

# A Multiscale Method for Highly Oscillatory Dynamical Systems Using a Poincaré Map Type Technique

G. Ariel · B. Engquist · S. Kim · Y. Lee · R. Tsai

Received: 14 February 2012 / Revised: 27 August 2012 / Accepted: 11 October 2012 /  
Published online: 23 October 2012  
© Springer Science+Business Media New York 2012

**Abstract** We propose a new heterogeneous multiscale method (HMM) for computing 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 ideas: a nonstandard splitting of the vector field (the right hand side of the ODEs); comparison of the solutions of the split equations; construction 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.

**Keywords** Oscillatory dynamical system · Averaging

## 1 Introduction

In many applications 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. There

---

G. Ariel  
Bar-Ilan University, Ramat Gan, 52900, Israel  
e-mail: [arielg@math.biu.ac.il](mailto:arielg@math.biu.ac.il)

B. Engquist · S. Kim · Y. Lee · R. Tsai (✉)  
Department of Mathematics and Institute for Computational Engineering and Sciences (ICES),  
The University of Texas at Austin, Austin, TX 78712, USA  
e-mail: [ytsai@math.utexas.edu](mailto:ytsai@math.utexas.edu)

B. Engquist  
e-mail: [engquist@ices.utexas.edu](mailto:engquist@ices.utexas.edu)

S. Kim  
e-mail: [skim@math.utexas.edu](mailto:skim@math.utexas.edu)

Y. Lee  
e-mail: [ylee@math.utexas.edu](mailto:ylee@math.utexas.edu)

has been significant amount of activities 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, 29–31, 37, 38, 41], 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 beyond the scope of 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 time interval of interest. Gear and Kevrekidis introduced such a technique in [26] and applied it successfully to dissipative problems. We will concentrate on highly oscillatory problems and build on the framework of E and Engquist [19], The Heterogeneous Multiscale Method (HMM).

An HMM utilizes an important consequence of scale separation in the given problems. If enough information about the fast scale influence on the slow scale dynamics can be obtained by performing localized simulations over short times, then it is possible to obtain a numerical complexity that is much smaller than direct simulations of the given systems. 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 [39] 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 differential equations [20]. In a sequence of papers, [2–6, 22], 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 include, e.g. [8, 9].

The requirement of using an 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 dynamics 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 Fig. 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. Iterating this procedure a few times, we obtain a sequence of points in the state space. From these points, we then construct locally an effective path along which the evolutions of the original system’s slow variables are accurately captured. Moreover, our algorithm achieves 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 [25] and FLAVORS [42] 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 split 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 split 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 Sect. 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 Sect. 2.1.2. Section 3 presents several numerical examples. Here we compare our method with existing methods, e.g., FLAVORS. We conclude in Sect. 4.

### 1.1 The HMM Framework

We consider the computation 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).

Significant amplification 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  $\epsilon$  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  $\epsilon^{-1}$ . 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  $U$  is a set of functions of  $u$ , they are commonly referred to as slow variables of the system. See [2, 4, 24, 27, 34–36]. 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  $|da/dt| \leq C$  for some constant  $C$  independent of  $\epsilon$  in  $t \in I$ . Moreover, a smooth function  $\xi(u) : \mathcal{D}_0 \rightarrow \mathbb{R}$  is called a *slow variable* with respect to  $u(t)$  if  $\xi(t) = \xi(u(t))$  is said to be 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 including some systems from molecular dynamics. In the HMM framework [3, 4, 22], one assumes a macroscopic model

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

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

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

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:

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

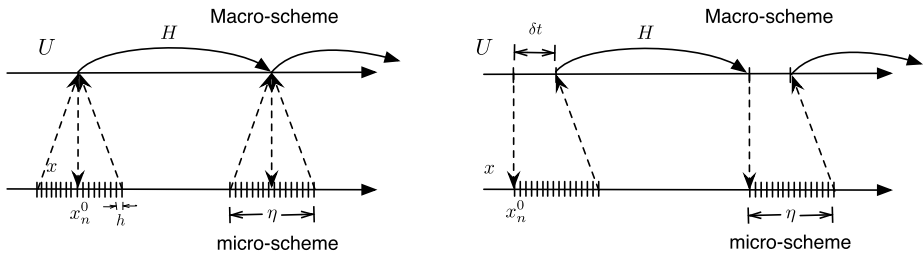
where  $D_R$  are the needed data that can be evaluated from  $u$ . Errors of this type of schemes generally take the structure

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

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 multiscale model, including the passing of information through  $\mathcal{R}$  and  $\mathcal{Q}$ . This approach has been used in a number of applications, such as contact line problems, epitaxial growth, thermal expansions, combustion [21], as well as homogenizations of wave propagation in long time intervals [23], 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 [18, 26, 39], and [7] are also of a similar structure. As a simple example, the forward Euler scheme applied to (1.2) would appear to be

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



**Fig. 1** Two typical structures of the proposed multiscale algorithm. The structure on the left is suitable for dissipative systems

where  $F$  contains the passage of  $\mathcal{Q}\Phi_t\mathcal{R}(U_n)$ ; reconstruction  $\mathcal{R}$ , evolution  $\Phi_t$ , and compression  $\mathcal{Q}$ , and  $H$  is the step size. If each evolution  $\Phi_t$  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 resolved 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  $\epsilon$ , all well known methods, assuming  $U = u$ , will converge as the step-size  $H \rightarrow 0$ , and there is no difference between stiff and non-stiff problems. In the related work [3, 4, 22], convergence for stiff problems ( $\epsilon \ll H$ ) is defined by the following relation: For the error

$$E(H) = \max_{0 \leq t_n \leq T} \left( \sup_{0 < \epsilon < \epsilon_0(H)} |U(t_n) - U_n| \right), \tag{1.7}$$

we have that  $E(H) \rightarrow 0$  as  $H \rightarrow 0$ . Here,  $\epsilon_0(H)$  is a positive function of  $H$ , serving as an upper bound for the range of  $\epsilon$ , 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  $\epsilon^{-1}$ .

### 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  $z$ ,  $x(t)$  is ergodic with respect to an invariant measure

$\mu$  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 for building multiscale scale methods; see [18, 43]. For these systems, 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, due to the ergodicity of the system,  $x^*$  can 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 [35], and more sophistication in the algorithm is needed. One can see the essence of this problem from the simple example,

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

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  $\alpha$  has only fast oscillations and averages to zero. 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  $\alpha$ . 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 [3, 22, 24, 42]. 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  $\epsilon$ . 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 [3, 4] for designing multiscale algorithms that compute the effective behavior of highly oscillatory dynamical systems by using slow variables. [4] shows how polynomial slow variables in (1.1) can be identified by a numerical algorithm and how short time solutions of (1.1) can be used to approximate the assumed effective equation. In particular, for autonomous systems, a diffeomorphism  $\Psi : u \rightarrow (\xi(u), \phi(u))$  of (1.1) from  $\mathbb{R}^d$  onto  $\mathbb{R}^{d-1} \times S^1$  is constructed so that

$$\begin{cases} \dot{\xi} = g_0(\xi, \phi), & \xi(0) = \xi_0, \\ \dot{\phi} = \epsilon^{-1} g_1(\xi, \phi) + g_2(\xi, \phi), & \phi(0) = \phi_0, \end{cases} \tag{1.10}$$

and all smooth slow variables depend on  $\xi$  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  $\xi$  along the dynamical system’s solutions,  $\xi \circ u$ , converge as  $\epsilon \rightarrow 0$ . We shall denote this limit as  $\bar{\xi}(t; u_0)$ . This expectation may come directly from the averaging theory [40].

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

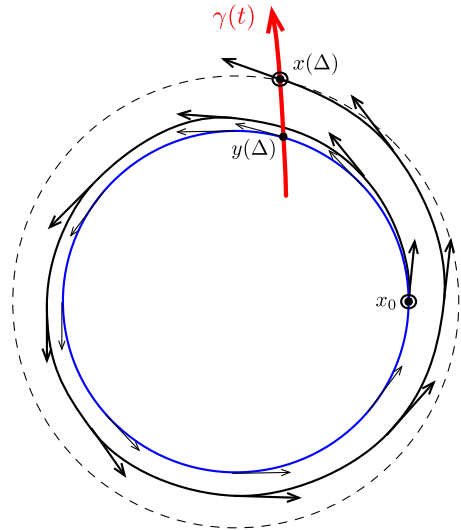
**Assumption 1** The trajectory passing through  $y_0 \in \mathcal{D}_0$  of the unperturbed equation

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

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

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

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



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

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

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  $\mathcal{M}(t)$  at  $s = t$ . Note that such  $\gamma$  is not unique and we shall construct one in the state space such that for any slow variable  $\xi$ , and finite time interval,  $\gamma$  satisfies the following conditions:

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

We shall refer  $\gamma$  as an effective path of the given dynamical system.

It has been observed in [4] 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  $\epsilon \rightarrow 0$ . *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  $\gamma$ . In the following algorithms,  $\gamma$  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 Fig. 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 [1] for a different problem.



**Algorithm 1** (Forward Euler BF HMM)

1. (Forward Euler macro-solver) Compute  $\gamma_{n+1}$  from  $\gamma_n$  at  $t_n = nH$ .

$$\gamma_{n+1} = \gamma_n(\Delta) + HF_n,$$

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

$$\frac{d}{dt}x^\epsilon = \epsilon^{-1} f_1(x_n) + f_0(x_n, t_n; \epsilon), \quad x_n(0) = \gamma_n,$$

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.

*Example 1* Our simple example to demonstrate the consistency of the Forward Euler BF HMM is an expanding spiral [3] 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  $\epsilon$ . We transform  $x^\epsilon$  into  $(\xi, \phi)$  where  $\xi = |x^\epsilon|$  and  $\phi = \arg(x^\epsilon)$ , and obtain

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

By Definition 1,  $\xi$  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  $\xi$  is given by

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

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  $\xi$ .

### 2.1 Higher Order Schemes

In this section we describe the construction of high order accurate BF HMMs with sublinear complexity in constant time intervals. In Algorithm 1, forward Euler scheme is used to

compute the effective path  $\gamma$  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  $\gamma$  that crosses a chosen point  $\gamma_0^*$ , which may either be given by the macroscopic integrator 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  $\gamma_k^*, k = 0, \pm 1, \dots, \pm p$ , is generated by the microscopic solver solving (2.1) and (2.2) for short time intervals of length  $\Delta$ . The generation of  $\gamma_k^*$  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  $\gamma_k^*$  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  $k = 0, 1, \dots, p - 1$ ,
  1. Solve Eq. (2.1) for  $x^\epsilon$  using  $\gamma_k^*$  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 Eq. (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  $x^\epsilon$  forward in time. This type of construction involving forward-backward flow can be recognized using the diagram shown in Fig. 3. In Fig. 4, we show two projections of  $\gamma$  thus constructed for the stellar orbits problem. See Sect. 3.2 for the stellar orbits model.

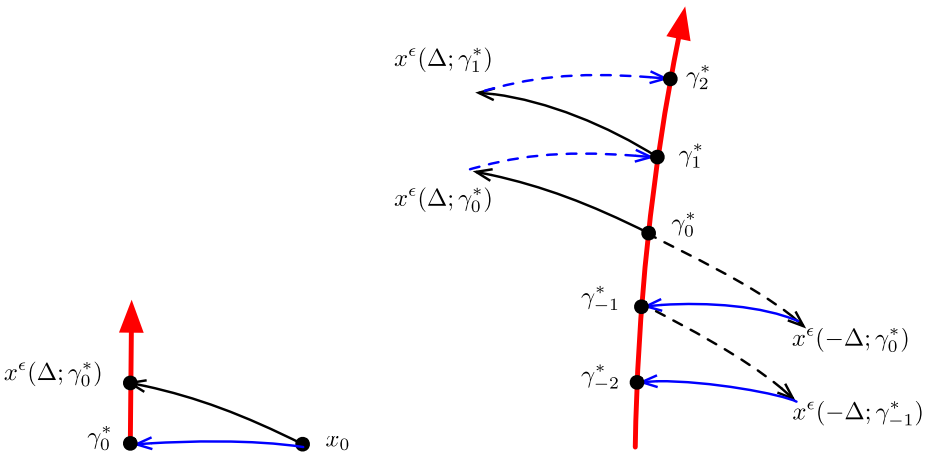
### 2.1.1 A Sampling Issue

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  $|\frac{d^v}{dt^v}\xi \circ x^\epsilon|$  be bounded uniformly in  $\epsilon$  for  $v \geq 2$ . Since slow variables are functions that do not depend on  $\epsilon$ , the boundedness of  $|\frac{d^v}{dt^v}\xi \circ x^\epsilon| = |\frac{d^v}{dt^v}\xi \circ \gamma|$  thus determines the boundedness of  $\gamma^{(v)}(t)$ . 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  $\Delta$  and the macroscopic step size,  $H$ .

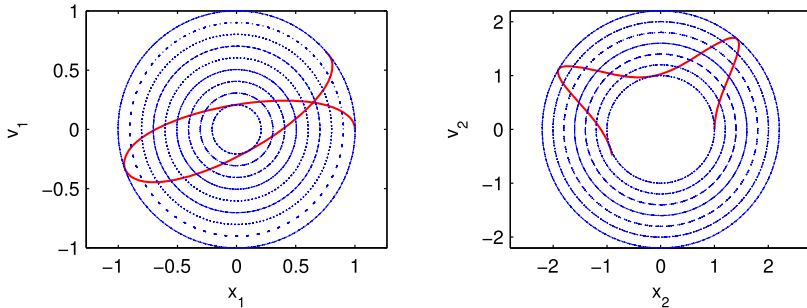
Nevertheless, the  $\mathcal{O}(\epsilon)$  oscillations will be sampled very irregularly by the interpolation points  $\gamma_k^*$  and will typically lead to an  $\mathcal{O}(\epsilon/\Delta)$  error in the approximation of  $d\xi/dt$  regardless of how many points we interpolate. This limitation of the 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  $\xi \circ x^\epsilon(t)$ .

### 2.1.2 A Novel On-the-Fly Filtering Approach

As we see from the discussion in Sect. 2.1.1, the bottleneck in the accuracy of this new algorithm is a consequence of the small-amplitude fast oscillations in  $\xi \circ x^\epsilon(t)$ . The accuracy



**Fig. 3** An illustration of the BF HMM construction for approximating an effective path that passes through  $\gamma_0^*$ . 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 Fig. 1. (Right) Blue curves symbolize mappings that involve the solutions of (2.2). The red curve depicts the trajectory of the computed effective path (Color figure online)



**Fig. 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*.  $\gamma$  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 Sect. 3.2 for the stellar orbits equation

of the proposed algorithm can be improved if  $\gamma_k$  sample 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 (2.1) by a filtered equation

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

with the identical initial condition as  $x^\epsilon$ ; 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 supported on the interval

$t^* - \Delta \leq t \leq t^*$ . 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 for linear equations can be understood by comparing

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

and the corresponding filtered equation in the interval  $0 \leq t \leq \Delta$ . Analysis of this new approach for more general nonlinear systems will be reported in a forthcoming paper. With  $x = e^{\frac{i}{\epsilon}t}w$  and  $\tilde{x} = e^{\frac{i}{\epsilon}t}\tilde{w}$ , we have

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

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

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

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

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

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

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  $K_\Delta$  performs two specific types of approximations corresponding to the integrals involving  $\bar{c}$  and  $\alpha$ .*

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

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

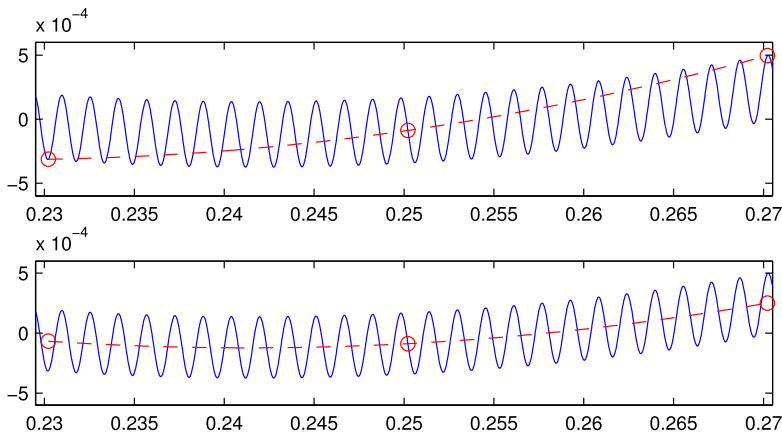
and then integration by parts yields

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

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

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

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



**Fig. 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

Thus for this type of problems, we may require what we called *the quadrature moment conditions for the filter  $K_\Delta$* :

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

We thus have the error

$$\left| \int_0^\Delta K_\Delta(\Delta - s)\bar{c}(s)ds - \int_0^\Delta \bar{c}(s)ds \right| \leq C\Delta^{p+2}.$$

For convenience of presentation below, let  $\tilde{\mathbb{K}}^{p,q}(I)$  denote the space of normalized  $C^q$  functions, supported on  $I = [0, 1]$  that have  $p$  moments specified by

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

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

We remark that with  $\Delta = \mathcal{O}(\epsilon)$ , 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  $|\frac{d^j}{dt^j}\xi \circ \bar{x} - \frac{d}{dt}\bar{\xi}|$  depends on  $\epsilon$ , and consequently, how the sizes of  $\Delta$  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 values of the slow variable  $\xi_1 \circ \mathbf{x}(t)$  defined in Sect. 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 Fig. 5 is obtained with the strategy to be discussed below.

**Algorithm 2** (Midpoint rule BF HMM)

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

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

$$\gamma_{n+1} = \gamma_n + H \mathcal{F}_{HMM}\left(\gamma_{n+\frac{1}{2}}, t_n + \frac{H}{2}\right)$$

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

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

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

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

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

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

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  $\gamma_{-1}^*$ .

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

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

for  $t \in [t^*, t^* + \Delta]$ . Denote the solution  $y_B(t^*)$  by  $\gamma_1^*$ .

(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  $\gamma^{n+1}$  from the given value  $\gamma^n$  at  $t = t_n$ .

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

where

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

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  $\gamma_0^*$  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

$$\begin{aligned} \gamma_j^* &:= (\tilde{S}_{t^*+\Delta}^{-\Delta} S_{t^*}^\Delta)^j \gamma_0^*, \\ \gamma_{-k}^* &:= (\tilde{S}_{t^*-\Delta}^\Delta S_{t^*}^{-\Delta})^k \gamma_0^*. \end{aligned}$$

Let  $\gamma_\Delta(t)$  be an interpolant of  $\gamma_j$  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  $\alpha$ -th order method with step size  $H$ , is given by  $H^\alpha$ .
- Global error in each micro-simulation: Using a  $\beta$ -th order method with step size  $h$ , we solve equations for  $x(t)$  and  $y(t)$  over micro interval  $\Delta$ . 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  $\gamma_k^*$ . 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  $\frac{\epsilon^{q+1}}{\Delta^q}$ , and quadrature error of order at most  $\Delta$ .
- Error in approximation of  $\gamma'(t)$  via interpolation: interpolating  $n + 1$  points by an  $n$ -th degree polynomial leads to an error of order  $\Delta^n$ .

In our setup for the multiscale problems, we consider a regime:  $0 \leq t \leq T$ ,  $\epsilon \rightarrow 0$ ,  $T \sim \mathcal{O}(1)$ ,  $\Delta \sim \mathcal{O}(\epsilon)$ , and  $H \sim \mathcal{O}(1)$ , assuming that  $\tilde{\xi}(t)$  has  $\nu$  derivatives bounded uniformly independent of  $\epsilon$ , and  $\nu \geq \alpha$ . In this regime, the dominating error terms would be that from micro-solver  $\mathcal{O}(h/\epsilon)^\beta$ , the filtering error  $\mathcal{O}(\epsilon)$ , 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 Sect. 2 to ODE systems and compare it with other methods.

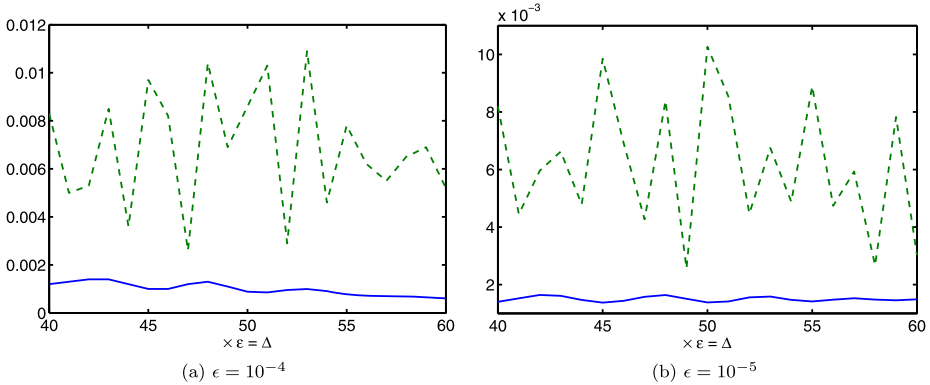
### 3.1 Nonlinear Expanding Spiral

Consider the following nonlinear equation in the complex plane

$$\dot{x} = i\epsilon^{-1}x|x| + (\sin x + \operatorname{Re}(x) \cdot x)|x|^{-2}, \tag{3.1}$$

**Table 1** (Section 3.1) BF HMM parameters

$T$	$H$	$\Delta$	$h$	Micro solver	RelTol, AbsTol (ODE45 parameters)	Macro solver
4	1/6	$40\epsilon$	$\epsilon/30$	ODE45	$10^{-13}, 10^{-10}$	Midpoint rule



**Fig. 6** (Section 3.1) Plot of an estimation error  $\|\bar{\xi}(\cdot) - \xi \circ \gamma(\cdot)\|_{L^\infty}$  with different  $\Delta$ 's over  $t \in [0, 4]$ . The dotted lines are the errors without the new filtering. A kernel  $K \in \mathbb{K}^{1,1}$  is used

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

$$\begin{cases} \dot{\xi} = \cos \phi + \xi^{-2} \{ \cos \phi \sin(\xi \cos \phi) \cosh(\xi \sin \phi) + \sin \phi \cos(\xi \cos \phi) \sinh(\xi \sin \phi) \}, \\ \xi(0) = 1, \\ \dot{\phi} = \epsilon^{-1} \xi + \xi^{-3} \{ \cos \phi \cos(\xi \cos \phi) \sinh(\xi \sin \phi) - \sin \phi \sin(\xi \cos \phi) \cosh(\xi \sin \phi) \}, \\ \phi(0) = 0. \end{cases} \tag{3.2}$$

We see immediately from Definition 1 that  $\xi$  is a slow variable. Note that (3.2) is never used in our algorithm; we use  $\xi$  in (3.2) only to show that the results computed by the algorithm is correct. The averaged equation for the slow variable  $\xi$  is

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

In this example, we used Algorithm 2, the Midpoint rule macro-solver and ODE45 micro-solver with quadratic polynomial interpolation for  $\gamma$  to compute the solution; however, in each micro-simulation, the micro-solver integrates the filtered equation

$$\dot{x}_n = i \epsilon^{-1} x_n |x_n| + K_\Delta(t - t_n) \{ (\sin x_n + \operatorname{Re}(x_n) \cdot x_n) |x_n|^{-2} \}, \quad t_n \leq t \leq t_n + \Delta,$$

the parameters in Table 1. The estimation errors for each different value of  $\Delta$  with respect to  $\epsilon = 10^{-4}$  and  $10^{-5}$  are illustrated in Fig. 6. Without the filtering technique, the resulting errors are highly oscillatory due to the sampling issue discussed in Sect. 2.1.1.

In Table 2, we show the effect of kernels on accuracy in the approximations of the slow variable  $\xi$ . Since the error tends to be dominated by the step size  $H$  of the macroscopic solver, taking a kernel with one-moment and one-regularity condition is enough to prevent the sampling error.



**Table 2** Table of  $\|\tilde{\xi}(\cdot) - \xi \circ \gamma(\cdot)\|_{L^\infty([0,4])}$  with various kernels  $\tilde{K}^{p,q}$

(a) $H = 0.2, \epsilon = 10^{-4}, \Delta = 40\epsilon$				(b) $H = 0.2, \epsilon = 10^{-5}, \Delta = 40\epsilon$			
	$q = 1$	2	3		$q = 1$	2	3
$p = 1$	2.00(-3)	1.84(-3)	1.99(-3)	$p = 1$	2.21(-3)	2.03(-3)	2.15(-3)
w/o filtering	6.76(-3)			w/o filtering	5.82(-3)		
(c) $H = 0.02, \epsilon = 10^{-4}, \Delta = 40\epsilon$				(d) $H = 0.02, \epsilon = 10^{-5}, \Delta = 40\epsilon$			
	$q = 1$	2	3		$q = 1$	2	3
$p = 1$	2.14(-4)	2.79(-4)	1.92(-4)	$p = 1$	4.85(-5)	6.13(-5)	9.05(-6)
w/o filtering	8.12(-4)			w/o filtering	1.16(-3)		

### 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, 33]):

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

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

$$\mathbf{x}' = \epsilon^{-1} \begin{bmatrix} 0 & a & 0 & 0 \\ -a & 0 & 0 & 0 \\ 0 & 0 & 0 & b \\ 0 & 0 & -b & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ x_2^2/a \\ 0 \\ 2x_1 x_2/b \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \tag{3.4}$$

where  $\mathbf{x} = [x_1, v_1, x_2, v_2]^T$ . One seeks approximation of the effective properties that takes 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 [4], three functionally independent slow variables are identified to be

$$\xi_1 = x_1^2 + v_1^2, \quad \xi_2 = x_2^2 + v_2^2, \quad \xi_3 = x_1 x_2^2 + 2v_1 x_2 v_2 - x_1 v_2^2 \tag{3.5}$$

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

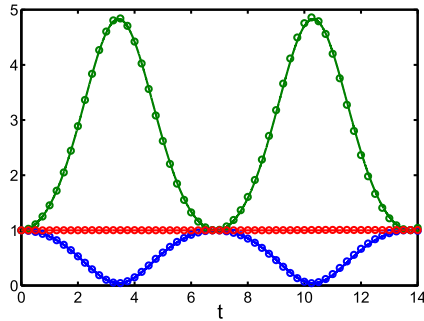
In Fig. 7, we present a result computed by our method and compare it with the results computed by FLAVORS [42] with two different sets of parameters. Figure 7a shows the BF HMM Mid-ODE45 (Midpoint rule macro-solver and ODE45 micro-solver with quadratic polynomial interpolation for  $\gamma$ ) result computed with the parameters tabulated in Table 3 and a kernel  $K \in \tilde{\mathcal{K}}^{1,4}$ . The resulting error in the slow variables is  $\max_{i=1,2,3} \|\xi_i(\cdot) - \xi_i \circ \gamma(\cdot)\|_{L^\infty([0,14])} = 0.049$ . The computational time on a one-year old desktop is about 7 s.

In Fig. 7b, we show the result computed by FLAVORS with the parameters within the recommended regimes. To be more precise, as stated in [42], the required conditions for FLAVORS are as follows:

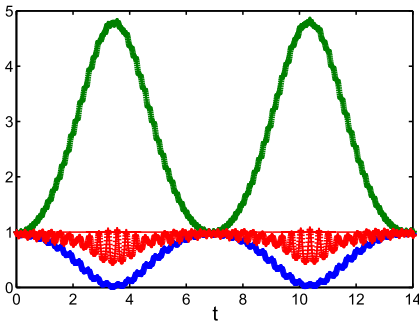
$$\Delta \ll \epsilon \ll H \ll 1 \quad \text{and} \quad \left(\frac{\Delta}{\epsilon}\right)^2 \ll H \ll \frac{\Delta}{\epsilon}. \tag{3.6}$$

**Table 3** (Section 3.2) BF HMM parameters for Sect. 3.2

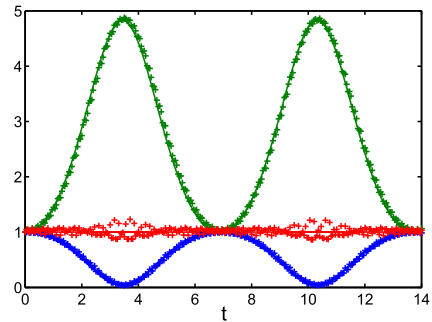
$\epsilon$	$T$	$h$	$\Delta$	$H$	Micro solver	RelTol	Macro solver
$10^{-4}$	14	$\epsilon/30$	$15\epsilon$	0.25	ODE45	$10^{-4}$	Midpoint rule



(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  $\Delta = 20\epsilon$ ,  $H = 0.05$ .

**Fig. 7** (Section 3.2) The dynamics of the slow variables  $\xi_1$ ,  $\xi_2$  and  $\xi_3$  in (3.4). Subfigures (b) and (c) FLAVORS fail to preserve the geometrical structure of the flow

In [42], the proposed empirical choice is given by  $\Delta = \gamma\epsilon$ ,  $H = \gamma\frac{\Delta}{\epsilon}$  where  $\gamma$  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  $\max_{i=1,2,3} \|\xi_i(\cdot) - \xi_i \circ \tilde{\mathbf{x}}(\cdot)\|_{L^\infty((0,14))} = 0.56$ . The computational time on the same machine is about 3.1 s.

In Fig. 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  $\max_{i=1,2,3} \|\xi_i(\cdot) - \xi_i \circ \tilde{\mathbf{x}}(\cdot)\|_{L^\infty((0,14))} = 0.23$ . The computational time is about 8.4 s.

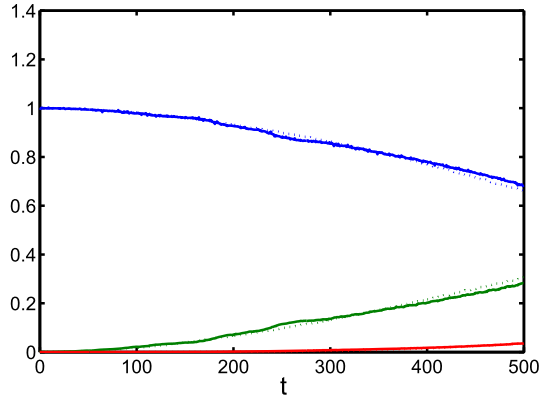
### 3.3 The Fermi-Pasta-Ulam Problem

The Fermi-Pasta-Ulam problem is a dynamical system which reveals 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 [29]. The model is derived from the

**Table 4** (Section 3.3) BF HMM parameters for Fig. 8

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

**Fig. 8** (Section 3.3) The *solid lines* correspond to the direct numerical simulation (DNS) solution with an exponential integrator. *Dotted lines* correspond to the HMM



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 [4], 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

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

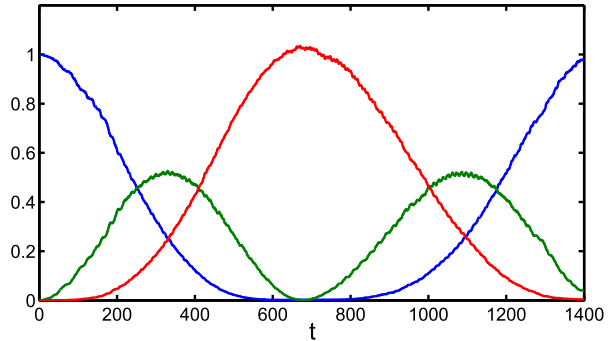
where  $I_i: \mathbb{R}^{12} \rightarrow \mathbb{R}$ . See [10, 11, 28] and references therein for some recent progress. With  $\epsilon$  denoting the time scale of the fast oscillations, the nontrivial energy transfer take place in the very long  $\epsilon^{-1}$  time scale. Even if one could afford 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  $\gamma$ ) with those by an exponential integrator with Deuffhard’s filter functions [17, 29] with the stepsize  $h = 5 \cdot 10^{-7}$ , 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

**Table 5** (Section 3.3) BF HMM parameters for Fig. 9

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

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



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 Fig. 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.

### 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. The main idea of this paper is summarized in the following:

- The given highly oscillatory system is regarded as a system with lower order perturbation. The solutions of the corresponding system without the lower order perturbation are assumed to stay on some invariant manifolds.
- We construct an effective path by comparing the solutions of the equations with and without lower order perturbation. This path discloses information about how the solutions of the given perturbed system traverse through the invariant manifolds of the unperturbed system.
- 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.

**Acknowledgements** 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. Ariel, G., Vanden-Eijnden, E.: Accelerated simulation of a heavy particle in a gas of elastic spheres. *Multiscale Model. Simul.* **7**(1), 349–361 (2008)
2. Ariel, G., Engquist, B., Tsai, R.: Numerical multiscale methods for coupled oscillators. *Multiscale Model. Simul.* **7**(3), 1387–1404 (2008)
3. Ariel, G., Engquist, B., Kreiss, H.-O., Tsai, R.: Multiscale computations for highly oscillatory problems. In: *Multiscale Modeling and Simulation in Science. Lect. Notes Comput. Sci. Eng.*, vol. 66, pp. 237–287. Springer, Berlin (2009)
4. Ariel, G., Engquist, B., Tsai, R.: A multiscale method for highly oscillatory ordinary differential equations with resonance. *Math. Comput.* **78**, 929–956 (2009)
5. Ariel, G., Engquist, B., Tsai, R.: A reversible multiscale integration method. *Commun. Math. Sci.* **7**(3), 595–610 (2009)
6. Ariel, G., Engquist, B., Tsai, R.: Oscillatory systems with three separated time scales: analysis and computation. In: *Numerical Analysis of Multiscale Computations. Lecture Notes in Computational Science and Engineering*, vol. 82, pp. 23–45. Springer, Berlin (2012)
7. Ariel, G., Sanz-Serna, J.M., Tsai, R.: A multiscale technique for finding slow manifolds of stiff mechanical systems. *Multiscale Model. Simul.* **10**(4), 1180–1203 (2012)
8. Artstein, Z., Kevrekidis, I.G., Slemrod, M., Titi, E.S.: Slow observables of singularly perturbed differential equations. *Nonlinearity* **20**(11), 2463–2481 (2007)
9. Artstein, Z., Linshiz, J., Titi, E.S.: Young measure approach to computing slowly advancing fast oscillations. *Multiscale Model. Simul.* **6**(4), 1085–1097 (2007)
10. Bambusi, D., Ponso, A.: On metastability in FPU. *Commun. Math. Phys.* **264**(2), 539–561 (2006)
11. Bambusi, D., Ponso, A.: Resonance, metastability and blow up in FPU. In: *The Fermi-Pasta-Ulam Problem. Lecture Notes in Phys.*, vol. 728, pp. 191–205. Springer, Berlin (2008)
12. Bond, S.D., Leimkuhler, B.J.: Molecular dynamics and the accuracy of numerically computed averages. *Acta Numer.* **16**, 1–65 (2007)
13. Chu, J., Engquist, B., Prodanovic, M., Tsai, R.: A multiscale method coupling network and continuum models in porous media I—Single phase flow. *Multiscale Model Simul.* **10**(2), 515–549 (2012)
14. Chu, J., Engquist, B., Prodanovic, M., Tsai, R.: A multiscale method coupling network and continuum models in porous media II—Single and two phase flow. In: *Advances in Applied Mathematics, Modeling, and Computational Science. Fields Institute Communications*, vol. 66, pp. 161–185 (2013)
15. Cohen, D., Jahnke, T., Lorenz, K., Lubich, C.: Numerical integrators for highly oscillatory Hamiltonian systems: a review. In: *Analysis, Modeling and Simulation of Multiscale Problems*, pp. 553–576. Springer, Berlin (2006)
16. Condon, M., Deaño, A., Iserles, A.: 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. Deuffhard, P.: A study of extrapolation methods based on multistep schemes without parasitic solutions. *Z. Angew. Math. Phys.* **30**(2), 177–189 (1979)
18. E, W.: Analysis of the heterogeneous multiscale method for ordinary differential equations. *Commun. Math. Sci.* **1**(3), 423–436 (2003)
19. E, W., Engquist, B.: The heterogeneous multiscale methods. *Commun. Math. Sci.* **1**(1), 87–132 (2003)
20. E, W., Vanden-Eijnden, E.: Numerical techniques for multiscale dynamical systems with stochastic effects. *Commun. Math. Sci.* **1**(2), 385–391 (2003)
21. E, W., Engquist, B., Li, X., Ren, W., Vanden-Eijnden, E.: Heterogeneous multiscale methods: a review. *Commun. Comput. Phys.* **2**(3), 367–450 (2007)
22. Engquist, B., Tsai, Y.-H.: Heterogeneous multiscale methods for stiff ordinary differential equations. *Math. Comput.* **74**(252), 1707–1742 (2005)
23. Engquist, B., Holst, H., Runborg, O.: Multi-scale methods for wave propagation in heterogeneous media over long time. In: Engquist, B., Runborg, O., Tsai, R. (eds.) *Numerical Analysis of Multiscale Computations. Lect. Notes Comput. Sci. Eng.*, vol. 82. Springer, Berlin (2011)

24. Fatkullin, I., Vanden-Eijnden, E.: A computational strategy for multiscale chaotic systems with applications to Lorenz 96 model. *J. Comput. Phys.* **200**, 605–638 (2004)
25. García-Archilla, B., Sanz-Serna, J.M., Skeel, R.D.: Long-time-step methods for oscillatory differential equations. *SIAM J. Sci. Comput.* **20**(3), 930–963 (1999)
26. Gear, C.W., Kevrekidis, I.G.: Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *SIAM J. Sci. Comput.* **24**(4), 1091–1106 (2003). (electronic)
27. Gear, C.W., Kevrekidis, I.G.: Constraint-defined manifolds: a legacy code approach to low-dimensional computation. *J. Sci. Comput.* **25**(1–2), 17–28 (2005)
28. Hairer, E., Lubich, C.: On the energy distribution in Fermi-Pasta-Ulam lattices. *Arch. Ration. Mech. Anal.* **205**(3), 993–1029 (2012)
29. Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration*. Springer Series in Computational Mathematics, vol. 31. Springer, Berlin (2002)
30. Hochbruck, M., Ostermann, A.: Exponential integrators. *Acta Numer.* **19**, 209–286 (2010)
31. Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie-group methods. In: *Acta Numerica*, 2000. *Acta Numer.*, vol. 9, pp. 215–365. Cambridge Univ. Press, Cambridge (2000)
32. Kevorkian, J., Cole, J.D.: *Perturbation Methods in Applied Mathematics*. Applied Mathematical Sciences, vol. 34. Springer, New York (1981)
33. Kevorkian, J., Cole, J.D.: *Multiple Scale and Singular Perturbation Methods*. Applied Mathematical Sciences, vol. 114. Springer, New York (1996)
34. Kreiss, H.-O.: Problems with different time scales for ordinary differential equations. *SIAM J. Numer. Anal.* **16**(6), 980–998 (1979)
35. Kreiss, H.-O.: Problems with different time scales. In: *Acta Numerica*, 1992, pp. 101–139. Cambridge Univ. Press, Cambridge (1992)
36. Kreiss, H.-O., Lorenz, J.: Manifolds of slow solutions for highly oscillatory problems. *Indiana Univ. Math. J.* **42**(4), 1169–1191 (1993)
37. Leimkuhler, B., Reich, S.: *Simulating Hamiltonian Dynamics*. Cambridge Monographs on Applied and Computational Mathematics, vol. 14. Cambridge University Press, Cambridge (2004)
38. Marsden, J.E., West, M.: Discrete mechanics and variational integrators. *Acta Numer.*, 357–514 (2001)
39. Petzold, L.R., Jay, L.O., Yen, J.: Numerical solution of highly oscillatory ordinary differential equations. In: *Acta Numerica*, 1997. *Acta Numer.*, vol. 6, pp. 437–483. Cambridge Univ. Press, Cambridge (1997)
40. Sanders, J.A., Verhulst, F.: *Averaging Methods in Nonlinear Dynamical Systems*. Applied Mathematical Sciences, vol. 59. Springer, New York (1985)
41. Sanz-Serna, J.M., Calvo, M.P.: *Numerical Hamiltonian Problems*. Applied Mathematics and Mathematical Computation, vol. 7. Chapman & Hall, London (1994)
42. Tao, M., Owhadi, H., Marsden, J.E.: 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)
43. Vanden-Eijnden, E.: Numerical techniques for multi-scale dynamical systems with stochastic effects. *Commun. Math. Sci.* **1**, 385–391 (2003)