

EQUATION-FREE, COARSE-GRAINED MULTISCALE COMPUTATION: ENABLING MICROSCOPIC SIMULATORS TO PERFORM SYSTEM-LEVEL ANALYSIS*

IOANNIS G. KEVREKIDIS[†], C. WILLIAM GEAR[‡], JAMES M. HYMAN[§], PANAGIOTIS
G. KEVREKIDIS[¶], OLOF RUNBORG^{||}, AND CONSTANTINOS THEODOROPOULOS^{**}

Abstract. We present and discuss a framework for computer-aided multiscale analysis, which enables models at a *fine* (microscopic/stochastic) level of description to perform modeling tasks at a *coarse* (macroscopic, systems) level. These macroscopic modeling tasks, yielding information over *long* time and *large* space scales, are accomplished through appropriately initialized calls to the microscopic simulator for only *short* times and *small* spatial domains. Traditional modeling approaches first involve the derivation of macroscopic evolution equations (balances closed through constitutive relations). An arsenal of analytical and numerical techniques for the efficient solution of such evolution equations (usually Partial Differential Equations, PDEs) is then brought to bear on the problem. Our equation-free (EF) approach, introduced in [1], when successful, can bypass the derivation of the macroscopic evolution equations *when these equations conceptually exist but are not available in closed form*. We discuss how the mathematics-assisted development of a computational superstructure may enable alternative descriptions of the problem physics (*e.g.* Lattice Boltzmann (LB), kinetic Monte Carlo (KMC) or Molecular Dynamics (MD) microscopic simulators, executed over relatively short time and space scales) to perform systems level tasks (integration over relatively large time and space scales, “coarse” bifurcation analysis, optimization, and control) directly. In effect, the procedure constitutes a system identification based, “closure-on-demand” computational toolkit, bridging microscopic/stochastic simulation with traditional continuum scientific computation and numerical analysis. We will briefly survey the application of these “numerical enabling technology” ideas through examples including the computation of coarsely self-similar solutions, and discuss various features, limitations and potential extensions of the approach.

1. Introduction

The purpose of this work is to analyze a computational framework linking conventional, continuum numerical methods/analysis with microscopic complex systems modeling, dynamics and control. Based on this framework, it is possible—under conditions discussed below—to perform the computer-assisted analysis of an evolution equation for the coarse, macroscopic level, closed description of a physical/material phenomenon *without having this closed description explicitly available*. This is accomplished by extracting from short bursts of simulation using a different, fine level, microscopic description (*e.g.* MD, KMC, kinetic theory based Lattice Gas (LG) or LB-BGK (Bhatnagar, Gross and Krook, [2] codes) the information that traditional numerical procedures would obtain through direct evaluation of the macroscopic evolution equation, had this equation been available. This type of information includes, for example, numerical approximations of residuals, (actions of) Jacobians and Hes-

*Received: August 15, 2002; accepted (in revised version): October 13, 2003. Communicated by Shi Jin.

[†]Department of Chemical Engineering and Program in Applied and Computational Mathematics; also Mathematics, Princeton University, Princeton, NJ 08544, USA. To whom correspondence should be addressed: (yannis@princeton.edu)

[‡]Department of Chemical Engineering, Princeton University, Princeton, NJ, USA; NEC Research Institute, Princeton, NJ, USA.

[§]CNLS and T-7, Los Alamos National Laboratory Los Alamos, NM 87545, USA.

[¶]Department of Mathematics, University of Massachusetts, Amherst, MA, USA.

^{||}Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, USA. Current address: Numerical Analysis and Computer Science, KTH, 100 44 Stockholm, Sweden.

^{**}Department of Process Integration, UMIST, Manchester M60 1QD, UK.

sians, partial derivatives with respect to parameters etc. Circumventing the derivation of the macroscopic-level description lies at the heart of the approach. We collectively refer to this class of methods as “Equation Free (Multiscale)” Methods (EF methods or EFM). The purpose of this computational framework is, therefore, to enable microscopic-level codes to perform system-level analysis directly, without the need to pass through an intermediate, macroscopic-level, explicit, “conventional” evolution equation description of the dynamics.

A persistent feature of complex systems is the emergence of macroscopic, coherent behavior from the interactions of microscopic agents, e.g. molecules, cells, individuals in a population, between themselves and with their environment. The implication is that macroscopic rules (description of behavior at a high level) can somehow be deduced from microscopic ones (description of behavior at a much finer level). For some problems, like Newtonian fluid mechanics, the successful macroscopic description (the Navier Stokes equations) predated its microscopic derivation from kinetic theory. In many important current problems, ranging from ecology to materials science, and from chemistry to engineering, the physics are known at the microscopic/individual level, but the closures required to translate them to a high-level macroscopic description are unknown. Severe limitations arise in trying either to find these closures directly, or to solve these problems at the scale at which the questions of interest are asked using microscopic simulations only. These limitations constitute a major stumbling block in current complex system modeling.

In this work, using as a tool the “*coarse time-stepper*” [1] we discuss the formulation, and in some instances demonstrate the implementation, of a set of “mathematics assisted” computational superstructure that can be wrapped around existing and future detailed atomistic/stochastic computer models. The procedure is intended to be used when a macroscopic description is *conceptually possible yet unavailable in closed form*. If *accurate* macroscopic models *are* available in closed form, one should probably work with them directly. It is interesting, however, even in this case, to explore the performance of the EF methods we describe here.

If what we did was to run a full (completely resolved in space and time) microscopic simulation, and merely observe the macroscopic behavior from it, the size of the resolved computation would obviously limit us to small systems over short times. A key feature of this work is that microscopic simulations are performed only in small *patches* in space-time. A *patch* is a small space domain (a “box” or a “tooth”) over a short time period. The solution is evolved in each patch, and then macroscopically interpolated across the patches (which can be thought of as nodes of a coarse mesh) to report the evolution of the macroscopic fields of interest; consecutive such reports can be used to estimate the time-derivative of the unavailable equation for the evolution of the coarse fields. These time derivative estimates can then be used as input to (a) a large time-step integrator—the projective integrator [3, 4]; (b) a zero-finder, to determine steady states, [1]; (c) an Arnoldi type eigensolver, to determine stability; and, more generally, to macroscopic task codes (controller design software, optimization software etc.). The detailed simulation should use the best current microscopic description of the physics available, and the code implementing it; our procedure will be wrapped around this microscopic simulator calling it as a subroutine. As microscopic simulators improve, the overall results will also improve. If the simulator does not embody correct physics, the EFM wrapper will extract, hopefully quickly and efficiently, the macroscopic consequences of the incorrect physics; this can help in realizing what is missing and modifying the microscopic model or code. The macroscopic computations

will take advantage of the arsenal of tools that have been developed for continuum evolutionary equation (usually Ordinary and Partial Differential Equation, ODE and PDE) descriptions of physical phenomena (e.g. [6, 7, 8, 9]). Standard tools include discretization and integration techniques, numerical linear algebra (most importantly, iterative large scale linear algebra, matrix-free solvers and eigensolvers), contraction mappings, continuation and numerical bifurcation analysis, as well as optimization (local and possibly global), and controller design computations (e.g. the solution of large scale Riccati equations and in general optimal control techniques).

A practical modeler is interested in the behavior of a system over time and space scales relevant to a particular application; ideally, a quantitatively predictive model is tailored to the scales of the application itself. Yet the relevant physical processes occur over a vast range of time and space scales, e.g. from the Angstroms and femtoseconds of molecular interactions to the millimeters and seconds relevant for typical macroscopic chemical reactors or laminar flows. Bridging this enormous range of dynamically coupled scales lies at the heart of physical modeling and continues to be a driving force in the development of mathematical tools and techniques, from homogenization to averaging, renormalization and the theory of inertial manifolds [10, 11, 12, 13]. These techniques underpin many successful explicit closures in equilibrium and nonequilibrium statistical mechanics (e.g. [14, 15, 16, 17, 18, 19, 20]). When explicit closures are not available, simulations must be done that seamlessly integrate different levels of physical modeling, as well as the different mathematical techniques appropriate for the different modeling levels/scales (e.g. [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]). The quasi-continuum methods of Phillips and coworkers [31, 32, 33] is such a pioneering multiscale/multilevel approach, in which the quantities required in solving for macroscopic fields are obtained by sampling atomistic computations over parts of the computational domain. Another such example of multiscale/multilevel techniques are those of Oettinger and coworkers, coupling continuum fields with stochastic evolution of molecular orientation in complex fluids [34, 35, 36, 37], see also [38]. The optimal prediction methods of Chorin and coworkers [39, 40, 41] provide a connection between microscopic/stochastic evolution and macroscopic observables. The review [42] on extracting macroscopic dynamics, from the projection operation formalism of Mori and Zwanzig [43, 44] to modern reduction methods for stochastic differential equation methods (e.g. [45, 46, 47]) contains a discussion of the strong relation between this approach and the coarse timestepper we will discuss below. Many other ideas associated with the exploitation of separation of scales become gradually integrated in modern multiscale computations; for example, a strong qualitative analogy exists between approximate inertial manifolds in deterministic PDEs and the quasi-equilibrium states of Gorban, Karlin and coworkers [48, 49, 50, 51]; use of these ideas in nonequilibrium thermodynamics can be traced back to Ehrenfest [52].

In this paper we borrow many of these mathematical and computational ideas, along with techniques that we have developed in our group, to put together a *systematic* computational framework for the equation-free analysis of general multiscale/complex system models. Optimal prediction over an appropriate time-horizon will, for example, be combined with the quasi-continuum idea of computation over small spatial domains and coarse projective integration to give rise to accelerated simulation through what we will call *patch dynamics*. Coarse timesteppers based on patches will be integrated with matrix-free solvers and eigensolvers to enable coarse stability and bifurcation analysis. Matrix-free eigensolvers, based on time-stepping results for different initializations and reporting horizons will be used to adaptively

explore the level at which macroscopic descriptions close. We hope that the *synthesis* proposed in our computational framework has new tools, useful points of view and insights to offer multiscale/complex system modelers. Ours is a system-identification based, “closure-on-demand” framework. It is interesting to draw an analogy with an established field of systems theory: the design of experiments for optimal parameter estimation/model building [53]. In our computational framework a microscopic simulation code is a *computational* experiment that one can initialize and run at will. The outer level code, the wrapper, designs these short bursts of computational experimentation, executes them, and processes their results, so as to design new computational experiments towards the ultimate goal. Different ultimate goals (long term integration, fixed point evaluation, stability, continuation, bifurcation, optimization and control tasks) require different design protocols. These protocols for designing computational experiments with an inner microscopic/stochastic simulator are templated on traditional continuum numerical analysis methods, and employ system identification tools [54, 55, 56].

The paper is organized as follows. In Section 2 we review our recent work on the coarse timestepper, and its use in coarse integration and coarse bifurcation analysis of unavailable macroscopic evolution equations through microscopic simulators. In Section 3 we discuss the “gap-tooth” scheme (first presented in [5]) and subsequently show how to combine it with coarse integration (for patch dynamics) and coarse bifurcation methods (for patch bifurcations). These steps enable the EF computation of macroscopic behavior over large spatial domains and long time intervals by post-processing the results of appropriately initialized and executed bursts of microscopic simulation over small spatial domains and short time intervals. Our main illustrative example will be the coarse LB-BGK simulation of a reaction-diffusion problem. We will then briefly survey an anthology of results from recent investigations outlining the scope of the EF methods in analyzing various types of problems, both in terms of the type of inner (microscopic) simulator, but also in terms of alternative tasks, like control-related tasks, homogenization—more precisely effective equation analysis—and the computation of coarsely self-similar solutions. In conclusion we will discuss various features and limitations of the EFM framework, as well as suggest areas in which we hope the methods may be fruitfully applied.

2. Background

We are interested in the coarse dynamics of large complex systems. In this section we assume that only solvers describing the *microscopic* behavior of the systems are available. Those could be based on, e.g., Monte-Carlo simulations, molecular dynamics, (Lattice-)Boltzman or PDEs with rapidly varying coefficients. Equations for the *macroscopic*, coarse, quantities of interest, such as the density, momentum, velocity and energy of a fluid (first moments) or the solution of the homogenized PDE, are, however, *not explicitly available*. We review here, for completeness, previous results so that they can be compared later with results obtained through more recent approaches. The focus is on general dynamic systems that are described without reference to specific spatial variation/structure.

Let $u(t)$ be the microscopic, detailed solution. To simplify the exposition, we will describe the case where u is given by a large system of ordinary differential equations,

$$\frac{du}{dt} = f(u), \quad u \in \mathbb{R}^m, \quad m \gg 1. \quad (2.1)$$

We denote this equation the *detailed* equation, and introduce the solution operator

(or *time stepper*, or time- t map) \mathcal{T}_d^t for which

$$u(s+t) = \mathcal{T}_d^t u(s),$$

if u solves (2.1). The main premise is that (2.1) is too expensive to solve for long times or that we do not have any precise knowledge of f ; typically (2.1) and \mathcal{T}_d would be given as a “black box,” legacy code, for which we can only control initial data and integration time.

The detailed equation (2.1) could, for instance, be a molecular dynamics simulation or a finely discretized PDE. The system could also represent an ensemble of identical, large systems that are run with different initial data. We should emphasize that there exist microscopic systems that cannot be written in the form (2.1), which can still be treated by the general methodology described here, e.g. stochastic systems simulated via Monte-Carlo or Brownian Dynamics methods.

The first thing we need to do in order to perform macroscopic level simulations based on (2.1), is to select the statistics of interest for describing the coarse behavior of the system and an appropriate representation for them. We will call this the macroscopic description, $U \in \mathbb{R}^M$, with $M \ll m$. For example, in a gas simulation at the particle level, the statistics would probably be the pressure, density, and velocity. For our LB-BGK reaction-diffusion and KMC lattice gas illustrative problems, the statistics of interest are the zeroth moments of the distribution (i.e. the concentrations). (We will use lower case-letters to denote microscopic variables and functions into the microscopic domain, and upper-case letters to name macroscopic variables and functions into the macroscopic domain. Usually the macroscopic, or coarse, variables have a lower dimension than the microscopic variables and our choice of terms will respect that. Since a primary purpose of the coarse variables is computational speedup, the lower dimensionality is important. However, we will sometimes indicate that coarse description variables may be functions (fields) —which have a higher dimension than any finite-dimensional particle model. In that case, the computational model will use a low-dimensional approximation of the function with finite differences, finite elements, or other standard computational tools.)

The choice of the coarse variables determines a *restriction* operator, which we denote \mathcal{M} . It takes a microscopic-level description u to the corresponding macroscopic description U ,

$$U = \mathcal{M}u.$$

This operator may involve averaging over microscopic space and/or microscopic time and/or number of realizations if an ensemble of simulations has been used. The maximum likelihood inference based field estimator of Li *et al.* [23, 24, 59], and the cumulative probability distributions in [60], are instances of such restriction operators.

If the detailed system can be successfully modeled at the coarse level through the macroscopic variable set, U , the remaining statistics should be well approximated as functionals of the selected ones; i.e. a fast-attracting slow manifold, parametrized by the selected macroscopic statistics, prevails when we observe the dynamics of the detailed microscopic problem. On this manifold, which embodies the closure of (2.1), all components of U are slowly varying. For example, suppose that there exists a change of variables $u \mapsto v$ in (2.1) to get the ODE $v' = g(v_1, v_2)$ where $v = (v_1, v_2)$ is a partition of v into M and $m - M$ components and that the compatible partition of

g is $(g_1, g_2/\varepsilon)$ with $\varepsilon \ll 1$. Then we have

$$\begin{aligned}\frac{dv_1}{dt} &= g_1(v_1, v_2), \\ \frac{dv_2}{dt} &= \frac{1}{\varepsilon} g_2(v_1, v_2).\end{aligned}\tag{2.2}$$

Suppose that v_2 closes on v_1 . (This is a much stronger condition than needed, and is used for illustration at this point.) Then we will have $v_2 = r(v_1)$ where $g_2(v_1, r(v_1)) = 0$. Identifying v_1 with the macroscopic variable U reduces (2.2) to the coarse equation

$$\frac{dU}{dt} = \frac{dv_1}{dt} = g_1(v_1, r(v_1)) = F(U).\tag{2.3}$$

Note that the functions g , r , and F are not available and neither (2.2) nor (2.3) can be used as a basis for direct numerical simulation.

Our problem can now be stated thus:

Problem A: We want to simulate (2.3) for large, or infinite, times, but we are only able to simulate (2.1) for short times.

2.1. Coarse Time Stepper. The main tool that allows the performance of numerical tasks at the macroscopic level using microscopic/stochastic simulation codes is the so-called coarse timestepper discussed in [1] (see also [57, 58, 4]). The coarse timestepper, denoted \mathcal{T}_c^τ , implements an approximation of the time- τ map for (2.3): the equation that governs the macroscopic evolution for the problem of interest, which is unavailable in closed form. The parameter $\tau > 0$ is the *time horizon* of the coarse time stepper. It is a length of time that is short enough to make the computation of $\mathcal{T}_d^\tau u$ tractable, yet large enough to be commensurate with the macroscopic dynamics (i.e. to allow enough change in the macroscopic dynamics so that we can locally estimate them).

The coarse time stepper relies on the detailed timestepper, \mathcal{T}_d for (2.1), the detailed microscopic/stochastic evolution of the system, as well as an appropriate, non-unique, *lifting* operator, μ , that maps the macroscopic description, U , to a consistent microscopic description, u . By consistent we mean that $\mathcal{M}\mu = I$, i.e. lifting from the macroscopic to the microscopic and then restricting down again should have no effect (except roundoff). For many types of microscopic descriptions, the lifting operator constructs distributions conditioned on (one or more of their) lower moments. For example, in a gas simulation using pressure etc. as the macroscopic-level variables, μ would probably be chosen to make random particle assignments consistent with the macroscopic statistics.

Given a macroscopic condition (e.g. concentration profile), $U(t)$, at some time t , the coarse timestepper consists of the following basic elements:

- (a) *Lift.* Transform the initial data through lifting to one (or more) fine, consistent microscopic realizations. When necessary to emphasize that the u obtained by lifting depends on the macroscopic variable at the lifting time, we will write this as $u(0, t) = \mu U(t)$, although when the meaning is clear, we will write $u(s + t)$ for $u(s, t)$.
- (b) *Evolve.* Use the microscopic simulator (the detailed timestepper) to evolve these realizations for the desired short macroscopic time τ , generating the value(s) $u(\tau, t) = \mathcal{T}_d^\tau u(0, t)$.
- (c) *Restrict.* Obtain the restriction of u and define the coarse timestepper solution as $\mathcal{T}_c^\tau U(t) := \mathcal{M}u(\tau, t)$.

In other words, the coarse time stepper is defined by

$$\mathcal{T}_c^\tau = \mathcal{M}\mathcal{T}_d^\tau\mu.$$

The choice of μ determines the initial data for v_2 at t in (2.2). For $s \gg \varepsilon$, and the assumption that v_2 approximately closes on v_1 , $g_2 \approx 0$, independent of initial data; the solution quickly approaches the slow manifold for any lifting operator μ . Except for an initial transient, the dynamics will well approximate the real dynamics of (2.3), and $U(t+s) \approx \mathcal{M}u(s,t)$. More importantly, the time derivatives of $\mathcal{M}u(s,t)$ will also be good approximations of the time derivatives of $U(t+s)$ after the initial transient, when the fast modes have been damped, e.g. $dU(t+s)/ds \approx d\mathcal{M}u(s,t)/ds$ when $s \gg \varepsilon$. The hope is that also in more general settings the coarse solution $(\mathcal{T}_c^\tau)^n U(0)$ at the discrete points in time $t = n\tau$ come from an evolution equation (a semigroup) like (2.3). In other words, we hope that a closed evolution equation—a formula for the time derivative—exists and closes for $U(t)$, defined for all t , and that its solution agrees well with our coarse timestepper at $t = n\tau$, i.e. $U(n\tau) \approx (\mathcal{T}_c^\tau)^n U(0)$.

Example. Let us consider a simple example. Suppose $u = (u_1, u_2)^T$ and that the microscopic equation (2.1) has the form

$$\begin{aligned} \frac{du_1}{dt} &= -u_1 - u_2 + 2, \\ \frac{du_2}{dt} &= \frac{1}{\varepsilon}(u_1^3 - u_2). \end{aligned} \tag{2.4}$$

We let the first component of u be the variable we observe the full dynamics on, our coarse variable, $U = \mathcal{M}u = u_1$, and we define the lifting as $\mu U = (U, 1)^T$. An example of a solution of this system is shown in Fig. 2.1, and the corresponding coarse time stepper solution is given in Fig. 2.2. The solution of (2.4) rapidly moves to the slow manifold $u_2 = u_1^3$ for any initial data when $\varepsilon \ll 1$. The coarse time stepper solution agrees better with the full solution (or its restriction) when ε is small or τ is large. More sophisticated lifting procedures (e.g. short runs constrained on the values of the coarse variables, in the spirit of algorithms like SHAKE [61], or of consistent initialization of differential-algebraic equations [62]) can help alleviate the lifting error. In this paper we will assume fast off-manifold dynamics (ε is small) lead to quick “healing” of the lifting error directly.

Through the coarse time stepper, our Problem A above has now been reduced to

*Problem B: We want to simulate (2.3) for large, or infinite, times,
but we are only able to simulate it for short discrete times, $t = n\tau$.*

We note that the coarse time stepper applied to (2.1) can now be viewed as a black box solver for (2.3). This is the legacy code view of our approach: given a good black-box timestepper for the problem, how to best use it (short of rewriting it) to explore the coarse behavior of (2.3)?

2.2. Coarse Projective Integration. In this section we discuss the first part of Problem B above: How to integrate (2.3) for large (macroscopic) times $t \gg \tau$, where τ is the time horizon in the coarse time-stepper \mathcal{T}_c^τ . Coarse projective integration thus enables the “short time” coarse timestepper to perform “long time” tasks.

Let $\Delta T \gg \tau$ be a large time step (commensurate with the slow coarse dynamics) and $\delta t \leq \tau$ be a small time step (commensurate with the relaxation of the fast dynamics). Set $t_k = k\delta t$. The small δt could for instance be the numerical time step used when solving the detailed equation (2.1). Introduce the numerical approximations of

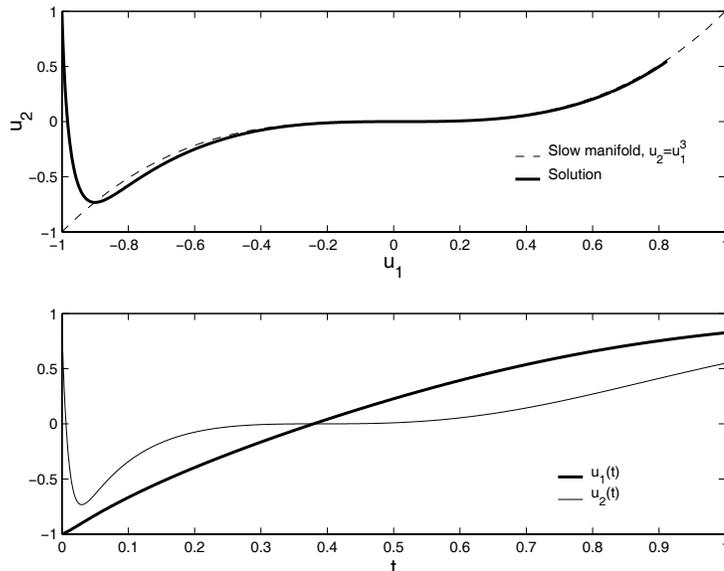


FIG. 2.1. Solution of (2.4) with $\varepsilon = 0.01$, $u_1(0) = -1$ and $u_2(0) = 1$. The top figure shows the solution plotted in the (u_1, u_2) phase space. The bottom figure shows the solution as a function of time.

the coarse solution $U(t)$ as $U^N \approx U(N\Delta T)$ and let $u^{0,N} = \mu U^N$ be the lifted fine approximation. The microscopic time stepper permits

$$u^{k,N} = \mathcal{T}_d^{t_k} u^{0,N} \quad (2.5)$$

to be computed. As long as the number of small time steps, k , is such that $t_k = k\delta t = O(\tau)$, we can afford this computation. We define the coarse approximations $U^{k,N}$ at small time steps as

$$U^{k,N} := \mathcal{M}u^{k,N} \approx U(N\Delta T + t_k). \quad (2.6)$$

By consistency $U^{0,N} = U^N$. We cannot compute an approximation of U^{N+1} in this way, however, since $t_k \ll \Delta T$. Instead we extrapolate U^{N+1} by using a coarse scheme of the type

$$U^{N+1} = U^{k,N} + (\Delta T - t_k)\tilde{F}(U^{k,N}), \quad (2.7)$$

where \tilde{F} approximates F . For $t_k \geq \tau$ the fast modes in the detailed equations have died out so that differences of $U^{k,N}$ are useful for approximating derivatives of the exact coarse solution. We use differencing, setting

$$\tilde{F}(U^{k,N}) := \frac{U^{k+1,N} - U^{k,N}}{\delta t} \approx \frac{dU(N\Delta T + t_k)}{dt} = F(U(N\Delta T + t_k)) \approx F(U^{k,N}). \quad (2.8)$$

The method in (2.7) and (2.8) is called Projective Forward Euler, and it is the most simple instance of a class of methods known as projective integration methods. A detailed study of this type of algorithms can be found in [3, 87]; while implicit versions of these algorithms are possible, and are partially discussed there, the heart of

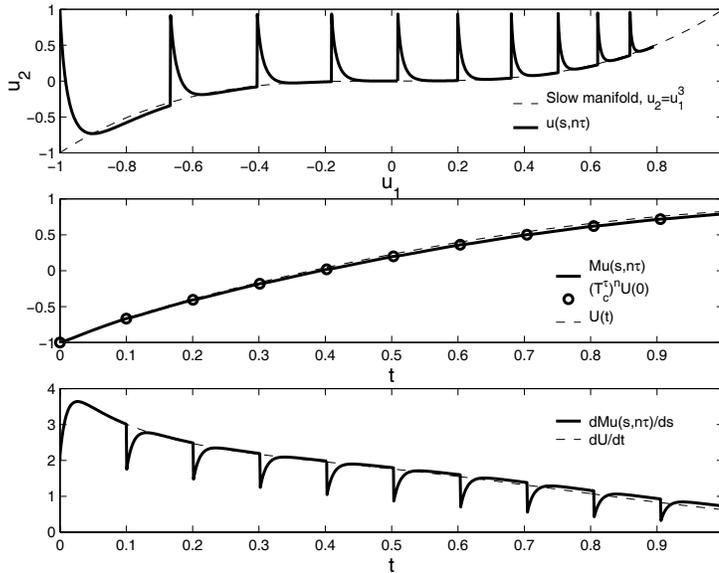


FIG. 2.2. Results with the coarse time stepper applied to (2.4) using $\tau = 10\varepsilon = 0.1$ and $U(0) = -1$. The top figure shows the lifted solution $u(s, n\tau)$ plotted in the (u_1, u_2) phase space. At times $t = n\tau$, the solution is restricted and lifted, which here amounts to setting $u_2(n\tau) = 1$. The middle figure shows the restricted solution $Mu(s, n\tau)$ as a function of time $t = n\tau + s$. Circles correspond to the coarse time stepper approximation, $(\mathcal{T}_c^5)^n U(0) = Mu(0, n\tau)$. These results are compared with the exact coarse solution $U(t) = u_1(t)$ from Fig. 2.1. The bottom figure shows the same comparison for the derivative of the functions.

the process is an *explicit* outer integrator, here (2.7), built around calls to an inner integrator, here (2.5), a version of the coarse time stepper.

Note that the fast modes in u may not be completely removed by applying the restriction operator \mathcal{M} ; fast modes are still present in the coarse solution U , although they are small in size. For stability reasons those fast modes need to be sufficiently damped by the coarse integrator (2.6) to neutralize their growth in the large, projective, extrapolation step in (2.7). In fact, each small time step of the coarse integrator multiplicatively reduces the fast components, so the error reduction scales with a power of the number of steps, k , whereas the growth in the extrapolation is linear in the size of the large step, $\Delta T - t_k$. As such, the algorithms bear many resemblances to explicit codes for stiff systems [88, 89, 90]; yet ours have been devised with multiscale problems and a microscopic/stochastic inner integrator in mind. The main enabling step is the coarse time-stepper through the “lift–evolve–restrict” evaluation sequence. There are some significant differences from explicit RK codes, including the applicability in the legacy code or noisy inner integrator cases, and ease in dynamically selecting step sizes (see [87] for details). It would also be interesting to compare our projective integrators with Krylov/exponential integrators for large systems with separations of time scales [91, 92, 93, 94].

Higher order versions of (2.7) can be constructed based on polynomial extrapolation [4]. Let $P(t)$ be the q -th order polynomial interpolating the points

$$P(N\Delta T + t_{k+s}) = U^{k+s, N}, \quad s = 0, \dots, q,$$

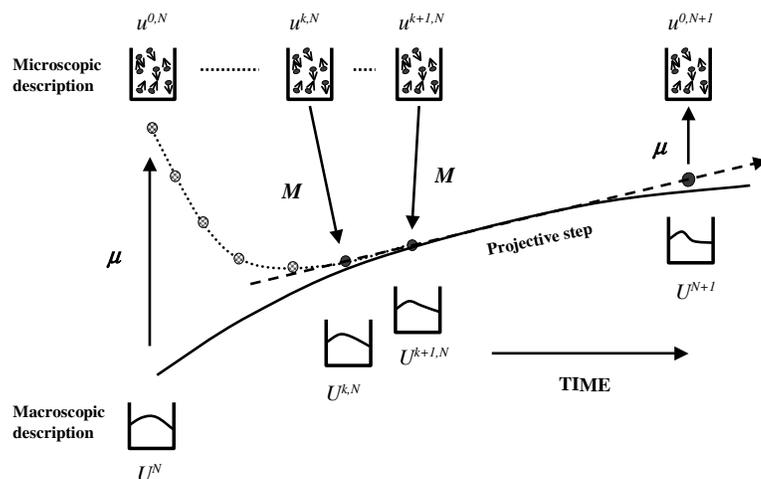


FIG. 2.3. Schematic representation of an explicit projective integrator. Notice the lifting, the evolution with successive restrictions, the estimation of the coarse time derivative, and the projection step followed by a new lifting.

and subsequently use it to extrapolate the coarse solution at the next time level,

$$U^{N+1} = P((N+1)\Delta T).$$

The Projective Forward Euler method is the case when $q = 1$. Alternatively, a multistep version of (2.7) can be used [95].

The general procedure is illustrated in Fig. 2.3: an initial condition is taken in coarse space (e.g. a density field) and lifted to a consistent distribution in microscopic space (e.g. cells for a chemotaxis problem). The microscopic code is used to evolve the distribution *long enough* for the higher moments, that have been incorrectly initialized, to heal. A few more evolution steps are then taken, and the solutions restricted to coarse space (densities). Successive density profiles are used to estimate the time derivatives of the (unavailable) density equation (again, an identification step). The profile of the (coarse) density function can be represented by the basis functions used in any of the traditional numerical discretization techniques, including finite differences, finite volumes, finite elements and spectral methods. The time-derivatives for the coefficients of the basis function will be estimated, and used in the projective integration step; finite difference (nodal), finite element as well as empirical basis function representations have been illustrated in [4].

We have analyzed and used such two-tier (and multi-tier) EFM projective integrators for what we term “coarse integration” of the macroscopic equations using microscopic as well as stochastic detailed time-steppers. Fig. 2.4, taken from [4] shows time-series as well as long-term attractors of such an “inner LB, outer FEM” one stage

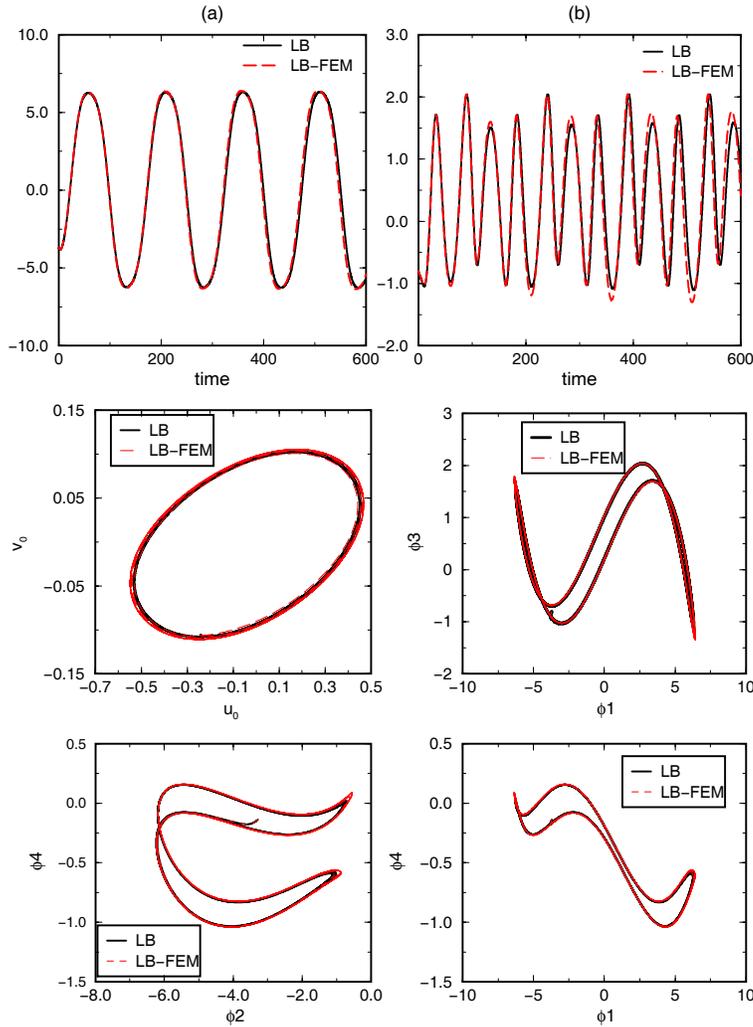


FIG. 2.4. A comparison of FHN dynamics computed through (1) an LB-BGK code and (2) a 100-50-350 inner-LB - outer FEM (explicit Euler) projective integrator (from [4]). The comparisons are performed by projection on subspaces spanned by a few empirical orthogonal eigenfunctions (EOFs or PODs, see text). Top two panels: time series for (1, solid) and (2, dashed) starting from the same initial conditions. Bottom four panels: various phase space projections (in POD-mode space) of the attractors of the LB-BGK (1, solid) and the 100-50-350 LB-FEM projective scheme (2, dashed).

projective forward Euler integrator. Because of the nature of the outer description, we call these two-tier codes *micro-Galerkin methods*. We have also explored using empirical basis functions (EOFs) for the representation of the macroscopic field. These are also known as KL modes (from the Karhunen-Loeve expansion), POD modes (from the Proper Orthogonal Decomposition) or simply PCs (from Principal Component Analysis) see [96, 97]. Such basis functions (modes) are useful for the parsimonious representation of macroscopic fields in complicated geometries, and are extensively

used in nonlinear identification and model reduction (for data compression, identification, bifurcation, control and optimization tasks), e.g. [98, 99, 100, 101, 102].

2.3. Coarse Bifurcation Analysis. In this section we discuss the second part of Problem B above: How to find the solution of (2.3) for infinite times. More specifically, we assume that the coarse equation (2.3) also includes a real parameter λ ,

$$\frac{dU}{dt} = F(U; \lambda), \quad U \in \mathbb{R}^M, \quad (2.9)$$

and we are interested in the qualitative behavior of its solutions for large times, and their dependence on λ : the stability of steady states of (2.9) and bifurcation diagrams with respect to λ . Note that a macroscopic, coarse, steady state can exist even though there are no true microscopic steady states (e.g. in the case of stochastic microscopic simulations). We recall that $F(U; \lambda)$ is not known explicitly. We only have a timestepper \mathcal{T}_c^τ for the equation and we can only afford to evolve U for short times, $\tau \ll 1$. The coarse bifurcation analysis with RPM that we discuss in this section will enable this short-time coarse timestepper to perform infinite-time tasks.

Let us assume that (2.9) has a steady state, which we denote by $U^*(\lambda)$. At the steady state, $dU/dt = 0$. Then U^* is obviously also a fixed point of \mathcal{T}_c^τ ,

$$U^* = \mathcal{T}_c^\tau U^*, \quad (2.10)$$

and the *stability* of U^* is given by the eigenvalues of the Jacobian of \mathcal{T}_c^τ ,

$$\rho \left(\frac{d\mathcal{T}_c^\tau(U^*)}{dU} \right) < 1, \quad \Rightarrow \quad \text{stability.}$$

The question that we are faced with is hence how to solve the fixed-point problem (2.10). Viewing the problem like this is the basis for time stepper based bifurcation analysis from which our original inspiration comes [69, 70, 71, 72, 73].

One naive way to solve (2.10) is to use fixed point iteration (or successive substitution), starting with some initial data U^0 and for $n \geq 0$ setting

$$U^{n+1} = \mathcal{T}_c^\tau U^n. \quad (2.11)$$

This corresponds to *direct simulation* of (2.9), using the coarse time stepper to evolve an initial profile in time. Since we can only realize \mathcal{T}_c^τ for small τ by assumption, this is, however, not a feasible approach. Indeed, the convergence rate for fixed point iterations is only linear. Moreover, we will not be able to find unstable solutions to (2.10) in this way.

Alternatively, we could use a Newton type iteration to solve (2.10), e.g.

$$U^{n+1} = U^n - \left[\frac{\partial \mathcal{T}_c^\tau(U^n)}{\partial U} - I \right]^{-1} [\mathcal{T}_c^\tau U^n - U^n].$$

This gives fast (quadratic) convergence, also when U^* is an unstable solution. However, the method needs the Jacobian of \mathcal{T}_c^τ , which is not available in closed form in our case, and it would be expensive to approximate it numerically.

A compromise between fixed point iteration and Newton type methods (e.g. see the monograph [74] for Newton-Krylov methods) is Shroff and Keller's Recursive Projection Method (RPM) [69]. It gives fast convergence (between linear and quadratic)

and converges also when U^* is slightly unstable. The Jacobian of \mathcal{T}_c^τ never needs to be evaluated directly. Moreover, as a by-product, RPM produces approximations of the largest eigenvalues of the Jacobian, which give stability information for the steady state.

Let $\mathbb{P} \subset \mathbb{R}^d$ be the maximal invariant subspace corresponding to the largest $d \ll M$ eigenvalues of $\partial\mathcal{T}_c^\tau(U)/\partial U$, and let \mathbb{Q} be its orthogonal complement in \mathbb{R}^M . The space \mathbb{P} represents the direction along which time evolution is slowest, possibly slightly unstable. Under the loose assumptions described below, the results of repeated calls to the time stepper can be used to adaptively approximate it. Decompose the solution U as

$$U = PU + QU =: p + q, \quad p \in \mathbb{P}, \quad q \in \mathbb{Q},$$

where P and Q are the projections onto \mathbb{P} and \mathbb{Q} , respectively. The philosophy of RPM is to do fixed point iterations in \mathbb{Q} and Newton type iterations in \mathbb{P} ,

$$\begin{aligned} p^{n+1} &= p^n - [J^*(p^n + q^n) - I]^{-1} [PT_c^\tau(p^n + q^n) - p^n], \\ q^{n+1} &= QT_c^\tau(p^n + q^n), \end{aligned}$$

where $J^* = P(\partial\mathcal{T}_c^\tau(U)/\partial U)P \in \mathbb{R}^{d \times d}$ is a small size Jacobian. The invariant subspace \mathbb{P} , J^* and d are dynamically updated by monitoring e.g. the convergence rate of $\{q^n\}$.

The combination of (approximate) Newton iteration in the low-dimensional, slow subspace, and fixed-point iteration (integration) to provide a contraction in its complement, and its extensions, are being successfully used to perform continuation, stability and bifurcation analysis of dissipative PDEs and low-index PDAEs close to low-codimension bifurcations [69, 78, 79, 80, 81, 82, 83, 84, 85].

Standard time stepper based bifurcation analysis using RPM wrapped around a PDE time stepper, is represented by the (b) branch in Fig. 2.5. Substituting the coarse time stepper, described by branch (a), for the PDE time stepper turns RPM into a coarse bifurcation code. If the the coarse time stepper is accurate enough a computational superstructure like RPM can indeed be wrapped around it in this way and enable it to perform the time stepper based bifurcation analysis of the (unavailable in closed form) coarse description of the problem. This has been demonstrated in e.g. [1, 4, 57, 58, 63, 64, 65]. Through the lift–evolve–restrict procedure, we enable a code doing time evolution at a fine level of description, to perform bifurcation analysis at a completely different, coarse level of description.

The main assumption for RPM to perform efficiently is that the system has a clear separation of time scales: a few eigenvalues of the Jacobian of F in (2.9) lie in a narrow strip around the imaginary axis (possibly some unstable ones too); then comes a *spectral gap*, and the rest of the eigenvalues are far to the left in the complex plane. For the Jacobian of the coarse time stepper \mathcal{T}_c^τ , this translates to having a few eigenvalues in a strip around the unit circle, then a gap, and then many eigenvalues in a small disk around zero for the time-stepper.

While so much structure seems like a lot to expect *a priori* from a model, (or from the physical process that the model springs from) the usually dissipative PDEs modeling reaction and transport (and including diffusion, viscous dissipation, heat conduction etc.) often do possess such a separation of time scales. Similar assumptions, for example, underpin the theory of Inertial Manifolds and Approximate Inertial Manifolds [75, 13, 76], and many singularly perturbed systems that arise in engineering modeling. In addition, as we will qualitatively discuss below, even if the problem

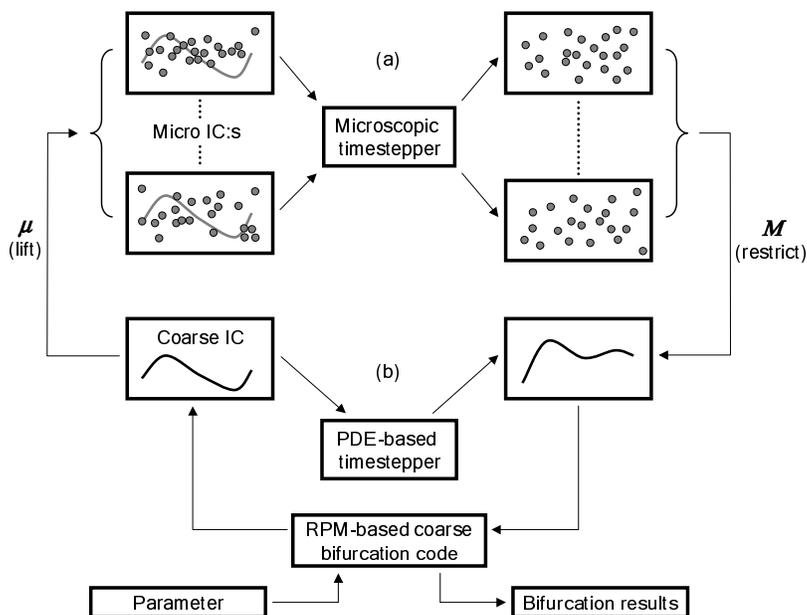


FIG. 2.5. Schematic description of time-stepper based bifurcation analysis, fine and coarse. Notice the lifting of a macroscopic initial condition to an ensemble of consistent microscopic ones, as well as the restriction of the microscopic integration results back to the macroscopic (usually moments-based) description.

itself does not have a separation of time scales, it is possible that the Fokker-Planck equation for the evolution of the statistics of the problem may have a separation of time scales. The EF approach will then be used on a time stepper for that level of (coarse, averaged) description of the statistics of the problem [77].

3. Coarse equation-free computations in space and time

The discussion up to this point combined several elements of published work, to form an appropriate background for the new results that follow. So far we have been concerned with systems without any reference to spatial structure; we have enabled short-time coarse time steppers (through the lift–evolve–restrict approach, identification, and connection with traditional scientific computation) to perform infinite-time tasks (find steady states and bifurcations) as well as long-time tasks (coarse integration).

We will now explicitly consider also the spatial structure of the system. In this context we discuss a different enabling technology that we will refer to as the gap-tooth scheme. It enables a “small space, short time” time stepper to perform “large space, short time” tasks [5]. The gap-tooth scheme is in fact an instance of the coarse time stepper in the setting where the system has a spatial structure that we can capitalize on.

When later we will combine the gap-tooth scheme with coarse projective integration we will obtain *patch dynamics*, which enables the “small space, short time”

gap-tooth time stepper to perform “large space, large time” tasks. Similarly, the combination of the gap-tooth scheme and methods like RPM enables the performance of “large space, infinite time” tasks.

The microscopic equation will again be exemplified by a large system of ODEs,

$$\frac{du}{dt} = f(u), \quad u \in \mathbb{R}^m, \quad m \gg 1, \quad + \text{ boundary conditions.} \quad (3.1)$$

However, the unknowns are now connected to spatial points and there is also an associated set of boundary conditions. The unknowns u in (3.1) can themselves include spatial locations, as in a molecular dynamics simulation, or they can simply correspond to fixed spatial points, as in a discretization of a fine scale PDE. When the spatial domain in which (3.1) is solved increases in size, the dimension n of the problem also increases. We therefore assume that (3.1) can only be solved in small domains, for short times. As in Section 2, the ODE-based description is a simplification; the procedure is applicable for microscopic evolution laws that do not fit in this example framework, notably Monte-Carlo simulations of stochastic systems.

When we refer to “space” we do not necessarily mean *physical space* but the domain over which the solution is defined at any time instant. For example, the solution could be in a frequency domain or it could be a probability distribution.

As before, we let u denote the coarse quantities that we are interested in, e.g. concentrations or moments of the particle distribution functions. This defines the restriction operator \mathcal{M} . The coarse solution U now also depends on the spatial variable and \mathcal{M} takes $u(t)$ to $U(t, x)$, a continuous function in x . We assume that U satisfies some coarse evolutionary equation, possibly a PDE which, in one dimension takes the form

$$U_t = L(t, x, U, \partial_x U, \dots, \partial_x^p U). \quad (3.2)$$

for some L and integer p ; integrodifferential evolutionary equations also fit this framework. Even though this equation conceptually exists, it is not explicitly available in closed form. At this point we are thus faced with

Problem C: We want to simulate (3.2) for large or infinite times over large spatial domains, but we can only solve (3.1) in small spatial domains and for short times.

Our strategy will follow that in Section 2: construct accurate approximate time steppers for a semi-discretization of (3.2) *without* explicitly deriving it through analytical mean-field or averaging procedures.

3.1. Gap-tooth Scheme. In this section we introduce the gap-tooth scheme. We use the microscopic rules (3.1) 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. We will present in some detail an application of the gap-tooth scheme to the diffusion equation, to motivate the scheme. We subsequently discuss how it, in more general cases, can lead to hybrid, two-tier (e.g. finite difference–molecular dynamics or finite difference–Monte Carlo) timesteppers for (3.2).

Consider a discretization of (3.2) on the full one-dimensional domain. Let the computational grid $\{x_j\}$ have a spatial stepsize H , so that $x_j = jH$, which is capable of accurately approximating the true coarse solution. This is our coarse mesh and we think of it as a set of adjacent boxes of size H , centered at the grid points. Note that it is fine enough for the numerical approximation to the solution to be accurate

for macroscopic purposes. We also let τ be the macroscopic time step, which is assumed small enough for a coarse explicit finite difference scheme to be stable, i.e. one satisfying the CFL condition.

In a finite difference scheme, the exact coarse solution would be approximated by a grid function $\{U_j^N\}$ at the coarse nodal positions, $U_j^N \approx U(N\tau, jH)$. What happens with the solution in each H -sized box during a single time step τ is computed (approximated) using fluxes, proportional to derivatives evaluated through finite differences. For reaction-diffusion problems, reaction source-sink term would also be evaluated at the nodal point(s).

Centered around each discretization point of the coarse mesh, within each H -sized box, we now consider a *small box* of size $h \ll H$: the interval $[x_j - h/2, x_j + h/2]$. These small boxes are the “teeth” of the scheme. They are separated by the “gaps”, each of size $H - h$.

We want to represent what happens in the big box of size H through what happens in the small box of size h , appropriately weighting the various terms by box size. This is *another* way of approximating the exact coarse solution $U(t, x)$. The small box in this approximation plays the role of the nodal point of the coarse finite difference scheme.

As in the coarse time stepper case, we base the gap-tooth step on the evolution of the microscopic equation in the tooth together with a lifting operator. In addition, the gap-tooth scheme must also construct *boundary conditions* for the microscopic equation (3.1) based on the coarse solution. Otherwise we follow the same steps as in Section 2.1. Given a finite-dimensional representation $\{U_j^N\}$ 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.

- (a) *Boundary conditions.* Construct boundary conditions for each small box based on the coarse representation $\{U_j^N\}$.
- (b) *Lift.* Use lifting to map the coarse representation $\{U_j^N\}$ to initial data for each small box.
- (c) *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 (a) and (b).
- (d) *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 $t = \tau$.

These steps thus reduce Problem C to

Problem D: We want to simulate (3.2) for large or infinite times over large spatial domains, but we can only simulate it for short discrete times, $t = N\tau$, over large domains.

We recognize that this problem is actually the same as Problem B, and we should be able to treat it with the same tools as in Section 2. This leads us to the patch dynamics, that we will discuss below in Section 3.2.

We should comment on the choice of boundary conditions and lifting in steps (a) and (b) above. If (3.2) is a balance equation (diffusion, reaction diffusion) the local boundary conditions should be the “correct” slopes (determining the fluxes) in and out of each *small box*. The derivative $s_{j+\frac{1}{2}}$ at the midpoint $x_{j+\frac{1}{2}} = \frac{1}{2}(x_j + x_{j+1})$ between two adjacent coarse nodes x_j and x_{j+1} can be approximated by the corresponding centered finite differences

$$s_{j+\frac{1}{2}} = (U_{j+1}^N - U_j^N)/H, \quad (3.3)$$

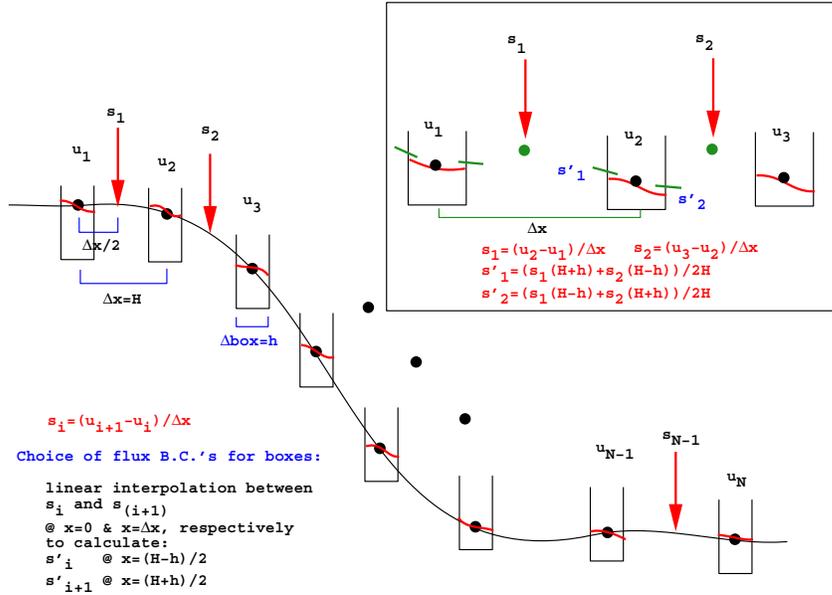


FIG. 3.1. An original schematic description of the gap-tooth scheme (from the 2000 AIChE Area 10 plenary lecture, [5]); the notation is slightly different from the one in the paper. Notice the macro-mesh and the smoothly interpolated macro solution, the teeth surrounding the macro mesh, and the interpolated boundary conditions at the outer edges of the teeth or “patches”. For this continuum description the buffer zones are not depicted.

if $\{U_j^N\}$ represents nodal values. Let the left and the right slope boundary conditions for each small box be s_j^- and s_j^+ . They can then be computed by linear interpolation between $s_{j-\frac{1}{2}}$ and $s_{j+\frac{1}{2}}$ (see inset of Fig. 3.1). Therefore

$$s_j^- = \frac{(H + h)s_{j-\frac{1}{2}} + (H - h)s_{j+\frac{1}{2}}}{2H} \tag{3.4}$$

$$s_j^+ = \frac{(H - h)s_{j-\frac{1}{2}} + (H + h)s_{j+\frac{1}{2}}}{2H}. \tag{3.5}$$

See also the discussion in Section 3.3.

Initial conditions for the simulation in the small boxes with these boundary conditions are given by lifting in step (b). It is reasonable to expect that the average over the small box of the quantity we are studying will be initially equal to the nodal value U_j^n . Any initial condition with the correct average ought to do a good job in the scheme we envision. As we discuss below, it is also possible that we may have to run an *ensemble* of such macroscopically consistent initial conditions and average over their results. If the initial condition does not satisfy the boundary conditions at $t = 0$ there will be sharp startup transients close to the boundary, but in principle the time-reporting horizon for the simulation in the small box (corresponding to one time step of a finite difference scheme for the big box) should be enough for these transients to smoothen out. We will return extensively to the discussion of the time-reporting horizon of the simulation in each small tooth.

3.1.1. An Illustration: the Diffusion Equation. We now consider a one-dimensional example where the coarse equation (3.2) exists and we know it in closed form. More precisely, let us study the linear diffusion equation,

$$U_t = U_{xx}. \quad (3.6)$$

The corresponding microscopic equation (3.1) could be a Brownian motion model, in which case U would be the zeroth moment of the single particle distribution function. To further simplify the analysis, we will, however, temporarily assume that the detailed description (3.1) is *also* the diffusion equation (3.6). In each small box we thus solve the same transient governing balance equation as for the whole domain.

We assume that the grid function $\{U_j^N\}$ represents nodal values of the coarse solution at discrete times, $U_j^N \approx U(N\tau, jH)$. Furthermore, we let $v_j(t, y)$ denote the continuous, detailed, solution in the small box j with appropriate boundary and initial conditions where the spatial coordinate y is local to each box $0 \leq y \leq h$. We select the coarse variable to represent an average over this detailed solution. The restriction operator is an integral that averages $v_j(t, y)$ over each small box and identifies it with U_j^N .

Given a small box size h , for each box j , we perform the following steps.

- (a) *Boundary conditions.* Following the remark at the end of the previous section, we use Neumann boundary condition at the small box edges. We thus require that

$$\partial_y v_j(t, 0) = s_j^-, \quad \partial_y v_j(t, h) = s_j^+, \quad t > 0. \quad (3.7)$$

The prescribed slopes, s_j^- and s_j^+ , are computed from $\{U_j^N\}$ using (3.3), (3.4) and (3.5).

- (b) *Lift.* We want to lift the coarse representation to an initial profile that is smooth and consistent with the boundary data and whose average agrees with U_j^N . It is reasonable (from all possible such initial conditions) to choose the simplest polynomial. In this case it is a quadratic polynomial that we denote $Q(y)$,

$$v_j(0, y) = Q(y) := Ay^2 + By + C \quad (3.8)$$

where A , B and C are constants. For consistency, it should satisfy

$$Q'(0) = s_j^-, \quad Q'(h) = s_j^+, \quad \frac{1}{h} \int_0^h Q(y) dy = U_j^N. \quad (3.9)$$

We get

$$A = \frac{s_j^+ - s_j^-}{2h}, \quad B = s_j^-, \quad C = U_j^N - \frac{1}{h} \int_0^h Ay^2 + By \, dy.$$

Note that A , B and C all depend linearly on elements of $\{U_j^N\}$ via (3.3), (3.4) and (3.5).

- (c) *Evolve.* Integrate the equation in the small box for time τ . During this time the boundary conditions remain constant and equal with their values at the beginning of the step. Hence, solve

$$\partial_t v_j = \partial_{yy} v_j, \quad y \in (0, h), \quad t \in (0, \tau],$$

with initial data (3.8) subject to boundary conditions (3.7). In general the solution would need to be approximated numerically, e.g. through a *fine* FD scheme, *within* each box, and with well-chosen fine time-step, or through a reasonable truncation of the solution obtained through separation of variables. Here, however, we are lucky: for the diffusion equation with the particular initial data in (3.8), it has an *explicit* solution:

$$v_j(t, y) = Q(y) + Q''(y)t = Q(y) + t \frac{s_j^+ - s_j^-}{h}. \quad (3.10)$$

- (d) *Restrict.* Compute U_j^{N+1} , the discrete approximation of $U(t, x_j)$ at time $(N + 1)\tau$, by averaging in space over the small box solution $v_j(t, y)$ at the final time τ . From (3.10) we obtain

$$U_j^{N+1} = \frac{1}{h} \int_0^h v_j(\tau, y) dy = \frac{1}{h} \int_0^h Q(y) dy + \tau \frac{s_j^+ - s_j^-}{h}. \quad (3.11)$$

By also using (3.9), (3.4) and (3.5), this gives

$$U_j^{N+1} = U_j^N + \tau \frac{s_j^+ - s_j^-}{h} = U_j^N + \tau \frac{s_{j+\frac{1}{2}} - s_{j-\frac{1}{2}}}{H}.$$

Finally, from (3.5) we get the new coarse mesh quantities,

$$U_j^{N+1} = U_j^N + \tau \frac{U_{j+1}^N - 2U_j^N + U_{j-1}^N}{H^2}. \quad (3.12)$$

The procedure then repeats itself. We interpolate to obtain a new coarse field, i.e. use the U_j^{N+1} to compute the new slopes s_j^+ and s_j^- and so on.

The final scheme (3.12) is *exactly* the same as a standard finite difference method for the diffusion equation. The method can be seen as a method of lines discretization of (3.6) using forward Euler in time and central differencing in space. Hence, in this simple case the hybrid scheme reduces exactly to a corresponding coarse FD discretization.

In the example we used a nodal representation of the macroscopic field as well as linear interpolation of coarse slopes to determine the patch boundary conditions. These features are shared by the FD scheme that was used to motivate the approach. We refer to this as the “*nodal-Exact*” gap-tooth scheme. The first component of the name, “nodal,” refers to our chosen representation of the macroscopic field which guides the number and location of coarse nodes as well as the patch boundary condition procedure. The second component, “exact,” is the way by which we solve the equation in each small box.

In the simple diffusion example, the truncation error for nodal-Exact is $O(H^2 + \tau)$, independent of h . To analyze the accuracy of the scheme, consider first the limiting case when $h = H$ i.e., when there are no gaps between the teeth. The scheme will then have truncation error $O(H^{p_0} + \tau^{p_1})$. The values of p_0 and p_1 depend on the order with which the boundary conditions are interpolated in space and extrapolated in time from the data U_j^N . Furthermore, an asymptotic analysis reveals that the difference between nodal-Exact with $h < H$ and with $h = H$ in one timestep is $O(\tau(H^q - h^q))$, for some q (for $h, H \ll 1$). Thus the total truncation error for nodal-Exact is bounded by

$$O(H^{p_0} + \tau^{p_1} + |H^q - h^q|). \quad (3.13)$$

When the nodal–Exact scheme described above is applied to a reaction-diffusion equation, $U_t = U_{xx} + F(U)$, we will have $p_0 = q = 2$ and $p_1 = 1$. Since $p_0 \leq q$, we note that in this case the order of accuracy is still asymptotically independent of h .

If we solve problems where the solution within each small box is not explicitly available, we also need to compute it numerically using, for example, a finite difference scheme on fine mesh, with N_f nodes within each small box and time step τ_f . This would be the “nodal–FD” scheme. In this case, the truncation error of the FD scheme will be $O((h/N_f)^{r_0} + \tau_f^{r_1})$, with r_0 and r_1 depending on the scheme. Assuming stability, this is also the error for one coarse time step τ within each small box. The total truncation error for nodal–FD is then

$$O\left(H^{p_0} + \left(\frac{h}{N_f}\right)^{r_0} + \tau^{p_1} + \tau_f^{r_1} + |H^q - h^q|\right). \quad (3.14)$$

In addition to the consistency of the schemes as expressed by (3.13) and (3.14), we also need to consider their stability at both the coarse level and at the fine level independently. Stability conditions will typically restrict the possible sizes of τ , H , τ_f and N_f . Optimal h/H as well as τ_f and N_f will appropriately balance errors and computational cost under these constraints.

3.1.2. General Systems. The diffusion equation example was chosen to illustrate the steps in the gap-tooth scheme and to motivate it in a simple case. Of course, the interesting applications of the scheme arise when the macroscopic balance equations are not explicitly available. The two-tier, inner-outer structure of the gap-tooth scheme is a persistent feature in EFM computations. In general, the inner solver will not come from a finer description of the *same* representation of the physics (i.e. the diffusion equation) but from a different level representation of the physics, i.e. random walkers or molecular dynamics. In these situations the steps in the gap-tooth scheme will be more involved.

- The initial condition in each small box is somehow lifted to a *microscopic* initial condition consistent with the coarse profile (or an appropriate ensemble of such consistent, microscopic initial conditions—distributions conditioned on a few low moments); boundary conditions for each small box are computed from the coarse field.
- Microscopic versions of the BC are applied to the small box problems. New ensembles for doing, say, MC under various boundary conditions will need to be invented, see for example the work of Heffelfinger et al. on Dual-Volume Grand Canonical Molecular Dynamics or -Monte Carlo as an example of such ensembles, [106, 107, 108, 109].
- The small box problems are propagated in time using microscopic evolution rules (MD, MC, LB) with, in principle, constant boundary conditions over the reporting horizon (corresponding to one coarse time step).
- The results at the end of the coarse time interval are averaged within each small box, subsequently interpolated in space between these averaged values. This gives the coarse field timestepper evolution for one step. The restriction operator maps the fine to the coarse description levels.

The procedure is then repeated.

It is conceivable that we may have to average over several realizations—possibly over several initial conditions—of the simulations in each small box to get good variance reduction (i.e., good final expected values for the coarse quantity of interest in this tooth; good enough for the interpolation in coarse space that follows).

It is important, as was discussed above, that the time-reporting horizon, τ , for the microscopic simulation is long enough for any higher order moments incorrectly initialized in the lifting process to have time to heal, i.e. to become functionals of the governing moments, the ones for which we believe that an evolution equation exists and closes. A comparable “healing” process should occur in terms of the spatial interpolation—the time integration should be long enough for interpolation errors in the higher spatial frequency components of the coarse field to get slaved to lower spatial frequency components. Similar thoughts (through a center manifold argument) for dissipative PDEs, at a single level of physics description, have been discussed by Roberts [110].

While this scheme appears computationally intensive (solving in each small box, averaging and interpolating is more expensive than the corresponding coarse FD solution), it is aimed towards the simulation of *microscopic systems* on large computational domains. An extra advantage is that the small box computations for the reporting time can be performed in parallel (a separate processor for each box or patch) significantly reducing the total wall-clock computational time. What was described here was templated on an *explicit* outer solver; in the spirit of implicit projective integrators, we expect that implicit (predictor-corrector) versions of this procedure can also be constructed.

Clearly, the gap-tooth scheme can also be combined with what are traditionally called *hybrid codes* [32, 25, 27, 115]: codes that combine a continuum with a microscopic description at different parts of the domain, matching them at computational interfaces. If in parts of the domain an available coarse description is valid, we evolve the mesh quantities there with the coarse equation outer scheme directly. In the regions where the explicit macroscopic description fails, we use teeth (boxes, patches in higher dimensions) and the lift–evolve–restrict approach. It appears as if the entire domain has been evolved with the outer scheme, but the numbers that this outer scheme crunches come in part from simple function evaluations (in parts of the domain where the coarse equation is valid) and in part from the appropriately initialized and evolved microscopic timestepper (in the regions of the domain where the coarse description is inaccurate and we need to revert to the microscopic physics). Many current hybrid applications have only one tooth: the *entire* portion of the domain where the macroscopic equation fails is evolved microscopically, and matched to the continuum. If this portion of the domain is large, however, the gap-tooth approach may reduce the extent of physical space over which microscopic simulations must be evolved. One can think of the gaptooth scheme as a limiting case of a hybrid code: the macroscopic interpolation between teeth appears (to the computation in each tooth) as the continuum it is coupled with.

One can also envision (although the details need to be worked out) that multi-tiered *outer* solvers (e.g. multigrid solvers), can be combined with the inner microscopic small-box timestepper. As a matter of fact, in the spirit of telescopic projective integrators, different levels of physics descriptions can be used at different levels of a composite, multitier solver. In established multigrid approaches one works with different coarseness levels of *the same* description; here one would work with *different levels of description*; the inner, “ultimate microscopic” solver provides the basic detailed physics; and it is coaxed through the computational superstructure to also provide the relaxation of the basic detailed physics to the coarse slow system-level dynamics we want to study. That such a coarse slow description exists, and the level at which it exists, is a basic premise of the entire numerical skeleton we build

around the microscopic timestepper. This basic premise requires validation during any computation.

There are several levels at which the procedure can fail; some of them are traditional numerical failures (inadequate coarse discrete approximations in space and/or time) and the ways to deal with them on-line during a computation through refinement or adaptive meshing are parts of traditional scientific computation. The more interesting and new feature, that has to be combined with these more traditional numerical aspects, is the “closure failure”, i.e. the possibility that equations do not close any more (as conditions change) at the level of coarse description we have been using up to now. A typical example would be that we need to write equations, in chemical kinetics, not just in terms of densities, but of densities and pair probabilities. Alternatively, in fluid mechanics, that we would need to write equations for stresses as independent variables as the Deborah number grows, and not just have them slaved to momentum gradients. These considerations have been discussed to some extent in [58]; we believe that testing such constitutive failures can be reasonably done numerically, and we discuss how to do it in [65, 86]. We believe that the ability to adaptively test for, detect, and possibly remedy such constitutive failures is one of the strong points of the computational framework we discuss. We only mention here that ways to test for constitutive failures have to be integrated in the framework, along with tests for more traditional discretization-based numerical failures.

There is one more issue that is important to discuss: the repeatedly imposed, macroscopically motivated, boundary conditions for the patches, and the appropriate microscopic ensembles for imposing them. This discussion will follow the next section on Patch Dynamics, since patch boundary conditions are also important in that context.

3.2. Patch Dynamics: A Composition. In Section 2.3 we were able to use short-time computations, embodied in calls to the microscopic timestepper, in order to perform coarse stability and bifurcation calculations. Since the results of this procedure are invariant objects (including elements of the ω -limit set of the unavailable coarse equation) we can consider that the coarse bifurcation results are infinite-time. Correspondingly, projective integration as presented above allows the use of short-time computations, in order to perform numerical integration over larger macro-time steps.

The gap-tooth time-stepper discussed in the previous section is a “short space, short time” to “large space, short time” computational framework. One could, in principle, use the gap-tooth timestepper repeatedly, tiling longer and longer intervals in time completely. In the spirit, however, of our coarse projective integration discussion, it is preferable to combine the gap-tooth timestepper with projective integrator templates to provide an EF framework bridging “small space, short time” simulations and “large space, long time” evolution. The gap-tooth timestepper can also be combined with coarse bifurcation calculations to provide a “short space, short time” simulation “large space, infinite time” computational framework. We call these combinations *patch dynamics*.

A schematic summary of the space-time features of the algorithms can be seen in Fig. 3.2. In the middle (a) we see that, in order to evolve the problem with the detailed physics description, we must tile the space-time area with extremely fine grids in both directions. This would be practically computationally intractable. The basic building block of our EF computational enabling technology is the same detailed description, but in a small box in space for a short time (a small space-time region). Projective

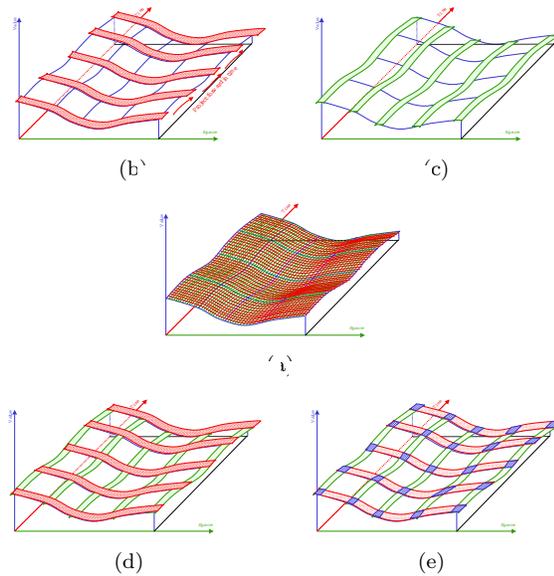


FIG. 3.2. Schematics of coarse integration and patch dynamics techniques. Full microscopic simulation (a) requires impractically fine discretizations. Coarse integration (b) simulates in large space for short times, and then projects to longer times. The gap-tooth scheme (c) simulates in short spacial domains and periodically reinterpolates between the domains. Combining the gap-tooth scheme with coarse integration (d) is better seen in (e): microscopic simulations are performed over short space domains (patches) and for relatively short times (sparse space time elements). Interpolation in coarse space and projection in coarse time is used to advance the macroscopic quantities.

integration is shown in caricature in (b), where we evolve in large space but for short times, and then we *project*, and start again (the lifting part has been suppressed). The gap-tooth scheme is shown in caricature in (c), where we evolve in small boxes for a short time horizon, and then we stop, estimate coarse fields, interpolate to compute new box boundary conditions (let the boxes talk to each other) and start again. Putting the two together is shown in (d) and—really—in (e): exploiting smoothness in time we need only to integrate small times, and exploiting smoothness in space we need only to integrate in boxes—as a result small space-time patches emerge from the overlap of the temporal and the spatial strips.

It is only *within* these space-time patches (small space, short time) that the actual microscopic evolution takes place—the results from this are postprocessed by the identification part of the software, and passed on to the traditional, continuum numerical routines for number-crunching. Since this way the closures for the unavailable macroscopic equations have been estimated in real time (“on demand”), the macro-solvers proceed normally, but with *estimated* numerical information rather than information from functional evaluations.

The higher the dimensions of the problem we are evolving, the higher the potential savings compared to full space microscopic simulations. Consider for example, agent based models arising in ecology or epidemiology, where the required solution moments are distributed not only over space and time, but also over several features of the populations in question (e.g. cell size, age, or income); if we only solve in a fraction

h/H of each coarse dimension, the savings grow as powers of this fraction. It will be extremely interesting to explore and devise patch boundary conditions for such alternative descriptions of distributed systems.

Fig. 3.3 contains a schematic summary of the patch dynamics procedure which we now outline. First we need to define a few of the components used in the scheme.

- (a) *Coarse variables.* Select the coarse variables that we are interested in, as well as a finite dimensional representation $\{U_j^N\}$ of them as in Section 2.1. The variables $U(t, x)$ now also depend on coarse space x . The choice of coarse statistics defines the restriction operator \mathcal{M} .
- (b) *Lifting operator.* Select the lifting operator μ that translate the finite representation of the coarse variables, $\{U_j^N\}$, to one or more consistent microscopic initial conditions for each small box, $\{v_j(0, y)\}$, where y is microscopic space corresponding to the macroscopic space x . The basic idea is that a coarse point in x_j corresponds to an interval, a box, in y .
- (c) *Type of patch boundary conditions.* Select the type of boundary condition to use at the small box edges, e.g. Neumann conditions.

The subsequent steps below define the gap-tooth scheme. We assume that a macroscopic initial profile $U(t = 0, x)$ has been prescribed and that its finite dimensional representation is $\{U_j^0\}$. For each N and j we then do the following steps.

- (d) *Boundary conditions.* Compute the values needed in the patch boundary conditions from the coarse field. For instance, interpolate between $\{U_j^N\}$ to obtain the size of the slopes used in implementing effective Neumann conditions.
- (e) *Lift.* The values $\{U_j^N\}$ are lifted to profiles $\{v_j(0, y)\}$ in the small boxes corresponding to the coarse mesh points $\{x_j\}$. The profiles should be conditioned on the values $\{U_j^N\}$, and it is a good idea that they are also conditioned on the boundary conditions selected in (c) and (d) that are motivated by the coarse field (e.g. be consistent with coarse slopes at the boundaries of the microdomain that come from the coarse field, see the discussion below).
- (f) *Evolve.* Each of these micro-patches is evolved for a short time based on the microscopic description (3.1), and through ensembles that enforce the macroscopically inspired boundary conditions—and thus generate $\{v_j(\tau, y)\}$.
- (g) *Restrict.* Use \mathcal{M} to map v_j in each patch to the coarse variables $U_j^{N+1} = \mathcal{M}v_j(\tau, y)$ (or more generally $\{U_j^{N+1}\} = \mathcal{M}\{v_j(\tau, y)\}$).

We now proceed by combining the gap-tooth scheme with the Projective Integrator ideas in the last two steps.

- (h) *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.)
- (j) *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.

The whole procedure, (d) to (j), is then repeated.

Microscopic simulations in each patch yield averaged quantities on a macroscopic sampling mesh. These quantities are then interpolated to define the macroscopic fields. The macroscopic time derivatives can be estimated at the nodes directly from the detailed computational data in the patches. Alternatively, they can be estimated from coefficient time series of the basis function representation of the macroscopic

fields.

3.3. Patch Boundary Conditions. In the description, above, of the nodal-Exact gap-tooth scheme, we directly introduced slope boundary conditions for the microscopic computational domains (teeth, boxes). We will now briefly discuss *how* one would repeatedly impose such boundary conditions in the case of microscopic simulators, and then *what* boundary conditions can be used for the little boxes. The nomenclature is quickly summarized by the one-dimensional sketch in Fig. 3.4. The boundary conditions for the microscopic dynamics will be imposed using buffer regions surrounding the patches; converting the macroscale interpolant into a microscale boundary condition is clearly a key algorithm in this framework. The appropriate boundary conditions for the patches will, of course, be problem specific.

The macroscale average quantities are interpolated (e.g. through piecewise linear or cubic Hermite interpolation) across the spatial gaps to give macroscopic fields. The approach is straightforward in higher-dimensional problems, when the patches are centered at the gridpoints of a rectangular or cuboid grid. The coarse, macroscale variables will be defined at the grid points and interpolated through, for example, tensor product piecewise Hermite polynomials. These interpolations are fast and efficient on regular grids for high dimensions (e.g. [112]).

If the inner solver for the patches is based on a known partial differential equation, then this equation acts as a guide in defining appropriate boundary conditions for the patches. However, even for known PDEs, in many cases it is not clear how to best choose the boundary conditions to tie the inner solution to the macro solution. If the macroscale field is sufficiently accurate, using it to overpose the BCs for a microscale patch may yield acceptable results if the frequent communication between the scales regularizes the solution. That is, if the errors resulting from the temporarily frozen BC grow slowly, so that they remain small and confined to the buffer regions, their effect will be minimized at the next reporting horizon. Such a self-regularizing effect is reminiscent of AMR simulations [113], where the coarse grid solution is interpolated in time and used to overpose the boundary conditions for the fine scale imbedded solution.

Fig. 3.5, borrowed from [24] shows an example of current practice in imposing arbitrary field, macroscopically inspired, boundary conditions to microscopic simulations. Such methodologies have resulted from extensive research in the coupling of continuum with microscopic simulations, leading to current hybrid simulations practice [114, 115, 24, 116, 25, 117]. The so-called *extended boundary conditions* (EBS) are applied in buffer regions, which are not real, physical domains, and which surround the actual computational patch. It is the *core region* over which we will eventually restrict in order to obtain coarse field information. By acting away from the region of interest, across a buffer zone where the microscopic dynamics are allowed to relax, these conditions attempt to minimize disturbances to the microscopic dynamics in the core region. Knowing that the microscopic distribution functions will relax within a few collision lengths dictates the size of the buffer regions. Various methods have been proposed to impose the actual boundary conditions in the buffer regions. These approaches include using distribution functions [116, 25] when these are known, constrained dynamics [114], and feedback control (the Optimal Particle Controller of [24]). Imposing a known distribution will, of course, be accurate; we hope that the theoretical underpinnings of other approaches will gradually become more firm. The *Dual Volume Grand Canonical* ensembles also use buffer zones to impose “chemical potential Dirichlet” boundary conditions.

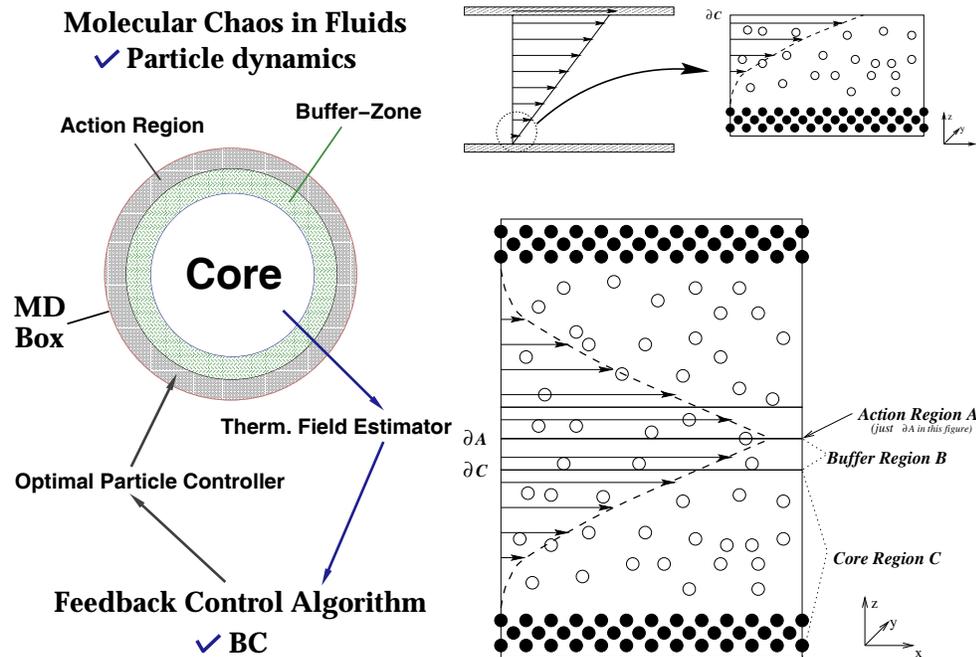


FIG. 3.5. A schematic of the “extended boundary conditions” from [24] and a fluid flow application.

The ingredients of the EBS implementation for continuum/MD fluid simulations involve (a) A field estimator, based on maximum likelihood inference; this corresponds to our restriction operator, obtaining moment fields from particle realizations. For Monte Carlo simulations we have also been using the cumulative probability density function, projected on the first few of a hierarchy of appropriate basis functions as the field estimator. (b) Three distinct regions only the first of which is physical: (the core region of interest, the action region of the OPC, and a buffer zone between the two); a particular illustration is also shown in the Fig. 3.4; and (c) A method for enforcing the macroscopic BC in the action region.

There is an overhead involved with imposing nontrivial (e.g. slope) boundary conditions to microscopic simulations, certainly in the MD case. The conceptual steps will be the same for any microscopic simulator; the implementation may be easier or more difficult depending on the nature of the simulator and on the problem under study. It might make sense to trade worse boundary conditions for larger buffer zones. In the case, for example, of the diffusion problem, instead of lifting to particles over a small box, and imposing slope boundary conditions, we may lift consistently with the coarse profile over a much larger box, but make no effort to impose particular boundary conditions at the outer boundary of the buffer region (i.e. let the particles spill out). The inaccurate boundary conditions will slowly degrade the accuracy of the quantities of interest in the patch; the reporting horizon is chosen so that the computation in the core region remains accurate. Imposing better boundary conditions will permit smaller buffer region and/or longer reporting horizons.

The effect of the boundary conditions on the quality of the computation also depends on how often they are updated (compare, for example, the long term attractors in Fig. 3.9 and Fig. 3.10). This can be analyzed in the ideal case where an exact inner continuum solver is available (for example, the “inner exact” diffusion problem mentioned earlier). Keeping the slope boundary conditions fixed for the exact solver, will eventually result in an infinite (or zero) average in the box; it is vital that the exact simulation in each box is stopped periodically, a restriction followed by interpolation to a new coarse field made, and a new round of inner simulations started. The length of time over which the slope boundary conditions can be used in each box without reinterpolation (without letting the boxes “talk” to each other) is clearly bounded by the accuracy and stability of the outer scheme; for the diffusion equation example, gap-tooth reduces to the standard finite difference method (3.12) with well known stability and accuracy criteria. For this example, there exists a direct connection between updating the BC in the gap-tooth scheme and the stability of an associated finite difference scheme.

When microscopic ensembles imposing macroscopically-motivated boundary conditions are used, the quality of the computation will be affected by the same factors as for an inner continuum gap-tooth scheme, and, in addition, by how the particular macroscopic BC are imposed on the microscale (the particular EBS ensemble).

In summary the quality of the overall patch dynamics scheme depends on the interplay of these factors: (a) the choice of macroscopic boundary conditions; (b) the size of the buffer region at the outer boundary of which they will be imposed; (c) how often the BC are updated from the macroscale (how patches communicate with each other) and (d) the details of how they are imposed on the microscale. The first three factors would also arise in analyzing the gap-tooth scheme for a continuum problem. It is conceptually possible (by analogy to implicit projective integrators) that implicit patch schemes can also be constructed; the interplay between the overall patch size and the time reporting horizon will be different for these.

For the diffusion equation, the slope boundary conditions are natural (they govern the flux of physical material in and out of the patch). In general, however, we can only guess at the number and type of boundary conditions we should use, since we do not know what the coarse equations actually are. What we *do* know (or, at least, postulate) is what the coarse variables are (at what level of coarse description a deterministic equation exists and closes). New challenges therefore arise for identification: from just the nature of the coarse variables and the microscopic simulator, we must extract enough information about the unavailable equation to determine the appropriate boundary conditions. This is an ongoing research subject, which also has a counterpart in the legacy code context: if we are given a black box PDE solver, which we can initialize at will, how can we use the solver to find out how many, and what, boundary conditions to impose in a patch dynamics context? A methodology for designing computational experiments to obtain answers to this type of qualitative question (what boundary conditions to use, is the unavailable equation Hamiltonian etc.) can be found in [118].

If the microscale model is in divergence form, then the natural boundary conditions would be in terms of fluxes. For example, the boundary conditions for the heat equation, $u_t = (Ku_x)_x$, would involve prescribing the flux (Ku_x) at the boundary. Similarly, for the KdV equation, $u_t = (u^2 + u_{xx})_x$, boundary conditions for the flux $u^2 + u_{xx}$ result in a well posed problem. Another consideration for these conservation laws is to retain global conservation of the relevant quantities, including the material

not being simulated between the patches. Not keeping track of this material between the patches will result in only approximately conservative algorithms. For the full calculation to preserve global conservation, the material in the gaps between the patches must be accounted for. This can easily be done in one dimension by calculating auxiliary conserved variables in the gaps between the patches; these conserved variables in each gap are updated on every macroscopic timestep by continuously accounting for the fluxes in and out of its neighboring patches. These auxiliary variables play a key role in constructing the macroscale interpolant defining the patch boundary conditions. A conservative interpolant is constructed to agree with the average values of the material in each patch *and* the integral of the interpolant in the gaps between the patches. Conservative patch dynamics algorithms still need to be developed in higher dimensions. A formulation for conservation problems based on the estimation of fluxes, leading to a generalized Godunov scheme, is proposed in [119]. For a recent discussion of a particular type of “effective smoothness” boundary conditions for gaptooth particle simulations, see [120].

3.4. Numerical Examples. We use as our basic illustrative example a simple, well-known set of nonlinear coupled partial differential equations (the FitzHugh-Nagumo, FHN) system in one spatial dimension [1, 66, 67].

$$u_t(t, \mathbf{x}) = u_{xx}(t, \mathbf{x}) + u(t, \mathbf{x}) - u^3(\mathbf{x}, t) - v(t, \mathbf{x}) \quad (3.15)$$

$$v_t(t, \mathbf{x}) = \delta v_{xx}(t, \mathbf{x}) + \varepsilon(u(t, \mathbf{x}) - a_1 v(t, \mathbf{x}) - a_0) \quad (3.16)$$

The boundary conditions are $u_x = 0$ and $v_x = 0$ at $x = 0$ and $x = L$. Here $u(t, \mathbf{x})$ and $v(t, \mathbf{x})$ are the local concentrations of the two participating reactants (the “activator” and the “inhibitor”), δ is a diffusion coefficient and the remaining constants pertain to the kinetic terms; we can clearly recognize the terms corresponding to diffusion and those corresponding to chemical kinetics. In all the simulations that will be presented in this paper, the diffusion coefficient is set to $\delta = 4.0$ and the values of the kinetic parameters are chosen to be $a_0 = -0.03$ and $a_1 = 2.0$. Finally, ε was varied as in some of our simulations it represented the continuation/bifurcation parameter. The use of such distributed parameter models for both analysis and design purposes then hinges on the exploitation of numerical discretization techniques to turn them into large sets of ordinary differential or differential-algebraic equations.

One of the reasons for choosing this model in [1] was that it is possible to analyze it through both the closed form PDE (3.15), (3.16), as well as through a kinetic theory-motivated Lattice Boltzmann-BGK code [68]. The PDE, at the appropriate limit, is a closed equation for the zeroth moments of the discrete velocity distribution—so it can be considered as a coarse closed deterministic model; the LB-BGK model is the “fine” model in this case. Another reason for choosing this example is that it exhibits a rich variety of nonlinear dynamic patterns, including multiplicity of spatially structured steady states (sharp, front-like ones, both stable and unstable) as well as spatiotemporal oscillations. The rich nonlinear dynamics and the spatial sharpness of the solutions make this simple but nontrivial one dimensional example a good testing ground for new tools.

The LB-BGK model we have used here (and in the works referenced above) is based on the discrete formulation on a lattice, of the Boltzmann equation [2], a continuous equation describing the evolution (on a lattice grid) of the distribution function of a single particle. The evolution equation for u (and respectively for v) is:

$$N_i^u(t+1, x+c_i) - N_i^u(t, x) = -\omega^u [N_i^u(t, x) - N^{u,eq}(t, x)] + R_i^u(N^{u,eq}, N^{v,eq}) \quad (3.17)$$

In the above equation $i = 1, \dots, 3$ for a one-dimensional model. $N_i^u(t, x)$ is the average population of particles with velocity c_i at position x on the lattice at time t . In LB models, particles can move to their adjacent sites in the lattice according to their velocity c_i ; this is called *streaming*. In our one-dimensional implementation, there are only two adjacent sites, left and right. For unitary spatial and temporal increments $c_i = \{0, 1, -1\}$. Therefore, particles N_1 , with velocity $c_1 = 0$ will stay in place (rest particles) while particles N_2 ($c_2 = 1$) and N_3 ($c_3 = -1$) will move to the right and to the left, respectively. The quantity $N^{u,eq}(t, x) = (1/3) \sum_{i=1}^3 N_i^u(t, x)$ is the local equilibrium population, homogeneous in all velocity directions. Thus, the first term in the right-hand side of equation (3.17) expresses the relaxation of N_i towards a local pseudo-equilibrium; ω^u (and correspondingly ω^v) is the BGK relaxation parameter which is proportional to the diffusion equation ($0 < \omega \leq 2$). From the left-hand side and the first term of the right hand-side of equation (3.17), the diffusion equation can be obtained through a Chapman-Enskog expansion [16]. The reaction term $R_i^u(N^{u,eq}, N^{v,eq})$ can be viewed as a forcing term for the diffusion process. The kinetic parameters ε , a_0 and a_1 are incorporated in the corresponding term $R_i^v(N^{u,eq}, N^{v,eq})$. The (strong) assumption that reactions occur locally, respecting the local diffusive equilibrium is used in this formulation [68]. The two concentrations (u and v) at each lattice site can be uniquely computed as the zeroth moments of the corresponding population densities (6 at each lattice site):

$$u(x, t) = \sum_{i=1}^3 N_i^u(x, t) \quad v(x, t) = \sum_{i=1}^3 N_i^v(x, t) \quad (3.18)$$

This is the *restriction* process $\mathcal{M}N^u$ and $\mathcal{M}N^v$. The lifting process μu and μv , where we extract the population densities from the macroscopic solution is given by

$$N_i^u(x, t) = w_i u(x, t) \quad N_i^v(x, t) = w_i v(x, t) \quad (3.19)$$

where, the weights, w_i can be any three numbers summing up to 1. Thus, there is more than one way to compute population balances from the concentrations. As mentioned above, for our LB-BGK problem the choice of weights is not crucial, since all initial disturbances caused by the random choice of weights decay much faster ($t \approx 0.1$ as shown in Fig. 3.6) than the macroscopic reporting horizons we have used in all our simulations ($t = 15 - 25$). So, in all the simulations presented below we have chosen $w_i = 1/3$.

To validate EF computations for any given problem we always start in a parameter regime where we *know* a valid macroscopic description explicitly, and we reproduce its results. We then perform homotopy/continuation towards the regime of interest in parameters/operating conditions, constantly “numerically” as well as “constitutively” validating the procedure. As we gain more experience from applications, our knowledge of levels of closure appropriate at different parameter regimes will give enough confidence to avoid the overhead of constant validation.

3.4.1. Gap-tooth scheme computations. We begin with results for the gap-tooth scheme without coupling it with projective integration or RPM. Fig. 3.7 shows our first gap-tooth integration results for the FHN equations. In the simulations the domain length was $L = 20.0$ and the parameter ε was set to $\varepsilon = 0.5$. At this parameter value a steady front is produced. The domain was discretized in 30 nodes. Therefore 30 boxes were used, each of them having length, h equal to $1/4$ of the

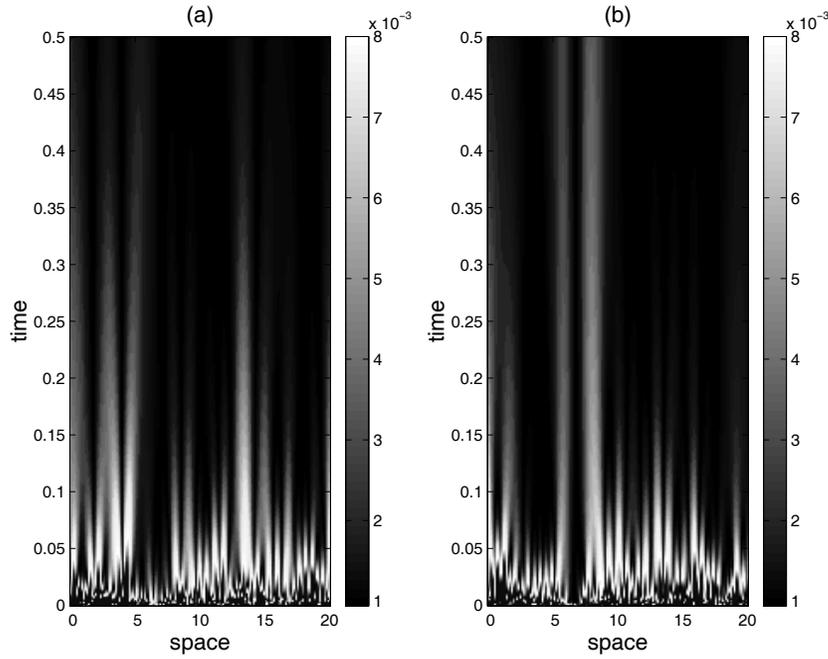


FIG. 3.6. Decay (“healing”) of the lifting error (from [4]): comparison between the evolving restrictions of a “mature” LB-BGK trajectory, and two distinct liftings of its restriction: (a) a local equilibrium fine lifting and (b) a more coarse random lifting. Healing has visibly taken place at times much shorter than the timestepper reporting horizon (here 0.5).

internode distance, H (i.e. $h = 0.1724$). For each overall step, starting from values of the density (zeroth moment for the LB-BGK code) on a coarse grid we computed a coarse density profile through interpolation (taking advantage of smoothness); we also used this coarse profile to compute temporary (slope) boundary conditions for the little boxes using (3.3) - (3.5). We lifted the density profile to a consistent LB profile in each box, and then evolved the problem through LB-BGK for one reporting horizon in each box, keeping the temporary boundary conditions constant; the outer scheme here is an explicit finite difference scheme. Implicit (iterated to self-consistency) versions of this evolution may also be possible [3, 4].

After the problem was left to evolve for one time horizon ($T_b = 10^{-3}$) in each box using the inner solver, we *restricted* the solution, computing u and v in each box using equations (3.18). We then computed the averages of the coarse solution (u and v) for each box, by integrating over the length of the box to obtain the new coarse profile. We have thus created an “inner solver-assisted” map from coarse grid to coarse grid. The procedure was then repeated.

We verified the effectiveness of this approach in a nodal-FD mode (inner code: FD in (fine) space, trapezoidal in (fine) time; nodal macroscopic field representation) as a sanity check. We then performed the computations in a nodal-LB code (we will call this an LB-BOX code): inner LB in fine space and fine time, with lifting and with slope boundary conditions; and outer nodal representation of the coarse fields. Some of the results presented in [5] are shown in Fig. 3.7 both for “inner FD” and “inner

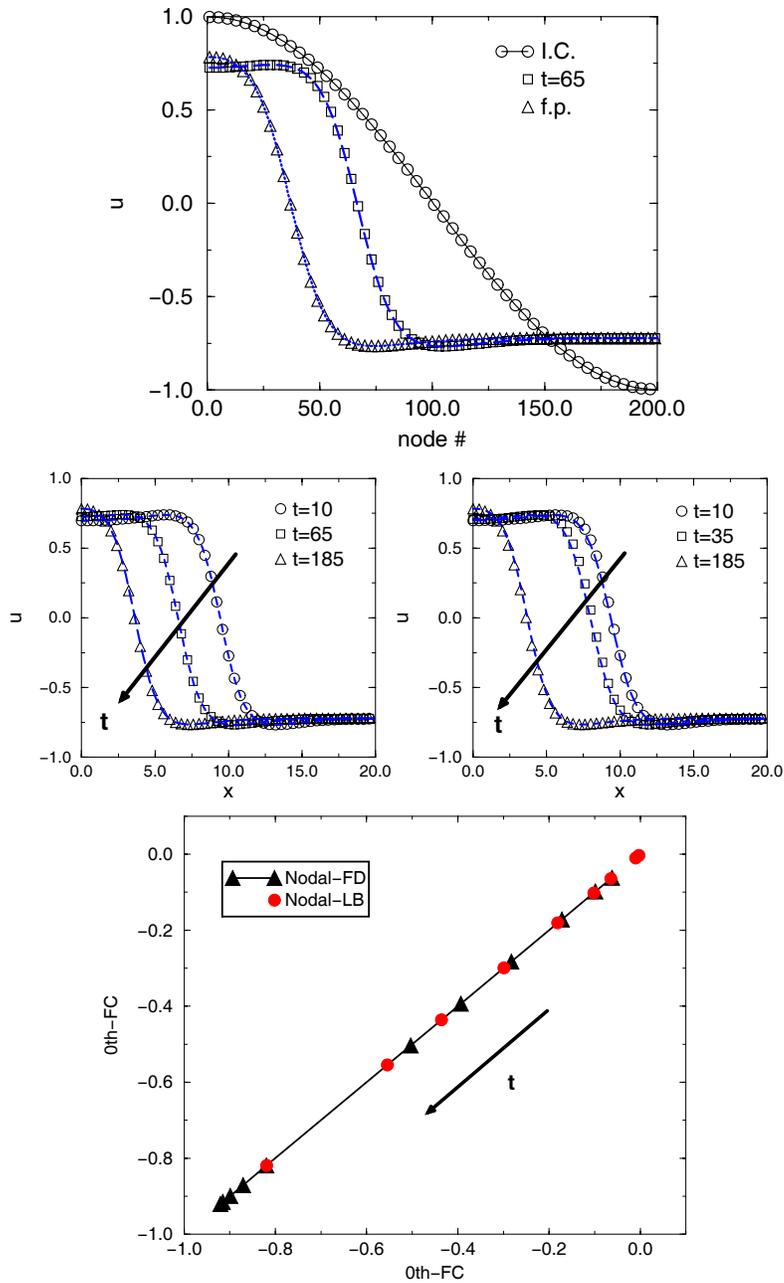


FIG. 3.7. “Inner LB” gap-tooth integration results for the FHN equation (symbols) compared to full LB simulations (curves) for various initial conditions. The last figure shows a phase portrait for this simulation for two different gap-tooth schemes: an “inner LB” (FD-LB) and an “inner FD” (FD-FD) continuum scheme.

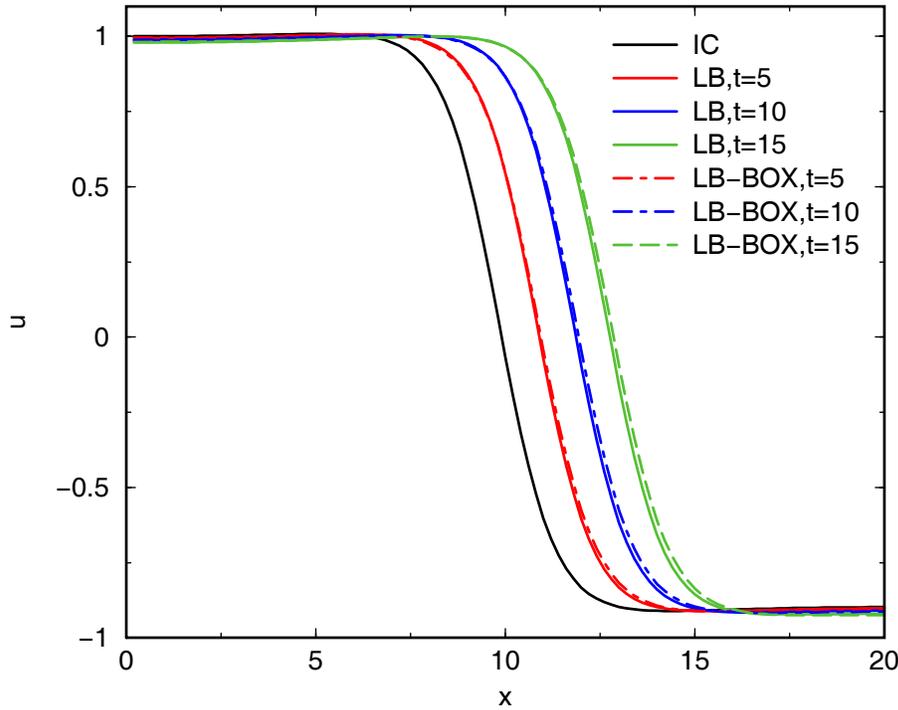


FIG. 3.8. Comparisons of LB and LB-BOX spatial profiles of u at various times in the unstable (oscillatory) FHN region ($\varepsilon = 0.01$) showing a palindromic movement of the reaction front.

LB” evolvers for comparison purposes.

We then embarked to more elaborate LB-BOX simulations to demonstrate the effectiveness of the process. In the simulations for the gap-tooth procedure shown in subsequent figures, 101 boxes were used with $h/H = 0.1$, i.e. the box length was $h = 0.02$. Fig. 3.8 shows comparisons of full LB and LB-BOX profiles of the variable u at various times in the *oscillatory* regime of the FHN ($\varepsilon = 0.01$). The solution at this parameter is a spatiotemporal limit cycle, and Fig. 3.8 shows a portion of it, involving the palindromic movement of a sharp reaction front. Fig. 3.9 shows a projection of this spatiotemporal long-term periodic attractor computed through full LB and through the gap-tooth LB-BOX approach. The effect of the coarse time reporting horizon, $T_c = 7.5 \times 10^{-4}$, can be seen in Fig. 3.10 where a comparison with the *long-term attractor* of the same scheme *but with a larger coarse time horizon* ($T_c = 2.5 \times 10^{-3}$) is shown. In the latter case the communication between the boxes, by constructing the new coarse profile and the new box boundary conditions for the next microscopic simulation “era”, occurs a factor of three less frequently. It is important in comparing these two figures to remember that what is shown are *long term attractors*. The short term integration with the two different reporting horizons will, for quite some macroscopic time, appear visually practically the same at the resolution of these plots.

Comparable results have been obtained for the diffusion equation and also for the

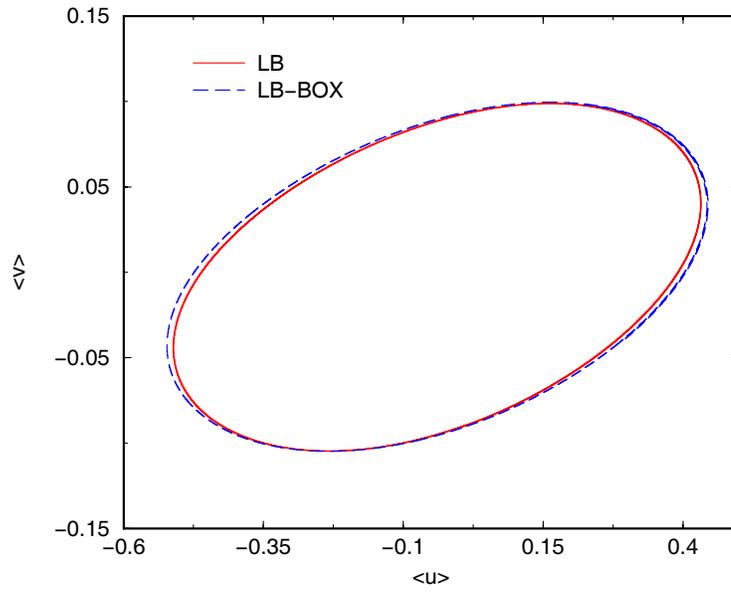


FIG. 3.9. Comparison of phase space projections of the LB (solid) and LB-BOX (dashed) long-term attractors. Reporting horizon for communication between boxes $T_c = 7.5 \times 10^{-4}$.

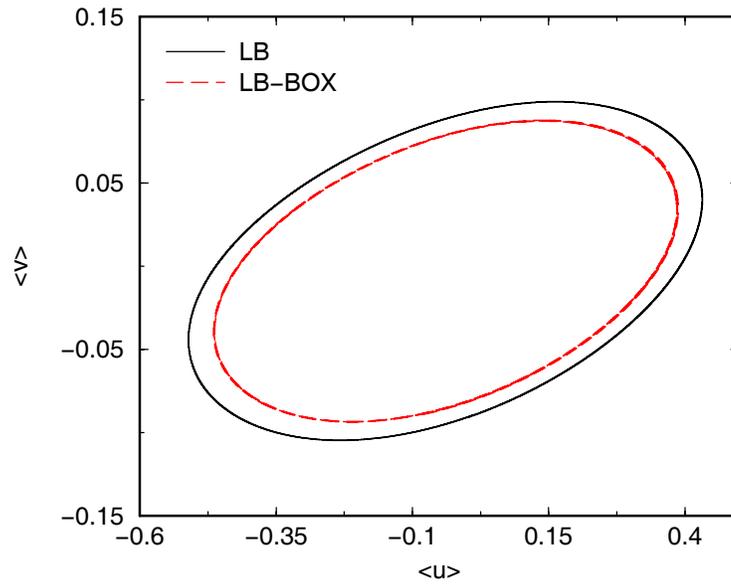


FIG. 3.10. Comparison of phase space projections of the LB (solid) and LB-BOX (dashed) long-term attractors. Reporting horizon for communication between boxes $T_c = 2.5 \times 10^{-3}$.

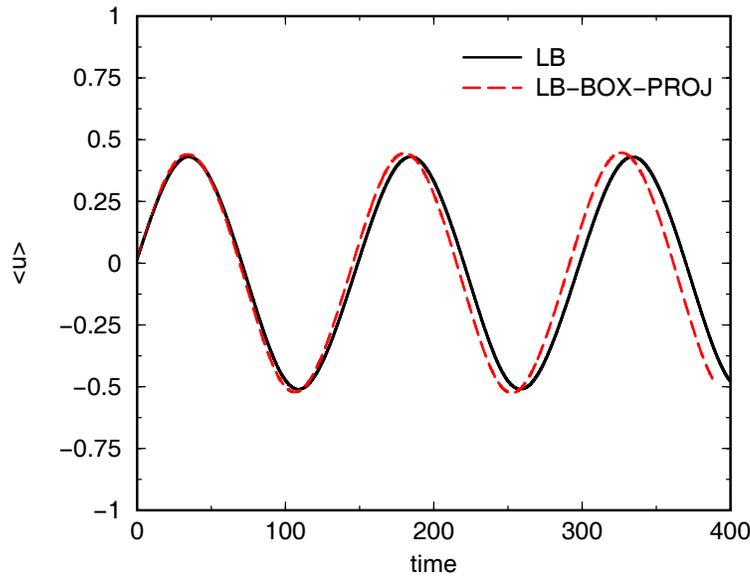


FIG. 3.11. Computed time-series comparison of the oscillatory FHN dynamics ($\varepsilon = 0.01$) computed by a full LB code (solid line) and by a 100-100-200 LB-BOX projective scheme (dashed line).

Allen-Cahn equation [111] of the form

$$u_t = u_{xx} + 2(u - u^3), \quad (3.20)$$

where the relaxation of the macroscale profile to the exact steady state of the form $u = \tanh(x - x_0)$ was observed.

3.4.2. Patch Dynamics Projective Integration. We next combine the gap-tooth scheme with projective integration to perform patch dynamics simulations. Fig. 3.11 compares a full LB simulation with such a combination. The algorithm now has several tiers: There are two tiers in time: the inner integrator is a gap-tooth LB (with 101 boxes, $h/H = 0.1$ and $T_c = 7.5 \times 10^{-4}$), and the outer integrator is forward Euler on the coarse profile (to be exact, on the macroscopic mesh density values). There are, however, tiers also in space: the coarse density profile interpolated from the density values at the coarse mesh points; the (auxiliary) coarse profile in each box; and the Lattice-Boltzmann profile in each box, the one that we lift to, and which we evolve using the microscopic evolution physics. In particular we call this a 100-100 inner 200 outer integrator, since the forward Euler step is equivalent to 200 time steps of the inner integrator (see also [4]). Fig. 3.12 shows snapshots of the corresponding profiles at different times, and Fig. 3.13 shows the long-term periodic attractor, successfully captured by the LB-BOX-Projective scheme.

3.4.3. Patch Dynamics Stability and Bifurcation Analysis. We finally turn to coarse bifurcation analysis of our FHN equation, based on the LB model discussed above together with the gap-tooth scheme and RPM. We have performed this coarse bifurcation analysis (including stable and unstable steady states, limit

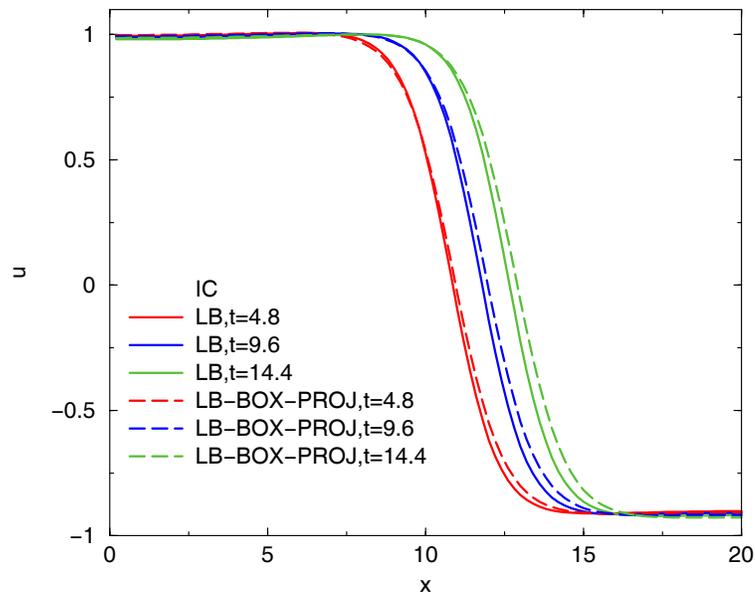


FIG. 3.12. Comparison of spatial profiles of u at various times in the unstable (oscillatory) FHN region ($\varepsilon = 0.01$) computed by a full LB code (solid lines) and by a 100-100-200 LB-BOX projective code (broken lines). The boxes cover 10% of the macroscopic computational domain.

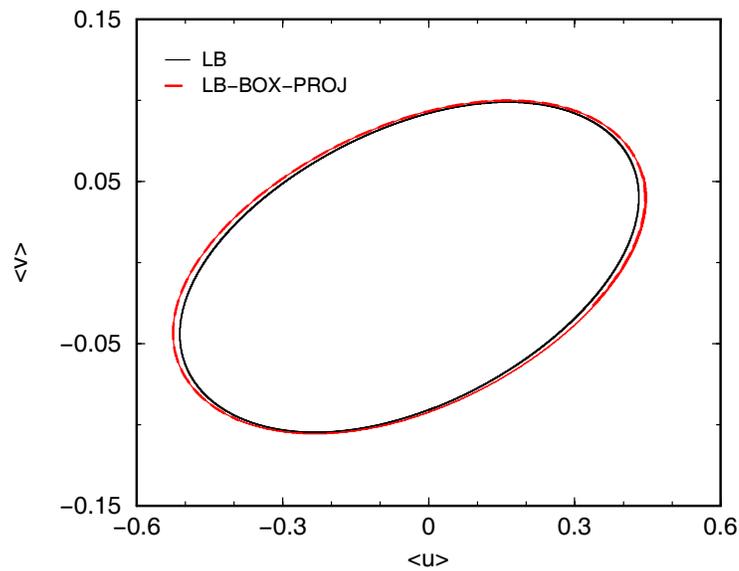


FIG. 3.13. Comparison of phase space projections of the long term attractors of a full LB (solid line) and the 100-100-200 LB-BOX projective (dashed line) attractors. Reporting time for communication between boxes $T_c = 7.5 \times 10^{-4}$.

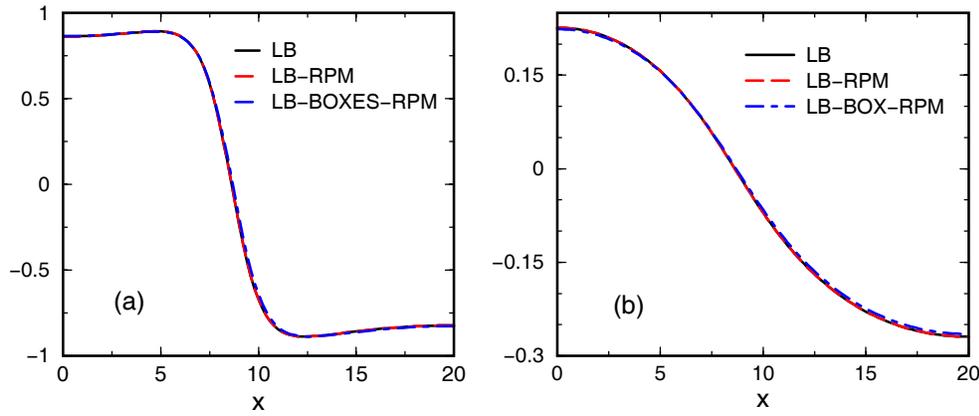


FIG. 3.14. Comparison of the coarse steady state spatial profiles of (a) u and (b) v at $\varepsilon = 0.05$ (a stable steady state) computed with direct LB simulations (solid lines, LB combined with RPM (LB-RPM) simulations (dashed lines) and our LB-BOX gap-tooth scheme combined with RPM (LB-BOX-RPM dash-dotted lines).

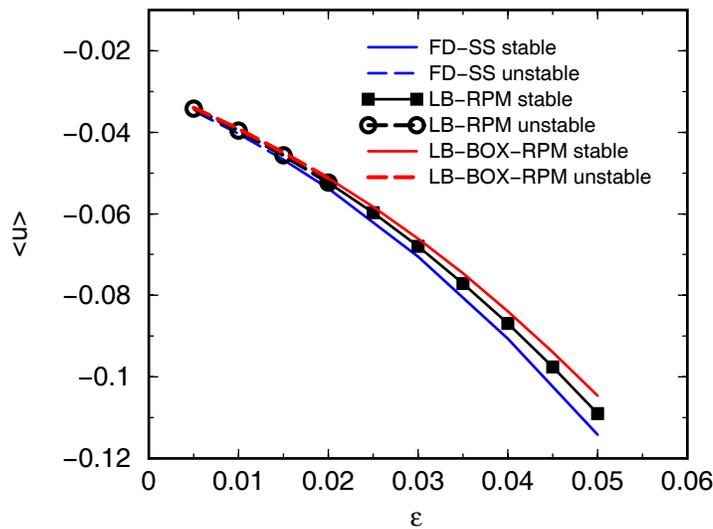


FIG. 3.15. FHN bifurcation diagram near a Hopf bifurcation point computed with a steady Finite Difference (FD-SS) code (bottom line), our LB-RPM code (middle line) and our LB-BOX gap-tooth scheme combined with RPM, called LB-BOX-RPM (top line). Stable (unstable) steady states: solid (broken) lines.

cycles, computation and continuation of codimension one coarse bifurcation points) and compared it with Newton-based analysis results using FD discretizations of the PDE, (3.15) and (3.16), [1, 4]. A summary of these results is included in Fig. 3.15 (bifurcation diagram of the FHN with respect to the bifurcation parameter ε , showing a Hopf bifurcation), Fig. 3.14 (showing representative u and v spatially varying steady

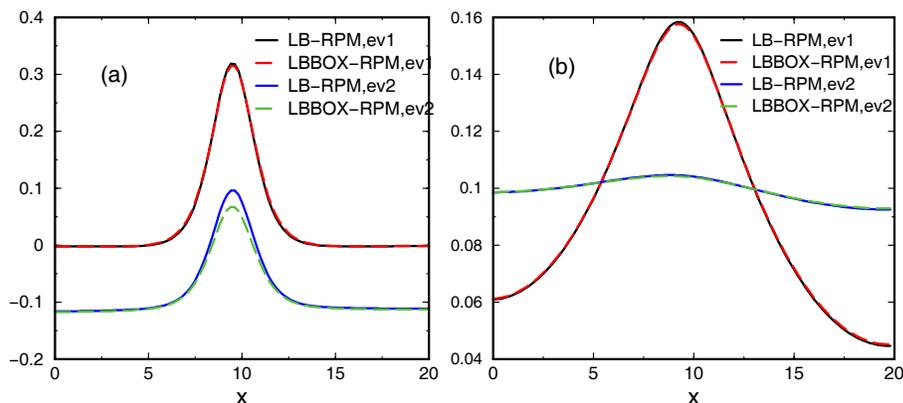


FIG. 3.16. Comparison of (a) u and (b) v components of the critical coarse eigenvectors ($ev1$ and $ev2$) at $\varepsilon = 0.01$ obtained upon convergence of the LB-RPM and LB-BOX-RPM computations.

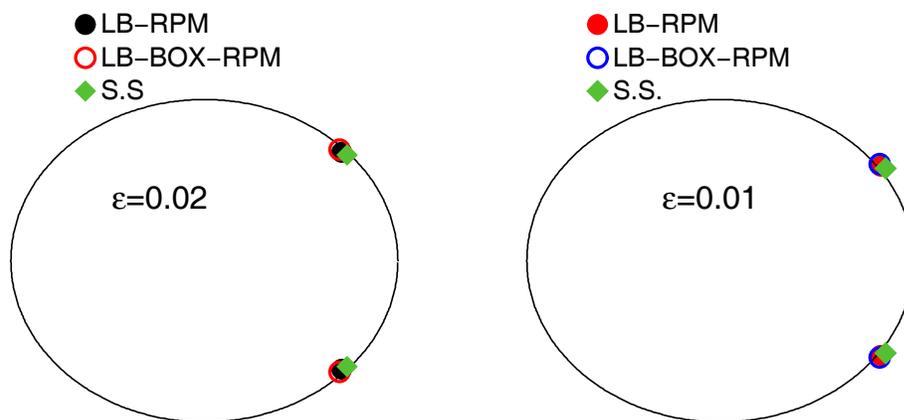


FIG. 3.17. Leading coarse eigenvalues before ($\varepsilon = 0.02$) and after ($\varepsilon = 0.01$) the Hopf point, obtained upon convergence of LB-RPM (filled circles) and LB-BOX-RPM (open circles). The exponentials of the eigenvalues (for the same reporting horizon) of a steady FD code (based on the PDE) are also depicted for comparison (diamonds, SS).

states for a value of the parameter), Fig. 3.17 (showing the leading eigenspectrum just before and just after the Hopf bifurcation) and Fig. 3.16 (showing the corresponding critical eigenvectors (both the ones identified by the PDE-based method and by LB-RPM)). In all the simulations performed, 200 discretization nodes were used for the FD simulations and 100-400 lattice points for the large-space LB coarse timestepper simulations with and without RPM. The reporting horizons LB-RPM (without gap-tooth) were $T = 15 - 25$. For the small-space patch dynamics simulations, which we call LB-BOX-RPM, (a two-tier nodal-LB scheme) 101 lattice points were used for the LB timestepper and correspondingly 101 boxes with $h/H = 0.1$ (so the inner solver operates on only one tenth of the full spatial domain). The time horizon in each box

was $T_c = 7.5 \times 10^{-4}$ and the reporting horizon of the LB-BOX-RPM timestepper for communicating with the bifurcation algorithm (RPM) was $T = 15$.

It is important to notice that, beyond the main features of the diagram, the actual shapes of the computed spatially varying steady states, both stable and unstable, are well captured as it can be seen in Fig. 3.14. Furthermore, their stability (embodied in the coarse linearization) is also well captured. This is true both for the LB and for the LB-BOX-RPM scheme. The LB-BOX-RPM scheme clearly captures quantitatively the linearized stability of the detailed problem at low wavenumbers (at the coarse level) as well as the large-space coarse LB scheme does.

This completes our illustrative computations with the simulation part of the EFM toolkit. The results have been obtained in a regime where the coarse PDE is known and can be used as a check, where a lot of variance reduction is not really necessary, and where we therefore find a context for troubleshooting the computational procedure and exploring the numerical analysis of the multi-tiered schemes. The accuracy and stability analysis of patch dynamics algorithms obviously will rely on the corresponding analyses of the gap-tooth scheme and the projective integrator schemes, and is the subject of current research.

4. Some applications

Over the last few years we have tested elements of the EF computational enabling technology using several types of microscopic simulators in contexts ranging from surface chemistry to small molecule folding in solution. The oscillatory behavior of a gas bubble rising in a liquid has been studied using an inner LB-BGK simulator and RPM-based coarse stability bifurcation analysis in [63, 64, 121, 122]. The coarse bifurcation analysis and coarse integration (both forward and backward in time) of kinetic Monte Carlo simulations of catalytic surface reactions, including the computation of coarse limit cycles (boundary value problems in time) has been the subject of a series of publications [58, 65, 95, 123]. An alternative coarse computational task (coarse controller design) for the same kinetic Monte Carlo simulations modeling surface catalytic reaction kinetics can be found in [124]; see also [125] for the interplay between control-based techniques and the tracing of coarse bifurcation diagrams. Unstable stationary states and estimates of the slow linearization around them are used in local linear observer and controller design, stabilizing these saddle-type states. Bifurcation diagrams for stochastic PDEs, arising in the context of the rheology of nematic liquid crystals have also been computed through Equation Free methods in [86], where coarse integration as well as coarse control of the corresponding Smoluchowski equation are demonstrated. The use of a coarse timestepper based on an inner molecular dynamics simulator has also been demonstrated as a method for performing free energy, kinetics, and long-time dynamics computations in the context of folding of an alanine dipeptide molecule in water [126].

Equation-free computational homogenization, or, more precisely, the solution of effective medium equations, can also be performed using coarse timesteppers without explicitly obtaining a closed effective (at some limit, homogenized) equation. This type of “on demand” effective medium computation was introduced and illustrated in [57] for reaction-diffusion pulse motion in periodic media. An extension of the approach to the effective modeling of pulse propagation in *spatially discrete* media can be found in [128].

In the context of legacy code acceleration, we have assembled an EF timestepper based wrapper around a pressure-swing adsorption model implemented in gPROMS (a state-of-the-art commercial plant simulator). We demonstrated there the acceleration

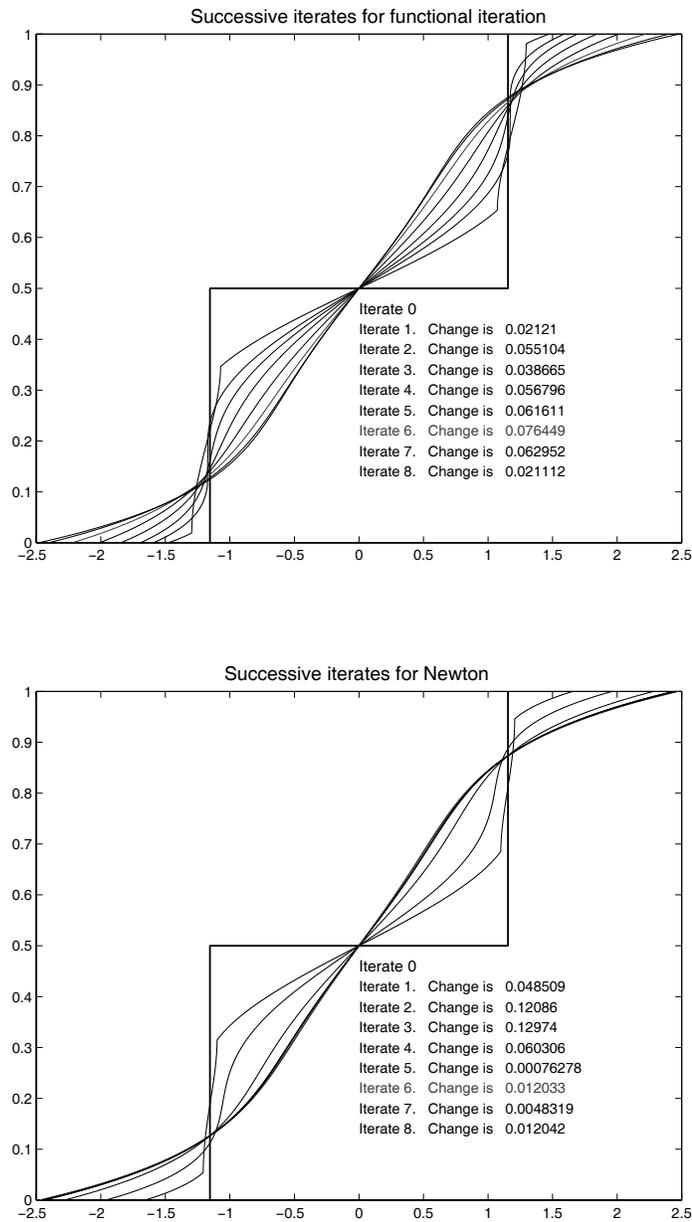


FIG. 4.1. Coarse renormalization flow and coarse self-similar solution calculations based on KMC random walk simulations. The top panel shows the results of coarse renormalization flow projective integration for the one-dimensional diffusion equation, while the bottom panel shows iterations of a fixed point algorithm converging on the same coarse self-similar solution. The evolution of the rescaled cumulative probability density is plotted at various times above, and at various iterations of the fixed point scheme below. The restriction operator was the cumulative distribution function.

of periodic computations as well as the coarse integration of *envelope* equations for large, rapidly oscillatory DAE systems [127].

The last application we include is very simple, but has, we believe, important implications. Fig. 4.1 shows the computation of a *coarse self similar solution*—the decaying self-similar solution of the diffusion equation—computed through a microscopic timestepper (a Monte Carlo random walk simulator). We have recently proposed the use of a template based dynamic renormalization approach to compute self-similar solutions for evolutionary PDEs [129, 130]. This template-based approach was motivated by the work of Rowley and Marsden for problems with translational invariance and traveling solutions [131]. The solution is rescaled either continuously, or discretely (after some time) based on minimizing its distance from a (more or less arbitrary) template function (see also [132]). Here we combine this idea with the lift–evolve–restrict approach as follows: we start with a random walker density profile (a coarse initial condition); for convenience, the coarse variable will be the cumulative distribution function (percent total walkers to the left of a given location in space). We then lift it to random walker location, and let the random walkers evolve for some time. We interpolate to cumulative distribution function (using appropriate orthogonal polynomials) and rescale the solution based on template minimization. The procedure is then repeated giving a discrete-time renormalization evolution, which converges quickly (the solution is stable). Successive coarse portraits are used to extract the relevant self-similarity exponents. Finally, we find the same solution not through successive coarse timestepping, but through a contraction mapping: a secant-based quasi-Newton method, for the profile that is invariant under finite time evolution and template-based rescaling—see the bottom panel in Fig. 4.1. This work [133] is in collaboration with Prof. Aronson in Minnesota, and we hope that it will allow the coarse study of many phenomena that involve self-similarity (“from cells to stars”, [134]) when microscopic physics descriptions exist but the macroscopic equations are not available. It is conceivable that even asymptotically self-similar solutions might be treatable in some cases. We have recently located an interesting bifurcation (the onset of dynamic self-similarity from steady state solutions) in the context of the focusing Nonlinear Schroedinger (NLS) equation [135]. It is conceivable that similar bifurcations, involving the onset of dynamic coarse self-similarity from coarse steady states (e.g. in molecular dynamics equilibrium simulations) might share some features with this instability and offer a useful computational perspective of the transitions involved.

5. Conclusions, discussion, outlook

We have described an equation-free framework for the computer-aided analysis of multiscale problems, in which the physics are known at a microscopic level, but where the questions asked about the expected behavior are at a much higher, macroscopic, coarse level. This is a closure-on-demand, system identification based framework, which provides a bridge between microscopic/stochastic simulators and macroscopic scientific computation and numerical analysis. The basic idea is to use short, intelligent, bursts of appropriately initialized microscopic simulations (the inner code), and process their results through system identification techniques to estimate quantities (residuals, time derivatives, right-hand-sides, matrix-vector products, actions of coarse slow Jacobians and Hessians) that, if a macroscopic model was available, would simply be evaluated from closed formulas. The quantities thus estimated are passed to an outer level code, usually a traditional macroscopic computational code that performs integration, controller design, steady state calculation or optimization

tasks. This outer code proceeds to perform its task without realizing that the model it is studying is not available in closed form at the macroscopic level. We showed also that these short bursts of microscopic simulation may, under some circumstances, be performed for small patches of computational space; these patches repeatedly communicate with each other, as the computation evolves, through macroscopically inspired boundary conditions (patch dynamics).

In our discussion, the main tool was the coarse timestepper, that evolves fields of the macroscopic variables through microscopic simulation. Separation of time scales (often associated with spatial scales) constitutes the main underpinning of effective coarse behavior. Whether (a) higher order moments of microscopically evolving distributions become quickly functionals of lower order, slow, governing moments; or (b) higher wavenumber modes of a dissipative PDE quickly evolve towards a slow manifold, becoming functions of the lower wavenumber, slow, master modes; or (c) higher moments of a Smoluchowski equation for the probability distribution of a stochastically evolving problem become slaved to lower order moments, it is the physical dynamics that lead to the quick, on demand closure that makes coarse equations meaningful. If the timesteppers are legacy simulators (one of the original motivations of time-stepper based enabling algorithms, like the adaptive condensation of Jarausch and Mackens, the RPM of Shroff and Keller, but also the Arnoldi based stability computations of Christodoulou and Scriven [136]) we are in the realm of “numerical analysis of legacy codes”. EF-type algorithms can significantly accelerate the convergence of large scale industrial simulators to fixed points (e.g. periodic steady states for pressure swing adsorption or reverse flow reactor simulations) as well the stability analysis of these fixed points (e.g. [127]).

This paper was focused on situations where the macroscopic model is not available in closed form, and one needs to use microscopic simulation over the entire spatial domain, or over patches distributed across the domain. As discussed above, it is also conceptually straightforward to construct coarse timesteppers using existing *hybrid* simulation frameworks. The map from current to future coarse fields is obtained through the combination of the lift–evolve–restrict approach in the “molecular” regions and continuum solvers in the continuum regions. If the molecular regions are large, and the macroscopic expected solution is smooth enough across these regions, one can conceivably evolve the microscopic simulation only in a grid of patches along this region; one then has a hybrid continuum-patch overall coarse timestepper.

System identification based on the observation of the microscopic simulation, variance reduction, and the extraction of local coarse models, are the second important feature of our closure-on-demand approach. On a different note, a connection between coarse timesteppers and what is termed “gray-box identification” also exists. Frequently, we may know (e.g. from irreversible thermodynamics, or from experimental experience) a lot about the form of several terms in the RHS; and only miss a particular phenomenological term, such as an effective viscosity, to have a useful closure. In such cases, microscopic simulations are typically done to extract a law for this phenomenological coefficient. For engineering purposes, discrete-time data obtained through the coarse timestepper can be fed to discrete time gray-box identification schemes, with built-in partial knowledge of the RHS of the macroscopic equation; only the unknown parts of the postulated equation will then be identified (fitted). A particular framework for doing this, based on recurrent artificial neural networks templated on numerical integration schemes can be found in [137, 138].

We discussed mostly parabolic problems in presenting coarse integration results.

Yet matrix-free fixed point solvers (Newton-Krylov methods like GMRES) can be used to solve elliptic problems that arise as steady states of parabolic problems. More generally, numerical tasks that can be formulated in terms of the discrete time-one map of the solution of an equation (independent of its type) can in principle be solved through matrix free algorithms.

Most of the “mathematics assisted” enhancement we expect from these algorithms comes from the estimation of coarse derivatives (in time, in space, with respect to parameters or with respect to the coarse variables themselves) through calls to the microscopic timestepper. Variance reduction through sheer number of copies, but also through the best estimation we can have, is thus vital in establishing computational savings [34, 139]; its importance cannot be overestimated.

In most of our exposition we assumed that the coarse variables of choice would be low order moments of the distributions evolved through the microscopic timesteppers. There will, however, exist problems where intelligently chosen phase field variables will provide a much leaner set of equations than moments (in the same sense that good empirical basis functions can provide a much more parsimonious description of a PDE discretization). We believe that the discovery (through statistical algorithms that search for nonlinear correlations across data fields) of good coarse variables from which to lift is one of the requests this technology will have from modern computer science. The fact that we need not come up with explicit formulas for the time derivatives in terms of the variables generates a lot of freedom in allowing the study of systems currently intractable (because good closures could not be explicitly computed, and full microscopic simulations were too expensive to perform over relevant space-time scales).

A simple case in point is the attempt to create timesteppers for expected evolution of effective equations for the energy spectra of DNS or LES simulations of the Navier Stokes equations; here DNS is the inner, microscopic simulator. Many other promising directions can and should be pursued, including agent based models in ecology, economics, epidemiology and evolutionary biology (the timesteppers do not have to be in physical time, they can involve mutations in a population). While most of the problems we have suggested so far come from nonequilibrium statistical mechanics, equation-free dynamic approaches may play a role in studying *equilibrium phenomena* through dynamics (see recent work in [140, 141, 142, 143] as well as in [144]). It is also conceivable that appropriate limits of quantum problems, such as the weak coupling limit of the N -particle Schroedinger, which in the semi-classical limit yields the 1-body Vlasov equation [145], can be studied through coarse timesteppers. In fact the Vlasov equation (as well as the Fokker-Planck-Vlasov) has analytically known equilibrium states; one could try to retrieve them using time-steppers and compare to the explicit analytic results. Another possibility is the semiclassical limit of the scalar wave equation or a single Schroedinger equation, which, using the Wigner transform, gives rise to a Vlasov equation. Further ad-hoc closures yield a system of conservation laws with a dispersive correction [146, 147, 148]; coarse timesteppers could sidestep (and/or validate) the derivation of such closures.

The availability of continuum equations underlies many of the methods developed and the results obtained in the modeling of physical systems. These continuum equations are, in many cases, only caricatures (sometimes excellent caricatures) of “better but dirtier” descriptions of the physics, such as those provided by microscopic/stochastic models. Equation-Free approaches hold the promise of applying our best computational/mathematical tools (whose development was targeted at con-

tinuum evolution equations) to the “dirty” description directly, circumventing the necessity of passing through a caricature. The transition between analytical, paper and pencil, solution methods (based on perturbation theory and special functions) to computer-aided analysis (based on large scale linear algebra and PDE discretization techniques) discussed in [149, 150] serves, in some sense, as a prototype for a new transition: from equation-based computer-aided analysis to equation-free computer aided analysis. In the first transition, quantities that would be computed by paper and pencil if the *solution* was explicitly available (e.g. eigenvalues of a linearization) were computed numerically. In the second transition, quantities that would be computed numerically if the *equation* was explicitly available (e.g., again, eigenvalues of a *coarse* linearization) are estimated numerically through appropriate calls to microscopic/stochastic simulators.

Beyond the type of systems that one can attempt to approach through this computational enabling technology, the type of tasks that are affected go beyond coarse simulation, steady state calculation and bifurcation analysis. We have illustrated coarse control, and timesteppers also naturally fit in a coarse optimization framework, where optimal control problems (such as those arising in rare events) can be solved without ever having to derive explicit models in terms of coarse coordinates that meaningfully parametrize the transitions in question. The ability to find coarse self-similar solutions by combining coarse timesteppers with renormalization flow approaches for PDEs opens up the playing field even more. It would be interesting to see, as experience is gathered and developments are made, how much of the conceptual promise of these techniques becomes reality.

Acknowledgements. The work reported in this paper spans several years and involves interactions with several people, whom we would like to acknowledge. Original discussions with Herb Keller about RPM (at the IMA in Minneapolis) and with Jim Evans at Iowa State about his hybrid MC simulators of surface reactions played an important role. Discussions over the years with Profs. J. Keller, K. Lust, D. Maroudas, D. McLaughlin, A. Majda, L. Ju, S. Yip, R. Armstrong, A. Panagiotopoulos, M. Graham, R. Kapral, H.-C. Oettinger, V. Yakhot, C. Foias, E. Titi, P. Constantin, M. Aizenman, J. Burns, F. Alexander, N. Hadjiconstantinou, A. Makeev, C. Siettos, C. Pantelides, C. Jacobsen, and A. Armaou have affected this work. In the course of the last three years we have discussed this research direction with our colleagues, Professors Engquist and E at PACM in Princeton. In the companion paper [119] they present an alternative formulation for conservation laws which, in the time-dependent case, corresponds to a generalized Godunov scheme. The role of AFOSR (Dynamics and Control, Drs. Jacobs and King), to whom this work was proposed in 1999, and which they supported through the years, has been crucial. Partial support by NSF through a KDI and an ITR grant, by DARPA, by a Humboldt Forschungspreis to I.G.K, and Los Alamos National Lab (through LDRD-DR-2001501) are also gratefully acknowledged.

Note added in proof: The originally submitted (August 2002) version of this paper, containing an extensive qualitative discussion of the underpinnings of the approach can be found as physics/0209043 at arXiv.org. Between the original submission and this publication many additional developments have taken place; a current listing of “equation-free papers” can be found in <http://arnold.princeton.edu/~yannis/>. In particular, we would like to mention developments in the numerical analysis of the gap-tooth scheme, reverse coarse projective integration and some computational biology and computational materials science problems (physics/0312004, physics/0310014,

physics/0308040, cond-mat/0310473, nlin.AO/0310011, nlin.CG0307016, nlin.CD0302055 and more at arXiv.org).

REFERENCES

- [1] C. Theodoropoulos, Y.H. Qian, and I.G. Kevrekidis, Proc. Nat. Acad. Sci., 34:9840–9843, 2000.
- [2] P.L. Bhatnagar, E.P. Gross, and M. Krook, Phys. Rev., 94:511–525, 1954.
- [3] C.W. Gear and I.G. Kevrekidis, NEC Technical Report NECI-TR 2001-029 (March 2001); SIAM J. Sci. Comp., 24(4):1091–1106, 2003.
- [4] C.W. Gear, I.G. Kevrekidis, and C. Theodoropoulos, Comp. Chem. Eng., 26(7-8):941–963, 2002.
- [5] I.G. Kevrekidis, *Plenary Lecture, CAST Division of the AIChE*, 2000 AIChE annual meeting, Los Angeles, 2000, also www.princeton.edu.
- [6] Matlab, *The Language of Technical Computing*, the Math Works Inc. Natick, MA, 2000.
- [7] E.J. Doedel, Cong. Num., 30:265–284, 1981. E.J. Doedel, H.B. Keller, and J.P. Kernevez, Int. J. Bif. Chaos., 1:493–520, 1991.
- [8] A.G. Salinger, N.M. Bou-Rabee, E.A. Burroughs, R.B. Lehoucq, R.P. Pawlowski, L.A. Romero, and E.D. Wilkes, *Sandia National Labs Technical Report*, 2001, also R.B. Lehoucq and A.G. Salinger, Int. J Numer. Meth. Fluids., 36:309–327, 2001.
- [9] C.C. Pantelides, *Proceedings of CHEMPUTERS EUROPE III*, Frankfurt, 1996.
- [10] A. Bensoussan, J.L. Lions, and G. Papanicolaou, *Asymptotic Analysis of Periodic Structures*, North-Holland, 1978.
- [11] F.A. Bornemann, *Homogenization in time of singularly perturbed conservative mechanical systems*, Lecture Notes in Math., 1687, Springer, 1998.
- [12] N.D. Goldenfeld, *Lectures on Phase Transitions and the Renormalisation Group*, Addison-Wesley, 1992.
- [13] P. Constantin, C. Foias, B. Nicolaenko, and R. Temam, *Integral Manifolds and Inertial Manifolds for Dissipative Partial Differential Equations*, Springer Verlag, NY, 1988.
- [14] R. Balescu, *Equilibrium and Nonequilibrium Statistical Mechanics*, John Wiley, NY, 1975.
- [15] P. Gaspard, *Chaos, Scattering and Statistical Mechanics*, Cambridge University Press, NY, 1998.
- [16] S. Chapman and T.G. Cowling, *The Mathematical Theory of Nonuniform Gases*, Cambridge U. Press, 1964.
- [17] W.G. Vincenti and C.H. Kruger Jr, *Introduction to Physical Gas Dynamics*, John Wiley, NY, 1965.
- [18] J.P. Boon and S. Yip, *Molecular Hydrodynamics*, McGraw-Hill, NY, 1980.
- [19] N.G. Van Kampen, *Stochastic Processes in Physics and Chemistry*, North Holland, 1992.
- [20] H. Spohn, *Large Scale Dynamics of Interacting Particles*, Springer, 1991.
- [21] K. Xu and K. Prendergast, J. Comp. Phys., 114:9–17, 1994.
- [22] V. Bulatov, F.F. Abraham, L. Kubin, B. Devincere, and S. Yip, Nature, 391:669, 1998.
- [23] J. Li, D. Liao, and S. Yip, Phys. Rev., (E)57:7259–7267, 1998.
- [24] J. Li, D. Liao, and S. Yip, Mat. Res. Soc. Symp. Proc., 538:473, 1998.
- [25] A.L. Garcia, J.B. Bell, W.Y. Crutchfield, and B.J. Alder, J. Comp. Phys., 154:134–155, 1999.
- [26] F.F. Abraham, N. Bernstein, J.Q. Broughton, and D. Hess, Mat. Res. Soc. Bulletin, 25:27, 2000.
- [27] F.F. Abraham, J.Q. Broughton, N. Bernstein, and E. Kaxiras, Computers in Physics, 12:538, 1998.
- [28] K. Xu, J. Comp. Phys., 171:289–335, 2001.
- [29] G.S. Smith, E.B. Tadmor, N. Bernstein, and E. Kaxiras, Acta Mater., 49:4089, 2001.
- [30] W.A. Curtin and R.E. Miller, Modelling Simul. Mater. Sci. Eng., 11:R33, 2003.
- [31] R. Phillips, *Crystals, Defects and Microstructures*, Cambridge University Press, 2001.
- [32] V.B. Shenoy, R. Miller, E.B. Tadmor, D. Rodney, R. Phillips, and M. Ortiz, J. Mechanics and Phys. Solids, 47, 1999.
- [33] M. Ortiz and R. Phillips, Adv. Appl. Mech., 36:1, 1999.
- [34] H.C. Oettinger, *Stochastic Processes in Polymeric Fluids*, Springer, 1996.
- [35] M. Laso and H.C. Oettinger, JNNFM, 47:1, 1993.
- [36] H.C. Oettinger, B.H.A.A. van den Brule, and M.A. Hulsen, JNNFM, 70:255–261, 1997.
- [37] M.A. Hulsen, A.P.G. van Heel, and B.H.A.A. van den Brule, JNNFM, 70:79, 1997.
- [38] R.M. Jendrejack, J.J. de Pablo, and M.D. Graham, JNNFM, 108:123, 2003.
- [39] A. Chorin, A. Kast, and R. Kupferman, Proc. Natl. Acad. Sci., USA, 95:4094–4098, 1998.

- [40] A. Chorin, O. Hald, and R. Kupferman, Proc. Natl. Acad. Sci., USA, 97:2968–2973, 2000.
- [41] A. Chorin, O.H. Hald, and R. Kupferman, Physica, (D)166:239, 2002.
- [42] D. Givon, R. Kupferman, and A. Stuart, Warwick Preprint 11/2003, <http://www.maths.warwick.ac.uk/~stuart/extract.pdf>.
- [43] H. Mori, Progr. Theor. Phys., 33:423, 1965.
- [44] R. Zwanzig, J. Stat. Phys., 9:215, 1973.
- [45] A.J. Majda, I. Timofeyev, and E. Vanden-Eijnden, Comm. Pure Appl. Math., 54:891, 2001.
- [46] T. Shardlow and A.M. Stuart, SIAM J. Num. Anal., 37:1120–1137, 2000.
- [47] W. Huisinga, C. Schuette, and A.M. Stuart, Comm. Pure Appl. Math., 56, 2003.
- [48] A.N. Gorban, I.V. Karlin, H.C. Ottinger, and L.L. Tatarinova, Phys. Rev., (E)63:066124, 2001.
- [49] A.N. Gorban, I.V. Karlin, P. Ilg, and H.C. Oettinger, JNNFM, 96:203–219, 2001.
- [50] A.N. Gorban and I.V. Karlin, Phys. Rev. (E)65:026116, 2002.
- [51] A.N. Gorban and I.V. Karlin, 2003, <http://arxiv.org/abs/cond-mat/0308331>.
- [52] P. Ehrenfest and T. Ehrenfest, in *Encyklopaedie der Mathematischen Wissenschaften*, reprinted in P. Ehrenfest, 1959, Collected Scientific Papers, North Holland, 1911.
- [53] G.E.P. Box, W.G. Hunter, and J.S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*, John Wiley, 1978.
- [54] R.E. Kalman and R.S. Bucy, Trans. ASME Ser. D, J. Basic Eng., 83:95–107, 1961.
- [55] K.J. Astrom, *Introduction to Stochastic Control Theory*, Academic Press, NY, 1970.
- [56] L. Ljung, *System identification: theory for the user*, 2nd ed., Prentice Hall, NY, 1999.
- [57] O. Runborg, C. Theodoropoulos, and I.G. Kevrekidis, Nonlinearity, 15:491–511, 2002.
- [58] A.G. Makeev, D. Maroudas, and I.G. Kevrekidis, J. Chem. Phys., 116(23):10083–10091, 2002.
- [59] J. Li, D. Liao, and S. Yip, J. Comp. Aided Mat. Des., 6:95–102, 1999.
- [60] C.W. Gear, NEC Technical Report NEC-TR 2001-130, 2001.
- [61] J.P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen, J. Comput. Phys., 23:327, 1977.
- [62] P.N. Brown, A.C. Hindmarsh, and L.R. Petzold, SIAM J. Sci. Comp., 19:1495, 1998.
- [63] C. Theodoropoulos, S. Sankaranarayanan, S. Sundaresan, and I.G. Kevrekidis, *Proc. of the 3rd Pan-Hellenic Conference in Chemical Engineering*, 221–224, 2001.
- [64] C. Theodoropoulos, S. Sankaranarayanan, S. Sundaresan, and I.G. Kevrekidis, Chem. Eng. Sci., 2003, in press, also nlin.PS/0111040 at arXiv.org
- [65] A.G. Makeev, D. Maroudas, A.Z. Panagiotopoulos, and I.G. Kevrekidis, J. Chem. Phys., 117(18):8229–8240, 2002.
- [66] FitzHugh and R. Biophys. J., 1:445–466, 1961.
- [67] J.S. Nagumo, S. Arimoto, and S. Yoshizawa, Proc., IREE, 50:2061–2070, 1962.
- [68] Y.H. Qian and S.A. Orszag, J. Stat. Phys., 81:237–253, 1995.
- [69] G.M. Shroff and H.B. Keller, SIAM J. Numer. Anal., 30:1099–1120, 1993.
- [70] H.B. Keller, *Dynamical Systems Program*, Lecture presented at the IMA, Minneapolis, 1997, also, personal communication IGK.
- [71] H. Jarausch and W. Mackens, Numer. Math., 50:633–653, 1987.
- [72] H. Jarausch and W. Mackens, in *Large Scale Scientific Computing*, P. Deuffhard, B. Engquist, Progress in Scientific computing, 7 Birkhäuser Verlag, 1987.
- [73] L.S. Tuckerman and D. Barkley, in *IMA Volumes in Mathematics and its Applications*, 119:453–466, 1999.
- [74] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM Publications, Philadelphia, 1995.
- [75] R. Temam, *Infinite Dimensional Dynamical Systems in Mechanics and Physics*, Springer Verlag, NY, 1988.
- [76] M.S. Jolly, I.G. Kevrekidis, and E.S. Titi, Physica (D)44:36, 1991.
- [77] P. Constantin, I.G. Kevrekidis, and E.S. Titi, Disc. Cont. Dyn. Sys., 2003, in press, also <http://people.cs.uchicago.edu/~const/smof.pdf>.
- [78] H.von Sosen, *The Recursive Projection Method Applied to Differential Algebraic Equations and Incompressible Fluid Mechanics*, Ph.D. Thesis, Part II, Caltech, Pasadena, 1994.
- [79] K. Lust, D. Roose, A. Spence, and A.R. Champneys, SIAM J. Sci. Comput., 19:1188–1209, 1997.
- [80] K. Lust, *Numerical Bifurcation Analysis of Periodic Solutions of Partial Differential Equations*, Ph.D. Thesis, Leuven, 1997.
- [81] K. Burrage, J. Erhel, B. Pohl, and A. Williams, SIAM J. Sci. Comput., 19:1245–1260, 1998.
- [82] J. Erhel, K. Burrage, and B. Pohl, J. Comp. Appl. Math., 69:303, 1996.
- [83] B.D. Davidson, SIAM J. Num. Anal., 34:2008–2027, 1997.
- [84] P. Love, *Bifurcations in Kolmogorov and Taylor Vortex Flows*, Ph.D. Thesis, Caltech, Pasadena, 1999.

- [85] E.D. Koronaki, A.G. Boudouvis, and I.G. Kevrekidis, *Comp. Chem. Eng.*, 27:951–964, 2003.
- [86] C. Siettos, M.D. Graham, and I.G. Kevrekidis, *J. Chem. Phys.*, 118:10149–10157, 2003.
- [87] C.W. Gear and I.G. Kevrekidis, NECI Research Report NEC-TR 2001-122, October 2001; *J. Comp. Phys.*, 187:95–109, 2003.
- [88] V.I. Lebedev, in *Numerical Analysis and its Applications*, Proc. WNAA 96, Bulgaria, Vulkov et al. eds., Springer, Berlin, 274–283, 1997.
- [89] A.L. Medovikov, *BIT*, 38(2):372–390, 1998.
- [90] J.G. Verwer, *Appl. Num. Math.*, 22:359–379, 1996.
- [91] M. Hochbruck, C. Lubich, and H. Selhofer, *SIAM J. Sci. Comp.*, 19:1552–1574, 1998.
- [92] W.S. Edwards, L.S. Tuckerman, R.A. Friesner, and D.C. Sorensen, *J. Comp. Phys.*, 110:82–102, 1994.
- [93] E. Gallopoulos and Y. Saad, *SIAM J. Sci. Comp.*, 13:1236–1264, 1992.
- [94] S.H. Lam and D.A. Goussis, *Int. J. Chem. Kinetics*, 26:461–486, 1994.
- [95] R. Rico-Martinez, C.W. Gear, and I.G. Kevrekidis, *J. Comp. Phys.*, 2003, in press, also nlin.CG/0307016 at arXiv.org
- [96] P.J. Holmes, J.L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 1996.
- [97] L. Sirovich and J.D. Rodriguez, *Phys. Lett.*, A(120):211, 1989.
- [98] A.E. Deane, I.G. Kevrekidis, G.M. Karniadakis, and S. Orszag, *Phys. Fluids*, 3:2337, 1991.
- [99] N. Aubry, P.J. Holmes, J.L. Lumley, and E. Stone, *J. Fluid Mech.*, 192:115, 1988.
- [100] S. Shvartsman and I.G. Kevrekidis, *AIChE J.*, 44:1579–1595, 1998.
- [101] S.Y. Shvartsman, C. Theodoropoulos, I.G. Kevrekidis, R. Rico-Martínez, E.S. Titi, and T.J. Mountziaris, *J. Proc. Control*, 10:177–184, 2000.
- [102] S. Yesilyurt and A.T. Patera, *CMAME*, 121:231–257, 1995.
- [103] C.I. Siettos, C.C. Pantelides, and I.G. Kevrekidis, presented at the AIChE annual meeting, Reno, 2001, also *Ind. Eng. Chem.*, 2003, in press, also math.OC/0302040 at arXiv.org.
- [104] J.K.C. Suen, Y.L. Joo, and R.C. Armstrong, *Annu. Rev. Fluid Mech.*, 34, 2002.
- [105] M. Grosso, L. Russo, P.L. Maffettone, and S. Crescitelli, in *Proc. of COC 2000*, St., Petersburg, IEEE Publications, 561–564, 2000.
- [106] G.S. Heffelfinger and F.van Swol, *J. Chem. Phys.*, 100(10):7548–7552, 1994.
- [107] G.S. Heffelfinger and D.M. Ford, *Molecular Physics*, 94(4):659–671, 1998.
- [108] G.S. Heffelfinger and D.M. Ford, *Molecular Physics*, 94(4):673–683, 1998.
- [109] A.P. Thompson, D.M. Ford, and G.S. Heffelfinger, *J. Chem. Phys.*, 109(15):6406–6414, 1998.
- [110] A.J. Roberts, *Appl. Numer. Math.*, 37:371–396, 2001.
- [111] S.M. Allen and J.W. Cahn, *Acta Metallurgica*, 27:1085, 1979.
- [112] C.de Boor, *A Practical Guide to Splines*, revised ed., Springer, NY, 2001.
- [113] M. Berger and J. Olinger, *J. Comp. Phys.*, 53:484, 1984.
- [114] S.T. O’Connell and P.A. Thompson, *Phys. Rev.*, E(52)6:R5792–95, 1995.
- [115] N.G. Hadjiconstantinou and A.T. Patera, *Int. J. Mod. Phys.*, C(8)4:967–976, 1997.
- [116] N.G. Hadjiconstantinou, *J. Comp. Phys.*, 154:245–265, 1999.
- [117] W. E and Z. Huang, *Phys. Rev. Lett.*, 87:135501, 2001.
- [118] J. Li, P.G. Kevrekidis, C.W. Gear, and I.G. Kevrekidis, *SIAM MMS*, 1:391–407, 2003.
- [119] W. E and B. Engquist, *Comm. Math. Sci.*, 1(1):87–132, 2003.
- [120] C.W. Gear, J. Li, and I.G. Kevrekidis, *Physics Letters A*, 316:190–195, 2003.
- [121] K. Sankaranarayanan, X. Shan, I.G. Kevrekidis, and S. Sundaresan, *J. Fluid Mech.*, 452:61–96, 2002.
- [122] K. Sankaranarayanan, I.G. Kevrekidis, S. Sundaresan, J. Lu, and G. Tryggvason, *Int. J. Multiphase Flow*, 29:109, 2003.
- [123] C.W. Gear and I.G. Kevrekidis, *Phys. Letters A*, 2003, submitted, also nlin.CD/0302055 at arXiv.org.
- [124] C.I. Siettos, A. Armaou, A.G. Makeev, and I.G. Kevrekidis, *AIChE J.*, 49:1922, 2003; also nlin.CG/0207017 (arXiv.org).
- [125] C.I. Siettos, D. Maroudas, and I.G. Kevrekidis, *Int. J. Bifurcations and Chaos*, 2003, in press, also physics/0212034 in arXiv.org
- [126] G. Hummer and I.G. Kevrekidis, *J. Chem. Phys.*, 118:(23)10762–10773, 2003.
- [127] C.I. Siettos, C. Pantelides, and I.G. Kevrekidis, *Ind. Eng. Chem.*, 2003, in press, also math.OC/03020404 at arXiv.org
- [128] J. Moeller, O. Runborg, P.G. Kevrekidis, K. Lust, and I.G. Kevrekidis, *Nonlinearity*, 2003, submitted, also physics/0307153 at arXiv.org
- [129] D.G. Aronson, S. Betelu, and I.G. Kevrekidis, *PNAS*, 2003, submitted, also manuscript nlin.AO/0111055 at arXiv.org.

- [130] C.W. Rowley, I.G. Kevrekidis, J.E. Marsden, and K. Lust, *Nonlinearity*, 16(4):1257–1275, 2003.
- [131] C.W. Rowley and J.E. Marsden, *Physica (D)*142:1–19, 2000.
- [132] L.Y. Chen and N. Goldenfeld, *Phys. Rev.*, E(51)6:5577–5581, 1995.
- [133] C.W. Gear, S. Betelu, D.G. Aronson, and I.G. Kevrekidis, manuscript in preparation, 2003.
- [134] M.P. Brenner, P. Constantin, L.P. Kadanoff, A. Schenkel, and S. Venkataramani, *Nonlinearity*, 12(4):1071–1098, 1999.
- [135] C.I. Siettos, I.G. Kevrekidis, and P.G. Kevrekidis, *Nonlinearity*, 16(2)497–506, 2003.
- [136] K.N. Christodoulou and L.E. Scriven, *Quart. Appl. Math.*, 9:17, 1998.
- [137] R. Rico-Martinez, I.G. Kevrekidis and K. Krischer, in *Neural Networks in Chemical Engineering*, A. B. Bulsari ed., Elsevier, 409–442, 1995.
- [138] R. Rico-Martinez, J.S. Anderson, and I.G. Kevrekidis, in *Proc. IEEE Workshop, Neural Networks for Signal Processing*, IEEE Publ., vol. III, 596–605, 1994.
- [139] J.C. Spall, in *Johns Hopkins APL Technical Digest*, 19, 1998, also *IEEE Trans. Aut. Control*, 45:1839–1853, 2000.
- [140] F.A. Escobedo, *J. Chem. Phys.*, 110:11999, 1999.
- [141] D.A. Kofke, *Molec. Phys.*, 78(6):1331–1336, 1993.
- [142] L.I. Kioupis and E.J. Maginn, *Fluid Phase Equilibria*, 200:75–92, 2002.
- [143] L.I. Kioupis, G. Arya, and E.J. Maginn, *Fluid Phase Equilibria*, 200:73–110, 2002.
- [144] C. Jarzynski, *Phys. Rev. Lett.*, 78(14):2690–2693, 1997.
- [145] C. Bardos, F. Golse, and N.J. Mauser, *Methods Appl. Anal.*, 7:275–293, 2000.
- [146] B. Engquist and O. Runborg, *J. Comput. Appl. Math.*, 74:175–192, 1996.
- [147] O. Runborg, *M2AN Math. Model. Numer. Anal.*, 34:1203–1231, 2000.
- [148] S. Jin and X. Li, *Multi-phase computations of the semiclassical limit of the Schrödinger equation and related problems: Whitham vs. Wigner*, *Physica D*, 182:46–85, 2003.
- [149] R.A. Brown, L.E. Scriven, and W.J. Silliman, in *New Approaches to Nonlinear Problems in Dynamics*, P.J. Holmes ed., SIAM, Philadelphia, 1980.
- [150] L.E. Scriven, *J. W. Gibbs Lecture presented to the AMS*, New Orleans Meeting, 1986.