A multiscale computation for highly oscillatory dynamical systems using EMD-type methods

Seong Jun Kim and Haomin Zhou*

May 25, 2015

Abstract

In this paper, we propose a numerical method, that combines the heterogeneous multiscale method (HMM) and the empirical mode decomposition (EMD) type of filtering techniques, to compute the slow dynamics in a multiscale system. The main idea is that we apply the Adaptive Local Iterative Filtering (ALIF) algorithm, an EMD-like nonlinear signal analysis strategy which decomposes a signal into several intrinsic mode functions (IMFs), to extract essential information, and then use it in the HMM framework to compute the coarse scale behavior without fully resolving the fine scale solutions. Our numerical examples demonstrated that the new method has a number of advantages over the existing ones. 1) The effective rate of change for the slow dynamics is calculated as the result of the local solution decomposition by ALIF, and an effective ordinary differential equation (ODE) is obtained on-the-fly by analyzing the trend of IMFs. 2) ALIF can find *hidden* intermediate time scales, with possible non-integer exponents, and the method can treat multiple (> 2) time scale systems hierarchically. 3) The time-frequency analysis in ALIF allows us to identify some frequency related slow variables, such as the relative phase. This is especially useful for the systems where the slow variables are not explicitly known.

1 Introduction

In many areas of engineering and science, we sometimes face the dilemma: we are interested in the macroscale behavior of a multiscale system, but the available empirical macroscale models are inadequate to calculate the targeted behavior for various reasons. A common one is that information regarding how the microscale dynamics influence the macroscale behavior is lacking. On the other hand, we cannot really rely on the microscopic models because they are far from being practical, either computationally or analytically. In this paper, we tackle this matter by merging two strategies, the heterogeneous multiscale method (HMM) [17] and the empirical mode decomposition (EMD) [28] for nonlinear signal analysis, so that it combines the efficiency of macroscale models and the accuracy of microscale models for problems modeled by an ordinary differential equation (ODE).

In the HMM framework, it is often assumed that the form of the macroscale model is roughly known, but some of its details related to the microscale model are missing. The missing information can be supplied by solving a microscale model. To reduce the computational cost, the microscale model is solved locally over the time intervals just large enough for the required accuracy for the

*School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA. skim396@math.gatech.edu (Seong Jun Kim), hmzhou@math.gatech.edu (Haomin Zhou).

Zhou is partially supported by NSF Awards DMS–1042998, DMS–1419027, and ONR Award N000141310408.

macroscale model. The time intervals for the microscale model are well separated in the physical space, and they do not have any direct communications among them, except via the macroscale solver.

Since its proposal more than ten years ago, the HMM for highly oscillatory ODEs has been studied intensively with many applications [2, 3, 4, 5, 6, 7, 18, 38]. It typically tackles the computational difficulty by utilizing an important consequence of scale separation. If the influence of the fast scale on the slower scale dynamics can be obtained by performing short time simulations, it is possible to obtain a numerical complexity that is much smaller than direct simulations of the given systems. This feature exploits special structures of highly oscillatory problems. In the envelope methods [36], fast oscillations are sampled cleverly in order to extrapolate in a much larger time step. Similar techniques are also used in stochastic differential equations [40]. See also a parallel-in-time (parareal) version of multiscale algorithm in [9].

While the HMM approach is attractive, there are still challenges due to its "multi-grid" type of coupling, namely at each macro time step, the solver acquires the necessary information by resolving microscale models. The first lies in the way of extracting approximations to the needed data from the microscale simulations. Even though a method using high order accurate kernels is discussed in [18], it is most useful for the fast dynamics that have specific structure such as one-dimensional periodicity. The second is that the time scales each model aims at have to be "predetermined". For example, if the time scale of the microscale model is either unknown or chosen incorrectly, one may not expect the resulting HMM to produce reliable results. The third is the need of an explicit form of slow variables. The existing HMM algorithms use a set of slow variables for computing the macroscale behavior of a highly oscillatory dynamical system. To achieve this, the set of slow variables must be either analytically derived, or the corresponding polynomials must be numerically determined a priori.

To overcome such difficulties in HMM, we propose a new method that combines the HMM with an EMD-type method, called the Adaptive Local Iterative Filtering (ALIF) algorithm [14]. EMD, first proposed by Huang and collaborators in [28], is a signal analysis strategy complementary to the classical Fourier and wavelet based method. It aims at decomposing a signal into several components called intrinsic mode functions (IMFs). Each IMF is an oscillatory function, similar to an amplitude-modulation (AM) and phase-modulation (PM) function, admitting better instantaneous frequencies. In the last two decades, EMD has been used in many applications, especially to problems involving nonlinear and non-stationary signals [27, 44]. In recent years, an ensemble EMD method (EEMD) was proposed to improve the stability of original EMD method [43]. Optimization framework has been introduced [23, 24], and adaptive wavelet techniques, such as the syncrosqueezed wavelets, have been developed too [15, 20]. A review of the related works can be found in [29]. Among various versions for EMD, ALIF uses an iterative filtering strategy with an adaptive and data driven filter lengths to extract IMFs.

In this paper, we integrate ALIF into the framework of HMM. Our main idea is to use ALIF to identify existing time scales in the system, and then design different solvers for them. We also use ALIF to pass information between scales. Although, HMM and ALIF are originally designed for completely different purposes. we find that their merging leads to additional benefits. Firstly, our new algorithm can identify fast and intermediate time scales, by performing ALIF hierarchically, from the fine scale to coarse scale. It can capture hidden intermediate scales and/or scales with non-integer exponents. Secondly, the trend component, the last IMF obtained in ALIF, provides the effective rate of change for the slow variables. It is used to supply the missing information in the macroscale model, and construct the effective equation on-the-fly while solving the problem forward in time. Lastly, for multiscale problems in which the slow variables are not explicitly known, or not available, ALIF allows us to construct some slow variables, such as the relative phase, so as

to carry out the large scale computations without a macroscale model. Those added advantages of our method overcome many challenges faced by the existing methods. We shall use numerical examples to illustrate those features. To the best of our knowledge, this study is the first reported one on using HMM and EMD in an integrated manner.

To be self-contained, we provide a short review of HMM and ALIF in the next section. We describe the detailed strategy merging them in Section 3. The resulting algorithm and many numerical examples are presented in Sections 4 and 5 respectively. Finally, we make a short conclusion in Section 6.

2 Brief reviews of HMM and ALIF

We begin with brief reviews of HMM and ALIF which are two essential ingredients for our method. We also use this opportunity to introduce notations used throughout the paper.

2.1 The HMM framework

Let us consider a multiscale problem given by the following differential equation

$$\frac{d}{dt}u = f(u, \epsilon), \tag{2.1}$$

with initial condition $u(0) = u_0 \in \mathcal{D}_0 \subset \mathbb{R}^d$ and ϵ a small parameter. Here u is the solution at the fine scale, which is often called a microscale variable in literature. We assume that (2.1) has a unique bounded solution in $[0, T]$, which is the time interval of interest and also referred as the $\mathcal{O}(1)$ time scale in the literature due to the independence of ϵ. Usually, equation (2.1) is considered as a stiff problem that requires to use a small time step, often at the finest scale such as $O(\epsilon^k)$, $k > 0$, to compute the solution correctly if the conventional numerical solvers are employed. This implies that the computational complexity over the interested time interval is at least of the order $O(\epsilon^{-k})$. This is inefficient, especially when one is only interested in a set of slowly changing quantities ξ that can be derived from u. For example, ξ could be the averaged kinetic energy of a particle system u.

When only two time scales are involved, such as the fine scale $O(\epsilon)$, in (2.1), a quantity ξ is commonly referred to as a slow variable if itself and its derivative $d\xi/dt$ are bounded for $0 < \epsilon \leq \epsilon_0$. For example, in a system containing $x_1(t) = \cos(t)$ and $x_2(t) = \cos(t/\epsilon)$, $x_1(t)$ is regarded as slow, and $x_2(t)$ is fast.

When additional time scales are involved, however, this bounded derivative characterization is not sufficient anymore. There may exist variables whose derivatives are not bounded for $0 < \epsilon \leq \epsilon_0$ but they are still regarded as slow variables. For example, consider a system which contains $x_1(t)$ and $x_2(t)$ as given before, and $x_3(t) = \cos(t) + \epsilon^{0.5}\cos(t/\epsilon)$ with $dx_3/dt = \mathcal{O}(\epsilon^{-0.5})$. There exists an intermediate $O(\epsilon^{0.5})$ scale. In this case, we regard $x_3(t)$ as slow because $|x_3(t) - \cos(t)| = \mathcal{O}(\epsilon^{0.5})$ which is slow compared to the fast variable $x_2(t)$. More examples can be found in [2, 7]. This motivates us to give the following definition for slow variables if multiple (> 2) time scales are presented. For simplicity, we will focus only on three time scales $O(\epsilon^k)$, $O(\epsilon^q)$ and $O(1)$ with $k \geq 1$ and $0 < q < k$. More scales can be considered in a similar manner.

Definition 1. A smooth function $a(t)$ is said to be slow if there exists a smooth function $b(t)$ such that $|db/dt| \leqq C$ and $|a(t) - b(t)| \leq C\epsilon^q$ for some constant C independent of ϵ with $0 < \epsilon \leq \epsilon_0$ in $t \in [0, T]$. Moreover, a smooth function $\xi(u) : \mathcal{D}_0 \to \mathbb{R}$ is called a slow variable with respect to $u(t)$ if $\xi(t) = \xi \circ u(t)$ is slow.

In many problems, there exists a diffeomorphism $\Psi : u \to (\xi(u), \phi(u))$, which explicitly separates the time scales of (2.1), such that the dynamics satisfies an ODE of the form

$$\begin{cases} \frac{d}{dt}\xi = g_0(\xi, \phi, \epsilon), & \xi(0) = \xi_0, \\ \frac{d}{dt}\phi = \epsilon^{-k} g_1(\xi, \phi, \epsilon) + g_2(\xi, \phi, \epsilon), & \phi(0) = \phi_0, \end{cases} \tag{2.2}$$

where the slow variables $\xi \in \mathbb{R}^{d-n}$ and the fast variables $\phi \in \mathbb{R}^n$. It is assumed that the fast variables are ergodic with respect to an "invariant" manifold which slowly changes along with the slow variables, and this is often referred as the ergodicity assumption in the literature. For highly oscillatory dynamical systems, such an manifold is diffeomorphic to an n-torus.

The objective of HMM is to construct and compute the macroscale model for the slow variables ξ. The HMM assumes the macroscale model

$$\frac{d}{dt}\xi = F(\xi, D_u), \tag{2.3}$$

where D_u is the data that supplies the necessary information passing from the microscale (2.1) to the macroscale (2.3) models. If the mathematical expression of F is explicitly known and the data D_u can be easily obtained, then one can solve (2.3). Unfortunately, this is not the case in many situations, especially the types of multiscale problems that we are interested in. Therefore, the main task is how to approximate the unknown F by processing D_u from localized simulations of (2.1).

One of the tools widely used by the existing multiscale methods is the theory of averaging [10, 11, 33, 37]. Under the scale separation and the ergodicity assumptions, and with sufficiently smooth right hand side, $\xi(t)$ converges, in the limit of $\epsilon \to 0$, to the solution of an averaged/effective equation given by

$$\frac{d}{dt}\bar{\xi} = G(\bar{\xi}), \quad \bar{\xi}(0) = \xi_0 \tag{2.4}$$

where

$$G(\bar{\xi}) = \int g_0(\bar{\xi}, \phi)\mu_{\bar{\xi}}(d\phi).$$

Moreover, the approximation is of order ϵ in the supremum norm, i.e.

$$\sup_{t\in[0,T]} \left|\xi(t) - \bar{\xi}(t)\right| \le C\epsilon^p, \tag{2.5}$$

where $C > 0$ is a constant that typically depends on T but is independent of ϵ, and $p > 0$ depends on the property for ϕ to fill an invariant manifold that is diffeomorphic to an n-torus. Accordingly, we choose the effective ODE (2.4) as the macroscale model (2.3) because the solutions of the systems remain close for the time interval $[0, T]$ of interest. Moreover, (2.4) can be integrated with a larger step size than the one required for (2.1). However, a difficulty still exists because (2.4) may not be explicitly available.

The HMM framework tackles the difficulty by coupling macro and micro solvers in which the effective force G is estimated by solving the microscale system (2.1). The schematic scratch for the general method is illustrated in Figure 1 for a two scale scenario. The upper axis sketches the time marching to compute $\bar{\xi}$ by an ODE solver for (2.4) with time step H, the macroscale time step. When one evaluates the right hand side $G(\bar{\xi})$, the necessary information, such as D_u, is acquired from the microscale solution u on a local time interval of length η, and u is solved by a different ODE solver with time step h depicted on the lower axis. We use $\mathcal{Q}$ to denote the operator extracting information D_u on the coarse scale from u. On the other hand, when one computes the

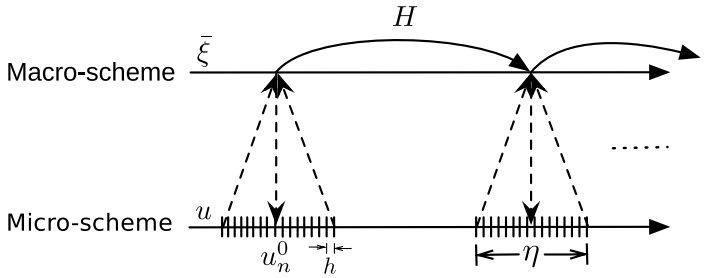


Figure 1: A typical structure of the HMM. This structure is suitable for two scale systems.

local solution u of (2.1), an initial value must be provided. This can be done by reconstructing the fine scale solution from the slow variable $\bar{\xi}$ and some conditions that couple the macro and micro scale models. We denote this procedure by $\mathcal{R}$. Essential questions including the choice of $\mathcal{R}$ and $\mathcal{Q}$ that transfer the information between different scales and the construction of a well-defined F are widely considered in the related work [3].

The known drawback of the existing HMMs for highly oscillatory ODEs is that some of them only work under restrictive assumptions: 1) efficient and accurate data processing is achievable only if the fast dynamics is ergodic with respect to a one-dimensional torus with fixed ξ, 2) time scales are well-separated in terms of integer powers of ϵ and explicitly given, and 3) the slow variables must be explicitly identified before the computation takes place. The purpose of this paper is to introduce new tactics for the HMM which relaxes those restrictions.

2.2 ALIF for signal decomposition

EMD aims at decomposing a signal into a sum of finitely many IMFs [28, 41, 42, 43]. Huang's EMD algorithm has an iterative structure, called the sifting process, which can be described as follows: Let $\mathcal{L}$ be an operator to obtain the moving (local) average of a signal $f(t)$ and $\mathcal{S}$ be an operator capturing the fluctuation part, defined by $\mathcal{S}(f)(t) = f(t) - \mathcal{L}(f)(t)$. Then the first IMF is given by

$$I_1(t) = \lim_{n\to\infty} \mathcal{S}^n(f)(t)$$

where $\mathcal{S}^n(f) = \mathcal{S} \circ \cdots \circ \mathcal{S}(f)$, meaning S is applied to the signal recursively for n times. The limit is reached in the sense that applying $\mathcal{S}$ does not change the outcome any more, and $I_1(t)$ is the first IMF. The subsequent IMFs are obtained one after another by

$$I_k(t) = \lim_{n\to\infty} \mathcal{S}^n(f(t) - I_1(t) - ... - I_{k-1}(t))$$

The process stops when the reminder $r(t) = f(t) - I_1(t) - ... - I_m(t)$, also called trend, has at most one local maximum or one local minimum. Then, the decomposition of $f(t)$ is expressed by

$$f(t) = \sum_{k=1}^{m} I_k(t) + r(t).$$

In the original EMD paper [28], each IMF $I_k(t)$ is designed to satisfies two properties: (1) the numbers of extrema and zero crossings must either equal or differ at most by one; (2) a local average, such as the one obtained by averaging the upper and lower envelopes in Huang's original EMD method, needs to be (near) zero at any point.

In literature, various methods have been proposed to compute the moving average $\mathcal{L}(f)(t)$. In particular, we adopt the ALIF procedure proposed in [14], in which the moving average is calculated by convolutions with low pass filters with adaptive filter lengths, i.e.

$$\mathcal{L}(f)(t) = \int_{-l(t)}^{l(t)} f(t+s)w(t,s)ds,$$

where $w(t,s)$ is a low pass filter and $l(t)$ the filter length changed adaptively depending on the signal itself. We refer to [14] for the details of the selections of $w(x,t)$, $l(x)$ and implementations of ALIF. We want to mention that we choose ALIF in our implementation because of its simplicity and stability. The proposed method of combining HMM and EMD may work as well for other EMD algorithms, e.g., [23, 25, 26].

3 Merging HMM and ALIF

In the following sections, we propose that by combining the HMM with ALIF, it is possible to design a new multiscale method which can tackle several challenges that the existing multiscale algorithms cannot handle.

3.1 The main ideas

The purpose of using ALIF is to supply the needed data D_u in the macroscale system of the HMM. A signal is calculated by solving the microscale model locally, and the trend function resulted from ALIF captures the effective rate of change of slow variables. Indeed, we decompose a short time solution of the slow variable $\xi(t)$ into the IMFs $I_k(t)$, and the trend which is identified as $\bar{\xi}(t)$. That is,

$$\xi(t) = \sum_{k=1}^{m} I_k(t) + \bar{\xi}(t). \tag{3.1}$$

The behavior of the slowest time scale is encoded in $\bar{\xi}(t)$, and a group of IMFs correspond to the behaviors of the fast time scales. Figure 2 shows an example of a signal involving three time scales decomposed into the IMFs by ALIF.

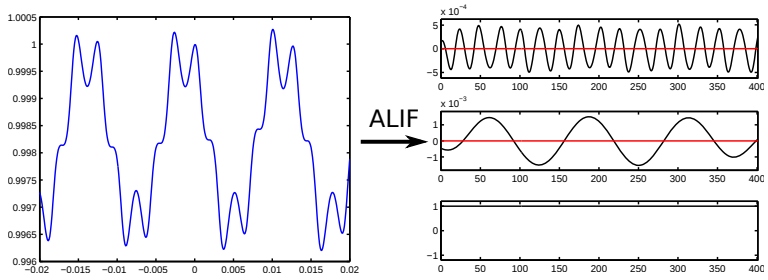


Figure 2: A signal from the trajectory of a multiscale dynamics. The resulting three IMFs show the fast, intermediate, and slow time scale behaviors, respectively.

The advantage of using a time-frequency analysis in the computation of highly oscillatory dynamical systems is three fold. Firstly, the effective rate of change for the slow dynamics is calculated as the result of the local solution decomposition by ALIF, and an effective ODE is obtained on-the-fly by analyzing the trend of a signal; Secondly, ALIF can find hidden intermediate time scale. This

motivates us to design a numerical method which applies the two-scale algorithms hierarchically to multiple (> 2) timescale systems. In [2, 7], the time scales of the system are assumed to be $\mathcal{O}(\epsilon^2)$, $\mathcal{O}(\epsilon)$, and $\mathcal{O}(1)$. This will be generalized into the system whose multiscale features are not specified as integer powers of ϵ, e.g., in nearly resonant coupled harmonic oscillators [32, 34]. Indeed, we assume that the time scales are of $\mathcal{O}(\epsilon)$, $\mathcal{O}(\epsilon^q)$, and $\mathcal{O}(1)$ with $0 < q < 1$. The unknown q is approximated by performing a signal decomposition of the slow trajectory with different ϵ's. Lastly, the time-frequency analysis allows us to identify the frequency-related slow variables, e.g., the relative phase. In [4], a numerical method of determining slow variables was developed. Such method can determine the slow variables which are only represented by polynomials. Other approaches to find slow variables include, e.g., [12, 13]. We propose a method to identify the slow variables originated from the resonance in frequencies.

3.2 Numerical averaging method using the ALIF

The ALIF is directly applied to the averaging theory for (2.2) whose solutions possess slow variables or observables as well as fast oscillations with multi-frequencies that are almost rationally resonant. The main purpose of numerical averaging is to approximate the effective ODE (2.4) and make the macroscale model explicitly available. Our generalization is along the lines of the so called averaging over multiple angles in the case of ϵ-dependent variable frequencies. For the relatively simple case of an 1-torus, a circle, averaging can be accelerated using convolution with respect to specially constructed one-dimensional averaging kernels [2, 3, 4, 5, 6, 7, 18, 38]. However, generalizations to higher dimensional n-torus, $n \geq 2$, is not trivial due to new challenges in the construction and implementation of the numerical averaging. In contrast, the numerical averaging strategy presented in this section is seamless in the sense that it does not require knowing the dimension of the torus.

To study the slow behavior of solutions, we first solve the microscale system for short time $[t - \eta, t + \eta]$ using suitable initial data and plug the trajectories of slow variables into the ALIF. Here we assume that the slow variables are available using the algorithm in [4] for the convenience of presentation. This assumption will be relaxed in the next subsection. Using the ALIF, the slow variables is decomposed into as (3.1),

$$\xi(t) = \sum_{k=1}^{m} I_k(t) + \bar{\xi}(t). \tag{3.2}$$

The trend $\bar{\xi}(t)$ shows the averaged-out effective trajectory of the slow variable. Therefore, the time derivative of $\bar{\xi}$ in (2.4) is approximated by

$$G(\bar{\xi}(t)) \approx \frac{\bar{\xi}(t + \bar{\eta}) - \bar{\xi}(t - \bar{\eta})}{2\bar{\eta}} \tag{3.3}$$

where $\bar{\eta}$ is strictly less than η.

We remark that in (3.3) sampling the $\bar{\xi}(t)$ not in both endpoints but $\bar{\eta}$ amount inside is typically due to the endpoint effect in the decomposition. As an EMD-like method, there also exists a problem of endpoint effect in the ALIF which makes the result of decomposition sensitive to the end points. Thus, our treatment to avoid the endpoint effect is sampling the trend at the interior points which are automatically suggested by the algorithm.

In the next section, we will focus on the resulting IMFs.

3.3 Identification of the intermediate time scale

We introduce a method to identify the hidden intermediate time scale in highly oscillatory dynamical systems. This technique is necessary, for example, in the appropriate averaging procedure depending on the type of near-resonances present and the time scale of interest [8, 11]. The wrong averaging method may yield an incorrect approximation for the highly oscillatory terms in the averaged equation and thus the wrong effective dynamics.

We assume that the system (2.1) exhibits three time scales $\mathcal{O}(\epsilon)$, $\mathcal{O}(\epsilon^q)$ and $\mathcal{O}(1)$ where $0 < q < 1$ is unknown and that both the ϵ and ϵ^q scales are oscillatory. For fixed t, and $\eta \gg \epsilon$, multiple sets of local simulations for short time $[t-\eta, t+\eta]$ with different ϵ's $\epsilon_l = M^{-l}\epsilon_0$, $l = 0, 1, \cdots, L$, are implemented. We then plug the trajectories of slow variables into the ALIF and look at how the frequencies of each IMF vary with respect to different ϵ's. The key point is that since the number of extrema and the number of zero crossings in the IMFs must either equal or differ at most by one, variations in the frequency of IMFs must be reflected in the different number of extrema. Denote the number of the extrema in the IMF by

$$\#(I_i; l) = \underset{s\in[t-\eta,t+\eta]}{\text{nminmax}} \{I_i(\xi(s); l)\}$$

where nminmax is the number of extrema. If the oscillation takes in $\mathcal{O}(\epsilon^q)$ time scale, then the period is given by the length of the time interval divided by $\#(I_i; l)$,

$$\frac{2\eta}{\#(I_i; l)}.$$

With the different $l_1 < l_2$, we thus have

$$M^{q(l_2-l_1)} = \frac{\#(I_i; l_2)}{\#(I_i; l_1)},$$

and the approximation for the intermediate time scale is

$$q = \frac{1}{l_2 - l_1} \log_M \frac{\#(I_i; l_2)}{\#(I_i; l_1)}.$$

In Sections 5.2 and 5.3, we apply above method to identify the hidden intermediate time scale due to the resonance in two coupled harmonic oscillators.

The discussion on identifying the intermediate time scales motivates a numerical method which applies the previous two-scale HMM algorithms [1, 3, 4, 5, 6, 18] hierarchically to multiple timescale systems. In Section 4, we introduce the HMM to evaluate the effective rate of change of $\xi(t)$.

3.4 Identification of the hidden slow variables

A common goal of multiscale methods is to accurately compute all slow variables. To achieve this goal, some multiscale methods require to identify a complete set of slow variables a priori. A difficulty lies in the fact that the system may have hidden slow variables. In some literature, this issue is caused by a presence of resonance [3, 18, 19, 39]. In this section, we propose a method to identify a set of frequency related slow variables which are not trivial as the amplitude related slow variables.

Consider a system of ODEs of the form

$$\frac{d}{dt}u = \epsilon^{-1}A(\epsilon)u + f(u, t), \tag{3.4}$$

where $u \in \mathbb{R}^d$, and A is a real, diagonalizable $d \times d$ matrix whose eigenvalues are non-zero and are pure imaginary. We assume that (3.4) is a corresponding first order linear system of r second order ODEs, so $d = 2r$. The following lemma considers the slow variables in the oscillators.

Lemma 1. *There exists a neighborhood $\mathcal{U}$ of u and a diffeomorphism*

$$\Phi : u \mapsto (\Xi, \Phi, \phi) \in \mathbb{R}^r \times \mathbb{R}^{r-1} \times S^1,$$

such that $\Xi = (\Xi_1, \Xi_2, \cdots, \Xi_r) \in \mathbb{R}^r$ and $\Phi = (\Phi_1, \Phi_2, \cdots, \Phi_{r-1}) \in \mathbb{R}^{r-1}$ are slow variables with respect to (3.4).

Proof. See [4] for the proof. □

A typical way for mapping the dynamics into $2r-1$ slow variables and a single fast variables in S^1 is

$$\begin{aligned} \Xi_k &= x_{2k-1}^2 + x_{2k}^2, \quad k = 1, \cdots, r, \\ \Phi_k &= \arg(x_{2k-1} + i x_{2k}) - c_k \arg(x_{2k+1} + i x_{2k+2}), \quad k = 1, \cdots, r-1. \end{aligned} \tag{3.5}$$

In this change of coordinates, phase related slow variables $\Phi_k(t; c_k)$ are nontrivial due to the unknown phase ratio c_k, and our goal is to identify these slow variables using the time-frequency analysis.

The main idea is from the fact that the more inaccurate the ratio of two phases is, the more extrema in the IMFs are generated if we decompose a 2π-modulo of Φ_k. The desired c_k is then sought by minimizing the number of extrema and the phase difference. Consequently, the following two-stage minimization strategy is proposed. We first solve

$$\min_{c_k \in \mathbb{R}} \{|\text{nminmax}\{I_1(\Phi_k(t; c_k))\}|\} \tag{3.6}$$

where $\Phi_k(t; c_k)$ is a short time solution of (3.4) over $[t_0 - \eta, t_0 + \eta]$ and $I_1(\Phi_k(t; c_k))$ is the first IMF obtained by applying the ALIF to $\Phi_k(t)$. If the minimization of (3.6) is achieved with $c_k = c^*$, we solve the following minimization problem with c^* as the initial condition.

$$\min_{c_k \in \mathbb{R}} \left\{ |\max\{I_1(\Phi_k(t; c_k))\} - \min\{I_1(\Phi_k(t; c_k))\}|^2 \right\}. \tag{3.7}$$

We propose to solve the above two-stage minimization problem by letting

$$E_1 = |\text{nminmax}\{I_1(\Phi_k(t; c_k))\}|, \quad E_2 = |\max\{I_1(\Phi_k(t; c_k))\} - \min\{I_1(\Phi_k(t; c_k))\}|^2.$$

Then the best c_k is to minimize the functional E_1 and then E_2. Since c_k lies in one-dimensional space, the search direction, denoted by p, is either $+1$ or -1 in the following algorithm.

Algorithm of identifying phase related slow variables

1. Minimization of (3.6).
   (a) Set initial $c_k^{(0)}$, the search direction $p^{(0)}$, and the step length $h^{(0)}$.
   (b) Iterate $c_k^{(i)} = c_k^{(i-1)} - h^{(i-1)} p^{(i-1)}$ until $E_1 = 0$ is reached. The sign of $p^{(i)}$ and the size of $h^{(i)}$ are adaptively changed toward the direction of decreasing E_1.
   (c) Denote the solution by $c_k^{(0)} = c^*$, and the last iteration by N.

2. Minimization of (3.7).

   (a) Set $c_k^{(N)} = c^*$ and the step length $h^{(N)}$.

   (b) Iterate $c_k^{(i)} = c_k^{(i-1)} - h^{(i-1)}\nabla E_2(c_k^{i-1})$, $i \geq N+1$, until a fixed point is reached. The size of $h^{(i)}$ is adaptively changed toward the direction of decreasing E_2.

3. Repeat for $k = 1, \cdots, r-1$.

In Step 1, the nminmax function is inappropriate for using a steepest descent method because it is not differentiable for all values. Detailed reviews and further references on the better optimization techniques can be found in the active field of "line search algorithms" in [35].

Two scenarios in the usage of above algorithm will be considered: (i) when two time scales are involved in (3.4), the dynamics of the given system is decomposed into a set of slow variables, including the variables found by above method and a single fast oscillating mode defined on a circle. With these slow variables, a two scale multiscale algorithm can be applied, and (ii) when three scales are involved, the intermediate time scales are identified by the method in Section 3.3. Accordingly, a multiscale algorithm requires to resolve the variables evolving on the intermediate time scale. We thus focus on the identification of these variables. It turns out that the value of phase ratio c_k depends on ϵ, and we will construct the variables on the intermediate time scale by modifying the form of Φ_k. See Section 5 for the practical applications of the algorithm.

4 The multiple scale algorithm

Once the intermediate time scale is identified, the relevant multiscale algorithm is constructed as a family of multilevel solvers which resolve the different time scales and efficiently computes the effective behavior of the slow time scale. The main difference from multiple time scale algorithms in [2, 7] is that our algorithm *determines* time scales and *decides* which time scale is resolved by each solver.

It is also worth pointing out that although our method approximate the effective ODE by averaging out the fast oscillations in the dynamics and this is very similar to the algorithms reported in [2, 3, 4, 5, 6, 7, 18], we do not discard the fast oscillations like the other methods do. Instead, the fast oscillations are retained and analyzed in order to obtain the additional information on the dynamics. We emphasize that the computational cost of our algorithm remains as effective as other two-scale algorithms whose complexity is sub linear to ϵ^{-1}, although our algorithm considers multiple time scales.

The multiscale method to be constructed should evaluate the effective rate of change of the slow variables $\xi(t)$. For simplicity of presentation, we only consider the systems with three time scales. The hierarchical structure for three scale systems is illustrated in Figure 3. The downward pointing arrows depict the determination of an initial condition for a lower, fast scale from data in an upper tier working on a slower time scale. The upward pointing arrows from 2nd tier to 1st tier and 1st tier to 0th tier relate the evaluation of averages with respect to $\mathcal{O}(\epsilon)$ and $\mathcal{O}(\epsilon^q)$, respectively. Below we detail the equations solved in each tier. We denote η_i and h_i the range of integration and the step size used in the i-th tier, respectively.

The 0th tier numerically approximates the $\mathcal{O}(1)$ time scale behavior by integrating the effective ODE (2.4),

$$\frac{d}{dt}\bar{\xi} = G(\bar{\xi}), \quad \bar{\xi}(0) = \xi_0 \tag{4.1}$$

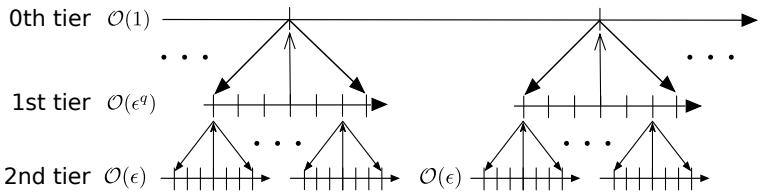


Figure 3: An illustration of a three scale algorithms.

in $t \in [0, \eta_0] = [0, T]$. To achieve this, we resolve fast $\mathcal{O}(\epsilon^q)$ and $\mathcal{O}(\epsilon)$ time behaviors in the lower tiers. Two approaches for numerically averaging the fast oscillations are considered. The first is to use the ALIF as presented in Section 3.2 which does not require the knowledge of the manifold of fast variables. The second is to apply the averaging kernels as presented in the various HMM literatures. This method yields more accurate averaging for the system if the fast variable is ergodic with respect to a circle for fixed slow variables. The numerical experiment to compare the ALIF with the recent HMM using a Poincaré map [1] is presented in Section 5.1. In the following, we present an algorithm which applies both the ALIF and the Poincaré map in the averaging procedures.

The 1st tier numerically approximates the effective ODE for the $\mathcal{O}(\epsilon^q)$ time scale. Indeed, the computation of $\xi(t)$ with $t \in [t_n - \eta_1, t_n + \eta_1]$ is achieved by the Poincaré 1st order method (Forward Euler for the ϵ^q time scale and ODE45 2nd tier solver with linear polynomial interpolation for the effective path). We compute γ_{m+1} from γ_m at $t_{n,m} = nh_0 + mh_1$, $m = -\eta_1/h_1 + 1, \cdots, -1, 0, 1, \cdots, \eta_1/h_1 - 1$,

$$\gamma_{m+1} = \gamma_m + h_1 \mathcal{F}^1_{HMM}(\gamma_m, t_{n,m}), \tag{4.2}$$

where $\mathcal{F}^1_{HMM}$ is defined in the 2nd tier, and the initial value at $t = t_{n,0}$ is an approximation of $u(t)$ at $t_n = nh_0$ obtained from the 0th tier.

The 2nd tier numerically approximates the given ODE at the initial time $t_{n,m}$ in a time segment $t \in [t_{n,m} - \eta_2, t_{n,m} + \eta_2]$ and evaluates $\mathcal{F}^1_{HMM}$. With a chosen C^q kernel $K(t)$ supported on $[0, \eta_2]$ with p moments, we approximate the effective path using linear interpolation as follows:

1. Solve
$$\frac{d}{dt}\tilde{u} = \epsilon^{-1} f_1(\tilde{u}) + K_{\eta_2}(t - t^*) f_0(\tilde{u}; \epsilon), \quad \tilde{u}(t^*) = \gamma_0^* \tag{4.3}$$
for $t \in [t^*, t^* + \eta_2]$. The initial condition γ_0^* is an approximation of $u(t)$ at $t_{n,m}$ obtained from the 1st tier. Denote the solution at $t = t^* + \eta_2$ by $\tilde{u}(\eta_2; \gamma_0^*)$.

2. Solve
$$\frac{d}{dt} z = \epsilon^{-1} f_1(z) \tag{4.4}$$
with $z(t^* + \eta_2) = \tilde{u}(\eta_2; \gamma_0^*)$ for $t \in [t^*, t^* + \eta_2]$. Denote the solution at $t = t^*$ by γ_1^*.

3. Evaluate $\mathcal{F}^1_{HMM}$:
$$\mathcal{F}^1_{HMM}(\gamma_0^*, t^*) := \frac{\gamma_1^* - \gamma_0^*}{\eta_2}. \tag{4.5}$$

We refer the readers to [1] for further details in the Poincaré map technique. Our basic algorithm is summarized below. This algorithm consists of Forward Euler in 0th tier, Forward Euler in 1st tier and ODE45 in 2nd tier as well as the averaging procedure in 1st tier using the ALIF and in 2nd tier using the linear interpolation for the effective path (Poincaré map). Thus, we shall call

our algorithm ALIF-BF HMM FE-FE-ODE45 for brevity.

ALIF-BF HMM FE-FE-ODE45

1. (Forward Euler 0th tier solver) $u_{n+1} = u_n + H\delta u_0$ where δu_0 is the least squares solution to the linear system
$$\delta u_0 \cdot \nabla \xi = \langle \xi' \rangle .$$
Approximate $\langle \xi' \rangle$ by $G(\bar{\xi}(t))$ of (3.3). The input $\xi(t)$ for the ALIF is obtained from Step 2.
2. (Forward Euler 1st tier solver) Computation of $\xi(t)$ over $t \in [t_n - \eta_1, t_n + \eta_1]$.
   (a) Compute (4.2) using the Poincaré 1st order method.
   (b) Evaluate the slow variable $\xi(t) := \xi \circ \gamma_m$.
3. (ODE45 2nd tier solver) Evaluation of $\mathcal{F}^1_{HMM}$ in Step 2(a) using ODE45 method over $t \in [t_{n,m} - \eta_2, t_{n,m} + \eta_2]$.
4. Repeat.

The next two remarks are concerned with some issues about the extensions: The above algorithm is easily generalized to the second order Midpoint rule or Verlet BF HMM using the similar strategy proposed in [1], and any two-scale method can be combined with the ALIF following our approach. For example, one can combine the ALIF with FLAVORS [39] or impulse method [21].

In our setup for the multiscale problems, we consider a regime: $0 \le t \le T$, $\epsilon \to 0$, $\eta_0 = T \sim \mathcal{O}(1)$, $\eta_1 \sim \mathcal{O}(\epsilon^q)$, $\eta_2 \sim \mathcal{O}(\epsilon)$, $h_0 = H \sim \mathcal{O}(1)$, $h_1 \sim \mathcal{O}(\epsilon^q)$, and $h_2 \sim \mathcal{O}(\epsilon)$, assuming that $\bar{\xi}(t)$ has the first derivative bounded uniformly and independent of ϵ. In this regime, the proposed method utilizes the slow varying property and generates accurate approximation of $\xi(u(t))$ with an $\mathcal{O}(1)$ complexity as $\epsilon \to 0$.

5 Numerical examples

In this section, four numerical examples are presented to illustrate the features achieved by the proposed algorithm.

5.1 Stellar orbits in a galaxy

In this example, our goal is to identify the hidden slow variables, and approximate an effective ODE for the slow variables by extracting the trend of the IMFs using the ALIF. The following system is taken from the theory of stellar orbits in a galaxy [30, 31]:

$$\begin{cases} \frac{d^2}{d\tau^2} r_1 + a^2 r_1 = \epsilon r_2^2, \\ \frac{d^2}{d\tau^2} r_2 + b^2 r_2 = 2\epsilon r_1 r_2. \end{cases}$$

With the change of variables $[x_1, v_1, x_2, v_2]^T = [r_1, \frac{d}{d\tau} r_1/a, r_2, \frac{d}{d\tau} r_2/b]^T$ and after a rescaling of time, $t = \epsilon\tau$, the system can be written in the following form

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} = \epsilon^{-1} \begin{bmatrix} 0 & a & 0 & 0 \\ -a & 0 & 0 & 0 \\ 0 & 0 & 0 & b \\ 0 & 0 & -b & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ x_2^2/a \\ 0 \\ 2x_1x_2/b \end{bmatrix} \tag{5.1}$$

where the initial condition at $t = 0$ is given by $[1, 0, 1, 1/2]^T$. In [4], when $a = 2$ and $b = 1$, resonance of oscillatory modes take effect in the lower order term. We try to identify the slow variables $\xi_i : \mathbb{R}^4 \to \mathbb{R}$, $i = 1, 2, 3$, evolving on the $\mathcal{O}(1)$ time scale. Two slow variables are related to the square of the amplitude of the two harmonic oscillators,

$$\xi_1 = x_1^2 + v_1^2, \qquad \xi_2 = x_2^2 + v_2^2. \tag{5.2}$$

There exists an additional slow variable which is a phase related variable and is assumed to be in the form of

$$\xi_3 = \arg(x_1 + iv_1) - c \arg(x_2 + iv_2). \tag{5.3}$$

The algorithm for the identification of slow variables in Section 3.4 approximates the constant $c = 2.0002$ which is independent of ϵ. Figure 4 shows $c^{(i)}$ plotted with respect to the iterations in the identification of ξ_3.

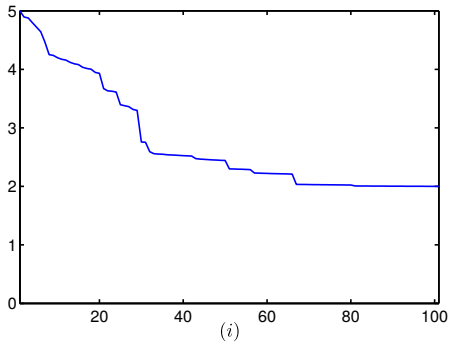
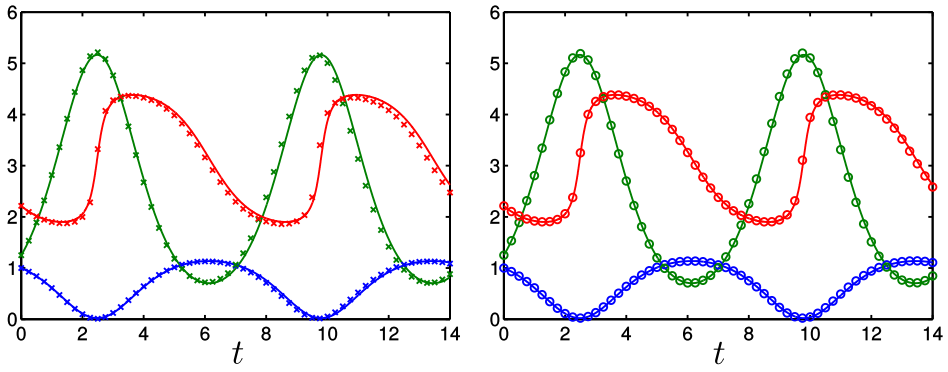


Figure 4: The trajectory of $c^{(i)}$ plotted with respect to the iterations in the algorithm in Section 3.4.

In Figure 5, we present a result computed by our two scale method and compare it with a multiscale integrator using a Poincaré map [1]. Figure 5a shows the ALIF HMM Mid-ODE45 (Midpoint rule in 0th tier and ODE45 in 1st tier with averaging procedure using the ALIF) result computed with the parameters tabulated in Table 1. Figure 5b is computed by BF HMM Mid-ODE45 (Midpoint rule in 0th tier and ODE45 in 1st tier with quadratic polynomial interpolation in Poincaré) with the same parameters, and C^3 kernel with $p = 1$ is used for the filtered equation.

Table 1: (Section 5.1) Parameters for HMM with the ALIF in the averaging.

ϵ	T	H	η_1	h_1	Micro solver	RelTol	Macro solver	Averaging method
10^{-4}	14	0.25	30ϵ	$\epsilon/20$	ODE45	1e-4	Midpoint	ALIF



(a) ALIF-HMM Mid-ODE45 (b) BF HMM Mid-ODE45

Figure 5: (Section 5.1) The dynamics of the slow variables ξ_1, ξ_2 and ξ_3 in (5.1).

Both methods are based on the numerical averaging strategy. The system (5.1) has the slow coordinate (ξ_1, ξ_2, ξ_3) which is a vector of three functionally independent slow variables, and thus one fast variable is ergodic with respect to a circle. As presented in Figure 5, the BF HMM generates more accurate approximation in this example. This is because a Poincaré map is specialized for the system where the fast oscillation in $\mathcal{O}(\epsilon)$ time scale is on a circle. The ALIF, however, can be applied in the general system where the fast dynamics are ergodic with respect to a n-torus, but generates less accurate approximations for the one dimensional case.

5.2 Harmonic oscillators I

In this example, we first identify the hidden intermediate time scale using the ALIF as described in Section 3.3, and then design a corresponding multiscale method compute the slow variables. The fast oscillations will be averaged out by the ALIF via sampling a trend. We compare the result with a two scale HMM which does not identify the intermediate time scale.

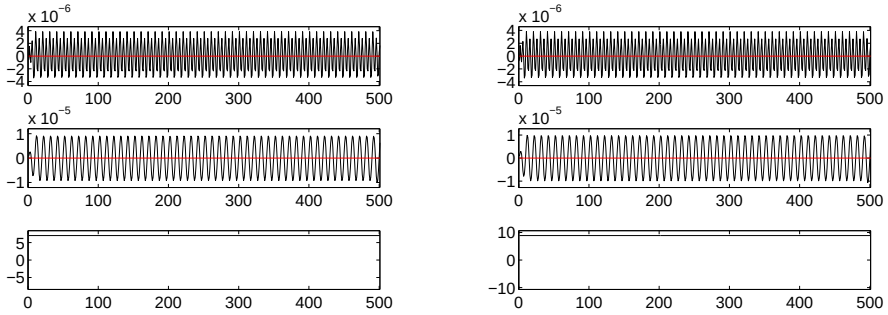
Consider the following system describing two coupled harmonic oscillators whose hidden fast dynamics is ergodic on 2-torus, $\mathbb{T}^2$.

$$\begin{cases} \frac{d}{dt}x_1 = -2\pi\epsilon^{-1}x_2 - x_1^3x_2^2(x_1^2+x_2^2)^{-3}, \\ \frac{d}{dt}x_2 = 2\pi\epsilon^{-1}x_1 - y_2(x_1^2+x_2^2)(y_1^2+y_2^2)^{-1} - x_2^3x_1^2(x_1^2+x_2^2)^{-3}, \\ \frac{d}{dt}y_1 = -2\pi\lambda\epsilon^{-1}y_2 + y_1x_2^2(y_1^2+y_2^2)^{-1}, \\ \frac{d}{dt}y_2 = 2\pi\lambda\epsilon^{-1}y_1 + x_2 + y_4x_2^2(y_1^2+y_2^2)^{-1}, \end{cases} \tag{5.4}$$

with irrational λ depending on ϵ and initial conditions $[x_1, x_2, y_1, y_2]^T = [1, 0, 2, 0]^T$. Due to the presence of λ, the near resonance between two phases happens, so the dynamics of (5.4) involves a nontrivial intermediate time scale. To see how resonances occur, we replace (x_1, x_2) and (y_1, y_2) by the polar coordinates (r_1, θ_1) and (r_2, θ_2) respectively. We then get $\dot{\theta} = h(\theta)$ where $\theta \in \mathbb{T}^2$:

$$\begin{cases} \frac{d}{dt}\theta_1 = \frac{1}{\epsilon} - \frac{r_1}{2\pi r_2}\cos 2\pi\theta_1 \sin 2\pi\theta_2, \\ \frac{d}{dt}\theta_2 = \frac{\lambda}{\epsilon} + \frac{r_1}{r_2}\sin 2\pi\theta_1 \cos 2\pi\theta_2. \end{cases} \tag{5.5}$$

The ϵ dependent variable frequency λ is irrational so that the fast variable is ergodic with respect to an invariant manifold which is diffeomorphic to $\mathbb{T}^2$. We stress that (5.5) is only used in order to demonstrate the relation between frequencies. This is not used in the numerical approximation.



(a) $I_1(t)$ (b) $I_2(t)$

Figure 6: The resulting IMFs of the slow variables I_1 and I_2 at $t = 0$.

The system (5.4) admits two hidden slow variables on the ϵ^0 scale which are the squares of the amplitude of the harmonic oscillators,

$$I_1 = x_1^2 + x_2^2, \;\; I_2 = y_1^2 + y_2^2.$$

We perform localized simulations of I_1 and I_2 over short times, and apply the ALIF to decompose these local solutions into the components. Figure 6 shows the IMFs obtained using the ALIF. The second component reveals the oscillation in the intermediate time scale.

Using the method described in Section 3.3, Table 2 shows the results of identifying the intermediate time scale. We denote $\tilde{q}$ by the approximated time scale obtained by our algorithm.

l	# of extreme points	Resulting $\tilde{q}$
0	28	
1	20	0.5146
2	14	0.4854
3	10	0.5146

(a) $\epsilon_0 = 8 \cdot 10^{-3}$, $\eta = 50\epsilon$, $M = 2$, exact $q = 1/2$

l	# of extreme points	Resulting $\tilde{q}$
0	64	
1	52	0.7004
2	40	0.6215
3	32	0.6781

(b) $\epsilon_0 = 8 \cdot 10^{-3}$, $\eta = 50\epsilon$, $M = 2$, exact $q = 2/3$

Table 2: Identification of the intermediate time scale using the ALIF applied to (5.4).

We assume the form of the other slow variable as

$$\xi_3 = \arg(x_1 + iy_1) - c\arg(x_2 + iy_2). \tag{5.6}$$

Then, the algorithm for the identification of the slow variables in Section 3.4 approximates the constant $c = (1 + \epsilon^{1-\tilde{q}})^{-1} \approx 1 - \epsilon^{1-\tilde{q}}$. By Definition 1, we have that $d\xi_3/dt = d\theta_1/dt - (1 - \epsilon^{1-\tilde{q}})d\theta_2/dt = \mathcal{O}(1)$ with $\theta_1 = \arg(x_1 + iy_1)$ and $\theta_2 = \arg(x_2 + iy_2)$. Therefore, the alternative 1:1 relative phase,

$$\theta = \arg(x_1 + iy_1) - \arg(x_2 + iy_2) \tag{5.7}$$

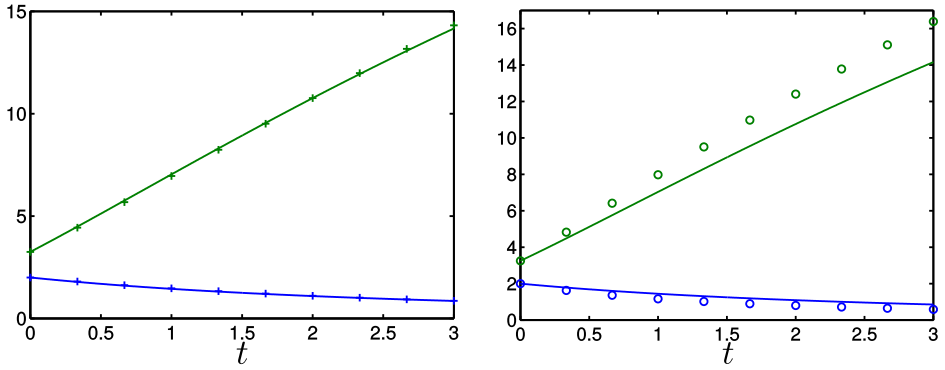
evolves in the intermediate $\mathcal{O}(\epsilon^{\tilde{q}})$ time scale. To sum up, we find two variables I_1, I_2 evolving in the slow time scale and θ evolving in the intermediate time scale. θ together with the variable in the finest time scale are on a 2-dimensional torus.

In Figure 7, we present a result computed by our 3-tire multiscale solver and compare it with the result computed by the two scale method [4]. Figure 7a shows the ALIF-ALIF Mid-Mid-ODE45

(Midpoint Rule in 0th & 1st tiers and ODE45 in 2nd tier as well as the averaging procedure in 1st & 2nd tier using the ALIF) result computed with the parameters in Table 3. In Figure 7b, we show the result computed by the two scale HMM with $\eta = 20\epsilon$, $h = \epsilon/150$ and the exponential kernel [5]. It is evident that the later fails to capture the slow variables accurately.

Table 3: (Section 5.2) Parameters for the 3-tier HMM. $\lambda = 1 + \frac{1+\sqrt{5}}{2}\epsilon^{1/3}$.

$\epsilon = 10^{-4}$	η_i	h_i	Averaging method	ODE Scheme
0th tier	3	1/3	-	Midpoint
1st tier	$3\epsilon^q/5$	$\epsilon^q/100$	ALIF	Midpoint
2nd tier	$\epsilon/5$	$\epsilon/150$	ALIF	ODE45



(a) ALIF-ALIF Mid-Mid-ODE45 (b) HMM Mid-ODE45

Figure 7: (Section 5.2) The dynamics of (5.4) on the ϵ^0 time scale. Subfigure (b) two scale HMM fails to approximate the slow variables.

5.3 Harmonic oscillators II

Consider the following system describing two coupled harmonic oscillators in resonance [7].

$$\begin{cases} \frac{d}{dt}x_1 &= -\epsilon^{-1}y_1 + \epsilon^{-1/2}y_2^2 - 3x_1x_2^2, \\ \frac{d}{dt}y_1 &= \epsilon^{-1}x_1 + y_1/2, \\ \frac{d}{dt}x_2 &= -\left(\epsilon^{-1} + \epsilon^{-1/2}\right)y_2 - x_2, \\ \frac{d}{dt}y_2 &= \left(\epsilon^{-1} + \epsilon^{-1/2}\right)x_2 - y_2 + 2x_1^2y_2. \end{cases} \tag{5.8}$$

All four state variables oscillate with a period which is of the order of ϵ. Hence, x_1, y_1, x_2 and y_2 evolve on the ϵ time scale. Initial conditions are $[x_1, y_1, x_2, y_2]^T = [0, 1, 0, 1]^T$.

As previous examples, two slow variables corresponding the oscillatory energies are

$$I_1 = x_1^2 + y_1^2,\ I_2 = x_2^2 + y_2^2.$$

We assume the form of the other slow variable as

$$\xi_3 = \arg(x_1 + iy_1) - c\arg(x_2 + iy_2). \tag{5.9}$$

Then, the algorithm for the identification of the time scale and slow variables in Section 3.3-3.4 approximate the constant $c = (1 + \epsilon^{0.502})^{-1}$. Therefore, the 1:1 relative phase θ

$$\theta = \arg(x_1 + iy_1) - \arg(x_2 + iy_2) \tag{5.10}$$

is evolving in the intermediate $\mathcal{O}(\epsilon^{0.502})$ time scale. The time evolution of I_1 and I_2 on the slow $\mathcal{O}(1)$ time scale is depicted in Figure 8. In addition, the figure shows the results of the 3-tier HMM integrator described in Section 4. Simulation parameters are detailed in Table 4.

Table 4: (Section 5.3) Parameters for the 3-tier HMM.

$\epsilon = 10^{-4}$	η_i	h_i	Averaging method	ODE scheme
0th tier	5	1/2	-	Midpoint
1st tier	$10\epsilon^q$	$\epsilon^q/5$	ALIF	Midpoint
2nd tier	35ϵ	$\epsilon/10$	ALIF	ODE45

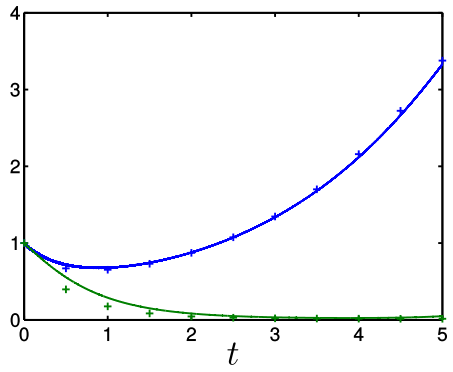


Figure 8: (Section 5.3) The dynamics of (5.8) on the ϵ^0 time scale. $\epsilon = 10^{-4}$. Plus signs are results of a 3-tier HMM.

5.4 The Fermi-Pasta-Ulam problem

The Fermi-Pasta-Ulam (FPU) problem is a challenging model in statistical mechanics which exhibits highly unexpected multiple scale (>2) behaviors [22]. We consider a long time simulation of $2k+1$ springs connected with alternating soft $k+1$ nonlinear and stiff k linear springs, and both ends are soft ones and fixed. The equation of motion is derived from the following Hamiltonian:

$$H(p,q) = \frac{1}{2}\sum_{i=1}^{2k} p_i^2 + \frac{1}{4}\epsilon^{-2}\sum_{i=1}^{k}(q_{2i} - q_{2i-1})^2 + \sum_{i=1}^{k}(q_{2i+1} - q_{2i})^4. \tag{5.11}$$

Using the change of variables given in [4], we have the following equations

$$\begin{cases} \frac{d}{dt}y_i = u_i, \\ \frac{d}{dt}x_i = \epsilon^{-1}v_i, \\ \frac{d}{dt}u_i = -(y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + (y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3, \\ \frac{d}{dt}v_i = -\epsilon^{-1}x_i + (y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + (y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3. \end{cases} \tag{5.12}$$

Both fixed ends yield $y_0 = x_0 = y_{k+1} = x_{k+1} = 0$. The system admits $2k-1$ slow variables evolving on the $\mathcal{O}(\epsilon^{-1})$ time scale. They are total energies of the stiff springs given by

$$I_i = x_i^2 + v_i^2, \quad i = 1, 2, \cdots, k \tag{5.13}$$

where $I_i : \mathbb{R}^{4k} \to \mathbb{R}$, and the relative phases between the different stiff springs, $\phi_i = x_1 x_i + v_1 v_i$, $i = 1, \cdots, k-1$. On the other hand, all the degrees of freedom which are related to the soft springs: y_i and u_i, $i = 1, \cdots, k$ evolve on the $\mathcal{O}(1)$ time scale. These variables thus lie on a $2k$ dimensional manifold for fixed slow variables.

Several numerical methods have been reported as being applied to the FPU problem. However, the existing results are mainly based upon the two-scale method [4, 39] and the corresponding computational cost is $\mathcal{O}(\epsilon^{-1})$. Therefore, it is meaningful to break through the need of $\mathcal{O}(\epsilon^{-1})$ steps in the two-scale algorithm and to devise a multiple-scale algorithm so that the longer time scale $\mathcal{O}(\epsilon^{-1})$ behavior can be approximated in a truly multiscale fashion. Unfortunately, the classical HMM using an 1D averaging kernel cannot be generally extended to the ϵ^{-1} time scale due to the dynamics embedded on the multi-dimensional torus. To this end, we use the ALIF to *resolve the oscillations in the intermediate time scale.* i.e., averaging over $2k$-torus will be achieved by the ALIF.

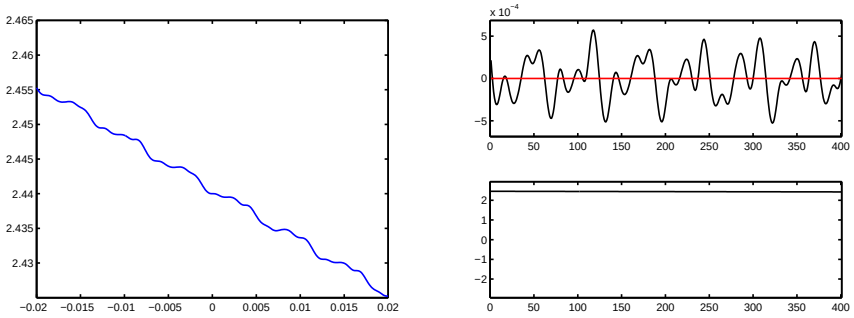
We rescale the time so that the slowest time of interest is independent of ϵ. Accordingly, the three time scales $\mathcal{O}(\epsilon^2)$, $\mathcal{O}(\epsilon)$ and $\mathcal{O}(1)$ will be considered. When we resolve $\mathcal{O}(\epsilon^2)$ time scale behavior in the 2nd tier, a Poincaré map technique is used because the dynamics in this time scale is equivalent to a rotation on a circle and thus more effective than the ALIF. A pair of the perturbed-unperturbed equations is picked as

Perturbed equation:

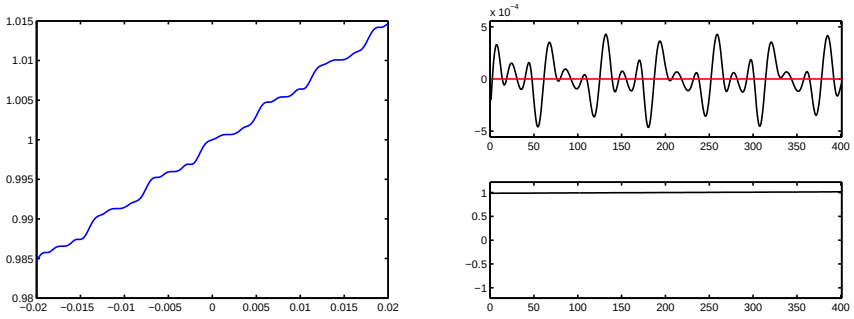
$$\begin{cases} \frac{d}{dt} y_i = \epsilon^{-1} u_i, \\ \frac{d}{dt} x_i = \epsilon^{-2} v_i, \\ \frac{d}{dt} u_i = -\epsilon^{-1}(y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + \epsilon^{-1}(y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3, \\ \frac{d}{dt} v_i = -\epsilon^{-2} x_i + \epsilon^{-1}(y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1})^3 + \epsilon^{-1}(y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i)^3, \end{cases} \tag{5.14}$$

Unperturbed equation:

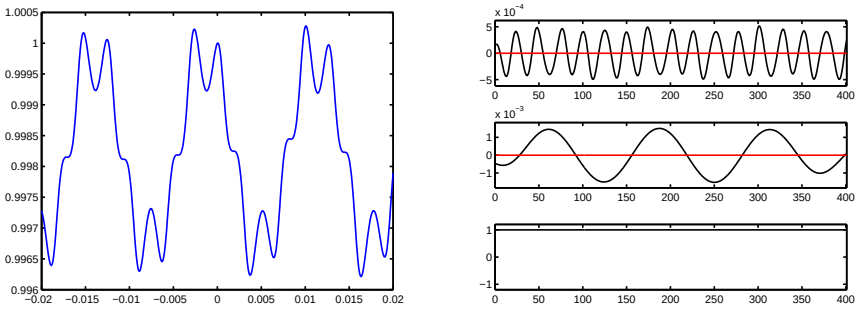
$$\begin{cases} \frac{d}{dt} y_i = 0, \\ \frac{d}{dt} x_i = \epsilon^{-2} v_i, \\ \frac{d}{dt} u_i = 0, \\ \frac{d}{dt} v_i = -\epsilon^{-2} x_i. \end{cases} \tag{5.15}$$



(a) Dynamics of $I_1(t)$ and the IMFs



(b) Dynamics of $I_2(t)$ and the IMFs



(c) Dynamics of $\phi(t)$ and the IMFs

Figure 9: IMFs of the slow variables on the interval $[-\eta_1, \eta_1]$. IMFs show the oscillations in the intermediate ϵ time scale, and the last component shows the trend in ϵ^0 time scale.

In the 1st tier, we resolve the $\mathcal{O}(\epsilon)$ time behavior using the ALIF. Figure 9 shows the decompositions of I_1, I_2 and ϕ in the 1st tier. Note that the fastest oscillations in $\mathcal{O}(\epsilon^2)$ scale are already resolved in the 2nd tier by a Poincaré map technique. Figure 10 illustrates the energy exchange of the stiff springs over $T = 10$ with $k = 2$ and $\epsilon = 5 \cdot 10^{-3}$. We compare the results computed by the ALIF-BF HMM Verlet-Verlet-ODE45 (Verlet in 0th & 1st tiers and ODE45 in 2nd tier as well as the averaging procedure in 1st tier using the ALIF and in 2nd tier using the quadratic interpolation in Poincaré) with those by an exponential integrator with Deuflhard's filter functions [22] with the stepsize $h = \epsilon/20000$, which we used as a reference solution. The initial conditions are $[y_1, x_1, u_1, v_1, y_2, x_2, u_2, v_2]^T = [1, 1, 0, 1.2, 1, 1, 1, 0]^T$. Our result is computed with the parameters given in Table 5, and a C^1 kernel with $p = 1$ is used for the filtered equation that corresponds to (5.14).

Table 5: (Section 5.4) ALIF-BF HMM parameters for Figure 10.

$\epsilon = 10^{-3}$	η_i	h_i	Averaging method	ODE scheme
0th tier	10	1/4	-	Verlet
1st tier	20ϵ	$\epsilon/10$	ALIF	Verlet
2nd tier	$15\epsilon^2$	$\epsilon^2/20$	Poincaré map	ODE45 (Reltol=10^{-7})

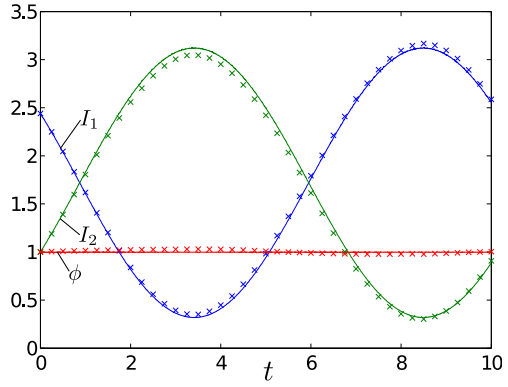


Figure 10: (Section 5.4) The solid lines correspond to the direct numerical simulation solution with an exponential integrator. Crosses correspond to the results of the proposed method.

6 Conclusion

In this paper, we proposed a new numerical method that combines the HMM and the ALIF on the purpose of computing slowly changing quantities. Compared to other multiscale schemes, this HMM and ALIF combination has several advantages compared to other multiscale schemes.

- The proposed algorithm offers a convenient method to calculate the effective rate of change for the slow dynamics as the result of the local solution decomposition. An effective ODE is obtained on-the-fly using the trend function.
- By analyzing the IMFs, the proposed algorithm can find hidden intermediate time scale and suggest a multiscale method which treats multiple (>2) time scale systems hierarchically.
- In some applications, the proposed algorithm allows us to identify hidden slow variables. This is especially useful for the systems where the slow variables are not explicitly known.

It is also important to study the errors of our proposed schemes which generally consist of

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

where E_H is the error of the macroscopic model (2.3), E_h is the errors from solving (2.1), E_{HMM} contains the errors in the the multiscale model, including the passing of information through $\mathcal{R}$ and $\mathcal{Q}$, and E_{ALIF} is the error from the decomposition using the ALIF. However, the accuracy of the decomposition using ALIF is not fully established yet. This is one of the future research directions we plan to take, so we postpone the rigorous error analysis for later.

References

[1] G. Ariel, B. Engquist, S. Kim, Y. Lee, and R. Tsai. A multiscale method for highly oscillatory dynamical systems using a Poincaré map type technique. *J. Sci. Comput.*, 54(2-3):247–268, 2013.

[2] G. Ariel, B. Engquist, S. J. Kim, and R. Tsai. Iterated averaging of three-scale oscillatory systems. *Commun. Math. Sci.*, 12(5):791–824, 2014.

[3] 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.

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

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

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

[7] G. Ariel, B. Engquist, and R. Tsai. Oscillatory systems with three separated time scales: Analysis and computation. In *Numerical Analysis of Multiscale Computations*, volume 82 of *Lecture Notes in Computational Science and Engineering*, pages 23–45. Springer Berlin Heidelberg, 2012.

[8] G. Ariel, S. J. Kim, and R. Tsai. Numerical averaging over embedded invariant tori under non-mixing dynamics. preprint.

[9] G. Ariel, S. J. Kim, and R. Tsai. Parareal methods for highly oscillatory ordinary differential equations. Submitted, `http://arxiv.org/abs/1503.02094`.

[10] V. I. Arnold, V. V. Kozlov, and A. I. Neishtadt. *Mathematical aspects of classical and celestial mechanics*, volume 3 of *Encyclopaedia of Mathematical Sciences*. Springer-Verlag, Berlin, third edition, 2006. [Dynamical systems. III], Translated from the Russian original by E. Khukhro.

[11] V.I. Arnol'd. *Mathematical methods of classical mechanics*. Springer-Verlag, New York, 1989.

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

[13] 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.

[14] A. Cicone, J. Liu, and H. Zhou. Adaptive local iterative filtering for signal decomposition and instantaneous frequency analysis.

[15] I. Daubechies, J. Lu, and H.-T. Wu. Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool. *Appl. Comput. Harmon. Anal.*, 30(2):243–261, 2011.

[16] P. Deuflhard. A study of extrapolation methods based on multistep schemes without parasitic solutions. *Z. Angew. Math. Phys.*, 30(2):177–189, 1979.

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

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

[19] 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.

[20] P. Flandrin, G. Rilling, and P. Goncalves. Empirical mode decomposition as a filter bank. *Signal Processing Letters, IEEE*, 11(2):112–114, Feb 2004.

[21] 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.

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

[23] T. Y. Hou and Z. Shi. Adaptive data analysis via sparse time-frequency representation. *Adv. Adapt. Data Anal.*, 3(1-2):1–28, 2011.

[24] T. Y. Hou and Z. Shi. Data-driven time-frequency analysis. *Appl. Comput. Harmon. Anal.*, 35(2):284–308, 2013.

[25] T. Y. Hou, Z. Shi, and P. Tavallali. Convergence of a data-driven time-frequency analysis method. *Appl. Comput. Harmon. Anal.*, 37(2):235–270, 2014.

[26] T. Y. Hou, M. P. Yan, and Z. Wu. A variant of the EMD method for multi-scale data. *Adv. Adapt. Data Anal.*, 1(4):483–516, 2009.

[27] N. E. Huang, Z. Shen, and S. R. Long. A new view of nonlinear water waves: the Hilbert spectrum. In *Annual review of fluid mechanics, Vol. 31*, volume 31 of *Annu. Rev. Fluid Mech.*, pages 417–457. Annual Reviews, Palo Alto, CA, 1999.

[28] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, 1998.

[29] N. E. Huang and Z. Wu. A review on hilbert-huang transform: Method and its applications to geophysical studies. *Reviews of Geophysics*, 46(2), 2008.

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

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

[32] S. J. Kim. *Numerical methods for highly oscillatory dynamical systems using multiscale structure*. PhD thesis, University of Texas at Austin, 2013.

[33] P. Lochak and C. Meunier. *Multiphase averaging for classical systems*, volume 72 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1988. With applications to adiabatic theorems, Translated from the French by H. S. Dumas.

[34] J. A. Murdock. *Perturbations*, volume 27 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999. Theory and methods, Corrected reprint of the 1991 original.

[35] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[36] R. L. Petzold, O. J. Laurent, and Y. Jeng. Numerical solution of highly oscillatory ordinary differential equations. *Acta Numerica*, 6:437–483, 1997.

[37] 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.

[38] R. Sharp, Y.-H. Tsai, and B. Engquist. Multiple time scale numerical methods for the inverted pendulum problem. In *Multiscale methods in science and engineering*, volume 44 of *Lect. Notes Comput. Sci. Eng.*, pages 241–261. Springer, Berlin, 2005.

[39] M. Tao, H. Owhadi, and J. E. 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.

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

[41] Z. Wu and N. E. Huang. A study of the characteristics of white noise using the empirical mode decomposition method. *Proceedings: Mathematical, Physical and Engineering Sciences*, 460(2046):pp. 1597–1611, 2004.

[42] Z. Wu and N. E. Huang. Statistical significance test of intrinsic mode functions. In *Hilbert-Huang transform and its applications*, volume 5 of *Interdiscip. Math. Sci.*, pages 107–127. World Sci. Publ., Hackensack, NJ, 2005.

[43] Z. Wu and N. E. Huang. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Advances in Adaptive Data Analysis*, 01(01):1–41, 2009.

[44] Z. Wu, N. E. Huang, S. R. Long, and C.-K. Peng. On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proceedings of the National Academy of Sciences*, 104(38):14889–14894, 2007.