The Heterogeneous Multiscale Method and the "Equation-free" Approach to Multiscale Modeling

Weinan E
Princeton University

Contents

1 Introduction **3**

2 Examples of Multiscale Methods **5**
2.1 The Car-Parrinello molecular dynamics . . . 5
2.2 The quasicontinuum method . . . 7
2.3 The kinetic scheme . . . 9

3 The Renormalization Multi-grid Methods **10**
3.1 The basic philosophy . . . 11
3.2 The general strategy . . . 12

4 The "Equation-free" Approach **12**
4.1 The main components of the "equation-free" approach . . . 12
4.2 Coarse projective integration . . . 13
4.3 Gap-tooth scheme . . . 14
4.4 Patch dynamics . . . 15

5 The Heterogeneous Multiscale Methods **16**
5.1 The main components of HMM . . . 16
5.2 A unified view of some existing multiscale algorithms . . . 19
5.3 Modifying traditional algorithms to deal with multiscale problems . . 20
5.4 Stability and accuracy . . . 23

6 Discussions **24**
6.1 Comparison between the three approaches . . . 24
6.2 Difficulties with RMG . . . 27
6.3 Difficulties with HMM . . . 27
6.4 Difficulties with the "equation-free" approach . . . 28
6.5 What are projective integrators? . . . 30
6.6 "Equation-free" or "equation-based"? . . . 32

7 Conclusion 32
7.1 Top-down vs. bottom-up coupling 33
7.2 Multi-grid style of coupling vs. seamless coupling 34
7.3 Sequential vs. concurrent coupling 34

Abstract

In many problems of multiscale modeling, we are interested in capturing the macroscale behavior of the system with the help of some accurate microscale models, bypassing the need of using empirical macroscale models. This paper gives an overview of the recent efforts on establishing general strategies for designing such algorithms. After reviewing some important classical examples, the Car-Parrinello molecular dynamics, the quasicontinuum method for modeling the deformation of solids and the kinetic schemes for gas dynamics, we discuss three attempts that have been made for designing general strategies: Brandt's renormalization multi-grid method (RMG), the heterogeneous multiscale method (HMM) and the "equation-free" approach. We will discuss the relative merits and difficulties with each strategy and we will make an attempt to clarify their similarities and differences. We end with some perspectives about this kind of approach.

1 Introduction

In many areas of science and engineering, we face the problem that on one hand, we do not have an sufficiently explicit and accurate macroscopic model for the macroscale quantities that we are interested in, but 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 accurate electronic structure model such as 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 a constitutive relation which we often do not have. Instead, we have an accurate atomistic model, such as molecular dynamics. The same situation exists for solids.

Many different methods have been developed to deal with such problems. Most well-known among these methods are the Car-Parrinello molecular dynamics [4], the quasicontinuum method for studying the deformation of solids [43, 31] and the kinetic scheme for studying gas dynamics [8]. All these methods share the following features:

1. They allow us to model the macroscale quantities of interest, by coupling with a microscale model instead of using ad hoc macroscale models.

2. They make use of scale separation, by either modifying some small parameters in the problem (as is done in the Car-Parrinello method), or by solving the microscopic model on small spatial-temporal domains (as in the Knap-Ortiz version of the quasicontinuum method [31] and the Prendergast-Xu version of the kinetic scheme [46]).

The success of these methods and the success of more traditional multiscale methods such as the multi-grid method has given impetus to establishing some general framework for multiscale methods [3, 9, 11, 27]. The hope is that as was the case of finite difference and finite element methods for solving differential equations, the general framework might lead to general designing principles for multiscale methods and general guidelines for carrying out error analysis. In [3], Achi Brandt reviewed a general strategy for extending the multi-grid method to multi-physics problems and problems with scale separation. The new strategy in principle allows the use of atomistic models such as Monte Carlo methods or molecular dynamics at the finest level. This strategy does not require explicit macroscale models to begin with. In fact, Brandt remarked that one might be able to construct the effective macro 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. Brandt remarked that "few sweeps are enough, due to the fast CMC (conditional Monte Carlo) equilibration. This fast equilibration also implies that the interpolation can be done just over a restricted *subdomain*, serving as *window*: In the window interior fine-level equilibration is reached." As in traditional multi-grid methods, RMG follows an "interpolation-equilibration-restriction" strategy, except that in the "equilibration" step, microscopic models are used. In addition, the macro and micro grid sizes might be very far apart.

The philosophy of the "equation-free" approach is very similar, except for an added "extrapolation step" for the time evolution of the macro variables, making it a "lifting - equilibration (run microscopic model) - restriction - extrapolation" procedure. In space, the "equation-free" approach also makes use of the fact that the microscopic simulation can be done on small windows, or the "teeth" as it is called in the "gap-tooth" scheme. The gaps in between are filled in using interpolation.

In the heterogeneous multiscale method [9, 11], one begins with an assumption about the form of the macro 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 used during the computation to supply whatever data that are missing from the macro model. Scale separation is exploited by observing that in the data estimation step, the computational domain for the microscopic model is totally decoupled from the physical domain for the macroscale solver, and 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 that are carried out for estimating data at different locations at the macroscale computational domain. 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 [12].

There are obvious similarities between all these approaches. It is interesting to see whether they are different and how different they are. We will make an attempt to address these questions in the present paper.

Before ending the introduction, let us remark that the most popular way of incorporating microscale information to macroscale models is to use *sequential coupling* or *precomputing*: The missing parameters or coefficients or a predetermined macroscale model are computed beforehand using the microscopic model and standard numerical techniques are then used to solve the macroscale model. For example, for elliptic homogenization problems, we know that the macroscale model should be of the form:

$$\nabla(A(\mathbf{x})\cdot\nabla)U(\mathbf{x}) = f(\mathbf{x})$$

But we usually do not know the coefficients A. This, however, can be computed beforehand using the microscale model. This is a very popular strategy in many areas of scientific modeling. This paper will focus mostly on concurrent coupling techniques [1].

2 Examples of Multiscale Methods

2.1 The Car-Parrinello molecular dynamics

In Car-Parrinello molecular dynamics (CPMD) [4], the macroscale quantities of interest are the positions and velocities f the nuclei, which obey Newton's second law

$$M_I\ddot{\mathbf{R}}_I = -\nabla_{\mathbf{R}_I}V$$

Here M_I and $\mathbf{R}_I$ are respectively the mass and position of the I-th atom. The unknown component of the model is the inter-atomic potential V or the force field $-\nabla_{\mathbf{R}_I}V$. We assume that we have at our disposal a sufficiently accurate electronic structure model, such as the density functional theory model. Since the atoms interact via Coulomb forces, we should be able to compute the force on the nuclei once we know the electron structure. Car and Parrinello devised a very elegant way of doing this, i.e. coupling molecular dynamics with electronic structure models [4].

First some remarks about the electronic structure model. In principle, we could start with the quantum many-body problem which would be the true first principle. In practice, one often makes various one-electron approximations. Most successful among these approximate models is the Kohn-Sham density functional theory [34], in which the N-body wavefunction is replaced by the Kohn-Sham orbitals $\{\psi_n\}_{n=1}^N$ that minimize the energy functional

$$E_{KS}\{\mathbf{R}_I, \psi_n\} = \sum_{n=1}^{N}\int \psi_n^*(\mathbf{r})(-\frac{1}{2}\Delta)\psi_n(\mathbf{r})d\mathbf{r} + J[\rho] + E_{xc}[\rho] \tag{1}$$

where $\rho(\mathbf{r}) = \sum_{n=1}^N |\psi_n(\mathbf{r})|^2$ is the electron density,

$$J[\rho] = \frac{1}{2}\int\int \frac{(\rho(\mathbf{r}_1) - m(\mathbf{r}_1))(\rho(\mathbf{r}_2) - m(\mathbf{r}_2))}{|\mathbf{r}_1 - \mathbf{r}_2|}d\mathbf{r}_1 d\mathbf{r}_2$$

E_{xc} is the exchange-correlation functional, usually given by some empirically derived local functional of ρ. $m(\mathbf{r})$ is the potential that represents the nuclei. Usually it takes the form:

$$m(\mathbf{r}) = \sum_I m_I^a(\mathbf{r} - \mathbf{R}_I)$$

where m_I^a is the potential for the I-the nucleus. The electronic structure of a material or a molecule can be obtained by minimizing the functional above, subject to the constraint that the orbitals are orthonormal

$$\int \psi_n^*(\mathbf{r})\psi_m(\mathbf{r})d\mathbf{r} = \delta_{mn} \tag{2}$$

Going back to the original problem, namely the dynamics of the nuclei, the most obvious idea for coupling nuclei dynamics with electronic structure analysis is via the Born-Oppenheimer approximation: Since the nuclei is much heavier than the electrons, one may assume that the electronic structure is in the ground state given by the nuclei. Under this approximation, the electronic structure is slaved by the state of the nuclei:

$$M_I \ddot{\mathbf{R}}_I = -\nabla_{\mathbf{R}_I} E_{KS}\{\mathbf{R}_I, \psi_n\} \tag{3}$$

$$\{\psi_n, n = 1, 2, \cdots, N\} = \operatorname{argmin} E_{KS} \tag{4}$$

A natural way of implementing such a Born-Oppenheimer dynamics is:

1. Select an integrator for the molecular dynamics equation (3), for example, the Verlet scheme.

2. Calculate the force on the nuclie by solving the electronic structure problem (4) using some iterative method. To solve the electronic structure problem at each time step, one has to: (A) initialize the iterative procedure, (B) run the iterative procedure until it converges and then (C) compute the force on the nuclei, i.e. the right hand side of (3).

We will see later that such a procedure is very much reminiscent of the heterogeneous multiscale method.

Instead of following strictly the Born-Oppenheimer dynamics, Car and Parrinello developed a much more seamless approach. They defined the extended phase space for both the nuclei positions and the Kohn-Sham orbitals, and introduced the extended Lagrangian:

$$L\{\mathbf{R}_I, \psi_n, \dot{\mathbf{R}}_I, \dot{\psi}_n\} = \frac{1}{2}\sum_I M_I |\dot{\mathbf{R}}|_I^2 + \frac{1}{2}\sum_n \mu \int \dot{\psi}_n^2(\mathbf{r}) d\mathbf{r} - E_{KS}\{\mathbf{R}_I, \psi_n\} \tag{5}$$

where μ is the "mass" for the Kohn-Sham orbitals. Car-Parrinello molecular dynamics (CPMD) is obtained by following standard procedures in classical mechanics for this this Lagrangian:

$$\begin{aligned} M_I \ddot{\mathbf{R}}_I &= -\nabla_{\mathbf{R}_I} E_{KS} \\ \mu \ddot{\psi}_n &= -\frac{\delta E}{\delta \psi_n^*} + \sum_m \Lambda_{nm} \psi_m \end{aligned} \tag{6}$$

Here the $\Lambda_{m,n}$'s are the Lagrange multipliers for the orthogonality constraint (2). This formulation has the advantage that the electrons and nuclei are treated in the same footing.

So far we have only considered the multi-physics aspect of CPMD. There is also a multi-scale aspect, and that is associated with the disparity between the time scales for the nuclei and the electrons. The natural choice for the value of the parameter μ should be the value of the electron mass, which is at least three orders of magnitude smaller than the nuclei mass. However, since we are only interested in the dynamics of the nuclei, we may use other value of μ as long as we still obtain an accurate enough approximation for the dynamics of the nuclei. The Born-Oppenheimer approximation means setting the value of μ to 0. Car and Parrinello adopted an opposite strategy, which is often more convenient in practice, namely increasing the value of μ to some fictitious value much larger than the electron mass. The actual value is determined by the accuracy requirement.

CPMD is still the most compelling example of concurrent coupling techniques: Since the inter-atomic potential is in principle a function of all the atomic positions, it is unfeasible to obtain this function by precomputing (i.e. sequential coupling), and it has to be calculated "on-the-fly". Many other examples in this category (which are called type B problems in [11]) do not share this feature, particularly in light of the efficient sequential coupling techniques proposed in [21].

2.2 The quasicontinuum method

Our next example is the (local) quasicontinuum (QC) method [43, 31]. In this example our interest is on the macroscale deformation of solids. This can be done by solving a variational problem for the displacement field $\mathbf{u}, \Omega \rightarrow R^3$, where Ω is the domain that defines the undeformed position of the solid:

$$\min \int_{\Omega} (W(\nabla \mathbf{u}(\mathbf{x})) - \mathbf{f}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}))\, d\mathbf{x} \tag{7}$$

$\mathbf{f}$ is the applied force. This requires knowing W, the stored energy density. Traditionally W is obtained from empirical considerations. The main purpose of QC is to bypass such empirical strategies, and instead rely on some sufficiently accurate atomistic models.

To begin with, QC starts with a piecewise linear finite element space on a coarse mesh. The mesh is generated by selecting the so-called representative atoms (rep-atom), identified by $\{\mathbf{x}_\alpha\}$ via their undeformed positions. The displacement of the rest of the atoms are determined by the trial functions in the finite element space. The key question is how to compute the energy of a trial function. Two different approaches have been proposed to deal with this problem.

The first is to use the so-called Cauchy-Born rule. Since the deformation gradient $\mathbf{A} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ is uniform within each element, one may approximate the energy on each element using the energy density of a crystal which is uniformly deformed with the deformation gradient $\mathbf{A}$ on that element. This is the Cauchy-Born rule. Denote by the $\mathcal{E}(\mathbf{A})$ the strain energy density obtained using the Cauchy-Born rule, the approximate total energy is then obtained by summing over the elements,

$$E_{tot} \approx \sum_{k=1}^{N_e} \mathcal{E}(\mathbf{A}_k)|\Omega_k| \tag{8}$$

where N_e is the number of elements, $|\Omega_k|$ denotes the volume of the k-th element.

The second is to compute the energy associated with each representative atoms by performing direct summation of the interatomic potential over a small cluster of atoms around the rep-atom. The total energy is computed approximately using:

$$E_{tot} \approx \sum_{\alpha=1}^{N_{rep}} n_\alpha E_\alpha \tag{9}$$

where E_α is the approximate energy associated with the rep-atom indexed by α, $\{n_\alpha\}$ is a set of suitably chosen weights. Roughly speaking, n_α should be the number of atoms $\mathbf{x}_\alpha$ is supposed to represent.

To compute E_α, one reconstructs the positions of atoms in a small cluster that surrounds $\mathbf{x}_\alpha$, and then perform direct summation in this small cluster (see the attachd figure). This is the Knap-Ortiz version of the quasi-continuum method [31].

To summarize, the Knap-Ortiz QC procedure works as follows:

1. Given a candidate trial function in the finite element space, use interpolation to find the position of the atoms in a small cluster around each rep-atom.

2. Using the atomistic model to find the force acting on the rep-atoms by summing the interaction force over the small clusters of atoms.

3. Update the trial function.

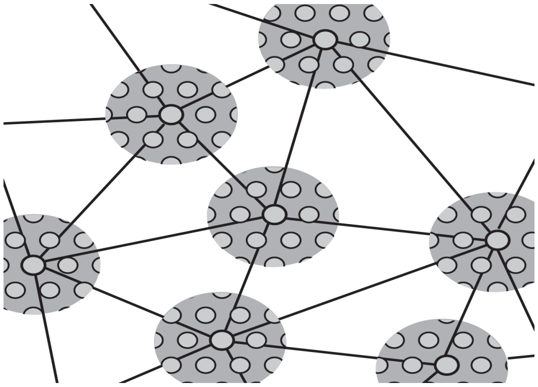

Figure 1: Schematic illustration of the cluster summation rule in QC (courtesy of M. Ortiz). Only atoms in the small cluster need to be visited during the computation.

We have only discussed one component (the simpler one) of QC. Another important component of QC is to use adaptive mesh refinement to resolve the atomistic features near defects (type A problems). Since this is not the kind of questions discussed in this paper, we will omit that part.

2.3 The kinetic scheme

Our next example is the derivation of numerical schemes for gas-dynamics that uses only the kinetic model. Such schemes are called kinetic schemes (see for example [40, 36, 35, 46, 8], see also the related work on lattice Boltzmann methods [5, 41]). Here the macroscopic quantities of interest are the density, pressure and velocity fields of the gas. We will base our construction on a kinetic model, such as the Boltzmann equation:

$$\partial_t f + \mathbf{v} \cdot \nabla f = \frac{1}{\varepsilon} C(f) \tag{10}$$

where f is the one-particle phase-space distribution function, which is also our microscale state variable; $C(f)$ is the collision kernel; ε is the mean-free path between collisions in the gas. The macroscale state variables U are the usual hydrodynamic variables of mass, momentum and energy densities, which are related to the microscale state variable f by:

$$\rho = \int f d\mathbf{v}, \quad \rho\mathbf{u} = \int f\mathbf{v} d\mathbf{v}, \quad E = \int f \frac{|\mathbf{v}|^2}{2} d\mathbf{v}. \tag{11}$$

From the Boltzmann equation, we have:

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ E \end{pmatrix} + \nabla \cdot \mathbf{F} = 0 \tag{12}$$

where

$$\mathbf{F} = \int_{\mathbb{R}^3} f \begin{pmatrix} \mathbf{v} \\ \mathbf{v} \otimes \mathbf{v} \\ \frac{1}{2}|\mathbf{v}|^2\mathbf{v} \end{pmatrix} d\mathbf{v} \tag{13}$$

When $\varepsilon \ll 1$, the distribution function f is close to local equilibrium states, or the local Maxwellians,

$$M(\mathbf{x}, \mathbf{v}, t) = \frac{\rho(\mathbf{x}, t)}{(2\pi\theta(\mathbf{x}, t))^{3/2}} \exp\Big(-\frac{(\mathbf{v} - \mathbf{u}(\mathbf{x}, t))^2}{2\theta(\mathbf{x}, t)}\Big) \tag{14}$$

with θ being the absolute temperature.

We will focus on the one-dimensional case. We first divide the computational domain in the physical space into cells of size Δx. We will denote by x_j the center position of the j-th cell, and $x_{j+1/2}$ the cell boundary between the j-th and $j+1$-th cells. For a first order method, we represent the solution as piece-wise constants, i.e.

$$\mathbf{U}_j = (\rho_j, \rho_j u_j, E_j), \quad x \in (x_{j-1/2}, x_{j+1/2}]$$

The finite volume scheme takes the form:

$$\begin{cases} \rho_j^{n+1} - \rho_j^n + \dfrac{\Delta t}{\Delta x}\big(F_{j+1/2}^{(1)} - F_{j-1/2}^{(1)}\big) = 0, \\ (\rho u)_j^{n+1} - (\rho u)_j^n + \dfrac{\Delta t}{\Delta x}\big(F_{j+1/2}^{(2)} - F_{j-1/2}^{(2)}\big) = 0, \\ E_j^{n+1} - E_j^n + \dfrac{\Delta t}{\Delta x}\big(F_{j+1/2}^{(3)} - F_{j-1/2}^{(3)}\big) = 0 \end{cases} \tag{15}$$

where $\mathbf{F}_{j+1/2} = (F_{j+1/2}^{(1)}, F_{j+1/2}^{(2)}, F_{j+1/2}^{(3)})^T$ is the numerical flux at the cell boundary $x_{j+1/2}$.

The fluxes $\mathbf{F}_{j+1/2}$ will be computed by solving the kinetic equation using the following three-step procedure:

1. Initialize the kinetic equation using the local Maxwellian with parameters (ρ, u, θ) given by $\mathbf{U}_j$.

2. Solve the kinetic equation locally around cell boundaries where the values of the fluxes are needed.

3. Using (13) to compute the numerical fluxes.

This gives rise to the following expression for the numerical fluxes:

$$\mathbf{F}_{j+1/2} = \mathbf{F}_{j+1/2}^{+} + \mathbf{F}_{j+1/2}^{-}, \quad \text{with } \mathbf{F}_{j+1/2}^{\pm} = \int_{\mathbb{R}^{\pm}} f(x_{j+1/2}^{\mp}, v, t) \begin{pmatrix} v \\ v^2 \\ \frac{1}{2}v^3 \end{pmatrix} dv, \tag{16}$$

$$\mathbf{F}^{\pm} = \begin{pmatrix} \rho u A^{\pm}(S) \pm \dfrac{\rho}{2\sqrt{\pi\beta}} B(S) \\ (p + \rho u^2) A^{\pm}(S) \pm \dfrac{\rho u}{2\sqrt{\pi\beta}} B(S) \\ (pu + \rho u e) A^{\pm}(S) \pm \dfrac{\rho}{2\sqrt{\pi\beta}} (\dfrac{p}{2\rho} + e) B(S) \end{pmatrix} \tag{17}$$

where

$$A^{\pm} = \frac{1 + \mathrm{erf}(S)}{2}, \quad B(S) = e^{-S^2}, \quad S = \frac{u}{\sqrt{2RT}}, \quad p = \rho RT.$$

This is the simplest kinetic scheme. The schematic is shown in Figure 2.

3 The Renormalization Multi-grid Methods

Given the success of these different multiscale methods, it is natural to ask whether we can formulate general designing principles or general guidelines. In the case of solving differential equations, such general designing principles have proved to be very useful, e.g., for finite difference and finite element methods. In the past several years,

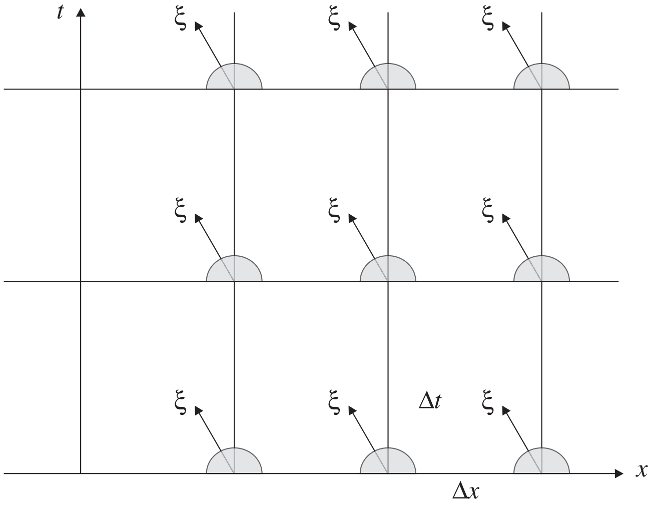

Figure 2: Schematics for the derivation of kinetic scheme: A finite volume method is imposed in the $x - t$ domain, and the kinetic equation is solved (e.g. analytically) over the shaded region to give the fluxes needed in the finite volume method. The ξ axis (which should have been v) indicates the extra velocity variable in the kinetic model, which represents the microstructure for the present problem.

several attempts have been made on constructing general framework for multiscale methods. In the following, we will discuss some examples of such general strategies, the extension of multi-grid method by Brandt, the heterogeneous multiscale method, and the "equation-free" approach.

Please note that what we call the "renormalization multi-grid" here should be what Brandt is now calling "systematic upscaling". "Renormalization multi-grid" is a term that Brandt used earlier. In any case, the object we will discuss below is found in Brandt's review article [3].

3.1 The basic philosophy

One of the first attempts for constructing a general framework for multiscale modeling was the extension of the multi-grid method by Achi Brandt during the 1990's. In its original form [2, 28], multi-grid method is an efficient way of solving the algebraic equations obtained from discretizing partial differential equations (PDE). The objective was to find accurate solutions of the PDE. Brandt extended this traditional notion of multi-grid method in a number of directions:

1. The models or processes at the different scales may be different in nature. For example, the process at the finest scale may be discrete, modeled by kinetic Monte Carlo methods or molecular dynamics, whereas at the macroscale, the process may be continuous.

2. Instead of resolving the detailed processes at the finest scale, in RMG, we are often interested only in capturing the large scale behavior. This is particularly useful in cases when closed-form macroscale models are not available. In that case, one may be able to reconstruct the effective macroscale model at the end of the such multiscale computation: "At sufficient coarse level, this entire algorithm effectively produces *macroscopic 'equations'* for the simulated system .... This can yield a macroscopic numerical description for the fluid even for those cases where the traditional derivation of closed-form differential equations is inapplicable."

3. One may take advantage of the scale separation by limiting the microscopic simulation at the fine scale to small spatial-temporal domains, As was remarked by Brandt: "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 *subdomain*, serving as *window*: In the window interior fine-level equilibration is reached."

3.2 The general strategy

Brandt described a general strategy implemented in the spirit of the multi-grid method. Each macro cycle consists of the following steps:

1. *Interpolation*: The current values of the local macro state variables are used to initialize the micro model.

2. *Equilibration*: Iterate the microscale model over small windows for a few sweeps.

3. *Restriction (projection)*: Project the micro variables back to the macro grid.

Brandt also described applications to many different areas, including electronic structure analysis, solving integro-differential equations, modeling high frequency wave propagation, Monte Carlo methods in statistical mechanics, complex fluids and image processing.

4 The "Equation-free" Approach

4.1 The main components of the "equation-free" approach

The "equation-free" approach is another framework to address the same kind of questions based on a similar philosophy. It consists of a set of techniques including coarse bifurcation analysis, projective integrators, the gap-tooth scheme and the patch dynamics. At an abstract level, "equation-free" is built on a three-stage "lift-evolve-restrict" coarse time-stepper procedure [27] very similar to the "interpolate-equilibrate-restrict" procedure in RMG: Given a macroscopic state, $U(t)$, at some time t, the coarse time-stepper consists of the following basic elements [27]:

1. *Lift*: Transform the initial data through lifting to one or more consistent microscopic realizations.

2. *Evolve*: Use the microscopic simulator (the detailed time-stepper) to evolve these realizations for the desired short macroscopic time τ, generating the value(s) $u(\tau, t)$.

3. *Restrict*: Obtain the restriction of u and define the coarse time-stepper solution as $\mathcal{T}_c^\tau U(t) = \mathcal{M}u(\tau, t)$.

Here $\mathcal{M}$ is the restriction operator that maps micro-state variables to macro-state variables.

Computational savings come from exploiting scale separation in the problem using interpolation and extrapolation techniques.

4.2 Coarse projective integration

In coarse projective integrators, one generates an ensemble of microscopic solutions for short times, by running the microscopic solver with an ensemble of initial data that are consistent with the current macro-state. One computes the average values of the coarse variables over this ensemble. The time derivatives for the coarse variables are computed using these averaged values and these coarse time derivatives are used to extrapolate the coarse variables over a much larger time step. “The dynamic information in the replica runs $\psi_i(t; \psi_0)$ can also be used to extrapolate toward longer times. Instead of propagating each of the replicas, we extrapolate the *averaged* position of the slow variable, for instance linearly (exploiting regularity of the expected coarse dynamics with time),

$$\overline{\psi(t'; \psi_0)} \approx \overline{\psi(t; \psi_0)} + \frac{t' - t}{t}[\overline{\psi(t; \psi_0)} - \psi_0]$$

A long ‘projective step’ $t' - t$ is then effected by reinitializing an ensemble at the extrapolated value; this is the simplest ‘projective forward Euler method’ ” (from [30]). In the same way, one can construct coarse molecular dynamics.

More specifically, the projective integration procedure is as follows.

1. At each macro time step, say the n-th time step, use the microscale model to obtain an approximation of the time derivative of the coarse variables by running the microscale model for short period of time δt, and (ensemble) average the micro-states obtained. Denote the result by $\tilde{U}^n_{\delta t}$.

2. Extrapolate to obtain the numerical solution for the coarse variable at the next time step, $\{U_j^{n+1}\}$:

$$U^{n+1} = U^n + \Delta t \frac{\tilde{U}^n_{\delta t} - U^n}{\delta t} \tag{18}$$

or more generally:

$$U^{n+1} = U^n + \Delta t \frac{\tilde{U}^n_{\delta t} - \tilde{U}^n_{\alpha\delta t}}{(1-\alpha)\delta t} \tag{19}$$

where α is some numerical parameter between 0 and 1.

This is also the basis for the coarse molecular dynamics developed in [30].

In the case of stiff ordinary differential equations (ODEs), for which projective integrators were initially developed [22], one would simply solve the whole system with say, forward Euler for a number of steps using small time steps that resolve the fast component of the dynamics, compute the time derivative by taking divided difference of the last two steps, and use the result to extrapolate the whole system over a macro time step. There is no need to distinguish slow and fast variables explicitly. Higher order methods can be constructed using higher order extrapolation, as described in [22]. In this context, very similar ideas were proposed in [20]. In general, the extrapolation step is only done for the macro variables.

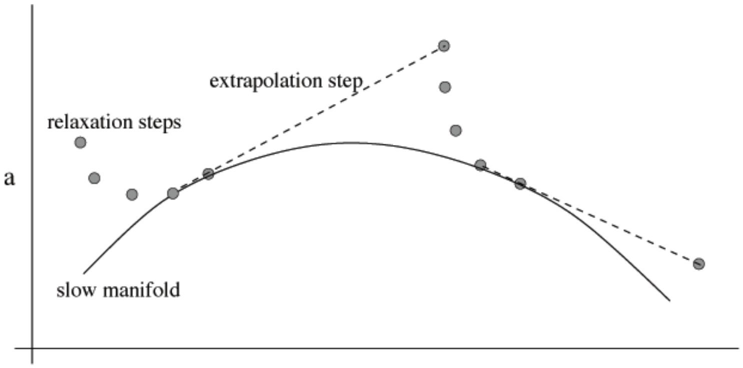

Figure 3: Schematic illustration of the projective integration scheme

4.3 Gap-tooth scheme

The basic idea of the gap-tooth scheme is to “use the microscopic rules themselves, in smaller parts of the domain and, through computational averaging within the subdomains, followed by interpolation, we evaluate the coarse field $U(t, x)$, the timestepper, and the time derivative field over the entire domain” ([27], page 729).

“Given a finite dimensional representation $\{U^N_j\}$ of the coarse solution (e.g. nodal values, cell averages, spectral coefficients, coefficients for finite elements or empirical basis functions) the steps of the gap-tooth scheme are the following.

1. *Boundary conditions.* Construct boundary conditions for each small box based on the coarse representation $\{U^N_j\}$.

2. *Lift.* Use lifting to map the coarse representation $\{U_j^N\}$ to initial data for each small box.

3. *Evolve.* Solve the detailed equation (3.1) for time $t \in [0, \tau]$ in each small box $y \in [0, h] \equiv [x_j - h/2, x_j + h/2]$ with the boundary conditions and initial data given by steps (1) and (2).

4. *Restrict.* Define the representation of the coarse solution at the next time level by restricting the solutions of the detailed equation in the boxes at $t = \tau$." ([27], page 730).

An interesting application of the gap-tooth scheme is presented in [23] in which a particle model was used to capture the macroscale dynamics of the viscous Burgers equation. This is done by performing particle simulation on an array of small boxes (teeth) together with a macroscopically interpolative mechanism for communication between the boxes at the particle level. Particles exiting one box are distributed into neighboring boxes and the original box. In this way, the microscopic simulations on the different boxes are all coupled and together they mimic a microscopic simulation performed over the whole domain.

4.4 Patch dynamics

Patch dynamics is a combination of the projective integrators and the gap-tooth scheme. It consists of the following (from [27], page 739):

1. *Short time steps.* Repeat the gap-tooth time stepper, points (d) through (g), a few times. (Compute patch boundary conditions, lift to patches, evolve microscopically, restrict).

2. *Extrapolate.* Advance coarse fields a long time step into the future through projective integration. This first involves estimation of the time-derivatives for the coarse field variables, using the successively reported coarse fields in (h) followed by a large projective step, as discussed in Section 2.2.

A detailed analysis of the "equation-free" approach is somewhat complicated by the fact it has been continuously evolving. What began as an "extrapolation in time, interpolation in space" strategy [33] has gradually evolved to a "macro solver- micor solver - data estimator" strategy [24], which is the hallmark of HMM, as we will see below.

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

Figure 4: Schematics of HMM framework

5 The Heterogeneous Multiscale Methods

5.1 The main components of HMM

We now turn to the framework of the heterogeneous multiscale method (HMM). The general setting is as follows. We are given an accurate microscale model, which can be abstractly written as

$$f(u, b) = 0 \tag{20}$$

where u is the microscopic state variable and b is the set of constraints, such as boundary conditions. We are not interested in the microscopic details of u, but rather the macroscopic state of the system which we denote by U. Using our existing knowledgement about the problem, we make an assumption about the form of the macroscopic model:

$$F(U, D) = 0 \tag{21}$$

where D stands for the macroscopic data that are necessary to complete the model. For example, in models of complex fluids, D might be the stress.

The goal of HMM is to compute U using the assumed form of F together with the microscale model. It consists of two main components.

1. *A macroscopic solver.* Even though the macroscopic model is not available completely, we can still use whatever knowledge that is available on the form of F to select a suitable macroscale solver.

2. *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 some macroscale data is needed, perform a series of microscopic simulations which are constrained so that they are consistent with the local macroscopic state, i.e.

$b = b(U)$. When formulating the constraints, one also uses the assumption about the form of the macroscale model, i.e. the form of F.

(b) *Data processing*: Use the microscale data generated from the microscopic simulations to extract the needed macroscale data.

The significance of the choosing the right macroscale solver can be appreciated from the following:

1. Even if we know completely the macroscale model in closed form, it may still be a non-trivial matter to select a suitable numerical algorithm for the macroscale model.

2. More importantly, the macroscale solver allows us to target the specific form of the macroscale model. If we write the macroscale model in the form:

$$U_t = F(U)$$

the "equation-free" approach is based on computing U_t. It specifically makes the point of avoiding the computation of F [] or using the specific structure of F . In the case when the macroscale model is a deterministic ODE, knowing U_t is equivalent to knowing F, and this difference between HMM and "equation-free" may not be that significant. This is also the reason why "equation-free" can work in this case. But if F has a more interesting structure, e.g. F might be the sum of a drift term and a noise term for the case when the macroscale model is a stochastic ODE, or F might represent a hyperbolic system, then thinking about computing U_t alone may not be sufficient. One needs to think about the whole macroscale solver. In the case of stochastic ODEs, this means that one has to treat the drift and diffusion terms separately. In the case of hyperbolic systems, this means that one has to use stable schemes.

Let us take the example of complex fluids in which the macroscale model is a continuum model for the macroscale velocity field U in the form of:

$$\partial_t \mathbf{U} + (\mathbf{U} \cdot \nabla)\mathbf{U} + \nabla P = \nabla \cdot \tau$$

$$\nabla \cdot \mathbf{U} = 0$$

These are simply statements of conservation of momentum and mass. The unknown data is the stress τ. We will assume that

$$\tau = \tau(\nabla \mathbf{U})$$

This key assumption will be used in formulating constraints on the microscale model.

Let us say that the micro model is a molecular dynamics model:

$$m_i \ddot{\mathbf{r}}_i = \mathbf{f}_i, \quad i = 1, 2, \cdots, N \tag{22}$$

Here $m_i, \mathbf{r}_i$ are respectively the mass and position of the i-th particle, $\mathbf{f}_i$ is the force acting on the i-th particle.

Given that the macroscale model is in the form of an incompressible fluid equation, we will use the projection method as the macro-solver [6]. In the implementation of the projection method, we will need the values of τ at the appropriate grid points. These are the data that need to be estimated. Since we have assumed that τ is only a function of $\nabla \mathbf{U}$, we will constraint the molecular dynamics so that the average strain rate is given by the value of $\nabla \mathbf{U}$ at the relevant grid point. In general, implementating such constraints is the most difficult step in HMM. For the present example, a possible strategy is discussed in [37].

From the results of the microscale model, we can extract the needed value of stress using the Irving-Kirkwood formula:

$$\begin{aligned}\tilde{\tau}(\mathbf{y}, t; \mathbf{x}) = \; & -\sum_i (m_i \mathbf{v}_i \otimes \mathbf{v}_i) \delta(\mathbf{r}_i - \mathbf{y}) \\ & -\frac{1}{2} \sum_{j \neq i} ((\mathbf{r}_i - \mathbf{r}_j) \otimes \mathbf{f}_{ij}) \int_0^1 \delta\left(\lambda \mathbf{r}_i + (1-\lambda)\mathbf{r}_j - \mathbf{y}\right) d\lambda\end{aligned} \tag{23}$$

where $\mathbf{v}_i = \dot{\mathbf{r}}_i - A\mathbf{r}_i$ is the thermal velocity of the i-th particle, $\mathbf{f}_{ij}$ is the force acting on the i-th particle by the j-th particle.

In summary, three main ingredients are required in using HMM:

1. a macro-solver, here the projection method;
2. a micro-solver, here the constrained molecular dynamics;
3. a data estimator, here the Irving-Kirkwood formula.

Next we consider the application of HMM to ODEs with multiple time scales. Consider the system [44]

$$\begin{cases} \dot{X}^\varepsilon = f(X^\varepsilon, Y^\varepsilon, \varepsilon), & X^\varepsilon(0) = x \\ \dot{Y}^\varepsilon = \dfrac{1}{\varepsilon} g(X^\varepsilon, Y^\varepsilon, t/\varepsilon, \varepsilon), & Y^\varepsilon(0) = y \end{cases} \tag{24}$$

where $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$. Here the macroscale variable of interest is X. We will assume that its dynamics is approximated accurately by an ODE of the form:

$$\dot{\bar{X}} = B(\bar{X}), \qquad \bar{X}(0) = x \tag{25}$$

Since the macroscale equation takes the form of an ODE, we will select stable ODE solvers as the macro-solver. This suggests a HMM that consists of the following:

1. The macro-solver:

$$\hat{X}_{n+1} = \hat{X}_n + \hat{B}(\hat{X}_n, \varepsilon) \Delta t, \qquad \hat{X}_0 = x, \tag{26}$$

where Δt is the macro-time-step, and $\hat{B}(x, \varepsilon)$ denotes the approximate value of $B(x)$ which needs to be estimated at each macro-time-step.

2. The micro-solver: This is obtained by considering the second equation in (24) with $x = \hat{X}_n$. In the simplest case of a forward Euler scheme with micro-time-step $\Delta\tau$, we have

$$\hat{Z}^{\varepsilon}_{n,m+1} = \hat{Z}^{\varepsilon}_{n,m} + g(\hat{Z}^{\varepsilon}_{n,m}, m\Delta\tau, \varepsilon)\Delta\tau. \tag{27}$$

   By considering only the second equation in (24), we automatically constrain the macro variable X.

3. An estimator: In the simplest case, we may use time-averaging:

$$\hat{B}(\hat{X}_n, \varepsilon) = \frac{1}{M}\sum_{m=0}^{M-1} f(\hat{X}_n, \hat{Z}^{\varepsilon}_{n,m}, \varepsilon). \tag{28}$$

   Of course, more sophisticated data processing techniques can be used [18].

If the ODE has a stochastic component, then the macroscale model might be in the form of stochastic ODEs. In this case, one should use a stochastic ODE solver as the macro-solver. The simplest example is the Euler-Maruyama scheme:

$$\hat{X}_{n+1} = \hat{X}_n + \hat{B}(\hat{X}_n, \varepsilon)\Delta s + \hat{\sigma}(\hat{X}_n, \varepsilon)\Delta W_n, \tag{29}$$

The data that need to be estimated are $\hat{B}$ and $\hat{\sigma}$. This can be done in a similar way as described above. The detailed formulas can be found in [44].

5.2 A unified view of some existing multiscale algorithms

HMM was partly motivated by the desire to provide a unified view for some of the existing multiscale algorithms. This is partly successful. Indeed the presentation in section 2 is already influenced by the HMM viewpoint. Let us now reexamine the algorithms in section 2 from a HMM perspective.

The Born-Oppenheimer dynamics can now be viewed as a special case of HMM in which the macro solver is the molecular dynamics algorithm chosen for (3), the data to be estimated is the forces acting on the nuclei, and data estimation is done through an iterative algorithm for the electronic structure model. For local QC, the macro solver is the piecewise linear finite element method, the data to be estimated is the energy associated with the trial function or the forces on the representative atoms. Data estimation can be done either using Cauchy-Born rule or by summing the inter-atomic potential over small clusters of atoms around the representative atoms. For the kinetic scheme, the macro solver is the finite volume method. The data to be estimated are the numerical fluxes and data estimation is done by solving the kinetic equation locally around the point of interest.

Considering the fact that these algorithms are proposed for drastically different applications, this unified view is interesting. It also offers some substances.

1. One may use the framework for analyzing the stability and accuracy of HMM to analyzing the accuracy of these existing methods. An example is done in [14] for analyzing the local QC.

2. HMM provides an alternative perspective on how to improve these existing methods. For example, if we are interested in developing high order kinetic schemes, we might start with a higher macro solver and solve the microscale model to higher order accuracy. This may not be that different from what experts in kinetic schemes would do anyway. But it does make the ideas a bit more systematic.

There is one very notable exception though, and that is the Car-Parrinello molecular dynamics. As we will discuss at the end of this paper, most algorithms discussed in this paper use a "multi-grid" style of macro-micro coupling, i.e. the "interpolation-equilibration-projection" procedure, except CPMD which is much more seamless. Indeed extending the seamless coupling procedure in CPMD is one of the motivation for developing the general seamless coupling strategy in [17].

5.3 Modifying traditional algorithms to deal with multiscale problems

HMM can be viewed as a philosophy for modifying traditional numerical algorithms for the purpose of handling efficiently multiscale problems. To illustrate this point, we will discuss two examples.

The FMM-HMM. Fast multipole method (FMM) is among the most effective algorithm for evaluating the Coulomb potential due to a distribution of charges [26]. Consider the following problem:

$$\phi(\mathbf{x}) = \int_\Omega \frac{q(\mathbf{y}, \frac{\mathbf{y}}{\varepsilon})}{|\mathbf{x}-\mathbf{y}|} d\mathbf{y} \tag{30}$$

Here q is a smooth function which is periodic in the second variable with period $I = [0,1]^d$ and $\varepsilon << 1$. Problems of this type arise in analyzing the electronic structure of solids [13]. Applying FMM directly to the evaluation of ϕ will require a cost of $O(\varepsilon^{-d})$ in d dimension, since the smallest boxes used in FMM should resolve the smallest scale of q, which is of $O(\varepsilon)$. A much more efficient approach is to combine FMM with HMM [29].

In the spirit of HMM, we select FMM as the macroscale solver. The data that are needed in FMM are the coefficients of multipole expansion:

$$M^p_{k,j} = \int_{C_{k,j}} q(\mathbf{y}, \frac{\mathbf{y}}{\varepsilon})(\mathbf{y} - \mathbf{x}_{k,j})^p d\mathbf{y}$$

where $(C_{k,j}, \mathbf{x}_{k,j})$ =j-th (box, box-center) at the k-th level. They can be approximately evaluated using, for example:

$$M^p_{k,j} \simeq \int_{C_{k,j}} \int_I q(\mathbf{y}, \mathbf{z})(\mathbf{y} - \mathbf{x}_{k,j})^p d\mathbf{y} d\mathbf{z}$$

In this way, the total cost is $O(1)$.

The approach described above is not necessarily the most efficient or accurate way of evaluating ϕ defined in (30). Nevertheless, this example serves as a good illustration of how to modify traditional algorithms for multiscale problems, based on the HMM philosophy.

Finite element HMM. Consider an elliptic equation with multiscale coefficients:

$$\begin{cases} -\operatorname{div}\big(a^\varepsilon(\mathbf{x})\nabla u^\varepsilon(\mathbf{x})\big) = f(\mathbf{x}) & \mathbf{x} \in D \subset \mathbb{R}^d, \\ u^\varepsilon(\mathbf{x}) = 0 & \mathbf{x} \in \partial D. \end{cases} \tag{31}$$

Here ε is a small parameter that signifies explicitly the multiscale nature of the coefficient $a^\varepsilon(\mathbf{x})$: It is the ratio between the scale of the microstructure and the scale of the physical domain D.

We are interested in capturing the macroscale behavior of the solution to this problem. Abstract homogenization theory tells us that the macroscale component of the solution satisfies an effective equation of the form:

$$-\operatorname{div}\big(A(\mathbf{x})\nabla U(\mathbf{x})\big) = f(\mathbf{x}) \qquad \mathbf{x} \in D \tag{32}$$

where $A(\mathbf{x})$ is the effective coefficient at the macroscale. If these coefficients are explicitly known, we may apply the finite element method directly to (32). In cases when these coefficients are not explicitly known, we may proceed as follows, in the spirit of HMM.

As macro-solver, we use the standard finite element method. As the simplest choice, we will use the standard C^0 piecewise linear element, on a triangulation T_H where H denotes the element size. We will denote by X_H the finite element space. The size of the elements should resolve the macroscale computational domain D, but they do have to resolve the small scales.

The needed data is the stiffness matrix for the finite element method:

$$\mathcal{A} = (A_{ij}) \tag{33}$$

where

$$A_{ij} = \int_D (\nabla\Phi_i(\mathbf{x}))^T A_H(\mathbf{x}) \nabla\Phi_j(\mathbf{x})\, d\mathbf{x} \tag{34}$$

and $A_H(\mathbf{x})$ is the effective coefficient (say conductivity) at the scale H, $\{\Phi_i(\mathbf{x})\}$ are the standard nodal basis functions of X_H. Had we known $A_H(\mathbf{x})$, we could have

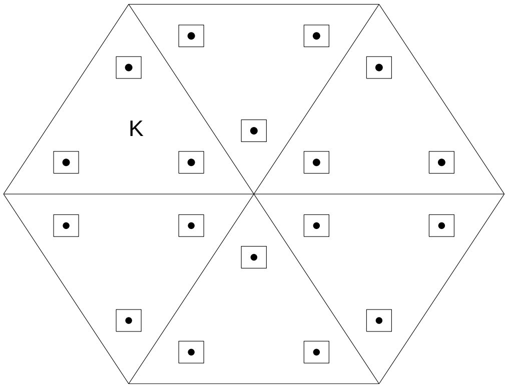

Figure 5: Illustration of HMM for solving (31). The dots are the quadrature points. The little squares are the microcell $I_\delta(x_\ell)$.

evaluated (A_{ij}) by numerical quadrature. Let $f_{ij}(\mathbf{x}) = (\nabla\Phi_i(\mathbf{x}))^T A_H(\mathbf{x}) \cdot \nabla\Phi_j(\mathbf{x})$, then

$$A_{ij} = \int_D f_{ij}(\mathbf{x}) d\mathbf{x} \simeq \sum_{K \in T_H} |K| \sum_{\mathbf{x}_\ell \in K} w_\ell f_{ij}(\mathbf{x}_\ell) \tag{35}$$

where $\{\mathbf{x}_\ell\}$ and $\{w_\ell\}$ are the quadrature points and weights respectively, $|K|$ is the volume of the element K. Therefore the data that we need to estimate are the values of $\{f_{ij}(\mathbf{x}_\ell)\}$. This will be done by solving the original microscopic model, properly reformulated, locally around each quadrature point $\{\mathbf{x}_\ell\}$, as shown in Figure 5.3.

For the micro-solver, let φ_i^ε be the solution of the following problem

$$-\mathrm{div}(a^\varepsilon(\mathbf{x})\nabla\phi^\varepsilon(\mathbf{x})) = 0 \quad \text{on } I_\delta(\mathbf{x}_\ell) \tag{36}$$

with boundary condition

$$a^\varepsilon(\mathbf{x})\frac{\partial\phi^\varepsilon}{\partial n} = \lambda^T \hat{\mathbf{n}} \quad \text{on } \partial I_\delta(\mathbf{x}_\ell) \tag{37}$$

where λ is the Lagrange multiplier for the constraints that $\frac{1}{\delta^d}\int_{I_\delta(\mathbf{x}_\ell)} \nabla\phi^\varepsilon \, d\mathbf{x} = (\nabla\Phi_i)(\mathbf{x}_\ell)$, $\hat{\mathbf{n}}$ is the outward normal on $\partial I_\delta(\mathbf{x}_\ell)$. Alternatively, we may use the periodic boundary condition: $\phi^\varepsilon(\mathbf{x}) - \Phi_i(\mathbf{x})$ is periodic with period $I_\delta(\mathbf{x}_\ell)$.

From the solution to the microscale problem, we estimated the needed data $f_{ij}(\mathbf{x}_\ell)$ by

$$f_{ij}(\mathbf{x}_\ell) \simeq \frac{1}{\delta^d}\int_{I_\delta(\mathbf{x}_\ell)} (\nabla\varphi_i^\varepsilon(\mathbf{x}))^T a^\varepsilon(\mathbf{x}) \nabla\varphi_j^\varepsilon(\mathbf{x}) \, d\mathbf{x} \tag{38}$$

Knowing $\{f_{ij}(\mathbf{x}_\ell)\}$, we obtain the stiffness matrix $\mathcal{A}$ by (35).

5.4 Stability and accuracy

Because HMM is a macro-solver-based philosophy, it comes with a nice framework for carrying out stability and accuracy analysis. Of course, for this purpose we have to limit ourselves to cases when we do have good analytical control of the effective macroscale model. The basic idea, as explained in [9], is to compare the HMM solution with the solutions of the selected macroscale solver for the effective macroscale model. Their difference is caused by the fact that the HMM solution contains an additional error due to the fact that some quantities in the macroscale model are estimated from microscale models, not from some explicit analytical model. This new error term is called the HMM error, denoted by $e(\mathrm{HMM})$. Under fairly general conditions (basically the stability of the macro-solver), one can prove [9]

$$\|U - U_{\mathrm{HMM}}\| \leq C\big(H^k + e(\mathrm{HMM})\big) \tag{39}$$

where U is the solution of the effective macroscale model, H is the macroscale grid size. The first term on the right hand side is the conventional error of the macroscale solver applied on an explicit macroscale model. The second term is the additional error term due to the HMM procedure.

The key in getting concrete error estimates and thereby giving guidelines in constructing multiscale methods lies in the estimation of $e(\mathrm{HMM})$. However, this is very specific to each problem.

For the elliptic problem discussed above,

$$e(\mathrm{HMM}) = \max_{\substack{\mathbf{x}_\ell \in K \\ K \in \mathcal{T}_H}} \|A(\mathbf{x}_\ell) - \tilde{A}(\mathbf{x}_\ell)\|,$$

where $\tilde{A}(\mathbf{x}_\ell)$ is the estimated coefficient at $\mathbf{x}_\ell$ using (35) and (38), $A(\mathbf{x}_\ell)$ is the coefficient of the effective model. We have

Theorem 1. Assume that (31) and (32) are uniformly elliptic. *Denote by U_0 and U_{HMM} the solution of* (32) *and the* HMM *solution, respectively. If U_0 is sufficiently smooth, then there exists a constant C independent of ε, δ and H, such that*

$$\|U_0 - U_{\mathrm{HMM}}\|_1 \leq C\big(H^k + e(\mathrm{HMM})\big), \tag{40}$$

$$\|U_0 - U_{\mathrm{HMM}}\|_0 \leq C\big(H^{k+1} + e(\mathrm{HMM})\big). \tag{41}$$

Here $\|u\|_1$ and $\|u\|_0$ are standard H^1 and L^2 norms of u respectively.

This result is completely general. It does not require specific assumptions on the structure of the coefficients in (31). However, to estimate $e(\mathrm{HMM})$ quantitatively, we need to make specific assumptions on the structure of the coefficients in (31). In general, if we assume that $a^\varepsilon(\mathbf{x})$ is of the form $a(\mathbf{x}, \mathbf{x}/\varepsilon)$, then we have

$$|e(\mathrm{HMM})| \leq C\left(\frac{\varepsilon}{\delta}\right)^\alpha \tag{42}$$

where δ is the size of the computational domain for the microscale problem. The value of the exponent α depends on the rate of convergence of the relevant homogenization problem as well as the boundary conditions used in the microscopic problem. Some interesting results are obtained in [16] but there are still many open questions.

6 Discussions

6.1 Comparison between the three approaches

It is clear from these descriptions that there are many similarities between RMG, HMM and the "equation-free" approach.

1. They all evolve the macro state variable with the help of the micro model.

2. They all involve mapping between the macro and micro states, even though they use different terminologies (see Table 1). Incidently, "lifting" is an unfortunate terminology, since in multiscale modeling, we typically place macroscale models above, and microscale models below. Using "lifting" this way gives the wrong sense of direction. (In a similar context, it should be noted that "lifting schemes" are used in wavelet analysis for computing wavelet transforms from smaller to larger scales [42]).

3. They all make use of scale separation by restricting the simulation of the microscale model on small domains (boxes, windows) for short times (or few sweeps).

Moreover, the basic structure for RMG is remarkably close to that of the "equation-free" approach, as can be seen in Table 2. Note in particular that the purpose of the "evolution" step in "equation-free" is to equilibrate. The main new component added by the "equation-free" approach is the extrapolation step: Computing time derivative of the macro variables from microscale models and use that to extrapolate the macro variables for a large time step.

	Macro to micro	*micro to Macro*
RMG	interpolation	restriction (projection)
HMM	reconstruction	compression
Equation-free	lifting	restriction

Table 1: A comparison between the terminologies used in RMG, HMM and "equation-free".

RMG	*"Equation-free"*
Interpolation	Lifting
Equilibration	Evolutions (equilibration)
Restriction (projection)	Restriction
	Extrapolation

Table 2: A comparison between the basic structure of RMG and "equation-free".

Are there any differences? Even though the details of RMG are sometimes unclear, it is clear that they are based on a multi-grid strategy. Whether there are differences between HMM and "equation-free" depends on which version of "equation-free" we consider. The motivating question for HMM has been: How can we best integrate our knowledge about both the macro model and the micro model? At the initial stage, the focus of the "equation-free" approach was quite the opposite: What can we do if we do not have any explicit knowledge about the macro and micro model? Indeed one of the main motivation for "equation-free" was to handle legacy codes. HMM has always been a "top-down" approach. Its starting point is a macro model and a macro-solver. In other words, HMM is more of an "equation-based " technique, even though it does not require all the details about the equation. The original description of "equation-free" seemed more of a "bottom-up" strategy. It tries to "fool" the microscopic model to perform macroscopic tasks. One clear example is the gap-tooth scheme in [23] discussed earlier. The main thrust there was to couple the different microscopic simulations on small boxes in order to mimic a microscopic simulation performed over the entire physical domain. However, recent versions of "equation-free" are becoming closer and closer to HMM. In fact, as we will see later, the new "projective integration scheme" in [24] is exactly HMM.

To be precise, let us consider a simple ODE example:

$$\frac{dx}{dt} = f(x,y), \quad \frac{dy}{dt} = -\frac{1}{\varepsilon}(y - \phi(x)) \tag{43}$$

This is the case for which "equation-free" and HMM are the closest. Here clearly x is the slow variable, y is the fast variable. The effective macroscale model should be of the form:

$$\frac{dx}{dt} = F(x) \tag{44}$$

At each macro time step, 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:

   $$y^{n,m+1} = y^{n,m} - \frac{\delta t}{\varepsilon}(y^{n,m} - \varphi(x^n)) \tag{45}$$

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

3. Force-estimator:
$$F^n = \frac{1}{M}\sum_{m=1}^{M} w_{m,n} f(x^n, y^{n,m}) \tag{46}$$
where $\{w_{m,n}\}$ is a set of weight coefficients. To facilitate comparison with "equation-free", let us simply take:
$$F^n = f(x^n, y^{n,M}) \tag{47}$$
4. Evolve macro-solver:
$$x^{n+1} = x^n + \Delta t F^n \tag{48}$$

Combining (47) and (48), we get:

$$x^{n+1} = x^n + \Delta t f(x^n, y^{n,M}) \tag{49}$$

One macro step of "equation-free" would proceed as follows:

1. Initialize the micro-solver, now the whole system (43), e.g. $x^{n,0} = x^n, y^{n,0} = y^{n-1,M}$.
2. Apply the micro-solver:
$$x^{n,m+1} = x^{n,m} + \delta t f(x^{n,m}, y^{n,m}), \quad y^{n,m+1} = y^{n,m} - \frac{\delta t}{\varepsilon}(y^{n,m} - \varphi(x^n)) \tag{50}$$
for $m = 0, 1, \cdots, M-1$.
3. Restrict: This means computing the values of the macroscopic variables that corresponds to $\{(x^{n,m}, y^{n,m})\}$. In this case, this simply means taking the x-component.
4. Extrapolate: A first order extrapolation is simply
$$x^{n+1} = x^n + (\Delta t - M\delta t)\frac{x^{n,M} - x^{n,M-1}}{\delta t} = x^n + (\Delta t - M\delta t) f(x^{n,M-1}, y^{n,M-1}) \tag{51}$$

The main difference in this version of HMM and "equation-free" differ mainly in whether the micro solver is constrained and where f is evaluated. However, if the microscale model is more complicated, for example, if the small scale feature is oscillatory in nature or has stochastic fluctuations, then the treatment of HMM and "equation-free" would be quite different:

1. HMM uses a suitably chosen macroscale solver according to the expected form of the macro model; "equation-free" uses polynomial interpolation [27].

2. HMM uses sophisticated averaging techniques in the micro solver to extract the needed macroscale data, whereas “equation-free” uses ensemble averaging to obtain the macro variables at the micro time steps to be used for extrapolation.

One clear example is the case when the y-component of the equation is changed to

$$\frac{dy}{dt} = -\frac{i}{\varepsilon}(y - \phi(x)) \tag{52}$$

In this case, one can still develop very accurate HMM techniques for capturing the dynamics of the slow component [18]. It is not clear at all what one can do with “equation-free”.

It is also interesting to note that since “equation-free” is based on a time-stepper strategy, it is not clear how to apply it to purely static problems. For example, it is not clear how to develped “equation-free” based QC or finite element methods for elliptic problems without adding artificial dynamics.

6.2 Difficulties with RMG

While the general philosophy is quite clear and very attractive, there are still questions about how these ideas should be implemented for concrete practical problems. The most extensively discussed example in [3] is the example of using atomistic models to capture the macroscale behavior of complex fluids. However, even for this example, it is not clear what the details of the algorithms are. This does not necessarily mean that there are essential difficulties in applying Brandt’s ideas. It is simply an indication that algorithmic and practical issues can be quite significant and require a lot of efforts.

6.3 Difficulties with HMM

The most significant shortcoming of HMM is that it is based on a presumed macroscale model. If the assumed form is incorrect, one can not 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, the stochastic component of the macroscale solution will not be captured by the HMM based on such an assumption. Because of this, HMM does not completely avoid making ad hoc assumptions about the macroscopic model.

For practical problems of interest, we often have accumulated some knowledge about how the macroscale model should be like. Such information can be used when making an assumption 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 behavior of the solution in the class of models considered. This philosophy bears some resemblence to that of the “optimal prediction” proposed by Chorin et al. [7]. In this sense, HMM is a way of addressing the question: What is the best we can do given the knowledge we have about the problem at all scales?

6.4 Difficulties with the "equation-free" approach

The most important problem with "equation-free" is the lack of a clear unifying principle. While RMG has been developed as an extension of the multi-grid method, HMM has been developed as a macro-solver-based top-down coupling procedure, there are no clear fundamental principles for "equation-free". At the present time, it stands more as a collection of papers, or the collected work of its developers.

Indeed, there are non-trivial logical inconsistencies in the "equation-free" papers:

1. The name "equation-free" suggests that it bypasses the need of using macroscale equations. This was clearly the intention in earlier papers and talks on "equation-free", and indeed the main selling point. (Recall that an earlier name for "equation-free" was "solving equations without equations"). However, as we will see below, not only the recent versions of "equation-free" clearly require the form of macroscale equations, as in HMM, a substantial number of the papers (for examples papers on free energy calculations such as [19], see below) simply did precomputing, i.e. sequential coupling, which are typical "equation-based" techniques and should have been the opposite philosophy of "equation-free". But they are still called "equation-free" in these papers.

2. How do we treat the boundary conditions for the microscale solver? This is commonly regarded as the most important technical issue in multiscale modeling. Two radically different proposals have been made in "equation-free" papers. The first is just to ignore the issue [38, 39]. The argument is that if we choose the microscale computational domains sufficiently large, then boundary conditions do not matter. The second proposal is to design boundary conditions that couple all the microscopic simulations in small boxes together, in order to mimic a microscopic simulation done on the whole domain. The only existing example of this kind is presented in [23]. This of course, makes the microscopic simulations much more complicated. We do not expect unified answers to the question of boundary conditions. The trouble with "equation-free" is that there is really no clear indication what the philosophy is supposed to be, and very few constructive suggestions on this important issue.

In the rest of this section, we will explain these points in detail.

A very important component of "equation-free" is the "lifting" operator. This is the analog of the interpolation operator in multi-grid method or the reconstruction operator in HMM. Their purpose is to reinitialize the microscale solver. In RMG and HMM, the only requirement on the reinitializd micro-states is consistency with the local macro-states. "Equation-free" requires a lot more for its reinitialization procedure, as we see below.

In [27, 38], the following lifting procedure was proposed: In the small domain around the macro grid point x_j, use the approximate Taylor expansion:

$$\tilde{u}_0(x) = \sum_{k=0}^{d} \frac{1}{k!} D_k (x - x_j)^k \tag{53}$$

Here D_k is some approximations to the derivatives of the macroscale profile at x_j, for example:

$$D_2 = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2}, \quad D_1 = \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x}, \quad D_0 = U_j^n - \frac{1}{24} h^2 D_2 \tag{54}$$

It was pointed out in [15] that if this lifting operator is used for studying the simple advection equation:

$$\partial_t u + \partial_x u = 0 \tag{55}$$

then patch dynamics gives rise to the following scheme:

$$U^{n+1} = U^n + \Delta t(-D_1 + \frac{1}{2} \delta t D_2) \tag{56}$$

Since $\delta t \ll \Delta t$, the last term is much smaller than the other terms, and we are left essentially with a scheme which is unstable under the standard CFL condition that $\Delta t \sim \Delta x$:

$$U^{n+1} = U^n - \Delta t D_1 \tag{57}$$

due to the central character of D_1.

Aside from the stability issue, there can also be problems with consistency. Consider the following example:

$$\partial_t u = -\partial_x^4 u \tag{58}$$

The macroscale model is obviously the same model. However, it is easy to see that if we follow the patch dynamics procedure with $d = 2$, we would be solving $\partial_t U = 0$, which is obviously inconsistent with the correct macroscale model.

These problems are fixed in the new version of patch dynamics [39]. In this new version, one assumes a macro model of the form:

$$\partial_t U = F(U, \partial_x U, \cdots, \partial_x^d U, t)$$

and chooses a "method-of-lines discretization" of this macro model:

$$\partial_t U_i = F(U_i, D^1(U_i), \cdots, D^d(U_i), t)$$

Here $D^k(U)$ is some suitable finite difference discretization of $\partial_x^k U$. For example, for advection equation, one should use some one-sided discretization. The operator D^k is then used in the lifting operator:

$$\bar{u}_\varepsilon^i(x, t^n) = \sum_{k=0}^{d} D_i^k(\bar{U}^n) \frac{(x - x_i)^k}{k!}, \quad x \in [x_i - \frac{H}{2}, x_i + \frac{H}{2}]$$

This version of "equation-free" approach does overcome the difficulties discussed above, but it also means that the reinitialization process not only has to take into account consistency with the local values of the macro variables, but also the some charateristics of the unknown macroscale model, such as the order of the macro equation, the direction of the wind if the macroscale model happen to be a first order equation; and we do not know what else needs to be taken into account. This is heavy lifting indeed.

This modification makes "patch dynamics" a strictly less useful tool than HMM. It has the same shortcomings as HMM, namely it relies on a presumed macro model and a macro solver. However, unlike HMM which has the freedom to use known macroscale information in the macro solver, the macroscale information in this patch dynamics is taken into account in a convoluted way through the lifting operator.

As an example of the difference between "equation-free" and HMM, let us consider how to capture macroscopic gas dynamics using molecular dynamics [11]. HMM would simply start with the macro model:

$$\partial_t U + \nabla \cdot \mathbf{F} = 0$$

where U is the mass, momentum and energy density, and use a finite volume method as the macro solver. In this case, one should choose a finite volume scheme that relies less on the details of the macro model. One possible choice is the Lax-Friedrichs scheme. The fluxes needed are then estimated from molecular dynamics, constrained so that the total mass, momentum and energy have the right values consistent with the local values of U. The fluxes can then be estimated using the Irving-Kirkwood formula. The overall strategy is quite similar to that of the kinetic scheme, except that the kinetic model is replaced by molecular dynamics. Numerical results can be found in [11].

It is not at all clear how to handle this problem in the "equation-free" approach. For one thing, it requires knowing the direction of the macroscale waves in the lifting step. It is also unclear how it can overcome the difficulties with shocks.

6.5 What are projective integrators?

The earlier papers on projective integration [22, 30, 27] made it clear that projective integrators are based on the idea of extrapolation. Time derivatives for the coarse variables are computed using the microscopic solver and used to extrapolate the coarse variables for large time steps. Indeed the first paper on projective integrators stated that [22]: "The reader might think that these should be called 'extrapolation methods', but that name has already been used [...]. Hence we call the proposed methods projective integration methods."

However, in a recent paper [24] dealing with stochastic ODEs with multiple time scales:

$$dx_t^\varepsilon = a(x_t^\varepsilon, y_t^\varepsilon)dt + b(x_t^\varepsilon, y_t^\varepsilon)dU_t \tag{59}$$

$$dy_t^\varepsilon = \frac{1}{\varepsilon}f(x_t^\varepsilon, y_t^\varepsilon)dt + \frac{1}{\sqrt{\varepsilon}}g(x_t^\varepsilon, y_t^\varepsilon)dV_t \tag{60}$$

the authors analyzed the following version of the "projective integration scheme":

1. "X_n is evolved in time by an Euler-Maruyama step,

   $$X_{n+1} = X_n + A(X_n)\Delta t + B(X_n)\Delta W_n$$

   where ΔW_n are Brownian displacements over a time interval Δt. We refer to (1.4) as the *macro-solver* (or, macro integrator)".

2. "The numerical solver used to generate the sequence Y_m^n is called the *micro-solver* (or micro-integrator). The simplest choice is again the Euler-Maruyama scheme,

   $$Y_{m+1}^n = Y_m^n + \frac{1}{\varepsilon}f(X_n, Y_m^n)\delta t + \frac{1}{\sqrt{\varepsilon}}g(X_n, Y_m^n)\Delta V_m^n$$

   where ΔV_m^n are Brownian displacements over a time interval δt".

3. "Having generated the trajectories Y_m^n, the functions $\bar{a}$ and $\bar{b}$ are estimated by

   $$A(X_n) = \frac{1}{M}\sum_{m=1}^{M} a(X_n, Y_m^n)$$

   $$B(X_n)B^T(X_n) = \frac{1}{M}\sum_{m=1}^{M} b(X_n, Y_m^n)b^T(X_n, Y_m^n)''$$

This "macro solver – micro solver –data estimator" strategy is clearly a HMM strategy, and is quite different from the extrapolation-based strategy of the original projective integration scheme. Indeed the algorithm discussed in [24] is basically the same as the one introduced in [44]. Throughout the paper, there is no mentioning of extrapolation. The analytical results obtained are still quite interesting but the misuse of the terminologies can easily lead to further confusion about what "equation-free" really is.

There is a reason why the authors abandoned the original extrapolation-based philosophy of projective integrators, and this has been pointed out in [15]. Basically if one follows the extrapolation strategy, then either the numerical solution converges to the wrong limit, or the numerical parameters have to have the right balance in which case there is no saving compared with solving the original problem using a brute force approach.

6.6 “Equation-free” or “equation-based”?

It is well-known that there are two class of techniques for multiscale coupling: Sequential coupling and concurrent coupling [1, 10]. In sequential coupling, one assumes a form of the macroscale model which contains some unknown components such as unknown coefficients, these are then obtained through the microscale model by precomputing. This is a typical “equation-based” multiscale technique that has been commonly used, perhaps ever since modeling became a series tool. At the very minimum, one would expect “equation-free” to be an alternative to such “equation-based” precomputing techniques. However, in many “equation-free” papers, particularly the ones on free energy calculations (see e.g. [19, 30]), what was done was to first precompute the drift $V(q)$ and diffusion coefficient $D(q)$ using microscopic simulations and then solve the Fokker-Planck equation

$$\partial_q(\partial_q[D(q)f(q)] - V(q)f(q)) = 0$$

to get the steady state (or equilibrium) probability distribution f and the free energy using $\Phi(q) = -k_B T \log f(q)$. This is still called “equation-free”, even though it is a standard precomputing procedure that has been around for many years. After all, this is not very different from precomputing the equation of state of a gas and then using Euler’s equation to study gas dynamics.

Aside from such extreme examples, as remarked earlier, the recent version of patch dynamics is also “equation-based”, i.e. one has to start with an assumed macro model and macro solver.

Some “equation-free” developers might argue that they do not need any preconceived notion of what the macroscale model might be like, they can extract such information using the microscale model, as is done in the “baby-bathwater scheme” [32], which is a scheme about finding the order of the highest derivative in the effective macroscale model using microscale simulation. This is an interesting idea, which is an extension of the well-known ideas that people have used to find out whether a system behaves convectively or diffusively by computing the effective diffusion constants. However, putting aside questions about whether the algorithms proposed in [32] actually work (see [15] for some discussions), there are some important philosophical issues. For example, in order to extract such macroscale information, one has to design a series of preconceived tests on the microscopic model. These preconceived tests have already limited the effective macroscale model to some preconceived form. Therefore, in the end, this approach is not very different from the “equation-based” approach described earlier.

7 Conclusion

We have seen that the general philosophies of RMG, HMM and the “equation-free” approach are all quite similar. In addition, the structure of RMG and “equation-free" is very close (see Table 2). The real advance made in "equation-free" over RMG is in coarse projective integration, i.e. the "extrapolation" step. For a large class of problems for which the effective macroscopic dynamics is deterministic, this is a simple but effective strategy for overcoming the numerical difficulty with time scale separation. As for spatial dependence, "equation-free" basically took the same viewpoint, except to say that the microscopic simulation is supposed to be done on small boxes. However, it is not clear how these microscopic simulations on small boxes are supposed to be set up. In this regard, it has not gone significantly beyond RMG.

The strategy of HMM is to put all these under a macroscale solver. Within HMM, the microscale simulations are independent and are indeed decoupled from the macroscopic computational domain: One can think of them as been done on some "virtual" space. The HMM strategy is not ideal: It relies on some assumpations about how the macroscale model should look like. These assumptions are not only used in selecting the macroscale solver, but also used in formulating the constraints on the microscopic solver. The HMM strategy represents a compromise: Ideally one would like to avoid completely such assumptions, but at the present time, that does not seem to be feasible in general.

7.1 Top-down vs. bottom-up coupling

For the class of problems we are interested in here, as long as we have selected the right set of macroscale variables, we can always write down the effective macroscale model in the form of

$$\partial_t U = F(\{U\}, x, t)$$

or some discrete analog. The real question is how much do we need to know about the specific form of F. One viewpoint, which seems to be the underlying principle of in the early "equation-free" papers with the exception of the papers on the baby-bathwater scheme [32], is that it is really not necessary to know the form of F, all we need is a way to compute U_t, and this can be done using the microscale model. It is not difficult to see that the "time-stepper" ideas, the coarse bifurcation methods, and the projective integrators are all developed along this line of thought. In space, the role of "extrapolation" is replaced by "interpolation". The initial idea of the "gap-tooth" scheme was to design an array of coupled microscale simulations on small boxes, linked together by interpolation and boundary conditions, to mimic a microscale simulation on the whole physical domain. This is indeed a "bottom-up" strategy, and it would be very interesting to see how far one can go with this and what class of problems can be handled with such a strategy.

HMM took the viewpoint that on one hand for most problems of interest, we have already gathered a lot of information about how F should be like; on the other hand, even if we do know the macroscale model completely, solving it numerically can be a very non-trivial task (think about numerical methods for solving nonlinear conservation laws). This motivated the macro solver-based HMM strategy, which is a "top-down" strategy. This top-down strategy is not just useful for HMM, it is also useful for other approaches such as the seamless coupling method proposed in [17].

7.2 Multi-grid style of coupling vs. seamless coupling

With the exception of the Car-Parrinello method, the methods we discussed in this note use the "multi-grid" style of coupling, by which we mean that they require converting explicitly between macro and micro states using the reconstruction (lifting, interpolation) and compression (projection, restriction) operators, at each macro cycle. While this is quite natural, it can become a rather difficult step in practice, particularly when reinitializing the microscale model given the values of the macro variables. This is no longer necessary in the new seamless strategy proposed in [17], which is more in the spirit of the Car-Parrinello method.

7.3 Sequential vs. concurrent coupling

What we have discussed so far are mostly concurrent strategies to multiscale modeling. Naturally they should be compared to the sequential strategies. The usual argument against sequential strategies are:

1. It requires a preconceived form for the macro model.
2. It is too expensive to precompute the needed information from the microscale model.

For the first point, as we remarked earlier in this section, the recent work on HMM and "equation-free" still requires some preconceived form of the macroscale model, although usually to a lesser degree than needed in most existing precomputing strategies. As for the second point, it is possible to develop rather efficient precomputing strategies. One strategy pursued in [21] is the use of sparse grids. With that it becomes quite feasible to precompute constitutive relations that depend on 5 or 6 variables. Currently, none of the examples in the HMM or "equation-free" papers have gone beyond that. In fact, in most concrete examples the constitutive relations depend on very few variables and can be easily precomputed. This does not mean that HMM and "equation-free" are useless, it is simply one benchmark that we should always keep in mind.

Acknowledgement: My work on multiscale modeling has been supported mainly by ONR and DOE, with some additional support from NSF. I am grateful to Bjorn Engquist, who introduced me to multiscale problems while I was a graduate student, and to Eric Vanden-Eijnden for the many intensive discussions. I am also grateful to the many collaborators of mine for sharing their views and their work. In addition, I would like to mention Roberto Car, Emily Carter, Shi Jin, Tim Kaxiras, Yannis

Kevrekidis, Sidney Yip and many others, who have brought different literature to my attention. In the case of Kevrekidis, it was the literature on CONNFFESSIT. I would like to mention in particular an NSF panel chaired by Bob Kohn on multiscale modeling in either 2000 or 2001, the discussions there greatly enhanced my knowledge about the literature.

References

[1] F. F. Abraham, J. Q. Broughton, N. Bernstein and E. Kaxiras, "Concurrent Coupling of Length Scales: Methodology and Application," *Phys. Rev. B*, vol. 60, no. 4, pp. 2391–2402, 1999.

[2] A. Brandt, "Multi-Level Adaptive Solutions to Boundary Value Problems," *Math. Comp.*, vol. 31, no. 138, pp. 333–390, 1977.

[3] A. Brandt, "Multiscale Scientific Computation: Review 2001," in *Multiscale and Multiresolution Methods: Theory and Applications*, Yosemite Educational Symposium Conf. Proc., 2000, Lecture Notes in Comp. Sci. and Engrg., T.J. Barth, et.al (eds.), vol. 20, pp. 3–96, Springer-Verlag, 2002. (An earlier version of this paper was: "Multiscale Scientific Computations: Review 2000).

[4] R. Car and M. Parrinello, "Unified Approach for Molecular Dynamics and Density-Functional Theory," *Phys. Rev. Lett.*, vol. 55, no. 22, pp. 2471–2474, 1985.

[5] S. Chen and G. D. Doolen, "Lattice Boltzmann Methods for Fluid Flows", *Ann. Rev. Fluid Mech.*, vol. 30, pp. 329-364, 1998.

[6] A. J. Chorin, "Numerical Solution of the Navier-Stokes Equations," *Math. Comp.*, vol. 22, pp. 745–762, 1968.

[7] A.J. Chorin, A.P. Kast and R. Kupferman, "Optimal Prediction of Underresolved Dynamics," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 8, pp. 4094-4098, 1998.

[8] S. M. Deshpande, "Kinetic Flux Splitting Schemes", *Computational Fluid Dynamics Review*, John Wiley & Sons, Chechester, 1995.

[9] W. E and B. Engquist, "The Heterogeneous Multi-Scale Methods," *Comm. Math. Sci.*, vol. 1, pp. 87–133, 2003.

[10] W. E and B. Engquist, "Multiscale Modeling and Computation," *Notices of the AMS*, vol. 50, no. 9, pp. 1062-1070, 2003.

[11] W. E, B. Engquist, X. Li, W. Ren and E. Vanden-Eijnden, "Heterogeneous multiscale methods: A review", *Comm. Comput. Phys.*, vol. 3, No. 3, pp. 367-450, 2007.

[12] W. E and J. Lu, “Seamless multiscale modeling via dynamics on fiber bundles”, *Comm. Math. Sci.*, vol. 5, No. 3, pp. 649-663, 2007.

[13] W. E and J. Lu, “The continuum limit and QM-continuum approximation of quantum mechanical models of solids”, *Comm. Math. Sci.* vol. 5, no. 3, pp. 679-696, 2007.

[14] W. E and P.B. Ming, “Analysis of the local quasicontinuum methods”, *Frontiers and Prospects of Contemporary Applied Mathematics*, Tatisen Li and Pingwen Zhang eds., Higher Education Press, World Scientific, Singapore, 2005, pp. 18-32.

[15] W. E and E. Vanden-Eijnden, “Some critical issues for the “equation-free” approach to multiscale modeling”, preprint, 2008.

[16] W. E, P.B. Ming and P.W. Zhang, “Analysis of the Heterogeneous Multiscale Method for Elliptic Homogenization Problems,” *J. Amer. Math. Soc.*, published electronically on September 16, 2004, in print: vol. 18, No. 1, pp. 121-156, 2005.

[17] W. E, W. Ren and E. Vanden-Eijnden, “A general seamless strategy for multiscale modeling”, preprint, 2008.

[18] B. Engquist and R. Tsai, “Heterogeneous multiscale method for a class of stiff ODEs”, *Math. Comp.*, vol. 74, 1707-1742 (2005).

[19] R. Erban, I. G. Kevrekidis, D. Adalsteinsson and T. C. Elston, “Gene regulatory networks: A coarse-grained, equation-free approach to multiscale computation”, *J. Chem. Phys.,*, vol. 124, 084106 (2006).

[20] K. Eriksson, C. Johnson and A. Logg, “Explicit Time-stepping for stiff ODEs”. *SIAM J. Sci. Comp.* vol. 25(4), 1142-1157 (2003).

[21] C. Garcia-Cervera, W. Ren, J. Lu and W. E, “Sequential multiscale modeling using sparse grids”, submitted to Comm. Comput. Phys.

[22] C.W. Gear and I.G. Kevrekidis, “Projective Methods for Stiff Differential Equations: Problems with Gaps in Their Eigenvalue Spectrum,” *SIAM J. Sci. Comput.*, vol. 24, no. 4, pp. 1091–1106, 2003.

[23] C. W. Gear, J. Li and I. G. Kevrekidis, “The Gap-Tooth Scheme for Particle Simulations”, *Phys. Lett. A*, vol. 316, pp.190-195, 2003.

[24] D. Givon, I. G. Kevrekidis, and R. Kupferman, “ Strong Convergence of Projective Integration Schemes for Singularly Perturbed Stochastic Differential Systems”. *Comm. Math. Sci.* **4**: 707–729 (2006).

[25] D. Givon and I. G. Kevrekidis, “Multiscale integration schemes for jump-diffusion systems”, preprint.

[26] L. Greengard, Lecture notes on fast multipole method, unpublished.

[27] I.G. Kevrekidis, C.W. Gear, J.M. Hyman, P.G. Kevrekidis, O. Runborg and C. Theodoropoulos, "Equation-Free, Coarse-Grained Multiscale Computation: Enabling Microscopic Simulators to Perform System-Level Analysis," *Comm. Math. Sci.*, vol. 1, no. 4, pp. 715–762, 2003.

[28] W. Hackbusch, "Convergence of Multigrid Iterations Applied to Difference Equations," *Math. Comp.*, 34(150), 425–440, 1980.

[29] J. Huang, private communications.

[30] G. Hummer and Y. Kevrekidis, "Coarse molecular dynamics of a peptide fragment: Free energy, kinetics, and long-time dynamics computations", *J. Chem. Phys.*, vol. 118, No. 23, pp. 10762-10773, 2003.

[31] J. Knap and M. Ortiz, "An Analysis of the Quasicontinuum Method," *J. Mech. Phys. Solids.*, vol. 49, no. 9, pp. 1899–1923, 2001.

[32] J. Li, P. G. Kevrekidis, C. W. Gear and I. G. Keverkidis, "Deciding the nature of the coarse equation through microscopic simulations: The baby-bathwater scheme", *Multiscale modeling and simulation*, **1** (3): 391-407 (2003).

[33] H. C. Ottinger, *Beyond Equilibrium Thermodynamics*, Wiley, 2005.

[34] R. G. Parr and W. Yang, *Density Functional Theory of Atoms and Molecules*, Oxford University Press, 1989.

[35] B. Perthame and F. Coron, "Numerical Passage from Kinetic to Fluid Equations," *SIAM J. Numer. Anal.*, vol. 28, pp. 26–42, 1991.

[36] D. I. Pullin, "Direct Simulation Methods for Compressible Inviscid Ideal Gas Flow", *J. Comput. Phys.*, vol. 34, pp. 231-244, 1980.

[37] W. Ren and W. E, "Heterogeneous Multiscale Method for the Modeling of Complex Fluids and Microfluidics", *J. Comput. Phys.*, vol. 204, No. 1, pp. 1-26, 2005.

[38] G. Samaey, I. G. Kevrekidis and D. Roose, Patch dynamics with buffers for homogenization problems, *J. Comput. Phys.*, **213**, 264-287 (2006).

[39] G. Samaey, D. Roose and I. G. Kevrekidis, "Patch dynamics finite difference methods for advection homogenization problems", preprint.

[40] R. H. Sanders and K. H. Prendergast, "On the Origin of the 3-Kiloparsec Arm", *Astrophysics Journal*, vol. 188, pp. 489-500, 1974.

[41] S. Succi, O. Filippova, G. Mith and E. Kaxiras, “Applying the Lattice Boltzmann Equation to Multiscale Fluid Problems”, *Computing in Science and Engineering*, vol. 3, 26, 2001.

[42] W. Sweldens, “The lifting scheme: A construction of second generation wavelets”, *SIAM J. Math. Anal.*, vol. 29, No. 2, pp. 511-546, 1997.

[43] E. B. Tadmor, M. Ortiz and R. Phillips, “Quasicontinuum Analysis of Defects in Crystals,” *Philos. Mag.*, A73 , 1529–1563, 1996.

[44] E. Vanden-Eijnden, “Numerical Techniques for Multiscale Dynamical Systems with Stochastic Effects,’ *Comm. Math. Sci.*, vol. 1, no. 2, pp. 385-391, 2003.

[45] E. Vanden-Eijnden, “On HMM-like integrators and projective integration methods for systems with multiple time scales.” *Comm. Math. Sci.*, **5**: 495-505 (2007).

[46] K. Xu and K.H. Prendergast, “Numerical Navier-Stokes Solutions from Gas Kinetic Theory,” *J. Comput. Phys.*, vol. 114, pp. 9–17, 1994.