"Coarse" Integration/Bifurcation Analysis via Microscopic Simulators: micro-Galerkin methods

C. W. Gear[1,2], Ioannis G. Kevrekidis[1] *
and Constantinos Theodoropoulos[1]
[1]Department of Chemical Engineering
Princeton University, Princeton, NJ 08544
[2]NEC Research Institute

Abstract

We present a time-stepper based approach to the "coarse" integration and stability/bifurcation analysis of distributed reacting system models. The methods we discuss are applicable to systems for which the traditional modeling approach through macroscopic evolution equations (usually partial differential equations, PDEs) is not possible *because the PDEs are not available in closed form.* If an alternative, microscopic (e.g. Monte Carlo or Lattice Boltzmann) description of the physics is available, we illustrate how this microscopic simulator can be enabled (through a computational superstructure) to perform certain integration and numerical bifurcation analysis tasks directly at the coarse, systems-level. This approach, when successful, can circumvent the derivation of accurate, closed form, macroscopic PDE descriptions of the system. The direct "systems level" analysis of microscopic process models, facilitated through such numerical "enabling technologies", may, if practical, advance our understanding and use of nonequilibrium systems.

1 Introduction

Textbook models of reaction and transport processes typically come in the form of conservation laws (mass, species, momentum, energy) closed through constitutive equations (e.g. the representation of viscous stresses for Newtonian fluids, or mass-action chemical kinetics expressions). In contemporary engineering modeling we have entered an era — ushered through materials modeling as well as systems biology modeling — where the time-honored macroscopic conservation equations are often not available any more. In this paper we will explore computational approaches that promise to bypass the derivation of such macroscopic equations, while still being able to deliver macroscopic level information.

to whom correspondence should be addressed, yannis@princeton.edu

Macroscopic conservation models often take the form of ordinary, or (for spatially distributed systems) partial differential equations (PDEs) in space and time. A representative reaction-diffusion problem that we will use as our illustrative example in this paper is described by the Fitzhugh-Nagumo PDE in one dimension [1, 2, 3].

$$u_t(\mathbf{x},t) = u_{xx}(\mathbf{x},t) + u(\mathbf{x},t) - u^3(\mathbf{x},t) - v(\mathbf{x},t) \tag{1}$$

$$v_t(\mathbf{x},t) = \delta v_{xx}(\mathbf{x},t) + \epsilon(u(\mathbf{x},t) - a_1 v(\mathbf{x},t) - a_0) \tag{2}$$

Here $u(\mathbf{x},t)$ and $v(\mathbf{x},t)$ 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. The *diffusion terms* and the *kinetic terms* in this system of coupled PDEs, representing distinct physical processes, are clearly recognizable. The use of such models for both analysis and design purposes then hinges on the exploitation of numerical analysis techniques to turn them (through discretization) into large sets of ordinary differential or differential-algebraic equations. A number of tasks (integration in time, direct solution of the steady state problem, linearization at steady state to determine stability) directly follow, and additional tasks, such as systematic parametric analysis, optimization and controller design, are also performed on the basis of these (discretized in space and time) models, possibly after further reduction. Extensive software packages (from general purpose scientific computing libraries such as Nag [4] or environments like MATLAB [5] to continuation/bifurcation codes like AUTO or LOCA [6, 7] and simulation/optimization environments like gPROMS [8]) have been and are constantly being developed, pushing the limits of what a user can *easily* obtain from such models. We collectively refer to these macroscopic evolution PDE models and their discretizations as "systems level models", and the tasks performed with such models, e.g. simulation (integration and steady state solution), continuation, stability and bifurcation analysis, controller design, optimization, as "systems level tasks".

An important point on macroscopic systems level modeling is the difference between what we refer to as "direct simulation" and "computer-assisted analysis". These colloquial terms can be interpreted in a number of ways, but here they are used in a particular sense: *Direct simulation* means setting parameters and initial conditions in a dynamic model, and then starting (and monitoring) time integration. In other words, *direct simulation* is the computational analog of an actual laboratory experiment: Setting the reactant concentrations in the reactor by *filling* the reactor (initial conditions), fixing the experimental conditions (setting parameter values, like valve settings for desired flow rates) and observing the system behavior (integrating) in time. Direct simulation is an important tool for the modeler, but not the only one, and not necessarily the "best possible" use of the model. The best computational approach depends on the task, i.e. the question asked. If, for example, steady states are sought experimentally, we are constrained to letting the experiment evolve to stationarity. Computationally we can perform the same series of tasks, i.e. integrate repeatedly until stationarity is reached. Alternatively, solving the steady state equations through a contraction mapping, such as a Newton-Raphson iteration, may prove much more efficient. Furthermore, if a limit of stability (e.g. a reactor ignition) is sought in parameter space, locating it, experimentally or computationally, through direct simulation is an extremely time consuming task. On the other hand, numerical bifurcation calculations, through the formulation of augmented systems —computational superstructures— around the (steady state form of the) same mathematical model, can be immensely more efficient for specific tasks, like the determination and continuation of stability boundaries. In summary, the extraction of certain types of information from the model can be performed through scientific computation (embodying applied mathematics considerations into and/or "around" the model) much more efficiently than through direct simulation. These latter *mathematics assisted* computational approaches will be referred to as "computer-assisted analysis". An early and lucid articulation of the power of such approaches can be found in ref. [10, 11].

Why might macroscopic equations not be available ? Two distinct possibilities may be responsible:

- coarse, systems level descriptions are simply not feasible (e.g. for small systems, say of nanometer size);
- systems level descriptions are *conceptually* possible, but we have no explicit *closed* equations at the system level (that is, the kinetics are not available as functions of just concentration, or the form of the viscous terms is not just a Newtonian or known non-Newtonian constitutive expression).

In these cases we often do have *correct* or approximately correct *microscopic* descriptions of the true, molecular level physics through "microscopic models", such as Molecular Dynamics (MD), Monte Carlo (MC), or kinetic theory based Lattice Boltzmann (LB) models. The macroscopic conservation equations for moments, like concentration, momentum, or energy, of the microscopic variable distributions are, however, *not available in closed form.*

If such a microscopic description is available, one could implement it in computer algorithms, and obtain results through direct simulation. The problem, in this case, is bridging the much-discussed gap between the time and space scales at which the microscopic models operate, and the macroscopic scales at which the "systems level" behavior is desired. It is clear that, for systems level *questions*, such as the location of a macroscopic steady state, a coarse macroscopic description (a set of PDEs) —if available in closed form— can be exploited to provide the best *computational answer.* If a system of PDEs describing transport and reaction in a microreactor accurately enough was available, then we should discretize this PDE system, and either integrate it in time, or, even better, perform a Newton-Raphson to solve for its steady states.

If, on the other hand, a PDE model is not available in closed form, but we do have microscopic models, alternative strategies have to be devised. Running the microscopic simulation for macroscopic space-time scales is (and will continue to be) inconceivable. Even if this were possible, it would still only be *direct simulation.* Practically no systematic ways exist today for implementing the "augmented" computational superstructures, discussed above (which are crucial for performing systems level tasks) around a microscopic (e.g. a molecular dynamics) code.

Traditionally, attacking such a problem involves (a) mathematical work, often enhanced by physically motivated considerations/assumptions, through, for example, multiscale analysis, to obtain closed meso/macroscopic continuum equations; and (b) using the systems level numerical tools discussed above to solve these continuum equations.

The alternative methodology we discuss in this paper aims at bypassing the first step of the traditional approach.

The idea is to perform certain systems level tasks, (typically performed through the continuum, macroscopic systems description) *directly* through the microscopic simulator. We will demonstrate that certain such systems level tasks (that we call "coarse integration" and "coarse bifurcation analysis") can indeed be performed at the systems level, without having to explicitly first derive a mesoscopic or macroscopic system description. The idea is to construct a different type of computational superstructure that will enable the microscopic simulator produce the information we would have needed from the meso/macroscopic description *had it been available in closed form.* It might be appropriate to refer to this approach as an "unavailable model motivated processing" of the microscopic simulations [12].

The remainder of this paper is organized as follows: in Section 2 we briefly discuss our recent results in the so-called time-stepper based *coarse* bifurcation analysis. This will illustrate our use of microscopic timesteppers and set the stage for Section 3, where "coarse integration" is briefly described, implemented and illustrated. In Section 4 we return to coarse bifurcation analysis using alternative macroscopic discretization techniques, thus completing what we will refer to as "micro-Galerkin" methods. We will then conclude with a discussion of current avenues of research, including the implications of coarse time-stepping for systems level control calculations.

2 TIME-STEPPER BASED COARSE BIFURCATION ANALYSIS

This section briefly outlines the approach we introduced in [3] and have also followed in our coarse bifurcation analysis of two-phase flow simulations [9]. It sets the stage for the introduction of "coarse integration" in Section 3, and will be revisited in our discussion and results of micro-Galerkin methods in Section 4. We start with a discussion of direct time-stepper based bifurcation analysis [13, 15] and proceed to its "coarse" modification.

Time-stepper based bifurcation analysis is a collective description of a class of numerical bifurcation techniques that stem from the following basic question: if we have a dynamic model of a process (a *time-stepper*) *i.e.* a subroutine which, given initial conditions returns the state of the system at the end of a time interval, can we use it to perform "system level" tasks like stability, continuation or bifurcation analysis ? Using the timestepper in the *traditional* way, *i.e.* taking the results of an integration and feeding them back as initial conditions for the next step, repeating the process, and thus following trajectories, is, as argued above, a difficult and sometimes impractical way of performing these systems level tasks.

A clear motivation for posing this question, that will resonate with large-scale industrial engineering modelers, is the existence of large legacy time-dependent simulation codes. These validated codes may contain the best available first principles model of the process and they represent man-years of effort. Thus, rewriting them from scratch to efficiently perform systems level tasks like computing operating diagrams is daunting. It makes sense then to seek to construct a computational superstructure, an intelligent *co-processing* (rather than *post-processing*) tool that will enable these codes to perform tasks they were not originally designed to do. We can think of this branch of numerical analysis as *an enabling technology for timesteppers*, and this was the motivation [14] of the paper [15] that constitutes the starting point of our work. Another original paper (where the term "adaptive condensation" was used for symmetric matrix problems) was published in 1987 [16], and the subject was relatively recently reviewed by Tuckerman and Barkley [17]. From the numerical analysis point of view, this work resonates with iterative large scale linear algebra and Newton-Krylov methods for solving nonlinear equations [18] as well as with Krylov integration methods [19, 21, 20]. This is a flourishing branch of numerical analysis research, which we do not review here, and to which our presentation may not do justice. In this section, we chose to present the procedure from the point of view of "smart utilization of legacy simulation codes"; we believe this makes a first description conceptually easier, and does bring out correctly the mathematical underpinnings and computational tasks involved.

Briefly and qualitatively, the starting point is the simple observation that a steady state of the set of coupled, nonlinear differential equations

$$u_t(\mathbf{x},t) = \mathcal{L}(u(\mathbf{x},t),\lambda)) \tag{3}$$

is also a fixed point

$$u^*(\mathbf{x}) = \varphi_\tau(u^*(\mathbf{x});\lambda) \tag{4}$$

of the time-τ map,

$$\begin{gathered} u^{\nu+1} = \varphi_\tau(u^\nu;\lambda) \\ (u^\nu \equiv u(\mathbf{x},t), u^{\nu+1} \equiv u(\mathbf{x},t+\tau)) \end{gathered} \tag{5}$$

the result of integration of equation 3 above with initial conditions $u(\mathbf{x},t)$ for time τ. Practically, we should think of equation 3 as the large set of coupled ordinary differential equations (ODEs) resulting from discretizing our macroscopic PDE description of the problem, and of equation 5 as the output of a subroutine that is used to accurately integrate the problem. Extensions of the discussion to systems of differential algebraic equations (DAEs) are also possible [30] but for simplicity we will deal here only with large systems of ODEs.

Suppose that we want to avoid writing "from scratch" a steady state solver (a contraction mapping like a Newton-Raphson iteration). We also want to avoid writing a direct solver of equation 3 whose Jacobian would involve variational integrations. Furthermore, suppose that the physical problem we want to solve has a clear separation of time scales. This rather loose expression translates to the key mathematical assumption about the qualitative nature of the (unknown) spectrum of the linearization

$$\left.\frac{\partial \mathcal{L}(u,\lambda)}{\partial u}\right|_{u=u^*} \tag{6}$$

of the dynamics around the steady state we want to find: there exist a few eigenvalues in a narrow strip around the imaginary axis, some are possibly unstable with positive real parts; then comes a *spectral gap*, and the rest of the eigenvalues are "far to the left" in the complex plane. In terms of the eigenvalues of the linearization of the timestepper,

$$\left.\frac{\partial \varphi_T(u,\lambda)}{\partial u}\right|_{u=u^*} \tag{7}$$

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. Eigenvalues at zero for equation 3 (at 1 for equation 4) are special, and can be dealt with. 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 [22, 23, 24], and many singularly perturbed systems that arise in engineering modeling.

In the so-called Recursive Projection Method (RPM) of Shroff and Keller [15] the user repeatedly calls the timestepper routine from several nearby initial conditions and for relatively short times. Under the loose assumptions described above, the results of these repeated calls can be used to adaptively approximate a low-dimensional subspace $\mathcal{P}$ of the linearization of the system along which time evolution is slowest, possibly slightly unstable. One then performs a contraction mapping (e.g. a Newton iteration based on a small approximate Jacobian) in this subspace, aided by the integration itself (Picard iteration for the timestepper mapping -equation 4) in its orthogonal complement $\mathcal{Q}$. This combination of (approximate) Newton iteration in the low-dimensional, slow subspace, and Picard iteration (integration) to provide a contraction in its complement, justifies the classification of such iterative methods as "Newton-Picard" methods. Such procedures, and their extensions, are being successfully used to perform continuation, stability and bifurcation analysis of dissipative PDEs close to low-codimension bifurcations [15, 25, 26, 27, 28, 29, 30].

The Coarse Time-Stepper: Suppose that we want to solve a problem described by partial differential equations of the type

$$u_t(\mathbf{x}, t) = \mathcal{L}(u(\mathbf{x}, t; \lambda)). \tag{8}$$

In particular, suppose that a computational code (a subroutine) exists that implements an accurate time-stepper for this PDE. That is, it takes as input initial conditions for the field(s) $u(\mathbf{x}, t = 0) = u_0(\mathbf{x})$, parameter values λ and a reporting time horizon T and produces as output an accurate approximation of the field(s)

$u(\mathbf{x}, T)$. The discussion above, in the case of certain dissipative problems, can be summarized as follows: it is possible to perform the bifurcation analysis of the discretized problem by writing a non-problem-specific computational superstructure, whose only information about the system comes from calls to the time-stepper subroutine. The Recursive Projection Method is one protocol for these calls, and for the processing that the superstructure performs for the resulting output. It is, in effect, the blueprint of such a computational superstructure. The work of Roose and coworkers [25, 31] for the Newton-Picard computation of limit cycles is another such blueprint.

In the case of the Fitzhugh-Nagumo PDE in one spatial dimension, the fields of interest are the average local concentrations of the two chemical species involved: the activator, $u(\mathbf{x}, t)$ and the inhibitor, $v(\mathbf{x}, t)$. The time-stepper of the PDE (obtained, for example, through a finite difference, finite element or spectral discretization in space, and the method of lines in time) will take as input a finite vector containing a truncated description of the fields at $t = t_0$ (e.g. the values u_i, v_i, $i = 1, ...N$, N being the number of discretization points) as well as parameter values, and will produce as output the same truncated description of the fields at $t = T$.

In chemical engineering modeling practice, we are used to this level of description of reaction and transport. We do not constantly have to remind ourselves that what really occurs is molecular motion and collisions; that the unknowns (or "variables") of the problem really are the positions and velocities of a distribution of molecules. We are used to the coarse description, where we can write deterministic equations for moments of the distribution of molecules of reacting species. A typical such moment is the local concentration, the zeroth moment of the distribution of molecules integrated over all possible velocities, and averaged over some microscopic space scale. It is remarkable (and the fact that it was empirically known does not make it less so) that equations describing the evolution in time of a few moments of the distribution, e.g. concentration, the zeroth moment, *close* using only the same few moments themselves. Indeed, concentration trajectories in time can be followed using chemical kinetics expressions that depend only on concentrations; only the zeroth moment, and not the entire distribution. These expressions (as also the expressions for diffusion in terms of the Laplacian of the zeroth moment field) are certainly approximate - but they have been so succesful over such a long time and in so many problems, that we have come to sometimes regard *them* as first principles, and not as the coarse, closed approximations of the detailed problem that they really are. The branch of chemical engineering probably most aware of this closure problem, and the need of a dialogue between molecular models and macroscopic conservation equations, has tradititionally been rheology [32, 33].

Suppose now that a problem arises, for which a microscopic, molecular level description and a code that implements it are available. This code evolves the entire molecular distribution in time and we call it the *fine timestepper.* It processes a more detailed description of the system than the coarse one. The coarse description — typically the first few moments of the fine description — is, in some sense, a projection, or map, of the fine description. The fine timestepper can be a molecular dynamics code, or a Monte Carlo code, or a kinetic theory based Lattice Boltzmann scheme.

Such an LB scheme will constitute our example in this work. Suppose that we also know, or have reason to believe, that, for this model, a macroscopic description in terms of moments of the molecular distributions (e.g. reaction-diffusion equations in terms of concentration fields) conceptually exists. That is, we believe it is possible to write deterministic equations for the evolution in time of concentration fields, but that the right hand sides of these equations are not explicitly available. This would occur, for example, if mass action kinetics was inaccurate in describing the reaction rate. Can we analyze these equations without first obtaining them explicitly?

We argued above that this can be done if we are given a subroutine implementing a timestepper of the PDEs (the coarse, closed moment equations). We will now argue that, in the absence of a coarse, PDE-level moment description, the timestepper of the (unavailable) moment equations can be approximated through the microscopic simulator. The procedure for implementing this coarse timestepper involves the following steps:

- Prescribe a coarse initial condition (e.g. a concentration profile), say $\mathbf{u}(t_0)$.
- Transform (we will use the word "lift" and the symbol μ) this initial condition into one (or more) fine, *consistent* microscopic realizations of it $\mathbf{U} = \mu(\mathbf{u})$; that is, create microscopic distributions conditioned on the coarse initial profile (i.e. prescribed values of some of their lower moments).
- Evolve this (these) realization(s) using the microscopic evolution code (the fine time-stepper, Φ_T) for the desired amount of macroscopic time T.
- Project (we will use the word "restrict" and the symbol $\mathcal{M}$) the results back to moment space, appropriately averaging over fine space and/or fine time and/or number of realizations $\mathbf{u} = \mathcal{M}(\mathbf{U})$.

This constitutes the "coarse timestepper", or "coarse time-T map". If this map is accurate enough, it is immediately obvious (and it was demonstrated in [3, 9]) that a computational superstructure like RPM can be "wrapped around" it and enable it to perform the time-stepper based bifurcation analysis of the coarse description of the problem. Through this "lift-run-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 procedure is summarized in Fig. 1: If a macroscopic PDE *is* available, we can perform its bifurcation analysis using its timestepper and RPM around it. If a macroscopic PDE *is not* available, we substitute the discretized PDE timestepper with the coarse timestepper, and use exactly the same code around it, to analyze the, *unavailable in explicit form*, evolution equation for the coarse, moments-level, description of the problem.

We illustrate this procedure by repeating a one-parameter bifurcation analysis of the Fitzhugh-Nagumo PDE in one dimension. Here the PDE represents the "coarse" description. The "fine" description is a Lattice-Botzmann BGK (LB-BGK) one, where we do not evolve overall concentrations in space, but probability densities of kinetic entities ("particles", "molecules") distributed over a discrete lattice (in space) and

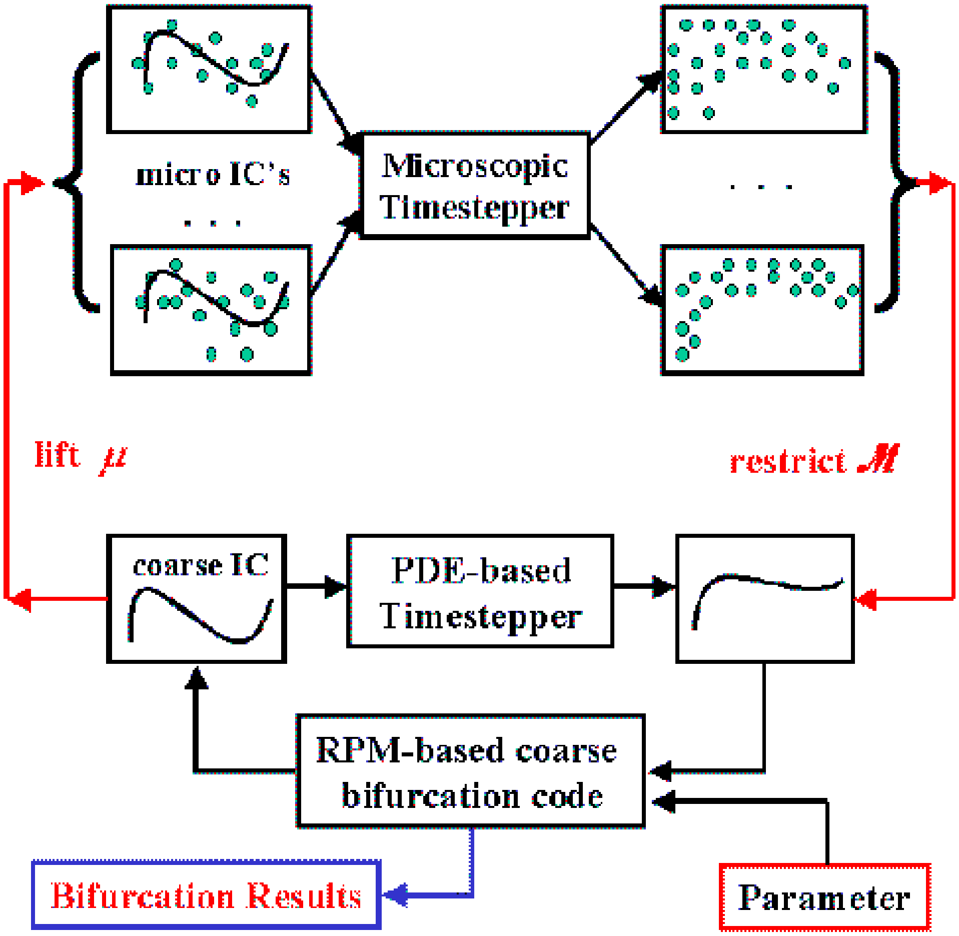

Figure 1: Schematic description of time-stepper based bifurcation analysis, fine and coarse. Notice the lifting of a macroscopic initial condition to an ensemble of consistent microscopic ones, as well as the restriction of the microscopic integration results back to the macroscopic (usually moments-based) description.

over a discrete number of velocities (for the 1-D problem just three: left-moving, right-moving and non-moving). The evolution rules for the LB-BGK model [34, 35] we have used are as follows:

$$
\begin{aligned}
N_i^u(x+c_i, t+1) - N_i^u(x,t) = -\omega^u[N_i^u(x,t) - N_i^{u,e}(x,t)] \\
+R_i^u(N_j^u, N_k^v)
\end{aligned} \tag{9}
$$

(and correspondingly for v). Here, $N_i^u(x,t)$ is the population density of activator (u) particles at position x on the lattice at time t, with velocity c_i. In this one-dimensional system, particles can only move towards the two available adjacent sites (N_2, N_3) or (allowing for rest particles) stay in place (N_1). For unitary LB temporal and spatial increments, therefore, $c_i = \{0, 1, -1\}$ and $i = 1, 2, 3$ respectively. $N_i^{u,e}$ is the local equilibrium u population, homogeneous in all velocity directions, and $\omega^{u(v)}$ is the $u(v)$ BGK relaxation parameter. The first part of the right hand side computes post-collision populations. R_i^u is the reaction term; we use the (strong) assumption [35] of local diffusive equilibrium for the reaction term. The local (in space) concentrations can be *uniquely* computed as the 0-th moments of the populations, e.g.

$$
u(x,t) = \sum_{i=1}^{3} N_i^u(x,t) \text{ or } \mathbf{u} = \mathcal{M}(\mathbf{N}) \tag{10}
$$

The opposite is, however, not true. If a "coarse" initial condition (concentrations, zeroth moments) is specified, *any* 3 random numbers (weights) w_i summing up to 1 would be a possible choice, i.e.

$$
N_i^u(x,t) = w_i u(x,t) \text{ or } \mathbf{N} = \mu(\mathbf{u}) \tag{11}
$$

This example has been chosen because the evolution of the moments (over velocities) of the LB-BGK procedure (equation 9) asymptotically approximate the evolution of the u and v fields for the FHN PDE [3]. We chose to work in a regime where the coarse description exhibits intense nonlinear effects, so that we can test how our coarse timestepper procedure captures these effects. The phenomena of interest include (a) sharply spatially varying, front shaped steady states; (b) steady state multiplicity through turning point bifurcations; and (c) spatiotemporal oscillations via a Hopf bifurcation. The values of the kinetic parametes used were $a_0 = 0.03$ and $a_1 = 2.0$; the diffusion coefficient was $\delta = 4.0$ and the time-scale ratio ϵ was our bifurcation parameter.

Fig. 2 depicts a comparison of the nonlinear spatiotemporal behavior of the FHN model in the regimes of interest, obtained through (a) the discretized PDE directly using the Galerkin Finite Element (FEM) in 101 nodal points; and (b) the microscopic LB timestepper and the coarse RPM-LB scheme. Figures 2a and b show the spatiotemporal behavior of $u(\mathbf{x}, t)$ on a limit cycle ($\epsilon = 0.01$ in equation 2) computed through the FEM and the LB timesteppers, respectively.

Branches of steady state solutions in the regime of the turning point and the Hopf bifurcation are depicted in Figures 3a and 3b, respectively. The solid (broken) lines

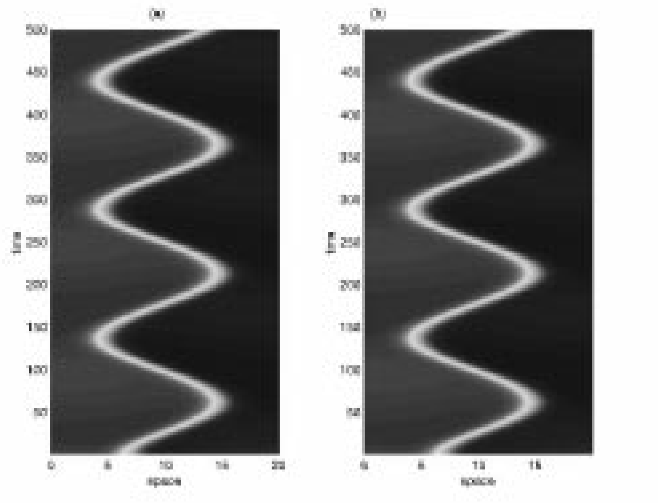

Figure 2: Space-time plot of a limit cycle of the Fitzhugh-Nagumo equation, showing the back-and-forth motion of the unstable front. The figure compares the u-component solution of the PDE computed (a) through a finite element discretization and (b) the projection of the Lattice-Boltzmann BGK solution onto the corresponding moment.

represent stable (unstable) steady states obtained from the solution of the (steady state form of the) FEM-discretized PDEs. Pseudo-arclength continuation was used to follow the branch past the turning point. The filled (open) circles represent stable (unstable) steady state solutions obtained from the coarse RPM-LB scheme. It is noteworthy that employing RPM around the microscopic timestepper we were able to calculate both stable *and unstable* coarse steady states and locate the bifurcation points by performing eigenvalue analysis on the small approximate Jacobian of the $\mathcal{P}$ subspace. The inset in Fig. 3b depicts a representative steady state from solution of the PDE (solid lines) and RPM-LB (broken lines).

In order to make a meaningful comparison between the results from the solution of the PDEs and those obtained from the microscopic timestepper, it is important to be able to translate macroscopic parameters (for the PDE evolution) into microscopic ones (for the LB-BGK evolution).

The most important step in the entire procedure is the lifting. In all the work described above, the fine description, in which the time evolution was performed, involved 400 lattice sites, two species, and three velocity bins per site, a total of 2400 unknowns. The coarse description (in which the RPM and the stability analysis was performed) involved 800 (2×400) finite element coefficients. Clearly, indeterminacy is generated (information has to be injected) every time we lift from 800 to 2400 numbers (see equation 11): how do we choose how to initialize all of these additional degrees of freedom ? The answer is that, under certain conditions, it does not matter. We can initialize these extra degrees of freedom (the higher moments of our

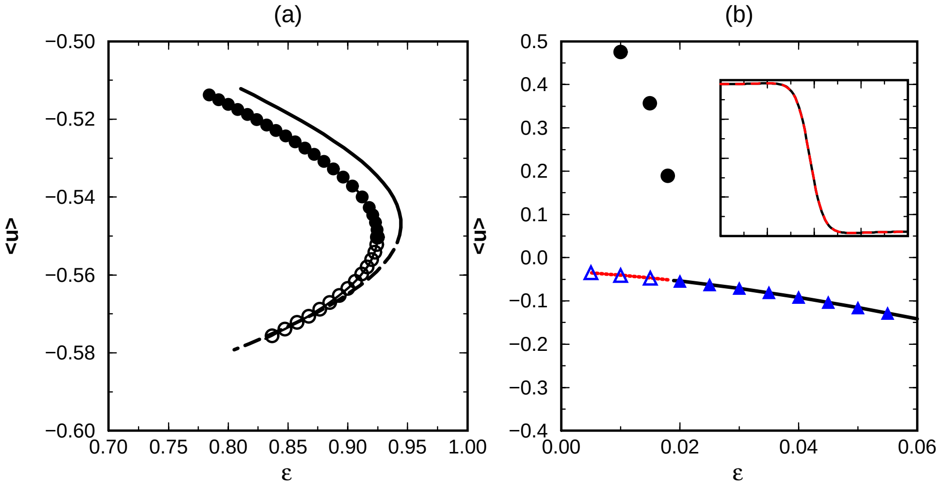

Figure 3: Comparison of the FHN bifurcation diagrams showing: (a) A turning point copmuted through FEM discretization (solid line:stable branch, broken line:unstable branch) and LB-RPM (filled circles stable and open circles unstable branch). (b) A Hopf bifurcation. Solid (broken) line: FEM computed stable (unstable) branch. Filled (open) triangles: LB-RPM computed stable (unstable) branch. Filled circles depict the amplitude of corresponding limit cycles. In the inset: comparison of a representative steady state u-profile computed through FEM (red solid line) and LB-RPM (black dashed line).

distribution) any way we want. The reason is that we expect that, very quickly, over the initial stages of our integration towards the timestepper reporting horizon, these higher order correlations will relax to functionals of the lower order ones (the ones we prescribed, the ones we conditioned our initial distribution on). For our particular example this is illustrated both for *fine* (Fig. 4a) and *coarse* (locally space-averaged, Fig. 4b) initializations at $\epsilon = 0.01$, taking as initial condition (I.C.) the coarse steady state at a different parameter value ($\epsilon = 0.1$). Figure 4a shows the difference in the moments, $u(\mathbf{x}, t) - u'(\mathbf{x}, t)$, of two spatiotemporal evolutions: (a) a microscopic profile encountered in the simulation, giving rise to $u'(\mathbf{x}, t)$; and (b) what we will call a "fine" local equilibrium initialization consistent with $u'(\mathbf{x}, 0)$: triads of equal weights ($w_i = 1/3$, $i = 1, 2, 3$ in equation 11), were chosen to consistently lift the density profile $u'(\mathbf{x}, 0)$ to Lattice Boltzmann space. In Fig. 4b the difference between $u(\mathbf{x}, t)$ and a *coarse* random lifting (initialization) is depicted. For the coarse initialization, starting from the same density profile, ($\epsilon = 0.1$), locally averaged values of u and v ($\overline{u}$ and $\overline{v}$) every 4 consecutive lattice points were calculated. These averages were used to initialize (in a "step" fashion) the corresponding lattice site populations through random positive triads of weights (a different triad for every lattice point, each summing up to 1). In both cases the absolute difference in the evolution $u - u'$ coming from different initializations drops below 2×10^{-3} after time T=0.1. Practically, the discrepancy between the projections on the zeroth moments of different trajectories, caused by the different initializations, has decayed after times much shorter than the RPM-LB reporting horizon (after $t \approx 0.1$ as compared to $T = 15 - 25$).

For this problem, no matter how the initialization is performed, as long as the zeroth moment coarse fields are the same, the coarse result of the integration over a "mesoscopic" time scale is (more or less) the same. If necessary, averaging over an *ensemble* of microscopic simulations, conditioned to have the same coarse initial condition, can help with variance reduction.

There are too many caveats and too many conditions in the above discussion. We have presented how it works, but did not quantify under what conditions it is expected to work, or even proven to work. Discussions of this type are found at the basis of the statistical mechanical arguments underlying the derivation of chemical kinetics. They also lie at the basis of Chapman-Enskog expansions [36] closing the equations for the moments of the Boltzmann equation to yield the Navier-Stokes equations. We only state here, very qualitatively, that the assumptions under which our "coarse time-stepper" makes sense are part of the assumptions that underpin the existence of a deterministic description at a coarse level (closed evolution equations for a few moments of a distribution). There is a number of additional points we will return to in the Discussion section. For the moment, we have shown that the coarse time-stepper, if it exists, is a tool that enables a microscopic simulator (acting on a fine problem description) to analyze the same problem at a coarse level, where a description is not available in closed form. This computer assisted approach, working across scales, has enabled a microscopic simulator to perform systems level analysis tasks that it was not designed for, and that, in principle, were inaccessible to it.

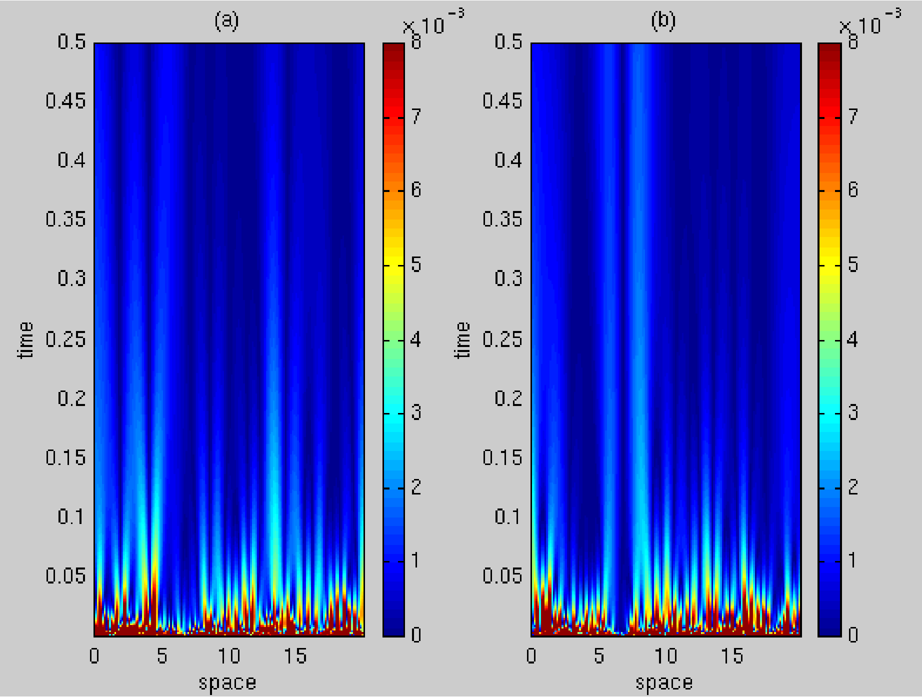

Figure 4: Decay ("healing") of the lifting error: comparison of the difference between an LB-BGK trajectory, and two distinct liftings of its restriction: (a) a local equilibrium fine lifting and (b) a more "coarse" random lifting —see text for discussion.— Healing has obviously been accomplished even at times much shorter than the timestepper reporting horizon.

3 COARSE PROJECTIVE INTEGRATION

Microscopic-level simulators, as described earlier, provide direct information about the short-term behavior of the simulated system. “Buried” in this output are clues to the coarse, long-term dynamics of the system; in the previous section we discussed a way of getting the infinite-time dynamics (elements of the ω-limit set, the steady states and limit cycles, the attractors and bifurcations) of the problem. Let us now turn to the *coarse trajectories* of the system, *e.g.* time histories of concentrations on the way to the eventual steady state. These *coarse* trajectories can be obtained from the microscopic evolution code directly, by running it over long periods of time, and occasionally producing as output the moments of interest (e.g. concentrations). It is not usually feasible, however, to continue the microscopic-level simulation over very long time periods because of the computational effort involved. Instead, we propose an approach of which the following is the simplest variant:

1. Choose the statistics of interest for describing the long-term behavior of the system and an appropriate representation for them. For example, in a gas simulation at the particle level, the statistics would probably be the pressure, density, and velocity and we might choose to discretize them in a computational domain via finite elements. For our LB-BGK problem the statistics of interest are the zeroth moments of the distribution over velocities (i.e. the concentrations) and we chose to represent them via finite elements. We will call this the macroscopic description, $\mathbf{u}$. These choices determine a restriction operator, $\mathcal{M}$, from the microscopic-level description, $\mathbf{U}$, to the macroscopic description: $\mathbf{u} = \mathcal{M}\mathbf{U}$.

2. Choose an appropriate *lifting* operator, μ, from the macroscopic description, $\mathbf{u}$, to the microscopic description, $\mathbf{U}$. 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. In our problem the lifting operator μ was discussed above in our presentation of the coarse time-stepper; μ should have the property that $\mathcal{M}\mu$ is the identity ($\mathcal{M}\mu = I$). In other words, lifting from the macroscopic to the microscopic and then restricting (projecting) down again should have no effect (except roundoff).

3. From an initial value at the microscopic level, $\mathbf{U}(t_0)$, run the microscopic simulator (the fine timestepper) for a number of simulated time steps, generating the values $\mathbf{U}(t_i)$ for $i = 1, 2, \cdots, n$.

4. Obtain the restrictions $\mathbf{u}(t_k) = \mathcal{M}\mathbf{U}(t_k)$, for $k = n, n-s, n-2s, \cdots, n-qs$ for some integers s and q.

5. Let $\mathbf{u}(t)$ be the q-th order polynomial in time, t, through $\mathbf{u}(t_k)$.

6. Evaluate $\mathbf{u}(T)$, where $T = t_n + H$ and H is a large time step, appropriate for the macroscopic-level description.

7. Lift $\mathbf{u}(T)$ to get a new consistent microscopic $\mathbf{U} = \mu\mathbf{u}(T)$ and use it as a new starting value for repeating steps 3 to 6.

Steps 3 and 7, in the spirit of the discussion of Section 2 above, may be performed for an appropriately chosen ensemble of microscopic initial conditions $\mathbf{U}_i$, all consistent with the same macroscopic condition $\mathbf{u} : (\mathcal{M}\mathbf{U}_j = \mathbf{u}, \forall j)$.

The procedure is illustrated in Fig. 5. The lifting and restriction steps – linking the microscopic ($\mathbf{U}$) and macroscopic ($\mathbf{u}$) descriptions of a state – the microscopic integration, as well as the extrapolative ("projective") step are schematically depicted. This *projective* step is performed in the macroscopic description of the problem, and is followed by a new lifting step and the procedure is repeated. Notice how a projective step may cause the solution to deviate from the "slow manifold" on which higher order moments are functionals of the low order, governing moments. We then rely on the microscopic integration to "heal" these errors, and approach the slow manifold again. This is symbolized in the figure by the first few microscopic steps, that appear to be approaching the correct "low moment" trajectory. We let this "trajectory healing" go on for some time before we estimate numerical derivatives for the projective integration step.

The conditions for this method to be successful were presented in an earlier paper [37] that introduced Projective Methods. These methods are related to the one proposed above without involving the restriction and lifting operators. They were applied to classes of problems that had two clusters of eigenvalues, a fast set whose components were damped very rapidly because the eigenvalues had large negative real parts, and a slow set which determined the long-term dynamics of the system. In the Projective Methods we relied on a number of initial steps of an "inner integrator" (one using a very small time step, commensurate with the fast components) to annihilate the fast components so that the polynomial extrapolation (called a "Projection" in the cited paper) could be applied to the slow components. We showed that, as long as there were enough inner steps to damp the fast components sufficiently, the stability of the method was determined by the stability of the extrapolation (Projection) process. That process was called an "Outer Integrator" because it has the characteristics of a standard numerical integration process for the differential equation $y' = f(y)$. The derivatives required for the "outer" integration were computed by interpolating the results from the inner integration (as opposed to evaluation of $f(y)$).

In the problems discussed in this paper, we cannot evaluate the right hand side $f(y)$ because the differential equation for the macroscopic description is not available; we therefore have to rely on numerical differencing or, more generally, interpolation of the *results of integration* to estimate the unavailable derivatives. The computational gain is that this estimation requires a few *short* integrations, while, hopefully, the extrapolation with the coarse estimated derivatives will be over *significantly longer* times. The microscopic simulation steps here are equivalent to the inner integrator steps of the earlier paper, and the extrapolation step is the outer integrator. In this case, however, we cannot assume that inner integration steps will damp the fast components - indeed, in most cases involving molecular dynamics, for example, they are highly oscillatory and will never be damped in an exact simulation. Instead,

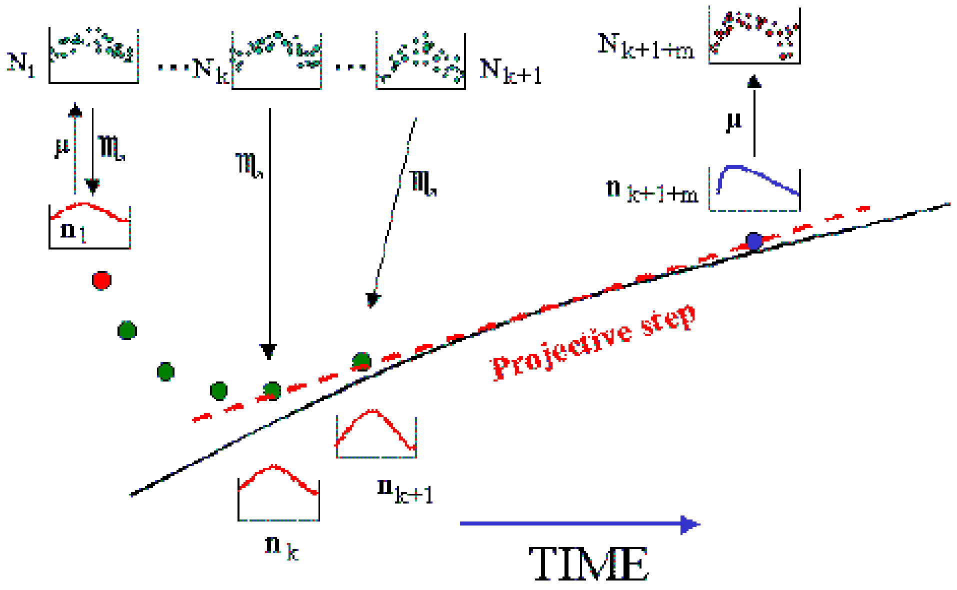

Figure 5: Schematic representation of an explicit projective integrator. Notice the correspondence between macroscopic and microscopic descriptions through lifting and restriction.

we must rely on the restriction operator $\mathcal{M}$ to annihilate the fast components. In what follows we will indeed work in cases where the coarse description (in terms of moments, such as concentrations) gives rise, due to irreversibility, to dissipative evolution equations (evidenced by the appearance in the macroscopic equations of terms containing diffusivity, viscosity, thermal conductivity etc.).

With sufficient assumptions on the problem, it is possible to prove that the method will work. An outline of the proof for a linear problem is as follows: We assume that the original microscopic problem is given by the differential equation $\mathbf{U}' = \mathbf{AU}$, while the macroscopic description is $\mathbf{u} = \mathcal{M}\mathbf{U}$. Then $\mathbf{u}$ satisfies the differential equation $\mathbf{u}' = \mathcal{M}\mathbf{A}\mu\mathbf{u}$. As long as the matrix $B = \mathcal{M}\mathbf{A}\mu$ only picks up the slow eigencomponents from the much higher-dimensional matrix A, then a large time step integration process will be stable if the step size is commensurate with those slow components. Thus there are conditions on the macroscopic description (which determines $\mathcal{M}$ and partially μ) that are equivalent to requiring that the macroscopic description not include any fast components — or when it does, the outer integration step must be correspondingly reduced. These are only conditions for stability of the coarse projective integrator. Accuracy conditions will also require that the macroscopic-level descriptions capture all of the active slow components. When nonlinearity is considered (and virtually all interesting problems are nonlinear), it is difficult to see how the analysis can handle other than small non-linearities. Also it is difficult to give useful criteria that will permit one to tell in advance when a problem is amenable to this approach.

An Illustration of Coarse Projective Methods. As in the case of timestepper based bifurcation analysis, RPM in particular, Projective Integration is a technique that can be used on a discretized PDE directly, should this PDE be available. It shares certain structural features with RPM: the same timestepper that was part of the RPM procedure is now the "inner integrator" for the projective methods. The "outer integrator" is now the computational superstructure which, after appropriately and repeatedly calling the inner timestepper, processes the results and performs the systems-level task (here projective integration over a long time horizon, over time scales characteristic of the slow modes). We therefore start by demonstrating that Projective Methods (and in this case explicit projective methods) can indeed facilitate the integration of a PDE for which an "inner", short horizon timestepper *is* available.

Explicit Projective Methods (Fine and Coarse). Figure 2 illustrates the results of integrating the Fitzhugh-Nagumo PDE in 1-dimension for the values of the kinetic parameters reported in the previous section. Here ϵ was set equal to 0.01. No flux boundary conditions were employed and the domain had length $S = 20$. The Galerkin FEM method was used to discretize the model in 50 elements and quadratic basis functions were used to interpolate the solution therefore yielding 101 computational nodes. An implicit Euler scheme was used to advance the solution in time. Under the above parametric conditions the solution is a stable limit cycle, shown in Fig 2: the sharp front characteristic of the steady state that lost stability in the nearby Hopf bifurcation moves back and forth over one period of the oscillation.

The "correct" converged solution, to which the projective results will be compared, is the 100 regular spatial mesh quadratic finite element solution, integrated in time through an implicit Euler method with step $\Delta t = 0.001$. Several discretizations with incresing number of elements, up to 400 elements (802 unknowns) were employed in order to ensure that the results were mesh independent; the integration results were also confirmed to be converged in terms of the temporal discretization through DASSL [38].

These results are compared first with projective integration results of the Fitzhugh-Nagumo PDE using the same FEM discretization scheme; the "inner" integrator is the implicit Euler with timestep dt=0.01 and the "outer" integrator is an explicit Euler method. To provide a sample of the new terminology associated with these integrators, the overall projective integrator in this figure would be characterized as a "50-50-100 inner finite element implicit Euler, outer finite element explicit Euler" or, more briefly, a "50-50-100 FEM-FEM" method. Fifty inner steps are taken, the solution recorded, fifty more inner steps are taken, the solution recorded again and then the solution is linearly extrapolated (projected forward) through an explicit Euler method, in which the derivative has been evaluated through numerically differencing the two recorded results over $H = 100$ more steps. In this approach we roughly do half the work of the detailed integration.

For reasons that will be discussed further below, we have also constructed empirical orthogonal global basis functions (so-called EOFs or POD modes) that can accurately represent the spatiotemporal dynamics of the problem in this regime. POD modes will be discussed in slightly more detail in Section 4 below. To facilitate the comparison, the results are shown as (i) time series, starting from the same initial condition (a stable steady state for $\epsilon = 0.1$), see (Fig6a and b); and (ii) as phase space projections in the space of these same first few global POD modes. We have chosen to plot, instead of the time series of concentration values at some representative point $u(x_i, t)$, the time series of the components of the solution along the first few members of these POD modes.

Figure 7 now shows a comparison of the detailed problem with a projective LB-FEM method. This is our first example where the inner integrator (a Lattice Boltzmann BGK) acts on a different level description (LB) than the outer one (FEM). The inner integrator is not, strictly speaking, a numerical integrator - it consists of the "primitive" dynamics of the Lattice Boltzmann methodology. In this case we are comparing the results of the "full" Lattice Boltzmann limit cycle (projected on the first few density POD modes) with the results of the LB-FEM simulation. Here we took 100 steps with the inner LB integrator, recorded the restriction of the solution to the 0th moments; 50 more steps were taken, the solution restricted and again. The solution was then linearly extrapolated over 350 steps. This can be therefore called a "100-50-350 LB-FEM" method. There is an important distinction between the projective integrator of the FHN PDE above, and the coarse projective integrator described here. In this case the extrapolative step is performed *not on the primitive Lattice Boltzmann variables*, but on their projections to coarse variables (densities) discretized through finite elements (hence the characterization of this scheme as an

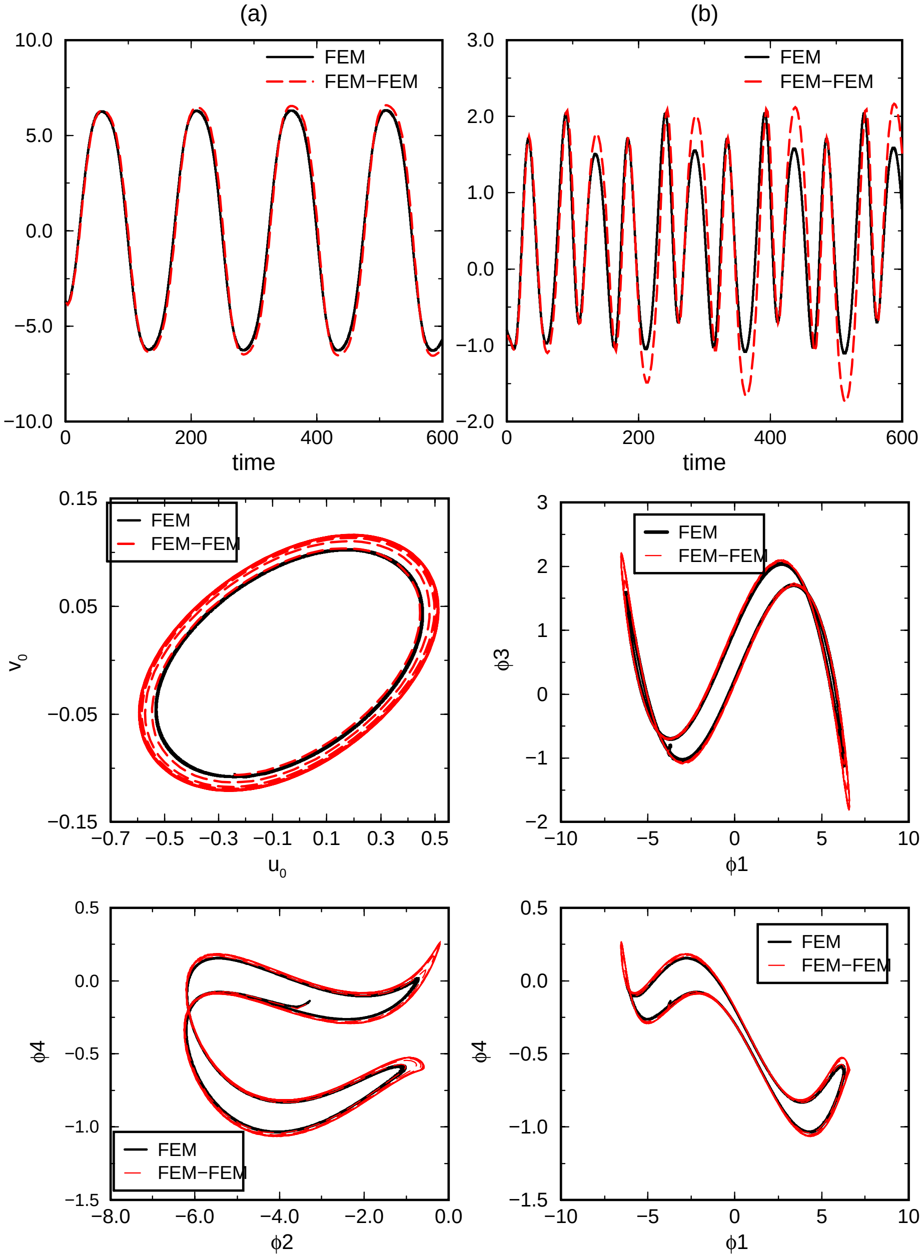

Figure 6: A comparison of FHN dynamics computed through (1) an implicit Euler integrator and (2) a 50-50-100 inner-FEM (implicit Euler) - outer FEM (explicit Euler) projective integrator. The comparisons are performed by projection on subspaces spanned by a few empirical orthogonal eigenfunctions (EOFs 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 discretized PDE (1, black) and the projective FEM-FEM scheme (2,red).

LB-FEM one, more accurately as an "inner LB, outer FEM explicit Euler" scheme.

It is interesting to consider the differences between short-term and long-term dynamic predictions of the two codes (the "full" LB and the projective LB-FEM)), as illustrated in Fig. 8

Clearly, the LB-FEM can be thought of as a perturbation of the LB scheme. The perturbation is obviously due to the projective step in time; but it also comes from the restriction and lift steps that constitute the communication between descriptions. Once more, we are counting on irreversibility and separation of time scales to "heal" the errors we make in the higher order moments upon lifting. By comparison of the short-term and the long-term results of the two methods we see that (as with any two numerical schemes) the pointwise predictions gradually deteriorate forward in time, while the long-term attractors (as sets in the same phase space projection) remain close to each other.

Implicit Coarse Methods: The extrapolation-based integration method described above is equivalent, from a stability point of view, to a q-th order Taylor series method (which has the same properties as the q-th order, q-stage Runge-Kutta method that exists for $q \leq 5$. For many problems, this is an adequate collection of methods, but for some the limited range of absolute stability makes implicit methods desirable. Implicit Projective methods exist, and in the context of this paper they take the form:

- (i) Compute some number, n, of microscopic simulations steps from an initial state $\mathbf{U}(t_0)$, say $\mathbf{U}(t_i)$.
- (ii) Starting from $\mathbf{u}$(T) (the as-yet-unknown macroscopic result of one outer integration step) compute $\mathbf{U}(\mathrm{T}) = \mu\mathbf{u}(\mathrm{T})$, and then compute a further n microscopic simulation steps forward from T, say $\mathbf{U}(T_i)$.
- (iii) Using the restrictions of some subset of q+1 of $\mathbf{u}(t_i) = \mathcal{M}\mathbf{U}(t_i)$ and $\mathbf{u}(T_i) = \mathcal{M}\mathbf{U}(T_i)$ determine the q-th degree polynomial $\mathbf{u}$(t) that passes through them and evaluate $\mathbf{u}$(T).
- (iv) Use whatever implicit solution method seems appropriate to ensure that the $\mathbf{u}$(T) used in step (ii) is the $\mathbf{u}$(T) generated in step (iii). For example, functional iteration could be used by simply repeating steps (ii) to (iv) starting with the $\mathbf{u}$(T) computed in the last iteration. Typically the first iterate will be computed with an explicit method - the predictor. The implicit formula is then called the corrector.

For technical reasons mainly to do with the rate of convergence of any implicit iteration, it seems to be preferable to modify step (iii) slightly in the following way: Take a subset of q+2 restrictions $\mathbf{u}$, with at least 2 from the set $\mathbf{u}(T_i)$, but use only differences of the values of the set from $\mathbf{u}(T_i)$. This will be illustrated with the analog of the trapezoidal rule below.

Although we specified the same number of microscopic integration steps, n, in (i) and (ii) above, this is not essential. Nevertheless, if the same type of outer integration

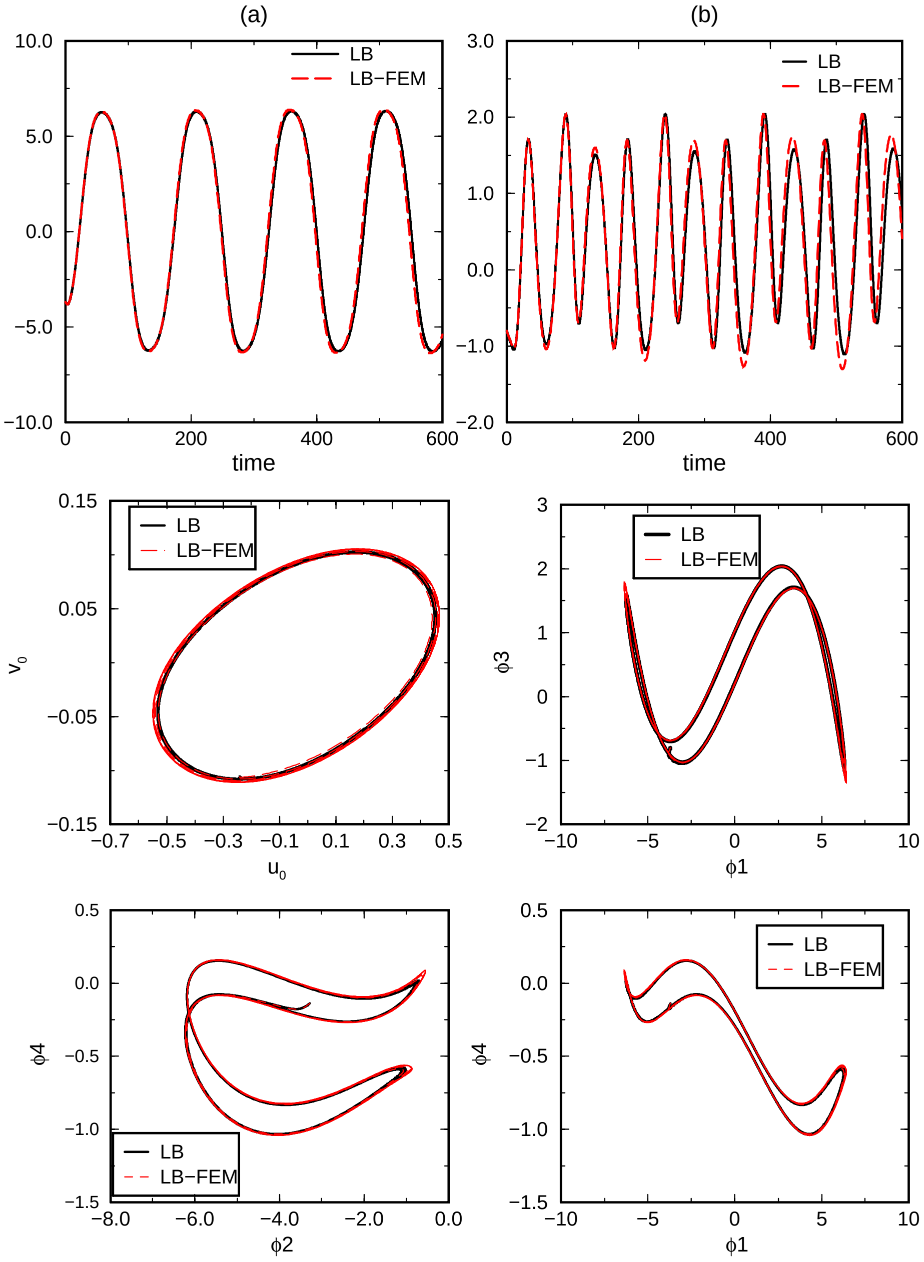

Figure 7: A comparison of FHN dynamics computed through (1) an LB-BGK code and (2) a 100-50-350 inner-LB - outer FEM (explicit Euler) projective integrator. The comparisons are performed by projection on subspaces spanned by a few empirical orthogonal eigenfunctions (EOFs or PODs, see text). Top two 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).

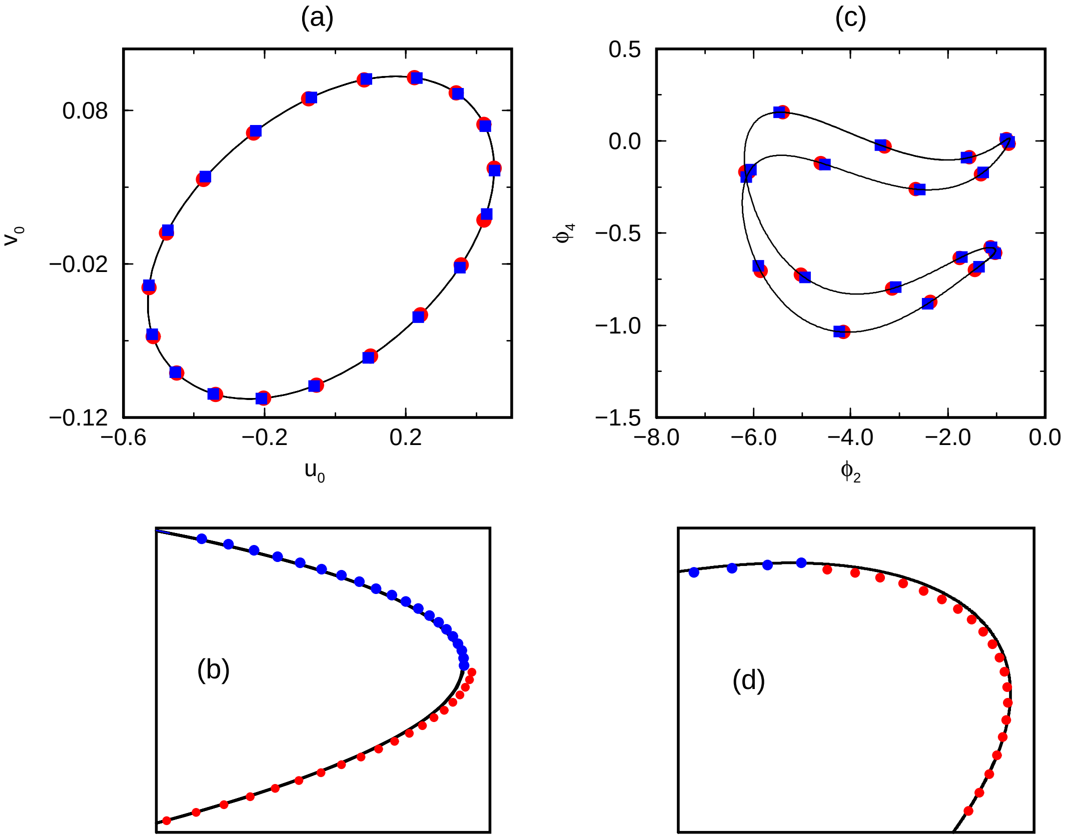

Figure 8: Short-term dynamic predictions of a 100-50-250 LB-FEM scheme. 20 equidistant points are taken on the converged LB limit cycle, and LB-FEM is performed for approximately 1/20 of the period ($t \approx 8$); this is equivalent to 20 of our projective steps. The initial points in (a,c) are denoted by the red circles and the final points by the blue squares. (a) Phase space projection on u_0 and v_0. (b) Magnification of panel (a) showing the LB (solid line) and the LB-FEM (color dots) trajectories, to emphasize the short-term error of the projective integration. (c) Phase space projection on POD modes ϕ_2 and ϕ_4; and (d) corresponding magnification of the trajectories of the two integrators starting at two nearby points on the LB limit cycle.

step is to be performed again, using the same n means that algorithm step i of the next outer integration has already been done in the last step of the previous corrector.

Example - The Coarse Projective Trapezoidal Rule: We advance from $\mathbf{U}$(t) to $\mathbf{U}$(t+T) by the following procedure:

- (i) Using an inner integration step of h compute $\mathbf{U}$(t+ih) for i = 1, 2, 3, ...n, ... , n+s.
- (ii) Form $\mathbf{u}$(t+nh) = $\mathcal{M}\mathbf{U}$(t+nh) and $\mathbf{u}$(t+(n+s)h) = $\mathcal{M}\mathbf{U}$(t+(n+s)h).
- (iii) Predict $\mathbf{u}(t+T) = \mathbf{u}(t+(n+s)h) + (T-(n+s)h)D_n$ where D_n is an approximation to the derivative at t+nh obtained by differencing, namely $D_n = (\mathbf{u}(t+(n+s)h) - \mathbf{u}(t+nh))/(sh)$.
- (iv) Using this as a first iterate, and setting $H = T + nh$ solve the corrector equation $\mathbf{u}(t+T) = \mathbf{u}(t+(n+s)h) + (T-(n+s)h)[aD_n + (1-a)D_H]$ where D_H is an approximation to the derivative at t+H obtained by differencing $\mathbf{u}$(t+T+(n+s)h) and $\mathbf{u}$(t+T+nh),

For $a = 0.5$ a method resembling to the trapezoidal rule is obtained. As described above, this trapezoidal rule does not have second order since the difference estimates of the derivatives are at the point t+(n+s/2)h and t+T+(n+s/2)h, rather than at t+nh and t+T. The coefficients can be corrected to obtain second order accuracy (see [37]) with the formula

$$a = \frac{nh+T}{2T} \tag{12}$$

We have implemented the above algorithm in a 100-50-700 LB-IMPLICIT FEM integrator with the 2nd order correction described in equation 12, for the same initial conditions and parameters used earlier in this section. It should be denoted here that by using a 2nd order accurate implicit projective step a much larger accurate projection (700 compared to 350) could be achieved compared to explicit outer integrators. In these implicit integration runs a tolerance of 10^{-6} was employed for convergence of the corrector iteration, and each implicit projective step coverged in approximately 5 corrector iterations. Figure 9, analogous to Figures 6 and 7 above, shows the results of this integration: it compares the results of full LB and projective trapezoidal "fine LB, coarse FEM" integration, both for short term (startup) and for long term dynamics (the limit cycle attractor).

4 MICRO-GALERKIN METHODS

We illustrated above how to use a microscopic time-stepper in order to perform coarse bifurcation and stability analysis as well as coarse integration. We emphasized the importance of the lifting step, in which information is generated (since the higher order moments of the distribution have to somehow be initialized). The relatively easy restriction step (from microscopic "fine" to coarse description) was only briefly described; we will revisit this now. It might be conceptually useful to consider this

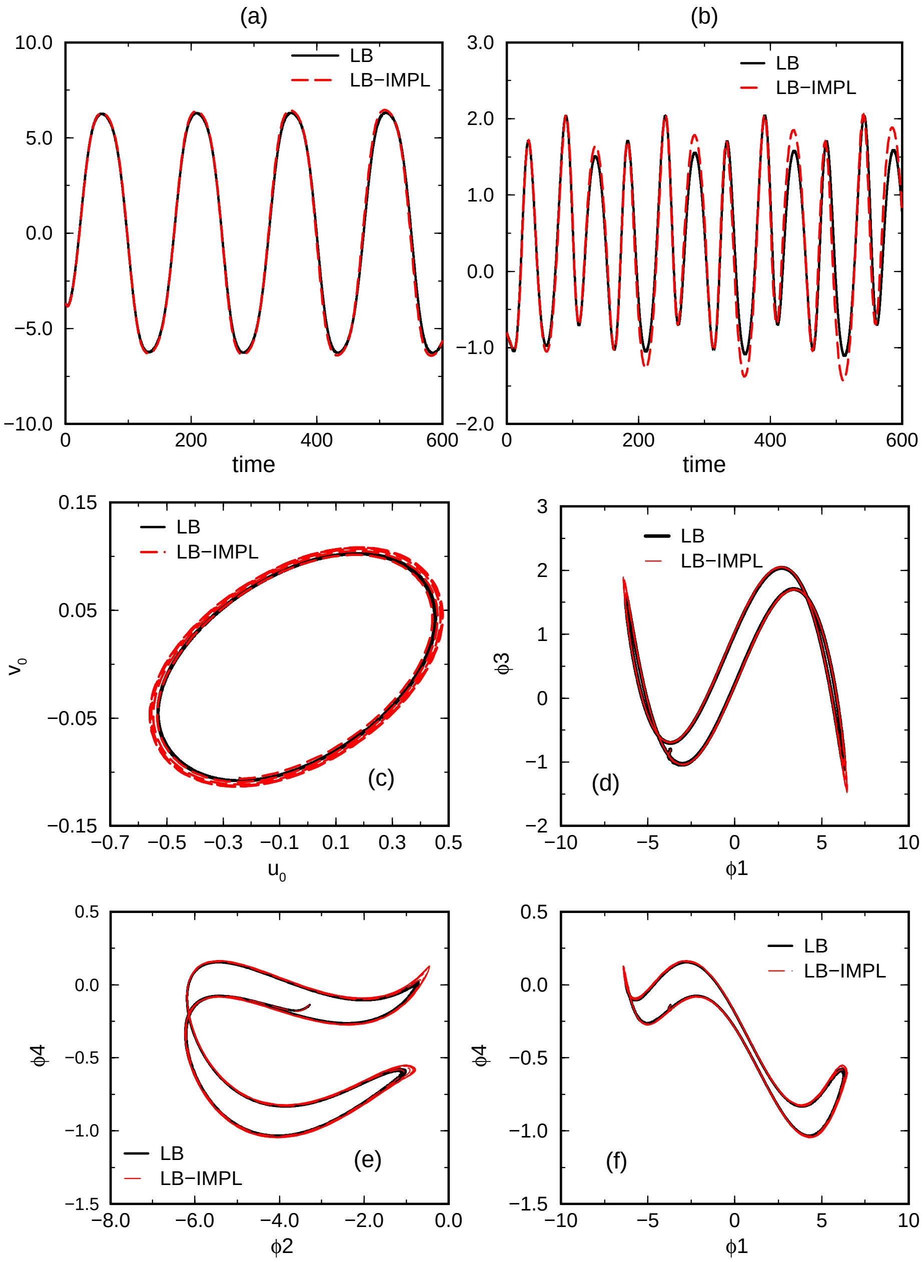

Figure 9: A comparison of FHN dynamics computed through (1) an LB-BGK code and (2) a 100-50-700 inner-LB - outer implicit projective integrator using the implicit projective trapezoidal rule. The comparisons are performed by projection on subspaces spanned by a few empirical orthogonal eigenfunctions (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 100-50-700 LB-implicit projective scheme (2, red).

restriction as taking place in two stages. The first stage consists of obtaining moments, actually fields of moments of the distributions in question (like concentration fields u(x,t) and v(x,t) in our example); for the sake of simplicity let us consider these fields as being obtained as values on a very fine mesh, much too fine for, say, coarse finite difference or finite element computations. The second stage involves the Galerkin projection of this "over- resolved" coarse description on a mesh more practical for the coarse computations performed by our computational superstructure (RPM or outer integrator). For the above, we used a "generic" library of basis functions (quadratic finite elements in one dimension) for the coarse computational work. We have also performed the same tasks with finite differences as the coarse representation of choice. A multitude of coarse representations can easily be included as alternatives in the superstructure; the only difference will be the set of coarse basis functions that we take inner products with.

It makes sense to us to colloquially term our coarse multi-scale methods "micro-Galerkin" methods: the coarse level computational superstructure (whether for bifurcation or for integration purposes) acts on Galerkin projections of the results of the microscopic evolution scheme on some "reasonable" set of coarse basis functions.

Since all these coarse procedures can be thought of as "on-line" model reduction procedures exploiting the separation of time scales, we chose to illustrate here a micro-Galerkin method that contains, by nature, a second level of model reduction: the *micro-POD* method. The idea is to perform coarse bifurcation analysis as well as coarse integration using empirical basis functions for the spatial description of the coarse variables.

The so-called method of empirical orthogonal eigenfunctions (EOFs), also termed Proper Orthogonal Decomposition (POD) and Karhunen-Loève expansion (KL) has long been proposed as a method of model reduction for nonlinear PDEs with low-dimensional long-term dynamics. The method consists of two stages, one being effectively data compression, and the other derivation of predictive low-dimensional dynamic models (see the monograph [39] for a recent review).

The first stage consists of statistical analysis of extensive simulation databases (through, essentially, Principal Component Analysis) to global basis functions. For nonlinear PDEs these databases may come either from simulation e.g. [40, 41] or from experiment e.g. [42]. For microscopic/molecular systems the data may also come from simulation or from experiment. It is obvious, though, that macroscopic PODs for experimental velocity fields in fluid mechanics or concentration fields in chemical reactions truly come from molecular systems; it is our *measurements* of the molecular systems that appear continuous. The only reason we think of them as continuum fields is because of the level at which the experimental system is observed - we directly observe moments of the microscopic distributions, since that is what our instruments are capable of measuring. Therefore, the PODs from field measurements of experimental observations are, truly, already "coarse description" PODs obtained through microscopic realization/evolution of the system. Projecting the solution onto the first few POD modes constitutes data compression: an experimental movie is reduced to a few time series for the amplitudes of its projection on the most energetic modes.

The second stage, that pertains more to systems analysis, is the exploitation of these eigenfunctions —and the resulting few time series— for the extraction of accurate, reduced, dynamic models. For macroscopic problems, where the "coarse" PDE-level dynamic model is available, this is done through a POD-Galerkin projection (see [43] and [44] for some of the original references). This is a subject of intense research by many groups on several applications; work by our group has concentrated on complex geometries for bifurcation/stability applications [40, 41] as well as on control related applications [45, 46, 47].

If, however, the macroscopic, coarse description is not available, coarse POD modes are only useful in data compression. The only way to construct low-dimensional predictive models would appear to be empirical - that is, the fitting (through some nonlinear system identification method) of a right-hand-side for the evolution of the coarse POD amplitudes based on the available few time series. One instance where this was accomplished using an artificial neural network (ANN approach) is described in [42, 48]. Video sequences of real-time micrometer scale experimental data from the catalytic oxidation of CO on Pt (110) were analyzed through the POD procedure to produce four time series. These time series were used (through delay reconstruction methods) to fit discrete-time dynamic models of the process. In a further reduction step, nonlinear principal components (autoregressive neural networks) were used to further reduce the degrees of freedom to three, and recurrent neural networks templated on numerical integration schemes [42] were used to extract a continuous-time right-hand-side for the equations. Thus three scalar initial conditions were enough to "replay" the video sequence.

The empirical identification of a right hand side from coarse POD time series, however, leaves something to be desired. If a microscopic simulation code is available, it should be possible to use it in constructing a coarse dynamic model. This is precisely where micro-Galerkin projective integrators come in play. The procedure involves:

1. **Off line:**

- Collection of a representative ensemble of microscopic simulation data;
- Projection of the data to appropriate moments (*e.g.* concentration fields);
- Extraction of empirical eigenfunctions (POD modes) for these fields.

2. **On-line:**

- Start with a POD (reduced moment space) initial condition (a small vector);
- Reconstruct the full coarse initial condition (through the POD modes);
- Lift the full coarse initial condition -as was discussed above- to one or more microscopic initial conditions;
- Evolve this(ese) microscopic initial condition(s) for a time horizon, recording the solution(s) and its projection onto moment space, and in particular on the reduced POD moment hyperplane.

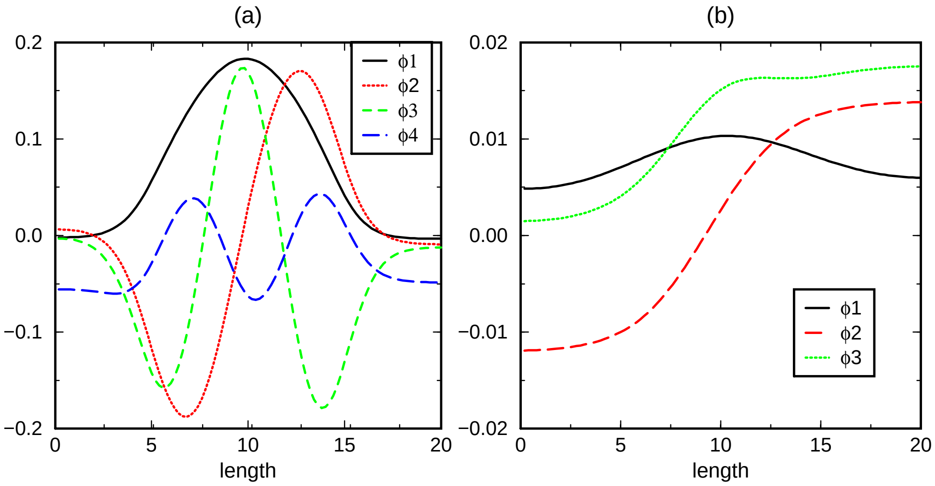

Figure 10: Spatial structure of the leading POD modes —(a): u-component, (b): v-component— for the LB simulation

- Perform extrapolative integration in the POD hyperplane, through either an explicit or an implicit projective method, and then
- Go back to the second step.

To illustrate this "doubly reduced" micro-Galerkin method (first microscopic to macroscopic, to moments space, and then from moments discretized space to moments POD hyperplane) we again turn to our LB simulation of the FHN model. We start through the collection of a representative set of microscopic (LB) simulations for a range of parameter values both before and after the Hopf bifurcation, including both transient simulations "on the way" to the stable steady states (when they exist) or away from them towards the unstable limit cycle, as well as data on the attractors themselves. We then project the LB data on moments (finely discretized concentration fields, $u(x)$ and $v(x)$) and then perform an SVD on these projections to determine the hierarchy of the POD modes. In this paper we find joint, concatenated $[u, v]$ POD modes. Table 1 shows the energy content of the POD modes and Fig. 10 shows the shape of the first few modes. We mention that the shape of the first few modes obtained through the projected microscopic simulations are in reasonable agreement with the modes obtained in a completely macroscopic study of the same model [45]. As shown in Table 1, 20 POD modes capture $\tilde{9}9.9999$ % of the energy of the system.

We start with an LB-POD integration, doing 100 inner LB steps recording the solution and performing a double projection: first restriction to the 0-th moments and then to the 20 most energetic POD modes; then 50 more inner LB steps are taken, the solution restricted to 0-th moments and projected, once more. The outer integration projective step involving 20 POD coefficients only, spans 350 timesteps. Figure 11 shows, in the spirit of Figure 6 above, the short- and long-term integration results of the "100-50 inner LB - 350 outer explicit Euler POD" integration.

In.	Normalized Eigenvalues	Cumulative Energy
1	0.7647237	0.7647237
2	0.1493851	0.9141088
3	0.043923813	0.9580327
4	0.015614404	0.9736471
5	0.014191233	0.9878383
6	0.005535833	0.99337416
7	0.00345	0.9968232
8	0.00185447	0.9986776
9	0.00077568	0.9994533
10	0.00032258	0.999776
11	0.00012495	0.9999010
12	4.824210E-05	0.999949
13	1.949975E-05	0.999968
14	1.487844E-05	0.999983
15	7.433011E-06	0.999991
16	3.058370E-06	0.9999941
17	2.443162E-06	0.9999965
18	1.111155E-06	0.9999977
19	1.015913E-06	0.9999987
20	7.143144E-07	0.9999994

Table 1: Normalized Eigenvalues of POD modes and cumulative energy of the system

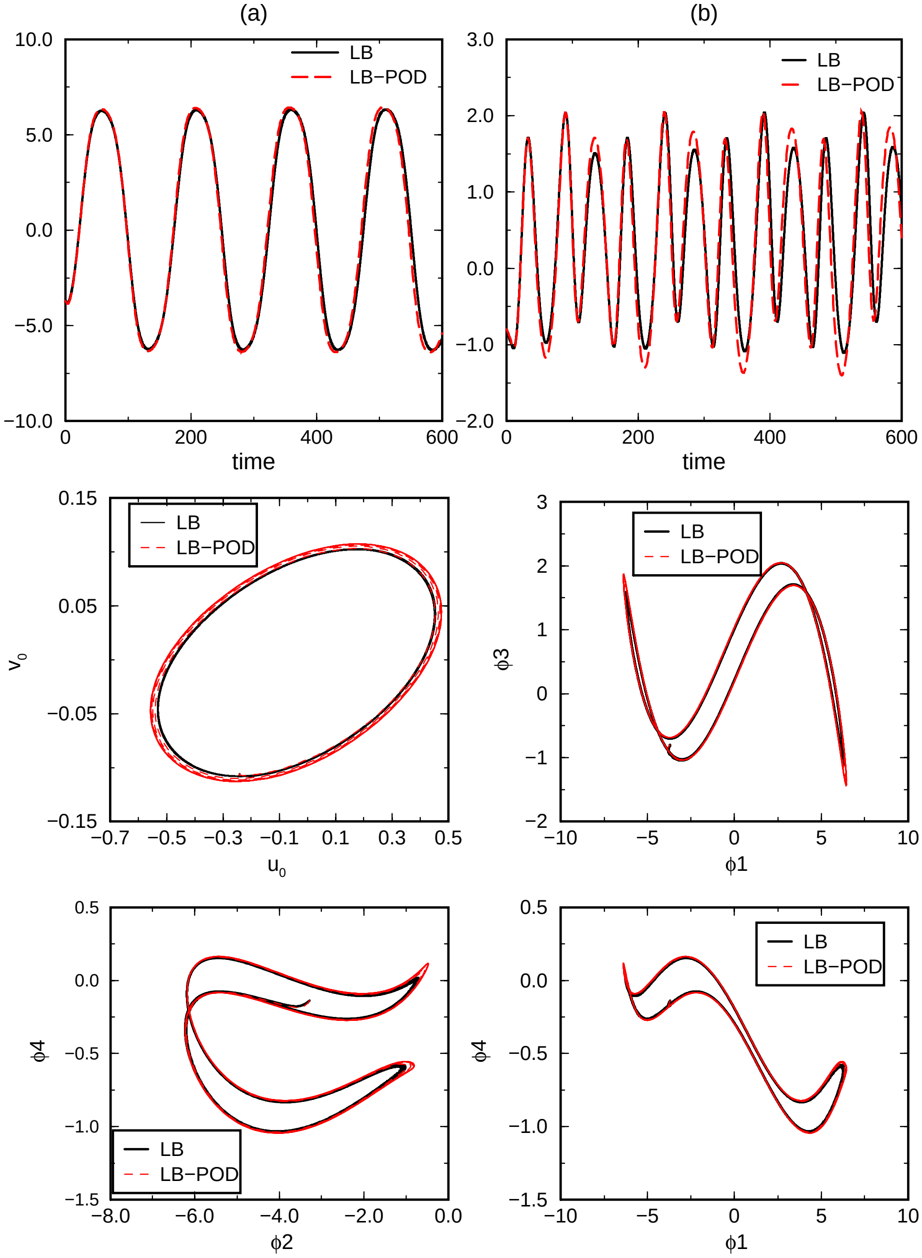

Figure 11: A comparison of FHN dynamics computed through (1) an LB-BGK solver and (2) a 100-50-350 inner LB-BGK - outer POD (explicit Euler) projective integrator. The comparisons are performed by projection on subspaces spanned by a few empirical orthogonal eigenfunctions. Top two panels: time series 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-POD projective scheme (2,red).

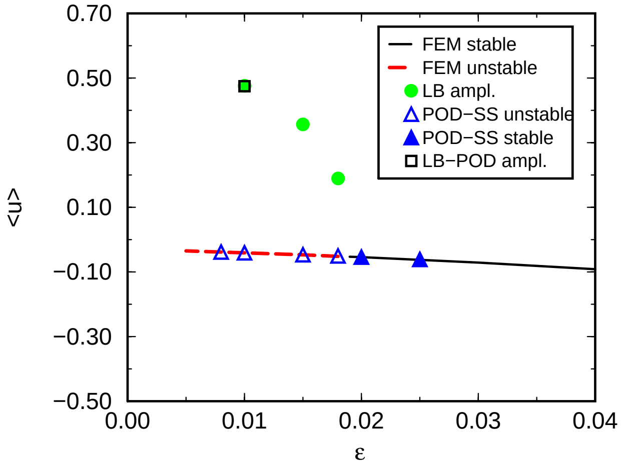

Figure 12: Comparison of the FEM-based and the LB-POD based bifurcation diagrams for the FHN PDE. Solid (broken) line: Stable (unstable) branch of steady states computed through the FEM discretization. Filled (open) triangles: Stable (unstable) steady states computed through LB-POD. Open square:amplitude of the corresponding limit cycle computed through the 100-50-350 LB-POD projective integration.

In a sense this is "as fast as we can go" exploiting our LB-BGK microscopic simulation code across scales.

Is it clear what we do here? We conclude with the long-term dynamics (the bifurcation diagram) of our micro (LB)-POD Galerkin scheme. Figure 12 shows the bifurcation diagram of the coarse (in the POD hyperplane of the moment space) LB-BGK-based timestepper, and compares it with the FHN FEM and the FHN-LB bifurcation diagrams.

This time the space dimension is small enough that RPM is actually not necessary: centered finite differences are quite practical in producing the full (20x20) numerical Jacobian in this low-dimensional coarse space. The stable and unstable steady states, the Hopf bifurcation value as well as the amplitude of the oscillations in the POD hyperplane coarse projection are successfully captured. So also are the coarse critical eigenvectors at the Hopf bifurcation (see Fig.13)

5 SUMMARY AND DISCUSSION

We have demonstrated the implementation of a suite of computational tools (which we collectively characterized as micro-Galerkin methods) for the *coarse* integration, stability and bifurcation analysis of atomistic level, microscopic simulations. The methods we described were based on the idea and implementation of a coarse time stepper, which evolves the system in time at one level of description, and a computational superstructure that performs bifurcation computations (e.g. the Recursive

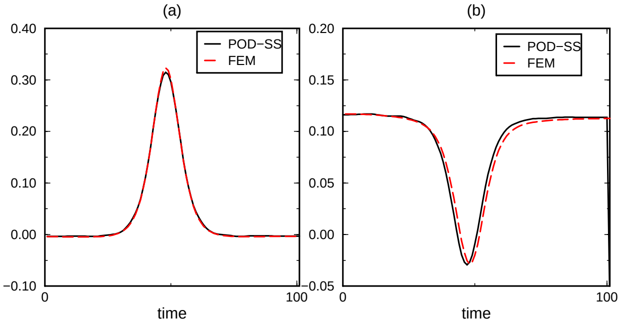

Figure 13: Comparison of the Hopf bifurcation eigenvectors (a) real and (b) imaginary; the corresponding eigenvalue is: $0.25 \pm 1.025i$, $\epsilon = 0.018$. The data have been obtained through a FEM (solid lines) as well as an LB-POD (broken lines) based discretization.

Projection Method) or coarse integration (e.g. explicit or implicit projective integration) at another level of description. Throughout this paper the transition was from microscopic to moments space, since moment closures underpin some of the most standard approaches to multiscale dynamics. It is important, however, to note, that if some heuristic set of describing variables (and not just the obvious moments of the distribution) are known to do a good job in parametrizing the coarse dynamics, the method can be used to go between microscopic and *empirical coarse* description level without change. An example would be the coarse description of a problem through an "order parameter field", say, a dislocation density field, instead of the first few of a hierarchy of moments of the molecular distribution.

The communication across scales is performed through a *lifting* procedure from coarse to fine, and a *restriction*, from fine to coarse. The lifting is a very important component of the overall computation, which conceptually lies at the heart of the approach. The idea is that errors made during lifting (the *ad hoc* initialization of the higher moments conditioned on the lower, determining ones) are healed since higher order correlations quickly evolve to functionals of lower order ones. Such a lifting step has been carefully considered by M. D. Graham and coworkers in their OSCM scheme in the context of non-Newtonian rheological simulations [49]. They use an optimization approach for their initialization, a procedure which resonates with considerations that can be found in the statistical mechanical literature [50]. In our experience with Monte Carlo methods, initialization is sometimes straightforward (e.g. when one needs to initialize only the average consistently) but may also require solving an optimization problem (e.g. making a different type of Monte Carlo run) if higher order moments (e.g. pair probabilities on a lattice) also need to be initialized [52].

The various aspects of the procedures we describe raise a number of points for discussion. What if the simulation in question is a stochastic one, such as a Monte Carlo simulation? The coarse time stepper from any single realization is then quite noisy. In this case, variance reduction plays a vital role in approximately ("almost always" solving the *noisy* fixed point equations, and stochastic approximation algorithms become important [51]. Suffice it to say that, in our experience with Monte Carlo methods [52], variance reduction obtained through many simultaneous realizations of trajectories starting at the same coarse initial condition hold the key to variance reduction. At the same time, this underscores the potential of the method for fruitful utilization of massively parallel computational resources, since many, non-interacting copies of a simulation starting at the same coarse initial condition are required. A very interesting interplay arises then between the number of copies that one uses, and the reporting horizon of the coarse timestepper. The many copies have to run for a long enough time for the higher order correlations to *heal*, but not so long that they sample the entire ultimate probability density function over the entire phase space, thus losing all phase information and becoming trivial; the (temporal) "density of collapses" (the time we run microscopic realizations before we average them and restrict them to moment space, to lift and run again, is vital for the coarse description to be successful [53, 54].

Some of the points to be made concern the "macro" side of the computation. It is worth noting, for example, that not only coarse steady states, but also coarse limit cycles can, and have been computed through this approach. The idea is to locate (through Newton-Picard iterations [25]) a fixed point not of an arbitrary time T map (thus a coarse steady state), but of the Poincaré return map, for the special return time that is either the period of the coarse oscillation or an integer multiple of it. It is also important to note that further reductions at the coarse level can also be done. For example, instead of truncated flat Galerkin methods with traditional or empirical global basis functions (like the finite element or the POD modes used above) it is possible to use nonlinear Galerkin methods [24, 41] to obtain an even smaller overall coarse description.

It is fitting that we close the discussion with a brief mention of additional systems-level tasks such as controller design and optimization. We have already observed that the slow coarse Jacobian with its eigenvalues and eigenvectors as well coarse parametric derivatives are a natural byproduct of the process in a continuation environment. It is precisely these slow coarse linearizations that can be put to good use in controller design. The concomitant separation of time scales (the coarse problems is a singularly perturbed one) points to a rich control literature [55] where these slow coarse Jacobians can be used for pole placement, solution of Riccati equations and LQR design, as well as for the construction of "coarse" observers. And while we described everything so far in terms of coarse Jacobians, coarse Hessians, based on the time-stepper approach, can also be computed, and local optimization techniques involving these matrices used for design level computations for a description for which we have no available closed form model.

We are actively pursuing several avenues in this work. We are working on LB multiphase flow results in collaboration with S. Sundaresan [9]; on lattice gas models of reaction kinetics with A. Malevanets and R. Kapral; on timestepper based homogenization with O. Runborg [56]; on coarse bifurcation analysis of discrete lattices with K. Lust; on finite temperature crystal lattice transitions through MD with D.

Maroudas; and so on. In addition to solving various problems, we seek out the common features as well as the differences of the basic steps (lifting, microscopic evolution, averaging, time-horizon selection issues, variance reduction issues) that arise in the various fields. We hope that, as the experience with these time-stepper based methods develops, they may evolve into a "tool of choice" for bridging scales "on line" complementing the traditional "off-line" bridging provided by analytical closures. This is of course a mainstream avenue in current research, where innovative multiscale techniques like [57] and novel stochastic integration techniques [58] are constanly being proposed and explored.

It is worth mentioning that this approach allows us to construct coarse bifurcation diagrams with respect to *microscopic* model parameters directly. We note, in closing, that while our techniques were presented as a way to co-process microscopic simulation codes, such as MD, MC, LB-BGK etc., they really are a tool for co-processing time evolution codes across scales. The better the "fine scale" code (e.g. biased MC, Brownian configuration fields), the better the results this approach can give. It is finally worth mentioning that in what we presented here, timesteppers were co-processed across *two* descrition scales; there is nothing to prevent a "telescoping" co-processing across more than one scale gap; a first view of this telescoping procedure can be found in [59].

Acknowledgements. This work was partially supported by the Air Force Office of Scientific Research (Dynamics and Control) the National Science Foundation and a Humboldt Foundation Prize. The assistance of Dr. Yue-Hong Qian with his FHN LB model is gratefully acknowledged.

References

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

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

[3] C. Theodoropoulos, Y.H. Qian and I.G. Kevrekidis (1999). *Proc. Nat. Acad. Sci.* **34**,

[4] NAG FORTRAN Library Manual, http://www.nag.co.uk/numeric/fl/manual/pdf/genint/foreword_fl19.pdf

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

[6] 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.

[7] 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.

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

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

[10] 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.

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

[12] This term was suggested by Prof. D. Maroudas.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[41] A. K. Bangia, P. F. Batcho, I. G. Kevrekidis and G. M. Karniadakis (1997) *SIAM J. Sci. Comput.* **18** 775.

[42] K. Krischer, R. Rico-Martinez, I. G. Kevrekidis, H. H. Rotermund, G. Ertl and J. L. Hudson (1993) *AICHE J.* **39** 89-98.

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

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

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

[46] 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.

[47] H. Aling, S. Banerjee, A. K. Bangia, V. Cole, J. L. Ebert, A. Emami-Naeini, K. F. Jensen, I. G. Kevrekidis and S. Shvartsman (1997) *Proc. ACC* 2223.

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

[49] R. M. Jendrejack, J. J. de Pablo, and M. D. Graham (2000) in *Proc. XIIIth Int. Cong. Rheology*, D. M. Binding et al., eds. 2.214-2.216.

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

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

[52] A. Makeev, D. Maroudas and I. G. Kevrekidis (2001) *J. Appl. Phys.* submitted.

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

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

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

[56] O. Runborg, C. Theodoropoulos and I. G. Kevrekidis (2001) *Nonlinearity* submitted.

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

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

[59] C. W. Gear and I. G. Kevrekidis (2001) NECI Research Report, October 2001.