A multiscale method for highly oscillatory dynamical systems using a Poincaré map type technique

G. Ariel[*], B. Engquist[†], S. Kim[†], Y. Lee[†], and R. Tsai[†]

Abstract

We propose a new heterogeneous multiscale method (HMM) which is devised to compute the effective behavior of a class of highly oscillatory ordinary differential equations (ODEs). Without the need for identifying hidden slow variables, the proposed method is constructed based on the following three ideas: a nonstandard splitting of the vector field (the right hand side of the ODEs); comparison of the solutions of the splitted equations; consrtuction of effective paths in the state space whose projection onto the slow subspace has the correct dynamics; and a novel on-the-fly filtering technique for achieving a high order accuracy. Numerical examples are given.

1 Introduction

In many application the preservation of the long-time behavior of the flow is more important than the approximation of the trajectory or a particle itself. Even for a numerical scheme with a high order accuracy, the preservation of invariance does not hold automatically. Thus, one of the major thrusts is in developing numerical methods that allow long time computation of oscillatory solutions to Hamiltonian systems. These methods typically attempt to approximately preserve some analytical invariance of the solutions: e.g. the total energy of the system, symplectic structures, or the reversibility of the flow. Detailed reviews and further references on this active field of geometric integration can be found in [12, 29, 30, 37],[28, 36, 40], and [15]. Another approach based on asymptotic expansions in inverse powers of the oscillatory parameter is given in [16] and references within.

Here we develop computational methods for a class of highly oscillatory ODEs, including non-Hamiltonian systems that are not covered by the standard geometric integrators or the classical methods for stiff equations. We focus on problems for which the range of scales is so large that the finest scale cannot be resolved over the entire computational domain. Gear and Kevrekidis introduced such a technique in [25] and applied it successfully to dissipative problems.We will concentrate on highly oscillatory problems and build on the framework of E and Engquist [17].

In this paper, we tackle this difficulty by exploiting separation of scales. It requires that enough information about the fast scale influence on the slow scale dynamics can be obtained by performing localized simulations over short times, and thereby achieve better efficiency. The numerical complexity of these methods is therefore much smaller than direct simulations of the given systems that induce the fine scale dynamics. This feature exists implicitly in the classical stiff solvers as the resulting algebraic systems of equations are solved by efficient nonlinear solvers exploiting special structures. In the envelope methods [38] for highly oscillatory problems fast oscillations are sampled cleverly in order to extrapolate in a much larger time scale. Similar techniques are also used in stochastic

[*] Gil Ariel
Bar-Ilan University, Ramat Gan, 52900, Israel email:arielg@math.biu.ac.il

[†] Bjorn Engquist · Seong Jun Kim · Yoonsang Lee · Richard Tsai
Department of Mathematics and Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, TX 78712, USA email: engquist@ices.utexas.edu, skim@math.utexas.edu, ylee@math.utexas.edu, ytsai@math.utexas.edu

differential equations [18]. In a sequence of papers, [1, 3, 4, 5, 6, 21], we have introduced and developed multiscale algorithms which use a set of slow variables for computing the effective behavior of a highly oscillatory dynamical system. The set of slow variables can be either analytically derived, or numerically determined. In addition to our previous work, other approaches to find slow variables includes, e.g. [8, 9].

The requirements of explicit form of slow variables is no longer needed in the methods proposed in this paper. Instead, some assumptions are placed on the vector fields defined by the dynamical systems. The originally given dynamical system is thought of as having a lower order perturbation. By ignoring a lower order perturbative part of the vector field, an "unperturbed" dynamical system is defined. It is assumed that the dynamic of the "unperturbed system" is ergodic and yields an invariant measure on a manifold. The essential part of our new algorithm is to run the original and the "unperturbed" systems from the same initial conditions for short time intervals, and compare the resulting solutions. See Figure 2 for an illustration. Furthermore, by reversing the "unperturbed problems, the "unperturbed" system yields new initial data from which a new comparison of the two dynamical systems can be performed as described earlier. We then construct locally an effective path along which the evolutions of the original system's slow variables are accurately captured. Moreover, our algorithm achieve higher order accuracy by connecting these ideas with a novel filtering technique.

We remark that our method shares some formal procedural similarity to many algorithms that use splitting techniques, but ours is fundamentally different from these algorithms. The impulse method [24] and FLAVORS [41] are two of the well-known methods using splitting technique. The impulse method for Newtonian dynamics splits the force field into fast and slow parts. As with the conventional splitting strategy, the basic impulse algorithm then integrates alternately in time intervals of equal length, the fast equation and the slow equation; the integration for the fast equation starts with the solution generated by the slow equation, and vice versa. In the context of impulse method, it is assumed that the slow part of the vector field comes from long range interaction potential and is relatively costly to evaluate compared to the fast force. In this paper, the focus is on bypassing the need to evaluate the fast forces in time intervals that are asymptotically longer than the shortest periods in the oscillations, which is also dealt with in FLAVORS. FLAVORS integrate the whole system and the splitted non-stiff system alternately as well. More precisely, during the integration, stiff forces in the given system are "turned on" over a microscopic time interval time and then "turned off" over a mesoscopic time step. As in the impulse method, the solution produced at the end of a microscopic interval by "turning on" the stiff force in the system is continued by serving as the initial condition for the whole system in the following mesoscopic time integration.

We emphasize that while our algorithm does require splitting of the vector field, our ultimate numerical solutions are not constructed from continuing alternately the solutions computed by the splitted equations. Furthermore, we point out that both the impulse methods and FLAVORS are lower order methods. In essence FLAVORS can be regarded as a Monte-Carlo type algorithm while our method is deterministic in nature and does compute higher order accurate solutions.

The layout of the paper is as follows. In the remaining of this section, we present the basic ideas in designing an HMM. In Section 2, we present our new algorithm for tracking slow variables. This is based on splitting the original equation (1.1) into stiff and non-stiff parts. In order to achieve high order accuracy, a novel on-the-fly filtering technique is introduced in Section 2.1.2. Section 3 presents several numerical examples. Here we compare our method with existing methods, e.g., FLAVORS. We conclude in Section 4.

1.1 The HMM framework

We consider the computations of the effective long time properties of a class of dynamical system, formally written in the form

$$\frac{d}{dt}u = \epsilon^{-1} f_1(u) + f_0(u), \tag{1.1}$$

with initial condition $u(0) = u_0 \in \mathcal{D}_0 \subset \mathbb{R}^d$. It is assumed that a unique bounded solution exists in a time segment $I = [0, T]$. In many examples, it is not clear how to characterize the slow parts of the dynamics in systems (1.1). To this end we define slow variables as below.

Significant amplication of numerical errors occurs when a classical integrator is applied to approximate the long-time behavior of (1.1) The accuracy and stability requirements of the integrator dictate the use of a time step of order ϵ due to the stiff part. This fact implies that the computational complexity for (1.1) over a fixed time T is at least of the order of ϵ^{-1}. However, in many situations, one is interested only in a set of slowly changing quantities U that are derived from the solutions of the given stiff system (1.1). In the case where U constitute of a set of functions of u, they are commonly referred to as slow variables of the system. See for example [2, 3, 23, 26, 33, 34, 35]. For example, U could be the averaged kinetic energy of a particle system u. Formally, slow variables of a dynamical system can be defined as below.

Definition 1. Let $u(t) \in \mathcal{D}_0$ denote the solution of (1.1) for some initial conditions. A smooth function $a(t)$ is to be **slow** if $|da/dt| \leqq C$ for some constant C independent of ϵ in $t \in I$. Moreover, a smooth function $\xi(u) : \mathcal{D}_0 \to \mathbb{R}$ is called a **slow variable** with respect to $u(t)$ if $\xi(t) = \xi(u(t))$ is slow.

Our objective is to construct and analyze ODE solvers that integrate the system

$$\frac{d}{dt}U = F(U, D), \tag{1.2}$$

where D is the data that can be computed by solving (1.1) locally in time. U may be some function or functional of u, and is called the macroscopic variable. U typically describes some effective behavior of (1.1) that is of relevance to the application.

If F is well-defined and has convenient explicit mathematical expression, then there is no need to solve the stiff system (1.1); one only needs to solve (1.2). In many situations, the dependence of U on u is not explicitly available. Our proposed strategy involves setting up a formal numerical discretization for (1.2), and evaluate F from short time histories of u with properly chosen initial conditions.

We will generalize the scope of this type of algorithms by providing a more general systematic analysis that is applicable to a much wider class of applications that includes some systems from molecular dynamics. In the HMM framework [1, 3, 21], one assumes a macroscopic model

$$F(U, D) = 0, \;\; U \in \Omega_{(M)} \tag{1.3}$$

which may not be explicitly given, but can be evaluated from a given microscopic model,

$$f(u, d) = 0, \;\; u \in \Omega_{(m)} \tag{1.4}$$

where u are the microscopic variables. $D = D(u)$ and $d = d(U)$ denote the set of data or auxiliary conditions that further couple the macro- and microscopic models. Model (1.3) is formally discretized at a macroscopic scale, and the adopted numerical scheme dictates when the necessary information $D(u)$ should be acquired from solving (1.4), locally on the microscopic scale with auxiliary conditions $d(U)$. As part of $d(U)$ and $D(u)$, the macro- and microscopic variables are related by reconstruction and compression operators:

$$\mathcal{R}(U, D_R) = u, \;\; \mathcal{Q}(u) = U, \;\; \mathcal{Q}(\mathcal{R}(U, D_R)) = U, \tag{1.5}$$

where D_R are the needed data that can be evaluated from u. Errors of this type of schemes generally take the structure

$$\text{Error} = E_H + E_h + E_{HMM},$$

where E_H is the error of the macroscopic model (1.3), E_h is the errors from solving (1.4), and E_{HMM} contains the errors in the the multiscale model, including the passing of information through $\mathcal{R}$ and

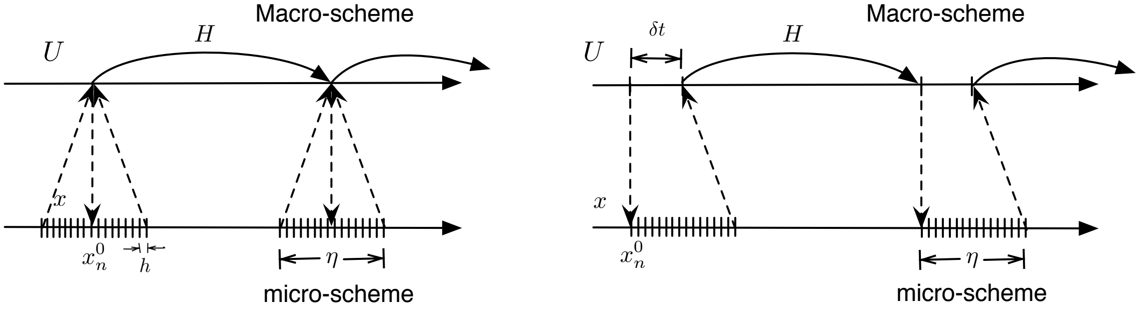


Figure 1: Two typical structures of the proposed multiscale algorithm. The structure on the left is for dissipative problems.

$\mathcal{Q}$. This approach has been used in a number of applications, such as contact line problems, epitaxial growth, thermal expansions, combustion, reviewed in [43], and homogenizations of wave propagation in long time intervals [20], and coupling network models for macroscopic multiphase flows in porous media [13, 14].

Figure 1 shows two typical structures of such algorithms. In our context, an ODE solver for U lies on the upper axis and constructs approximations of U at the grid points depicted there. The fine meshes on the lower axis depict the very short evolutions of (1.1) with initial values determined by $\mathcal{R}(U(t_n))$. The reconstruction operator then takes each short time evolution of u and evaluates F and U. The algorithms in [19],[25], and [38] are also of a similar structure. As a simple example, the forward Euler scheme applied to (1.2) would appear to be

$$U_{n+1} = U_n + H \cdot F(U_n), \tag{1.6}$$

where F contains the passage of $\mathcal{Q}\Phi_t\mathcal{R}(U_n)$; reconstruction $\mathcal{R}$, evolution Φ_t, and compression $\mathcal{Q}$, and H is the step size. If each evolution Φ_t of the full scale system (1.1) is reasonably short, the overall complexity of such type of solvers would be smaller than solving the stiff system (1.1) for all time, thereby gaining computational efficiency.

Essential questions that need to be resolve for each scheme include:

- If only the microscopic model is given, how to systematically derive a corresponding macroscopic model for the application in question? What are $\mathcal{R}$ and $\mathcal{Q}$?
- With the system for u, and a choice of $U(u)$, is F well-defined by the procedure defined above? If not, how can it be properly defined?
- How long should each evolution be computed?
- What do consistency, stability. and convergence mean?

For a fixed $\epsilon > 0$, all well known methods, assuming $U = u$, will converge as the step-size $H \to 0$, and there is no difference between stiff and non-stiff problems. In the related work [1, 3, 21], convergence for very stiff problems ($\epsilon \ll H$) is defined by following error:

$$E(H) = \max_{0 \le t_n \le T} \Big(\sup_{0 < \epsilon < \epsilon_0(H)} |U(t_n) - U_n| \Big). \tag{1.7}$$

Here, $\epsilon_0(H)$ is a positive function of H, serving as an upper bound for the range of ϵ, and $U(t_n)$ and U_n denote respectively the analytical solution and the discrete solution at t_n. With this notion, it is clear that a sensible method has to utilize the slow varying property of U and generate accurate approximation with a complexity sublinear to ϵ^{-1}.

1.2 Slow variables and invariant measures

We start by considering a class of system that has an explicit slow-fast grouping in the solution's components:

$$\begin{cases} \frac{d}{dt}x &= \epsilon^{-1}f(x,z,t)+g(x,z,t), \\ \frac{d}{dt}z &= h(x,z,t). \end{cases} \tag{1.8}$$

Here the x components are real valued and stay bounded but are highly oscillatory, and the z components are called the slow variables in the system, since their time derivatives are formally bounded. If for fixed z, $x(t)$ yields an invariant measure μ supported on some manifold $\mathcal{M}(z)$ of the same dimension as that of x, then $z(t)$ can be consistently approximated in any constant time interval by an averaged equation

$$\frac{d}{dt}\bar{z} = \bar{h}(\bar{z},t) := \int_{\mathcal{M}(\bar{z})} h(x,\bar{z},t)d\mu(x,\bar{z}). \tag{1.9}$$

Such systems are widely studied to build multiscale scale methods. See [19][42]. In this case, it is reasonable to use $\bar{z}$ as the macroscopic variable; i.e. $U=\bar{z}\simeq z$, and $\mathcal{R}(U,D_R)=(x^*,z)$ where D_R gives the value $x^*\in\mathcal{M}$. For example, x^* may be taken from the x values in the previous microscale simulation. The compression $\mathcal{Q}$ may simply be $\mathcal{Q}(x,z)=z$. The operator F in (1.6) plays the role of approximating the average right hand side by time averaging the microscopic evolution using a suitable filtering kernel. From the computational point of view, averaging methods inspire efficient numerical schemes for integrating the slow components of slow-fast systems without fully resolving all fast oscillations.

However, if there are resonances among the oscillations, $x(t)$ is likely not to remain on any invariant manifold [34], and more sophistication in the algorithm is needed. One can see the essence of this problem from the simple example,

$$\begin{cases} \frac{d}{dt}x &= i\epsilon^{-1}x+g(x), \;\; x(0)=1, \\ \frac{d}{dt}z &= h(x), \;\; z(0)=z_0. \end{cases} \Longrightarrow \begin{cases} \frac{d}{dt}w &= e^{-\frac{i}{\epsilon}t}g(e^{\frac{i}{\epsilon}t}w), \;\; x(t)=e^{\frac{i}{\epsilon}t}w(t), \\ \frac{d}{dt}z &= h(e^{\frac{i}{\epsilon}t}w), \;\; z(0)=z_0. \end{cases}$$

Let us formally decompose $e^{-\frac{i}{\epsilon}t}g(e^{\frac{i}{\epsilon}t}w)=\bar{g}(w)+\alpha(e^{\frac{i}{\epsilon}t},w)$, where $\bar{g}$ does not depend on any fast oscillations but α has only fast oscillations. Resonance in this system corresponds to the case where $\bar{g}$ is non-zero. If $\bar{g}\equiv 0$, $w(t)$ stays close to 1, due to the strong self-averaging in α. Thus (1.9) corresponds to averaging h over the unit circle, and $d\mu$ is the arc-length element. Consequently, the averaging has to be performed with the correct measure

$$\frac{d}{dt}\bar{z} = \bar{h}(\bar{z},t) := \int_{\mathcal{M}(t)} h(x,\bar{z},t)d\mu(x,\bar{z};t).$$

For example, if $g(x)=x$, then $\bar{g}(w)=w$, and $\alpha\equiv 0$. Consequently, $\mathcal{M}(t)$ is a circle with radius equal to $w(t)=\exp(t)$. Without the knowledge of $w(t)$, it is impossible to define a consistent reconstruction operator $\mathcal{R}$, and consequently, it is impossible to build a convergent multiscale algorithm. In some literature, the issue caused by resonance is referred to as the system having hidden slow variables [1, 23, 21],[41]. It is essential that a multiscale method computes accurately the effect of the hidden slow variables.

We continue our discussion using the previous example, but instead, we rewrite the equation for x as a system in $\mathbb{R}^2$:

$$\begin{cases} \dot{x}_1 &= \epsilon^{-1}x_2+x_1, \\ \dot{x}_2 &= -\epsilon^{-1}x_1+x_2, \end{cases}$$

with initial conditions $(x_1(0),x_2(0))=(0,1)$. Thus $(x_1(t),x_2(t))=(e^t\sin\epsilon^{-1}t, e^t\cos\epsilon^{-1}t)$.

Taking $I=x_1^2+x_2^2$, we notice that I has a bounded derivative along the trajectory of the solution; i.e., $\dot{I}:=(d/dt)I(x_1(t),x_2(t))=2I$ is independent of ϵ. Since I does not appear explicitly in the given ODEs, it is regarded as a hidden slow variable. For this particular example one can easily solve for I, $I(t) = I(0)e^{2t}$. In fact, the uniform bound on $\dot{I}$ indicates the slow nature of $I(x_1(t), x_2(t))$ when compared to the fast oscillations in $(x_1(t), x_2(t))$. This type of characterization of the effective dynamics of highly oscillation systems are commonly used in the literature. In this example it was easy to find the slow quantity I.

We refer the reader to [1, 3] for designing multiscale algorithms that compute the effective behavior of highly oscillatory dynamical systems by using slow variables. [3] shows how we identify polynomial slow variables in (1.1) and how we use the solution of (1.1) locally in time to approximate an assumed effective equation. In particular, for autonomous systems, a diffeomorphism $\Psi : u \to (\xi(u), \phi(u))$ of (1.1) from $\mathbb{R}^d$ onto $\mathbb{R}^{d-1} \times S^1$ is constructed so that

$$\begin{cases} \dot{\xi} = g_0(\xi, \eta), & \xi(0) = \xi_0, \\ \dot{\eta} = \epsilon^{-1} g_1(\xi, \eta) + g_2(\xi, \eta), & \eta(0) = \eta_0 \end{cases} \tag{1.10}$$

and all smooth slow variables depends on ξ up to some bounded lower order perturbative terms. Consequently, this set of slow variables characterizes the slowly changing effective behavior of the trajectory of the given oscillatory dynamical system. Typically, one may expect that the values of a slow variable ξ along the dynamical system's solutions, $\xi \circ u$, converge as $\epsilon \to 0$. We shall denote this limit as $\bar{\xi}(t; u_0)$. This expectation may come directly from the averaging theory [39].

In designing multiscale algorithms for this type of problems, it is often convenient to aim at constructing accurate approximation of $\bar{\xi}$ by suitable filtering of the oscillations in $u(t)$. This typically involves numerically averaging over the fast oscillations in the system. The resulting HMMs are quite efficient as reported in our previous work. Nevertheless, for large systems, analytical or numerical determination of a suitable close set of slow variables whose dynamics are closed along $u(t)$ can be difficult. The purpose of this paper is to introduce a new type of HMMs which do not use slow variables in the computation.

2 The BF HMM scheme

We consider the computation of the effective long time properties of a class of dynamical system, formally written in the form

$$\frac{d}{dt} x^\epsilon = \epsilon^{-1} f_1(x^\epsilon) + f_0(x^\epsilon, t; \epsilon), \tag{2.1}$$

with initial condition $x^\epsilon(0) = x_0 \in \mathcal{D}_0 \subset \mathbb{R}^d$.

Assumption 1. *The trajectories of the unperturbed equation*

$$\frac{d}{dt} y = \epsilon^{-1} f_1(y), \tag{2.2}$$

are ergodic on some invariant manifold $\mathcal{M}(y_0)$, where $y(0) = y_0$ is the initial condition. Furthermore, for points in $\mathcal{D}_0$, the Jacobian of f_1 has only purely imaginary eigenvalues bounded away from 0, independent of ϵ.

Assumption 2. *The invariant manifold $\mathcal{M}$ of (2.2) can be identified by the level sets of $\xi_1, \xi_2, \cdots, \xi_k$ with $k < d$ which are slow variables with respect to $x^\epsilon(t)$.*

Thus, for each time t, we may identify the manifold

$$\mathcal{M}(t) = \cap_{j=1}^{k} \{ z \in \mathbb{R}^d : \xi_j(z) = \xi_j \circ x^\epsilon(t) \},$$

and if we solve (2.1) and (2.2) with the same initial condition lying on $\mathcal{M}(t)$, it is then possible to track $\mathcal{M}(t)$ by comparison of $x^\epsilon(t)$ and $y(t)$ without explicitly knowing the slow variables. Thus the evolution of the slow variables, or equivalently, that of $\mathcal{M}(t)$, can be tracked at least locally in state space by a path $\gamma(s)$ which crosses $\mathcal{M}(t)$ for at $s = t$. Note that such γ is not unique and we shall construct one in the state space such that for any slow variable ξ, and finite time interval, γ satisfies the following conditions:

1. (Consistency) $\xi \circ \gamma(t) = \xi \circ x^{\epsilon}(t)$;
2. (Effectiveness) $\left|\frac{d^{(j)}\gamma}{dt^{(j)}}\right| \leq C$, for $1 \leq j \leq k$ for some positive integer k.

We shall refer γ as an effective path of the given dynamical system.

It has been observed in [3] that such a path can be constructed using an effectively closed system of explicitly identified slow variables. Furthermore, the constructed path is orthogonal to the level sets of the slow variables in the limit as $\epsilon \to 0$. *Our new algorithm does not require explicit form of any slow variables.* Instead, our new algorithm compares short time solutions of (2.1) and (2.2) to generate a sequence of points whose interpolation defines an approximation of γ. In the following algorithms, γ is not necessarily orthogonal to the level sets of slow variables. As we shall see further below, the more sophisticated form of our algorithm requires both the forward and backward in time solutions of (2.1) and (2.2). Thus, we shall call our algorithms BF HMMs for brevity.

Our basic algorithm is illustrated in Figure 2 and summarized below. This first algorithm does not involve any solution of the equations involved backward in time, but we shall still call it a BF HMM. We remark here that Algorithm 1 described below shares a similar strategy is that proposed in [7] for a different problem.

Algorithm 1. *(Forward Euler BF HMM)*

1. (Forward Euler macro-solver) Compute γ_{n+1} from γ_n at $t_n = nH$.
$$\gamma_{n+1} = y_n(\Delta) + HF_n,$$
where
$$F_n := \frac{x_n(\Delta) - y_n(\Delta)}{\Delta},$$
and $x_n(\Delta)$ and $y_n(\Delta)$ are evaluated from the micro-solver.
2. (Micro-solver) At $t_n = nH$, solve
$$\frac{d}{dt}x_n = \epsilon^{-1}f_1(x_n) + f_0(x_n), \quad x_n(0) = \gamma_n,$$
and
$$\frac{d}{dt}y_n = \epsilon^{-1}f_1(y_n), \quad y_n(0) = \gamma_n,$$
for $t \in [t_n, t_n + \Delta]$ with $0 < \epsilon \ll \Delta \ll H$.
3. Repeat.

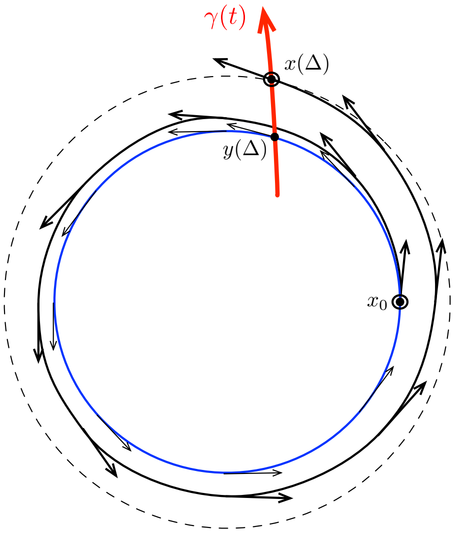


Figure 2: $\gamma(\Delta) := x(\Delta)$,$\gamma(0) := y(\Delta)$, and $\gamma(-\Delta) := x(-\Delta; y(\Delta))$.

Example 1. Our simple example to demonstrate the consistency of the Forward Euler BF HMM is an expanding spiral [1] in $\mathbb{C}$.

$$\frac{d}{dt}x^\epsilon = i\epsilon^{-1}x^\epsilon + x^\epsilon, \quad x^\epsilon(0) = x_0 \tag{2.3}$$

with $x_0 > 0$ independent of ϵ. We transform x^ϵ into (ξ, θ) where $\xi = |x^\epsilon|$ and $\theta = \arg(x^\epsilon)$, and obtain

$$\begin{cases} \dot{\xi} = \xi, & \xi(0) = |x_0|, \\ \dot{\theta} = \epsilon^{-1}, & \theta(0) = \arg(x_0). \end{cases} \tag{2.4}$$

By Definition **1**, ξ is a slow variable. In Step 1, we assume that all micro simulations of x and y are exact over $[t_n, t_n + \Delta]$, $\Delta \ll H$. Then the local truncation error in approximating a slow variable ξ is given by

$$\begin{aligned} \left|\xi \circ x^\epsilon(t_{n+1}) - \xi \circ \left(y_n(\Delta) + H \cdot \frac{x_n(\Delta) - y_n(\Delta)}{\Delta}\right)\right| &= \left|e^{t_n+H} - e^{t_n}\left(1 + H \cdot \frac{e^\Delta - 1}{\Delta}\right)\right| \\ &= \left|e^{t_n}\left(\frac{H^2 - \Delta \cdot H}{2} + \cdots\right)\right| \\ &\leq CH^2 \end{aligned}$$

for some positive constant C. Thus to leading order in H^2, Forward Euler BF HMM yields a correct $\gamma(t)$ for the slow variable ξ.

2.1 Higher order schemes

In this section we describe the construction of high order accurate BF HMMs which has sublinear complexity in constant time intervals. In Algorithm 1, forward Euler scheme is used to compute the effective path γ that passes through $y(\Delta)$. A lower order approximation of $d\gamma/dt$ when it crosses $y(\Delta)$ is approximated by F_n, which is a linear approximation. Thus, higher order BF HMMs require higher order approximation of $d\gamma/dt$. In order to do that, we systematically solve (2.1) and (2.2) both forward and backward in time to obtain points lying on an effective path that crosses a chosen point. Below, we outline this general procedure:

- The chosen macroscopic integrator is used to construct an effective path γ that crosses a chosen point γ_0^*, which may wither be given by the macroscopic intergrator directly, or come from solving (2.2) for a short time. The values of $\frac{d}{dt}\gamma$ at various quadrature points needed by the macro-integrator are computed as below.
- From the initial condition $\gamma(t^*) =: \gamma_0^*$. A sequence of points in the state space, denoted by γ_k^*, $k = 0, \pm 1, \cdots, \pm p$, is generated by the microscopic solver solving (2.1) and (2.2) for short time intervals of length Δ. The generation of γ_k^* will be described in detail later.
- $\frac{d}{dt}\gamma(t)$ is approximated by $\frac{d}{dt}\gamma_\Delta^*(t)$ for $t \in [t^* - p\Delta,\, t^* + p\Delta]$, where $\gamma_\Delta^*(t)$ is an interpolation of γ_k^* at $t = t^* + k\Delta$.

For simplicity of presentation, we only describe the procedure for $k > 0$. Assume that the value of $\gamma(t^*)$ is given, we start by defining $\gamma_0^* := \gamma(t^*)$.

- For $k = 0, 1, \cdots, p-1$,
  1. Solve equation (2.1) for x^ϵ using γ_k^* as the initial condition at $t = k\Delta$, and obtain the solution at time $(k+1)\Delta$, denoted by $x^\epsilon(\Delta; \gamma_k^*)$.
  2. Solve equation (2.2) for y backward in time, from $t = (k+1)\Delta$ to $k\Delta$, with the condition $x(\Delta; \gamma_k^*)$. Denote the solution at $t = k\Delta$ by $y(-\Delta; \gamma_k^*)$.
- Define $\gamma_{k+1}^* := y(-\Delta; \gamma_k^*)$.

The procedure for $k < 0$ involves first solving y backward in time, and then solving x^ϵ forward in time. This type of construction involving forward-backward flow can be recognized using the diagram shown in Figure 3. In Figure 4, we show two projections of γ thus constructed for the stellar orbits problem. See Section **3.2** for the stellar orbits model.

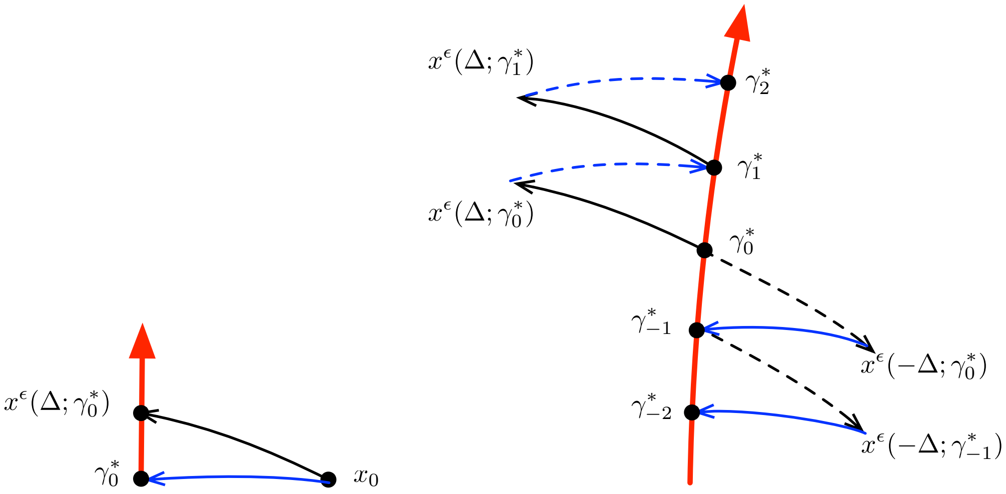

$x^\epsilon(\Delta;\gamma_0^*)$
γ_0^*
x_0
$x^\epsilon(\Delta;\gamma_1^*)$
γ_2^*
γ_1^*
$x^\epsilon(\Delta;\gamma_0^*)$
γ_0^*
γ_{-1}^*
$x^\epsilon(-\Delta;\gamma_0^*)$
γ_{-2}^*
$x^\epsilon(-\Delta;\gamma_{-1}^*)$

Figure 3: An illustration of the BF HMM construction for approximating an effective path that passes through γ_0^*. This construction will take place at every microscopic simulation in a BF HMM algorithm. Mappings that involve backward in time solutions of either (2.1) and (2.2) are depicted by the dashed arrow curves. (Left) This diagram summarizes the evaluation of F_n in Algorithm 1. Together with the chosen Forward Euler macro-solver, the structure corresponds to the HMM structure shown in the left subfigure of Figure 1. (Right) Blue curves symbolizes mappings that involve the solutions of (2.2). The red curve depicts the trajectory of the computed effective path.

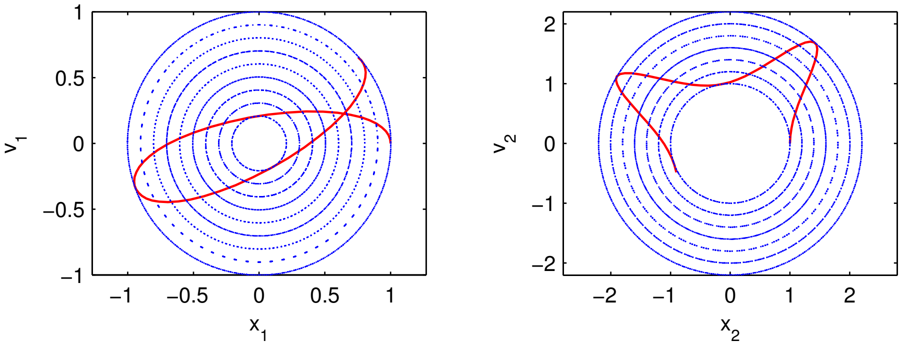


Figure 4: Projections of $\gamma(t)$ onto the x_1-v_1 and the x_2-v_2 planes, are shown by the solid curve. The level sets of the slow variables are shown by the dotted contours. γ is computed by a second order explicit Runge-Kutta method using macroscopic time step size $H = 0.25$, $\epsilon = 10^{-4}$. See Algorithm 2 for generating solid curves and Section 3.2 for the stellar orbits equation.

2.1.1 A sampling issue

In a typical application, the slow variables along $x^\epsilon(t)$ will possess $\mathcal{O}(\epsilon)$ oscillations around a smooth average; i.e. one cannot expect that $|\frac{d^\nu}{dt^\nu}\xi \circ x^\epsilon|$ is bounded uniformly in ϵ for $\nu \geq 2$. Since slow variables are functions that do not depend on ϵ, the boundedness of $|\frac{d^\nu}{dt^\nu}\xi \circ x^\epsilon| = |\frac{d^\nu}{dt^\nu}\xi \circ \gamma|$ thus determines the boundedness of $\gamma^{(\nu)}(t)$. In other words, for most applications, the effective path $\gamma(t)$ constructed by the algorithm outlined above will have fast oscillations of $\mathcal{O}(\epsilon)$ amplitudes. This poses some restriction to the lengths of Δ and the macroscopic step size, H.

Nevertheless, the $O(\epsilon)$ oscillations will be sampled very irregularly by the interpolation points γ_k^* and will typically lead to an $O(\epsilon/\Delta)$ error in the approximation of $d\bar{\xi}/dt$ regardless of how many points we use in an interpolation. This limitation of accuracy can be lifted by a novel filtering technique described in the following section, or by additional knowledge of the periodicities of the fast oscillations in $\xi \circ x^\epsilon(t)$.

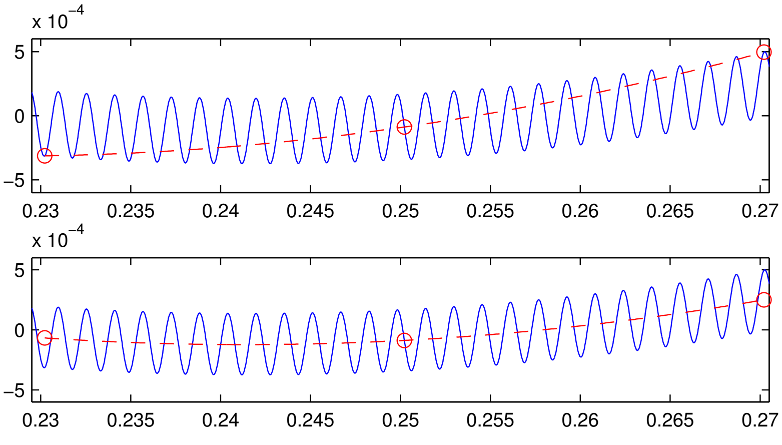


Figure 5: The blue curves are the trajectories of $\xi_1 \circ x^\epsilon(t)$ with $\epsilon = 10^{-3}$, showing fast oscillations with small amplitudes. The top plot shows the result obtained without the new filtering. The bottom plot is obtained with the new filtering.

2.1.2 A novel on-the-fly filtering approach

As we see from the discussion in Section 2.1.1, the bottleneck in accuracy of this new algorithm is a consequence of the small-amplitude fast oscillations in $\xi \circ x^\epsilon(t)$. The accuracy of the proposed algorithm can be improved if γ_k sample *implicitly* the smooth average $\bar{\xi}$ instead. Since we assume no explicit knowledge about the slow variables, $\bar{\xi}$ must be computed *intrinsically.*

Our idea is to average the vector field defined by the dynamical system "on-the-fly". More precisely, we propose to replace the (2.1) by a filtered equation

$$\frac{d}{dt}\tilde{x} = \frac{1}{\epsilon}f_1(\tilde{x}) + K_\Delta(t)f_0(\tilde{x}, t, \frac{t}{\epsilon}), \tag{2.5}$$

with the identical initial condition as x^ϵ; i.e. $\tilde{x}(t^*) = x^\epsilon(t^*)$. In the forward in time simulations for time interval $t^* \leq t \leq t^* + \Delta$, the filter $K_\Delta(t)$ will vanish outside of that interval. Similarly, in the backward in time simulations, the filter will be support in $t^* - \Delta \leq t \leq t^*$. We will develop the appropriate filters so that the smooth average $\bar{\xi}(t)$ of $\xi \circ x^\epsilon(t)$ is approximated accurately by $\xi \circ \tilde{x}(t)$ at $t = t^* \pm \Delta$.

The mechanism of this approach can be understood by comparing

$$x' = \frac{i}{\epsilon}x + c(t, \frac{t}{\epsilon})x,$$

and the corresponding filtered equation in the interval $0 \leq t \leq \Delta$. With $x = e^{\frac{i}{\epsilon}t}w$ and $\tilde{x} = e^{\frac{i}{\epsilon}t}\tilde{w}$, we have

$$w' = c(t, \frac{t}{\epsilon})w, \quad \text{and } \tilde{w}' = K_\Delta(t)c(t, \frac{t}{\epsilon})\tilde{w}.$$

Suppose $c(t, t/\epsilon) = \bar{c}(t) + \alpha(t/\epsilon)$, where α is a periodic function with zero average. Then

$$w(t) = w_0 \exp(\int_0^t \bar{c}(s)ds + \int_0^t \alpha(\frac{s}{\epsilon})ds) = y_0 \exp(\int_0^t \bar{c}(s)ds) + \mathcal{O}(\epsilon)), \tag{2.6}$$

$$\tilde{w}(t) = w_0 \exp(\int_0^t K_\Delta(s)\bar{c}(s)ds + \int_0^t K_\Delta(s)\alpha(\frac{s}{\epsilon})ds). \tag{2.7}$$

In this example, the lower order term containing α in the right hand side of (2.6) causes the sampling issue mentioned above. Thus, our main objective is to build high order scheme that computes the smooth part of w; i.e.

$$\bar{w}(t) := w_0 \exp(\int_0^t \bar{c}(s)ds).$$

In the algorithm that we outlined above, *we only need that the value of $\tilde{w}(t)$ to be close to $\bar{w}(t)$ at $t = \Delta$, the filter K_Δ should perform two specific types of approximations corresponding to the integrals involving $\bar{c}$ and α.*

The theory of averaging out oscillations that appears in the integral for $\alpha(t/\epsilon)$ is developed in [22]. It requires that K_Δ is compactly supported in the interval $[0, \Delta]$, and the effectiveness of averaging out the oscillations in α is determined by the regularity of K_Δ at $s = 0$ and Δ; i.e.

$$\frac{d^k}{dt^k}K_\Delta(0) = \frac{d^k}{dt^k}K_\Delta(\Delta) = 0, k = 0, 1, \ldots, q, \tag{2.8}$$

which imply, **using integration by parts,**

$$\left| \int_0^\Delta K_\Delta(s)\alpha(\frac{s}{\epsilon})ds \right| \leq \mathrm{C} \cdot \frac{\epsilon^q}{\Delta^{q-1}}. \tag{2.9}$$

High order accurate approximation of the integration of $\bar{c}$ requires different conditions. Taylor expansion of $\bar{c}(t)$ around $t = \Delta$ gives $\bar{c}(t) = \bar{c}(\Delta) + (t-\Delta)\bar{c}'(\Delta) + ...$ and

$$\int_0^\Delta \bar{c}(s)ds = \sum_j \frac{\bar{c}^{(j)}(\Delta)}{j!} \int_0^\Delta (s-\Delta)^j ds,$$

$$\int_0^\Delta K_\Delta(s)\bar{c}(s)ds = \sum_j \frac{\bar{c}^{(j)}(\Delta)}{j!} \int_0^\Delta K_\Delta(s)(s-\Delta)^j ds.$$

Thus for this type of problems, we may require what we called *the quadrature moment conditions for the filter* K_Δ:

$$\int_0^\Delta K_\Delta(s)s^j ds = \int_0^\Delta s^j ds, j = 0, 1, 2, \cdots, p. \tag{2.10}$$

For convenience of presentation below, let $\tilde{\mathbb{K}}^{p,q}(I)$ denote the space of normalized C^q functions, supported on I that have p moments specified by

$$\int K(t)t^r dt = \int_I t^r dt = \frac{1}{r+1}, 0 \le r \le p. \tag{2.11}$$

For $\Delta > 0$, $K_\Delta(t)$ denotes a rescaling of K as $K_\Delta(t) = K(\Delta^{-1}t)$.

We remark the estimate in (2.9) shows that it is more important to use a filter with higher regularity, as it directly impacts on how the error $|\frac{d^j}{dt^j}\xi \circ \tilde{x} - \frac{d}{dt}\bar{\xi}|$ depends on ϵ, and consequently, how the sizes of Δ and the step size H for the macro-solver should be chosen.

Figure 5 demonstrates a scenario for the stellar orbits example. In it, the blue curves correspond to the the values of the slow variable $\xi_1 \circ \mathbf{x}(t)$ defined in Section 3.2. The red circles show the values of $\xi_1(\gamma_k)$ at times $t_n + k\Delta$, and the dotted red curves are the quadratic interpolants of these values. The bottom plot in Figure 5 is obtained with the strategy to be discussed below.

Algorithm 2. *Midpoint rule BF HMM*

1. (Midpoint rule macro-solver) Compute γ_{n+1} from γ_n at $t_n = nH$.

$$\begin{aligned} \gamma_{n+\frac{1}{2}} &= \gamma_n + \frac{H}{2}\mathcal{F}_{HMM}(\gamma_n, t_n), \\ \gamma_{n+1} &= \gamma_n + H\mathcal{F}_{HMM}(\gamma_{n+\frac{1}{2}}, t_n + \frac{H}{2}) \end{aligned}$$

   where $\mathcal{F}_{HMM}$ is defined below.

2. (Micro-solver) Evaluation of $\mathcal{F}_{HMM}(\gamma_0^*, t^*)$. With a chosen filter $K_\Delta \in \tilde{\mathbb{K}}^{p,q}([0,\Delta])$, $p, q \ge 1$, and $\Delta > 0$:

   (a) (Forward solution of the perturbed equation) Solve

$$\frac{d}{dt}\tilde{x} = \epsilon^{-1}f_1(\tilde{x}) + K_\Delta(t-t^*)f_0(\tilde{x}, t, \frac{t}{\epsilon}), \quad \tilde{x}(t^*) = \gamma_0^*$$

   for $t \in [t^*, t^* + \Delta]$. Denote the solution at $t = t^* + \Delta$ by $\tilde{x}(\Delta; \gamma_0^*)$.

   (b) (Backward solution of the perturbed equation) Solve

$$\frac{d}{dt}\tilde{x} = \epsilon^{-1}f_1(\tilde{x}) + K_\Delta(t-t^*)f_0(\tilde{x}, t, \frac{t}{\epsilon}), \quad \tilde{x}(t^*) = \gamma_0^*$$

   for $t \in [t^* - \Delta, t^*]$. Denote the solution at $t = t^* - \Delta$ by $\tilde{x}(-\Delta; \gamma_0^*)$.

(c) (Forward solution of the unperturbed equation) Solve

$$\frac{d}{dt}y_F = \epsilon^{-1}f_1(y_F), \quad y_F(t^*-\Delta) = \tilde{x}(-\Delta;\gamma_0^*)$$

for $t \in [t^*-\Delta, t^*]$. Denote the solution $y_F(t^*)$ by γ_{-1}^*.

(d) (Backward solution of the unperturbed equation) Solve

$$\frac{d}{dt}y_B = \epsilon^{-1}f(y_B), \quad y_B(t^*+\Delta) = \tilde{x}(\Delta;\gamma_0^*)$$

for $t \in [t^*, t^*+\Delta]$. Denote the solution $y_F(t^*)$ by γ_1^*.

(e) Evaluate $\mathcal{F}_{HMM}$:

$$\mathcal{F}_{HMM}(\gamma_0^*, t^*) := \frac{\gamma_1^* - \gamma_{-1}^*}{2\Delta}.$$

3. Repeat.

Algorithm 3. *(Explicit s-stage Runge Kutta BF HMM)*

1. (Macro-solver: An explicit s-stage m-th order Runge-Kutta method defined by the Butcher's tableau involving the coefficients $(a_{i,j})$, b_i, and c_j, $1 \leq i,j \leq s$.)
Computes γ^{n+1} from the given value γ^n at $t = t_n$.

$$\gamma^{n+1} = \gamma^n + H\sum_{i=1}^{s} b_i k_i,$$

where

$$k_j = \mathcal{F}_{HMM}(\gamma^n + H\sum_{\ell=1}^{j-1} a_{j\ell}k_\ell, t_n + c_j H), \quad j = 1, 2, \cdots, s.$$

The values of $\mathcal{F}_{HMM}$ are computed from microscopic simulations.

2. (Micro-solver) Evaluate $\mathcal{F}_{HMM}(\gamma_0^*, t^*)$ at the given values of γ_0^* and t^*.
Let $S_{t^*}^{\Delta}$ be the operator that maps a given initial data at $t = t^*$ to the solution of the filtered perturbed equation (2.5) to $t^*+\Delta$, and let $\tilde{S}_{t^*}^{\Delta}$ be the operator that has the analogous function for the unperturbed equation (2.2). Define

$$\gamma_j^* := \left(\tilde{S}_{t^*+\Delta}^{-\Delta} S_{t^*}^{\Delta}\right)^j \gamma_0^*,$$

$$\gamma_{-k}^* := \left(\tilde{S}_{t^*-\Delta}^{\Delta} S_{t^*}^{-\Delta}\right)^k \gamma_0^*.$$

Let $\gamma_\Delta(t)$ be an interpolant of γ_j at $t^* + j\Delta$. Then

$$\mathcal{F}_{HMM}(\gamma_0^*, t^*) := \frac{d}{dt}\gamma_\Delta(t^*).$$

2.1.3 Formal accuracy estimate

Here we summarize errors produced by Algorithm 3.

- Global error in macro-simulation: Using an α-th order method with step size H, is given by H^α.

- Global error in each micro-simulation: Using an β-th order method with step size h, we solve equations of $x(t)$ and $y(t)$ over micro interval Δ. The global error is of order $\frac{\Delta h^\beta}{\epsilon^{\beta+1}}$.
- Filtering errors, by which we refer to the errors made in constructing γ_k^*. Using a filter $K_\Delta(t) \in \tilde{\mathbb{K}}^{p,q}$ with $p, q \geq 1$, we have a residual from averaging the oscillations of order $\frac{\epsilon^q}{\Delta^{q-1}}$, and quadrature error of order at most Δ.
- Error in approximation of $\gamma'(t)$ via interpolation: interpolating $n+1$ points by an n-th degree polynomial leads to an error of order Δ^n.

In our setup for the multiscale problems, we consider a regime: $0 \leq t \leq T$, $\epsilon \to 0$, $T \sim \mathcal{O}(1)$, $\Delta \sim \mathcal{O}(\epsilon)$, and $H \sim \mathcal{O}(1)$, assuming that $\bar{\xi}(t)$ has ν derivatives bounded uniformly independent of ϵ, and $\nu \geq \alpha$. In this regime, the dominating error terms would be that from micro-solver $\mathcal{O}(h/\epsilon)^\beta$, the filtering errors $\mathcal{O}(\epsilon)$, and the error from the macro-solver $\mathcal{O}(H^\alpha)$.

3 Numerical examples

In this section, we apply our BF HMM algorithm described in Section 2 to ODE systems and compare it with other methods.

3.1 Expanding spiral

Consider in the complex plane the following equation

$$\dot{x} = i\epsilon^{-1}x + \mathrm{Re}(x) \cdot x + x|x|^{-1}, \tag{3.1}$$

with the initial value $x(0) = 0.5$. As in Example 1, the dynamics of $x(t)$ can be analyzed by the corresponding system of slow and fast variables:

$$\begin{cases} \dot{\xi} = \xi^2 \cos\theta + 1, & \xi(0) = 0.5, \\ \dot{\theta} = \epsilon^{-1}, & \theta(0) = 0. \end{cases} \tag{3.2}$$

We see immediately that in constant time interval, ξ is well approximated by $\bar{\xi}$ such that

$$\dot{\bar{\xi}} = 1, \quad \bar{\xi}(0) = 0.5. \tag{3.3}$$

The small-amplitude fast oscillations in $\xi \circ x(t)$ are illustrated in Figure 6. In this example, we used Algorithm 1, the Forward Euler BF HMM to compute the solution; however, in each mico-simulation the micro-solver integrates the filtered equation

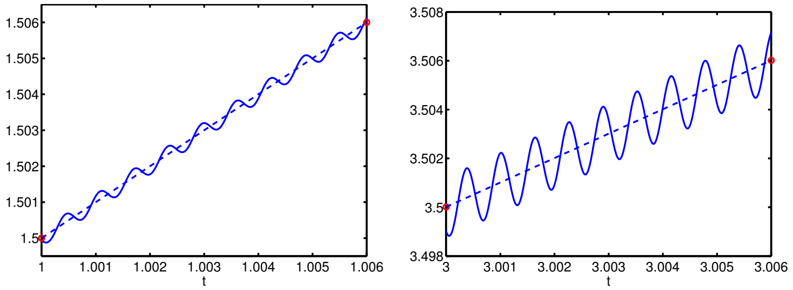
$$\dot{x}_n = i\epsilon^{-1}x_n + K_\Delta(t - t_n)\left(\mathrm{Re}(x_n) \cdot x_n + x_n|x_n|^{-1}\right), \;\; t_n \leq t \leq t_n + \Delta,$$

with the parameters in Table 1 and a filter $K \in \tilde{\mathbb{K}}^{1,5}$.

In Table 2, we show the accuracy in the approximations of the averaged slow variable $\bar{\xi}$ obtained with filters of different regularity.

Table 1: (Section 3.1) BF HMM parameters

ϵ	T	Δ	H	Micro solver	RelTol(ODE45 parameter)	Macro solver
10^{-4}	10	60ϵ	1	ODE45	1e-8	Forward Euler



(a) Plot of $\xi \circ x(t)$ over $t \in [1, 1+\Delta]$. (b) Plot of $\xi \circ x(t)$ over $t \in [3, 3+\Delta]$.

Figure 6: (Section 3.1) The dynamics of the slow variable ξ. Circles are results of the Forward Euler BF HMM in the respective time intervals. The dotted lines are the linear interpolants of these values. An amplitude of the fast oscillations around $\bar{\xi}$ increases as time goes by.

Table 2: Table of $||\bar{\xi}(\cdot) - \xi \circ \gamma(\cdot)||_{L^\infty([0,10])}$ with various kernels. An estimation error decreases with higher regularity conditions.

	$q = 1$	2	3	4	5
$p = 1$	1.1e-2	9.7e-4	7.6e-4	2.2e-4	1.3e-4

3.2 A simplified model for stellar orbits in a galaxy

The following extensively analyzed system is taken from the theory of stellar orbits in a galaxy (see [32, 31]):

$$\begin{cases} r_1'' + a^2 r_1 &= \epsilon r_2^2, \\ r_2'' + b^2 r_2 &= 2\epsilon r_1 r_2, \quad 0 < \tilde{t} < T/\epsilon. \end{cases}$$

Here $r_1(0,\epsilon) = r_2(0,\epsilon) = 1, r_1'(0,\epsilon) = r_2'(0,\epsilon) = 0$; r_1 stands for the radial displacement of the orbit of a star from a reference circular orbit, and r_2 stands for the deviation of the orbit from the galactic plane. The time variable $\tilde{t}$ actually denotes the angle of the planets in a reference coordinate system. After a rescaling of time, the system can be written in the following form

$$\mathbf{x}' = \epsilon^{-1} \begin{bmatrix} 0 & a & 0 & 0 \\ -a & 0 & 0 & 0 \\ 0 & 0 & 0 & b \\ 0 & 0 & -b & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ x_2^2/a \\ 0 \\ 2x_1x_2/b \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \tag{3.4}$$

where $\mathbf{x} = [x_1, v_1, x_2, v_2]^{\mathrm{t}}$. One seeks approximation of the effective properties that take place in a constant time interval. When $a = 2$ and $b = 1$, resonance of oscillatory modes take effect in the lower order term. Using the numerical algorithm proposed in [3], three functionally independent slow variables are identified to be

$$\xi_1 = x_1^2 + v_1^2, \qquad \xi_2 = x_2^2 + v_2^2, \qquad \xi_3 = x_1x_2^2 + 2v_1x_2v_2 - x_1v_2^2. \tag{3.5}$$

where $\xi_i : \mathbb{R}^4 \to \mathbb{R}$, $i = 1, 2, 3$.

In Figure 3.2, we present a result computed by our method and compare it with the results computed by FLAVORS [41] with two different sets of parameters. Figure 7a shows the BF HMM Mid-ODE45 (Midpoint rule macro-solver and ODE45 micro-solver with quadratic polynomial interpolation for γ) result computed with the parameters tabulated in Table 3 and a kernel $K \in \tilde{\mathbb{K}}^{1,5}$. The resulting error in the slow variables is $\max\limits_{i=1,2,3}||\xi_i(\cdot) - \xi_i \circ \gamma(\cdot)||_{L^\infty([0,14])}= 0.063$. The computational time on a one-year old desktop is about 5.9s.

In Figure 7b, we show the result computed by FLAVORS with the parameters within the recommended regimes. To be more precise, as stated in [41], the required conditions for FLAVORS are as follows:

$$\Delta \ll \epsilon \ll H \ll 1 \text{ and } \left(\frac{\Delta}{\epsilon}\right)^2 \ll H \ll \frac{\Delta}{\epsilon}. \tag{3.6}$$

In [41], the proposed empirical choice is given by $\Delta = \gamma\epsilon$, $H = \gamma\frac{\Delta}{\epsilon}$ where γ is small (0.1, for instance). Figure 7b is from their empirical choice $\Delta = \gamma\epsilon$, $H = \gamma\frac{\Delta}{\epsilon}$ where $\gamma = 0.1$. We obtained $\max\limits_{i=1,2,3}||\xi_i(\cdot) - \xi_i \circ \tilde{\mathbf{x}}(\cdot)||_{L^\infty([0,14])}$=0.56. The computational time on the same machine is about 3.1s.

In Figure 7c, we show the result computed by FLAVOR by a set of parameters which do not fall in the recommendation. With the parameters shown in the Figure, we obtained $\max\limits_{i=1,2,3}||\xi_i(\cdot) - \xi_i \circ \tilde{\mathbf{x}}(\cdot)||_{L^\infty([0,14])}$=0.23. The computational time is about 8.4s.

Table 3: (Section 3.2) BF HMM parameters for Section 3.2

ϵ	T	h	Δ	H	Micro solver	RelTol	Macro solver
10^{-4}	14	$\epsilon/30$	7ϵ	0.25	ODE45	1e-5	Midpoint rule

3.3 The Fermi-Pasta-Ulam problem

The Fermi-Pasta-Ulam problem is a dynamical system which revealed highly unexpected behavior. We consider a chain of $2k$ springs connected with alternating soft nonlinear and stiff linear springs, and both ends are fixed. This problem has been a test bed for evaluating the long-time performance of geometric integrators [28]. The model is derived from the following Hamiltonian:

$$H(p,q) = \frac{1}{2}\sum_{i=1}^{2k} p_i^2 + \frac{1}{4}\epsilon^{-2}\sum_{i=1}^{k}(q_{2i} - q_{2i-1})^2 + \sum_{i=1}^{k}(q_{2i+1} - q_{2i})^4. \tag{3.7}$$

Using the change of variables given in [3], we have the following equations of motion

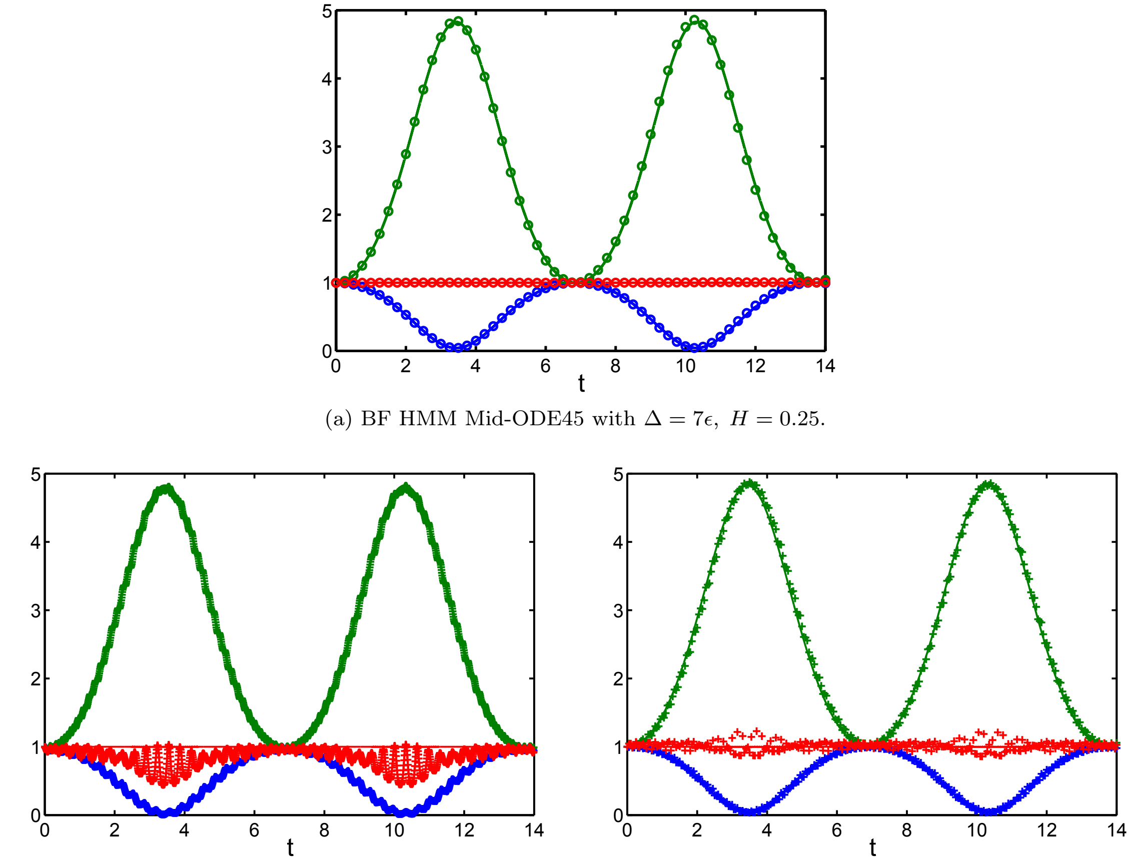
$$\begin{cases} \dot{y}_i = u_i, \\ \dot{x}_i = \epsilon^{-1} v_i, \\ \dot{u}_i = -(y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + (y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3, \\ \dot{v}_i = -\epsilon^{-1} x_i + (y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + (y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3. \end{cases} \tag{3.8}$$

Both fixed ends yield $y_0 = x_0 = y_{k+1} = x_{k+1} = 0$ and we choose $k = 3$ for an illustration. Initial conditions are $y_1 = x_1 = u_1 = 1$ and zero otherwise. Total energies of the stiff springs are given by

$$I_i = x_i^2 + v_i^2, \quad i = 1, 2, 3 \tag{3.9}$$

where $I_i : \mathbb{R}^{12} \to \mathbb{R}$. See [10, 11, 27] and references therein for some recent progress. With ϵ denoting the time scale of the fast oscillations, the nontrivial energy transfer take place in the very long ϵ^{-1} time scale. Even if one could afford the long computational time, it is unclear if the computational results retain enough effective accuracy. The FPU is a good model problem to study the proposed new algorithm for computation in $\mathcal{O}(\epsilon^{-1})$ timescale.

Figure 8 shows the energy exchange of the stiff springs over $T = \epsilon^{-1}$, with $\epsilon = 2 \cdot 10^{-3}$. We compare the results computed by the BF HMM Verlet-ODE45 (Verlet macro-solver and ODE45 micro-solver with quadratic polynomial interpolation for γ) with those by an exponential integrator



(a) BF HMM Mid-ODE45 with $\Delta = 7\epsilon,\ H = 0.25$.

(b) FLAVORS Mid-ODE45 with $\Delta = \gamma\epsilon$, $H = \gamma\frac{\Delta}{\epsilon}$, $\gamma = 0.1$.

(c) FLAVORS Mid-ODE45 with $\Delta = 20\epsilon$, $H = 0.05$.

Figure 7: (Section 3.2) The dynamics of the slow variables ξ_1, ξ_2 and ξ_3 in (3.4). Subfigures (b) and (c) FLAVORS fail to preserve the geometrical structure of the flow.

by Deuflhard [28] with the stepsize $h = 5\epsilon 10^{-7}$, which we used as a reference solution. We point out here that in order to obtain a reliable reference numerical solution, the aforementioned step size is needed. Furthermore, we had to use 128-bit precision for the variables in our computation in order to retain reasonable significant digits at time T in our computation with the exponential integrator. The BF HMM result is computed with the parameters given in Table 4, and with the filter $K^{cos} \in \tilde{\mathbb{K}}^{1,1}$ for the filtered equation that corresponds to (3.8). In this setup, the BF HMM runs approximately 30,000 times faster. The difference in the stiff springs' total energy between the HMM solution and the reference solution measured in the supremum norm is $\max_{i=1,2,3} ||I_i(\cdot) - I_i \circ \gamma(\cdot)||_{L^\infty([0,\epsilon^{-1}])} = 0.027$.

In Figure 9, with $\epsilon = 5 \cdot 10^{-3}$, we show a result computed by the same BF HMM algorithm for longer time and demonstrate the stability of our algorithm in a longer time interval. See Table 5 for simulation parameters.

Table 4: (Section 3.3) BF HMM parameters for Figure 8.

ϵ	T	h	Δ	H	Micro solver	RelTol	Macro solver
$2 \cdot 10^{-3}$	ϵ^{-1}	$\epsilon/10$	$6\pi\epsilon$	1/3	ODE45	1e-7	Verlet

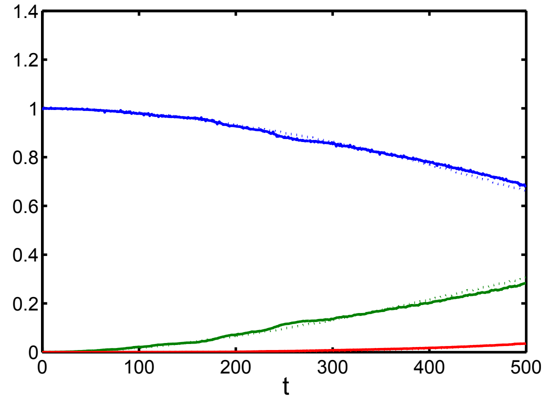


Figure 8: (Section 3.3) The solid lines correspond to DNS solution with an exponential integrator. Dotted lines correspond to the HMM.

Table 5: (Section 3.3) BF HMM parameters for Figure 9.

ϵ	T	h	Δ	H	Micro solver	RelTol	Macro solver
$5 \cdot 10^{-3}$	$7 \cdot \epsilon^{-1}$	$\epsilon/10$	15ϵ	0.3	ODE45	1e-7	Verlet

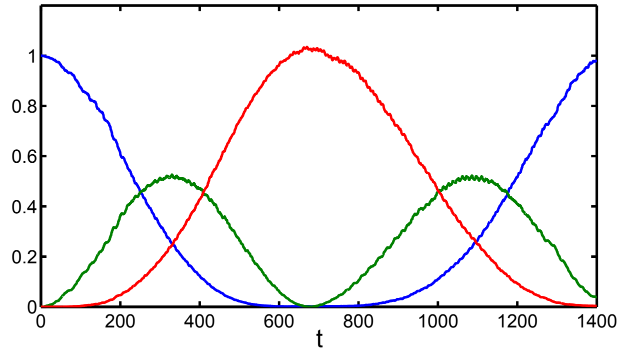


Figure 9: (Section 3.3) Long time simulation by BF HMM Verlet-ODE45 to $T = 7\epsilon^{-1}$.

4 Summary

We introduce and analyze a new class of multiscale methods that use a technique related to the construction of a Poincaré map. The proposed algorithms compute the effective slow behavior of highly oscillatory dynamical systems built on the HMM framework. The main idea of this paper is the following:

- The essential invariant manifolds are defined by the slow variables, and from the given initial data, the highly oscillatory dynamical system has an effective evolution across the different essential invariant manifolds.

- We construct an effective path by considering the solutions of the equations with and without lower order perturbation. This path discloses information about an effective evolution of the invariant manifolds in state space.

- A novel on-the-fly filtering technique is applied for achieving high order accuracy beyond other approaches that rely only on dynamical system's self-averaging property.

Finally, we point out that the proposed HMM methods are not limited to the few simple schemes that we presented here. One can use this methodology to build an HMM scheme for problems with more than two separated scales, and to capture numerically the slow behavior of stiff stochastic differential equations. Rigorous analysis of the proposed methods and their generalization will be reported in a forthcoming paper by the authors.

5 Acknowledgments

Kim and Tsai are partially supported by NSF grants DMS-0914840 and DMS-0914465. Engquist was partially supported by NSF grant DMS-0714612.

References

[1] G. Ariel, B. Engquist, H.-O. Kreiss, and R. Tsai. Multiscale computations for highly oscillatory problems. In *Multiscale modeling and simulation in science*, volume 66 of *Lect. Notes Comput. Sci. Eng.*, pages 237–287. Springer, Berlin, 2009.

[2] G. Ariel, B. Engquist, and R. Tsai. Numerical multiscale methods for coupled oscillators. *Multiscale Model. Simul.*, 7(3):1387–1404, 2008.

[3] G. Ariel, B. Engquist, and R. Tsai. A multiscale method for highly oscillatory ordinary differential equations with resonance. *Math. Comp.*, 78:929–956, 2009.

[4] G. Ariel, B. Engquist, and R. Tsai. Numerical multiscale methods for coupled oscillators. *Multi. Mod. Simul.*, 7:1387–1404, 2009.

[5] G. Ariel, B. Engquist, and R. Tsai. A reversible multiscale integration method. *Comm. Math. Sci.*, 7(3):595–610, 2009.

[6] G. Ariel, B. Engquist, and R. Tsai. Oscillatory systems with three separated time scales – analysis and computation. In B. Engquist, O. Runborg, and Y.H. R. Tsai, editors, *Numerical analysis of multiscale computations*, volume 82 of *Lecture Notes in Computational Science and Engineering.* Springer-Verlag, 2011.

[7] G. Ariel and E. Vanden-Eijnden. Accelerated simulation of a heavy particle in a gas of elastic spheres. *Multiscale Model. Simul.*, 7(1):349–361, 2008.

[8] Z. Artstein, I. G. Kevrekidis, M. Slemrod, and E. S. Titi. Slow observables of singularly perturbed differential equations. *Nonlinearity*, 20(11):2463–2481, 2007.

[9] Z Artstein, J. Linshiz, and E. S. Titi. Young measure approach to computing slowly advancing fast oscillations. *Multiscale Model. Simul.*, 6(4):1085–1097, 2007.

[10] D. Bambusi and A. Ponno. On metastability in FPU. *Comm. Math. Phys.*, 264(2):539–561, 2006.

[11] D. Bambusi and A. Ponno. Resonance, metastability and blow up in FPU. In *The Fermi-Pasta-Ulam problem*, volume 728 of *Lecture Notes in Phys.*, pages 191–205. Springer, Berlin, 2008.

[12] S. D. Bond and B. J. Leimkuhler. Molecular dynamics and the accuracy of numerically computed averages. *Acta Numer.*, 16:1–65, 2007.

[13] C. Chu, B. Engquist, M. Prodanovic, and R. Tsai. A multiscale method coupling network and continuum models in porous media I – single phase flow. *Under review*, 2011.

[14] C. Chu, B. Engquist, M. Prodanovic, and R. Tsai. A multiscale method coupling network and continuum models in porous media II – single and two phase phase flow. *Under review*, 2011.

[15] D. Cohen, T. Jahnke, K. Lorenz, and C. Lubich. Numerical integrators for highly oscillatory hamiltonian systems: A review. In *Analysis, Modeling and Simulation of Multiscale Problems*, pages 553–576. Springer Berlin Heidelberg, 2006.

[16] M. Condon, A. Deaño, and A. Iserles. On second-order differential equations with highly oscillatory forcing terms. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 466(2118):1809–1828, 2010.

[17] W. E and B. Engquist. The heterogeneous multiscale methods. *Commun. Math. Sci.*, 1(1):87–132, 2003.

[18] W. E and E. Vanden-Eijnden. Numerical techniques for multiscale dynamical systems with stochastic effects. *Comm. Math. Sci.*, 1(2):385–391, 2003.

[19] Weinan E. Analysis of the heterogeneous multiscale method for ordinary differential equations. *Commun. Math. Sci.*, 1(3):423–436, 2003.

[20] B. Engquist, H. Holst, and O. Runborg. Multiscale methods for wave propagation in heterogeneous media over long time. In B. Engquist, O. Runborg, and R. Tsai, editors, *Numerical analysis of multiscale computations*, volume 82 of *Lect. Notes Comput. Sci. Eng.* Springer-Verlag, 2011.

[21] B. Engquist and Y.-H. Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Math. Comp.*, 74(252):1707–1742, 2005.

[22] B. Engquist and Y.-H. Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Math. Comp.*, 74(252):1707–1742 (electronic), 2005.

[23] I. Fatkullin and E. Vanden-Eijnden. A computational strategy for multiscale chaotic systems with applications to Lorenz 96 model. *J. Comp. Phys.*, 200:605–638, 2004.

[24] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel. Long-time-step methods for oscillatory differential equations. *SIAM J. Sci. Comput.*, 20(3):930–963, 1999.

[25] C. W. Gear and I. G. Kevrekidis. Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *SIAM J. Sci. Comput.*, 24(4):1091–1106 (electronic), 2003.

[26] C. W. Gear and I. G. Kevrekidis. Constraint-defined manifolds: a legacy code approach to low-dimensional computation. *J. Sci. Comput.*, 25(1-2):17–28, 2005.

[27] E. Hairer and C. Lubich. On the energy distribution in Fermi-Pasta-Ulam lattices. *Preprint*, 2010.

[28] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2002.

[29] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010.

[30] A. Iserles, H. Munthe-Kaas, S.P. Nørsett, and A. Zanna. Lie-group methods. In *Acta numerica, 2000*, volume 9 of *Acta Numer.*, pages 215–365. Cambridge Univ. Press, Cambridge, 2000.

[31] J. Kevorkian and J. D. Cole. *Perturbation methods in applied mathematics*, volume 34 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1981.

[32] J. Kevorkian and J. D. Cole. *Multiple scale and singular perturbation methods*, volume 114 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1996.

[33] H.-O. Kreiss. Problems with different time scales for ordinary differential equations. *SIAM J. Numer. Anal.*, 16(6):980–998, 1979.

[34] H.-O. Kreiss. Problems with different time scales. In *Acta numerica, 1992*, pages 101–139. Cambridge Univ. Press, 1992.

[35] H.-O. Kreiss and J. Lorenz. Manifolds of slow solutions for highly oscillatory problems. *Indiana Univ. Math. J.*, 42(4):1169–1191, 1993.

[36] B. Leimkuhler and S. Reich. *Simulating Hamiltonian dynamics*, volume 14 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2004.

[37] J.E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, pages 357–514, 2001.

[38] L. R. Petzold, L. O. Jay, and J. Yen. Numerical solution of highly oscillatory ordinary differential equations. In *Acta numerica, 1997*, volume 6 of *Acta Numer.*, pages 437–483. Cambridge Univ. Press, Cambridge, 1997.

[39] J. A. Sanders and F. Verhulst. *Averaging Methods in Nonlinear Dynamical Systems*, volume 59 of *Applied Mathematical Sciences*. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1985.

[40] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian problems*, volume 7 of *Applied Mathematics and Mathematical Computation*. Chapman & Hall, London, 1994.

[41] M. Tao, H. Owhadi, and J. Marsden. Nonintrusive and structure preserving multiscale integration of stiff ODEs, SDEs, and Hamiltonian systems with hidden slow dynamics via flow averaging. *Multiscale Model. Simul.*, 8(4):1269–1324, 2010.

[42] E. Vanden-Eijnden. Numerical techniques for multi-scale dynamical systems with stochastic effects. *Comm. Math. Sci.*, 1:385–391, 2003.

[43] X. Li W. Ren W. E, B. Engquist and E. Vanden-Eijnden. Heterogeneous multiscale methods: A review. *Commun. Comput. Phys.*, 2(3):367–450, 2007.