \leq C\Delta x + \sqrt{C\Delta x + C\frac{e(\text{HMM})}{\Delta x}}. \tag{5.27}$$

5.3. Application to spall fractures

Spall fractures, in their simplest form, occur when two strong shock waves under strain conditions interact to produce a region of tension in the interior of a material body. The interacting waves arise when the system is subject to high-velocity impacts, for example. The shock waves lead to deformation of material at the highest possible rates, and the inertial effects become extremely important.

Experimental studies on fracture initiation due to rapid loading have been well documented (Antoun *et al.* 2003, Davison 2008). However, the computational approach remains a challenge due to the processes occurring on multiple physical scales.

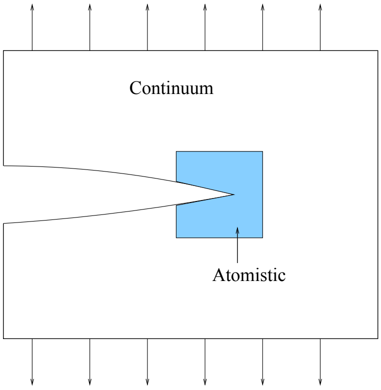


Figure 5.3. A coupled model.

This problem was studied by Xiantao Li using HMM, by integrating conventional methods for continuum models of solids, with an atomic level description: the molecular dynamics model. The coupling is shown in Figure 5.3. This method treats the crack tip area by explicitly incorporating the atomic interactions, to describe the interaction with the shock waves. Meanwhile, the integration with continuum models allows us to capture shock waves and simulate materials of realistic size.

In the continuum region, we solve the conventional computational model of the elastodynamics equation,

$$\rho_0 \frac{\partial u}{\partial t} = \nabla \cdot \sigma. \tag{5.28}$$

Here ρ_0 is the initial density and σ is the stress tensor.

The model is often combined with Griffith-type criteria to simulate crack growth: for example, see Camacho and Ortiz (1996) and Xu and Needleman (1994). However, many of the fundamental properties of cracks are determined by the detailed atomic interactions. Therefore, we introduce an atomistic region around the crack, where the system is modelled by molecular dynamics.

please check

Xiantao Li (private communication) used Roe's scheme as the macro solver away from the crack tip. Around the crack tip, numerical fluxes were evaluated with the help of MD. When the finite volume method is applied to the microscale model, the average momentum in a cell is determined by the total traction along the cell edges, which can be written as

$$t = \sum_{i,j} f_{ij}. \tag{5.29}$$

extra fig. ref. deleted

Here f_{ij} is the force between two atoms that lie on two sides of the interface. Such a coupling method maintains the continuity of the fluxes across the interface.

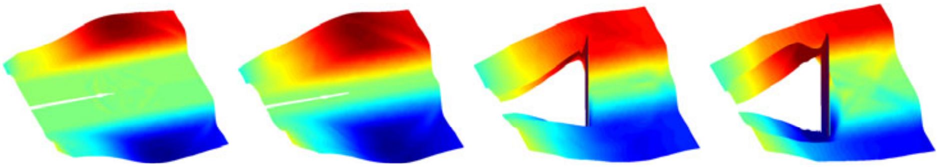


Figure 5.4. An open crack under mode I shock loading.

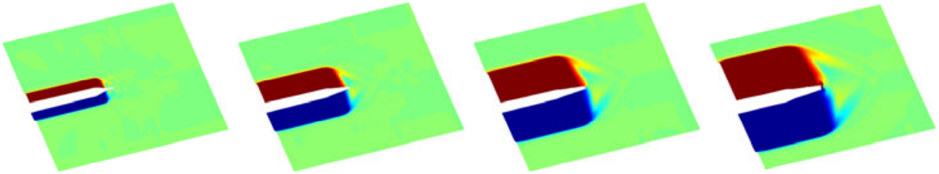


Figure 5.5. An open crack subject to face loading.

Another critical issue is the boundary condition for the atomistic region. This issue has been extensively studied in Li (2008), Li and E (2007) and Li and E (2006).

Numerical results. As the first application, Xiantao Li studied cracks in pure iron at low temperatures. The interatomic potential is the EAM model (Daw and Baskes 1984). The system is a rectangular sample with size 0.78 μm $\times$ 0.78 μm. The plane strain condition was assumed. For the MD model, this is enforced by a periodic boundary condition in the third direction with period equal to one atomic spacing. In the first experiment, tensile stress is rapidly applied from the top and bottom boundaries of the specimen. This generates two shock waves, which later propagate to the centre of the system to interact with a pre-existing crack. Below a certain threshold, no crack growth is observed after the two shocks collide. Instead of shock collisions, reflections off the crack faces are observed behind the crack tip. Above a certain threshold, one begins to observe crack growth, followed by a sequence of high-frequency lattice waves, the first two of which are of elliptical and diamond shapes. Experimental observation shows that spallation does not occur instantaneously when the tension exceeds the spallation strength: it occurs after a brief incubation period. This was also observed in our numerical experiments. A sequence of snapshots is shown in Figure 5.4.

In the second experiment, Li studied the response of a crack to loadings along the crack faces. These results are shown in Figure 5.5. In this case, the shocks are no longer of longitudinal type, and they propagate with a much slower speed. The magnitude required to produce fractures is much higher.

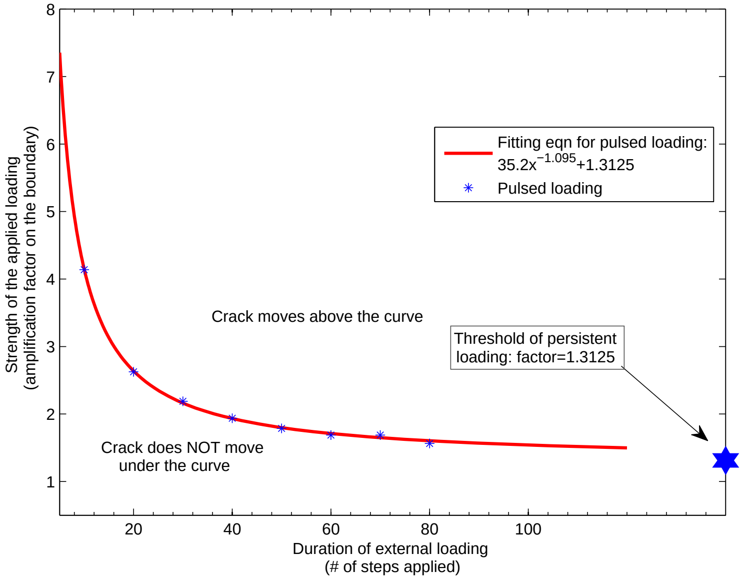


Figure 5.6. The onset of crack initiation.

Figure 5.6 summarizes the results under the impact of shocks of different magnitude. The goal is to investigate the strength of the material under various types of extreme loading conditions. The curve in the figure indicates the onset of the crack growth. In particular, for a given impact duration, the point on the curve indicates the magnitude for which the shock impact leads to fracture. For instantaneous shock loads, only very strong shocks are able to produce a growing crack. For continuing (slow) impacts, the threshold is much lower, and the curve approaches a plateau, which agrees with linear elasticity analysis. The results can be compared to the classical experimental results (Ravi-Chandar and Knauss 1984).

6. Conclusion

When designing multiscale methods, there are two important factors to consider.

1 From a modelling viewpoint, we would like to eliminate *ad hoc* modelling assumptions as much as possible.

2 From a numerical viewpoint, the algorithms have to be practical and must satisfy the standard requirements for numerical algorithms, such as stability and accuracy.

These two aims are not always compatible. Therefore compromises have to be made. This is one lesson we have learned over the past decade in multiscale modelling. HMM provides a general framework that allows us to make such compromises *and* make progress.

Although HMM has shown great promise, much work remains to be done in order to fully realize its potential. One should note, however, that serious obstacles remain before HMM can be applied to more challenging problems. For type A problems, namely problems involving local defects or singularities, the macrostates around the defects or singularities have to be inhomogeneous, and there are difficulties in formulating constrained microscopic models so that they are consistent with inhomogeneous macrostates. In addition, near defects or singularities, the local macroscale solutions are often singular too. Therefore, for numerical efficiency, local mesh refinements are needed. This is a technical problem that needs to be addressed. For type B problems, namely problems for which microscopic models are needed to supplement constitutive relations, the main obstacle is still the cost associated with running the microscopic models.

A major issue for multiscale methods is the effect of fluctuations. This is manifested in several ways. Fluctuation effects are almost always present in microscopic models, such as molecular dynamics or Monte Carlo models. They are also present in detailed models of random media. However, they are absent from most effective macroscale models. In cases when fluctuation effects are important, these effective macroscale models have to be remedied to include the fluctuations (Donev, Bell, Garcia and Alder 2010). How to do this is still very much an open question. In particular, in HMM, to reduce costs, we would like to perform the microscopic simulations on domains as small as possible. But the Central Limit Theorem tells us that fluctuations are inversely proportional to the square root of the volume. Hence computational savings might come at the cost of artificially enlarging the size of the fluctuations (Bal and Jing 2011). These are all issues that will need to be carefully addressed.

Acknowledgement

We are grateful to Xiantao Li for his help in writing this article. The work of Abdulle is supported in part by the Swiss National Science Foundation under Grant 200021 134716/1. The work of E has been supported in part by ONR grant N00014-01-1-0674. The work of Engquist has been supported in part by NSF grant DMS1016577. The work of Vanden-Eijnden has been supported in part by grant DMS-0708140 from NSF and grant N00014-11-1-0345 from ONR.

REFERENCES[6]

A. Abdulle (2005*a*), 'Multiscale methods for advection–diffusion problems', *Discrete Contin. Dyn. Syst.* **suppl.**, 11–21.

A. Abdulle (2005*b*), 'On *a priori* error analysis of fully discrete heterogeneous multiscale FEM', *SIAM Multiscale Model. Simul.* **4**, 447–459.

A. Abdulle (2006), 'Analysis of a heterogeneous multiscale FEM for problems in elasticity', *Math. Models Methods Appl. Sci.* **16**, 615–635.

A. Abdulle (2008), 'Multiscale method based on discontinuous Galerkin methods for homogenization problems', *CR Math. Acad. Sci. Paris* **346**, 97–102.

A. Abdulle (2009), 'The finite element heterogeneous multiscale method: A computational strategy for multiscale PDEs', *GAKUTO Int. Ser. Math. Sci. Appl.* **31**, 135–184.

A. Abdulle (2011), '*A priori* and *a posteriori* error analysis for numerical homogenization: A unified framework', *Ser. Contemp. Appl. Math. CAM* **16**, 280–305.

A. Abdulle (2012), 'Discontinuous Galerkin finite element heterogeneous multiscale method for elliptic problems with multiple scales', *Math. Comp.* **81**, 687–713.

date changed A. Abdulle and Y. Bai (2012), 'Reduced basis finite element heterogeneous multiscale method for high-order discretizations of elliptic homogenization problems', *J. Comput. Phys.*, to appear.

A. Abdulle and W. E (2003), 'Finite difference heterogeneous multi-scale method for homogenization problems', *J. Comput. Phys.* **191**, 18–39.

A. Abdulle and B. Engquist (2007), 'Finite element heterogeneous multiscale methods with near optimal computational complexity', *SIAM Multiscale Model. Simul.* **6**, 1059–1084.

A. Abdulle and M. Grote (2011), 'Finite element heterogeneous multiscale method for the wave equation', *SIAM Multiscale Model. Simul.* **9**, 766–792.

A. Abdulle and A. Nonnenmacher (2009*a*), '*A posteriori* error analysis of the heterogeneous multiscale method for homogenization problems', *CR Math. Acad. Sci. Paris* **347**, 1081–1086.

A. Abdulle and A. Nonnenmacher (2009*b*), 'A short and versatile finite element multiscale code for homogenization problem', *Comput. Methods Appl. Mech. Engrg* **198**, 2839–2859.

A. Abdulle and A. Nonnenmacher (2011*a*), 'Adaptive finite element heterogeneous multiscale method for homogenization problems', *Comput. Methods Appl. Mech. Engrg* **200**, 2710–2726.

A. Abdulle and A. Nonnenmacher (2011*b*), *A posteriori* error estimate in quantities of interest for the finite element heterogeneous multiscale method. Preprint.

A. Abdulle and C. Schwab (2005), 'Heterogeneous multiscale FEM for diffusion problems on rough surfaces', *SIAM Multiscale Model. Simul.* **3**, 195–220.

A. Abdulle and G. Vilmart (2011*a*), Analysis of the finite element heterogeneous multiscale method for nonmonotone elliptic homogenization problems. Preprint.

[6] The URLs cited in this work were correct at the time of going to press, but the publisher and the authors make no undertaking that the citations remain live or are accurate or appropriate.

A. Abdulle and G. Vilmart (2011*b*), 'The effect of numerical integration in the finite element method for nonmonotone nonlinear elliptic problems with application to numerical homogenization methods', *CR Acad. Sci. Paris, Ser. I* **349**, 1041–1046.

A. Abdulle and G. Vilmart (2012*a*), 'Coupling heterogeneous multiscale FEM with Runge–Kutta methods for parabolic homogenization problems: A fully discrete space–time analysis', *Math. Models Methods Appl. Sci.*, to appear.

A. Abdulle and G. Vilmart (2012*b*), '*A priori* error estimates for finite element methods with numerical quadrature for nonmonotone nonlinear elliptic problems', *Numer. Math.*, to appear.

C. Abrams and E. Vanden-Eijnden (2010), 'Large-scale conformational sampling of proteins using temperature-accelerated molecular dynamics', *Proc. Natl Acad. Sci.* **107**, 4961.

M. Ainsworth and R. Rankin (2012), 'Guaranteed computable bounds on quantities of interest in finite element computations', *Internat. J. Numer. Methods Engrg* **89**, 1605–1634.

T. Antoun, L. Seaman, D. Curran, G. Kanel, R. Razorenov and A. Utkin (2003), *Spall Fracture*, Springer.

G. Ariel, B. Engquist and R. Tsai (2009*a*), 'A multiscale method for highly oscillatory ordinary differential equations with resonance', *Math. Comp.* **78**, 929–956.

G. Ariel, B. Engquist and R. Tsai (2009*b*), 'Numerical multiscale methods for coupled oscillators', *Multiscale Model. Simul.* **7**, 1387–1404.

G. Ariel, B. Engquist, H.-O. Kreiss and R. Tsai (2009*c*), Multiscale computations for highly oscillatory problems. In *Multiscale Modeling and Simulation in Science*, Vol. 66 of *Lecture Notes in Computational Science and Engineering*, Springer, pp. 237–287.

G. Ariel, J. M. Sanz-Serna, and R. Tsai (2012), A multiscale technique for finding slow manifolds of stiff mechanical systems. Under review.

D. Arnold, F. Brezzi, B. Cockburn and D. Marini (2001/2002), 'Unified analysis of discontinuous Galerkin methods for elliptic problems', *SIAM J. Numer. Anal.* **39**, 1749–1779.

Z. Artstein, I. Kevrekidis, M. Slemrod and E. Titi (2007*a*), 'Slow observables of singularly perturbed differential equations', *Nonlinearity* **20**, 2463–2481.

Z. Artstein, J. Linshiz and E. Titi (2007*b*), 'Young measure approach to computing slowly advancing fast oscillations', *Multiscale Model. Simul.* **6**, 1085–1097.

I. Babuška (1976), Homogenization and its application: Mathematical and computational problems. In *Numerical Solution of Partial Differential Equations III: Proc. Third Sympos. SYNSPADE 1975*, pp. 89–116.

I. Babuška and J. Osborn (1983), 'Generalized finite element methods: Their performance and their relation to mixed methods', *SIAM J. Numer. Anal.* **20**, 510–536.

G. Bal and W. Jing (2011), 'Corrector theory for MsFEM and HMM in random media', *Multiscale Model. Simul.* **9**, 1549–1587.

R. Becker and R. Rannacher (2001), An optimal control approach to *a posteriori* error estimation in finite element methods. In *Acta Numerica*, Vol. 10, Cambridge University Press, pp. 1–102.

A. Bensoussan, J.-L. Lions and G. Papanicolaou (1978), *Asymptotic Analysis for Periodic Structures*, North-Holland.

L. Bers, F. John and M. Schechter (1957), *Partial Differential Equations*, Vol. 3A of *Lectures in Applied Mathematics*, Wiley.

L. Boccardo and F. Murat (1981), 'Homogénéisation de problèmes quasi-linéaires', *Publ. IRMA, Lille* **3**, 13–51.

S. Boyaval (2008), 'Reduced-basis approach for homogenization beyond the periodic setting', *Multiscale Model. Simul.* **7**, 466–494.

D. Braess (2007), *Finite Elements: Theory, Fast Solvers, and Applications in Elasticity Theory*, third edition, Cambridge University Press.

A. Brandt (2002), 'Multiscale scientific computation: Review 2001', *Multiscale and Multiresolution Methods: Theory and Applications* **20**, 3–96.

M. P. Calvo and J. M. Sanz-Serna (2009), 'Instabilities and inaccuracies in the integration of highly oscillatory problems', *SIAM J. Sci. Comput.* **31**, 1653–1677.

M. P. Calvo and J. M. Sanz-Serna (2010), 'Heterogeneous multiscale methods for mechanical systems with vibrations', *SIAM J. Sci. Comput.* **32**, 2029–2046.

G. Camacho and M. Ortiz (1996), 'Computational modelling of impact damage in brittle materials', *Internat. J. Solids Struct.* **33**, 2899–2938.

Y. Cao, D. Gillespie and L. Petzold (2005), 'Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems', *J. Comput. Phys.* **206**, 395–411.

R. Car and M. Parrinello (1985), 'Unified approach for molecular dynamics and density-functional theory', *Phys. Rev. Lett.* **55**, 2471–2474.

P. Chartier, A. Murua and J. M. Sanz-Serna (2010), 'Higher-order averaging, formal series and numerical integration I: B-series', *Found. Comput. Math.* **10**, 695–727.

S. Chen, W. E and C.-W. Shu (2005*a*), 'The heterogeneous multiscale method based on the discontinuous Galerkin method for hyperbolic and parabolic problems', *SIAM Multiscale Model. Simul.* **3**, 871–894.

Z. Chen, W. Deng and H. Ye (2005*b*), 'Upscaling of a class of nonlinear parabolic equations for the flow transport in heterogeneous porous media', *Comm. Math. Sci.* **3**, 493–515.

A. Chorin (1967), 'A numerical method for solving incompressible viscous flow problems', *J. Comput. Phys.* **2**, 12–26.

P. Ciarlet and P. Raviart (1972), The combined effect of curved boundaries and numerical integration in isoparametric finite element methods. In *The Mathematical Foundations of the Finite Element Method: With Applications to Partial Differential Equations*, Academic Press, pp. 409–474.

D. Cioranescu and P. Donato (1999), *An Introduction to Homogenization*, Vol. 17 of *Oxford Lecture Series in Mathematics and its Applications*, Oxford University Press.

P. Clément (1975), 'Approximation by finite element functions using local regularization', *Rev. Fran. Automat. Informat. Recherche Opérationnelle Sér. RAIRO Analyse Numérique* **9**, 77–84.

D. Cohen, T. Jahnke, K. Lorenz and C. Lubich (2006), Numerical integrators for highly oscillatory Hamiltonian systems: A review. In *Analysis, Modeling and Simulation of Multiscale Problems*, Springer, pp. 553–576.

L. Davison (2008), *Fundamentals of Shock Wave Propagation in Solids*, Springer.

M. Daw and M. Baskes (1984), 'Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals', *PRB* **29**, 6443.

E. De Giorgi and S. Spagnolo (1973), 'Sulla convergenza degli integrali dell'energia per operatori ellittici del secondo ordine', *Boll. Un. Mat. Ital.* **4**, 391–411.

A. Donev, J. Bell, A. Garcia and B. Alder (2010), 'A hybrid particle-continuum method for hydrodynamics of complex fluids', *Multiscale Model. Simul.* **8**, 871–911.

M. Dorobantu and B. Engquist (1998), 'Wavelet-based numerical homogenization', *SIAM J. Numer. Anal.* **35**, 540–559.

J. Douglas and T. Dupont (1975), 'A Galerkin method for a nonlinear Dirichlet problem', *Math. Comp.* **29**, 689–696.

R. Du and P. Ming (2010), 'Heterogeneous multiscale finite element method with novel numerical integration schemes', *Comm. Math. Sci.* **8**, 863–885.

W. E (2003), 'Analysis of the heterogeneous multiscale method for ordinary differential equations', *Comm. Math. Sci.* **1**, 423–436.

W. E (2011), *Principles of Multiscale Modeling*, Cambridge University Press.

W. E and B. Engquist (2002), The heterogeneous multi-scale methods. UCLA CAM Report 02-15.

W. E and B. Engquist (2003), 'The heterogeneous multi-scale methods', *Comm. Math. Sci.* **1**, 87–133.

W. E and J. Lu (2007), 'Seamless multiscale modeling via dynamics on fiber bundles', *Comm. Math. Sci.* **5**, 649–663.

W. E, B. Engquist, X. Li, W. Ren and E. Vanden-Eijnden (2007*a*), 'Heterogeneous multiscale methods: A review', *Comm. Comput. Phys.* **3**, 367–450.

W. E, D. Liu and E. Vanden-Eijnden (2005*a*), 'Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates', *J. Chem. Phys.* **123**, 194107.

W. E, D. Liu and E. Vanden-Eijnden (2007*b*), 'Nested stochastic simulation algorithms for chemical kinetic systems with multiple time scales', *J. Comput. Phys.* **221**, 158–180.

W. E, P. Ming and P. Zhang (2005*b*), 'Analysis of the heterogeneous multiscale method for elliptic homogenization problems', *J. Amer. Math. Soc.* **18**, 121–156.

W. E, W. Ren and E. Vanden-Eijnden (2009), 'A general strategy for designing seamless multiscale methods', *J. Comput. Phys.* **228**, 5437–5453.

Y. Efendiev and T. Hou (2009), *Multiscale Finite Element Methods: Theory and Applications*, Vol. 4 of *Surveys and Tutorials in the Applied Mathematical Sciences*, Springer.

B. Engquist (1987), 'Computation of oscillatory solutions to partial differential equations', *Nonlinear Hyperbolic Problems* (*St. Etienne 1986*) **1270**, 10–22.

B. Engquist and O. Runborg (2002), Wavelet-based numerical homogenization with applications. In *Multiscale and Multiresolution Methods: Theory and Applications* Vol. 20 of *Lecture Notes in Computational Science and Engineering*, Springer, pp. 97–148.

B. Engquist and Y.-H. Tsai (2005), 'Heterogeneous multiscale methods for stiff ordinary differential equations', *Math. Comp.* **74**, 1707–1742.

K. Eriksson, D. Estep, P. Hansbo and C. Johnson (1995), Introduction to adaptive methods for differential equations. In *Acta Numerica*, Vol. 4, Cambridge University Press, pp. 105–158.

I. Fatkullin and E. Vanden-Eijnden (2004), 'A computational strategy for multiscale chaotic systems with applications to Lorenz 96 model', *J. Comput. Phys.* **200**, 605–638.

E. Fermi, J. Pasta and S. Ulam (1955), Studies of the nonlinear problems I. Los Alamos Report LA-1940. Later published in *Collected Papers of Enrico Fermi*, Vol. II (E. Segre, ed.), University of Chicago Press (1965), p. 978.

W. Gautschi (1961), 'Numerical integration of ordinary differential equations based on trigonometric polynomials', *Numer. Math.* **3**, 381–397.

C. Gear and K. Gallivan (1982), Automatic methods for highly oscillatory ordinary differential equations. In *Numerical Analysis: Dundee 1981*, Vol. 912 of *Lecture Notes in Mathematics*, Springer, pp. 115–124.

C. Gear and I. Kevrekidis (2005), 'Constraint-defined manifolds: A legacy code approach to low-dimensional computation', *J. Sci. Comput.* **25**, 17–28.

M. Geers, V. Kouznetsova and W. Brekelmans (2010), 'Multi-scale computational homogenization: Trends and challenges', *J. Comput. Appl. Math.* **234**, 2175–2182.

D. Gillespie (1976), 'A general method for numerically simulating the stochastic time evolution of coupled chemical reactions', *J. Comput. Phys.* **22**, 403–434.

A. Gloria (2011), 'Reduction of the resonance error 1: Approximation of homogenized coefficients', *Math. Models Methods Appl. Sci.* **21**, 1601–1630.

E. Godlewski and P. Raviart (1996), *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Springer.

W. Grabowski (2001), 'Coupling cloud processes with the large-scale dynamics using the cloud-resolving convection parameterization (CRCP)', *J. Atmos. Sci.* **58**, 978–997.

E. Hairer, C. Lubich and G. Wanner (2002), *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Vol. 31 of *Springer Series in Computational Mathematics*, Springer.

E. Haseltine and J. Rawlings (2002), 'Approximate simulation of coupled fast and slow reactions for stochastic kinetics', *J. Chem. Phys.* **117**, 6959–6969.

P. Henning and M. Ohlberger (2010), 'The heterogeneous multiscale finite element method for advection–diffusion problems with rapidly oscillating coefficients and large expected drift', *Netw. Heterog. Media* **5**, 711–744.

V. Hoang and C. Schwab (2005), 'High-dimensional finite elements for elliptic problems with multiple scales', *Multiscale Model. Simul* **3**, 168–194.

T. Hou, X. Wu and Z. Cai (1999), 'Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients', *Math. Comp.* **68**, 913–943.

J. Irving and J. Kirkwood (1950), 'The statistical mechanical theory of transport processes IV', *J. Chem. Phys.* **18**, 817–829.

V. Jikov, S. Kozlov and O. Oleinik (1994), *Homogenization of Differential Operators and Integral Functionals*, Springer.

I. Kevrekidis, C. Gear, J. Hyman, P. Kevrekidis, O. Runborg and C. Theodoropoulos (2003), 'Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis', *Comm. Math. Sci.* **1**, 715–762.

H.-O. Kreiss (1979), 'Problems with different time scales for ordinary differential equations', *SIAM J. Numer. Anal.* **16**, 980–998.

H.-O. Kreiss (1992), Problems with different time scales. In *Acta Numerica*, Vol. 1, Cambridge University Press, pp. 101–139.

H.-O. Kreiss and J. Lorenz (1993), 'Manifolds of slow solutions for highly oscillatory problems', *Indiana Univ. Math. J.* **42**, 1169–1191.

T. Kurtz (1973), 'A limit theorem for perturbed operator semigroups with applications for random evolutions', *J. Functional Analysis* **12**, 55–67.

O. Ladyzhenskaya (1985), *The Boundary Value Problems of Mathematical Physics*, Vol. 49 of *Applied Mathematical Sciences*, Springer.

B. Leimkuhler and S. Reich (2004), *Simulating Hamiltonian Dynamics*, Vol. 14 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press.

R. LeVeque (1990), *Numerical Methods for Conservation Laws*, Birkhäuser.

X. Li (2008), 'Variational boundary conditions for molecular dynamic in solids: treatment of the loading conditions', *J. Comput. Phys.* **227**, 10078–10093.

X. Li and W. E (2005), 'Multiscale modeling for dynamics of solids at finite temperature', *J. Mech. Phys. Solids* **56**, 1650–1685.

X. Li and W. E (2006), 'Variational boundary conditions for molecular dynamics simulations of solids at low temperature', *Comm. Comput. Phys.* **1**, 136–176.

X. Li and W. E (2007), 'Boundary conditions for molecular dynamics simulations at finite temperature: Treatment of the heat bath', *Phys. Rev.* B **76**, 104107.

L. Maragliano and E. Vanden-Eijnden (2006), 'A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations', *Chem. Phys. Letters* **426**, 168–175.

L. Maragliano and E. Vanden-Eijnden (2007), 'On-the-fly string method for minimum free energy paths calculation', *Chem. Phys. Letters* **446**, 182–190.

A.-M. Matache and C. Schwab (2002), 'Two-scale FEM for homogenization problems', *M2AN Math. Model. Numer. Anal* **36**, 537–572.

A.-M. Matache, I. Babuška and C. Schwab (2000), 'Generalized p-FEM in homogenization', *Numer. Math.* **86**, 319–375.

C. Miehe, J. Schröder and C. Bayreuther (2002), 'On the homogenization analysis of composite materials based on discretized fluctuations on the microstructure', *Acta Mechanica* **135**, 1–16.

P. Ming and X. Yue (2006), 'Numerical methods for multiscale elliptic problems', *J. Comput. Phys.* **214**, 421–445.

P. Ming and P. Zhang (2007), 'Analysis of the heterogeneous multiscale method for parabolic homogenization problems', *Math. Comp.* **76**, 153–177.

F. Murat and L. Tartar (1997), 'H-convergence, topics in the mathematical modeling of composite materials', *Progr. Nonlinear Differential Equations Appl.* **31**, 21–43.

H. Nessyahu and E. Tadmor (1990), 'Nonoscillatory central differencing for hyperbolic conservation laws', *J. Comput. Phys.* **87**, 408–463.

N. Neuss, W. Jäger and G. Wittum (2001), 'Homogenization and multigrid', *Computing* **66**, 1–26.

R. Nochetto, A. Veeser and M. Verani (2009), 'A safeguarded dual weighted residual method', *IMA J. Numer. Anal.* **29**, 126–140.

A. Nonnenmacher (2011), Adaptive finite element methods for multiscale partial differential equations. Thèse no. 5097, École Polytechnique Fédérale de Lausanne.

J. Oden and S. Prudhomme (2001), 'Goal-oriented error estimation and adaptivity for the finite element method', *Comput. Math. Appl.* **41**, 735–756.

J. Oden, S. Prudhomme, A. Romkes and P. Bauman (2006), 'Multiscale modeling of physical phenomena: Adaptive control of models', *SIAM J. Sci. Comput.* **28**, 2359–2389.

M. Ohlberger (2005), '*A posteriori* error estimates for the heterogeneous multiscale finite element method for elliptic homogenization problems', *Multiscale Model. Simul* **4**, 88–114.

G. Papanicolaou (1977), Introduction to the asymptotic analysis of stochastic differential equations. In *Modern Modeling of Continuum Phenomena*, Vol. 16 of *Lectures in Applied Mathematics* (R. DiPrima, ed.), AMS, pp. 109–149.

L. Petzold, O. Laurent and Y. Jeng (1997), Numerical solution of highly oscillatory ordinary differential equations. In *Acta Numerica*, Vol. 6, Cambridge University Press, pp. 437–483.

S. Prudhomme and J. Oden (1999), 'On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors', *Comput. Methods Appl. Mech. Engrg* **176**, 313–331.

K. Ravi-Chandar and W. Knauss (1984), 'An experimental investigation into dynamic fracture I: Crack initiation and arrest', *Internat. J. Fracture* **43**, 247.

P. Raviart (1973), The use of numerical integration in finite element methods for solving parabolic equations. In *Topics in Numerical Analysis* (J. Miller, ed.), Academic Press, pp. 233–264.

W. Ren and W. E (2005), 'Heterogeneous multiscale method for the modeling of complex fluids and microfluidics', *J. Comput. Phys.* **204**, 1–26.

H. Salis and Y. Kaznessis (2005), 'Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions', *J. Chem. Phys.* **122**, 054103.

A. Samant and D. Vlachos (2005), 'Overcoming stiffness in stochastic simulation stemming from partial equilibrium: A multiscale Monte Carlo algorithm', *J. Chem. Phys.* **123**, 144114.

Sandia National Laboratories (1997–2010), The CUBIT tool suite.
http://cubit.sandia.gov

J. M. Sanz-Serna (2009), 'Modulated Fourier expansions and heterogeneous multiscale methods', *IMA J. Numer. Anal.* **29**, 595–605.

R. Scheid (1983), 'The accurate numerical solution of highly oscillatory ordinary differential equations', *Math. Comp.* **41**, 487–509.

R. Srivastava, L. You, J. Summers and J. Yin (2002), 'Stochastic vs. deterministic modeling of intracellular viral kinetics', *J. Theor. Biol.* **218**, 309–321.

E. Tadmor, M. Ortiz and R. Phillips (1996), 'Quasicontinuum analysis of defects in crystals', *Phil. Mag. A* **73**, 1529–1563.

M. Tao, H. Owhadi and J. Marsden (2010), ‘Nonintrusive and structure preserving multiscale integration of stiff ODEs, SDEs, and Hamiltonian systems with hidden slow dynamics via flow averaging’, *Multiscale Model. Simul.* **8**, 1269–1324.

K. Terada and N. Kikuchi (2001), ‘A class of general algorithms for multiscale analyses of heterogeneous media’, *Comput. Methods Appl. Mech. Engrg* **190**, 5427–5464.

E. Vanden-Eijnden (2003), ‘Numerical techniques for multi-scale dynamical systems with stochastic effects’, *Comm. Math. Sci.* **1**, 385–391.

E. Vanden-Eijnden (2007), ‘On HMM-like integrators and projective integration methods for systems with multiple time scales’, *Comm. Math. Sci.* **5**, 495–505.

R. Verfürth (1996), *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley-Teubner.

T. Wallstrom, S. Hou, M. Christie, L. Durlofsky and D. Sharp (1999), ‘Accurate scale up of two phase flow using renormalization and nonuniform coarsening’, *Comput. Geosci.* **3**, 67–87.

Y. Xing, W. Grabowski and A. Majda (2009), ‘New efficient sparse space–time algorithms for superparameterization on mesoscales’, *Monthly Weather Rev.* **137**, 4307–4324.

J. Xu (1996), ‘Two-grid discretization techniques for linear and nonlinear PDE’, *SIAM J. Numer. Anal.* **33**, 1759–1777.

X.-P. Xu and A. Needleman (1994), ‘Numerical simulations of fast crack growth in brittle solids’, *J. Mech. Phys. Solids* **42**, 1397–1434.

X. Yue and W. E (2007), ‘The local microscale problem in the multiscale modeling of strongly heterogeneous media: Effects of boundary conditions and cell size’, *J. Comput. Phys.* **222**, 556–572.

V. Yurinskiĭ (1986), ‘Averaging of symmetric diffusion in a random medium’, *Sibirsk. Mat. Zh.* **27**, 167–180, 215.