Coarse-Grained Equation-Free Multiscale Computation: enabling microscopic simulators to perform system-level tasks

Ioannis G. Kevrekidis[1,2], *
C. William Gear[1,3], James M. Hyman[4], Panagiotis G. Kevrekidis[5],
Olof Runborg[2†], and Constantinos Theodoropoulos[6]

[1]Department of Chemical Engineering;
[2]Program in Applied and Computational Mathematics
Princeton University, Princeton, NJ 08544; [3]NEC Research Institute;
[4]CNLS and T-7, Los Alamos National Laboratory Los Alamos, NM 87545;
[5]Department of Mathematics, University of Massachusetts, Amherst, MA
[6]Department of Process Integration, UMIST, Manchester M60 1QD, UK.

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 systems identification based, "closure on demand" computational toolkit, bridging microscopic/stochastic simulation with traditional continuum scientific computation and numerical analysis. We illustrate these "numerical enabling technology"

* *to whom correspondence should be addressed, yannis@princeton.edu*
† Current address: Numerical Analysis and Computer Science, KTH, 100 44 Stockholm, Sweden.

ideas through examples from chemical kinetics (LB, KMC), rheology (Brownian Dynamics), homogenization and the computation of "coarsely self-similar" solutions, and discuss various features, limitations and potential extensions of the approach.

Contents

1 INTRODUCTION

The purpose of this work is to analyze a synergism that exists between "conventional" numerical methods/analysis and microscopic complex systems modeling, dynamics and control. Based on this synergism, 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 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 Hessians, 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 or EFM Methods). The purpose of this "mathematics assisted" computational technology 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" -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 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 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 super-structures (libraries) that can be wrapped around whatever the best computer model a scientist would come up with for her/his system - be it a Monte Carlo description of a chemical reaction or a kinetic theory based model of multiphase flow. 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.)

We discuss a systematic framework for extracting directly from microscopic simulations the information one would obtain from macroscopic models, had these macroscopic models been available in closed form. In a sense, the approach consists of "unavailable macroscopic model-motivated" processing of microscopic dynamical computations—it is a *system identification* based procedure, processing the results of short bursts of (appropriately initialized) microscopic simulation, that enables us to create an "equation-free" computational framework.

If what we did was to run a full (completely resolved in space and time) microscopic simulation, and merely extract 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.). These points are briefly summarized in Table 1, where "small space" and "short time" refer to the microscopic simulation, while "large space" and "long time" refer to the macroscopic tasks.

EFM Methods & Scope

Method	Actual		Enabled
Legacy timestepper + RPM Shroff and Keller, 1993 [37]	repeated (T)	$\longrightarrow$	T_∞
Coarse timestepper + RPM Theodoropoulos et al., 2000 [1] (coarse bifurcation)	repeated (τ)	$\longrightarrow$	T_∞
Coarse integration Gear and Kevrekidis, 2001, 2002[3, 4]	$\tau\tau\tau\ldots$	$\longrightarrow$	T
Gap-tooth scheme Kevrekidis, AIChE 2000 [5]	(ss...s)τ	$\longrightarrow$	Sτ
gap-tooth + coarse integration $\equiv$ patch dynamics	(ss...s)$\tau\tau\tau\ldots$	$\longrightarrow$	ST
gap-tooth + RPM $\equiv$ patch bifurcations	repeated ((ss...s)$\tau\tau\tau\ldots$)	$\longrightarrow$	ST_∞

Table 1: EFMs can be catalogued by their spatial—s(S) small(large)—and their temporal—τ(T,T_∞) short (intermediate,"infinite") computational scales. The method named on the left uses "actual" space-time computations to obtain "enabled" space-time results.

The microscopic simulation should use the best current microscopic description of the physics (and the code implementing it) available; 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)

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 through post-processing ("patching together") the results of appropriately initialized and executed bursts of microscopic simulation over small spatial domains and short time intervals. We will then present an anthology of results from ongoing investigations that illustrate the scope of the EFM methods to analyze 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 areas in which we hope the methods may be fruitfully applied.

Before we start, it is worth making certain disclaimers. Many of the mathematical and computational tools combined here (for example, system identification, or inertial manifold theory) are well established; we only borrow them as necessary. We hope that the *synthesis* we propose here has new tools and insights to offer. Its main point, the "on-line" bridging of scales through "on demand" closure is, of course, a mainstream avenue in current research, where innovative multiscale techniques (such as the quasi-continuum method of Philips and coworkers [10, 11], the kinetic-theory based solvers proposed by Xu and Prendergast [12, 13], the optimal prediction methods of Chorin [15, 16] and the novel stochastic integration techniques of Shardlow and Stuart [14]) are constantly being proposed and explored.

2 BACKGROUND

We are interested in the coarse dynamics of large complex systems. In this section we assume that only solvers describing the *microscopic* behaviour 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 systems that are described without reference to any specific spatial structure.

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

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

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 (1). We always assume that (1) is too expensive to solve for long times or that we do not have any precise knowledge of f; typically (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 (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 are microscopic systems that cannot be written on the form (1), which can still be treated by the general methodolgy described here, e.g. many stochastic systems solved via Monte-Carlo simulations.

The first thing we need to do in order to perform macroscopic level simulations based on (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 our 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 name 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 variables are functions - which have a higher dimension than any

finite-dimensional particle model. In that case, the computational model will use a low-dimension 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 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 Ju *et al.* [19, 20, 21], and the cumulative probability distributions in [22], are instances of this restriction operator.

The important features of the macroscopic variable set, U, are that it is sufficient to represent the closure of (1) on a slow manifold and that on that manifold all components of U are slowly varying. For example, suppose that there is a change of variables $u \mapsto v$ in (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. Suppose that the compatible partitiion of g gives $(g_1, g_2/\varepsilon)$ with $\varepsilon \ll 1$. Then we have

$$\begin{aligned} \frac{dv_1}{dt} &= g_1(v_1, v_2) \\ \frac{dv_2}{dt} &= \frac{1}{\varepsilon} g_2(v_1, v_2) \end{aligned} \tag{2}$$

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 $g_2(v_1, g(v_1)) = 0$. Identifying v_1 with the macroscopic variable U reduces (2) to the coarse equation

$$\frac{dU}{dt} = \frac{dv_1}{dt} = g_1(v_1, r(v_1)) = F(U). \tag{3}$$

Note that the functions g, r, and F are not available and neither (2) nor (3) can be used as a basis for direct numerical simulation.

Our problem can now be stated thus:

> Problem A: We want to simulate (3) for large, or infinite, times, but we are only able to simulate (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 [17, 18, 4]). The coarse timestepper, denoted $\mathcal{T}_c^\tau$, implements an approximation of the time-τ map for (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 comensurate with the macroscopic dynamics.

The coarse time stepper relies on the detailed timestepper, $\mathcal{T}_d$ for (1), the detailed microscopic/stochastic evolution of the system, as well as an appropriate, nonunique, *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. 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 $\mathcal{T}_c^\tau U(t+\tau) := \mathcal{M}u(\tau,t)$.

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). For $t \gg \varepsilon$, and the assumption that v_2 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 (3), and $U(t) \approx \mathcal{M}u(t)$. 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 (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. Suppose (2) has the form

$$\begin{aligned} \frac{du}{dt} &= -u - v + 2, & u(0) &= u_0, \\ \frac{dv}{dt} &= \frac{1}{\varepsilon}(u^3 - v), & v(0) &= v_0, \end{aligned} \tag{4}$$

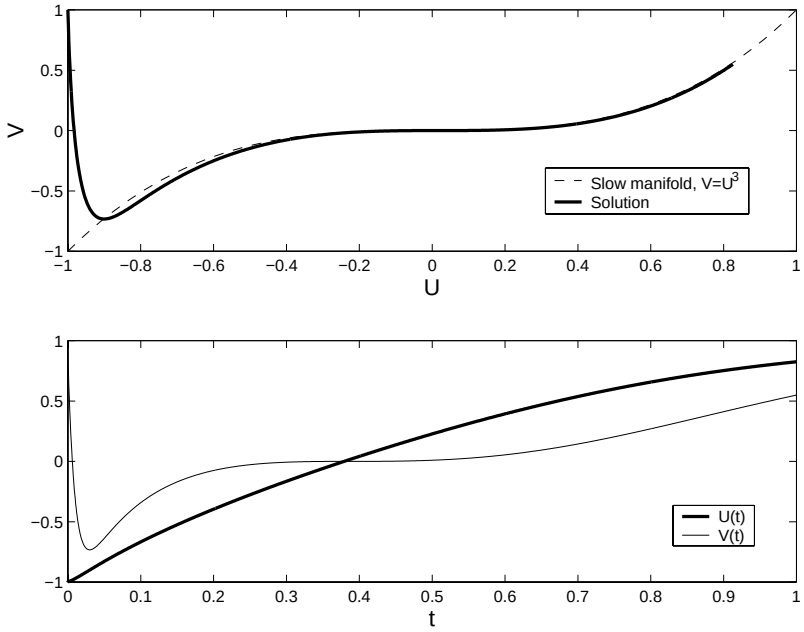


Figure 1: Exact solution of (4) with $\varepsilon = 0.01$, $u_0 = -1$ and $v_0 = 1$. Top figure shows solution plotted in the (u, v) phase space. Bottom figure shows solution as a function of time.

where for simplicity both u and v are scalars. The solution of (4) rapidly moves to the slow manifold $v = u^3$ for any initial data when $\varepsilon \ll 1$. An example of an exact solution of this system is shown in Fig. 1, and the corresponding coarse time stepper solution is given in Fig. 2.

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

> Problem B: We want to simulate (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 (1) can now be viewed as a "black box" solver for (3), much in the same way as a legacy code for (3).

2.1.1 Discussion about the Coarse Time Stepper

The coarse time-stepper procedure is a natural one, and it is easy to realize numerically. For problems in which several microscopic copies of the same coarse initial conditions must be evolved, it is also easy to parallelize: each "lifted" copy of the same coarse initial condition runs independently on a different processor. (As we will see below, an additional level of parallelism enters when we decimate a large computational domain to "patches", each of which can also be run on a different processor.) Such procedures have been employed for the construction of averaged equations from coarse-graining the master equation (e.g. [23, 24, 25]) and, similarly, for the derivation of hydrodynamic descriptions of interacting particle systems (e.g. [26, 27, 28, 29, 30]).

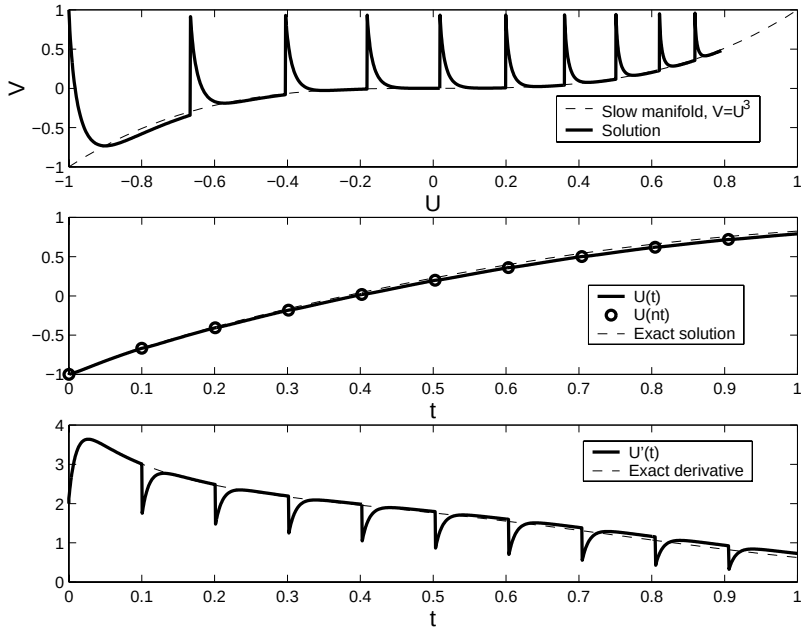


Figure 2: Results with coarse time stepper applied to (4) using $\tau = 10\varepsilon = 0.1$ and $u_0 = -1$. The lifting operator μ is chosen such that the initial condition for v is $v_0 = 1$. Top figure shows solution plotted in the (u, v) phase space. At times $t = n\tau$, the solution is restricted and lifted, which here amounts to setting $v = v_0 = 1$. Middle figure shows solution as a function of time. Bottom figure shows derivative of solution as a function of time. The coarse time stepper solution agrees better with the exact solution when ε is small or τ is large.

In this work we provide, through several illustrations, computational evidence for this idea, suggesting a program for the proof of existence of a semigroup for such a procedure (and for its "gap-tooth" and "patch dynamics" extensions that will be presented in later Sections).

We have discussed the one-many nature of the *lifting* component of the coarse time stepper in much more detail in [18, 4, 33, 55]. *In principle*, if there is enough of a separation of time scales, it should not matter what we initialize the higher moments of the distributions to; we expect that, if a coarse deterministic equation for the low moments exists and closes, the high moments will very quickly relax to functionals of the lower moments. We refer to this process as the "healing" of the higher moments, and it basically produces "mature" distributions (as opposed to "fresh" distributions that have the same low moments but "inaccurate" or "improbable" higher moments): cf. the spikes in Fig. 2. This is, of course, a singularly perturbed problem (the high moments quickly evolve to an attracting manifold parametrized by the low moments), and the problem of consistent initialization has been systematically studied in this context. Consistent initialization of DAEs is, essentially, the same problem [54]; initializing on slow manifolds in meteorology (what we call "mature" initial conditions

are referred to as "bred" initial conditions there, e.g. see [56, 57]) is also in essence the same problem, as is initializing as close to an inertial manifold as possible. Of course, after a short integration time the dynamics themselves will "bring us down" to the manifold; but the random initialization of the high moments, even though quickly healed, is a source of noise for the deterministic part of the computation, and reducing this noise is helpful in the "outer" computational tasks. The "strong" and "weak" cone conditions in the theory of inertial manifolds [43] provide a framework for the rate at which trajectories "off" the manifold approach it, and several playful analogies to airplanes and landing strips can be made. This issue has important practical implications for the effectiveness of the computations, and it has been studied in the statistical mechanical literature for a long time [58] (see also [59]).

Averaging over an ensemble of consistent (i.e. similarly conditioned) initial distributions is, of course, also a means to variance reduction in MC calculations. In the systems and control area a vast literature exists on optimal estimation and model identification (e.g. [60, 61, 62]), and good filtering is vital when one tries to estimate the numerical derivatives (or the approximations to the matrix-vector products) necessary to enable Newton/fixed point subspace iteration based computations. Our standard illustrative problem (the LB-FHN model) is extremely forgiving in terms of variance reduction and consistent initialization; this is illustrated in Fig. 15. A single simulation copy (a single realization) conditioned on the zeroth moments, at various levels of coarseness, initialized at local equilibrium or away from local equilibrium for the high moments, does "the same thing" as far as the RPM code is concerned at the level of accuracy in our computations.

Other problems (like the nematic liquid crystal closure [55] as well as the surface kinetic Monte Carlo examples [18, 33] we briefly discuss below) are much more sensitive to "mature" initial conditions. Indeed, care has to be taken (i.e. a separate Metropolis Monte Carlo code written) to "equilibrate" the consistent initialization [33]; as we discuss in [18] and demonstrate in [55], it may become necessary to include additional moments (which now have gradually become "slow") as independent coarse variables. For surface lattice gases, for example, we may lift with densities *and* pair probabilities as independent variables, and include a separate short MC step that imposes desired pair probabilities for every initialization ("lifting").

2.2 Coarse Projective Integration

In this section we discuss the first part of Problem B above: How to integrate (3) for large (macroscopic) times $\Delta T \gg \tau$, where τ is the time horizon in the coarse time stepper $\mathcal{T}_c^\tau$. In the terminology of Table 1, we show that coarse projective integration enable the "short time" coarse timestepper to perform "large time" tasks.

Let $\Delta T \gg \tau$ be a large time step and $\delta t \leq \tau$ be a small time step. Set $t_k = k\delta t$. The small δt could for instance be the numerical time step used when solving the detailed equation (1). Introduce the numerical approximations of 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

$$u^{k,N} = \mathcal{T}_d^{t_k} u^{0,N} \tag{5}$$

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

$$U^{k,N} := \mathcal{M} u^{k,N} \approx u(N\Delta T + t_k). \tag{6}$$

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

$$U^{N+1} = U^{k,N} + (\Delta T - t_k)\tilde{F}(U^{k,N}), \tag{7}$$

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

$$\tilde{F}(U^{k,N}) := \frac{U^{k+1,N} - U^{k,N}}{\delta t} \approx \frac{dU(N\Delta T + t_k)}{dt} = F(U(N\Delta T + t_k)) \approx F\left(U^{k,N}\right). \tag{8}$$

The method in (7) and (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, 63]; while implicit versions of these algorithms are possible, and are partially discussed there, the heart of the process is an *explicit* outer integrator, here (7), built around calls to an "inner" integrator, here (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 $\mathcal{M}$ to u; 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 (6) to neutralize their growth in the large, projective, extrapolation step in (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 [64, 65, 66]; 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 [63] for details). It would also be interesting to compare our projective integrators with Krylov/exponential integrators for large systems with separations of time scales [67, 68, 69, 70].

Higher order versions of (7) can be constructed based on polynomial extrapolation [4]. Let $P(t)$ be the q-th order polynomial interpolating the points

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

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 versions of (7) can be used.

The general procedure is illustrated in Fig. 16: 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. 17, taken from [4] shows time-series as well as long-term attractors of such an "inner LB, outer FEM" one stage projective forward Euler integrator; because of the nature of the outer description, we call these two-tier codes "micro-Galerkin methods". Using different sets of basis functions for the outer integrator is discussed in [4]. In particular, we have used 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 [71, 72]. 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. [73, 74, 75, 76, 77].

2.2.1 Discussion about Coarse Projective Integration

The smooth macroscopic fields can be estimated by low-dimensional projections of Cumulative Distribution Functions (CDFs) on appropriate basis functions [22]; alternative maximum-likelihood based methods for estimating smooth macroscopic fields from distributions include the "thermodynamic field estimator" [20]. This estimated time derivative is then used by the outer integrator to perform a *projection step* in density space—a step much longer than the time-step in the molecular code. What carries the day is the separation of time scales in the space of moments of the distribution (in that, for example, higher moments get very quickly slaved to density, the zeroth moment, here). Of course, even *within* the space of the evolution for the zeroth moment (density profiles) slaving and smoothness is important, and another level of

separation of time scales ensues (center manifolds and possibly inertial manifolds for the density PDE). But the important thing [3, 4, 63] is that the detailed, microscopic dynamics themselves "kill" (damp) the coarse strong stable modes enough during the initial, maturing stages of the integration. Eventually, for *coarse* problems with gaps in their spectrum, the only remaining limitation is the time scales of the coarse slow modes, and the "outer" integration steps can be taken (with care) commensurate to these.

The projective step can be run forwards or (interestingly) backwards in time. A reason for running it backwards would be to obtain an estimate of an earlier point *on* the "slow" or inertial manifold –when the integration was started off that manifold. If, for example, we used k steps, each of length h, of an inner integrator—where k was large enough for the fast components to significantly damp and approach the manifold—took one more step, and then took a projective step of length kh backwards in time using the chord from the last two integration steps, we would have an estimate of the initial values *on* the inertial manifold. Simple analysis (at least, in the linear case) shows that if the inner integrator is approximately correct for the fast component—i.e., it approximates $\exp(h\lambda)$—then the initial deviations from the inertial manifold are damped by a factor proportional to $kh\lambda\exp(kh\lambda)$. For negative λ this factor can be made arbitrarily small by increasing k.

This "backwards estimate" can also be used to estimate the *time derivatives* on the manifold at the "start". In this case, we must do at least 2 additional inner steps, and then interpolate a quadratic or higher polynomial through the last 3 (or more) points. Estimating the time derivative at the "start" by computing the derivative of this polynomial can be similarly shown to reduce the impact of any off-manifold discrepancies in the initial conditions by a comparable factor. An additional benefit of this "backwards estimate" is that one can now construct a subroutine which, given a coarse initial condition, returns an estimate of the coarse time-derivative *at this condition*. This would allow for the easier incorporation of the equation-free approach, based on the coarse time-stepper, into existing large-scale, equation based, scientific computing integration packages.

Note that the coarse projective integration procedure can be "telescoped" recursively over several tiers of description: not just densities to molecules, but (not too seriously) densities to molecules to wavefunctions, with the microscopic integration always performed at the inner level. One then has an hierarchical structure, where the inner timestepper does the real work, the "first outer" time-stepper "projects" inner time-stepper results, a "second-outer" timestepper projects "first outer" timestepper results and so forth. The key is to do enough inner integration (that is where the damping takes place) so that the iterative outer projections remain stable and accurate; we have some preliminary experience with these telescoping projective integrators, and we are currently studying them further [63]. It may be interesting (although not immediately useful) to remark that the stability regions of these telescoping integrators have asymptotically (at the infinite iteration limit) fractal boundaries.

2.3 Coarse Bifurcation Analysis

In this section we discuss the second part of Problem B above: How to integrate (3) for infinite times. More specifically, we assume that the coarse equation (3) also includes a real parameter λ,

$$\frac{dU}{dt} = F(U;\lambda), \qquad U \in \mathbb{R}^M, \tag{9}$$

and we are interested in the qualitative behaviour for large times, and its dependence on λ: the stability of steady states of (9) and bifurcation diagrams with respect to λ. Note that a macroscopic, coarse, steady state can exist even though there are no microscopic steady states. We recall that $F(U;\lambda)$ is not known explicitly. We only have a timestepper $\mathcal{T}_c^\tau$ 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 "inifinite time" tasks: see Table 1.

Let us assume that (9) has a steady state, which we denote by $U^*(\lambda)$. At the steady state, $dU/dt = 0$ and, then U^* is obviously also a fixed point of $\mathcal{T}_c^\tau$,

$$U^* = \mathcal{T}_c^\tau U^*, \tag{10}$$

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

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

The question that we are faced with is hence how to solve the fixed-point problem (10). Viewing the problem like this is the basis for time stepper based bifurcation analysis from which our original inspiration comes [37, 38, 39, 40, 41].

One naive way to solve (10) is to use fixed point iteration (or "successive substitution"), starting with some initial data U_0 and for $n \geq 0$ setting

$$U^{n+1} = \mathcal{T}_c^\tau U^n. \tag{11}$$

This corresponds to *direct simulation* of (9), using the coarse time stepper to evolve an initial profile in time. Since we can only realize $\mathcal{T}_c^\tau$ 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 (10) in this way.

Alternatively, we could use a Newton type iteration to solve (10), e.g.

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

This gives fast (quadratic) convergence, also when U^* is an unstable solution. However, the method needs the Jacobian of $\mathcal{T}_c^\tau$, 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, is Shroff and Keller's Recursive Projection Method (RPM) [37]. It gives fast convergence (between linear and quadratic) and converges also when U^* is slightly unstable. The Jacobian of $\mathcal{T}_c^\tau$ never needs to be evaluated directly. Moreover, as a by-product, RPM computes approximations of the largest eigenvalues of the Jacobian, which give stability information of 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

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

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 $\mathbb{P}$,

$$\begin{aligned} p^{n+1} &= p^n - \left[J^*(p^n + q^n) - I\right]^{-1} \left[P\mathcal{T}_c^\tau(p^n + q^n) - p^n\right], \\ q^{n+1} &= Q\mathcal{T}_c^\tau(p^n + q^n), \end{aligned}$$

where $J^* = P(\partial \mathcal{T}_c^\tau(U)/\partial U)P \in \mathbb{R}^{d\times d}$ is a "small" 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 [37, 46, 47, 48, 49, 50, 51, 52, 53].

Standard time stepper based bifurcation analysis using RPM on top of a PDE time stepper, is represented by the (b) branch in Fig. 14. Substituting the coarse time tepper, described by branch (a), for the PDE time stepper in 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, 17, 18, 31, 32, 33]. 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 work is that the system has a clear separation of time scales: a few eigenvalues of the Jacobian of F in (9) lies 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 $\mathcal{T}_c^\tau$, 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 [42, 43, 44], and many singularly perturbed systems that arise in engineering modeling. In addition, as we will qualitatively discuss below, even if the problem 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 timestepper for that level of (coarse, averaged) description of the statistics of the problem [45].

2.3.1 Discussion about Coarse Bifurcation Analysis

Fixed point iteration (11), or direct simulation of (9) amounts to doing on the computer what one would observe in an experiment for the problem: setting parameter values, setting initial conditions and evolving forward in time. Doing on the computer what "nature does" in a laboratory makes sense, but may not be the best or fastest way of discovering features like steady states or bifurcation points.

The Newton method is an example of an algorithm that use the same right hand side, $f(u;\lambda)$ of (9), as in direct integration, but in different ways. Algorithms based on contraction mappings or augmented systems for the detection of bifurcation points, are routine for systems where the right hand side (RHS) is available [7, 8]. Indeed, techniques like Newton-Raphson, or multiparameter continuation codes inspired by numerical bifurcation theory, can be thought of as "alternative simulation ensembles": they use the same system equations (the same RHS formula) but in different ways, that *do not* correspond to evolution in physical time. The detailed dynamics of these algorithms (e.g. the dynamics of a Newton Raphson iteration) do not make sense as physical dynamics; yet *the results* (the fixed points, the bifurcation points) are the same as the corresponding states of the original system—and they have been found "much easier" than an experimentalist, or a "direct simulator", would find them.

The plan is, therefore, to construct such alternative computational ensembles that will discover the information we want to obtain from the code "faster and easier" than direct simulation. This is routinely done today, for example through numerical bifurcation theory algorithms, or through optimization algorithms – using knowledge about the mathematical characteristics of the problem we want to solve (e.g. eigenvalue conditions for a boundary of stability, Karush-Kuhn-Tucker conditions for optimality) we construct algorithms that wrap additional considerations around the same right-hand-side and search more efficiently than direct simulation could search. In general, such codes (optimization codes, bifurcation codes, controller design codes) have to be written from scratch, even if they do use some of the subroutines that the direct simulation code uses.

The "numerical analysis of legacy codes" discussed later in Section V, and the associated "time-stepper based bifurcation analysis" combine the two approaches: direct simulation *and* alternative algorithm ("ensemble") construction in what we believe is an ingenuous "software engineering" way. One uses the direct simulator as a timestepper -a simple subroutine that integrates starting at an initial condition

in time, and reports the result a little later. The point of algorithms like RPM is to extract from the time-stepper what a fixed point Newton algorithm would need: residuals (that is easy) and the action of Jacobians (that is a little more intricate, and uses iterative linear algebra/matrix free ideas). Using the time-stepper as a tool that allows us to identify the quantities that we need in order to do numerical analysis of the "unavailable equation" lies, as we have already said, at the heart of the overall procedure.

One might, at some level, consider that what we are doing is adaptive control on the time-stepper using it as an experiment. The difference though –and it is a crucial difference– is that the time-stepper can be *initialized at will*, while a physical experiment, even in closed loop, is not so easy to initialize, and evolves as the physical dynamics dictate in real time.

Let us stress once more the two-tier structure of the EFM numerics: an inner numerical algorithm (the LB-BGK time integrator) is called repeatedly, and its results processed by an outer numerical algorithm (the RPM code). It is interesting that one can study the numerical analysis (stability, convergence) aspects of the method without having to worry about the exact nature of the "inner" code. The only element required to parametrize this inner code from the point of view of the outer one is the "coarse linearization" spectrum of the inner code; this can be obtained off-line, and using the inner code only [3, 63]. As we will discuss below, this two-tier "inner-outer" structure makes also sense in the case of "legacy code" numerical analysis, when the "inner" timestepper is not a microscopic/stochastic code, but rather, a continuum model solver.

3 COARSE EQUATION-FREE COMPUTATIONS IN SPACE AND TIME

The discussion up to this point involved results that have either been published, are in press or under review; we will now proceed to discuss new results. So far we have been concerned with systems without any reference to spatial structure. With reference to the top of Table 1, we have enabled "short time" coarse timesteppers (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" timestepper 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.

If we combine the gap-tooth scheme with the coarse projective integration we get the "patch dynamics" which enables the "small space, short time" gap-tooth timestepper to perform "large space, large time" tasks. Similarly, the combination gap-tooth scheme and RPM enables "large space, inifite time" tasks. See Table 1.

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

$$\frac{du}{dt} = f(u), \qquad u \in \mathbb{R}^m, \qquad m \gg 1, \quad + \text{ boundary conditions.} \tag{12}$$

However, the unknowns are now connected to spatial points and there is also an associated set of boundary conditions. The unknowns u in (12) 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 (12) is solved increases in size, the dimension n of the problem also increases. We therefore assume that (12) can only be solved in small domains, for short times. As in Section 2, the ODE description is a simplifcation and there are microscopic evolution laws that do not fit in this example framework, but for which the procedure described here is still applicable, 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 evolutionary, coarse, PDE which, in one dimension has the form

$$U_t = L(t, x, U, \partial_x U, \ldots, \partial_x^p U). \tag{13}$$

for some L and integer p. Even though this equation conceptually exist, it is not explicitly available in closed form. At this point we are thus faced with

> Problem C: We want to simulate (13) for large or infinite times over large spatial domains, but we can only solve (12) in small spatial domains and for short times.

Our strategy will follow that in Section 2: construct accurate timesteppers for a semi-discretization of (13) *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 (12) 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 (13).

Consider a discretization of (13) 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 together in the tooth together with a with a lifting operator. In addition, the gap-tooth scheme must also construct *boundary conditions* for the microscopic equation (12) 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

(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 (12) 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 (13) 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.

We should comment on the choice of boundary conditions and lifting in steps (a) and (b) above. If (13) 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

$$s_{j+\frac{1}{2}} = (U^N_{j+1} - U^N_j)/H, \tag{14}$$

if $\{U^N_j\}$ 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. 18). Therefore

$$s^-_j = \frac{(H+h)s_{j-\frac{1}{2}} + (H-h)s_{j+\frac{1}{2}}}{2H} \tag{15}$$

$$s^+_j = \frac{(H-h)s_{j-\frac{1}{2}} + (H+h)s_{j+\frac{1}{2}}}{2H}. \tag{16}$$

See also the discussion in Section 3.2.

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^n_j. *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 Diffusion Equation

We now consider a one-dimensional example where the coarse equation (13) exists and we know it in closed form. More precisely, let us study the linear diffusion equation,

$$U_t = U_{xx}. \tag{17}$$

The corresponding microscopic equation (12) 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 (12) is *also* the diffusion equation (17). 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 $0 \le y \le h$. 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 do the following steps.

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

$$\partial_y v_j(t,0) = s_j^-, \qquad \partial_y v_j(t,h) = s_j^+, \qquad t > 0. \tag{18}$$

The prescribed slopes, s_j^- and s_j^+, are computed from $\{U_j^N\}$ using (14), (15) and (16).

(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 $Q(y)$,

$$v(0,y) = Q(y) := Ay^2 + By + C \tag{19}$$

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

$$Q'(0) = s_j^-, \qquad Q'(h) = s_j^+, \qquad \frac{1}{h}\int_0^h Q(y)dy = U_j^N. \tag{20}$$

We get

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

Note that A, B and C all depend linearly on elements of $\{U_j^N\}$ via (14), (15) and (16).

(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

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

with initial data (19) subject to boundary conditions (18). 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 (19), it has an *explicit* solution:

$$v_j(t,y) = Q(y) + Q''(y)t = Q(y) + t\frac{s_j^+ - s_j^-}{h}. \tag{21}$$

(d) *Restrict.* Compute U_j^{N+1}, the discrete approximation of $U(t, x_j)$ at time $N\tau$, by averaging in space over the small box solution $v_j(t,y)$ at the final time τ. From (21) we obtain

$$U_j^{N+1} = \frac{1}{h}\int_0^h v_j(\tau,y)dy = \frac{1}{h}\int_0^h Q(y)dy + \tau\frac{s_j^+ - s_j^-}{h}. \tag{22}$$

By also using (20), (15) and (16), 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 (16) we get the new coarse mesh quantities,

$$U_j^{N+1} = U_j^N + \tau\frac{U_{j+1}^N - 2U_j^N + U_{j-1}^N}{H^2}. \tag{23}$$

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 (23) 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 (17) 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 accurcay 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

$$O\left(H^{p_0} + \tau^{p_1} + |H^q - h^q|\right). \tag{24}$$

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

$$O\left(H^{p_0} + \left(\frac{h}{N_f}\right)^{r_0} + \tau^{p_1} + \tau_f^{r_1} + |H^q - h^q|\right). \tag{25}$$

In addition to the consistency of the schemes as expressed by (24) and (25), 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 is 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, [94, 95, 96, 97].

- 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 [98].

While this scheme seems computationally intensive (solving in each small box, averaging and interpolating is more expensive than the corresponding "straight" 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 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" [11, 106, 99, 104]: 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 were 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 also envision (although the details need to be worked out) that multitiered *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 [18]; we believe that testing such "constitutive failures" can be reasonably done "on line", and we discuss how to do it in [33, 55]. We believe that the ability to detect, test for, 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 failure".

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 (see Table 1). 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 that the gap-tooth timestepper can be combined 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. 19. In the middle (a) we see that, in order to evolve the problem with the "detailed microscopic" 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). What enables the simulation to be done faster, if at all, is the basic premise that *coarse behavior* exists – that "emergent" behavior (for lack of a better word), that is smooth in space and time at much larger scales, exists for the moments of our microscopically evolving distributions. Projective 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 exploting 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 "intelligently post-processed" 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 computed in real time ("on demand"), the macro-solvers (the "outer" computational codes) do not realize that the information they process does not come from a closed formula.

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. 20 contains a schematic summary of the patch dynamics procedure which we now repeat in words. 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 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 lifiting operator μ that lifts the finite representation of the coarse variables, $\{U_j^N\}$, to one profile 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" that provides us with the "small space, short time" to "large space, short time" enabling technology. We assume that a macroscopic initial profile $U(t=0,x)$ has been prescribed and that its finite 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 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 (12), 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 a time sequence of the basis functional 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. 21. 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. [101]).

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 [102], where the coarse grid solution is interpolated in time and used to overpose the boundary conditions for the fine scale imbedded solution.

Fig. 22, borrowed from [20] 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 [103, 104, 20, 105, 106, 107]. 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 [105, 106] when these are known, constrained dynamics [103], and feedback control (the Optimal Particle Controller of [20]). 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.

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 Figure); 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. 9 and Fig. 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 (23) 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 "talk to" 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 "overall patch size" and "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 first step towards answering this question is made by observing that appropriate BC for macroscopic evolution problems are often closely related to the highest spatial derivative in the model equations. To estimate the highest order spatial derivative

in the right hand side $P(\partial_x, x, u)$ for evolution PDEs of the form $u_t = P(\partial_x, x, u)$ we developed the "baby-bathwater" algorithm (BBA) [108]. In the BBA we evolve ensembles of initial conditions in a periodic domain using a "black box" simulation code (this can be the micrscopic solver, or even a "legacy timestepper" for an unknown PDE). We construct ensembles of (spatially periodic) initial conditions which at some point in space x_0, share the same value; or share the same value and the same first spatial derivative value; the same value and the same first and second spatial derivative values; and so on. Let $\{u_j^M(0,x)\}_{i=1}^n$ denote an ensemble of n initial conditions such that $\partial_x^m u_j^M(0,x_0) = \partial_x^m u_j^M(0,x_0)$ for $i,j = 1,\ldots,n$ and $m = 0,\ldots,M$. We note that if $P(\partial_x, x, u)$ only depends on derivatives of order equal to or less than M, then so does $\partial_t u_j^M(0,x_0) = \partial_t u_j^M(0,x_0)$ for $i,j = 1,\ldots,n$. Therefore, tracking the results of a short time simulation at the point x_0, i.e. studying $\{u_j^M(t,x_0)\}_{i=1}^n$ with $t \ll 1$, shows at what level we have thrown the "baby" (the highest *significant* spatial derivative) along with the "bathwater" (the higher order, non-significant ones). We have already tested this algorithm both at the "legacy PDE" level and at the level of "coarse" studies of microscopic simulators [108].

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 [109].

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 [34, 35, 1].

$$u_t(t, \mathbf{x}) = u_{xx}(t, \mathbf{x}) + u(t, \mathbf{x}) - u^3(\mathbf{x}, t) - v(t, \mathbf{x}) \tag{26}$$

$$v_t(t, \mathbf{x}) = \delta v_{xx}(t, \mathbf{x}) + \epsilon(u(t, \mathbf{x}) - a_1 v(t, \mathbf{x}) - a_0) \tag{27}$$

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 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 (26), (27), as well as through a kinetic theory motivated Lattice Boltzmann-BGK code [36]. 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.

Insert description of LB-BGK model and its relation to FHN here.

Another reason for choosing this example is that it exhibits a rich variety of nonlinear dynamic patterns, including multiplicity of spatially structured steady states (and we will work with sharp, front like ones, both stable and unstable) as well as spatiotemporal oscillations consisting of the palindromic movement of sharp concentration fronts (we will also work with these). 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.

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.

Newton-based and RPM-based bifurcation analysis results (stable and unstable steady states, limit cycles, computation and continuation of codimension one "coarse

bifurcation" points) for our FHN test problem, as well as other problems, have appeared in the past [1, 4]. A summary of these results (to be compared to additional, "small space" gap-tooth results later) is included in Fig. 3 (bifurcation diagram of the FHN with respect to the bifurcation parameter ϵ, showing a Hopf bifurcation), Fig. 4 (showing representative u and v spatially varying steady states for a value of the parameter), Fig. 5 (showing the leading eigenspectrum just before and just after the Hopf bifurcation) and Fig. 6 (showing the corresponding critical eigenvectors (both the "PDE-based" and the "LB-RPM small Jacobian" identified ones).

Be careful to mention patch boundary conditions everywhere.

Fig. 7 shows our first gap-tooth integration results for the FHN equation (this computation did not involve projective integration). In those simulations the domain length was $L = 20.0$ and the parameter, ϵ was set to $\epsilon = 0.5$. 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 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. 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 has evolved for one time horizon in each box using the "inner" solver, we *restrict* the solution, computing the averages of the densities in each box. 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. 7 both for "inner FD" and "inner LB" evolvers for comparison purposes.

In the simulations for the gap-tooth procedure shown in the following Figures, 101 boxes were used with $h/H = 0.1$, i.e. the box length was $h = 0.02$. Fig. 8 shows comparisons of full LB and LB-BOX profiles of the variable u at various times in the *oscillatory* regime of the FHN ($\epsilon = 0.01$). The solution is a spatiotemporal limit cycle, and the figure shows a portion of it, involving the palindromic movement of a sharp reaction front. Fig. 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 comparing with Fig. 10 where 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 sumulation "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 itegration 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 Allen-Cahn equation [100] of the form

$$u_t = u_{xx} + 2(u - u^3), \tag{28}$$

where the relaxation of the macroscale profile to the exact steady state of the form $u = \tanh(x - x_0)$ was observed. The simulation used an "inner FD" scheme in each patch (with different numbers of nodes in the short space of the patch and appropriate time steps between each "outer" reporting horizon).

Results of "patch dynamics" and "patch bifurcation" computations for our FHN equation based LB-example are now presented and discussed. We already showed in Fig. 3 the coarse bifurcation diagram of the FHN obtained through a "large space" LB coarse timestepper; the figure also shows, for comparison, the same bifurcation diagram obtained with a "small space" LB coarse timestepper (a two-tier nodal-LB scheme that we call LB-BOX). Here, 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). 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. Furthermore, their stability (embodied in the coarse linearization) is also well captured. This can be seen both for the LB and for the LB-BOX scheme. The LB-BOX 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. The combination of the gap-tooth scheme with techniques like RPM allows us, therefore, to perform coarse bifurcation computations using "boxes" or "patches" in space- it provides a "small space, short time to long space, infinite time" framework.

Fig. 11 compares full LB with patch dynamics (gap-tooth combined with projective integration) results; the algorithm now has several tiers. There are two tiers in time: the inner integrator is a "gap-tooth" LB, 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 "microscopic evolution physics". In particular we call this a 100-100 "inner" 200 "outer" integrator, since the forward Euler "jump" is equivalent to 200 time steps of the inner integrator (see also [4]). Fig. 12 shows the corresponding short-term integration results, and Fig. 13 shows the long-term periodic attractor, successfully captured by the LB-BOX-Projective scheme. These are, of course, preliminary results in a regime where the "coarse PDE" is known, where a lot of variance reduction is not really necessary, and where therefore we find a context for troubleshooting the computational procedure and exploring the numerical analysis of the multi-tiered schemes.

This completes the "simulation" part of EFM toolkit. We now have, through the combination of the gap-tooth scheme with projective integrators and RPM what we call "patch dynamics" and "patch bifurcations" respectively: a "small space, short time" to "large space, long/infinite time" enabling methodology. 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. This "coarse simulation" technology can also be coupled with both controller design and optimization algorithms.

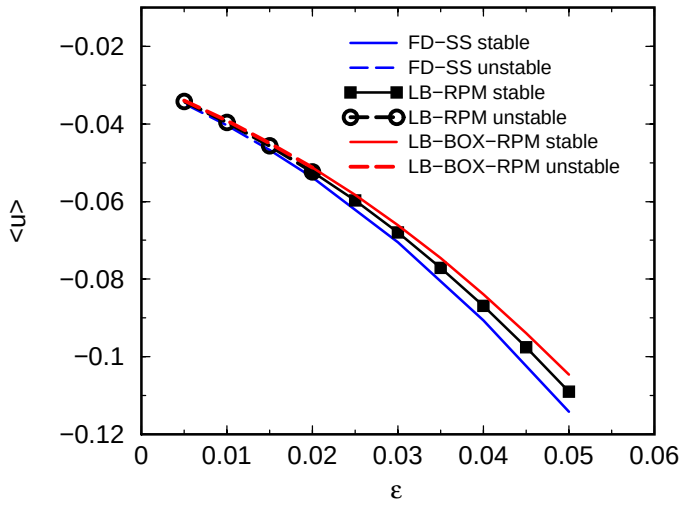


Figure 3: FHN bifurcation diagram near a Hopf bifurcation point computed with a steady Finite Difference (FD-SS) code (blue line), our LB-RPM code (black line) and our LB-BOX gap-tooth scheme combined with RPM, called LB-BOX-RPM (red line). Stable (unstable) steady states: solid (broken) lines.

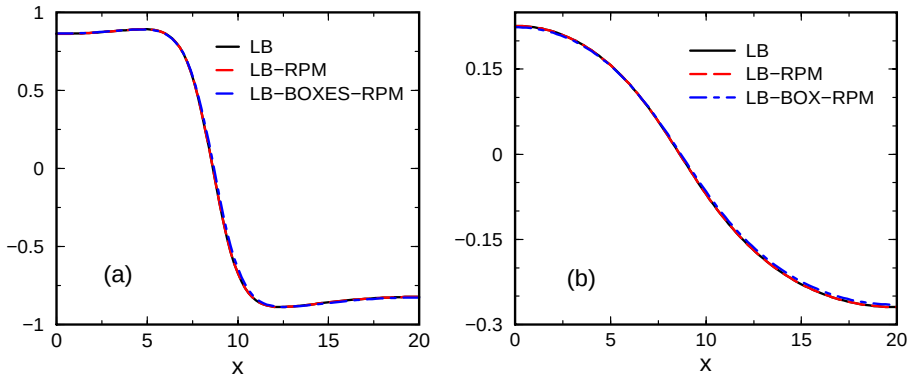


Figure 4: Comparison of the coarse steady state spatial profiles of (a) u and (b) v at $epsilon = 0.05$ (a stable steady state) computed with direct LB simulations (black solid lines, LB combined with RPM (LB-RPM) simulations (red broken lines) and our LB-BOX gap-tooth scheme combined with RPM (LB-BOX-RPM -blue broken lines).

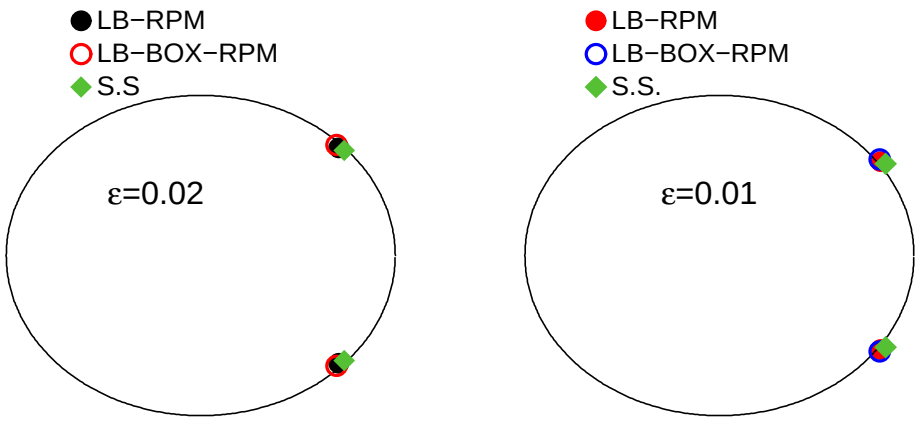


Figure 5: Leading coarse eigenvalues before ($\epsilon = 0.02$) and after ($\epsilon = 0.01$) the Hopf point, obtained upon convergence of LB-RPM (filled black circles) and LB-BOX-RPM (open red circles). The exponentials of the eigenvalues (for the same reporting horizon) of a steady FD code (based on the PDE) are also depicted for comparison (green diamonds, SS).

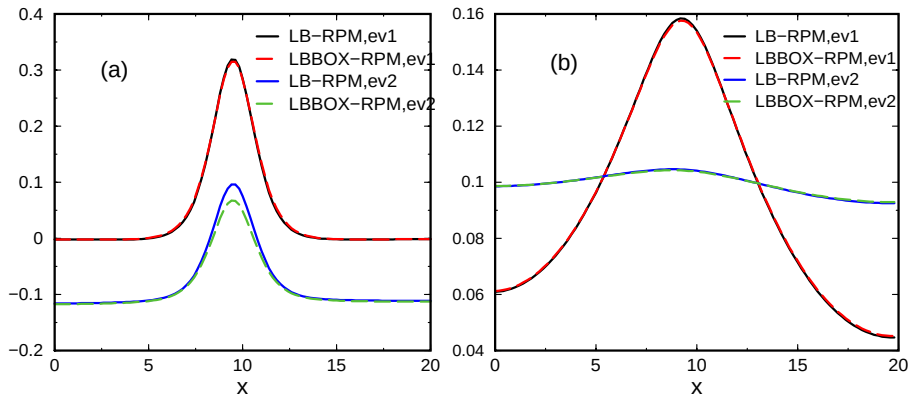


Figure 6: Comparison of (a) u and (b) v components of the critical coarse eigenvectors (ev1 and ev2) at $\epsilon = 0.01$ obtained upon convergence of the LB-RPM and LB-BOX-RPM computations.

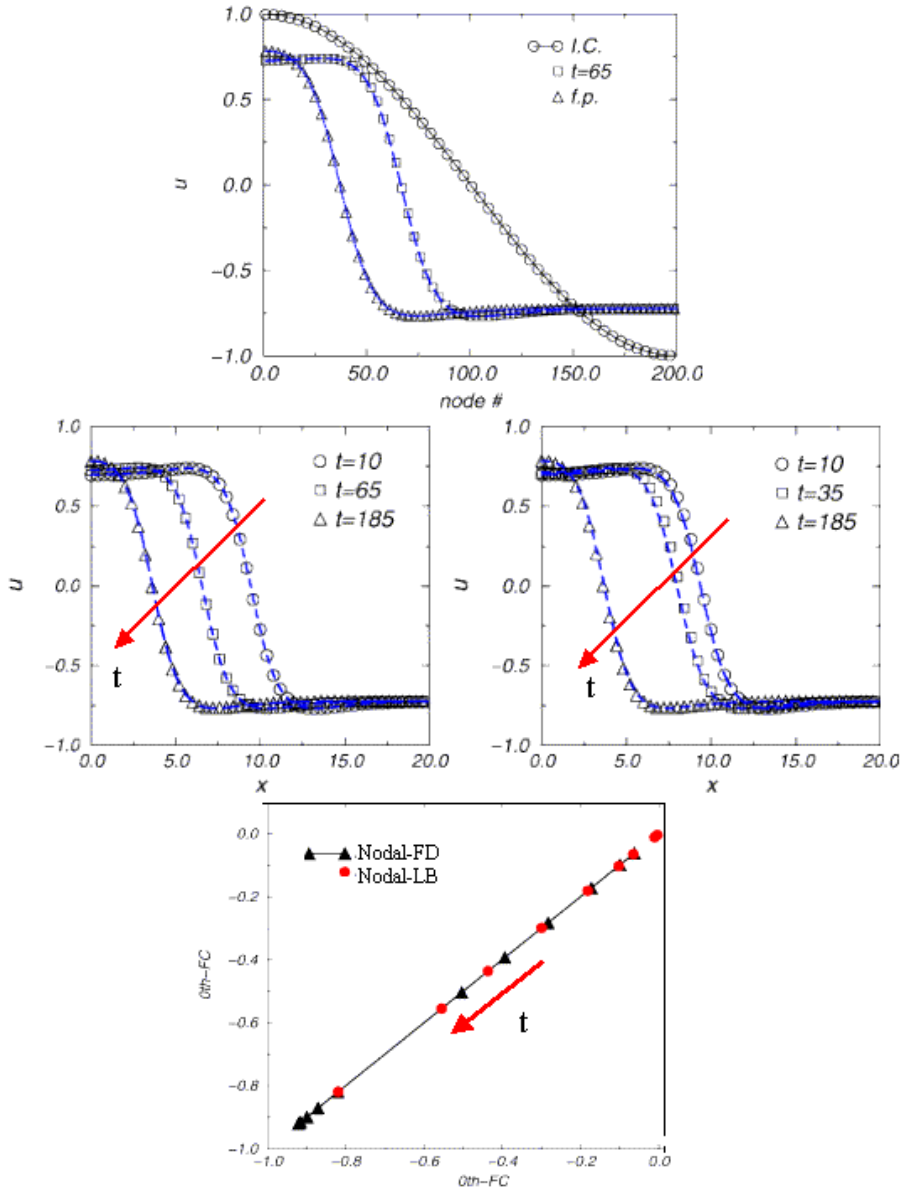


Figure 7: "Inner LB" gap-tooth integration results for the FHN equation (symbols) compared to full LB simulations (curves) for various initial conditions. The last figure shows a phase portrait for this simulation for two different gap-tooth schemes: an "inner LB" (FD-LB) and an "inner FD" (FD-FD) continuum scheme.

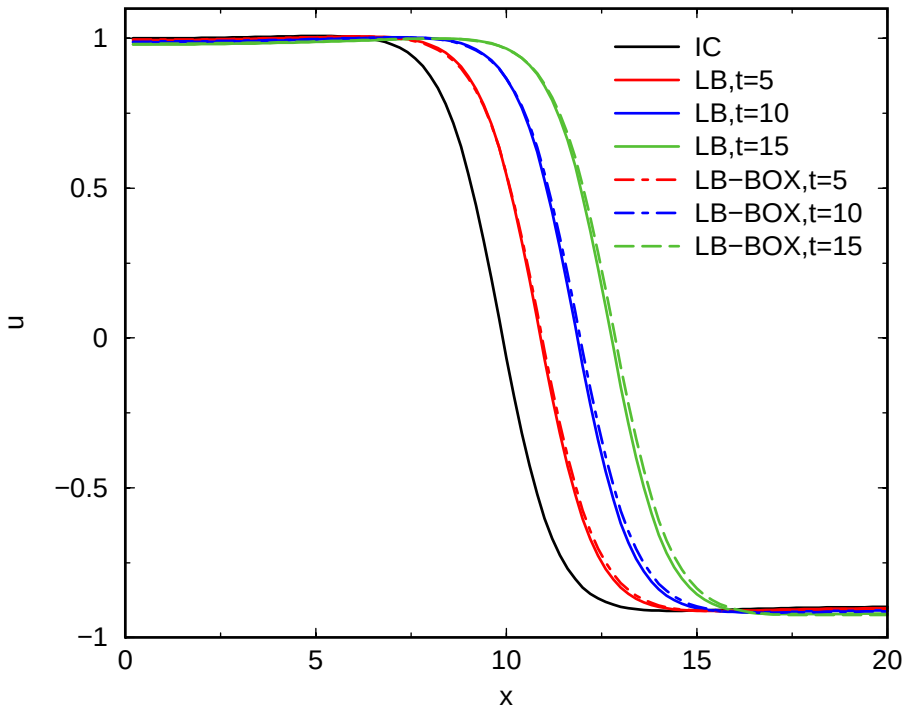


Figure 8: Comparisons of LB and LB-BOX spatial profiles of u at various times in the unstable (oscillatory) FHN region ($\epsilon = 0.01$) showing a palindromic movement of the reaction front.

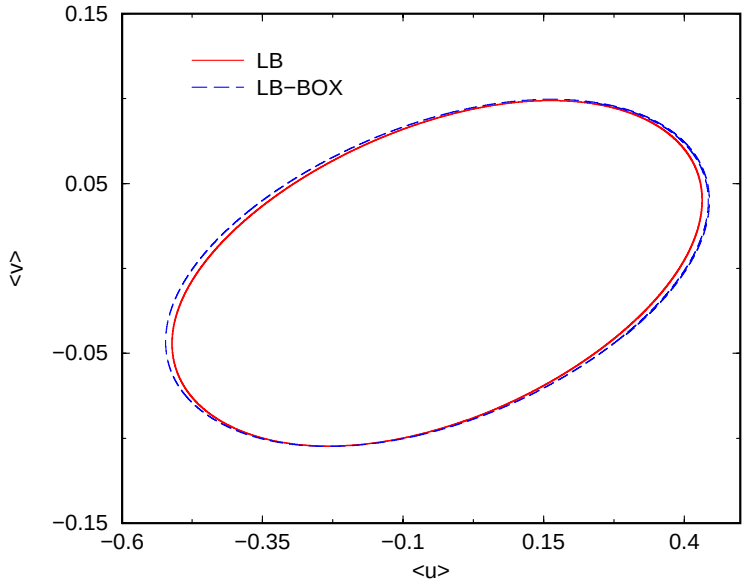


Figure 9: Comparison of phase space projections of the LB (black) and LB-BOX (red) long-term attractors. Reporting horizon for communication between boxes $T_c = 7.5 \times 10^{-4}$.

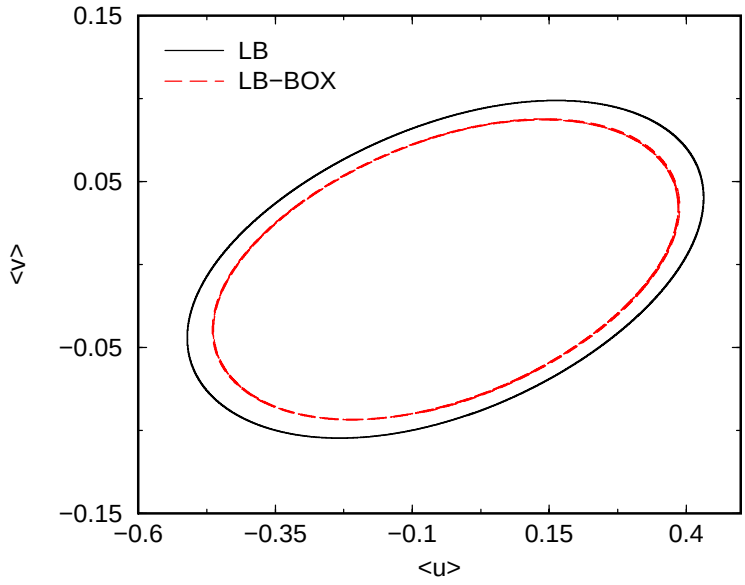


Figure 10: Comparison of phase space projections of the LB (black) and LB-BOX (red) long-term attractors. Reporting horizon for communication between boxes $T_c = 2.5 \times 10^{-3}$.

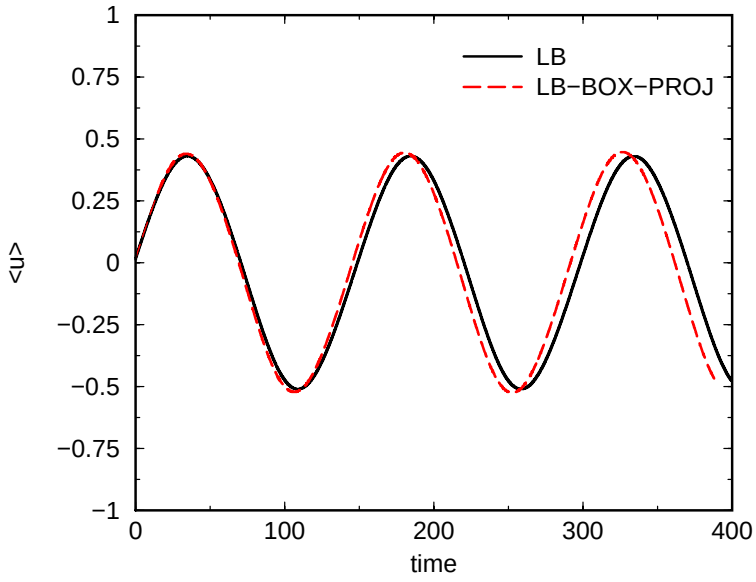


Figure 11: Computed time-series comparison of the oscillatory FHN dynamics ($\epsilon = 0.01$) computed by a full LB code (solid black line) and by a 100-100-200 LB-BOX projective scheme (broken red line).

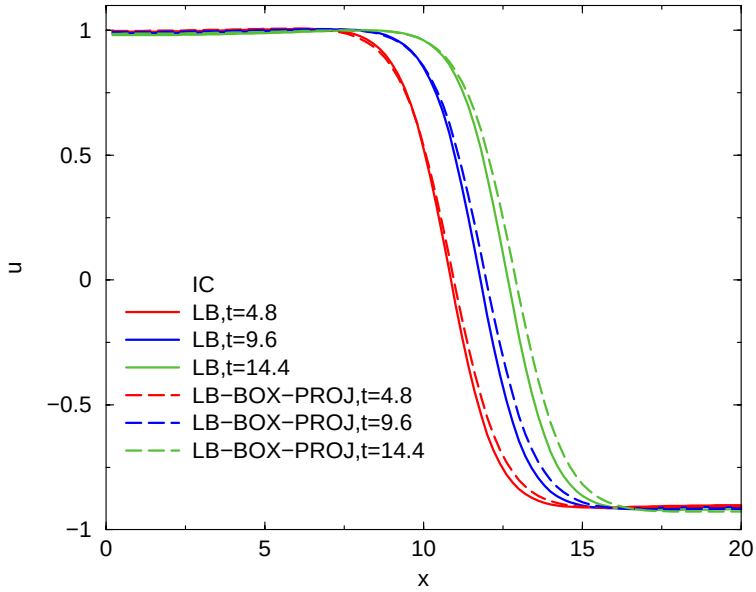


Figure 12: Comparison of spatial profiles of u at various times in the unstable (oscillatory) FHN region ($\epsilon = 0.01$) computed by a full LB code (solid lines) and by a 100-100-200 LB-BOX projective code (broken lines). The boxes cover 10% of the macroscopic computational domain.

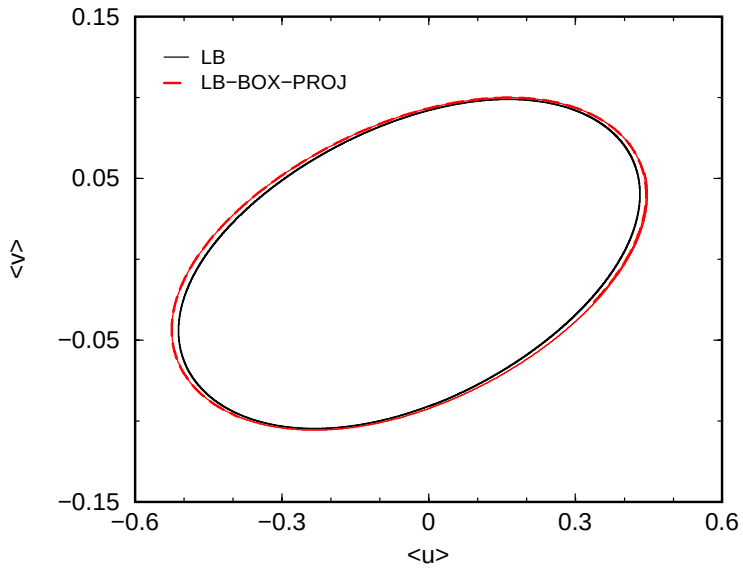


Figure 13: Comparison of phase space projections of the long term attractors of a full LB (black line) and the 100-100-200 LB-BOX projective (red line) attractors. Reporting time for communication between boxes $T_c = 7.5 \times 10^{-4}$.

4 A DISCUSSION OF THE INGREDIENTS

In this section we discuss several elements (all of them more or less well established) that underpin the "computational bridge" we want to construct between microscopic simulation and traditional, macroscopic scientific computing/numerical analysis. This discussion can be skipped at a first reading of the paper, and studied after a first pass of the new algorithms and results sections.

4.1 System Identification

The most important element comes from systems theory: system identification lies at the heart of what we propose. The point is to consider the microscopic simulator as an experiment, from which measurements are made. It is crucial that this computational experiment can be *repeatedly initialized* almost at will; this is to be contrasted to a physical experiment, which cannot practically be initialized at will. Using information obtained from several short dynamic runs of the microscopic code, runs that have been initialized "intelligently", possibly by post-processing results of previous runs, one can essentially identify (estimate) useful elements of the unavailable continuum model. This procedure has components of what are called "just in time" or "model on demand" approaches in learning theory [78, 79]. Whether because of the lifting procedure, because of the noise inherent in stochastic "inner" timesteppers, or even from the choice of the restriction operator, it is clear that accurate estimation of various coarse quantities and derivatives is crucial for the successful performance of the "outer" algorithm. We rely on theory developed in the systems area [60, 61, 62] for the best unbiased estimates of these quantities (see also the review [80]). Since we can initialize the simulation at will, however, we also have the option of selecting appropriate ensembles, so that we can improve variance reduction for a given computational effort; variance reduction plays a crucial role in the accurate estimation of coarse values and coarse derivatives.

Because in our case the missing element is the closure of the macroscopic equations -and that is what, in effect, is accomplished through the identification procedure-, we believe that "closure on demand" is an appropriate description for our computational framework. One then "passes" the on-demand identified coarse model quantities to the standard continuum scientific computing subroutines. These proceed to perform the computer assisted analysis "without realizing" that the numbers they crunch do not come from a subroutine evaluating an explicit coarse description, but rather from a subroutine that identifies these quantities "on demand" through short computational bursts of the microscopic description. Part of the macroscopic operations performed includes the post-processing of simulation results to construct new, useful initial conditions for the microscopic timestepper: for example, a new initial guess for an outer Newton iteration, an outer optimization algorithm, or for an outer Arnoldi eigensolver.

4.2 "Numerical Analysis of Legacy Codes"

The next important element is what, for lack of a better expression, could be called "the numerical analysis of legacy codes"; this is the context in which we first encountered such ideas. It is a context of great practical importance in itself, but at the same time it is truly valuable since it provides a natural framework for the numerical analysis of what we propose. The Recursive Projection Method (Shroff and Keller, 1993 [37], see also [39]) is representative of this context (and of what has come to be called "time-stepper-based" numerical bifurcation theory [41]). Consider that a company or National Laboratory (which, as we understand, was the case for the inception of RPM) has a large scale dynamic simulator of a physical process (what we will call "the legacy code"). This dynamic simulator requires often tens of man-years of development and validation, and it embodies the best physics description of the process available. As an illustration, consider that this code is used, say, to detect a steady state by transient integration; it is then conceivable that the acquisition of the desired result (the steady state) may be significantly accelerated by a Newton-Raphson procedure. However, rewriting the code as a Newton-Raphson iteration may be practically impossible - the original programmers may have left or retired, and rewriting the code from scratch might be an immense undertaking. Alternatively, the evolution code may not involve simple right-hand-side evaluations, but, for example, split-step iterations, and a steady solver will be far from a small modification. The alternative is to write a software wrapper "around" this code, calling it as a subroutine. The "wrapper" will identify, using "intelligent" calls to the code, what a Newton iteration would require; and thus exploit the simulator (the "time-stepper") making it the core of a continuation - stability - bifurcation code.

This combination of system identification with numerical analysis is motivated by the structure of large scale numerical linear algebra techniques, based on matrix vector product computation. These matrix-vector products can be approximated through timestepper calls in a "matrix free" environment. This was demonstrated in the work of Shroff and Keller where, exploiting separation of time scales for the original problem, it was proved that the overall procedure converges. The main element in the procedure is its two-tier structure: one has an "inner" simulator (the legacy code) and an "outer" processor (identification plus whatever we do with the identification results, here a contraction mapping). It is this two-tier structure that is important for numerical analysis purposes. There is an extensive and growing literature on the use of techniques like RPM in the inner "legacy timestepper" context (e.g. [46, 52, 51, 53]), not only for steady states but also for periodic solutions, whether autonomous [47, 48] or periodically forced [81].

In our research so far we have substituted the "inner simulator" with the (appropriately initialized) microscopic system model. However, one can perform the analysis of the two-tier process assuming a "general" inner simulator: for example, a "perfect" inner simulator as will be shown for the "gap-tooth scheme" below, or a "typical" inner integrator as can be found in our work on Projective Integrators [3, 63]. One can then proceed to analyze the stability and convergence of the procedure based on nominal representative properties of the inner scheme (like the amplification factor

for an integrator, as in [3]) without worrying at this level about what the detailed inner scheme will eventually actually be. The results of the overall process for a particular inner scheme may then be found through the properties (e.g. the integration amplification factor) of that scheme itself.

As we discussed above, RPM is a computational superstructure that enables a "short time" legacy code to perform "infinite time" calculations (find elements of the ω-limit set of the problem). What we call "projective integration" is similarly a computational framework that enables a "short time" legacy code to perform "longer time" calculations. Along the same lines, the so-called "gap-tooth" scheme is a computational framework that enables a "small space, short time" legacy code to perform "large space, short time" calculations. The gap-tooth–projective integration combination (that we call "patch dynamics") allows then the "small space, short time" legacy codes to perform "large space, long time" calculations, while the gap-tooth–RPM combination allows them to perform "large space, infinite time" calculations. The scope of these schemes is summarized in Table 1. It is not clear which of these names will withstand the test of time, but for the moment we will keep using them for brevity.

Over the last few years we have demonstrated that this "enabling legacy codes" approach can be used by substituting at the inner level an appropriately initialized microscopic code at the place of the legacy simulator. The "coarse timestepper" in effect, evolves distributions -and thus moments of distributions- conditioned on (initialized consistently with) initial values of typically a few of these moments; thus the microscopic code becomes our main tool for conditional probability estimation [15, 16]. Some of the combinations we consider (timestepper and RPM, projective integration) may make perfect sense for legacy codes; the rationale of other combinations (the "gap-tooth scheme") might appear a little warped in the context of legacy codes, but much more natural, as we shall see, and even quite powerful, in the context of microscopic inner simulators. Shroff and Keller's RPM, the Newton-Picard methods of Lust et al., our own Projective Integrator analysis [3, 4, 63] as well as the glimpses of gap-tooth numerical analysis presented below make, we hope, the point.

4.3 Separation of Scales

After system identification, the matrix-free computations that arise in the numerical analysis of legacy codes bring up the next important element in the overall procedure: separation of time scales (and the concomitant separation of space scales and the smoothness it induces on the expected behavior). As one sees in the numerical analysis of RPM, a gap in the eigenvalue spectrum of the linearization of the problem one studies is useful for the fast convergence of the method. The same appears in the study of projective integrators—and is also very important in the identification of the slow subspace of the evolution of the "unavailable coarse equation".

Separation of time (and sometimes concomitant space) scales arises in several important ways. Possibly the most important is in the way it appears in the RPM itself: as a spectral gap *within* the dynamics of the legacy code (or the "unavailable PDE"). This is typical of dissipative evolution equations (equations characterized by diffusion

-molecular or turbulent-, viscosity, heat conduction...) where the spatial operators have dispersion relations that give rise to large, increasingly negative eigenvalues. If only a few modes, with eigenvalues close to the imaginary axis (possibly unstable) are "slow", then a spectral gap exists; to the left of it follow the (infinitely many) fast, strongly stable modes, and RPM is shown to converge "fast". This dissipative PDE/separation of time scale context is also the realm of Inertial Manifolds/Forms and Approximate Inertial Manifolds/Forms (IMs, AIMs, IFs and AIFs respectively, [43, 42]). It is also the context in which the singularly perturbed control methodologies by Kokotovic and coworkers [82] have been developed over the years; these will become important in several ways below.

Time scale separation also arises through *irreversibility* and averaging, as one takes hydrodynamic limits of microscopically reversible processes (e.g. [26, 30]). In essentially all current microscopic/multiscale models we evolve distributions (of atoms, particles, cells, individuals) and try to analyze (simulate, design, optimize and control) closed deterministic equations for a few order parameters, usually the first few moments of these distributions (alternatively, well-chosen phase field variables). A typical example would be the distribution of molecules ("kinetic entities" for a kinetic theory) over physical space, time and velocity space. It is convenient to consider the Navier-Stokes as an example of explicitly closed macroscopic "coarse" equations for fields *of the first two moments* (density and momentum) of this distribution over velocity space—(a kind of *Approximate Inertial Form* closure of the moments hierarchy of the Boltzmann equation). We will discuss below the issue of choosing possibly "better" coarse variables than these first few moments; let us temporarily think in terms of writing an (infinite) hierarchy of (coupled) equations for the evolution of moments of the distribution.

If coarse, deterministic, macroscopic behavior exists (we fully realize this is a colloquial, and not a mathematical statement), the implication is that –for the Boltzmann/NSE example– one can indeed predict, over coarse space and time scales, the behavior of the first two moments of the distribution (density and momentum) based *only* on these two moments. It is the *closure* of the infinite moment hierarchy that allows us to write the Navier-Stokes: they contain a model of the effect of all the higher moments on the first two (Newton's law of viscosity). The fact that a *useful* deterministic equation exists and closes in terms of only the first two moments does not imply that the higher moments are negligible - far from that. It implies that the higher moments of the distribution get *very quickly* (compared to the time scale over which coarse deterministic predictions are made) slaved to (become functionals of) the lower moments—density and momentum.

The theory of inertial manifolds provides good "thought analogies" and nomenclature for such a situation. In the same way that one writes an (approximate) inertial form in terms of the first few eigenmodes of the linear part of the problem, one here writes the Navier-Stokes in terms of the first few moments of the evolving distribution. By analogy to "determining modes", we can call these first few moments *the "determining moments"* for the coarse evolution of the distribution.

By analogy to the parametrization of the (approximate) inertial manifold as a

function over the determining modes, we can think of the existence of an "almost always" forward attracting, smooth, invariant manifold in moments space (in the case of IMs, the manifold is Lipschitz and exponentially attracting [43, 42]). When the distribution is initialized "far away" from it, the higher moments get quickly attracted to this manifold, and thus become functionals of the lower (governing) moments. In that sense, we can think of the Navier-Stokes as some kind of "approximate inertial form" of the (non-closed) hierarchy of equations for the evolution of the moments of the distribution. Alternatively, this "slow moment manifold" can be thought of as embodying a closure in the mode hierarchy in the appropriate space (see [83] for a recent corresponding study for diffusion).

It stands to reason that such a slaving is at play when coarse macroscopic deterministic behavior is observed: if the higher moments *were truly individually important* then knowing only the first few (in effect, conditioning the distribution on the first few) would not be enough to have deterministic predictions later on in time. It is conceivable that one can come up with examples in statistical mechanics where closed equations can be obtained for some moments *even without* slaving; in most of the practical physical problems that we want to model, however, quick dynamic slaving of the high moments of the distributions, effected by the inner, microscopic timestepper, lies at the heart of macroscopic coarse determinism.

Perhaps the most typical example of this "irreversibility induced" separation of time scales is the derivation of the diffusion equation from molecular dynamics; under appropriate conditions, and after an initial few molecular collisions, the density field effectively satisfies the diffusion equation; all higher moments of the distribution have been slaved to the zeroth moment (density) field. Similarly, microscopic (MD,DSMC) simulators effectively "see" the Navier Stokes equations after a brief initial startup period [30, 84]. This is probably best illustrated by thinking of a (laminar, isothermal) fluid mechanical experiment as compared to a solution of the Navier-Stokes equations. We know that measuring the density and momentum with some resolution allows us to predict what they will become later in time: we interpolate (smoothness is important) the measurements, give them as initial conditions to the Navier Stokes, evolve the Navier-Stokes and obtain deterministic predictions for the same fields after some time. Clearly, the NS are "successful" closed equations for the expected values of two moments of the distribution; the actual positions and momenta of each molecule at the initial time are not necessary to make coarse deterministic predictions. It is worth reiterating that the microscopic (Newtonian collision) dynamics are fully reversible, yet the macroscopic coarse dynamics are dissipative.

There is a third way by which separation of time scales enters the problem, and this is the one that we have the most difficulty articulating; the discussion is still, we hope, informative. There exist physical contexts in which it does not make sense to attempt the derivation of an equation for the (expected value of a) *single* experiment; the system is "too noisy". While it is not possible to have deterministic predictions of the coarse state of a single experiment, we can still obtain good deterministic predictions for the *probability distribution* of coarse states of many experiments over said coarse state values (e.g. through Fokker-Planck equations). One evolves simultane-

ously several realizations of the problem with the same *coarse* initial conditions, and observes the evolution of the ensemble; noise comes from the *fine* initial conditions and/or the fine simulator. It is now the dynamics *for the probability distributions* of the coarse results that may acquire time-scale separation: higher order moments again quickly become functionals of lower order moments. In this case, however, these are moments of the evolving probability distributions of the coarse variables (alternatively, of the solution of the Fokker-Planck for the evolving probability distribution of the coarse variables).

A nice example arises in the rigid rod model of nematic liquid crystal rheology, [85, 86, 87], see [88, 55]. One cannot write a deterministic equation for the molecular orientation—but *can* write a good deterministic equation at the level of the single rod orientation distribution. This equation certainly has exponential attractors, and probably even has an inertial manifold. Macroscopic closures for the problem (that is, approximate inertial forms in terms of order parameters - which can be identified with low moments of the distribution) are well known and quite accurate in certain parameter regimes. So, inertial manifold "technology" may be invoked to explore whether indeed high moments of evolving probability density distributions get slaved to low moments of these distributions. In this case, it is the averaging over many realizations that "is responsible for" the time scale separation, the spectral gaps and the slaving; and it is the running and averaging of many similar experiments that allows us to make deterministic predictions for the *statistics* of the expected coarse state, rather than for the coarse state itself. One has to work at a different level of coarse description (statistics of evolving ensembles) in order to restore coarse determinism. Since the single realization is not "ergodic enough" any more, the modeler changes the problem by following many (similarly coarsely initialized) realizations.

What we do is to use the microscopic timesteppers to identify, on the fly, what amounts to the time-T map (the timestepper) and slow evolution subspace of the unavailable equation, be it an equation for the expected state of an experiment (like the velocity field in the Navier-Stokes) or the expected statistics of an experiment (like, possibly, the "energy versus wavenumber" spectrum in a turbulence calculation, as we will discuss below). We then use scientific computation to analyze/compute this equation as if we had it. Statistical mechanics and probability theory must therefore play a vital role in establishing or contesting the validity of the attempt to extend the"numerical analysis of legacy codes" into the "coarse numerical analysis of microscopic timesteppers".

4.4 Some Additional Considerations

It is interesting to consider that, in addition to direct simulation, most of the computational technology we use to enhance the exploration of our computational models is based on smoothness and the evaluation of derivatives. The Newton-Raphson is such a case, as are other steady solvers [89], continuation algorithms, parametric sensitivity, the integration of variational and Riccati equations, or optimization algorithms. Within these algorithms we need to evaluets Jacobians, Hessians, derivatives with respect to parameters etc. If the equations are not available, these derivatives have

to be identified from calls to the "inner" legacy code, and so numerical derivative estimation becomes important. In the context of microscopic/stochastic codes, we need to obtain accurate *coarse* derivative information directly from the timestepper: variance reduction is absolutely *crucial* for the practical usefulness of the ideas we discuss. There are several ways to accomplish variance reduction: in addition to large samples (something we want to avoid if possible for computational reasons) one could try "intelligent" (i.e. slightly biased) microscopic simulations, such as Brownian Configuration Fields [90, 91], see alse [92]. Systems theory comes once more to the aid of such computations; indeed, the filtering and estimation techniques that have been developed for noisy data processing and identification (from the famous Kalman filtering to the recent interest in synchronization and nonlinear observers [93]) can be brought to bear on the variance reduction problem.

One of the important issues in the "coarse observation" of microscopic simulations is the extraction of smooth, coarse, macroscopic fields (output) from, say, the detailed locations, velocities, and possibly recent histories of molecules in an MD simulation (the "microscopic state"). Such problems, using systems theory methods, are currently being tackled in materials science computations (see for example the "field estimator" [20] or the use of the Cumulative Distribution Function and filtering in [22]).

As we will see, additional elements of systems theory (in particular, controllers) will be combined with elements of continuum numerical analysis (the "gap-tooth" and "patch dynamics" schemes) and with new ensembles from microscopic simulation (in the spirit of the Dual Volume Grand Canonical MC or MD simulations [94, 95, 96, 97]). The crucial element here is the effective imposition of coarse boundary conditions on microscopic simulators; and the use of control-motivated methodologies to come up with the corresponding microscopic ensembles.

5 SOME APPLICATIONS

To demonstrate the scope of the techniques we have discussed, we present here a brief anthology of results that we have obtained doing "coarse" numerics on problems with various "inner timesteppers". This collage, and references to the corresponding publications, were selected to emphasize the broad scope of the problems that can be solved through the EFM computational approach.

Fig. 23 arises in the modeling of multiphase flows through LB-BGK simulations. It is a study of the dynamics and instabilities of a two-dimensional periodic array of gas bubbles rising, under the action of gravity, in a liquid. Time-dependent simulations both before and after this "coarse Hopf" instability, leading to macroscopic bubble shape oscillations, are shown, and the coarse bifurcation diagram shows both stable *and unstable* coarse steady bubble shapes. This is collaborative research with Dr. Sankaranarayanan and Prof. Sundaresan at Princeton [31, 32, 110, 111]

Fig. 24 shows bifurcation diagrams for lattice gas models of surface catalytic reactions (a caricature of CO oxidation) in the presence of adsorbate interactions. Here the "inner integrator" is a kinetic Monte Carlo code. One deals with average surface

concentrations (that is, coarse ODEs for expected coverages, as opposed to coarse PDEs in the examples above). It is interesting to compare the coarse bifurcation analysis obtained through our scheme with the results of "traditional" explicit closures included in the graph (mean field as well as quasi-chemical approximations) and with true KMC simulations (also included). Clearly, at these strong value of adsorbate interactions, microsctructure has started to develop on the surface; yet it is still possible (for a long enough reporting horizon) to analyze unavailable closed equations in terms of only the zeroth moments of the distribution (two coverages). This work is in collaboration with Dr. A. Makeev of Moscow State University, and Profs. D. Maroudas at UCSB and A. Panagiotopoulos at Princeton.

Fig. 25 shows the bifurcation diagram for an order parameter arising in the dynamics of nematic liquid crystals (more exactly, the Doi theory for rigid rods in shear). Here the inner time-stepper is Brownian Dynamics, the initial conditions are rod orientation distributions (over the unit sphere) evolved through a stochastic integrator, and the bifurcation parameter corresponds to the density of the rods. The Smoluchowski/Fokker-Planck equation for the evolution of the distribution has been analyzed through both spherical harmonic and wavelet methods ([88, 86]). This coarse bifurcation diagram (and the coarse eigenvalues computed in the process) confirms what is known from the numerically integrated Fokker-Planck itself. We have used this example to study the effect of closure conditioning on several moments, and have confirmed that the "next fastest" governing moment is approximately 500 times faster than the order parameter S [55]. This justifies one-mode closures in the literature, which can be rationalized as a one-dimensional approximate inertial manifold for the Smoluchowski equation. This work is in collaboration with Dr. C. Siettos and Prof. M. D. Graham at Wisconsin (and aspect of it are also studied with Prof. R. Armstrong and Dr. A. Gopinath at MIT).

Fig. 26 shows a "different" (non-microscopic) application of the EF methodology: a computation of "effective medium" behavior using coarse timesteppers. The example is taken from our recent paper [17]; it is a study of the dynamics of pulses in a reaction-diffusion problem, traveling in a one-dimensional periodic medium (consisting of alternating "stripes" of different reactivity). Assuming that an effective equation for a complicated medium exists, but is not available in closed form, we can construct an approximate timestepper for it by giving a coarse initial condition to an ensemble of realizations of the detailed medium, solving (for short times) the detailed equations, and averaging the result. In general, for a random medium, one would have to construct the ensemble of medium realizations through Monte-Carlo sampling; but in this example the medium is periodic, and so the different realizations simply correspond to shifts of the initial condition over the unit cell. The "effective equation" version of the coarse time-stepper is shown above in caricature, and the "effective" bifurcation diagram, showing a "coarse Hopf" bifurcation of a "coarsely traveling" wave to a "modulated coarsely traveling" wave is shown below. We believe that, at the appropriate limit, the time-stepper constructed through this procedure will approach the time-stepper arising from the discretization of the corresponding homogenized equations.

Fig. 27 is a sample of an "alternative" computational task—neither integration nor steady/bifurcation computation. It shows the coarse control of the expected value of a kinetic Monte Carlo simulation for a caricature of a catalytic surface reaction [18, 112]. Here the unstable coarse (expected) steady states, their linearization and bifurcations have been computed off-line using the coarse time-stepper. The identified linearization and coarse derivatives with respect to control parameters around an unstable steady state were used to design both an observer (a Kalman filter) and a (pole placement) controller for the system. The bifurcation parameter (corresponding to the gas phase pressure of a reactant) is used as the actuator in a regime where the coarse dynamics are oscillatory. The "coarse linear controller" is clearly capable of stabilizing the (unstable) coarse expected stationary state of the KMC simulation. This work is in collaboration with Dr. C. Siettos, A. Armaou and Prof. A. Makeev; work on the coarse control of distributed microscopic systems is also underway.

We are currently exploring, through collaborations, several problems involving several types of "inner timesteppers": with Prof. Katsoulakis we study coarse optimization and the computation of "coarse rare" events using timesteppers. With Prof. Bourlioux we explore turbulent combustion using coarse front evolution timesteppers. With Prof. Tannenbaum we study "coarse" computationns of microscopic edge detection algorithms; With Prof. Hadjiconstantinou and Drs. Alexander and Garcia we study coarse DSMC fluid simulation. Finite temperature MD stress induced crystal transformations is studied with Profs. Maroudas and Li Ju, and pattern formation in granular flows is studied with Professors Swinney and Swift.

The last application we include is very simple, but has, we believe, important implications. Fig. 28 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 [113, 114]. This template-based approach was motivated by the work of Rowley and Marsden for problems with translational invariance and traveling solutions [115]. The solution is rescaled either continuously, or discretely (after some time) based on minimizing its distance from a (more or less aritrary) template function (see also [116]). 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 successsive 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

the Figure. This work [117] is in collaboration with Prof. Aronson and Dr. Betelu at Minnesota, and we hope that it will allow the "coarse" study of many phenomena that involve self-similarity ("from cells to stars", [118]) 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 [119]. 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.

6 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 cicrumstances, 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. 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 [120]) 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. [81]).

This paper is 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 we briefly discussed, however, 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.

On a different note, a connection between coarse timesteppers and what is termed "gray-box identification" is best discussed considering an "unavailable macroscopic equation" of the form of (13). The coarse timestepper estimates the result of integrating the entire right-hand-side of the equation for time T. 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 [121, 122]. We are exploring the ability of this framework to practically close microsopic rheological simulations in collaboration with Prof. Oettinger.

While we discussed mostly parabolic problems in presenting coarse integration results, clearly coarse bifurcation algorithms like RPM provide the solutions to elliptic problems, and -as we briefly discussed in the section above- hyperbolic problems also fit naturally within this framework.

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 [123]; 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 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 energy spectra of DNS or LES simulations of the Navier Stokes equations by (a) conditioning NS initial condition ensembles on a given spectrum; (b) evolving the ensemble through DNS or LES for some time and (c) restricting back down to spectra. Aspects of this problem are being explored in collaboration with Profs. Martin, Yakhot, Jolly and Foias. The list of problems that can be attempted through this methodology (with no guarantee of solution, of course; the premise of "coarse behavior" has to be satisfied, and it cannot be *a priori* established) is vast. In the previous section we provided an anthology of systems we are working on currently; we have promising initial results on Monte Carlo models of bacterial chemotaxis (with Dr. Sima Setayeshgar and Prof. Hans Othmer) and in the derivation of translationally invariant effective equations for discrete systems (e.g. lattices of coupled neurons) in which we study the effect of front pinning and an analogy to the Portevin-le Chatelier effect (with K. Lust, J. Moeller and D. Srolovitz). Agent based models in ecology, epidemiology and evolutionary biology (the timesteppers do not have to be in physical time, they can involve mutations in a population) are also a possibility, which we are exploring with Prof. Levin; another possibility might be the coarse integration of unavailable equations for the expected backbone shapes of oligopeptides. 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* throgh dynamics (see recent work in [124, 125, 126, 127] as well as in [128]). 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 [129], 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 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 [130]; 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 continuum 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 [131, 132] 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 illustrated "coarse control" in the previous section, 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 parametrize the transitions in question. Just guessing what a set of sufficient such coordinates might be, would be enough. 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 two years we have discussed this research direction with our colleagues, Professors Engquist and E at PACM in Princeton. In the companion paper [109] they present an alternative formulation 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.

References

[1] C. Theodoropoulos, Y.H. Qian and I.G. Kevrekidis (2000). *Proc. Nat. Acad. Sci.* **34**, 9840-9843.

[2] P.L. Bhatnagar, E.P. Gross and M. Krook (1954) *Phys. Rev.* **94** pp. 511-525.

[3] C. W. Gear and I.G. Kevrekidis (2001) *SIAM J. Sci. Comp.* in press; also NEC Research Report NECI-TR 2001-029.

[4] C. W. Gear, I. G. Kevrekidis and C. Theodoropoulos (2002) *Comp. Chem. Eng.* **26**(7-8) 941-963.

[5] I. G. Kevrekidis, (2000) *Plenary Lecture, CAST Division of the AIChE*, 2000 AIChE annual meeting, Los Angeles; also www.princeton.edu

[6] Matlab, the Language of Technical Computing, the Math Works Inc. Natick, MA (2000).

[7] Doedel, E.J. (1981) *Cong. Num* **30** 265-284; Doedel, E.J., H. B. Keller and J.P. Kernevez (1991) *Int. J. Bif. Chaos* **1** 493-520.

[8] A.G. Salinger, N.M. Bou-Rabee, E.A. Burroughs, R.B. Lehoucq, R.P. Pawlowski, L.A. Romero, and E.D. Wilkes (2001), Sandia National Labs Technical Report; also R.B. Lehoucq and A.G. Salinger (2001) *Int. J Numer. Meth. Fluids* **36**, 309-327.

[9] C. C. Pantelides, (1996) Proceedings of CHEMPUTERS EUROPE III, Frankfurt.

[10] R. Phillips (2001) "Crystals, Defects and Microstructures" Cambridge University Press.

[11] V. B. Shenoy, R. Miller, E. B. Tadmor, D. Rodney, R. Phillips and M. Ortiz (1999) *J.Mechanics and Phys. Solids*, **47**.

[12] K. Xu and K. Prendergast (1994) *J. Comp. Phys.* **114** 9-17.

[13] K. Xu (2001) *J. Comp. Phys.* **171** 289-335.

[14] T. Shardlow and A. M. Stuart (2000) *SIAM J. Num. Anal.* **37** 1120-1137.

[15] A. Chorin, A. Kast and R. Kupferman (1998) *Proc. Natl. Acad. Sci. USA* **95** 4094-4098

[16] A. Chorin, O. Hald and R. Kupferman (2000) *Proc. Natl. Acad. Sci. USA* **97** 2968-2973

[17] O. Runborg, C. Theodoropoulos and I. G. Kevrekidis (2002) *Nonlinearity* **15** 491-511.

[18] A. G. Makeev, D. Maroudas and I. G. Kevrekidis (2002) *J. Chem. Phys.* **116**(23) 10083-10091.

[19] J. Li, D. Liao and S. Yip (1998) *Phys. Rev. E* **57** 7259-7267

[20] J. Li, D. Liao and S. Yip (1998) *Mat. Res. Soc. Symp. Proc.* **538** 473.

[21] J. Li, D. Liao and S. Yip (1999) *J. Comp.-Aided Mat. Des.* **6** 95-102

[22] C. W. Gear (2001) NEC Technical Report NEC-TR 2001-130

[23] I. Prigogine, R. Balescu, F. Henin and P. Resibois (1960) *Physica* **26** (suppl.) S36-S48.

[24] R. Balescu (1975) *Equilibrium and nonequilibrium statistical mechanics* John Wiley, NY.

[25] P. Gaspard, (1998) *Chaos, Scattering and Statistical Mechanics* Cambridge University Press, NY.

[26] J. L. Lebowitz, E. Presutti and H. Spohn (1988) *J. Stat. Phys.* **51** 841-862.

[27] J. L. Lebowitz (1993) *Physica* A **194** 1-27.

[28] S. Chapman and T. G. Cowling (1964) *The Mathematical Theory of Nonuniform Gases* Cambridge U. Press.

[29] W. G. Vincenti and C. H. Kruger Jr. (1965) *Introduction to Physical Gas Dynamics*, John Wiley, NY.

[30] J. P. Boon and S. Yip (1980) *Molecular Hydrodynamics* McGraw-Hill, NY.

[31] C. Theodoropoulos, S. Sankaranarayanan, S. Sundaresan and I.G. Kevrekidis (2001) *Proc. of the 3rd Pan-Hellenic Conference in Chemical Engineering* pp. 221-224.

[32] C. Theodoropoulos, S. Sankaranarayanan, S. Sundaresan and I.G. Kevrekidis (2002) *Phys. Rev. Lett.* submitted.

[33] A. G. Makeev, D. Maroudas, A. Z. Panagiotopoulos and I. G. Kevrekidis (2002) *J. Chem. Phys.* in press.

[34] FitzHugh, R.(1961) *Biophys. J.* **1** 445-466.

[35] Nagumo, J.S., Arimoto, S. and Yoshizawa, S. (1962) *Proc. IREE* **50** 2061-2070.

[36] Y. H. Qian and S. A. Orszag (1995) *J. Stat. Phys.* **81**, 237-253.

[37] G. M. Shroff and H. B. Keller (1993) *SIAM J. Numer. Anal.* **30**, 1099-1120.

[38] H. B. Keller, (1997) Lecture presented at the IMA (Dynamical Systems Program), Minneapolis; also, personal communication (IGK).

[39] H. Jarausch and W. Mackens (1987) *Numer. Math* **50** pp. 633-653.

[40] H. Jarausch and W. Mackens (1987) in *Large Scale Scientific Computing*, P. Deuflhard, B. Engquist, Progress in Scientific computing, **7** Birkhäuser Verlag.

[41] L.S. Tuckerman and D. Barkley (1999) in *IMA Volumes in Mathematics and its Applications* **119** 453-466.

[42] R. Temam (1988) *Infinite Dimensional Dynamical Systems in Mechanics and Physics* Springer Verlag, NY.

[43] P. Constantin, C. Foias, B. Nicolaenko and R. Temam (1988) *Integral Manifolds and Inertial Manifolds for Dissipative Partial Differential Equations*, Springer Verlag, NY.

[44] M. S. Jolly, I. G. Kevrekidis and E. S. Titi (1991) *Physica D* **44** 36.

[45] P. Constantin, I. G. Kevrekidis and E. S. Titi *in preparation.*

[46] H. von Sosen (1994) *The Recursive Projection Method applied to Differential Algebraic Equations and Incompressible Fluid Mechanics* Ph.D. Thesis, Part II, Caltech, Pasadena.

[47] K. Lust, D. Roose, A. Spence and A.R. Champneys (1997) *SIAM J. Sci. Comput.* **19**, 1188-1209.

[48] K. Lust (1997) *Numerical Bifurcation Analysis of Periodic Solutions of Partial Differential Equations*, Ph.D. Thesis, Leuven.

[49] K. Burrage, J. Erhel, B. Pohl and A. Williams (1998) *SIAM J. Sci. Comput.* **19** pp. 1245-1260.

[50] J. Erhel, K. Burrage and B. Pohl (1996) *J. Comp. Appl. Math.* **69** 303.

[51] B. D. Davidson (1997) *SIAM J. Num. Anal.* **34** 2008-2027.

[52] P. Love (1999) *Bifurcations in Kolmogorov and Taylor Vortex Flows*, Ph.D. Thesis, Caltech, Pasadena.

[53] E. D. Koronaki, A. G. Boudouvis and I. G. Kevrekidis (2002) *Comp. Chem. Eng.* submitted.

[54] L. R. Petzold (1983) in *Scientific Computing*, eds. R.S. Stepleman et al., North-Holland, 65-68.

[55] C. Siettos, M. D. Graham and I. G. Kevrekidis (2002) in preparation for *Phys. Rev. Lett.*

[56] Z. Toth and E. Kalnay (1993) *Bull. Am. Meteorol. Soc.* **72** 2317

[57] J. Patil, B. R. Hunt, E. Kalnay, J. A. Yorke and E. Ott *Phys. Rev. Lett.* **86**(26) 5878-5881.

[58] R. M. Lewis (1967) *J. Math. Phys.* **8** 1448-1459.

[59] A. N. Gorban, I. V. Karlin, P. Ilg and H. C. Oettinger (2001) *JNNFM* **96** 203-219.

[60] R. E. Kalman and R. S. Bucy (1961) *Trans. ASME Ser. D, J. Basic Eng.* **83** 95-107

[61] K. J. Astrom (1970) *Introduction to Stochastic Control Theory* Academic Press, NY.

[62] L. Ljung (1999) *System identification: theory for the user*, 2nd ed., Prentice Hall, NY.

[63] C. W. Gear and I. G. Kevrekidis (2001) NECI Research Report NEC-TR 2001-122, October 2001; also submitted to *J. Comp. Phys.*

[64] V. I. Lebedev (1997) in *Numerical Analysis and its Applications*, *Proc. WNAA 96, Bulgaria*, Vulkov et al. eds., Springer, Berlin, 274-283.

[65] A. L. Medovikov (1998) *BIT* **38**(2) 372-390

[66] J. G. Verwer (1996) *Appl. Num. Math.* **22** 359-379.

[67] M. Hochbruck, C. Lubich and H. Selhofer (1998) *SIAM J. Sci. Comp.* **19** 1552-1574.

[68] W. S. Edwards, L. S. Tuckerman, R. A. Friesner and D. C. Sorensen (1994) *J. Comp. Phys.* **110** 82-102.

[69] E. Gallopoulos and Y. Saad (1992) *SIAM J. Sci. Comp.* **13**, pp. 1236-1264.

[70] S. H. Lam and D. A. Goussis (1994) *Int. J. Chem. Kinetics* **26** 461-486.

[71] P.J. Holmes, J.L. Lumley and G. Berkooz (1996) *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge University Press.

[72] L. Sirovich and J. D. Rodriguez (1989) *Phys. Lett. A* **120** 211

[73] A.E. Deane, I. G. Kevrekidis, G. M. Karniadakis and S. Orszag (1991) *Phys. Fluids* **3** 2337.

[74] N. Aubry, P.J. Holmes, J.L. Lumley and E. Stone (1988) *J. Fluid Mech.* **192** 115.

[75] S. Shvartsman and I. G. Kevrekidis (1998) *AIChE J.* **44** 1579-1595.

[76] S. Y. Shvartsman, C. Theodoropoulos, I.G. Kevrekidis, R. Rico-Martínez, E. S. Titi and T. J. Mountziaris (2000) *J. Proc. Control* **10** 177-184.

[77] S. Yesilyurt and A. T. Patera (1995) *CMAME* **121** 231-257.

[78] G. Cybenko (1996) in *Identification, Adaptation and Learning: the Science of Learning Models from Data*, *NATO ASI Ser. F153* Springer, Berlin, 423-434.

[79] M. W. Braun, D. E. Rivera and A. Stenman (2001) *Int. J. Control* **74** 1708-1717.

[80] N. K. Sinha (2000) *Sadhana* **25** 75-83

[81] C. I. Siettos, C. C. Pantelides and I. G. Kevrekidis (2001) presented at the AIChE annual meeting, Reno, also *Ind. Eng. Chem.* in preparation.

[82] P. V. Kokotovic, H. K. Khalil and J. O'Reilly (1986) *Singular Perturbation Methods in Control: Analysis and Design* Academic Press, NY.

[83] A. M. Berezhkovskii and G. Sutmann (2002) *Phys. Rev. E* **65**(6) 060201.

[84] N. G. Hadjiconstantinou (2000) *Phys. Fluids* **12** 2634-2638.

[85] R. G. Larson (1999) *The structure and rheology of complex fluids* Oxford U. Press.

[86] J.K.C. Suen, Y.L. Joo, and R.C. Armstrong (2002) *Annu. Rev. Fluid Mech.* **34**.

[87] H. C. Oettinger (1996) *Stochastic Processes in Polymeric Fluids*, Springer, Berlin.

[88] M. Grosso, L. Russo, P. L. Maffetone and S. Crescitelli (2000) in *Proc. of COC 2000, St. Petersburg*, IEEE Publications, 561-564.

[89] C. T. Kelley (1995) *Iterative Methods for Linear and Nonlinear Equations* SIAM Publications, Philadelphia.

[90] H. C. Oettinger, B.H.A.A. van den Brule and M. A. Hulsen (1997) *JNNFM* **70** 255-261.

[91] M. A. Hulsen, A.P.G. van Heel and B.H.A.A. van den Brule (1997) *JNNFM* **70** 79.

[92] R. M. Jendrejack, J. J. de Pablo and M. D. Graham (2002) *JNNFM* in press

[93] M.P. Kennedy and M. J. Ogorzalek, guest eds. (1997) *IEEE Trans. Circ. Syst.* **44**(10).

[94] G. S. Heffelfinger and F. van Swol (1994) *J. Chem. Phys.* **100**(10) 7548-7552.

[95] G. S. Heffelfinger and D. M. Ford (1998) *Molecular Physics* **94**(4) 659-671.

[96] G. S. Heffelfinger and D. M. Ford (1998) *Molecular Physics* **94**(4) 673-683.

[97] A. P. Thompson, D. M. Ford and G. S. Heffelfinger (1998), *J. Chem. Phys.* **109**(15) 6406-6414.

[98] A. J. Roberts (2001) *Appl. Numer. Math.* **37** 371-396.

[99] F. F. Abraham, J. Q. Broughton, N. Bernstein and E. Kaxiras (1998) *Computers in Physics*, **12**, 538.

[100] S.M. Allen and J.W. Cahn, (1979) *Acta Metallurgica* **27**, 1085.

[101] C. de Boor (2001) *A Practical Guide to Splines*, revised ed., Springer, NY.

[102] M. Berger and J. Oliger (1984) *J. Comp. Phys.* **53** 484.

[103] S. T. O'Connell and P. A. Thompson (1995) *Phys. Rev. E* **52**(6) R5792-95.

[104] N. G. Hadjiconstantinou and A. T. Patera (1997) *Int. J. Mod. Phys. C* **8**(4) 967-976

[105] N. G. Hadjiconstantinou (1999) *J. Comp. Phys.* **154** 245-265.

[106] A. L. Garcia, J. B. Bell, W. Y. Crutchfield and B. J. Alder (1999) *J. Comp. Phys.* **154** p.134-155

[107] W. E and Z. Huang (2001) *Phys. Rev. Lett.* **87** 135501.

[108] Li Ju and I. G. Kevrekidis (2002) manuscript in preparation, see also http://long-march.mit.edu/Stuff/LANL.ppt

[109] W. E and B. Engquist (2002) submitted.

[110] K. Sankaranarayanan, X. Shan, I. G. Kevrekidis and S. Sundaresan (2002) *J. Fluid Mech.* **452** 61-96.

[111] K. Sankaranarayanan, I. G. Kevrekidis, S. Sundaresan, J. Lu and G. Tryggvason (2002) *Int. J. Multiphase Flow*, accepted for publication.

[112] C. I. Siettos, A. Armaou, A. G. Makeev and I. G. Kevrekidis (2002) *AIChE J.* submitted; also manuscript nlin.CG/0207017 (arXiv.org).

[113] D. G. Aronson, S. Betelu and I. G. Kevrekidis (2001) *PNAS* submitted; also manuscript nlin.AO/0111055 at arXiv.org.

[114] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden and K. Lust (2002) *Nonlinearity* submitted.

[115] C. W. Rowley and J. E. Marsden (2000) *Physica D* **142** 1-19.

[116] L.-Y. Chen and N. Goldenfeld (1995) *Phys. Rev. E* **51**(6) 5577-5581.

[117] C. W. Gear, S. Betelu, D. G. Aronson and I. G. Kevrekidis (2002) manuscript in preparation.

[118] M. P. Brenner, P. Constantin, L. P. Kadanoff, A. Schenkel and S. Venkataramani (1999) *Nonlinearity* **12**(4) 1071-1098.

[119] C. I. Siettos, I. G. Kevrekidis and P. G. Kevrekidis (2001) *Nonlinearity* submitted; also manuscript nlin.PS/0204030 at arXiv.org

[120] K. N. Christodoulou and L. E. Scriven (1998) *Quart. Appl. Math.* **9** 17.

[121] R. Rico-Martinez, I. G. Kevrekidis and K. Krischer (1995) in *Neural Networks in Chemical Engineering*, A. B. Bulsari ed. Elsevier, 409-442.

[122] R. Rico-Martinez, J. S. Anderson and I. G. Kevrekidis (1994) in *Proc. IEEE Workshop, Neural Networks for Signal Processing* IEEE Publ. vol.III, 596-605.

[123] J. C. Spall (1998) in *Johns Hopkins APL Technical Digest* **19**; also (2000) *IEEE Trans. Aut. Control.* **45** 1839-1853.

[124] F. A. Escobedo (1999) *J. Chem. Phys.* **110** 11999.

[125] D. A. Kofke (1993) *Molec. Phys.* **78**(6) 1331-1336

[126] L. I. Kioupis and E. J. Maginn (2002) *Fluid Phase Equilibria* in press.

[127] L. I. Kioupis, G. Arya and E. J. Maginn (2002) *Fluid Phase Equilibria* in press.

[128] C. Jarzynski (1997) *Phys. Rev. Lett.* **78**(14) 2690-2693.

[129] C. Bardos, F. Golse and N. J. Mauser (2000) *Methods Appl. Anal.* **7** 275-293.

[130] S. Jin and X. Li (2002) Preprint.

[131] R. A. Brown, L. E. Scriven and W. J. Silliman (1980) in *New Approaches to Nonlinear Problems in Dynamics*, P. J. Holmes ed., SIAM, Philadelphia.

[132] L. E. Scriven (1986) *J. W. Gibbs Lecture presented to the AMS*, New Orleans Meeting.

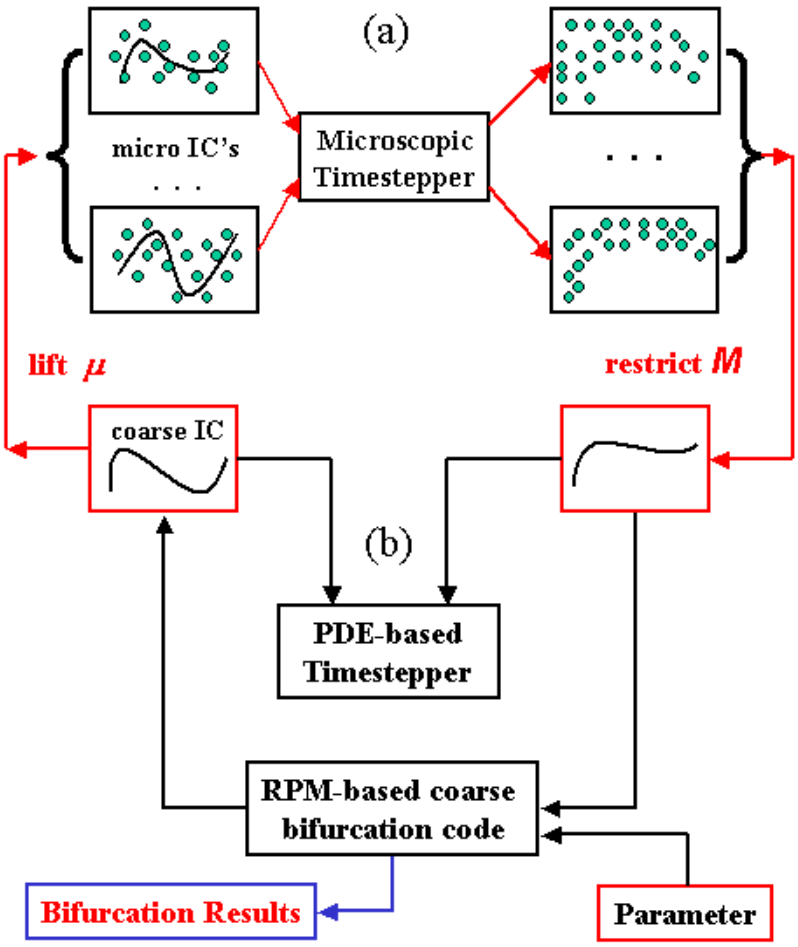


Figure 14: Schematic description of time-stepper based bifurcation analysis, fine and coarse. Notice the lifting of a macroscopic initial condition to an ensemble of consistent microscopic ones, as well as the restriction of the microscopic integration results back to the macroscopic (usually moments-based) description.

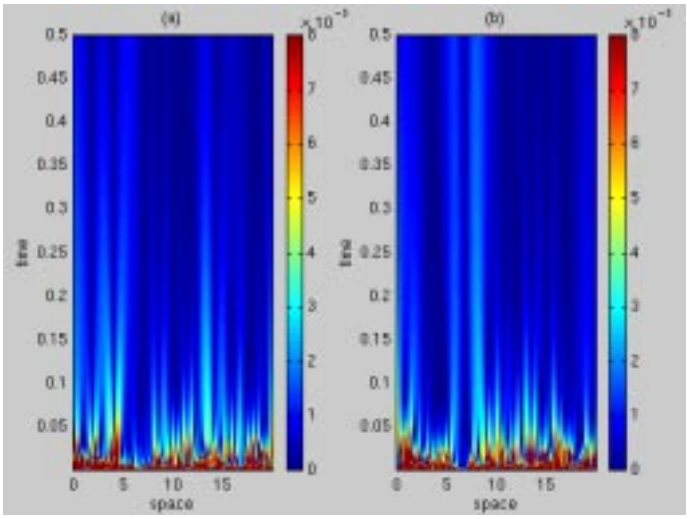


Figure 15: Decay ("healing") of the lifting error: comparison between the evolving restrictions of a "mature" LB-BGK trajectory, and two distinct liftings of its restriction: (a) a local equilibrium fine lifting and (b) a more "coarse" random lifting. Healing has visibly taken place at times much shorter than the timestepper reporting horizon (here 0.5).

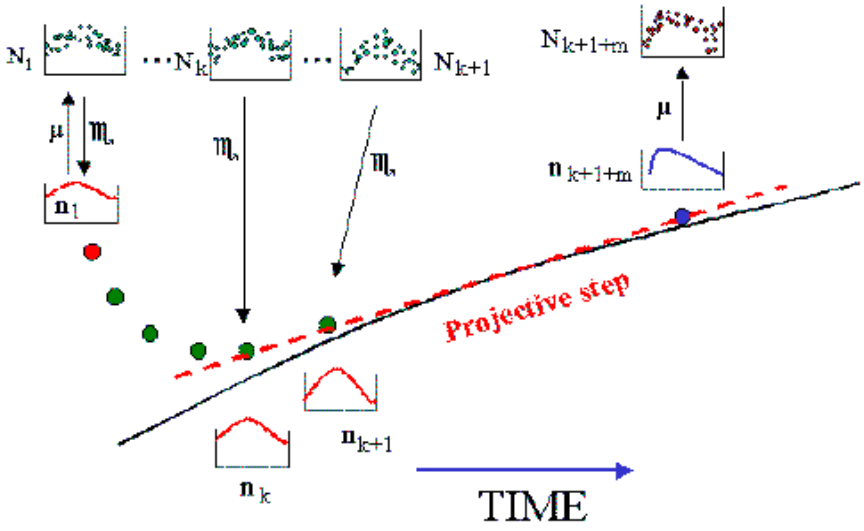


Figure 16: Schematic representation of an explicit projective integrator. Notice the lifting, the evolution with successive restrictions, the estimation of the "coarse time derivative", the projection step, followed be a new lifting.

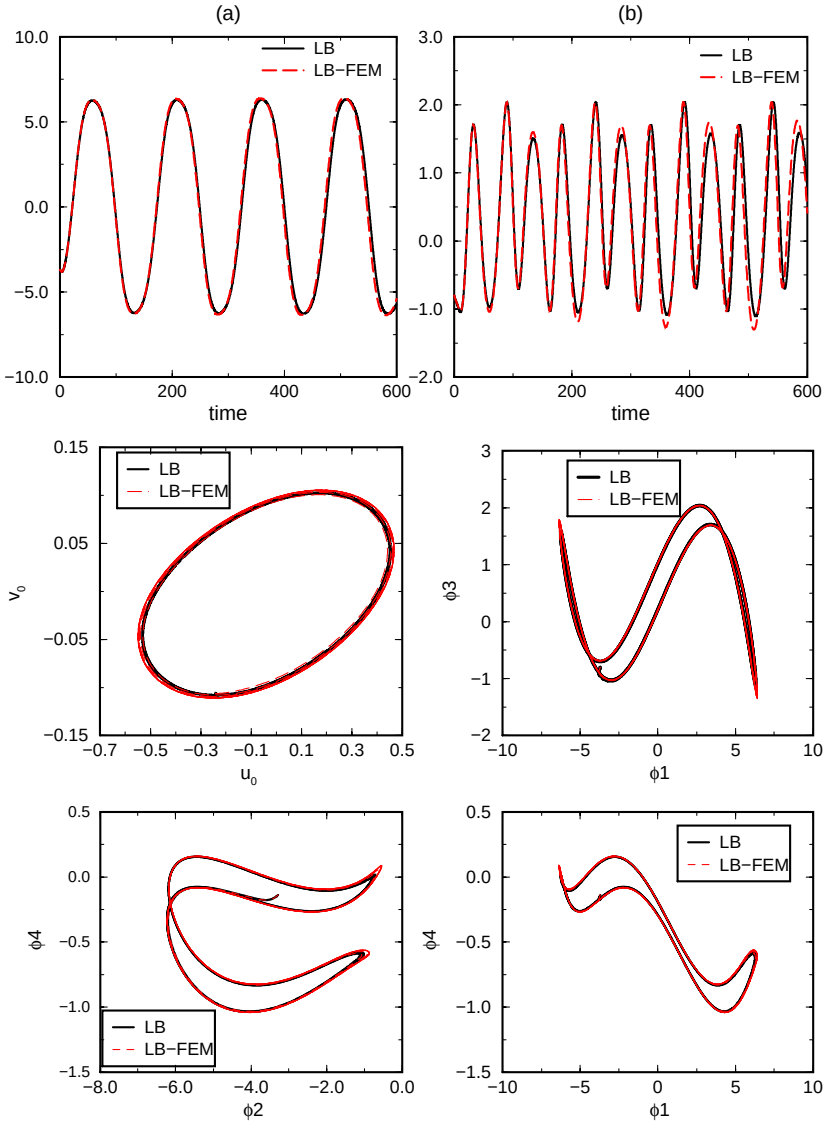


Figure 17: A comparison of FHN dynamics computed through (1) an LB-BGK code and (2) a 100-50-350 inner-LB - outer FEM (explicit Euler) projective integrator (from [4]). The comparisons are performed by projection on subspaces spanned by a few empirical orthogonal eigenfunctions (EOFs or PODs, see text). Top two panels: time series for (1, black) and (2, red) 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, black) and the 100-50-350 LB-FEM projective scheme (2,red).

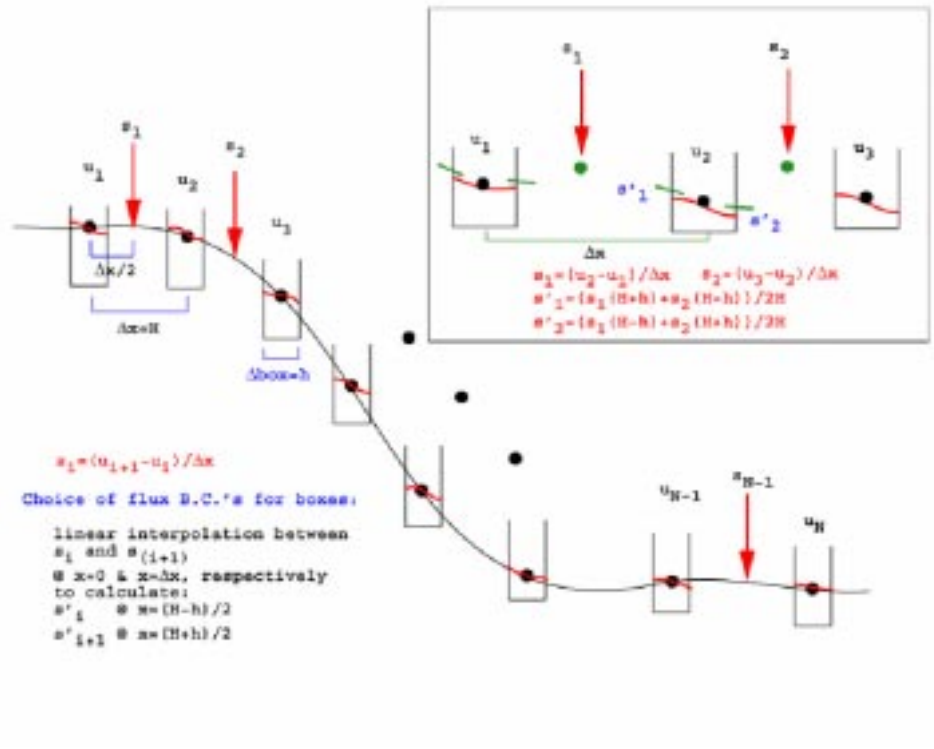


Figure 18: An original schematic description of the gap-tooth scheme (from the 2000 AIChE Area 10 plenary lecture, [5]); the notation is slightly different from the one in the paper. Notice the macro-mesh and the smoothly interpolated macro solution, the "teeth" surrounding the macro mesh, and the interpolated boundary conditions at the outer edges of the "teeth" or "patches". For this continuum description the buffer zones are not depicted.

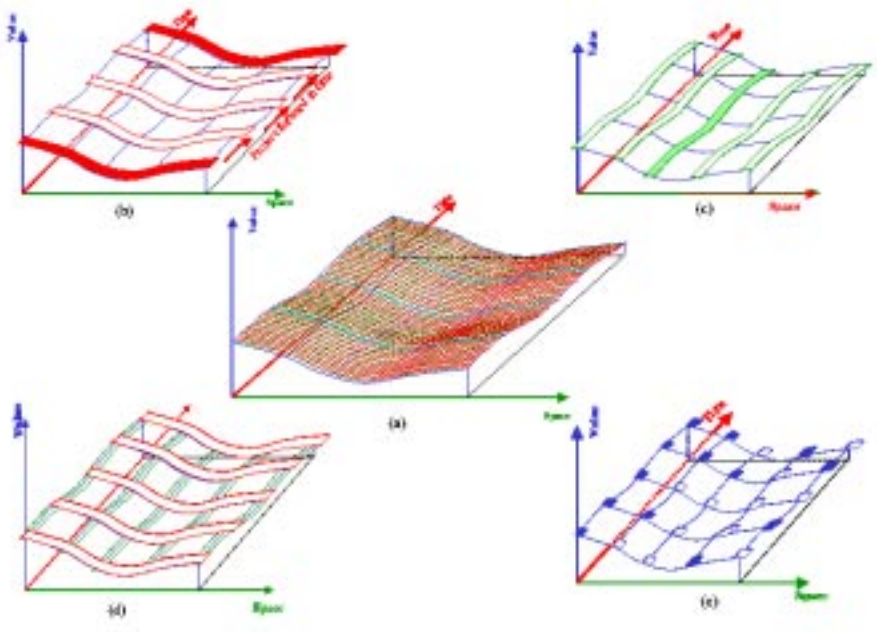


Figure 19: Schematics of coarse integration and patch dynamics techniques. Full microscopic simulation (a) requires impractically fine discretizations. Coarse integration (b) simulates in large space for short times, and then projects to longer times. The gap-tooth scheme (c) simulates in short spacial 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.

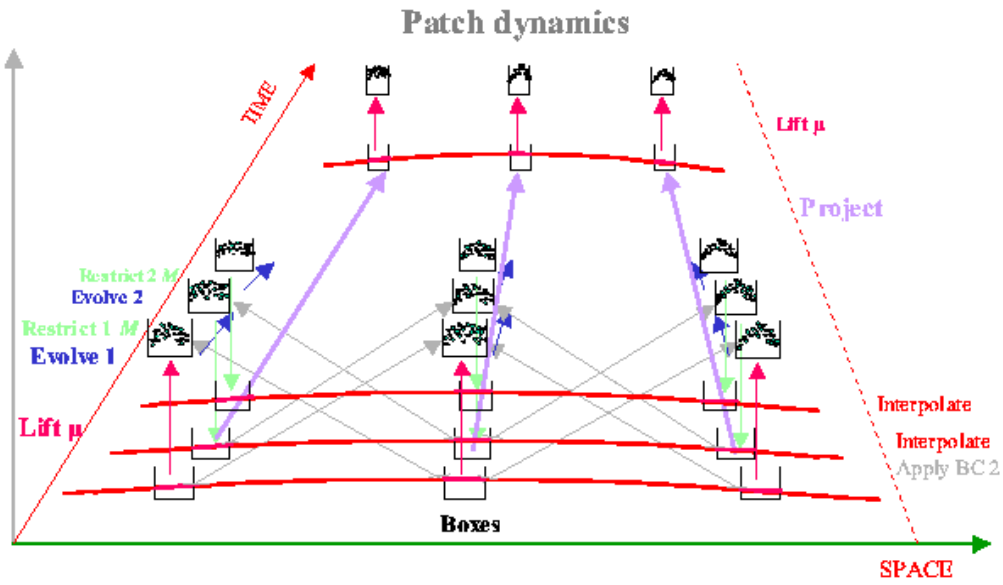


Figure 20: A schematic of the full patch dynamics simulation cycle, including lifting in patches, computation and imposition of patch BC, microscopic evolution, repeated restrictions and BC recomputations as necessary, and (after sufficient time has elapse so that the coarse time-derivatives can be estimated, a coarse projective step, followed by a new lifting.

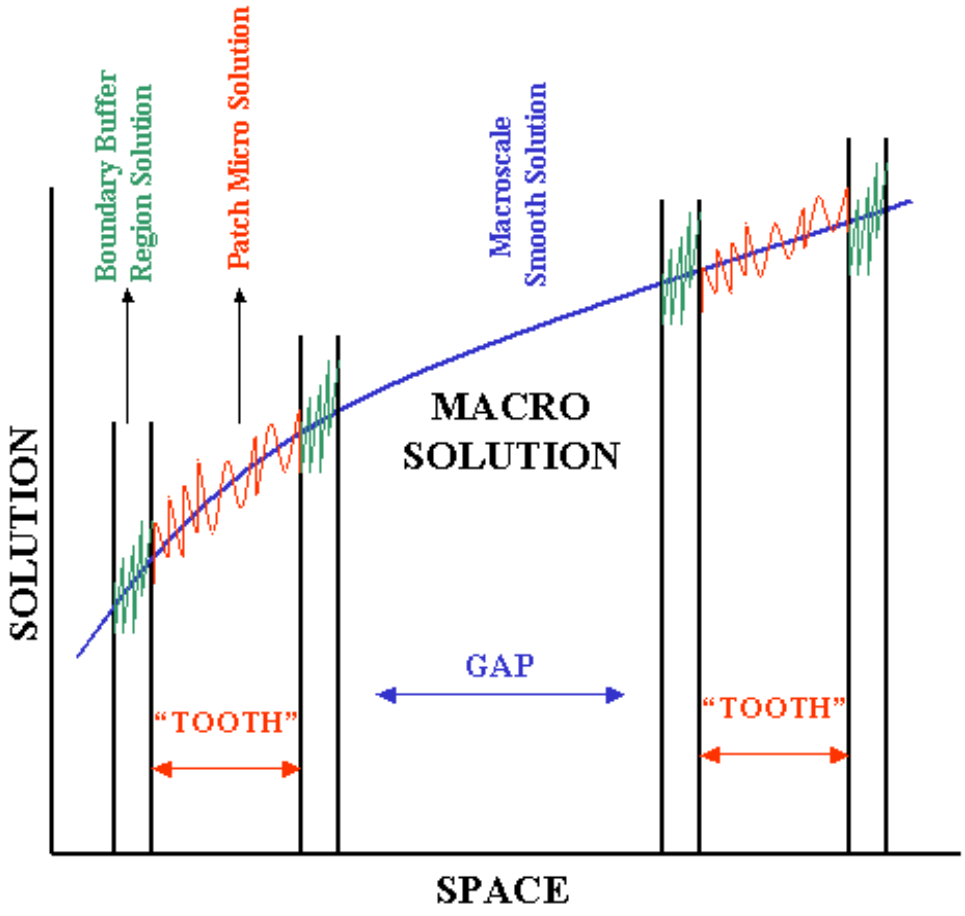


Figure 21: A schematic summary of the gap-tooth/patch dynamics nomenclature in 1D.

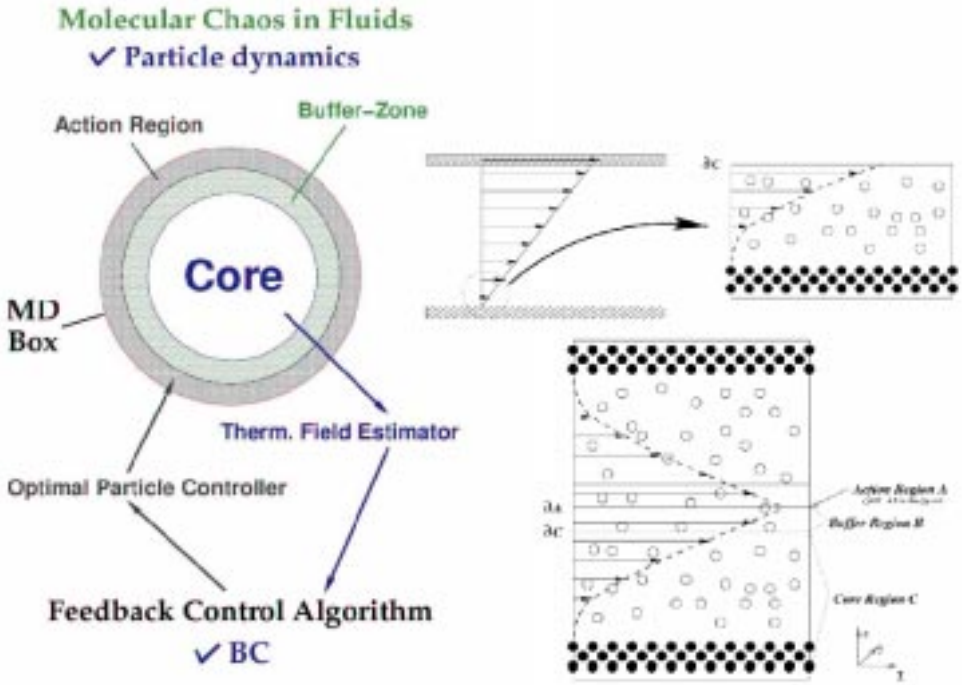
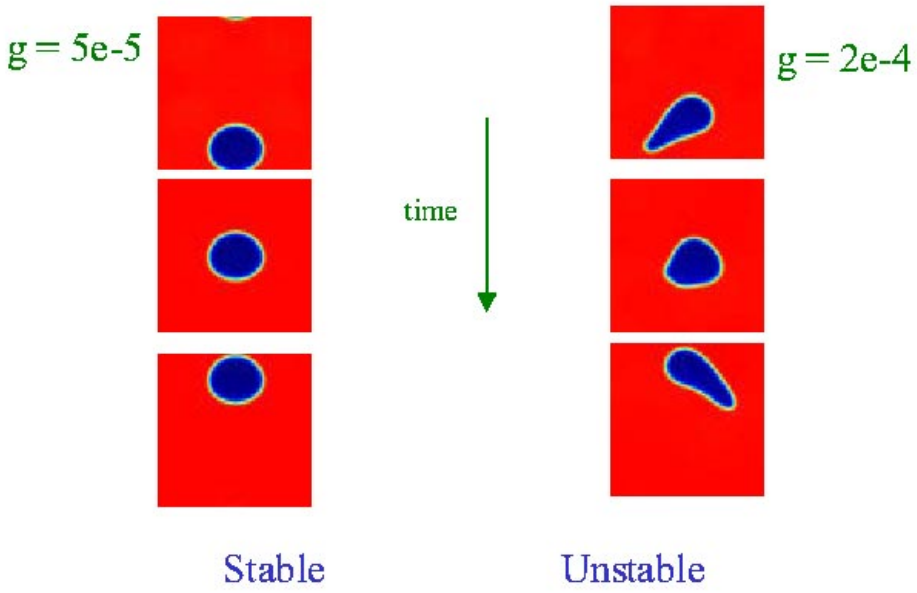


Figure 22: A schematic of the "extended boundary conditions" from [20] and a fluid flow application.



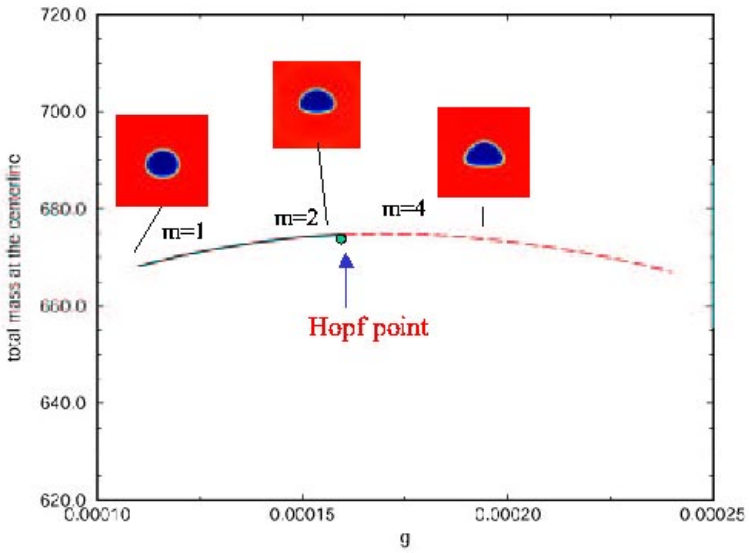


Figure 23: Coarse bifurcation diagram of a rising bubble flow instability (bottom) obtained with an inner LB-BGK simulator. The top panels show simulations of a steadily rising and an oscillating bubble before and after the bifurcation (from [32]). Solid(resp. broken): coarse stable(resp. unstable) steady rising bubbles. m is the dimension of the slow coarse subspace adaptively identified by RPM).

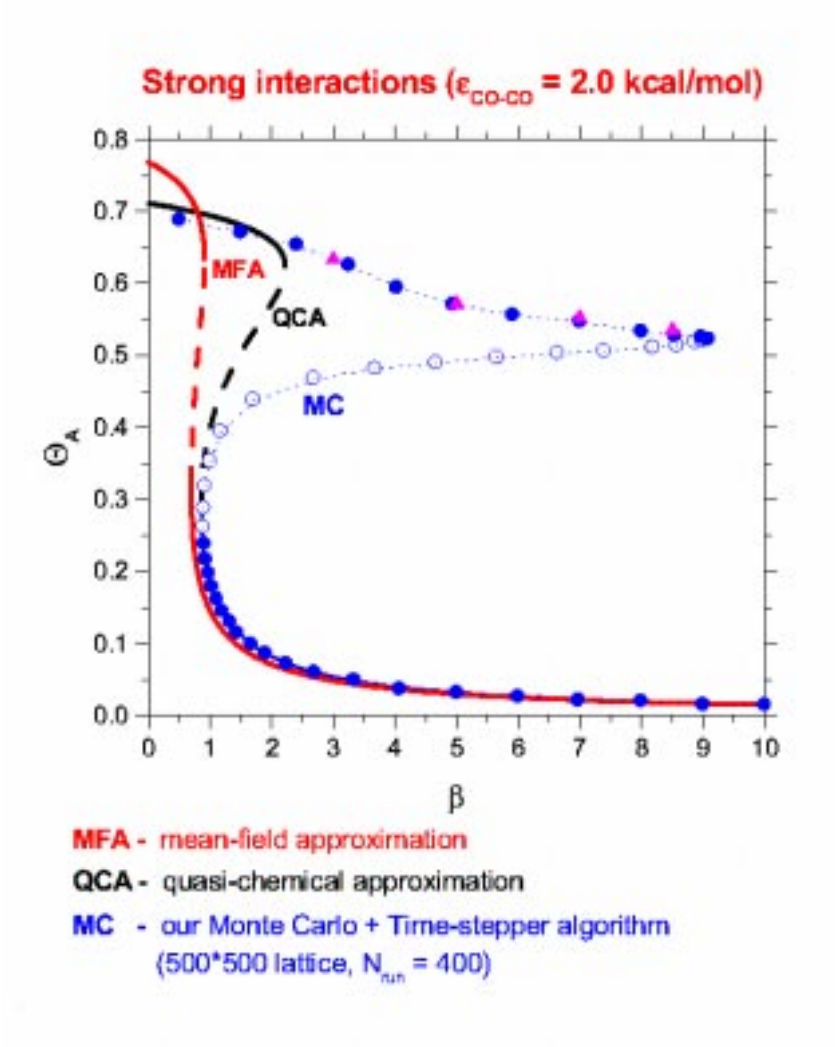


Figure 24: Coarse bifurcation diagram of a Kinetic Monte Carlo simulation of a model surface reaction (CO oxidation with repulsive adsorbate interactions). The diagram compares the bifurcation diagram obtained trough the coarse KMC timestepper with the mean field and the quasichemical bifurcation diagrams; the triangles are long-term KMC simulations (from [33]).

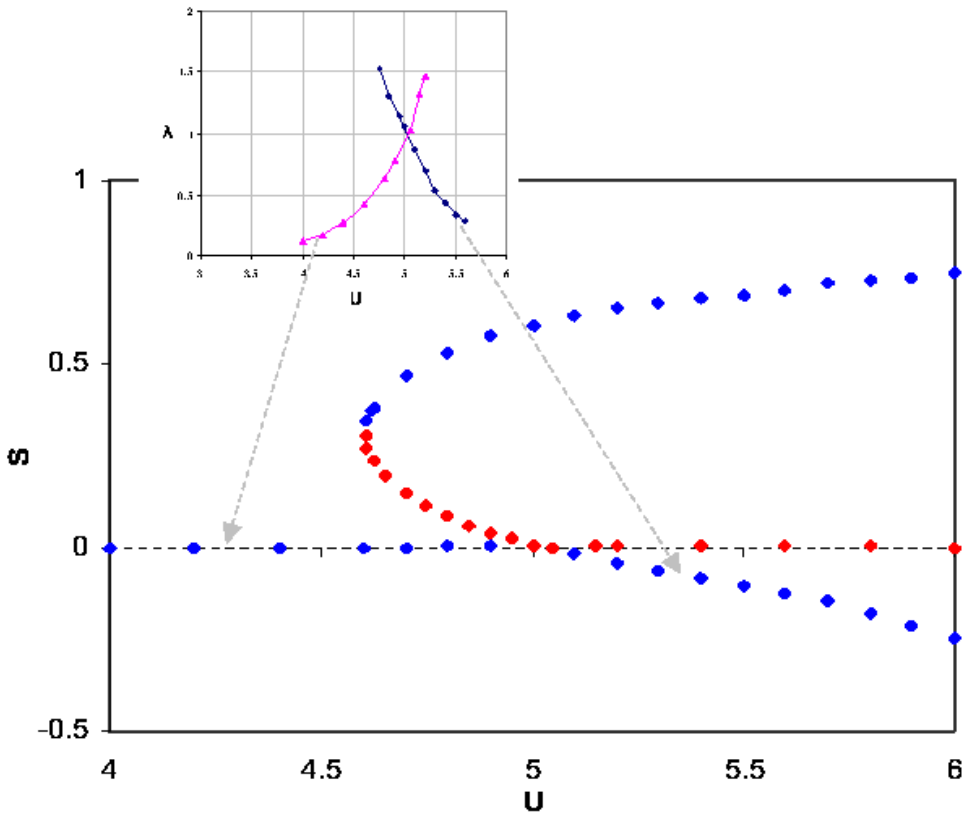


Figure 25: Coarse bifurcation diagram of a Brownian Dynamics simulation arising in the rigid rod model of nematic liquid crystals. Orientation distributions of rigid rods conditioned on one or two of their moments are used to construct the coarse timestepper with an inner stochastic Euler integrator. Stable and unstable branches of the order parameter S as a function of the rod density U are marked by color, and the corresponding coarse eigenvalues are also indicated (from [55]).

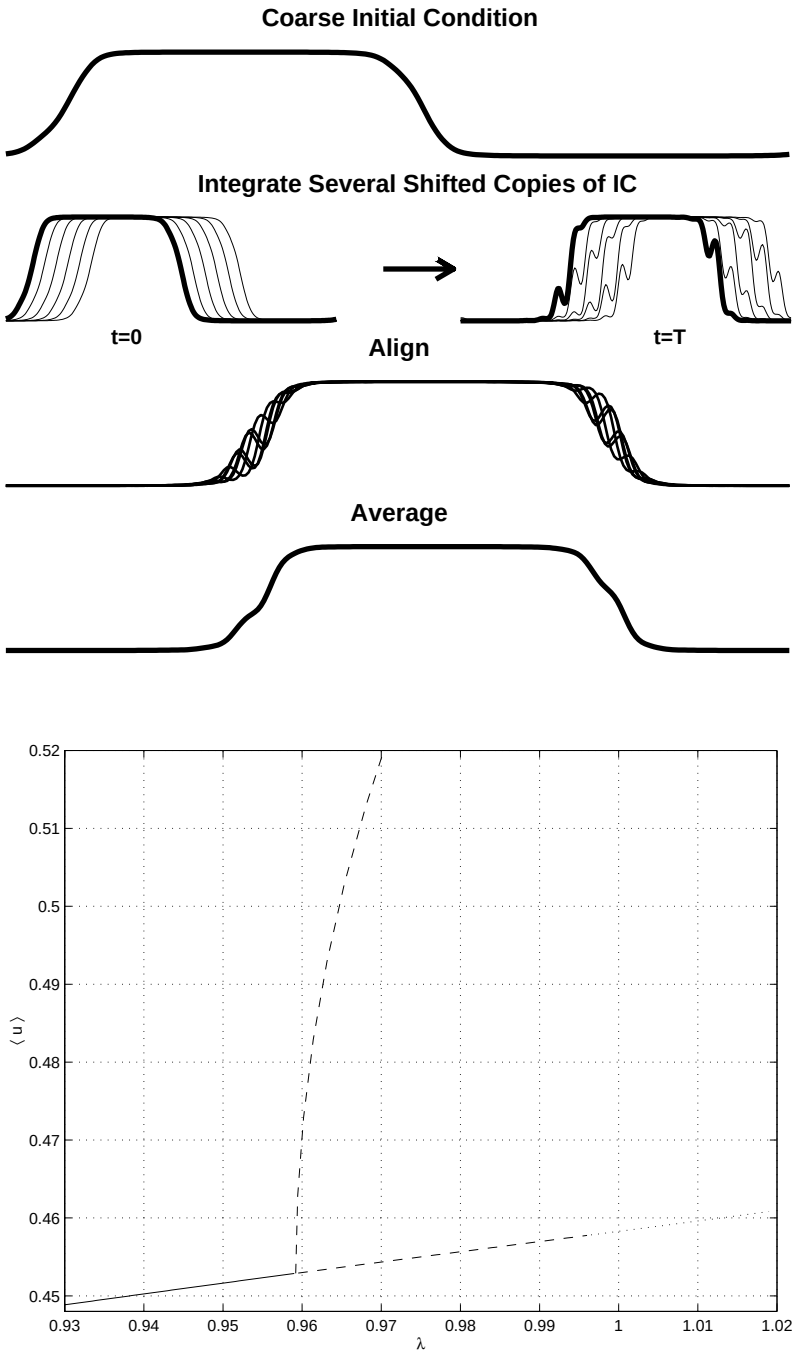


Figure 26: Schematic of the coarse time-stepper in a homogenization/effective equation analysis framework from [17]. The "effective bifurcation diagram" for a traveling reaction pulse through a periodic medium as a function of the medium periodicity is shown at the bottom, indicating a coarse instability and the birth of a modulated coarse traveling pulse.

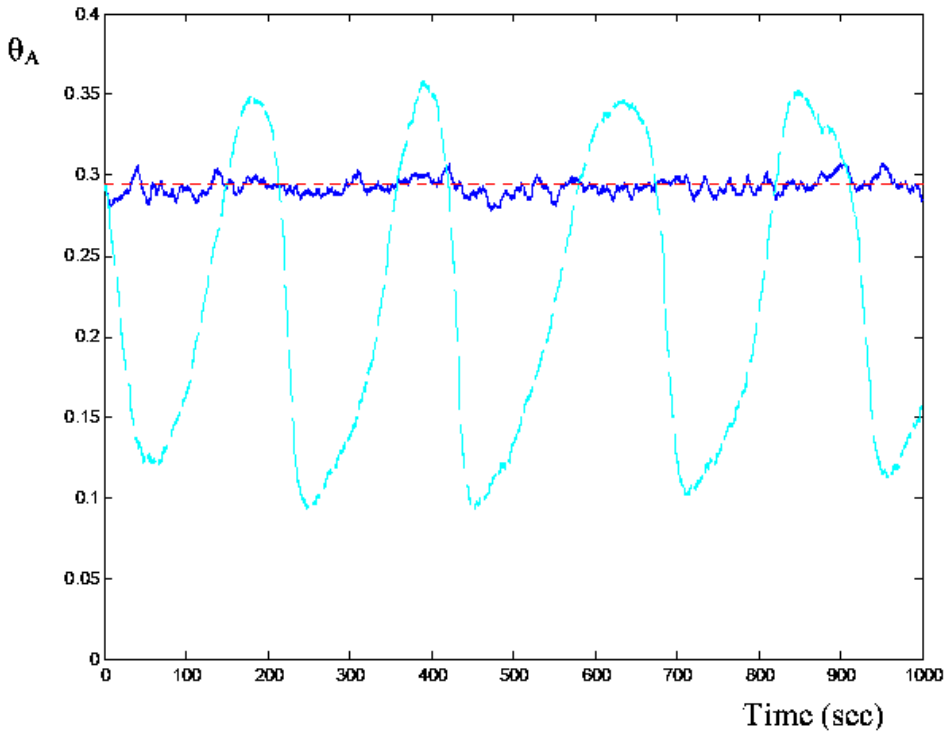


Figure 27: Coarse control of a KMC simulation of a surface reaction model [112]; the (coarse) unstable steady state to be stabilized was located, the estimator (a Kalman filter) and the controller (pole placement) constructed based on Newton-Raphson results obtained exploiting the coarse timestepper. The open loop behavior is a noisy oscillation (light blue), while the closed loop (using the bifurcation parameter as the actuator) clearly shows stabilization of the coarse steady state.

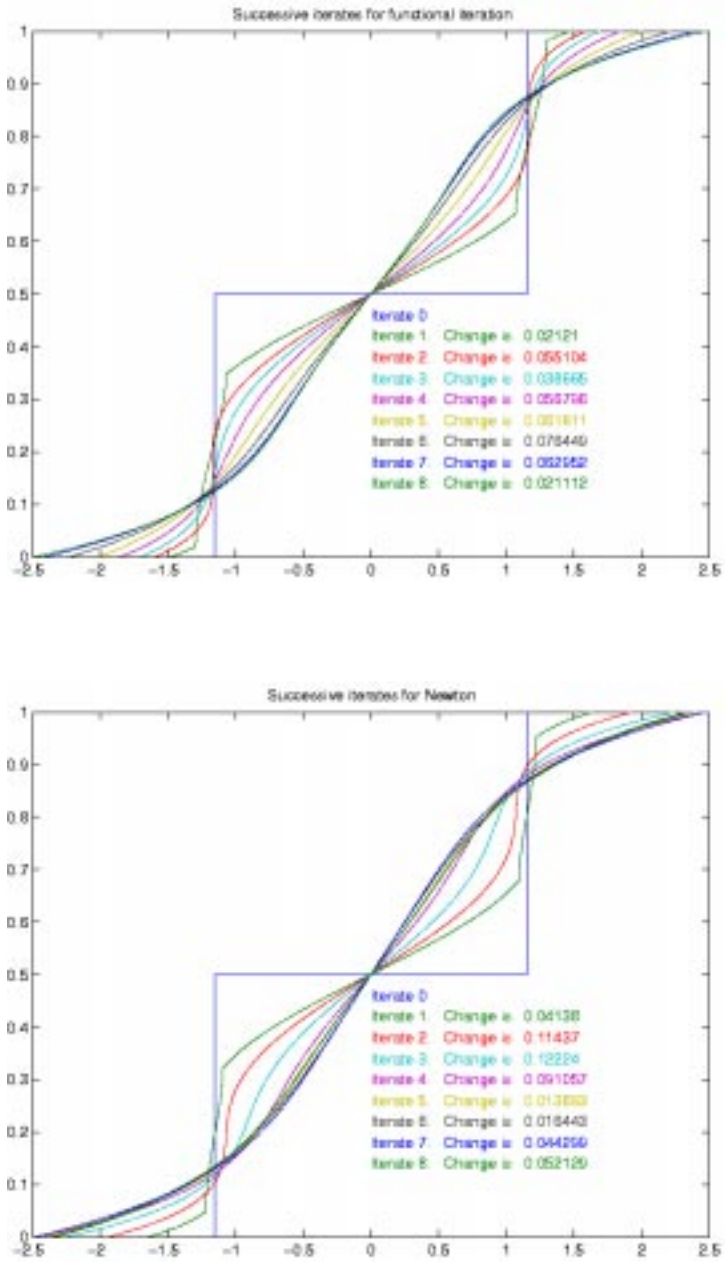


Figure 28: Coarse renormalization flow and coarse self-similar solution calculations based on KMC random walk simulations. The top panel shows the results of coarse renormalization flow projective integration for the one-dimensional diffusion equation, while the bottom panel shows iterations of a fixed 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.