Journal of Computational Physics 228 (2009) 5437–5453

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A general strategy for designing seamless multiscale methods

Weinan E [a,b], Weiqing Ren [c,*], Eric Vanden-Eijnden [c]

[a] Department of Mathematics and PACM, Princeton University, Princeton, NJ 08544, USA
[b] School of Mathematical Sciences, Peking University, Beijing 100871, China
[c] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA

ARTICLE INFO

Article history:
Received 2 August 2008
Received in revised form 3 February 2009
Accepted 17 April 2009
Available online 3 May 2009

Keywords:
Multiscale method
Seamless
Error estimates
Complex fluids

ABSTRACT

We present a new general framework for designing multiscale methods. Compared with previous work such as Brandt's systematic up-scaling, the heterogeneous multiscale method (HMM) and the "equation-free" approach, this new framework has the distinct feature that it does not require reinitializing the microscale model at each macro time step or each macro iteration step. In the new strategy, the macro- and micro-models evolve simultaneously using different time steps (and therefore different clocks), and they exchange data at every step. The micro-model uses its own appropriate time step. The macro-model runs at a slower pace than required by accuracy and stability considerations for the macroscale dynamics, in order for the micro-model to relax. Examples are discussed and application to modeling complex fluids is presented.

1. Introduction

The main purpose of this paper is to present a new framework for designing multiscale, multi-physics methods. Compared with other existing general strategies such as systematic up-scaling [1,2], the heterogeneous multiscale method [3–5] and the "equation-free" approach [6], the new framework has the distinct advantage that it does not require reinitializing the microscale model at each macro time step or each macro iteration step. Methods designed under this new framework are therefore quite seamless and much easier to implement.

In many areas of science and engineering, we face the problem that we are interested in analyzing the macroscale behavior of a given system, but we do not have an explicit and accurate macroscopic model for the macroscale quantities that we are interested in. On the other hand, we do have at our disposal a microscopic model with satisfactory accuracy – the difficulty being that solving the full microscopic model is far too inefficient. Most well-known examples include:

(1) In molecular dynamics, we need an accurate force field which we often do not have. Instead, we have an electronic structure model such as models from the density functional theory.
(2) When modeling the dynamics of real gases, we need the equation of state which we often do not have. Instead, we have an accurate kinetic model.
(3) In continuum models of complex fluids, we need the constitutive relations which we often do not have. Instead, we have an accurate atomistic model, such as a molecular dynamics model, for the system. The same situation exists for solids.

[*] Corresponding author.
E-mail addresses: weinan@math.princeton.edu (W. E), weiqing@cims.nyu.edu (W. Ren), weinan@math.princeton.edu (E. Vanden-Eijnden).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.04.030

Many different methods have been developed to deal with these situations. Most well-known among these methods are the Car–Parrinello molecular dynamics [7], the quasi-continuum method for studying the deformation of solids [8,9] and the kinetic scheme for studying gas dynamics [10]. All these methods share the following features:

(1) They allow us to model the macroscale quantities of interest, by coupling with a microscale model, not by using *ad hoc* macroscale models.
(2) They make use of the scale separation in the system, either by modifying some small parameters in the problem (as is done in the Car–Parrinello molecular dynamics), or by solving the microscopic model on small spatio-temporal domains (as in the Knap–Ortiz version of the quasi-continuum method [9]).

The success of these methods and the success of more traditional multiscale methods such as the multi-grid method have given impetus to finding general framework for multiscale methods. Most notable among these are the systematic up-scaling approach of Brandt [2], the heterogeneous multiscale method [3–5] and the "equation-free" approach [6]. The hope is that a general framework might lead to general designing principles for multiscale methods and guidelines for carrying out error analysis, as was the case of finite difference and finite element methods for solving differential equations. To a large extent, the heterogeneous multiscale method has indeed achieved these goals [4].

In Ref. [2], Achi Brandt reviewed a general strategy for extending the multi-grid method and renormalization group analysis to multi-physics problems. The new strategy, which is now called "systematic up-scaling", allows the use of discrete models such as Monte Carlo methods or molecular dynamics (besides traditional partial differential equation models) at the finest level of the calculation, and it does not require explicit macroscale models to begin with. In fact, Brandt remarked that one might be able to construct an effective macroscale model from the data accumulated during the computation. In addition, one can exploit scale separation by restricting the size of the spatial–temporal domain over which the microscale models are simulated: "few sweeps are enough, due to the fast CMC equilibration. This fast equilibration also implies that the interpolation can be done just over a restricted *sub-domain*, serving as *window*: In the window interior fine-level equilibration is reached."

In the heterogeneous multiscale method (HMM), one begins with an assumption about the form of the macroscale model (not the detailed expression), based on which one selects a macro-solver for the problem. Due to the fact that the macro-model is not explicitly known, the microscale model is invoked during the computation to supply whatever data that is missing but needed for the macro-solver. Scale separation is exploited by observing that in the data estimation step, the computational domain for the microscopic model can be totally decoupled from the physical domain for the macroscale solver. It only has to be large enough to guarantee the required accuracy for the data. For the same reason, there is no direct communication between the different microscopic simulations carried out for estimating data at different spatial locations. All communications are done through the macro-solver. This observation motivated the construction of the fiber bundle structure for the multiscale problems handled by HMM [11]. This fiber bundle structure is also quite relevant to the seamless approach discussed below.

In the "equation-free" approach, one tries to link together simulations of the microscopic models on small spatial–temporal domains in order to "perform system level tasks" [6]. This is done through interpolation in space and extrapolation in time of ensemble-averaged macroscale quantities obtained from the microscopic simulations.

In all three approaches, one basic underlying assumption is that the microscopic process is in equilibrium with the local macroscopic state of the system. In other words, the local relaxation time for the microscopic process is much smaller than the time scale for the macroscopic evolution of the system.

From a practical viewpoint, the most notable common feature of these strategies is that they all require going back and forth between the macro- and micro-states of the system, even though the terminologies used are different (see Table 1 for a comparison of the different terminologies). This can become rather difficult in practical implementations, particularly when constructing micro-states from the macro-variables and when the micro-states are discrete and the macro-variables are continuous quantities.

The main motivation of the current paper is to present a strategy that bypasses this difficulty. In a nutshell, the basic idea is as follows.

(1) Run the micro-solver using its own time step $\delta\tau$.
(2) Run the macro-solver using its own time step $\Delta' t$.
(3) Exchange data between the micro- and macro-solvers at every step.

Table 1
Terminologies used in multi-grid, HMM, and the equation-free approach.

Method	Macro to micro	Micro to macro
Systematic up-scaling	Interpolation	Restriction (projection)
HMM	Reconstruction	Compression
Equation-free	Lifting	Restriction

The macro time step $\Delta' t$ should be much smaller than what is required to accurately resolving the macroscale behavior of the system. For example, if Δt is the time step size required for accurately resolving the macroscale behavior of the system, we choose $\Delta' t = \Delta t/M$ where $M > \tau_\varepsilon/\delta\tau$, τ_ε being the relaxation time of the microscopic model. This is necessary in order to guarantee that the microscopic process has sufficient time to relax.

Two important features of this procedure are:

(1) There is no need to reinitialize the microscopic solver.
(2) The micro- and macro-solvers use different clocks.

Intuitively, the basic idea is to force the microscale model to accommodate the changes in the macroscale environment at a much faster pace. For example, assume that the characteristic macro time scale is 1 s and the micro time scale is 1 fs $= 10^{-15}$ s. In a brute force calculation, the micro-model will run 10^{15} steps before the environment changes. HMM makes use of the separation of the time scales by running the micro-model until it is sufficiently relaxed, which requires far fewer than 10^{15} steps, and then extracting the data in order to evolve the macro system over a macro time step of 1 s. The price one has to pay is that one has to reinitialize the microscale solver at each macro time step. In other words, even though HMM skips $10^{15} - M$ (M is the number of steps that HMM has to run in order for the microscopic model to relax) micro steps of calculation, it still exchanges data between the macro- and micro-solvers after every 10^{15} micro time step interval. One can do this differently and exchange data every M micro time steps. By doing so, we change the clock for the micro-model, but we no longer need to reinitialize the micro-model every macro time step. Since the cost of the macro-solver is typically very small compared with the cost of the micro-solver, we may as well run the macro-solver using a smaller time step (i.e. $1/M$ s) and exchange data every time step. In this way the data exchanged are more smooth. It turns out that this has the added advantage that it also reduces the statistical error (see the numerical results presented below).

At a mathematical level, aspects of these ideas can be found in the framework of fiber bundle dynamics, proposed in Ref. [11]. At the level of algorithms, ideas of a similar origin were used in Car–Parrinello molecular dynamics [7]. Most directly relevant to the present work are the works in [12] where similar ideas were used to study dynamical systems with disparate time scales, in [13] where they are used to design methods to sample free energies in the context of molecular systems, and in [14] where they are used to develop multiscale methods for studying the macroscale behavior of fluids made up of chain molecules. This work can be considered as a direct generalization of the work in [14].

The present paper is organized as follows: In Section 2, we present the general framework and discuss three simple examples, an ODE (ordinary different equations) example, a SDE (stochastic differential equation) example, and an example of the parabolic homogenization problem. In Section 3, we discuss errors in these methods and we focus on the simple ODE and SDE discussed in Section 2. We also compare the new strategy with HMM. In Section 4, we present the application of the new seamless algorithm to modeling the macroscopic behavior of complex fluids modeled by chain molecules. Complex fluids were used as examples in Ref. [14], but there the discussion was restricted to shear flows. Here we consider general incompressible flows. In Section 5, we briefly discuss type A problems, namely problems for which the microscopic models are used to help resolving local singularities or defects. Concluding remarks are made in Section 6.

2. The seamless algorithm

2.1. Setup and the general framework

We will follow a top-down coupling strategy in the style of HMM [3,4]. Let us begin with a general formulation of multiscale problems. We will use U to denote the macro-state variables, and u to denote the micro-state variables in the problem. At the macroscopic level, we have an incomplete macroscale model:

$$\partial_t U = L(U; D), \tag{1}$$

where D denotes the data needed for the macroscale model to be complete. For example, for complex fluids, U might be the macroscopic velocity field and D might be the stress tensor. In addition, we also have a microscopic model:

$$\partial_\tau u = \mathcal{L}(u; U), \quad Qu(0) = U_0. \tag{2}$$

Here the macroscale variable U may enter the system as constraints, and Q is the compression operator that maps micro-state variables to macro-states variables [3]. We have used different notations t and τ to denote the time variable for the macro- and micro-models, to emphasize the point that we do not have to think about them over the same physical domain.

In general, 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 two different cases. The first is when there is no time scale separation, i.e. $\tau_\varepsilon \sim t_M$. In this case, from the viewpoint of numerical efficiency, there is not much room to play with as far as time scales are concerned. The second case is when $\tau_\varepsilon \ll t_M$. This is the case we will focus on. In this case, we can play with the time variable τ, as we will see later.

The general philosophy of HMM is to couple the macro- and micro-models such that the macro-state provides the constraints for the micro-model and the micro-model provides the needed constitutive data D for the macro-model. HMM proceeds by selecting a macro-solver for (1), and then

(1) Solve (2) in the appropriate computational domain for the micro-model;
(2) Use the results to estimate D;
(3) Use the macro-solver to evolve U.

Each time the microscopic model is invoked, it has to be reinitialized. Thus, one can write down the HMM procedure formally as:

(1) Given the current state of the macro-variable $U(t)$, reinitialize the micro-variable:

$$u_t(0) = RU(t). \tag{3}$$

(2) Evolve the micro-variable for M micro time steps:

$$u_t((m+1)\delta\tau) = \mathcal{S}_{\delta\tau}(u_t(m\delta\tau); U(t)), \quad m = 0, \ldots, M-1; \tag{4}$$

(3) Estimate D:

$$D(t) = \mathcal{D}_M(u_t(0), u_t(\delta\tau), \ldots, u_t(M\delta\tau)). \tag{5}$$

(4) Evolve the macro-variable for one macro time step using the macro-solver:

$$U(t+\Delta t) = S_{\Delta t}(U(t); D(t)). \tag{6}$$

(5) Set the current state of the macro-variable to $U(t+\Delta t)$ and repeat.

Here R is some reconstruction operator that reinitializes the micro-model in a way that is consistent with the current state of the macro-variables, $\mathcal{S}$ is the micro-solver, which also depends on $U(t)$ through the constraints, as indicated. $\mathcal{D}_M$ is some data processing operator which in general involves time/ensemble averaging. This is sometimes referred to as the data estimator. Finally S is the macro-solver.

The reason why the micro-variables must be reinitialized is that at the end of one HMM iteration, one is left with $u_t(M\delta\tau)$ satisfying $Qu_t(M\delta\tau) = U(t)$, but what is needed for the next iteration is $u_{t+\Delta t}(0)$ which should satisfy $Qu_{t+\Delta t}(0) = U(t+\Delta t) \neq U(t)$.

The parameter M is an important parameter in this algorithm. It has to be large enough such that the micro-state has reached its quasi-equilibrium state. In practice, the requirement is that

$$\tau_\varepsilon \ll M\delta\tau. \tag{7}$$

We can formulate a simpler algorithm that does not require reinitialization:

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

$$u(\tau+\delta\tau) = \mathcal{S}_{\delta\tau}(u(\tau); U(t)). \tag{8}$$

(2) Estimate D:

$$D = \mathcal{D}(u(\tau+\delta\tau)). \tag{9}$$

(3) Evolve the macro-variable

$$U(t+\Delta' t) = S_{\Delta' t}(U(t); D). \tag{10}$$

(4) Set the current state of the micro- and macro-variables respectively to $u(\tau+\delta\tau)$ and $U(t+\Delta' t)$ and repeat.

In this algorithm, we alternate between the macro- and the micro-solvers, each running with its own time step (therefore the micro- and macro-solvers use different clocks). At every step, the needed macroscale data is estimated from the results of the micro-model (at that step) and is supplied to the macro-solver. The new values of the macro-state variables are then used to constrain the micro-solver. In essence, the micro-solver is forced to adapt to the changes of the macroscale environment at a much faster pace. An illustration of this procedure is shown in the lower panel Fig. 1.

Remarks: Some remarks are in order:

(1) To guarantee that the microscale model will have sufficient time to adapt to the changes of the macro-state, we have to run the macroscale model on a slower pace than is required for accurately resolving the dynamics of macroscale quantities. If Δt is the time step size suitable for resolving the macroscale dynamics (i.e. the one used in HMM), then

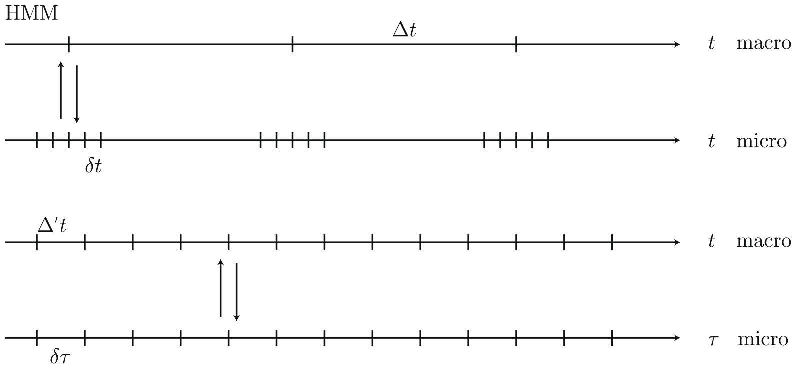

Fig. 1. Illustration of HMM (upper panel) and the seamless algorithm (lower panel).

we should take $\Delta' t = \Delta t/M$ where the value of M is similar to that in HMM and should be chosen according to (7) and the accuracy requirement. The dependence of the accuracy of the algorithm on the parameter M will be discussed in Section 3.

(2) The seamless algorithm is more costly than HMM, since we now evolve the macro-variables using much smaller time steps. However, this increase of cost is quite small since in most cases, there are far fewer macro-variables than micro-variables.

(3) Even though we do not use explicit time averaging to process the microscopic data in the new algorithm, the data is implicitly averaged over time in this algorithm, and the statistical error is comparable to that in HMM at equal cost. We will return to this issue in Section 3.

(4) One should note the difference between HMM, the seamless algorithm and the multiple time step methods [15] or asynchronous integrators [16]. In the seamless algorithm, the macro- and micro-models use different clocks. In HMM, the micro- and macro-models use the same clock, but the micro-model is only simulated over a small portion of the time interval. In the multiple time step methods or the asynchronous integrators, different time steps are used for the different terms in the model, but there is only one clock and all terms are propagated over the whole time-span.

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

$$C_S = \frac{\Delta' t}{\delta\tau} = \frac{\Delta t}{M\delta\tau}. \tag{11}$$

As an example, let us consider the case when the microscopic model is molecular dynamics, and the time step size is femtoseconds ($\delta\tau = 10^{-15}$ s). If one wants to simulate one second of physical time, then one needs to compute for 10^{15} steps. On the other hand, assume that the relaxation time is on the order of picoseconds (10^{-12} s) which is about 10^3 micro time steps, then $M = 10^4$ is a reasonable choice, and simulating one second of physical time using the seamless algorithm requires 10^4 steps. This is a factor of 10^{11} savings. The price to be paid is that we no longer obtain accurate information at the level of the microscopic details – we can only hope to get accurate information for the macro-state variables.

2.2. Example: stiff ODEs

To get a first impression about how the new algorithm works, let us consider the trivial example of stiff ODEs:

$$\begin{cases} \dfrac{dx}{dt} = f(x,y) \\ \dfrac{dy}{dt} = -\dfrac{1}{\varepsilon}(y - \varphi(x)). \end{cases} \tag{12}$$

Here $U = x, u = (x,y)$. In the style of Car–Parrinello molecular dynamics, one would change the small parameter ε to a bigger value ε':

$$\begin{cases} \dfrac{dx}{dt} = f(x,y) \\ \dfrac{dy}{dt} = -\dfrac{1}{\varepsilon'}(y - \varphi(x)), \end{cases} \tag{13}$$

where the size of ε' is determined by the accuracy consideration. This modified equation is then solved using standard methods.

We can look at this differently. Instead of changing the value of ε, we may change the clock for the micro-model, i.e. if we use $\tau = \varepsilon t/\varepsilon'$ in the second equation in (13), then (13) can be written as:

$$\begin{cases} \dfrac{dx}{dt} = f(x,y), \\ \dfrac{dy}{d\tau} = -\dfrac{1}{\varepsilon}(y - \varphi(x)). \end{cases} \tag{14}$$

If we discretize this equation using standard ODE solvers (e.g. the forward Euler) but with different time step sizes and denote by $y^n = y(n\delta\tau)$ and $x^n = x(n\Delta' t)$, we obtain the following algorithm:

$$y^{n+1} = y^n - \frac{\delta\tau}{\varepsilon}(y^n - \varphi(x^n)), \tag{15}$$

$$D^{n+1} = y^{n+1}, \tag{16}$$

$$x^{n+1} = x^n + \Delta' t f(x^n, D^{n+1}). \tag{17}$$

One can think of this as running the two solvers on different clocks with the same number of steps, and exchanging data every step.

2.3. Example: SDEs with multiple time scales

Next, consider:

$$\begin{cases} \dfrac{dx}{dt} = f(x,y), \\ dy = -\frac{1}{\varepsilon}(y - \varphi(x))dt + \sqrt{\frac{1}{\varepsilon}}dw, \end{cases} \tag{18}$$

where $w(t)$ is a Wiener process. The averaging theorems suggest that the effective macroscale equation should be in the form of an ODE:

$$\frac{dx}{dt} = F(x). \tag{19}$$

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:

$$y^{n,m+1} = y^{n,m} - \frac{\delta t}{\varepsilon}(y^{n,m} - \varphi(x^n)) + \sqrt{\frac{\delta t}{\varepsilon}}\xi^{n,m} \tag{20}$$

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

(3) Estimate $F(x)$:

$$F^n = \frac{1}{M}\sum_{m=1}^{M} f(x^n, y^{n,m}). \tag{21}$$

(4) Apply the macro-solver:

$$x^{n+1} = x^n + \Delta t F^n. \tag{22}$$

Here $x^n = x(n\Delta t)$.

In contrast, the seamless algorithm with forward Euler scheme is simply:

$$y^{n+1} = y^n - \frac{\delta\tau}{\varepsilon}(y^n - \phi(x^n)) + \sqrt{\frac{\delta\tau}{\varepsilon}}\xi^n, \tag{23}$$

$$x^{n+1} = x^n + \Delta' t f(x^n, y^{n+1}), \tag{24}$$

where $\{\xi^n\}$ are independent normal random variables with mean 0 and variance 1, and now $x^n = x(n\Delta' t)$.

2.4. The parabolic homogenization problem

Consider

$$\partial_t u^\varepsilon = \nabla \cdot \left(a\left(x, \frac{x}{\varepsilon}, t\right)\nabla u^\varepsilon\right), \tag{25}$$

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

$$\partial_t U = \nabla \cdot D, \tag{26}$$

$$D = \left\langle a\left(x, \frac{x}{\varepsilon}, t\right) \nabla u^{\varepsilon} \right\rangle, \tag{27}$$

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

As in HMM, if we choose a finite volume method as the macro-solver, then D needs to be evaluated at the cell boundaries [17]. We will make the assumption that the flux D depends on the local values of U and ∇U only. Consequently for the micro-model, we will impose the boundary condition that $u^{\varepsilon}(x,t) - Ax$ is periodic where $A = \nabla U$ evaluated at the location of interest.

Denote the micro-solver as:

$$u^{n+1} = \mathcal{S}_{\delta\tau,\delta x}(u^n; A). \tag{28}$$

In HMM, assume that we have the numerical approximation $U^n = U(t^n)$ (where $t^n = n\Delta t$) at the n-th macro time step, we obtain the numerical approximation at the next macro time step, $U^{n+1} = U((n+1)\Delta t)$ through the following steps:

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

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

with $m = 0, 1, \ldots, M-1$.
(4) Compute

$$D_{j-1/2}^{n+1} = \left\langle a\left(x, \frac{x}{\varepsilon}, t^n\right) \nabla u_j^{n,M} \right\rangle. \tag{29}$$

(5) Evolve the macro-state variables using

$$U_j^{n+1} = U_j^n + \Delta t \frac{D_{j+1/2}^{n+1} - D_{j-1/2}^{n+1}}{\Delta x}. \tag{30}$$

In contrast, if we use the proposed seamless strategy, then we have $U^n = U(n\Delta' t)$. Given $\{U_j^n\}$ where U_j^n denotes the numerical approximation at time $t^n = n\Delta' t$ inside the jth cell, the seamless algorithm produces the updated values U^{n+1} via the following steps:

(1) For each j, let $A_j^n = \left(U_j^n - U_{j-1}^n\right)/\Delta x$.
(2) Evolve the micro-state variable for one micro time step:

$$u_j^{n+1} = \mathcal{S}_{\delta\tau,\delta x}\left(u_j^n; A_j^n\right). \tag{31}$$

(3) Compute

$$D_{j-1/2}^{n+1} = \left\langle a\left(x, \frac{x}{\varepsilon}, t^n\right) \nabla u_j^{n+1} \right\rangle. \tag{32}$$

(4) Advance the macro-state for one reduced macro time step:

$$U_j^{n+1} = U_j^n + \Delta' t \frac{D_{j+1/2}^{n+1} - D_{j-1/2}^{n+1}}{\Delta x}. \tag{33}$$

3. Accuracy

Now let us discuss the accuracy of these methods. Instead of presenting a general framework for error analysis for the seamless method, we will focus on the simple examples of ODEs and SDEs discussed earlier. For these simple examples, the error for the seamless algorithm can be understood very simply using the following observation: Instead of thinking of the micro- and macro-solvers with different clocks, we can think of them as being running on the same clock with the same time step $\Delta' t$, but with a different value of ε that satisfies

$$\frac{\delta\tau}{\varepsilon} = \frac{\Delta' t}{\varepsilon'} \tag{34}$$

or

$$\varepsilon' = \frac{\Delta' t}{\delta\tau}\varepsilon. \tag{35}$$

For these simple examples, the seamless algorithm can be considered as standard algorithms applied to the modified problem with the modified parameter value. The error consists of two parts: The error due to boosting the parameter ε and the error due to numerical solution of the boosted model. For the stiff ODE (12), let us denote by x_h the numerical solution for the slow variable x and $\bar{x}$ the solution to the limiting equation

$$\frac{dx}{dt} = f(x, \varphi(x)). \tag{36}$$

Then we have

$$|x_h(t) - \bar{x}(t)| \leqslant C\left(\varepsilon' + \left(\frac{\Delta' t}{\varepsilon'}\right)^k + \Delta' t^\ell\right) = C\left(\frac{\Delta' t}{\delta\tau}\varepsilon + \left(\frac{\delta\tau}{\varepsilon}\right)^k + \Delta' t^\ell\right), \tag{37}$$

where the first term is due to the difference of the solution to the modified model and the solution to the limiting equation, the second and third terms are the error of the micro- and macro-solvers respectively when applied to the modified equation, and k is the order of accuracy of the micro-solver and ℓ is the order of accuracy of the macro-solver. In terms of the parameters entering the HMM algorithm, (37) can be written as

$$|x_h(t) - \bar{x}(t)| \leqslant C\left(\frac{\Delta t}{M\delta\tau}\varepsilon + \left(\frac{\delta\tau}{\varepsilon}\right)^k + \left(\frac{\Delta t}{M}\right)^\ell\right). \tag{38}$$

This is to be compared with the error for the HMM algorithm which is given by [18,4,19]:

$$|x_h(t) - \bar{x}(t)| \leqslant C\left(\Delta t e^{-M\delta\tau/\varepsilon} + \left(\frac{\delta\tau}{\varepsilon}\right)^k + \Delta t^\ell\right) \tag{39}$$

if we initialize the micro-solver such that $y^{n,0} = y^{n-1,M}$, and

$$|x_h(t) - \bar{x}(t)| \leqslant C\left(e^{-M\delta\tau/\varepsilon} + \left(\frac{\delta\tau}{\varepsilon}\right)^k + \Delta t^\ell\right) \tag{40}$$

if we initialize $y^{n,0}$ arbitrarily. Here the first term is due to the error in the data estimation process, the second term is the error in the micro-solver, and the last term is the error in the macro-solver. We see that under the usual circumstance and if we use high order macro-solver the last term in each expressions gives a small contribution, and since we can then take $M\delta\tau/\varepsilon \gg 1$, HMM gives slightly better accuracy because the first term in (39) or (40) is smaller than the first term in (38). However, this difference is quite insignificant since the other terms often dominate the error.

Let us now consider the SDE (18). We compare the numerical solution x_h with the exact solution $\bar{x}$ of the limiting equation:

$$\frac{dx}{dt} = F(x), \tag{41}$$

where $F(x)$ is given by [20]:

$$F(x) = \lim_{\varepsilon\to 0} \int f(x, y)\mu_x^\varepsilon(dy), \tag{42}$$

where $\mu_x^\varepsilon(dy)$ is the invariant measure of the dynamics of the fast variable y with x fixed. Using $\delta\tau/\varepsilon = \Delta' t/\varepsilon'$, we see that the seamless algorithm in (23) and (24) can be rewritten as:

$$y^{n+1} = y^n - \frac{\Delta' t}{\varepsilon'}(y^n - \varphi(x^n)) + \sqrt{\frac{\Delta' t}{\varepsilon'}}\xi^n, \tag{43}$$

$$x^{n+1} = x^n + \Delta' t f(x^n, y^{n+1}) \tag{44}$$

which can be considered as a standard discretization of (18) with the parameter value of ε boosted to ε':

$$\begin{cases} \dfrac{dx}{dt} = f(x, y), \\ dy = -\frac{1}{\varepsilon'}(y - \varphi(x))dt + \sqrt{\frac{1}{\varepsilon'}}dw. \end{cases} \tag{45}$$

Denote the solution of the above equation by $x_{\varepsilon'}$. It was shown in Ref. [20] that

$$E|x_{\varepsilon'}(t) - \bar{x}(t)| \leqslant C\sqrt{\varepsilon'}, \tag{46}$$

where $\bar{x}$ is the solution to the limiting Eq. (41). Therefore the error for the seamless algorithm is controlled by:

$$E|x_h(t) - \bar{x}(t)| \leqslant C\left(\sqrt{\varepsilon'} + \left(\frac{\Delta' t}{\varepsilon'}\right)^k + \Delta' t^\ell\right) = C\left(\sqrt{\frac{\varepsilon\Delta t}{M\delta\tau}} + \left(\frac{\delta\tau}{\varepsilon}\right)^k + \left(\frac{\Delta t}{M}\right)^\ell\right), \tag{47}$$

where the three terms represent the three contributions to the error: the first term is from (46), the second and third terms are the error of the micro- and macro-solvers respectively when applied to the modified Eq. (45), k is the weak order of accuracy of the micro-solver and l is the order of accuracy of the macro-solver. In comparison, the error for the HMM algorithm is [20]:

$$E|x_h(t) - \bar{x}(t)| \leqslant C\left(\sqrt{\frac{\varepsilon\Delta t}{M\delta\tau}} + \left(\frac{\delta\tau}{\varepsilon}\right)^k + \Delta t^\ell\right). \tag{48}$$

We see that the errors are comparable for this case.

Due to the stochastic nature of the micro-model and the estimated data therefrom, the numerical solution of both algorithms contains fluctuation or statistical error. This error is controlled by the first term in (47) and (48). In HMM, the statistical error can be made smaller by using a larger value of M, which increases the averaging time in the data estimation process and gives more accurate (i.e. less fluctuating) data. From the central limit theorem, the statistical error scales as $1/\sqrt{M\delta\tau}$ as shown in (48). In the seamless algorithm, we see from (47) that the parameter M plays the same role as in HMM and its value has the same effect on the accuracy of the numerical solution, though the data is not averaged explicitly over time. This is another nice feature of the seamless algorithm. It is generic and is not limited to the specific SDE example considered here. This will be demonstrated later in the numerical example in Section 4.3, where the micro-model is molecular dynamics.

4. Application to polymer fluids

4.1. Basic ideas

In this section, we apply the seamless algorithm just described to model the macroscopic fluid dynamics of chain molecules, under the assumption that the stress only depends on the rate of strain. The macroscopic model is that of hydrodynamics:

$$\begin{cases} \rho(\partial_t \boldsymbol{u} + \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u})) - \nabla\cdot\tau_s = 0, & \boldsymbol{x}\in\Omega, \\ \nabla\cdot\boldsymbol{u} = 0, \end{cases} \tag{49}$$

where ρ, $\boldsymbol{u}$ and τ_s are the fluid density, velocity and stress tensor, respectively (we use the subscript s in τ_s to distinguish between the stress tensor τ_s and the time τ of the micro-variables). These equations are simply the statement of mass and momentum conservation. The data that needs to be supplied from the micro-model is the stress: $D = \tau_s$.

The microscopic model we will use is a molecular dynamics model for chain molecules. Assume that we have N molecules, each molecule consists of k beads connected by springs. Each bead moves according to the Newton's equation:

$$m\frac{d^2\boldsymbol{x}_j}{d\tau^2} = -\frac{\partial V}{\partial \boldsymbol{x}_j}, \tag{50}$$

where $j = 1,\ldots,kN$ accounts for all the beads. The interaction potential consists of two parts:

(1) All beads interact via the Lennard–Jones (LJ) potential:

$$V^{\mathrm{LJ}}(r) = 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right), \tag{51}$$

where r is the distance between the beads, ϵ and σ are some energy and length parameters, respectively.

(2) There is an additional interaction between neighboring beads in each molecule via a spring force, modeled by the FENE (finitely extensible nonlinear elastic) potential:

$$V^{\mathrm{FENE}}(r) = \begin{cases} \frac{1}{2}kr_0^2 \ln\left(1 - \left(\frac{r}{r_0}\right)^2\right) & \text{if } r < r_0, \\ \infty & \text{if } r \geqslant r_0. \end{cases} \tag{52}$$

We use the following parameter values: The spring constant $k = 30$ and the maximum extensibility $r_0 = 1.5\sigma$.

From the molecular dynamics equation (50), one can derive a set of equations in the form of conservation laws. For that purpose, we define mass and momentum densities as follows:

$$\tilde{\rho}(\boldsymbol{x},\tau) = \sum_j m\delta(\boldsymbol{x} - \boldsymbol{x}_j(\tau)), \tag{53}$$

$$\tilde{\boldsymbol{m}}(\boldsymbol{x},\tau) = \sum_j m\boldsymbol{v}_j(\tau)\delta(\boldsymbol{x} - \boldsymbol{x}_j(\tau)), \tag{54}$$

Here $\boldsymbol{v}_j(\tau) = d\boldsymbol{x}_j/d\tau$ and $\delta(\cdot)$ is the Dirac delta function. One can prove as a consequence of the Newton's equation (50) that these fields satisfy the following:

$$\partial_\tau \tilde{\rho} + \nabla \cdot \tilde{\boldsymbol{m}} = 0, \tag{55}$$

$$\partial_\tau \tilde{\boldsymbol{m}} + \nabla \cdot \tilde{\sigma} = 0, \tag{56}$$

where we defined the analog of the stress tensor:

$$\tilde{\sigma}(\boldsymbol{x},\tau) = \sum_i m\boldsymbol{v}_i(\tau) \otimes \boldsymbol{v}_i(\tau)\delta(\boldsymbol{x} - \boldsymbol{x}_i(\tau)) + \frac{1}{2}\sum_i \sum_{j\neq i}(\boldsymbol{x}_i(\tau) - \boldsymbol{x}_j(\tau)) \otimes \boldsymbol{f}_{\;ij}(\tau)\int_0^1 \delta(\boldsymbol{x} - (1-\lambda)\boldsymbol{x}_j(\tau) - \lambda\boldsymbol{x}_i(\tau))d\lambda. \tag{57}$$

Here $\boldsymbol{f}_{\;ij}(\tau)$ is the force between the ith and the jth beads. (57) is the well-known Irving–Kirkwood formula [21].

These relations provide the needed link between the atomistic model and the continuum model: Upon (ensemble) averaging, (55) and (56) becomes

$$\partial_\tau \langle\tilde{\rho}\rangle + \nabla \cdot \langle\tilde{\boldsymbol{m}}\rangle = 0, \tag{58}$$

$$\partial_\tau \langle\tilde{\boldsymbol{m}}\rangle + \nabla \cdot \langle\tilde{\sigma}\rangle = 0. \tag{59}$$

Here $\langle\tilde{\sigma}\rangle$ is the total momentum flux. Stress is obtained if we subtract out the contribution due to momentum transport by the average velocity. Specifically we split the particle velocity into two parts:

$$\boldsymbol{v}_i = \boldsymbol{v}'_i + \boldsymbol{u}(\boldsymbol{x}_i,\tau), \tag{60}$$

where $\boldsymbol{v}'_i$ is the fluctuating part of the velocity and $\boldsymbol{u}$ is the mean velocity defined as:

$$\boldsymbol{u}(\boldsymbol{x},\tau) = \left\langle \sum_j \boldsymbol{v}_j(\tau)\delta(\boldsymbol{x} - \boldsymbol{x}_j(\tau)) \right\rangle. \tag{61}$$

The average momentum flux $\tilde{\sigma}$ can then be written as:

$$\langle\tilde{\sigma}\rangle = \langle\tilde{\rho}\rangle\boldsymbol{u} \otimes \boldsymbol{u} - \langle\tilde{\tau}_s\rangle, \tag{62}$$

where $\tilde{\rho}$ is given in (53) and $\tilde{\tau}_s$ is given by

$$\tilde{\tau}_s = -\sum_i m\boldsymbol{v}'_i(\tau) \otimes \boldsymbol{v}'_i(\tau)\delta(\boldsymbol{x} - \boldsymbol{x}_i(\tau)) - \frac{1}{2}\sum_i \sum_{j\neq i}(\boldsymbol{x}_i(\tau) - \boldsymbol{x}_j(\tau)) \otimes \boldsymbol{f}_{\;ij}(\tau)\int_0^1 \delta(\boldsymbol{x} - (1-\lambda)\boldsymbol{x}_j(\tau) - \lambda\boldsymbol{x}_i(\tau))d\lambda. \tag{63}$$

The ensemble average of above expression gives the stress tensor in the macro-model (49): $\tau_s = \langle\tilde{\tau}_s\rangle$.

To use this in HMM and the seamless algorithm, we have to set up the molecular dynamics in such a way that the *average velocity field coincides with the macroscopic velocity field*. We have assumed that the stress is only a function of the rate of strain. Therefore, associated with each macro grid point where the stress is needed, we make the approximation that the macroscopic velocity field is a linear function:

$$\boldsymbol{u} = \boldsymbol{u}_0 + A\boldsymbol{x}, \tag{64}$$

where A is the rate of strain tensor at the given macroscopic grid point. It is easy to see that without loss of generality, we can set $\boldsymbol{u}_0 = 0$. We will set up a molecular dynamics system which is constrained in such a way that its average strain rate coincides with A. Following Refs. [22,14], this constraint is imposed by using a periodic boundary conditions for particle positions with respect to an evolving computational box. The vertices of the MD box move according to the dynamics:

$$\frac{d\boldsymbol{X}}{d\tau} = A\boldsymbol{X}(\tau). \tag{65}$$

When a particle goes outside the simulation box, it is inserted back from the opposite side of the box and at the same time, its velocity is modified according to the imposed velocity gradient: $\bar{\boldsymbol{v}}_i = \boldsymbol{v}_i + A(\bar{\boldsymbol{x}}_i - \boldsymbol{x}_i)$.

From the results of the molecular dynamics, we then compute the local stress using (63). Since the MD is constrained so that the average velocity field is $\boldsymbol{u} = Ax$, the fluctuating velocity of the particles in (63) is simply given by $\boldsymbol{v}'_i = \boldsymbol{v}_i - A\boldsymbol{x}_i$.

In practice, in the seamless algorithm, we replace the ensemble average in (62) by spatial average (or spatial–temporal average, as is done in HMM):

$$\tau_s(n\Delta' t) = \frac{1}{V}\int_\omega \tilde{\tau}_s(\boldsymbol{x}, n\delta\tau)d\boldsymbol{x}. \tag{66}$$

Here ω is the (time-dependent) MD simulation box, V is the volume of ω.

There is a basic difference between HMM and the new seamless algorithm on how the stress is obtained from MD. In HMM, at each macro time step, a constrained MD is performed and the stress is then computed following the dynamics. During the MD, the matrix A in (65) is fixed at the value given by the macro-state at that macro time step. Usually, the above formula is also averaged over time to give better results (with less statistical fluctuations). In the seamless algorithm, however, (65) as well as the MD equations are solved simultaneously with the macro-model, and the data are exchanged every time step. Thus the matrix A changes at every step. But this change is very slow, due to the fact that we use a macro time step $\Delta' t$ which is smaller by a factor of M than the time step Δt used in HMM (see (6) and (10) and also the algorithms below).

4.2. Algorithms

In principle, we should work with a set of compressible flow type of equations at the macro-level, and an NVE ensemble at the micro-level, as was described above. In practice, we will make two approximations: First we assume the flow is incompressible at the macroscale, secondly we will work with the NVT ensemble in molecular dynamics, i.e. we will impose constant temperature in MD [23]. We will also limit ourselves to the situation when the macroscale flow is a two-dimensional flow. The molecular dynamics, however, is done in three dimensions, and simple periodic boundary condition is used in the third direction. The constant temperature constraint is imposed by modifying the Newton's equation for the component normal to the plane of the macroscale flow:

$$m\frac{d^2 y_j}{d\tau^2} = -\frac{\partial V}{\partial y_j} - \Gamma \dot{y}_j + \eta_j, \tag{67}$$

where $V = V^{\mathrm{LJ}} + V^{\mathrm{FENE}}$, Γ is a friction coefficient which is set to be 1 (in molecular units, i.e. $\sqrt{m\epsilon}/\sigma$) in the following numerical examples, and η_j is a random force with zero mean and variance $2m\Gamma k_B T$.

In HMM or the seamless method, we start with a macro-solver for the macro-model (49). Since the macroscale model is in the form of the equations for incompressible flows, it is natural to use the projection method as the macro-solver [24]. The projection method is a fractional step method. Let us denote the time step by Δt (it would be $\Delta' t$ in the seamless method), and the numerical solution at the time $t_n = n\Delta t$ by $\boldsymbol{u}^n$. In the projection method, we discretize the time derivative in the momentum equation using the forward Euler scheme:

$$\rho\frac{\tilde{\boldsymbol{u}}^{n+1} - \boldsymbol{u}^n}{\Delta t} + \nabla\cdot\left(\rho\boldsymbol{u}^n\otimes\boldsymbol{u}^n - \tau_s^n\right) = 0. \tag{68}$$

For the moment, the pressure as well as the incompressibility condition are neglected. Next, the velocity field $\boldsymbol{u}^{n+1}$ at the new time step $t_{n+1} = (n+1)\Delta t$ is obtained by projecting $\tilde{\boldsymbol{u}}^{n+1}$ onto the divergence-free subspace:

$$\rho\frac{\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}^{n+1}}{\Delta t} + \nabla p^{n+1} = 0, \tag{69}$$

where p^{n+1} is determined by the incompressibility condition:

$$\nabla\cdot\boldsymbol{u}^{n+1} = 0. \tag{70}$$

In terms of the pressure field, this becomes:

$$\Delta p^{n+1} = \frac{\rho}{\Delta t}\nabla\cdot\tilde{\boldsymbol{u}}^{n+1} \tag{71}$$

with a Neumann type of boundary condition.

The spatial derivatives in the above equations are discretized using standard central differences on a staggered grid as shown in Fig. 2. As mentioned earlier, we will consider problems in which the macroscale flow fields are two-dimensional flows. We denote the two components of the velocity field by u and w. In the staggered grid, u is defined at $\left(x_i, z_{j+\frac{1}{2}}\right)$, w is

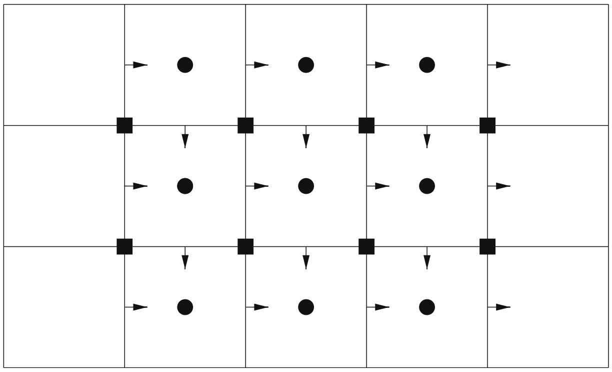

Fig. 2. A staggered grid for the discretization of the spatial derivatives in the macro-model. Let $\boldsymbol{u} = (u, w)$, then u is defined at $\left(x_i, z_{j+\frac{1}{2}}\right)$, w is defined at $\left(x_{i+\frac{1}{2}}, z_j\right)$, and p is defined at the $\left(x_{i+\frac{1}{2}}, z_{j+\frac{1}{2}}\right)$. The diagonals of the momentum flux are computed at the cell centers indicated by circles, and the off-diagonals are computed at the grid points indicated by squares.

defined at $\left(x_{i+\frac{1}{2}}, z_j\right)$, and p is defined at the center of each cell $\left(x_{i+\frac{1}{2}}, z_{j+\frac{1}{2}}\right)$. The diagonal terms of the momentum flux (τ_s plus the contribution of the convection) are defined at $\left(x_{i+\frac{1}{2}}, z_{j+\frac{1}{2}}\right)$, and the off-diagonal terms are defined at (x_i, z_j). The operators ∇ and Δ in Eqs. (68), (69) and (71) are discretized by standard central difference and the five-point formula respectively.

For the purpose of comparison, we first briefly describe the algorithm of HMM for this problem. More details can be found in Ref. [22]. In HMM, we first choose an initial value $\boldsymbol{u}^0$ for the macro-model and set $n = 0$. We then follow the following steps:

(1) Compute the velocity gradient $A^n = \nabla \boldsymbol{u}^n$ at each grid point where the stress is needed.
(2) Initialize an MD at each grid point.
(3) Solve each of the MD systems for M steps with a micro time step $\delta\tau$. Each MD is constrained by the local velocity gradient A^n through the boundary condition.
(4) Compute the stress τ_s^n from the MD results. This is done after sufficient number of relaxation steps using the formula (63). The stress is averaged over each MD box as in (66) and also over time to reduce the statistical fluctuation.
(5) Solve the macro-model for one macro step to obtain $\boldsymbol{u}^{n+1} = \boldsymbol{u}((n+1)\Delta t)$. This is done using the projection method (68), (69), (71) with a macro time step Δt and the stress computed from MD;
(6) Set $n := n + 1$, and go to step 1.

The seamless algorithm, on the other hand, works as follows. We start with some initial velocity field $\boldsymbol{u}^0$ for the macro-model. The macro-model is discretized on the staggered grid as shown in Fig. 2. At each grid point where the stress is needed, we initialize an MD system in accordance with the local velocity gradient $A^0 = \nabla \boldsymbol{u}^0$. Then we set $n = 0$ and proceed as follows:

(1) Compute the velocity gradient $A^n = \boldsymbol{u}^n$ at each grid point.
(2) Evolve each MD system and the equation for the MD box (65) for one micro time step $\delta\tau$. For the MD, periodic boundary condition is used with respect to the evolving box.
(3) Compute the stress τ_s^n from the MD results. Similar to HMM, the stress is averaged over the MD box.
(4) Update the macro-model by one macro step $\Delta' t$ to obtain $\boldsymbol{u}^{n+1} = \boldsymbol{u}((n+1)\Delta' t)$. This is done using the scheme in (68), (69), and (71) but with Δt replaced by a (smaller) time step $\Delta' t$.
(5) Set $n := n + 1$, and go to step 1.

Compared with HMM, the main difference of the above seamless algorithm is that the micro-solver (here the MD) runs continuously without reinitialization at each time step. The MD communicates with the macro-solver at each time step – the constraint (i.e. the velocity gradient) imposed on MD changes at each MD step while the instantaneous stress computed at each MD step is used to evolve the macroscale velocity field. In order for the MD to have sufficient time to relax and adapt to the new environment specified by the constraint, the imposed velocity gradient has to change slowly when viewed in the MD clock. This can be achieved by reducing the time step for the macro-model while keeping the MD time step fixed – the smaller the macro time step, the slower the velocity gradient changes when viewed in the MD clock. A small time step for the macro-model is also necessary in order to control the statistical error, as discussed earlier in Section 3 and demonstrated below in the numerical example. The simplest version of this algorithm for shear flows was introduced in [14].

Note that in the computation the MD box could become very skewed (though this did not happen in our computation for the driven cavity flow shown below). If this happens, the MD box has to be reset. The new box may be chosen based on the idea of reproducible lattice [25–27]. The new MD box is still cubic but will align in certain direction which is determined by the current local velocity gradient. This technique can also be used to choose the initial MD box.

4.3. Example: a driven cavity flow

As an example, we apply the above seamless algorithm to study the driven cavity flow. We consider two types of fluids: The first is composed of simple particles interacting via the Lennard–Jones potential, the second is a polymer fluid which at the microscale is modeled by the bead-spring model as described earlier in Section 4.1. Under normal conditions, the LJ fluid is Newtonian (i.e. the stress and the rate of strain have a linear relation) and the macro flow behavior can be accurately described by the Navier–Stokes (NS) equation. Therefore, we will regard the solution of the NS equation as the exact solution and use it as a benchmark for the multiscale method.

In the following, we will express all quantities in terms of LJ units. For example, the unit of length is σ; the unit of time is $\sigma\sqrt{m/\epsilon}$; for temperature it is ϵ/k_B where k_B is the Boltzmann constant, etc.

We consider the LJ fluid first. The fluid is confined in a square box which has a side length $L = 2000$. The flow is driven by the motion of the upper wall, which is sheared in the x-direction at the speed $U(t) = \min\{3 \times 10^{-4} t, 3\}$. The computational domain $[0, L]^2$ is covered by a uniform grid with 20×20 grid points. In the seamless multiscale method, the macro time step is taken as $\Delta' t = 0.5$. Each grid point is associated with one MD system which supplies the local stress. The MD is done in 3d at the temperature 1.1 and the density 0.81. Each system contains 1.2×10^4 particles. Initially these particles are put in a cubic box which has the side length 24.56. The box evolves according to the local velocity gradient. The MD time step is $\delta\tau = 0.005$.

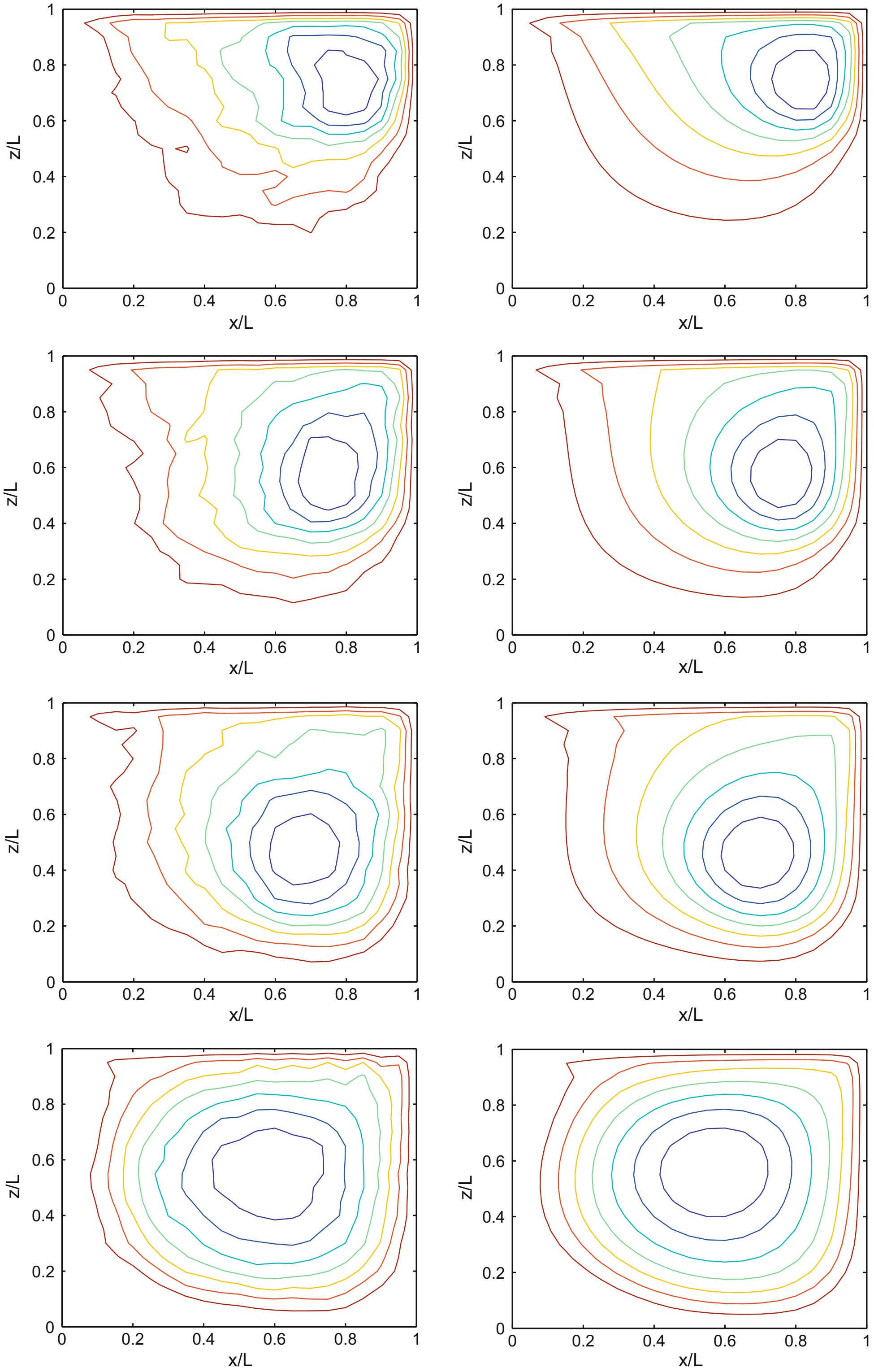

Fig. 3. Flow lines of a Lennard–Jones fluid in the driven cavity flow at different times: $t = 7.5 \times 10^3,\ 1.0 \times 10^4,\ 1.25 \times 10^4,\ 2.25 \times 10^4$ from top to bottom. The left column is the result of the seamless multiscale method, and the right column is the numerical solution of the Navier–Stokes equation.

In terms of real SI units, the size of our system is $L = 0.68$ micrometer, the maximal driving velocity is $U_m = 4.7 \times 10^2$ m/s. Obviously these values are a bit extreme and may not correspond to any realistic system. The only reason that we use these values is to have a large signal-to-noise ratio. In realistic systems, the signal (i.e. the needed data) is usually vanishingly small compared to the magnitude of the fluctuation. This imposes a severe difficulty when applying the multiscale method to these systems since it will require a long averaging time in HMM and a very small macro time step $\Delta' t$ (i.e. a large value of M) in the seamless algorithm. More efficient methods for overcoming this difficulty, such as advanced variance reduction techniques, will be left for future studies.

The numerical results are shown in Fig. 3. The first column of the figure shows the instantaneous flow lines obtained from the seamless multiscale method at several times. As a comparison, in the second column we show the numerical results of the NS equation at the same times. The parameters in the NS equation are measured from separate MD simulations of the LJ fluid at the same density and temperature. In particular the density is $\rho = 0.81$ and the measured viscosity is $\mu = 2.0$. From the figure we see that the result of the multiscale method agree very well with the solution of the NS equation.

To further assess the performance of the seamless method, we look at the velocity as a function of time at two locations. The two figures in Fig. 4 show the x-component of the velocity at the location (1000,1600) and (1500,500) respectively. The dashed curves are the solution of the seamless method. As a comparison, we also plot the solution of the NS equation (smooth solid curves). From the figures we see that the major difference of the two results is the fluctuation in the solution of the multiscale method. This is to be expected and is due to the statistical fluctuations in the stress tensor computed from MD. It can be improved in various ways, e.g. by employing ensemble average (i.e. many MD replica associated with each grid point), using larger MD system (consequently the instantaneous stress will be averaged over larger space, see (66)), or by reducing the macro time step (see the next example). Apart from the fluctuations, the multiscale result follows closely the solution of the NS equation.

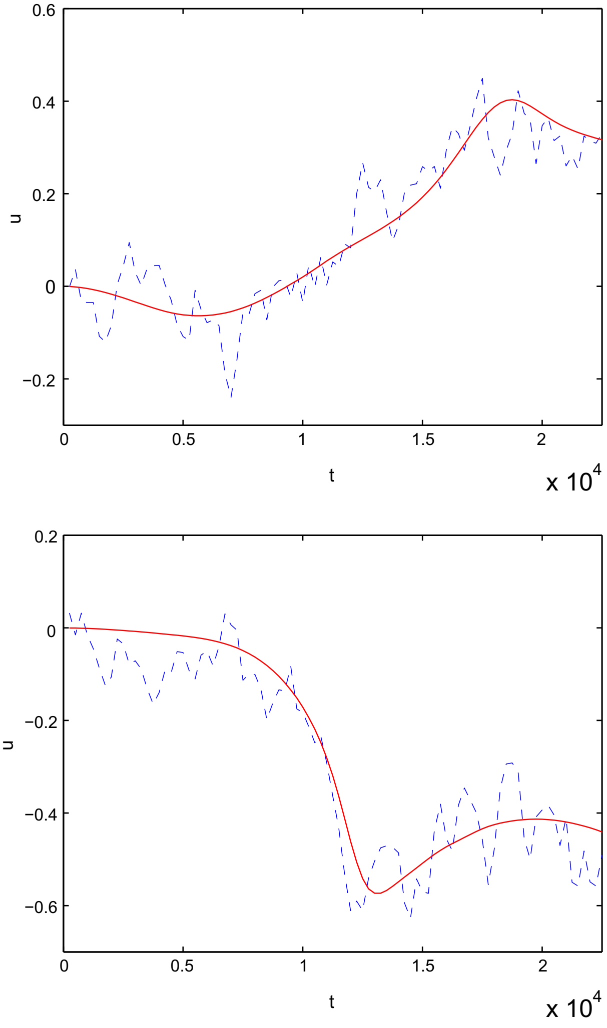

Fig. 4. The x-component of the velocity at the location (1000,1600) (upper panel) and (1500,500) (lower panel) as a function of time. The smooth solid curves are the solution of the NS equation; the dashed curves are the solution of the seamless multiscale method.

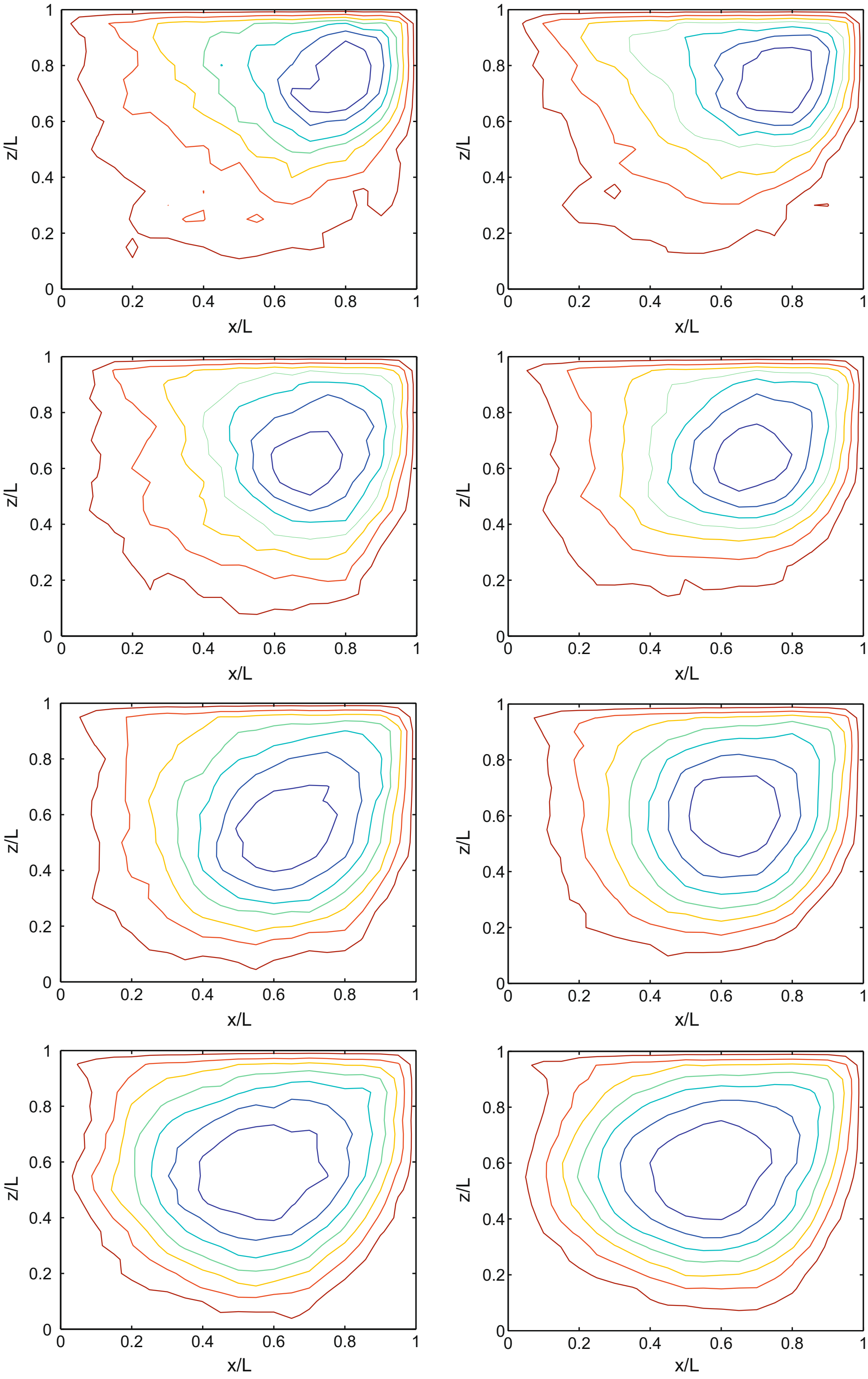

Fig. 5. Flow lines of a polymer fluid in the driven cavity flow at different times: $t = 7.5 \times 10^3,\ 1.0 \times 10^4,\ 1.25 \times 10^4,\ 2.0 \times 10^4$ from top to bottom. The left column is the result of the seamless multiscale method with the macro time step $\Delta' t = 0.5$; the right column is the result of the multiscale method with a smaller macro time step $\Delta' t = 0.25$;

Next we consider polymer fluids. The MD system at each grid point contains 1000 polymers; each polymer has 12 beads. The density of the beads is 0.81; the MD time step is 0.002. All the other parameters are the same as in the previous example for the LJ fluid. The numerical results are shown in Fig. 5. In this example, we used two different macro time steps: $\Delta' t = 0.5$ and $\Delta' t = 0.25$. The results are shown in the two columns respectively. These solutions are different from the Lennard–Jones fluid. In particular, the vortex in the LJ fluid travels further downward to the bottom of the box before reaching the steady state. A more qualitative comparison of the flows in a channel can be found in Ref. [14].

Comparing the two columns in Fig. 5, we see that their overall behavior agrees very well. We also see that the solution obtained using the smaller macro time step (the right column) has less fluctuations. This agrees with the discussion at the end of Section 3, namely, implicit averaging happens in the seamless algorithm and the stress tensor is implicitly averaged over time.

Given a macro time T, the MD simulation in the seamless algorithm is carried out for a total period of $T\delta\tau/\Delta' t$; therefore reducing the macro time step $\Delta' t$ while keeping the micro time step $\delta\tau$ fixed yields a longer MD simulation. This certainly increases the computational cost, but we gain accuracy – the numerical result has less statistical error. This is similar to HMM in which the statistical error can be reduced by increasing the value of the parameter M, the number of MD steps in one macro time step. This example shows that in the seamless algorithm, the same effect can also be achieved by reducing the macro time step. The difference is that in HMM one has to *explicitly* average the stress over time; but in the seamless method, this is unnecessary and the stress is averaged over time *implicitly*.

5. Type A problems

So far we have only discussed the so-called type B problems [4], for which the microscopic models are used to supply the constitutive relations, bypassing ad hoc modeling assumptions. We can also use the same kind of ideas for type A problems, for which the microscopic models are used to help resolving local singularities, defects, or other kinds of small regions of interest. In this case, we have some macroscopic models which are accurate enough over most of the computational domain, except around these local singularities or defects. For these kinds of problems, there are also several general strategies.

(1) Domain decomposition methods: The microscopic models are used on part of the computational domain, and the macroscopic models are used on the rest of the computational domain. These domains may or may not overlap. This is the most popular strategy, and it is hard to say when it got started.
(2) Algorithm refinement methods [28]: These are extensions of the mesh refinement methods, but with the additional option that more refined microscopic models can be used when needed. In particular, in this case, the whole computational domain is first covered by the macro-solver which is then refined locally.
(3) The heterogeneous multiscale methods [3,4]: The principle is the same as before, but now applied to a local region. In this case, the computational domain for the microscopic model has to be identified with the physical domain (or sub-domain) and it supplies the data needed in the macro-solver. If the time scale for the dynamics of the local region and the relaxation time scale for the microscopic model are comparable, then there is little difference with the adaptive model refinement approach. But if these time scales are separated, then within the HMM framework, one can naturally take advantage of the time scale separation, as was done for type B problems.

The seamless strategy discussed here provides yet another alternative. Like HMM it allows us to take advantage of the possible time scale separation.

Denote the whole physical domain of interest as Ω and on a sub-domain Ω_0, we would like to use a more refined microscopic model. Assume that $\tau_\varepsilon/t_M \ll 1$ where τ_ε and t_M are respectively the relaxation time of the microscopic model and the time scale for interesting dynamics in the local region. We then simply apply the strategy discussed in Section 2 on Ω_0. The computational savings come from the time scale separation: The microscopic solver runs on a fictitious time scale, even though it has a small time step, it has no obligation to cover the same amount of physical time as the macro-solver. If there is no time scale separation, we can set $M = 1$ and identify the fictitious time for the micro-solver with the physical time.

6. Conclusions

We have presented a new framework for designing multiscale methods. Compared with existing general strategies such as Brandt's systematic up-scaling, HMM and the "equation-free" approach, this new framework has the distinct advantage that it does not require reinitializing the microscale solver at every macro time step. At the same time, the accuracy and efficiency are generally comparable to that of HMM. We note, however, that a general drawback of all the on-the-fly multiscale methods is that it may be difficult to control the statistical error, to a point that systems of realistic sizes may be difficult to study at this point. Solving this problem will require the development of better estimators using e.g. variance reduction techniques. This will be the topic of future works.

Acknowledgments

W. E is supported in part by ONR Grant N00014-01-0674 and DOE Grant DE-FG02-03ER25587. W. Ren is supported in part by NSF Grant DMS-0604382, DMS-0806401 and the Sloan fellowship. E. Vanden-Eijnden is supported in part by NSF Grants DMS02-39625 and DMS07-08140, and by ONR Grant N00014-04-1-0565.

References

[1] A. Brandt, Multi-level adaptive solutions to boundary value problems, Math. Comput. 31 (1977) 333.
[2] A. Brandt, Multiscale scientific computation: review 2001, in: T.J. Barth et al. (Eds.), Multiscale and multiresolution methods: theory and applications, Yosemite Educational Symposium Conf. Proc., 2000, vol. 20, Lecture Notes in Comp. Sci. and Eng., Springer-Verlag, 2002, pp. 3–96.
[3] W. E, B. Engquist, The heterogeneous multi-Scale methods, Commun. Math. Sci. 1 (2003) 87.
[4] W. E, B. Engquist, X. Li, W. Ren, E. Vanden-Eijnden, Heterogeneous multiscale methods: a review, Commun. Comput. Phys. 3 (2007) 367.
[5] E. Vanden-Eijnden, Numerical techniques for multiscale dynamical systems with stochastic effects, Commun. Math. Sci. 1 (2003) 385.
[6] I.G. Kevrekidis, C.W. Gear, J.M. Hyman, P.G. Kevrekidis, O. Runborg, C. Theodoropoulos, Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis, Commun. Math. Sci. 1 (2003) 715.
[7] R. Car, M. Parrinello, Unified approach for molecular dynamics and density-functional theory, Phys. Rev. Lett. 55 (1985) 2471.
[8] E.B. Tadmor, M. Ortiz, R. Phillips, Quasicontinuum analysis of defects in crystals, Philos. Mag. A 73 (1996) 1529.
[9] J. Knap, M. Ortiz, An analysis of the quasicontinuum method, J. Mech. Phys. Solids 49 (2001) 1899.
[10] B. Perthame, F. Coron, Numerical passage from kinetic to fluid equations, SIAM J. Numer. Anal. 28 (1991) 26.
[11] W. E, J. Lu, Seamless multiscale modeling via dynamics on fiber bundles, Commun. Math. Sci. 5 (2007) 649.
[12] I. Fatkullin, E. Vanden-Eijnden, A computational strategy for multiscale systems with applications to Lorenz 96 model, J. Comput. Phys. 200 (2004) 605.
[13] L. Maragliano, E. Vanden-Eijnden, A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations, Chem. Phys. Lett. 426 (2006) 168.
[14] W. Ren, Seamless multiscale modeling of complex fluids using fiber bundle dynamics, Commun. Math. Sci. 5 (2007) 1027.
[15] M. Tuckerman, B. Berne, G. Martyna, Reversible multiple time scale molecular dynamics, J. Chem. Phys. 97 (1992) 1990.
[16] A. Lew, J.E. Marsden, M. Ortiz, M. West, Asynchronous variational integrators, Arch. Ration. Mech. Anal. 167 (2003) 85.
[17] A. Abdulle, W. E, Finite difference heterogeneous multi-scale method for homogenization problems, J. Comput. Phys. 191 (2003) 18.
[18] W. E, Analysis of the heterogeneous multiscale method for ordinary differential equations, Commun. Math. Sci. 1 (2003) 423.
[19] B. Engquist, Y.-H. Tsai, Heterogeneous multiscale methods for stiff ordinary differential equations, Math. Comput. 74 (2005) 1707.
[20] W. E, D. Liu, E. Vanden-Eijnden, Analysis of multiscale methods for stochastic differential equations, Commun. Pure Appl. Math 58 (2005) 1544.
[21] J.H. Irving, J.G. Kirkwood, The statistical mechanical theory of transport processes IV, J. Chem. Phys. 18 (1950) 817.
[22] W. Ren, W. E, Heterogeneous multiscale method for the modeling of complex fluids and micro fluidics, J. Comput. Phys. 204 (2005) 1.
[23] D. Frenkel, B. Smit, Understanding Molecular Simulation: From Algorithms to Applications, second ed., Academic Press, 2001.
[24] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745.
[25] A.M. Kraynik, D.A. Reinelt, Extensional motions of spatially periodic lattices, Int. J. Multiphase Flow 18 (1992) 1045.
[26] B.D. Todd, P.J. Daivis, Nonequilibrium molecular dynamics simulations of planar elongational flow with spatially and temporally periodic boundary conditions, Phys. Rev. Lett. 81 (1998) 1118.
[27] A. Baranyai, P.T. Cummings, Steady state simulation of planar elongation flow by nonequilibrium molecular dynamics, J. Chem. Phys. 110 (1999) 42.
[28] A.L. Garcia, J.B. Bell, W.Y. Crutchfield, B.J. Alder, Adaptive mesh and algorithm refinement using direct simulation Monte Carlo, J. Comput. Phys. 154 (1999) 134.