# Equation-Free Modeling For Complex Systems

YANNIS G. KEVREKIDIS
(with C. W. Gear and G. Hummer)
*Department of Chemical Engineering, Program in Applied and*
*Computational Mathematics, and Department of Mathematics*
*Princeton University*
*Princeton, New Jersey*

## ABSTRACT

*In current modeling, the best available descriptions of a system often come at a fine level (atomistic, stochastic, microscopic, individual-based), while the questions asked and the tasks required by the modeler (prediction, parametric analysis, optimization and control) are at a much coarser, averaged, macroscopic level. Traditional modeling approaches start by first deriving macroscopic evolution equations from the microscopic models, and then bringing an arsenal of mathematical and algorithmic tools to bear on these macroscopic descriptions. Over the last few years, and with several collaborators, we have developed and validated a mathematically inspired, computational enabling technology that allows the modeler to perform macroscopic tasks acting on the microscopic models directly. We call this the "equation-free" approach, since it circumvents the step of obtaining accurate macroscopic descriptions. We argue that the backbone of this approach is the design of (computational) experiments. Traditional continuum numerical algorithms can be viewed as protocols for experimental design (where "experiment" means a computational experiment set up and performed with a model at a different level of description). Ultimately, what makes it all possible is the ability to initialize computational experiments at will. Short bursts of appropriately initialized computational experimentation— through matrix-free numerical analysis and systems theory tools like variance reduction and estimation—bridge microscopic simulation with macroscopic modeling. Remarkably, if enough control authority exists to initialize laboratory experiments "at will," this computational enabling technology can become a set of experimental protocols for the equation-free exploration of complex system dynamics.*

A persistent feature of many complex systems is the emergence of macroscopic, coherent behavior from the interactions of microscopic "agents"—molecules, cells, individuals in a population—among themselves and with their environment. The implication is that macroscopic rules, a description of the system at a coarse-grained, high level, can somehow be deduced from microscopic ones, a description at a much finer level. For laminar Newtonian fluid mechanics, a successful coarse-grained description, the Navier-Stokes equations, was known on a phenomenological basis long before its approximate derivation from kinetic theory. Today we must frequently study systems for which the physics can be modeled at a microscopic, fine scale; yet it is practically impossible to explicitly derive a good macroscopic description from the microscopic rules. Hence, we look to the computer to explore the macroscopic behavior based on the microscopic description.

Macroscopic models of reaction and transport processes in our textbooks come in the form of conservation laws (species, mass, momentum, energy) closed through constitutive equations (reaction rates as a function of concentration, viscous stresses as functionals of velocity gradients). These models are written *directly* at the scale (alternatively, at the level of complexity) at which we are interested in practically modeling the system behavior. Because we *observe* the system at the level of concentrations or velocity fields, we sometimes forget that what is really evolving during an experiment is distributions of colliding and reacting molecules. We know, from experience with particular classes of problems, that it is possible to write predictive deterministic laws for the behavior (predictive over relevant space/time scales that are useful in engineering practice) observed at the level of concentrations or velocity fields. Knowing the right level of observation at which we can be *practically predictive*, we attempt to write closed evolution equations for the system at this level. The closures may be based on experiment (e.g., through engineering correlations) or on mathematical modeling and approximation of what happens at more microscopic scales (e.g., the Chapman-Enskog expansion). In many problems of current modeling practice, ranging from materials science to ecology and from engineering to computational chemistry, the physics are known at the microscopic/individual level, and the closures required to translate them to a high-level, coarse-grained, macroscopic description are not available. Sometimes we do not even know *at what* *level of observation* one can be practically predictive. Severe computational limitations arise in trying to bridge, through direct computer simulation, the enormous gap between the scale of the available description and the macroscopic, “system” scale at which the questions of interest are asked and the practical answers are required (see, e.g. Maroudas, 2000; Lu and Kaxiras, 2004). These computational limitations are a major stumbling block in current complex system modeling.

We will describe a computational approach for dealing with any complex, multiscale system whose collective, coarse-grained behavior is *simple* when we know in principle how to model such systems at a very fine scale (e.g., through molecular dynamics). We assume that we do not know how to write good *simple* model equations at the right coarse-grained, macroscopic scale for their collective, coarse-grained behavior. We will argue that, in many cases, the derivation of macroscopic equations can be circumvented; that by using short bursts of appropriately initialized microscopic simulation one can effectively solve the macroscopic equations *without ever writing them down*. A direct bridge can be built between microscopic simulation (e.g., kinetic Monte Carlo, agent-based modeling) and traditional continuum numerical analysis. It is possible to enable microscopic simulators to directly perform macroscopic, systems-level tasks. The main idea is to consider the microscopic, fine-scale simulator as a (computational) experiment that one can set up, initialize, and run at will. The results of such appropriately designed, initialized, and executed brief computational experiments allow us to *estimate* the same information that a macroscopic model would allow us to *evaluate* from explicit formulas.

The heart of the approach can be conveyed through a simple example. Consider a single, autonomous ordinary differential equation,

$$\frac{dc}{dt} = f(c).$$

Think of it as a model for the dynamics of a reactant concentration in a stirred reactor. Equations like this embody “practical determinism” as discussed above: given a finite amount of information—the state at the present time, *c(t=0)*—we can predict the state at a future time. Consider how this is done on the computer using, for illustration, the simplest numerical integration scheme, forward Euler:

$$c_{n+1} \equiv c([n+1]\tau) = c_n + \tau f(c_n).$$

Starting with the initial condition, $c_0$, we go to the equation and *evaluate* $f(c_0)$, the time derivative, or slope of the trajectory *c(t)*; we use this value to make a prediction of the state of the system at the next time step, $c_1$. We then repeat the process: go to the equation with $c_1$ to *evaluate* $f(c_1)$ and use the Euler scheme to predict $c_2$; and so on. Forgetting for the moment accuracy and adaptive step-size selection, consider how the equation is used: given the state we *evaluate* the time derivative; and then, using mathematics (in particular, Taylor series and smoothness to create a local linear model of the process in time) we make a prediction of the state at the next time step. A numerical integration code will “ping” a subroutine with the current state as input and will obtain as output the time-derivative at this state. The code will then process this value and use local Taylor series in order to make a prediction of the next state (the next value of *c* at which to call the subroutine evaluating the function *f*). Three simple things are important to notice. First, the *task* at hand (numerical integration) does not need a closed formula for *f(c)*—it only needs *f(c)* evaluated at a particular sequence of values $c_n$. Whether the subroutine evaluates *f(c)* from a single-line formula, uses a table lookup, or solves a large subsidiary problem, from the point of view of the integration code it is the same thing. Second, the sequence of values $c_n$ at which we need the time derivative evaluated is not known *a priori*.

It is generated as the task progresses, *from processing results of previous function evaluations* through the Euler formula. We know that protocols exist for designing experiments to accomplish tasks such as parameter estimation (Box et al., 1978). In the same spirit, we can think of the Euler method, and of explicit numerical integrators in general, as *protocols for specifying where to perform function evaluations* based on the task we want to accomplish (computation of a temporal trajectory). Last, the form of the protocol (the Euler method here) is based on mathematics, particularly on smoothness and Taylor series. The trajectory is locally approximated as a linear function of time; the coefficients of this function are obtained from the model using *function evaluations.*

Suppose now that we do not have the equation, but *we have the experiment itself.* We can fill up the stirred reactor with reactant at concentration $c_0$, run for some time, and record the time series of $c(t)$. Using the results of a short run (over, say, one minute), we can now *estimate* the slope, $dc/dt$ at $t=0$, and predict (using the Euler method) where the concentration will be in, say 10 minutes. Now, instead of waiting for nine minutes for the reactor to get there, we stop the experiment and immediately start a new one: reinitialize the reactor *at the predicted concentration*, run for one more minute, and use forward Euler to predict what the concentration will be 20 minutes down the line. We are substituting short, appropriately initialized experiments, and *estimation* based on the experimental results, for the function evaluations that the subroutine with the closed form $f(c)$ would return. We are in effect doing forward Euler again, but the coefficients of the local linear model are obtained using experimentation *"on demand"* (Cybenko, 1996) rather than function evaluations of an *a priori* available model.

Now we complete the argument. Suppose that the inner layer is not a laboratory experiment, but a *computational* one, with a model at a different, much finer level of description, for the sake of the discussion, a lattice kinetic Monte Carlo (kMC) model of the reaction. Instead of running the kMC model for long times, and *observing* the evolution of the concentration, we can exploit the procedure described above, perform only short bursts of appropriately initialized microscopic simulation, and use their results to evolve the macroscopic behavior over hopefully much longer time scales. It is much easier to initialize *a code* at will—a computational experiment—as opposed to initializing a new laboratory experiment. Many new issues arise, notably noise in the form of fluctuations, from the microscopic solver. The conceptual point, however, remains: even if we do not have the right macroscopic equation for the concentration, we can still *perform its numerical integration* without obtaining it in closed form. The skeleton of the wrapper (the integration algorithm) is the same one we would use if we had the macroscopic equation, but now function evaluations are substituted by short computational experiments with the microscopic simulator, whose results are appropriately processed for local macroscopic identification and estimation. If a large separation of time scales exists between microscopic dynamics (here, the time we need to run kinetic Monte Carlo to estimate *dc/dt*) and the macroscopic evolution of the concentration, this procedure may be significantly more economical than direct simulation.

Passing information between the microscopic and macroscopic scales at the beginning and the end of each computational experiment is a vitally important issue. It is accomplished through a *lifting operator* (macro- to micro-) and a *restriction operator* (micro- to macro-) as discussed below (Theodoropoulos et al., 2000; Kevrekidis et al*.,* 2003 and references therein). The proposed computational methodology consists of the following basic elements:

(a) 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 density and momentum (zeroth and first moment of the particle distribution over velocities), and we might choose to discretize them in a computational domain via finite elements. We call this the macroscopic description, $u$. These choices suggest possible *restriction* operators, $M$, from the microscopic-level description $U$, to the macroscopic description: $u = MU$;

(b) Choose an appropriate *lifting* operator, $\mu$,from the macroscopic description, $u$, to one or more consistent microscopic descriptions, $U$. For example, in a gas simulation using pressure etc. as the macroscopic-level variables, $\mu$ could make random particle assignments consistent with the macroscopic statistics. $\mu M=I$, i.e. lifting from the macroscopic to the microscopic and then restricting (projecting) down again should have no effect, except roundoff;

(c) Start with a macroscopic condition (e.g. concentration profile) $u(t_0)$;

(d) Transform it through lifting to one or more fine, *consistent* microscopic realizations $U(t_0) = \mu u(t_0)$;

(e) Evolve this(ese) realization(s) using the microscopic simulator for the desired short macroscopic time $T$, generating the value(s) $U(T)$;

(f) Obtain the restriction(s) $u(T)=MU(T)$ (and average over them).

This constitutes the *coarse time-stepper*, or *coarse time-T map.* If this map is accurate enough, we discussed above how to use it in a two-tier procedure to perform *Coarse Projective Integration* (Gear and Kevrekidis, 2003, Gear, 2001, Gear et al., 2002) (see Figure 1a for a schematic). Coarse projective integration and also coarse bifurcation computations (see Figure 1b) have been used to accelerate lattice kinetic Monte Carlo simulations of catalytic surface reactions (Makeev et al., 2002a, 2002b; Rico-Martinez et al., 2004), Brownian dynamics simulations of nematic liquid crystals (Siettos et al., 2003), and much more.

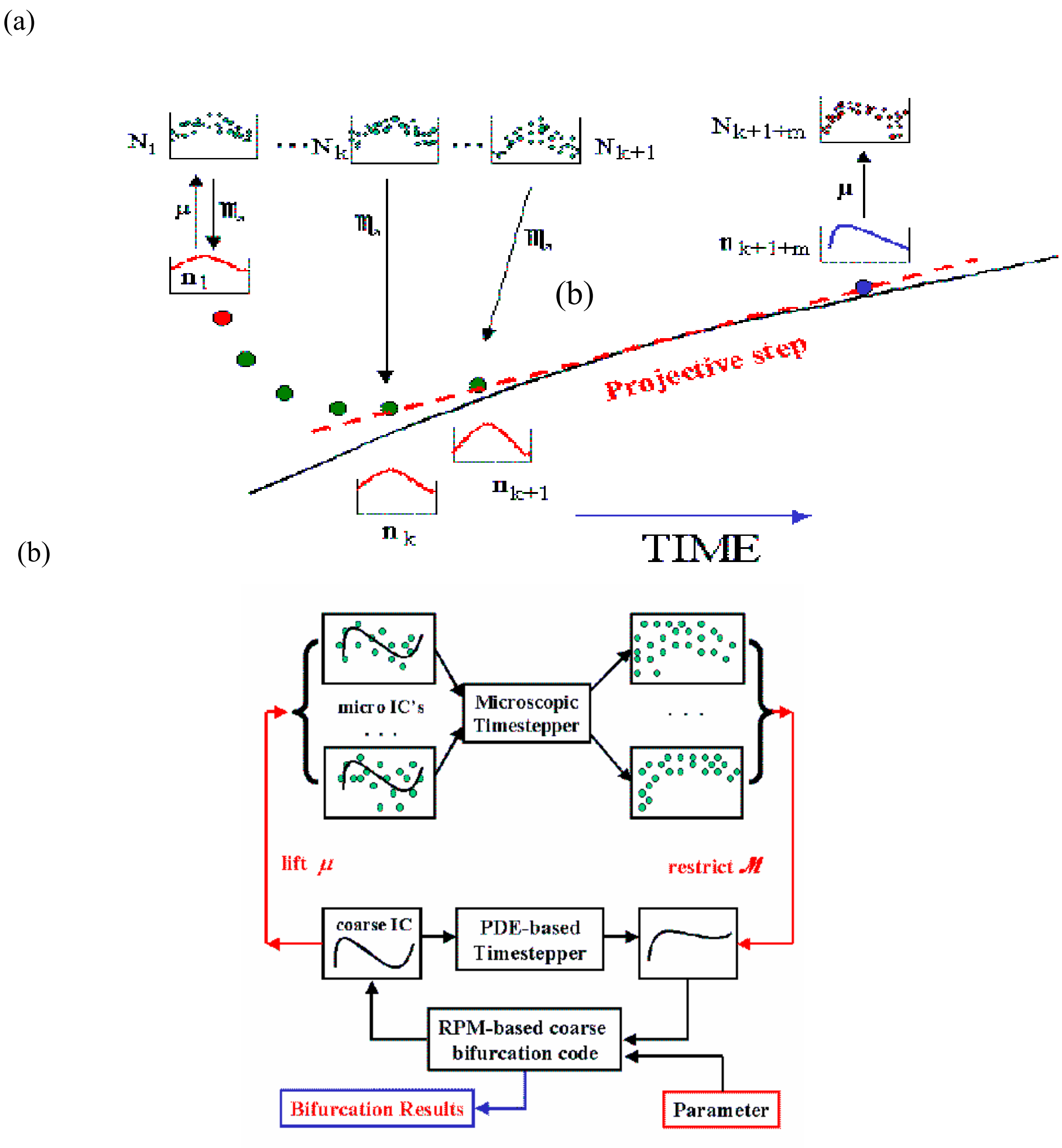

Figure 1. Schematic illustrations of (a) coarse projective integration;
(b) coarse timestepper-based bifurcation computations (see text).

Timestepper based methods are, in effect, alternative ensembles for performing microscopic (molecular dynamics, kMC, Brownian dynamics) simulations. Innovative multiscale/multilevel techniques proposed over the last decade that can be integrated in an equation-free, timestepper-based framework include the quasi-continuum methods of Phillips and coworkers (Phillips, 2001; Ortiz and Phillips, 1999) and the optimal prediction methods of Chorin and coworkers (Chorin et al., 1998; 2000) (see the discussion in Kevrekidis et al., 2003). If one has good macroscopic equations, one should use them. But when these equations are not available in closed form, and such cases arise with increasing frequency in contemporary modeling, the equation-free computational enabling technology we outlined here may hold the key to the engineering of *effectively simple* systems.

## ACKNOWLEDGMENTS

This work was partially supported over the years by AFOSR, through an NSF/ITR grant, DARPA, and Princeton University. An extended version of this article has appeared as a Perspective in the July 2004 issue of the *AIChE Journal* (with C. W. Gear and G. Hummer as co-authors).

## REFERENCES

Box, G.E.P., W. Hunter, and J.S. Hunter. 1978. Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building. Indianapolis, In: Wiley-Interscience.

Chorin, A., A. Kast, and R. Kupferman. 1998. Optimal prediction for underresolved dynamics. Proceedings of the National Academy of Sciences 95(8): 4094–4098

Chorin, A., O. Hald, and R. Kupferman. 2000. Optimal prediction and the Mori-Zwanzig representation of irreversible processes. Proceedings of the National Academy of Sciences 97(7): 2968–2973.

Cybenko, G. 1996. Just in Time Learning and Estimation. Pp. 423-434 in Identification, Adaptation and Learning: The Science of Learning Models from Data, NATO Advanced Studies Institute (ASI) Series F153, edited by S. Bittanti and G. Picci. Berlin: Springer-Verlag.

Gear, C.W. 2001. Projective Integration Methods for Distributions. NEC Technical Report 2001-130.

Gear, C.W., and I.G. Kevrekidis. 2003. Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. Society for Industrial and Applied Mathematics (SIAM) Journal on Scientific Computing 24(4): 1091-1106.

Gear, C.W., I.G. Kevrekidis, and K. Theodoropoulos. 2002. Coarse integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods. Computer and Chemical Engineering 26(7-8): 941–963.

Kevrekidis, I.G., C.W. Gear, J.M. Hyman, P.G. Kevrekidis, O. Runborg, and K. Theodoropoulos. 2003. Equation-free coarse-grained multiscale computation: enabling microscopic simulators to perform system-level tasks. Communications in Mathematical Sciences 1(4): 715–762.

Lu, G., and E. Kaxiras. 2004. An overview of multiscale simulations of materials. Available online at: <*http://arxiv.org/abs/cond-mat/0401073*>.

Makeev, A., D. Maroudas, and I.G. Kevrekidis. 2002. Coarse stability and bifurcation analysis using stochastic simulators: kinetic Monte Carlo examples. Journal of Chemical Physics 116(23): 10083–10091.

Makeev, A.G., D. Maroudas, A.Z. Panagiotopoulos, and I.G. Kevrekidis. 2002. Coarse bifurcation analysis of kinetic Monte Carlo simulations: a lattice gas model with lateral interactions. Journal of Chemical Physics 117(18): 8229–8240.

Maroudas, D. 2002. Multiscale modeling of hard materials: challenges and opportunities for chemical engineering. American Institute of Chemical Engineers Journal 46(5): 878–882.

Ortiz, M., and R. Phillips. 1999. Nanomechanics of defects in solids. Advances in Applied Mechanics 36: 1–79.

Phillips, R. 2001. Crystals, Defects and Microstructures. Cambridge, U.K.: Cambridge University Press.

Rico-Martinez, R., C.W. Gear, and I.G. Kevrekidis. 2004. Coarse projective KMC integration: forward/reverse initial and boundary value problems. Journal of Computational Physics 196(2): 474–489.

Siettos, C., M.D. Graham, and I.G. Kevrekidis. 2003. Coarse Brownian dynamics for nematic liquid crystals: bifurcation, projective integration and control via stochastic simulation. Journal of Chemical Physics 118(22): 10149–10156.

Theodoropoulos, K., Y.-H. Qian, and I.G. Kevrekidis. 2000. Coarse stability and bifurcation analysis using timesteppers: a reaction diffusion example. Proceedings of the National Academy of Sciences 97(18): 9840–9843.