ABSTRACT. A multiscale, time reversible method for computing the effective slow behavior of systems of highly oscillatory ordinary differential equations is presented. The proposed method relies on correctly tracking a set of slow variables that is sufficient to approximate any variable and functional that are slow under the dynamics of the system. The algorithm follows the framework of the heterogeneous multiscale method. Numerical examples show the efficiency of the multiscale method and the advantages of time reversibility.

1. INTRODUCTION

Time reversible and symplectic schemes have been proven to be highly valuable tools for integrating systems of ordinary differential equations (ODEs) [17] whose solutions possess similar properties. These methods are particularly useful for integration over long time segments. However, for general systems involving two or more time scales, most schemes require a step size that is of the order of the fastest scale, typically due to stability and accuracy considerations.

The challenges of multiscale numerical integration have been addressed by several different approaches. Stiff problems with fast transients can be optimally solved by implicit schemes [7, 18, 19]. The Chebyshev methods [1, 21] as well as the projective integrator approach [15] provide stable and explicit computational strategies for this class of problems in general. For harmonic oscillatory problems, traditional numerical approaches attempt to either filter out or fit fast oscillations to some known functions in order to reduce the complexity, e.g. [14, 20, 27], or use some notion of Poincaré map to determine slow changes in the orbital structure [16, 24]. A general class of approaches aiming at Hamiltonian systems are geometric integration schemes that preserve a discrete version of certain invariances. We refer the readers to [17] and [22] for an extensive list of literature. Many of the schemes specialized for finite dimensional mechanical systems can be conveniently derived from the viewpoint of variational integrators [23]. In certain applications, special considerations are given to the expensive cost of evaluating non-local potentials in large systems, see e.g. the impulse method and its derivatives [22]. For a recent review on numerical methods for highly oscillatory systems see [6].

In this paper we propose a numerical scheme for approximating the slow dynamics of highly oscillatory ODE systems. The scheme follows the framework of the Heterogeneous Multiscale Methods (HMM) [8, 9, 10, 12, 29]. It is multiscale in
the sense that, in order to gain efficiency, it does not fully resolve the fastest scale. The scheme is time reversible and enjoys some of the benefits of geometric integrators such as low energy dissipation. Recently, Calvo and Sanz-Serna suggested an HMM scheme that is both time reversible and symplectic [5]. Their method can be applied to some types of Hamiltonian systems, for example, mechanical systems that are driven by a single external fast oscillation.

We consider ODE systems of the form

\[ (1.1) \quad \dot{x} = \epsilon^{-1} f(x) + g(x), \quad x(0) = x_0, \quad t \in [0, T], \]

where \( 0 < \epsilon \leq \epsilon_0 \), \( x = (x_1, \ldots, x_d) \in \mathbb{R}^d \) and \( 0 < T < \infty \) is independent of \( \epsilon \). It is assumed that the solution of (1.1) remains in a domain \( D_0 \subset \mathbb{R}^d \) which is bounded independent of \( \epsilon \) for all \( t \in [0, T] \). For fixed \( \epsilon \) and initial condition \( x_0 \), the solution of (1.1) is denoted \( x(t; \epsilon, x_0) \). For brevity we will write \( x(t) \) when the dependence on \( \epsilon \) and \( x_0 \) is not directly relevant to the discussion. Furthermore, in this paper we only consider the case in which the fast dynamics is oscillatory in nature rather than dissipative. In particular, we assume that the unperturbed system, \( g(x) = 0 \), has a continuous family of periodic solutions or an asymptotically stable periodic limit cycle.

The general approach previously developed in [2], [3] and [11], is to identify a set of functions in the state space whose values change slowly along the oscillatory trajectories. The time evolution of these slow variables is used to guide the slow time-scale dynamics. The ODE (1.1) is then integrated following the HMM framework: a Macro-solver integrates the effective, but generally unknown evolution equation for the slow variables under the dynamics of (1.1). The rates of change for these slow variables are computed on-the-fly by a micro-solver that integrates the full ODE (1.1) for short time segments. For a recent review see [10].

To keep this paper self-contained, Section 2 reviews the main results and algorithms developed in [2], [3] and [11]. Section 3 describes a particular implementation of the method which is time reversible. A few examples are presented in Section 4. We end with concluding remarks in Section 5.

2. THE HMM SCHEME

In order to study the long time properties of (1.1) we need to distinguish between the fast and slow constituents of the dynamics. We say that a real valued smooth function (variable) \( \alpha(x) \) is slow with respect to (1.1) if there exists a non empty open set \( A \subset \mathbb{R}^d \) such that

\[ \max_{x_0 \in A, t \in [0, T]} \left| \frac{d}{dt} \alpha(x(t; \epsilon, x_0)) \right| \leq C_0, \]

where \( C_0 \) is a constant that is independent of \( \epsilon \). Otherwise, \( \alpha(x) \) is said to be fast. Similarly, we say that a quantity or constant is of order one if it is bounded independent of \( \epsilon \) in \( D_0 \) or \([0, T]\).
Of course, any function of slow variables is also slow. Therefore, it is reasonable to look for variables which are functionally independent, i.e., a vector of slow variables \( \xi = (\xi^{(1)}(x), \ldots, \xi^{(r)}(x)) \) such that \( \nabla \xi^{(1)}(x), \ldots, \nabla \xi^{(r)}(x) \) are linearly independent in \( \mathcal{A} \). Since \( r \) is bounded by the dimension, \( d \), it is useful to look at a set with a maximal number of functionally independent slow variables. Augmenting the slow variables with \( d - r \) fast ones \( z = (z_1, \ldots, z_{d-r}) \) such that \( \partial(\xi, z)/\partial x \) is non-singular in \( \mathcal{A} \), one obtains a local coordinate system, i.e., a chart of the states space. We will refer to a chart in which a maximal number of coordinates is slow as a maximal slow chart for \( \mathcal{A} \) with respect to the ODE (1.1). Covering the set \( \mathcal{D}_0 \) by maximal slow charts we obtain a maximal slow atlas for \( \mathcal{D}_0 \).

Next, we wish to establish the existence of an effective evolution equation for the slow variables \( \xi(x(t)) \) under the flow of (1.1). The assumption that the unperturbed dynamics is periodic implies that the only fast coordinate is equivalent to rotation on the unit circle with constant velocity, i.e., \( \phi \in S^1 \). This case is quite general since many weakly perturbed integrable systems in resonance fall into this category through the notion of action-angle variables. Then, an averaging principle can be used to prove that for small \( \epsilon \), \( \xi(x(t; \epsilon, x_0)) \) is well approximated in \( [0, T] \) by an effective equation of the form

\[
\dot{\xi} = F(\xi), \quad \xi(0) = \xi(x_0).
\]

See [2, 4, 25] for details. The requirement that \( (\xi, \phi) \) is a maximal slow chart is critical for the derivation of (2.3). Without it, there is no guaranty that the right hand side of the averaged equation does not depend on additional slow variables which may be hidden or unknown.

The effective equation (2.3) may not be available as an explicit formula. Instead, the idea behind the HMM algorithm is to evaluate \( F(\xi) \) by numerical solutions of the original ODE (1.1) on significantly reduced time intervals. In this way, the HMM algorithm approximates an assumed effective equation whose form is typically unknown. This strategy is advantageous if \( F(\xi) \) can be approximated efficiently. The next section describes such an algorithm.

2.1. The algorithm. Suppose \( \xi = (\xi^{(1)}(x), \ldots, \xi^{(r)}(x)) \) are the slow variables in a slow atlas for (1.1). The system is integrated using a two level algorithm, each level corresponding to a different time scale. The first is a Macro-solver which integrates the effective equation (2.3) for the slow variables \( \xi \). The second level is a micro-solver that is invoked whenever the Macro-solver needs an estimate of \( F(\xi) \). The micro-solver computes a short time solution of (1.1) using suitable initial data. Then, the time derivative of \( \xi \) is approximated by

\[
\dot{\xi}(t) \sim \langle \dot{\xi} \rangle_{\eta}(t) = \int_{-\eta/2}^{\eta/2} \dot{\xi}(t + \tau)K_{\eta}(t - \tau)d\tau;
\]
where, $K_\eta(\cdot)$ denotes a smooth averaging kernel with support on $[-\eta/2, \eta/2]$. Note that $\xi$ is not necessarily slow. However, it is bounded independent of $\epsilon$. The properties of averaging with respect to a kernel will be reviewed shortly.

To better explain the algorithm, denote the Macro-solver sample times by $t_0, \ldots, t_N$, $N = T/H$, and its output at corresponding times by $x_0, \ldots, x_N$. At the $n$-th Macro-step, the micro-solver can be implemented using any scheme with step-size $h$ and initial condition $x(t_n) = x_n$. It integrates the full ODE both backwards and forward in time to approximate the solution in $[t_n - \eta/2, t_n + \eta/2]$. The structure of the algorithm, depicted in Figure 1, is as follows.

1. Initial conditions: $x(0) = x_0$, $\xi_0 = \xi(x_0)$ and $n = 0$.

2. Force estimation:
   (a) micro-simulation: solve (1.1) in $[t_n - \eta/2, t_n + \eta/2]$ with initial conditions $x(t_n) = x_n$.
   (b) Averaging: approximate $\dot{\xi}(t_n)$ by $\langle \dot{\xi}(t_n) \rangle$. 

3. Macro-step (forward-Euler example): $\xi_{n+1} = \xi_n + H \langle \dot{\xi}(t_n) \rangle$.

4. Reconstruction: find $x_{n+1}$ consistent with $\xi_{n+1}$. take $x_{n+1} = x_n + H \tilde{F}_n$, where $\tilde{F}_n$ is the least squares solution of the linear system

\[
\frac{\partial \xi(x_n)}{\partial x} \tilde{F}_n = \langle \dot{\xi}(t_n) \rangle
\]

5. $n = n + 1$. Repeat steps (2) and (3) to time $T$.

Here, $\partial \xi/\partial x$ is a matrix whose $k$’th row is $\nabla \xi^{(k)}$. The scheme described above can be generalized to Macro-solvers with higher order accuracy.

Figure 1. The cartoon depicts the time steps taken by the HMM scheme. At the $n$-th Macro step, a micro-solver with step size $h$ integrates (1.1) to approximate $x(t)$ in a time segment $[t_n - \eta/2, t_n + \eta/2]$. This data is used to calculate $\langle \dot{\xi}(x) \rangle_{\eta}(t)$. Then, the Macro-solver takes a big step of size $H \tilde{F}_n$, where $\tilde{F}_n$ is consistent with $\langle \xi^{(k)} \rangle_{\eta}$ for all slow variables $\xi^{(k)}$ in the maximal slow chart.

2.2. **Updating state variables.** At the core of the HMM framework lies the idea that we are actually solving the effective equation $\dot{\xi} = F(\xi)$ at the macroscopic
time scale. Accordingly, one has the freedom of using his integrator of choice. For example, applying forward-Euler with step size $H$ yields a single-step rule
\begin{equation}
(2.5) \quad \xi_{n+1} = \xi_n + HF(\xi_n),
\end{equation}
where $\xi_n$ denotes the approximation for $\xi(t_n)$. Alternatively, one can use the familiar two-step leap-frog method, which is reversible in time
\begin{equation}
(2.6) \quad \xi_{n+1} = \xi_{n-1} + 2HF(\xi_n).
\end{equation}
As in the algorithm described above, $F(\xi_n)$ is approximated by solving the full system for a very short time window.

Once new values for the slow variables are determined, it is necessary to find a new set of state variables, $x_{n+1}$, that are consistent with the new slow state, $\xi_{n+1}$, i.e., $\xi(x_{n+1}) = \xi_{n+1}$. This stage is referred to as reconstruction. Instead, the algorithm described above bypasses this difficulty by evolving the state variables $x$ directly [2, 12]. Hence, step (3) of the algorithm above does not need to be performed. Of course, this has to be done in a way that is consistent with the slow dynamics (to some order in $H$). With single-step methods, the Macro-step takes the form
\begin{equation}
(2.7) \quad x_{n+1} = x_n + H\tilde{F}(x_n, \xi_n).
\end{equation}
Comparing with (2.5) we find that, to order $H^2$, $\tilde{F}(x_n, \xi_n)$ can be taken to be the least squares solution of the linear system
\begin{equation*}
\frac{\partial \xi(x_n)}{\partial x} \tilde{F}(x_n, \xi_n) = F(\xi_n).
\end{equation*}
Higher order schemes following Runke-Kutta methods are developed in [2].

The main goal of this paper is to develop a time reversible leap-frog scheme for the state variables which consistently embeds the leap-frog scheme for $\xi$, (2.6). The scheme can be written implicitly in the form
\begin{equation}
(2.8) \quad G(x_n, \xi_n; x_{n-1}, x_{n+1}) = 0,
\end{equation}
where we require that $x_{n-1}$ and $x_{n+1}$ are consistent with (2.6) to some known power of $H$. In Section 3, we suggest a possible form for $G$, which is skew-symmetric, i.e.,
\begin{equation*}
G(x_n, \xi_n; x_{n-1}, x_{n+1}) = -G(x_n, \xi_n; x_{n+1}, x_{n-1}).
\end{equation*}
As a result, the evolution operator is time reversible. The process of finding a microscopic state $x_{n+1}$, consistent with the Macroscopic $\xi_{n+1}$ is under-determined since different $x$ may correspond to the same slow coordinates $\xi$. Using (2.8) one picks a particular possible solution in such a way that the entire algorithm become reversible in time.

Stability of the new scheme is inherited from that of the leap-from method for $\xi$, (2.6). Note that the approximation is only consistent for the slow variables $\xi$ and not for the original state variables $x$, since any information on the fast coordinate is discarded.
2.3. **Averaging using kernels.** Let \( K(\cdot) \) denote a smooth kernel function with support on \([-1, 1]\) with unit mass, \( \int_{-1}^{1} K(\tau) d\tau = 1 \), and vanishing first moment, \( \int_{-1}^{1} K(\tau) \tau d\tau = 0 \). In this paper we restrict the discussion to kernels which are symmetric with respect to their mid-point. For example, the following smooth exponential kernel was found useful:

\[
K(t) = Z^{-1} \exp \left( -\frac{5}{4} \frac{1}{(t-1)(t+1)} \right),
\]

for \( t \in (-1, 1) \) and zero otherwise. Here, \( Z \) is a normalization constant. For \( \eta > 0 \) let,

\[
K_\eta(\tau) = \frac{2}{\eta} K\left( \frac{2}{\eta} \tau \right).
\]

We will take \( \eta \) to be \( \epsilon \) dependent such that \( 0 < \epsilon \ll \eta \ll 1 \). The convolution of a function \( a(t) \) with \( K_\eta \) is denoted as (recall (2.4))

\[
\langle a \rangle_\eta(t) = \int_{-\eta/2}^{\eta/2} a(t + \tau) K_\eta(t - \tau) d\tau.
\]

Typically, the fast dynamics in equations such as (1.1) is one of two types (compare to the linear case, \( f(x) = Ax \)). The first consists of modes that are attracted to a low dimensional manifold in a time scale of order one. These modes are referred to as transient or dissipative modes and will not be discussed in this paper. The second type consists of oscillators with constant or slowly changing frequencies. Averaging of oscillatory modes filters out high frequency oscillations. The errors introduced by the averaging are estimated in [2] and [11]. For example, for a function \( \beta(t) \) with period one and a kernel with \( q \) continuous derivatives, we have that

\[
\| K_\eta(\cdot) * \beta(\epsilon^{-1} \cdot) - \bar{\beta} \| \leq C \| \beta \|_\infty \| K \|_{W^{1,q}} \left( \frac{\epsilon}{\eta} \right)^q,
\]

where \( \bar{\beta} = \int_{0}^{1} \beta(\tau) d\tau \), \( \| \cdot \|_\infty \) denotes the sup norm in \( D_0 \),

\[
\| \beta \|_\infty = \sup_{x \in D_0} |\beta(x)|,
\]

and

\[
\| K \|_{W^{1,q}} = \int_{-1}^{1} K^{(q)}(t) dt.
\]

Here, \( K^{(q)} \) denotes the \( q \)-th derivative of \( K \).

3. **A REVERSIBLE Macro-solver**

In this Section we describe a time reversible Macroscopic solver. As a consequence, the entire HMM algorithm described in Section 2.1 is also time reversible. Let \( x(t) \) denote the exact solution of the full ODE system (1.1) with the initial condition
x(0) = x_0. In addition, let \( H \) denote the Macroscopic step size used in the Macro-solver, \( x_n = x(nH) \) and \( \Delta^+ x_{n-1} = x_n - x_{n-1} \).

For illustration purposes, consider the following linear system describing a slowly expanding spiral

\[
\begin{align*}
\dot{z}_1 &= -\epsilon^{-1}z_2 + z_1 \\
\dot{z}_2 &= \epsilon^{-1}z_1 + z_2,
\end{align*}
\]

with initial conditions \( z_1(0) = 1 \) and \( z_2(0) = 0 \). The exact solution of (3.1) is \( x(t) = (z_1(t), z_2(t)) = (e^t \cos \epsilon^{-1}t, e^t \sin \epsilon^{-1}t) \) and \( \xi(x) = z_1^2 + z_2^2 \) is a slow variable. Figure 2 depicts a forward-Euler type Macro-step for (3.1).

Let \( \gamma \) denote a smooth curve connecting \( x_0 \) and \( x_1 = x(H) \). The change in \( \xi(x) \) along the curve can be expressed as

\[
\Delta^+ \xi(x_0) = \xi(x_1) - \xi(x_0) = \int_\gamma \nabla \xi(x) \cdot dx.
\]

We are interested in two particular curves. The first is along the solution of the ODE. Taking \( \gamma = \{ x(t)|0 \leq t \leq H \} \), (3.2) yields

\[
\Delta^+ \xi(x_0) = \int_0^H \nabla \xi(x(\tau)) \cdot \dot{x} d\tau = \int_0^H \dot{\xi}(x(\tau)) d\tau.
\]

For \( 0 < \epsilon < \eta \ll H \), we obtain an estimate

\[
\Delta^+ \xi(x_0) = H \left\langle \dot{\xi}(x) \right\rangle_\eta (0) + O(H^2 + \epsilon),
\]

where \( e \) denotes the error from approximating the average \( \dot{\xi} \) using kernels. We assume \( e \) is negligible compared to \( H^2 \).

An alternative choice of curve follows the straight line in phase space connecting \( x_0 \) and \( x_1 \). Parameterizing the segment as \( s x_1 + (1 - s) x_0 \) with \( s \in [0, 1] \) yields

\[
\Delta^+ \xi(x_0) = \int_0^1 \nabla \xi(x_0 + s \Delta^+ x_0) \cdot (x_1 - x_0) ds = \nabla \xi(x_0) \cdot \Delta^+ x_0 + O(H^2).
\]

Comparing (3.4) and (3.5) we deduce that to second order in \( H \),

\[
H \left\langle \dot{\xi}(x) \right\rangle_\eta (0) = \nabla \xi(x_0) \cdot \Delta^+ x_0.
\]

Thus, solving for \( \Delta^+ x_0 \), (3.6) yields the forward-Euler Macro-step formula used for the algorithm described in Section 2.1 with a local truncation error that is second order in \( H \). For larger systems with several slow variables (3.6) generalizes to a linear system whose components comes directly from each of the slow variables, and (3.6) can be solved using singular value decomposition.

The derivation above can be generalized to high order Runge-Kutta type methods by improving the approximation in (3.2) and (3.5). This approach is developed in [2]. More attention is needed for multistep methods that uses directly the original variables of the full system at the Macroscopic level. In the following, we discuss our approach for designing such schemes.
Suppose the multiscale algorithm has already produced the first two Macroscopic steps, $x_0 = x(0)$ and $x_1 = x(H)$. We are looking for the next Macro-step $x_2 = x(2H)$. In analogy to the leap frog method, we would like to find the value for $x_2$ using $x_0$, $x_1$ and the derivative at the middle point $x_1$, $\langle \dot{\xi} \rangle_\eta(H)$. In particular, any reversible scheme cannot use the derivatives at $x_0$, $\langle \dot{\xi} \rangle_\eta(0)$. Figure 3 depicts a reversible Macroscopic two-step solver.

Following the discussion above, we consider the change in a slow variable $\xi$ between $x_0$ to $x_2$

$$\Delta^0 \xi = \xi(x_2) - \xi(x_0) = \int_0^\gamma \nabla \xi(x) \cdot dx,$$

where $\gamma$ is a smooth curve starting at $x_0$ and ending at $x_2$. Integrating along the solution of the ODE, $x(t)$, yields

$$\Delta^0 \xi = \int_0^{2H} \dot{\xi}(x(\tau)) d\tau = \int_0^{2H} \langle \dot{\xi}(x) \rangle_\eta(\tau) d\tau + O(He) = 2H \langle \dot{\xi}(x) \rangle_\eta(H) + O(H^3).$$

On the other hand, we expand $\xi(x)$ around the middle point $x_1$

$$\xi(x_1 + \delta x) = \xi(x_1) + \mathbf{v} \cdot \delta x + O(\delta^2 x),$$

where $\mathbf{v} = \nabla \xi(x_1)$. Hence,

$$\Delta^0 \xi = \mathbf{v} \cdot (x_2 - x_0) + O(\delta^2 x).$$

Assuming that $\xi(t)$ is uniformly Lipshitz in the domain of interest, $D_0$, we have that $\delta x = O(H)$. Therefore, comparing (3.8) and (3.10) yields, to order $H^2$,

$$2H \langle \dot{\xi}(x) \rangle_\eta(H) = \mathbf{v} \cdot (x_2 - x_0).$$

In order to achieve a local truncation error as small as possible, we look for the solution $x_2$ such that $|x_2 - x_1|$ is minimal. The reversibility of the algorithm hinges on the fact that (3.11) is anti-symmetric with respect to switching $x_0$ and $x_2$ and that both $|x_2 - x_1|$ and $|x_0 - x_1|$ are minimal. Integrating forward, $x_0$ and $x_1$ are given. Using the micro-solver we approximate the force at $x_1$, $\langle \dot{\xi}(x) \rangle_\eta(H)$, and solve (3.11) for $x_2$. Integrating backwards, $x_2$ and $x_1$ are given. Using the micro-solver we approximate the force starting at the same point, $x_1$, and solve (3.11) for $x_0$. Since (3.11) is a linear system, $x_2$ can be easily obtained using singular value decomposition.

Using (3.11), the error in each Macro step is of order $H^2$. Hence, the global error of the Macro-solver is of order $H$. In order to obtain a second order method, we expand the slow variable $\xi(x_1 + \delta x)$ to second order in $\delta x$

$$\xi(x_1 + \delta x) = \xi(x_1) + \mathbf{v} \cdot \delta x + \frac{1}{2} \delta^2 x \cdot A \delta x + O(\delta^3 x),$$

where $A$ is a symmetric matrix that both $\dot{\xi}(x_1 + \delta x)$ is anti-symmetric with respect to switching $x_0$ and $x_2$ and that both $|x_2 - x_1|$ and $|x_0 - x_1|$ are minimal.
FIGURE 2. A forward-Euler type Macro-step for the expanding spiral (3.1).

FIGURE 3. A leap-frog type Macro-step for the expanding spiral (3.1).

where $A$ is the Hessian of $\xi(x)$ evaluated at $x_1$. Using (3.7), (3.8) and (3.12) yields an equation for $x_2$ that is accurate to order $H^3$

\begin{equation}
2H(\langle \xi(x) \rangle_\eta H) = (x_2 - x_0) \cdot (v - Ax_1) + \frac{1}{2}x_2 \cdot Ax_2 - \frac{1}{2}x_0 \cdot Ax_0.
\end{equation}

As before, in order to have a reversible scheme, we look for a solution that is closest to the middle point $x_1$. With several slow variables $\xi^{(1)}, \ldots, \xi^{(r)}$, $x_2$ is the minimum of $|x_2 - x_1|^2$ under the constraint (3.13) for each of the slow variables. Since (3.13) is quadratic in $x_2$ and $x_0$, it can be shown that for small enough values of $H$ the sought constrained minimum exists and is unique. Hence, the scheme is reversible.
Using Lagrange multipliers, $x_2$ satisfies

$$\begin{align*}
(x_2 - x_0) \cdot (v - A x_1) + \frac{1}{2} x_2 \cdot A x_2 - \frac{1}{2} x_0 \cdot A x_0 &= \Delta^0 \xi^{(k)}, \quad k = 1 \ldots r \\
2x_2 - 2x_1 + \sum_{k=1}^{r} \lambda_k [v + A(x_2 - x_1)] &= 0,
\end{align*}$$

(3.14)

where $\Delta^0 \xi^{(k)} = 2H(\dot{\xi}^{(k)}(x))_\eta(H)$ and $\lambda_1 \ldots \lambda_r$ are the Lagrange multipliers. For systems of coupled oscillators, the slow variables correspond to amplitudes and the relative phase between the oscillators. In [2, 3], we show that for periodic systems there exists $r = d - 1$ functionally independent slow variables. Hence, (3.14) is a quadratic system of $2d - 1$ equations and $2d - 1$ unknowns. In the examples appearing at the next Section, (3.14) is solved using Newton-Raphson with a required accuracy $\epsilon$. Taking the second order approximation (3.11) and $\lambda_1 = \cdots = \lambda_r = 0$ is a good initial guess. Since the system is quadratic, convergence is usually rapid.

We summarize our method with the following algorithm. Notations are as in Section 2.

1. Initial conditions: $x(0) = x_0$ and $n = 0$.
2. Force estimation:
   a. micro-simulation: solve (1.1) in $[t_n - \eta/2, t_n + \eta/2]$ with initial conditions $x(t_n) = x_n$.
   b. Averaging: approximate $\dot{x}(t_n)$ by $\langle \dot{x}\rangle_\eta(t_n)$.
3. Macro-step: evolving $\xi$ and reconstructing $x_{n+1}$ (leap frog example)
   solve $G(x_n, 2H(\dot{\xi}(t_n))_\eta; x_{n-1}, x_{n+1}) = 0$ for $x_{n+1}$, where $G$ is given by (3.14).
4. $n = n + 1$. Repeat steps (2) and (3) to time $T$.

Finally, we remark that higher order quadrature methods can be constructed in a similar fashion using two or more steps for approximating (3.8) and additional terms in the Taylor expansion (3.12).

4. Examples

In this Section we apply the reversible HMM algorithm described above to several model systems.

4.1. The inverted pendulum. The following example considers a pendulum with a rigid arm that is attached at one of its ends to a mechanical motor. The setup is depicted in Figure 4. The motor causes the point of suspension of the arm to vibrate up and down with amplitude $\epsilon$ and frequency $\epsilon^{-1}$. Surprisingly, the fast vibrations of the motor can cause the pendulum to oscillate slowly (with a $O(1)$ frequency) around the inverted position, in which its arm is pointing up. Denoting by $\theta$ the
angle between the pendulum arm and the upward direction, the equation of motion for the system becomes

\[ l \ddot{\theta} = \left[ g + \epsilon^{-1} \sin(2\pi \epsilon^{-1} t) \right] \sin \theta, \]

where \( \theta \) denotes the angle between the arm and the upward direction, \( l \) is the arm’s length and \( g \) is the gravitational constant. Rewriting (4.1) as a first order autonomous system yields an ODE of the form

\[
\begin{aligned}
\dot{\theta}_1 &= \theta_2 \\
\dot{\theta}_2 &= l^{-1}(g + \epsilon^{-1} \psi_1) \sin \theta_1 \\
\dot{\psi}_1 &= 2\pi \epsilon^{-1} \psi_2 \\
\dot{\psi}_2 &= -2\pi \epsilon^{-1} \psi_1
\end{aligned}
\]

In [2] we describe a variational numerical method for identifying the slow variable for (4.2). The method identifies three slow variables that constitute a slow atlas:

\[
\begin{aligned}
\xi^{(1)} &= \theta_1 \\
\xi^{(2)} &= \psi_1^2 + \psi_2^2 \\
\xi^{(3)} &= \theta_2 + (2\pi l)^{-1} \psi_2 \sin \theta_1.
\end{aligned}
\]

Indeed, it is easily verified that \( (d/dt)\xi^{(k)}(x(t)) \) is bounded independent of \( \epsilon \) for \( k = 1, 2, 3 \). Figure 5b depicts the numerical HMM solution for (4.2) with \( g = 0.1 \) and \( l = 0.05 \) using the reversible, second order Macro-solver (3.13). Simulation parameters are \( \epsilon = 10^{-5}, h = \epsilon/25, H = 0.25, \eta = 6.2 \epsilon \) and the exponential kernel (2.9). The Newton-Raphson algorithm for solving (3.14) never requires more than two iterations. Initial conditions are \( \theta_1(0) = 0, \theta_2(0) = -0.4, \psi_1(0) = 0 \) and \( \psi_2(0) = 1 \). The value for \( \xi^{(2)} \) is practically constant with an error that is less than \( 10^{-8} \). Invariance of quadratic constants of motion is a typical advantage of reversible methods. With the above parameters the HMM algorithm runs over 5000 times faster than Verlet. Figure 5b depicts a similar numerical HMM solution for (4.2) using the exact same parameters, but with a Macro-solver applying the midpoint rule. All other parameters are the same. Additional approaches for applying the HMM strategy on this example can be found at [5, 26, 28].

4.2. Fermi-Pasta-Ulam. The Fermi-Pasta-Ulam model [13] is a one dimensional system of unit mass particles connected by springs. The springs alternate between stiff linear and soft non-linear ones. The model is derived from the following Hamiltonian

\[
H = \frac{1}{2} \sum_{i=1}^{2k} p_i^2 + \frac{1}{4} \epsilon^{-2} \sum_{i=1}^{k} (q_{2i} - q_{2i-1})^2 + \sum_{i=0}^{k} (q_{2i+1} - q_{2i})^4.
\]

The following linear change of variables is convenient since it separates the elongations of the \( k \) stiff springs and associated momenta:

\[
\begin{aligned}
x_i &= \epsilon^{-1} (q_{2i-1} - q_{2i})/\sqrt{2}, \\
v_i &= (p_{2i-1} - p_{2i})/\sqrt{2},
\end{aligned}
\]
Figure 4. The inverted pendulum has a rigid arm which is attached to a motor that is vibrating fast. The centrifugal force pulls the arm upwards.

Figure 5. (a) Comparison of the HMM approximation for the solution of (4.2) to the Verlet method with step size of order $\epsilon$. Solid curve: $\xi^{(1)} = \theta_1$, dotted curve: $\xi^{(3)} = \theta_2 + (2\pi l)^{-1}\psi \sin\theta_1$. $\xi^{(2)}$ is constant. (b) The same system with a Macro-solver applying the mid-point rule. All other parameters are the same.

and a second set of variables associated with the $k$ soft springs:

\begin{align}
    y_i &= \left( q_{2i-1} + q_{2i} \right)/\sqrt{2}, \quad u_i = \left( p_{2i-1} + p_{2i} \right)/\sqrt{2},
\end{align}

Defining $y_0 = x_0 = y_{2k+1} = x_{2k+1} = 0$, the equations of motion can be written as (4.7)

\begin{align}
    \dot{y}_i &= u_i \\
    \dot{x}_i &= \epsilon^{-1} u_i \\
    \dot{u}_i &= -\left( y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1} \right)^3 + \left( y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i \right)^3 \\
    \dot{v}_i &= -\epsilon^{-1} x_i + \left( y_i - \epsilon x_i - y_{i-1} - \epsilon x_{i-1} \right)^3 + \left( y_{i+1} - \epsilon x_{i+1} - y_i - \epsilon x_i \right)^3.
\end{align}
As discussed in [2], the slow atlas for the system consists of $4k - 1$ slow variables. First are all the degrees of freedom which are related to the soft springs: $y_i$ and $u_i$, $i = 1 \ldots k$. Second, the total energy (kinetic + potential) of the stiff springs, $I_i = x_i^2 + v_i^2$, $i = 1 \ldots k$. Finally, the relative phases between the different stiff springs, $\phi_i = x_1 x_i + v_1 v_i$, $i = 2 \ldots k$. Any other function $\alpha(x)$ which is slow under the dynamics of (4.7) can be written as a function of the $4k - 1$ variables described above.

On the $O(1)$ time scale the energy of the stiff springs and their relative phases are fixed, while the degrees of freedom that correspond to the soft springs oscillate in a complicated, non-harmonic way. On the $O(\epsilon^{-1})$ time scale the dynamics becomes more interesting as the energies $I_i$ begin to change [13, 17]. Figure 6a depicts our results for a system with three stiff springs, $k = 3$. Initial conditions are $x_1 = -1$, $y_1 = -0.5$, $y_1 = u_1 = x_2 = 1$, $v_3 = -0.5$ and zero otherwise. Simulation parameters are $\epsilon = 10^{-3}$, $h = \epsilon/50$, and $H = 0.1$, $\eta = 60\epsilon$ and the exponential kernel (2.9). The second order accurate reversible HMM (3.14) is compared to the Verlet solution with step size $\epsilon/200$. Note the significantly smaller step size required for Verlet. Figure 6b depicts the drift in the total energy of the entire system, which is smaller than 0.4%. Our method does not guarantee convergence on the $\epsilon^{-1}$ time scale. Accordingly, the purpose of this example is to demonstrate the benefits of the symmetric algorithm. Indeed, with non-reversible Macro-solvers the algorithm suffers from relatively high energy dissipation and the method is unpractical for computations on the $O(\epsilon^{-1})$ time scale. The reversible solver greatly improves energy conservation and a posteriori error analysis suggests that it is convergent.

The Newton-Raphson method used for solving the equations obtained by the Lagrange multipliers, (3.14), becomes inefficient if the partial derivatives matrix of the right hand side of (3.14) is close to singular. In principle, one should then use a different method for solving these equations. Since in this example we are mostly interested in demonstrating energy conservation, we bypass this difficulty by integrating the full system (4.7) using Verlet for a time $H$ whenever this problem occurs. In practice, the Verlet method is used in a few segments whose total length is less than 1% of $[0, T]$ and therefore does not reduce the efficiency considerably. In addition, these segments improve the stability of the algorithm as they smooth out oscillations between even and odd steps.

5. Conclusion

Previously, we have proposed an approach for identifying a change of variables that decomposes a vector field into its fast and slow constituents [2, 3]. The decomposition is used in an HMM algorithm that efficiently integrates the slow parts of the dynamics without fully resolving the fast parts. The algorithm applies a different integrator to each of the time scales in the problem. In this paper we further develop this approach and describe a method in which both the integrators and the feedback between the different scales are implemented in a time reversible way.
We demonstrate that this method enjoys many of the benefits of traditional time reversible integrators such as low energy dissipation. These properties are critical for integrating Hamiltonian systems over long time periods, as was demonstrated in the Fermi-Pasta-Ulam example.

ACKNOWLEDGMENTS

Support from NSF through Grant DMS-0714612 is gratefully acknowledged. Tsai’s research is partially supported by an Alfred P. Sloan Fellowship.

REFERENCES


Department of Mathematics, The University of Texas at Austin, Austin, TX, 78712, USA

E-mail address: ariel@math.utexas.edu

E-mail address: engquist@math.utexas.edu

E-mail address: ytsai@math.utexas.edu