

From Efficient Symplectic Exponentiation of Matrices to Symplectic Integration of High-dimensional Hamiltonian Systems with Slowly Varying Quadratic Stiff Potentials

Molei Tao¹, Houman Owhadi^{1,2}, and Jerrold E. Marsden^{1,2}

¹Control & Dynamical Systems, MC 107-81, California Institute of Technology, Pasadena, CA 91125, USA and ²Applied & Computational Mathematics, MC 217-50, California Institute of Technology, Pasadena, CA 91125, USA

Correspondence to be sent to: mtao@caltech.edu

We present a multiscale integrator for Hamiltonian systems with slowly varying quadratic stiff potentials that uses coarse timesteps (analogous to what the impulse method uses for constant quadratic stiff potentials). This method is based on the highly nontrivial introduction of two efficient symplectic schemes for exponentiations of matrices that only require $\mathcal{O}(n)$ matrix multiplications operations at each coarse time step for a preset small number n . The proposed integrator is shown to be (i) uniformly convergent on positions; (ii) symplectic in both slow and fast variables; (iii) well adapted to high-dimensional systems. Our framework also provides a general method for iteratively exponentiating a slowly varying sequence of (possibly high dimensional) matrices in an efficient way.

1 Introduction

One objective of this paper is to obtain an explicit and efficient numerical integration algorithm for the following multiscale Hamiltonian system:

$$M \begin{bmatrix} \dot{q}^{\text{fast}} \\ \dot{q}^{\text{slow}} \end{bmatrix} = \begin{bmatrix} p^{\text{fast}} \\ p^{\text{slow}} \end{bmatrix}$$

Received July 9, 2010; Revised April 12, 2011; Accepted May 16, 2011

$$\begin{bmatrix} \dot{p}^{\text{fast}} \\ \dot{p}^{\text{slow}} \end{bmatrix} = -\nabla V(q^{\text{fast}}, q^{\text{slow}}) - \epsilon^{-1} \nabla U(q^{\text{fast}}, q^{\text{slow}}) \quad (1)$$

where $q^{\text{slow}}, p^{\text{slow}}$ and $q^{\text{fast}}, p^{\text{fast}}$ are slow and fast degrees of freedom (in the sense that slow degrees of freedom have bounded time derivatives, whereas time derivatives of fast ones may grow unboundedly as $\epsilon \rightarrow 0$). Observe that a direct numerical integration of (1) becomes prohibitive as $\epsilon \downarrow 0$. Notice also that not all stiff Hamiltonian systems are multiscale, and whether a separation of timescales exists depends on specific forms of $V(\cdot), U(\cdot)$ and initial conditions. To the authors' knowledge, a generic theory that determines whether a stiff system is multiscale has not been fully developed yet.

We will mainly discuss and analyze the case where $U(q^{\text{fast}}, q^{\text{slow}}) = \frac{1}{2}[q^{\text{fast}}]^T K(q^{\text{slow}}) q^{\text{fast}}$, which we call, throughout this paper, a quasi-quadratic potential. In this case, the proposed method will be able to integrate the system using a coarse timestep. Notice that if K remains constant with respect to q^{slow} , then the impulse method [12, 15, 32, 35] allows for an accurate and symplectic (see, for instance, [16] for a definition) integration of (1) using coarse steps. The impulse method can, in principle, integrate the situation where K is a regular function of slow variables; however, its practical implementation requires a numerical approximation to the stiff system

$$\begin{aligned} \ddot{q}^{\text{fast}} &= -\epsilon^{-1} \partial U / \partial q^{\text{fast}}(q^{\text{fast}}, q^{\text{slow}}), \\ \ddot{q}^{\text{slow}} &= -\epsilon^{-1} \partial U / \partial q^{\text{slow}}(q^{\text{fast}}, q^{\text{slow}}), \end{aligned} \quad (2)$$

which generally needs to be based on a numerical integration with small steps. The advantage of the impulse method over Verlet is that ∇V only needs to be evaluated at coarse timesteps, but nevertheless its computational cost blows up as $\epsilon \rightarrow 0$.

To use a coarse integration timestep independent of ϵ , we adopt a splitting approach to treat the slow and fast variables separately. At each coarse step, we will require an exact solution or a numerical approximation to the following stiff system:

$$\begin{aligned} \ddot{q}^{\text{fast}} &= -\epsilon^{-1} \partial U / \partial q^{\text{fast}}(q^{\text{fast}}, \cdot), \\ \dot{p}^{\text{slow}} &= -\epsilon^{-1} \partial U / \partial q^{\text{slow}}(q^{\text{fast}}, \cdot), \end{aligned} \quad (3)$$

in which q^{slow} is fixed (different from the impulse method). To obtain such an approximation, we compute the exponential of a matrix that depends on $K(q^{\text{slow}})$ and $\partial K(q^{\text{slow}})$. Still, if not handled appropriately, the cost of this method blows up rapidly as ϵ decreases and/or the dimension of the system increases. Furthermore, symplecticity would also be jeopardized by inaccuracies of the numerical exponentiations.

In this paper, we propose an integrator well-adapted to high-dimensional systems, which computes the exponentiation in an efficient and symplectic way. Only $\mathcal{O}(n)$ matrix multiplication operations at each coarse time step are needed, where n is a preset small integer at most $\log \epsilon^{-1}$. Although simple in appearance, to guarantee the symplecticity (in all variables) of the resulting method without diagonalizing $K(q^{\text{slow}})$ (which is expensive) is a surprisingly difficult problem, and in fact it is highly non-trivial even when $K(q^{\text{slow}})$ is a scalar [22].

In addition to a solution to this problem, this paper also provides a general method for iteratively exponentiating a slowly varying sequence of (possibly high dimensional) matrices in an efficient way (see Section 2.4 and Appendix). This method works for any matrices, and it is not restricted to the integration of (1). The preservation of symplecticity associated with these two proposed matrix exponentiation schemes (both suit high-dimensional systems; the first one is in Section 2.3) is a core difficulty addressed in this paper.

Also, useful discrete geometric structures regarding the flow map of a parameter-dependent vectorial harmonic oscillator have been studied; for instance, derivatives of its flow map with respect to the parameter can be computed using matrix exponentials.

Although backward error analysis (relating symplecticity and energy conservation) does not apply directly to stiff systems (due to large Lipschitz constants), we numerically observe improved long-time behaviors for the proposed integrator, such as near-preservation of energy and conservation of momentum maps. We also note that modulated Fourier expansion [5] has been proposed to explain favorable long-time energy behaviors of some integrators for oscillatory Hamiltonian systems.

We prove the uniform (in ϵ^{-1}) convergence of the method and bound the global error on q^{slow} by CH , where H is a coarse integration timestep and C is a constant independent of ϵ^{-1} .

2 The Proposed Method

2.1 General methodology

Consider a Hamiltonian system with the following Hamiltonian:

$$\mathcal{H}(q, p) = \frac{1}{2} p^T M^{-1} p + V(q) + \epsilon^{-1} U(q^{\text{fast}}, q^{\text{slow}}), \quad (4)$$

where $0 < \epsilon \ll 1$, $q \in \mathbb{R}^d$, $p \in \mathbb{R}^d$. $q^{\text{fast}} \in \mathbb{R}^{d_f}$ and $q^{\text{slow}} \in \mathbb{R}^{d_s}$ are fast and slow variables. For the sake of clarity, we will assume that $q = (q^{\text{fast}}, q^{\text{slow}})$ (and hence $d = d_f + d_s$), but the method presented here can be generalized to the situation where $(q^{\text{fast}}, q^{\text{slow}}) = \eta(q)$, where η is a diffeomorphism explicitly known beforehand (for instance, see [7] for an example where a diffeomorphism puts an extensible pendulum into a quasi-quadratic form). The idea of separating the variables is that fast variables need to be integrated using $o(\sqrt{\epsilon})$ timesteps by a single-scale integrator, whereas slow variables can be resolved using $o(1)$ steps. Dividing variables into different timescales is a widely used technique in studying stiff and multiscale systems (e.g., [10, 20]). A rigorous definition of separation of timescales can be found in [31], for instance.

Without loss of generality, we can further assume M to be the identity matrix. The governing ODE system is

$$\begin{aligned} \dot{q}^{\text{fast}} &= p^{\text{fast}} \\ \dot{p}^{\text{fast}} &= -\epsilon^{-1} \frac{\partial U}{\partial q^{\text{fast}}} - \frac{\partial V}{\partial q^{\text{fast}}} \\ \dot{q}^{\text{slow}} &= p^{\text{slow}} \\ \dot{p}^{\text{slow}} &= -\epsilon^{-1} \frac{\partial U}{\partial q^{\text{slow}}} - \frac{\partial V}{\partial q^{\text{slow}}} \end{aligned} \tag{5}$$

which can be split into a sum of three vector fields:

$$\begin{aligned} \dot{q}^{\text{fast}} &= 0, & \dot{q}^{\text{fast}} &= 0, & \dot{q}^{\text{fast}} &= p^{\text{fast}} \\ \dot{p}^{\text{fast}} &= 0, & \dot{p}^{\text{fast}} &= -\frac{\partial V}{\partial q^{\text{fast}}}, & \dot{p}^{\text{fast}} &= -\epsilon^{-1} \frac{\partial U}{\partial q^{\text{fast}}} \\ \dot{q}^{\text{slow}} &= p^{\text{slow}}, & \dot{q}^{\text{slow}} &= 0, & \dot{q}^{\text{slow}} &= 0 \\ \dot{p}^{\text{slow}} &= 0, & \dot{p}^{\text{slow}} &= -\frac{\partial V}{\partial q^{\text{slow}}}, & \dot{p}^{\text{slow}} &= -\epsilon^{-1} \frac{\partial U}{\partial q^{\text{slow}}} \end{aligned}$$

such that the exact flow of each could be obtained, which is also symplectic at the same time. Indeed, denote the flow maps of all systems by $\phi^i(s)$, $i = 1, 2, 3$ over a time of s . It is easy to see that they are all symplectic.

Observe that ϕ^1 and ϕ^2 are analytically available. We consider only the case where ϕ^3 is also analytically or numerically known; more precisely, the numerical solution $\tilde{\phi}^3$ has to have a consistent uniform local error over a coarse time step $H = o(1)$, i.e., $\|\tilde{\phi}^3(H) - \phi^3(H)\| \leq CH^2$ for a constant C independent of ϵ^{-1} . This can be satisfied for arbitrary $U(\cdot)$ by a symplectic integration with a microscopic timestep $h = o(\sqrt{\epsilon})$, which is in the same spirit as the impulse method. On the other hand, for specific types of $U(\cdot)$,

such as quasi-quadratic stiff potentials (defined in Section 2.2), a method alternative to fine-scale integration can be proposed (see Sections 2.3 and 2.4).

Having the three flow maps at hand, one step update of the proposed method is obtained by composing them, i.e., $\phi^1(H) \circ \phi^2(H) \circ \phi^3(H)$. Notice that any split can result in a convergent numerical scheme, but this particular split treats two timescales independently and therefore is uniformly convergent at least in the quasi-quadratic stiff potential case (illustrated later); also, it results in a symplectic scheme.

Remark 2.1. If there were no slow variable, we would compose the flows of

$$\begin{aligned} \dot{q}^{\text{fast}} &= p^{\text{fast}}, & \dot{p}^{\text{fast}} &= 0 \\ \dot{p}^{\text{fast}} &= -\epsilon^{-1} \frac{\partial U}{\partial q^{\text{fast}}}, & \dot{q}^{\text{fast}} &= -\frac{\partial V}{\partial p^{\text{fast}}} \end{aligned}$$

and obtain a first-order version of the original impulse method. □

Remark 2.2. There are also alternative higher order ways of composing these flow maps; see, for instance, [16, 26]. In fact, the original impulse method is second-order and can be constructed from a second-order composition scheme. However, we will stick to first-order Lie–Trotter ($\phi^1(H) \circ \phi^2(H) \circ \phi^3(H)$) in this paper. □

2.2 Quasi-quadratic fast potentials

We will, from now on, discuss and analyze an analytically solvable case (so that a uniform coarse timestep could be used), in which $U = \frac{1}{2}[q^{\text{fast}}]^T K(q^{\text{slow}})q^{\text{fast}}$, where K is a positive definite d_f -by- d_f symmetric-matrix-valued function. This fast potential represents stiff harmonic oscillators with nonconstant, but slowly varying, frequencies. We call such potentials quasi-quadratic. In this case, the third split vector field is

$$\begin{aligned} \dot{q}^{\text{fast}} &= p^{\text{fast}} \\ \dot{p}^{\text{fast}} &= -\epsilon^{-1} K(q^{\text{slow}})q^{\text{fast}} \\ \dot{q}^{\text{slow}} &= 0 \\ \dot{p}^{\text{slow}} &= -\epsilon^{-1} \frac{1}{2}[q^{\text{fast}}]^T \nabla K(q^{\text{slow}})q^{\text{fast}} \end{aligned} \tag{6}$$

where the last equation is understood as $\dot{p}_i^{\text{slow}} = -\epsilon^{-1} \frac{1}{2}[q^{\text{fast}}]^T \partial_i K(q^{\text{slow}})q^{\text{fast}}$ for $i = 1, \dots, d_s$. The flow of this dynamical system on q^{fast} and p^{fast} is just an exponential map, which in this case corresponds to linear combinations of initial conditions with trigonometric coefficients. For p^{slow} , because q^{slow} (and hence $\nabla K(q^{\text{slow}})$) is fixed, one could

obtain its exact flow by analytically integrating a quadratic function of trigonometric functions.

When $d_f = 1$, the exact flow map of (6) over time H is (letting $\omega = \sqrt{\epsilon^{-1}K(q^{\text{slow}})}$):

$$\begin{aligned}
q^{\text{fast}} &\mapsto \cos(\omega H)q^{\text{fast}} + \sin(\omega H)/\omega p^{\text{fast}} \\
p^{\text{fast}} &\mapsto -\omega \sin(\omega H)q^{\text{fast}} + \cos(\omega H)p^{\text{fast}} \\
q^{\text{slow}} &\mapsto q^{\text{slow}} \\
p^{\text{slow}} &\mapsto p^{\text{slow}} - \epsilon^{-1} \frac{1}{2} \nabla K(q^{\text{slow}}) \frac{1}{4\omega^3} (2\omega(H[p^{\text{fast}}]^2 + p^{\text{fast}}q^{\text{fast}} + \omega^2 H[q^{\text{fast}}]^2) \\
&\quad - 2\omega p^{\text{fast}}q^{\text{fast}} \cos(2\omega H) + (-[p^{\text{fast}}]^2 + \omega^2[q^{\text{fast}}]^2) \sin(2\omega H))
\end{aligned} \tag{7}$$

where again the last equation is understood as

$$\begin{aligned}
p_i^{\text{slow}} &\mapsto p_i^{\text{slow}} - \epsilon^{-1} \frac{1}{2} \partial_i K(q^{\text{slow}}) \frac{1}{4\omega^3} (2\omega(H[p^{\text{fast}}]^2 + p^{\text{fast}}q^{\text{fast}} + \omega^2 H[q^{\text{fast}}]^2) \\
&\quad - 2\omega p^{\text{fast}}q^{\text{fast}} \cos(2\omega H) + (-[p^{\text{fast}}]^2 + \omega^2[q^{\text{fast}}]^2) \sin(2\omega H))
\end{aligned} \tag{8}$$

When $d_f \geq 2$, the obvious method to obtain the exact flow of (6) is based on a diagonalization of K . More precisely, since K is symmetric, we can write $\epsilon^{-1}K(q^{\text{slow}}) = \epsilon^{-1}Q(q^{\text{slow}})^T D(q^{\text{slow}})Q(q^{\text{slow}})$, where $\epsilon^{-1}D(q^{\text{slow}}) = \text{diag}[\omega_1^2, \dots, \omega_{d_f}^2]$. Then

$$\begin{aligned}
&\exp\left(\begin{bmatrix} 0 & HI \\ -\epsilon^{-1}HK(q^{\text{slow}}) & 0 \end{bmatrix}\right) \\
&= \begin{bmatrix} Q^T & 0 \\ 0 & Q^T \end{bmatrix} \exp\left(\begin{bmatrix} 0 & HI \\ -\epsilon^{-1}HD & 0 \end{bmatrix}\right) \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} = \begin{bmatrix} Q^T & 0 \\ 0 & Q^T \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} \text{diag}[\cos(\omega_1 H), \dots, \cos(\omega_{d_f} H)] & \text{diag}[\sin(\omega_1 H)/\omega_1, \dots, \sin(\omega_{d_f} H)/\omega_{d_f}] \\ \text{diag}[-\sin(\omega_1 H)\omega_1, \dots, -\sin(\omega_{d_f} H)\omega_{d_f}] & \text{diag}[\cos(\omega_1 H), \dots, \cos(\omega_{d_f} H)] \end{bmatrix} \\
&\quad \times \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix}
\end{aligned} \tag{9}$$

A similar (but lengthy) calculation will give the expression of the flow on p^{slow} .

If the diagonalization frame of $K(\cdot)$ is constant, i.e., Q does not depend on q^{slow} , then Q needs to be computed only once throughout the simulation, and then the calculation of the flow on q^{fast} and p^{fast} is dominated by the cost of two matrix multiplication operations per coarse step (at expense of $\mathcal{O}(d_f^{2.376})$ per multiplication by the state-of-art Coppersmith–Winograd algorithm [6]). However, if the frame varies (Q depends on

q^{slow}), then diagonalizing K at each time step can offset the gain obtained by the macro-time-stepping of the algorithm. This is especially true if d_f is large. Moreover, errors in numerical diagonalizations may accumulate and deteriorate the symplecticity of ϕ^3 .

In this paper, we address those difficulties by proposing a method, described below, for the numerical integration of (6) that is symplectic and that remains computationally tractable in high-dimensional cases (large d_f).

2.3 Fast numerical matrix exponentiation for the symplectic integration of (6)

The proposed method is based on matrix exponentiation. We will first describe its analytical formulation, and then present an accurate numerical approximation that is both symplectic and computationally cheap.

The first step of our method is based on the following property of matrix exponentials illustrated in [36]: if N and M are constant square matrices of the same dimension, then

$$\exp\left(\begin{bmatrix} -N^T & M \\ 0 & N \end{bmatrix} H\right) = \begin{bmatrix} F_2(H) & G_2(H) \\ 0 & F_3(H) \end{bmatrix} \tag{10}$$

with

$$\begin{aligned} F_2(H) &= \exp(-N^T H) \\ F_3(H) &= \exp(NH) \\ F_3(H)^T G_2(H) &= \int_0^H \exp(N^T s) M \exp(Ns) \, ds \end{aligned} \tag{11}$$

Therefore, ordering coordinates as $q^{\text{fast}}, p^{\text{fast}}$, taking

$$N := \begin{bmatrix} 0 & I \\ -\epsilon^{-1} K(q^{\text{slow}}) & 0 \end{bmatrix} \quad \text{and} \quad M_i := \begin{bmatrix} \epsilon^{-1} \partial_i K(q^{\text{slow}}) & 0 \\ 0 & 0 \end{bmatrix} \quad \text{with } i = 1, \dots, d_s$$

which indicates the component of the slow variable, we obtain that if

$$\begin{bmatrix} F_2(H) & G_{2,i}(H) \\ 0 & F_3(H) \end{bmatrix} := \exp\left(\begin{bmatrix} -N^T & M_i \\ 0 & N \end{bmatrix} H\right) \tag{12}$$

then the (linear) flow map on $q^{\text{fast}}, p^{\text{fast}}$ is given by

$$\exp(NH) = F_3(H) \tag{13}$$

and the drift on p^{slow} is given by

$$\begin{aligned} \int_t^{t+H} q^{\text{fast}}(s)^T \epsilon^{-1} \partial_i K(q^{\text{slow}}) q^{\text{fast}}(s) ds &= \int_0^H \begin{bmatrix} q^{\text{fast}}(t) \\ p^{\text{fast}}(t) \end{bmatrix}^T \exp(N^T s) M_i \exp(Ns) \begin{bmatrix} q^{\text{fast}}(t) \\ p^{\text{fast}}(t) \end{bmatrix} ds \\ &= \begin{bmatrix} q^{\text{fast}}(t) \\ p^{\text{fast}}(t) \end{bmatrix}^T F_3(H)^T G_{2,i}(H) \begin{bmatrix} q^{\text{fast}}(t) \\ p^{\text{fast}}(t) \end{bmatrix} \end{aligned} \quad (14)$$

Therefore, $\phi^3(H)$ is given by:

$$\begin{aligned} \begin{bmatrix} q^{\text{fast}} \\ p^{\text{fast}} \end{bmatrix} &\mapsto F_3(H) \begin{bmatrix} q^{\text{fast}} \\ p^{\text{fast}} \end{bmatrix} \\ q^{\text{slow}} &\mapsto q^{\text{slow}} \\ p_i^{\text{slow}} &\mapsto p_i^{\text{slow}} - \frac{1}{2} \begin{bmatrix} q^{\text{fast}} \\ p^{\text{fast}} \end{bmatrix}^T F_3(H)^T G_{2,i}(H) \begin{bmatrix} q^{\text{fast}} \\ p^{\text{fast}} \end{bmatrix} \end{aligned} \quad (15)$$

where in the last equation $i = 1, \dots, d_s$.

In addition, our specific choice of M_i is a symmetric matrix for each i , because $K(\cdot)$ is symmetric. Consequently, $\exp(N^T s) M_i \exp(Ns)$ is symmetric, and therefore

$$F_3(H)^T G_{2,i}(H) = (F_3(H)^T G_{2,i}(H))^T \quad (16)$$

Assuming we have F_3 and $G_{2,i}$ (which will be given by Integrator 2), (4) can be integrated by the following.

Integrator 1. Symplectic multi-scale integrator for (4) with $U = \frac{1}{2} [q^{\text{fast}}]^T K(q^{\text{slow}}) q^{\text{fast}}$. Its one-step update mapping q_k, p_k onto q_{k+1}, p_{k+1} with a coarse timestep H is given by:

$$\begin{aligned} q_k^{\text{slow}} &= q_k^{\text{slow}} + H p_k^{\text{slow}} \\ q_k^{\text{fast}} &= q_k^{\text{fast}} \end{aligned} \quad (17)$$

$$\begin{aligned} p_k^{\text{slow}} &= p_k^{\text{slow}} - H \partial V / \partial q^{\text{slow}}(q_k^{\text{slow}}, q_k^{\text{fast}}) \\ p_k^{\text{fast}} &= p_k^{\text{fast}} - H \partial V / \partial q^{\text{fast}}(q_k^{\text{slow}}, q_k^{\text{fast}}) \begin{bmatrix} q_{k+1}^{\text{fast}} \\ p_{k+1}^{\text{fast}} \end{bmatrix} = F_{3,k} \begin{bmatrix} q_k^{\text{fast}} \\ p_k^{\text{fast}} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} q_{k+1}^{\text{slow}} &= q_k^{\text{slow}} \\ p_{k+1,i}^{\text{slow}} &= p_{k,i}^{\text{slow}} - \frac{1}{2} \begin{bmatrix} q_k^{\text{fast}} \\ p_k^{\text{fast}} \end{bmatrix}^T F_{3,k}^T G_{2,k,i} \begin{bmatrix} q_k^{\text{fast}} \\ p_k^{\text{fast}} \end{bmatrix} \end{aligned} \quad (18)$$

where $F_{2,k}, G_{2,k,i}$ ($i = 1, \dots, d_s$) and $F_{3,k}$ are numerical approximations of that in (12) at each time step k (using q_k^{slow}), for instance, computed by Integrator 2. \square

To numerically approximate the above flow map (15), i.e., to obtain $F_{3,k}$ and $G_{2,k,i}$, we need to ensure two points: (i) an approximation of the matrix exponential (and hence $F_{3,k}$ and $G_{2,k,i}$) will not affect the symplecticity of the resulting approximation of ϕ^3 ; (ii) the numerical computation of the exponential (12) will not off-set the savings gained by using a coarse timestep. It is highly non-trivial to satisfy both simultaneously, because most matrix exponentiation methods will ruin the symplecticity of ϕ^3 unless high precision (much higher than the requirement on accuracy) is enforced, but then the computational cost will be high. In fact, a necessary and sufficient condition for symplecticity is given by Lemma 3.1, and it is unclear how most matrix exponentiation methods, for instance those based on matrix decompositions (e.g., diagonalization, QR decomposition) with computational costs of Cd_f^3 flops, will satisfy this condition (unless C is very large and the approximation is very accurate). Also, here is an illustration of a popular non-decomposition-based exponentiation method that fails to satisfy this symplecticity condition:

Example. MATLAB function “expm” [17] uses a scaling and squaring strategy based on the following identity:

$$\exp(X) = [\exp(X/2^n)]^{2^n} \quad (19)$$

where n is a big enough preset integer such that $X/2^n$ has a small norm, and therefore Padé approximation [17] could be employed to approximate $\exp(X/2^n)$. The simplest (1,0) Padé approximation, which is essentially Taylor expansion to first order, gives

$$\exp(X) \approx [I + X/2^n]^{2^n} \quad (20)$$

However, this approximation is not symplectic. For instance, consider a counterexample of $X = \begin{bmatrix} 0 & I \\ -\Omega^2 & 0 \end{bmatrix}$. Obviously, this corresponds to a vectorial harmonic oscillator, and $\exp(X)$ ought to be symplectic. However, it can be easily checked that $A := I + X/2^n$ does not satisfy $A^T J A = J$ and hence is not symplectic. \square

Our idea is to obtain $F_{2,k}$ and $F_{3,k}$ using a modified scaling and squaring strategy, in which the Padé approximation is replaced by a symplectic approximation originated from a reversible symplectic integrator (we use Velocity-Verlet). More precisely, suppose $h > 0$ is a small constant, then we have the following identity:

$$\begin{bmatrix} F_{2,k}(H) & G_{2,k,i}(H) \\ 0 & F_{3,k}(H) \end{bmatrix} = \begin{bmatrix} F_{2,k}(h) & G_{2,k,i}(h) \\ 0 & F_{3,k}(h) \end{bmatrix}^{H/h} \quad (21)$$

$F_{3,k}(h)$ can be approximated by the following:

$$\exp \begin{bmatrix} 0 & hI \\ -h\epsilon^{-1}K(q_k^{\text{slow}}) & 0 \end{bmatrix} \approx \begin{bmatrix} I - \frac{h^2}{2}\epsilon^{-1}K(q_k^{\text{slow}}) & h \left(I - \frac{h^2}{4}\epsilon^{-1}K(q_k^{\text{slow}}) \right) \\ -h\epsilon^{-1}K(q_k^{\text{slow}}) & I - \frac{h^2}{2}\epsilon^{-1}K(q_k^{\text{slow}}) \end{bmatrix}, \quad (22)$$

which can be easily checked to be symplectic thanks to the specific $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ corrections in the above expression.

It is a classical result (global error bound of Velocity-Verlet) that links $F_{3,k}(H)$ with the approximated $F_{3,k}(h)$:

$$\left\| \exp \begin{bmatrix} 0 & HI \\ -H\epsilon^{-1}K(q_k^{\text{slow}}) & 0 \end{bmatrix} - \begin{bmatrix} I - \frac{h^2}{2}\epsilon^{-1}K(q_k^{\text{slow}}) & h \left(I - \frac{h^2}{4}\epsilon^{-1}K(q_k^{\text{slow}}) \right) \\ -h\epsilon^{-1}K(q_k^{\text{slow}}) & I - \frac{h^2}{2}\epsilon^{-1}K(q_k^{\text{slow}}) \end{bmatrix} \right\|_2^{H/h} \leq \epsilon^{-1}C \exp(CH)h^2 \quad (23)$$

for some constant $C > 0$, because the approximation in (22) corresponds to the celebrated Velocity-Verlet integrator with updating rule:

$$\begin{aligned} x_{i+\frac{1}{2}} &= x_i + \frac{h}{2}y_i \\ y_{i+1} &= y_i - h\epsilon^{-1}K(q_k^{\text{slow}})x_{i+\frac{1}{2}} \\ x_{i+1} &= x_{i+\frac{1}{2}} + \frac{h}{2}y_{i+1} \end{aligned} \quad (24)$$

for the system

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -\epsilon^{-1}K(q_k^{\text{slow}})x, \end{aligned}$$

which is well known to have a second-order global error.

We can repeat the same procedure to get an approximation of $F_{2,k}(H)$ by using the following approximated $F_{2,k}(h)$:

$$\exp \begin{bmatrix} 0 & h\epsilon^{-1}K^T(q_k^{\text{slow}}) \\ -hI & 0 \end{bmatrix} \approx \begin{bmatrix} I - \frac{h^2}{2}\epsilon^{-1}K^T(q_k^{\text{slow}}) & h\epsilon^{-1}K^T(q_k^{\text{slow}}) \\ -h \left(I - \frac{h^2}{4}\epsilon^{-1}K^T(q_k^{\text{slow}}) \right) & I - \frac{h^2}{2}\epsilon^{-1}K^T(q_k^{\text{slow}}) \end{bmatrix} \quad (25)$$

To approximate $G_{2,k,i}(h)$, we follow the result of Lemma 3.1 that in the continuous case $G_{2,k,i} = -J \frac{\partial}{\partial q_{k,i}^{\text{slow}}} F_{3,k}$ and let

$$G_{2,k,i}(h) = -J \partial_i F_{3,k}(h) \approx \begin{bmatrix} h\epsilon^{-1} \frac{\partial}{\partial q_{k,i}^{\text{slow}}} K(q_k^{\text{slow}}) & \frac{h^2}{2} \epsilon^{-1} \frac{\partial}{\partial q_{k,i}^{\text{slow}}} K(q_k^{\text{slow}}) \\ -\frac{h^2}{2} \epsilon^{-1} \frac{\partial}{\partial q_{k,i}^{\text{slow}}} K(q_k^{\text{slow}}) & -\frac{h^3}{4} \epsilon^{-1} \frac{\partial}{\partial q_{k,i}^{\text{slow}}} K(q_k^{\text{slow}}) \end{bmatrix} \quad (26)$$

Notice that if (1,0) Padé approximation (i.e., first-order Taylor expansion) is used, we will get

$$G_{2,k,i}(h) \approx hM_i = \begin{bmatrix} h\epsilon^{-1} \frac{\partial}{\partial q_{k,i}^{\text{slow}}} K(q_k^{\text{slow}}) & 0 \\ 0 & 0 \end{bmatrix} \quad (27)$$

Naturally, (26) is a higher order correction of this.

$G_{2,k,i}(H)$ will also be accurate: since the accuracy of (20) is well established, the higher order corrections that we add in $F_{2,k}(H)$, $F_{3,k}(H)$, $G_{2,k,i}(H)$ will not lead a scheme less accurate. This can immediately be seen in the context of the numerical integration of a stable system, where a local error of $\mathcal{O}(h^2)$ will only lead to a global error of at most $\epsilon^{-1}CHh$ [21]. We also refer to [25, Appendix A] for an analogous error analysis if one prefers to directly work with matrices.

To sum up, the following numerical approximation of $F_{3,k}$ and $G_{2,k,i}$ will simultaneously guarantee symplecticity, accuracy, and efficiency.

Integrator 2. Matrix exponentiation scheme that complements the updating rule of Integrator 1. $n \geq 1$ is an integer controlling the accuracy of the approximation of the matrix exponentials. k is the same index as the one used in Integrator 1, and the following needs to be done for each k :

1. Evaluate $K_k := K(q_k^{\text{slow}})$ and $\partial_i K_k := \frac{\partial}{\partial q_{k,i}^{\text{slow}}} K(q_k^{\text{slow}})$. Let $h = H/2^n$,

$$A_k := \begin{bmatrix} I - \epsilon^{-1} K_k \frac{h^2}{2} & \epsilon^{-1} K_k h \\ -h \left(I - \epsilon^{-1} K_k \frac{h^2}{4} \right) & I - \epsilon^{-1} K_k \frac{h^2}{2} \end{bmatrix}, \quad (28)$$

$$C_k := \begin{bmatrix} I - \epsilon^{-1} K_k \frac{h^2}{2} & h \left(I - \epsilon^{-1} K_k \frac{h^2}{4} \right) \\ -\epsilon^{-1} K_k h & I - \epsilon^{-1} K_k \frac{h^2}{2} \end{bmatrix}, \quad (29)$$

and for $i = 1, \dots, d_s$,

$$B_{k,i} := \begin{bmatrix} \epsilon^{-1} \partial_i K_k h & \epsilon^{-1} \partial_i K_k \frac{h^2}{2} \\ -\epsilon^{-1} \partial_i K_k \frac{h^2}{2} & -\epsilon^{-1} \partial_i K_k \frac{h^3}{4} \end{bmatrix}. \quad (30)$$

2. Let $F_{2,k}^1 := A_k$, $G_{2,k,i}^1 := B_{k,i}$, $F_{3,k}^1 := C_k$, then repetitively apply

$$\begin{aligned} \begin{bmatrix} F_{2,k}^{j+1} & G_{2,k,i}^{j+1} \\ 0 & F_{3,k}^{j+1} \end{bmatrix} &:= \begin{bmatrix} F_{2,k}^j & G_{2,k,i}^j \\ 0 & F_{3,k}^j \end{bmatrix}^2 \\ &= \begin{bmatrix} F_{2,k}^j F_{2,k}^j & F_{2,k}^j G_{2,k,i}^j + G_{2,k,i}^j F_{3,k}^j \\ 0 & F_{3,k}^j F_{3,k}^j \end{bmatrix} \quad \text{for } j = 1, \dots, n. \end{aligned}$$

3. Define $F_{2,k} := F_{2,k}^{n+1}$, $G_{2,k,i} := G_{2,k,i}^{n+1}$, $F_{3,k} = F_{3,k}^{n+1}$. □

Remark 2.3. The trick for the computational save is that raising to the 2^n th power is computed by n self multiplications, which is due to the semi-group property of the exponentiation operation. An obvious upper bound to guarantee accuracy is $n \leq C \log \epsilon^{-1}$ (because the error of numerical exponentiation is bounded by $\epsilon^{-1} C h = \epsilon^{-1} C H / 2^n$). In all numerical experiments in this paper, $n = 10$ worked well, which is a value much smaller than $\log \epsilon^{-1}$, and this choice of n makes the computation cost of the same order as if K could be diagonalized by a constant orthogonal matrix. □

Remark 2.4. Observe that, for a finite-time simulation, the cost of computing ϕ^3 numerically with microscopic time-steps blows up with a speed of $\mathcal{O}(\epsilon^{-1})$, whereas the cost of matrix exponentiations via Integrator 2 blows up at a maximum speed of $\mathcal{O}(\log \epsilon^{-1})$. □

Theorem 3.7 shows that Integrator 2 not only ensures $F_{2,k}$ and $F_{3,k}$ to be symplectic, but also guarantees a symplectic approximation to ϕ^3 (Equation (15)).

Speed-up is obtained because at each step the computation cost is dominated by $2(d_s + 1)n$ matrix production operations (of $d_f \times d_f$ matrices), where n is a small integer. If the Coppersmith–Winograd algorithm is used to realize the matrix multiplication operation, then the time complexity for exponentiation at each step is $n\mathcal{O}(d_f^{2.376})$ (assuming $d_s = \mathcal{O}(1)$; the problem of matrix exponentiation is less difficult otherwise).

2.4 An alternative matrix exponentiation algorithm based on updating

An alternative way to approximate the flow map (15) is to use the slowly varying property of K to generate a symplectic update of the exponential computed at the previous step. This is based on a generic idea: given an arbitrary sequence of matrices $\{X_k\}$ that vary slowly, use the approximation

$$\exp(X_k) = [\exp(X_k/2^n)]^{2^n} \approx [\exp(X_{k-1}/2^n) \exp((X_k - X_{k-1})/2^n)]^{2^n} \quad (31)$$

where n is a preset constant. Again, we use the trick of self-multiplication for computing the 2^n th power, and efficiency is guaranteed exactly as before.

Accuracy is achieved because, as shown in the following theorem, the approximation error decreases at an exponential rate with respect to n .

Theorem 2.1 ([25, Theorem 5]).

$$\|\exp(A + B) - (\exp(A/2^n) \exp(B/2^n))^{2^n}\|_2 \leq 2^{-n-1} e^{\max(\mu(A+B), \mu(A)+\mu(B))} \|[A, B]\|_2 \quad (32)$$

where $\mu(X)$ is the maximum eigenvalue of $(X^* + X)/2$, and $[A, B] = AB - BA$ is the canonical Lie bracket. \square

Remark 2.5 (Generality). This exponentiation method based on corrections (31) is not limited to the integration of (4), but works for repetitive exponentiations of any slowly varying matrix. It would also work for a set of matrices, as long as they could be indexed to ensure a slow variation. \square

For our purpose of the symplectic integration of (6) (and hence (4)), X_k and A are identified with N in Section 2.3 at each timestep, and B is identified as the difference in N 's between consecutive steps. Since $K(q^{\text{slow}})$ (and hence N as well) is changing slowly, $\|B\|_2 \ll \|A\|_2$; furthermore, the calculation of $[A, B]$ (omitted; notice that B is nilpotent) shows that $\|[A, B]\|_2 \ll \|A\|_2$. Therefore, the error bound here (32) is much smaller than that based on scaling and squaring for the same n . Consequently, we will be able to further decrease the value of n by a few (not a lot because a decrease in n exponentially increases the error).

The reason that we do not identify X_k and A with $\begin{bmatrix} -N^T & M_i \\ 0 & N \end{bmatrix}$ is due to a consideration of symplecticity in all variables, because otherwise $G_{2,k,i}$, obtained as the upper-right block of the exponential, will not be exactly the derivative of $F_{3,k}$. Instead, we let $G_{2,k,i} = -J \frac{\partial}{\partial q_{k,i}^{\text{slow}}} F_{3,k}$, where $F_{3,k}$ is updated from $F_{3,k-1}$ using (31). Taking the derivative, however, incurs additional computation, because $F_{3,k}$ now depends on not only q_k^{slow}

but also q_{k-1}^{slow} , and therefore $\partial q_{k,i}^{\text{slow}} / \partial q_{(k-1)',j}^{\text{slow}}$ has to be computed so that a chain rule applies to facilitate the computation. In the end, the computational saving based on updating the exponentiation becomes less significant due to the extra cost in updating $\partial q_{k,i}^{\text{slow}} / \partial q_{(k-1)',j}^{\text{slow}}$, but the implementation becomes more convoluted. We leave the details to Appendix.

3 Analysis

For a concise writing, we carry out matrix analysis in block forms in this section. Coordinates are ordered as $q^{\text{fast}}, p^{\text{fast}}, q^{\text{slow}}, p^{\text{slow}}$, and therefore $\mathbb{J} = \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix}$ is the coordinate representation of the canonical symplectic 2-form on the full phase space (abusing notations, we use $J := \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$ to represent the symplectic 2-form on both the fast subspace (for $q^{\text{fast}}, p^{\text{fast}}$) and the slow subspace (for $q^{\text{slow}}, p^{\text{slow}}$); this should not affect the clarity of the analysis). We also recall that a map $x \mapsto \phi(x)$ is symplectic if and only if $\phi'(x)^T \mathbb{J} \phi'(x) = \mathbb{J}$ or $\phi'(x)^T J \phi'(x) = J$ for all x 's (depending on whether x represents all variables or only slow or fast variables).

Lemma 3.1. The numerical approximation to ϕ^3 given by (18) is symplectic on all variables if and only if $F_{3,k}$ is symplectic and, for $i = 1, \dots, d_s$, $G_{2,k,i} = -J \frac{\partial F_{3,k}}{\partial q_{k,i}^{\text{slow}}}$ (note that for a fixed i , $G_{2,k,i}$, $\frac{\partial F_{3,k}}{\partial q_{k,i}^{\text{slow}}}$ and J are $d_f \times d_f$ matrices). □

Proof. For conciseness and convenient reading, write q_k^{fast} and p_k^{fast} as q_f and p_f , $\partial / \partial q_{k,i}^{\text{slow}}$ as ∂_i , and $G_{2,k,i}$ and $F_{3,k}$ as $G_{2,i}$ and F_3 in this proof.

The Jacobian of the numerical approximation to $\phi^3 : q_k, p_k \mapsto q_{k+1}, p_{k+1}$ given by (18) can be computed as:

$$A = \left(\begin{array}{c|ccc} F_3 & & & \\ \hline \begin{pmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} & \partial_1 F_3 \begin{pmatrix} q_f \\ p_f \end{pmatrix} & \cdots & \partial_{d_s} F_3 \begin{pmatrix} q_f \\ p_f \end{pmatrix} & \left(\begin{array}{ccc} 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{array} \right) \\ \hline \begin{array}{c} -(q_f^T \quad p_f^T) F_3^T G_{2,1} \\ \vdots \\ -(q_f^T \quad p_f^T) F_3^T G_{2,d_s} \end{array} & & & \begin{array}{c|c} I & 0 \\ \hline -* & I \end{array} \end{array} \right) \tag{33}$$

where $(*)_{i,j} = \frac{1}{2} [q_f; p_f]^T \partial_j (F_3^T G_{2,i}) [q_f; p_f]$, and the 0's in the upper right block, the lower left block, and the lower right block, respectively, corresponds to d_f -by-1, 1-by- d_f , and

d_s -by- d_s zero matrices. Notice that we have $F_3^T G_{2,i}$ in the lower left block because $F_3^T G_{2,i}$ is symmetric (their exact values satisfy this because of (16), and their numerical approximations satisfy this because of Lemma 3.6).

Symplecticity is equivalent to $A^T \mathbb{J} A = \mathbb{J}$, whose left-hand side writes out to be

$$\begin{aligned}
 A^T \mathbb{J} A &= \frac{\begin{array}{c|c} F_3^T J & G_{2,1}^T F_3 \begin{pmatrix} q_f \\ p_f \end{pmatrix} \cdots G_{2,d_s}^T F_3 \begin{pmatrix} q_f \\ p_f \end{pmatrix} \\ \hline (q_f^T \ p_f^T) \partial_1 F_3^T J \\ \vdots \\ (q_f^T \ p_f^T) \partial_{d_s} F_3^T J \\ \hline \begin{pmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} \end{array}}{\begin{array}{c|c} F_3^T J F_3 + 0 & (F_3^T J \partial_1 F_3 + G_{2,1}^T F_3) \begin{pmatrix} q_f \\ p_f \end{pmatrix} \cdots (F_3^T J \partial_{d_s} F_3 + G_{2,d_s}^T F_3) \begin{pmatrix} q_f \\ p_f \end{pmatrix} \\ \hline \Delta & \frac{([q_f; p_f]^T \partial_i F_3^T J \partial_j F_3 [q_f; p_f])_{i=1, \dots, d_s; j=1, \dots, d_s}}{0} \end{array}} \times \frac{\begin{array}{c|c} \begin{pmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{pmatrix} \\ \hline \begin{array}{c|c} *^T & I \\ \hline -I & 0 \end{array} \end{array}}{0 \quad \begin{array}{c|c} - *^T + * & I \\ \hline -I & 0 \end{array}} \\
 & \tag{34}
 \end{aligned}$$

where Δ is naturally negative the transpose of the upper-right block because $A^T \mathbb{J} A$ is skew-symmetric for any A .

This is equal to \mathbb{J} if and only if the upper-left block and the bottom-right block are both J and the upper-right block and the bottom-left block are both zero. The requirement on upper-left block is

$$F_3^T J F_3 = J \tag{35}$$

By the arbitrariness of q_f and p_f , the requirement on upper-right and bottom-left blocks translates to:

$$F_3^T J \partial_i F_3 + G_{2,i}^T F_3 = 0 \tag{36}$$

which further simplifies to

$$G_{2,i} = -J \partial_i F_3 \tag{37}$$

because $F_3^T J \partial_i F_3 = \partial_i (F_3^T J F_3) - \partial_i F_3^T J F_3 = -\partial_i F_3^T J F_3$, F_3 is invertible due to (35), and $J^T = -J$.

The bottom-right block needs to be J , and this requirement is equivalent to

$$[q_f; p_f]^T (\partial_i F_3^T J \partial_j F_3 + \frac{1}{2} \partial_i (F_3^T G_{2,j}) - \frac{1}{2} \partial_j (F_3^T G_{2,i})) [q_f; p_f] = 0 \quad (38)$$

By (37), the above left-hand side rewrites as

$$\begin{aligned} & [q_f; p_f]^T (\partial_i F_3^T J \partial_j F_3 - \frac{1}{2} \partial_i F_3^T J \partial_j F_3 - \frac{1}{2} F_3^T J \partial_i \partial_j F_3 + \frac{1}{2} \partial_j F_3^T J \partial_i F_3 + \frac{1}{2} F_3^T J \partial_j \partial_i F_3) [q_f; p_f] \\ &= [q_f; p_f]^T (\frac{1}{2} \partial_i F_3^T J \partial_j F_3) [q_f; p_f] + [q_f; p_f]^T (\frac{1}{2} \partial_j F_3^T J \partial_i F_3) [q_f; p_f] \end{aligned} \quad (39)$$

Since what are summed up above are just two real numbers, the second number remains the same after taking its transpose, which due to $J^T = -J$ yields

$$[q_f; p_f]^T (\frac{1}{2} \partial_j F_3^T J \partial_i F_3) [q_f; p_f] = -[q_f; p_f]^T (\frac{1}{2} \partial_i F_3^T J \partial_j F_3) [q_f; p_f] \quad (40)$$

Therefore, (38) does hold. ■

Lemma 3.2. In Integrator 2, all A_k and C_k are symplectic; moreover, all $F_{2,k}$ and $F_{3,k}$ are symplectic, too. □

Proof. Straightforward computation using (28) and (29) shows that $A_k^T J A_k = J$ and $C_k^T J C_k = J$. Moreover, since the product of symplectic matrices is symplectic, all $F_{2,k}$ and $F_{3,k}$, being powers of A_k and C_k , are symplectic. ■

Lemma 3.3. In Integrator 2, $A_k^T C_k = I$ (and equivalently $C_k A_k^T = I$) for all k ; moreover, $F_{2,k}^T F_{3,k} = I$ (and equivalently $F_{3,k} F_{2,k}^T = I$). □

Proof. Straightforward computation using (28) and (29) shows that $A_k^T C_k = I$. Therefore, $(A_k A_k)^T C_k C_k = A_k^T I C_k = I$, and by induction $(A_k^{2^n})^T C_k^{2^n} = I$, i.e., $F_{2,k}^T F_{3,k} = I$. ■

Lemma 3.4. In Integrator 2, $B_{k,i} = -J \frac{\partial}{\partial q_{k,i}^{\text{slow}}} C_k$ for all k and i , and $G_{2,k,i} = -J \frac{\partial}{\partial q_{k,i}^{\text{slow}}} F_{3,k}$ for all k and i . □

Proof. Use the short-hand notation $\partial_i := \frac{\partial}{\partial q_{k,i}^{\text{slow}}}$. Straightforward computation using (30) and (29) shows that $B_{k,i} = -J \partial_i C_k$ for all k and i .

Since $\begin{bmatrix} F_{2,k} & G_{2,k,i} \\ 0 & F_{3,k} \end{bmatrix} = \begin{bmatrix} A_k & B_{k,i} 2^n \\ 0 & C_k \end{bmatrix}$ for all i , by induction, it is only necessary to prove that $G_{2,k,i} = -J \partial_i F_{3,k}$ when $n = 1$. In this case, $G_{2,k,i} = A_k B_{k,i} + B_{k,i} C_k$ and $F_{3,k} = C_k C_k$, and the equality can be proved by the following.

Because $B_{k,i} = -J\partial_i C_k$, $C_k^T A_k = I$ (Lemma 3.3) and $J = C_k^T J C_k$ (Lemma 3.2), we have

$$C_k^T A_k B_{k,i} = -C_k^T J C_k \partial_i C_k \quad (41)$$

Since symplectic matrix is nonsingular, this is

$$A_k B_{k,i} = -J C_k \partial_i C_k \quad (42)$$

Adding $B_{k,i} C_k = -J\partial_i C_k C_k$, we have

$$A_k B_{k,i} + B_{k,i} C_k = -J\partial_i (C_k C_k) \quad (43)$$

Hence, the induction works. ■

Lemma 3.5. In Integrator 2, $C_k^T B_{k,i} = B_{k,i}^T C_k$ for all k and i . □

Proof. This can be shown by straightforward computation using (30) and (29). ■

Lemma 3.6. In Integrator 2, $F_{3,k}^T G_{2,k,i} = G_{2,k,i}^T F_{3,k}$ for all k and i . □

Proof. By Lemma 3.5, $C_k^T B_{k,i} = B_{k,i}^T C_k$ for all k and i . By Lemma 3.3, $A_k^T C_k = I$ and $C_k^T A_k = I$.

Since $\begin{bmatrix} F_{2,k} & G_{2,k,i} \\ 0 & F_{3,k} \end{bmatrix} = \begin{bmatrix} A_k & B_{k,i} 2^n \\ 0 & C_k \end{bmatrix}$ for all i , by induction, it is only necessary to prove that $F_{3,k}^T G_{2,k,i} = G_{2,k,i}^T F_{3,k}$ when $n=1$. In this case, $G_{2,k,i} = A_k B_{k,i} + B_{k,i} C_k$ and $F_{3,k} = C_k C_k$, and this equality can be proved upon observing for all i :

$$\begin{aligned} C_k^T C_k^T (A_k B_{k,i} + B_{k,i} C_k) &= C_k^T B_{k,i} + C_k^T C_k^T B_{k,i} C_k = B_{k,i}^T C_k + C_k^T B_{k,i}^T C_k C_k \\ &= B_{k,i}^T A_k^T C_k C_k + C_k^T B_{k,i}^T C_k C_k = (A_k B_{k,i} + B_{k,i} C_k)^T C_k C_k \end{aligned} \quad (44)$$
■

Theorem 3.7. The proposed method (Integrator 1 + 2) is symplectic on all variables. □

Proof. By Lemmas 3.2, 3.4, 3.1, and 3.6, the numerical approximation to ϕ^3 given by (18) is symplectic on all variables.

The flow given by (17) is symplectic on all variables as well, because it is the composition of ϕ^1 and ϕ^2 , which, respectively, correspond to Hamiltonians $\mathcal{H}_1(q^{\text{fast}}, p^{\text{fast}}, q^{\text{slow}}, p^{\text{slow}}) = [p^{\text{slow}}]^2/2$ and $\mathcal{H}_2(q^{\text{fast}}, p^{\text{fast}}, q^{\text{slow}}, p^{\text{slow}}) = V(q^{\text{fast}}, q^{\text{slow}})$, and hence both are symplectic.

Consequently, the proposed method, which composes (17) and (18), is symplectic. ■

3.1 Uniform convergence

This integrator is convergent due to splitting theory [34], i.e., the global error on $q^{\text{slow}}, q^{\text{fast}}, p^{\text{slow}}, p^{\text{fast}}$ is bounded by $\epsilon^{-1}CH$ for some constant $C > 0$ in Euclidean norm.

Moreover, this integrator is uniformly convergent in q under typical or reasonable assumptions, and hence H can be chosen independently from ϵ for stable and accurate integration.

Condition 3.1. We will prove a uniform bound of the global error on position for Integrator 1 under the following (classical) conditions:

1. Regularity: In the integration domain of interest, $\nabla V(\cdot)$ is bounded and Lipschitz continuous with coefficient L , i.e., $\|\nabla V(a) - \nabla V(b)\|_2 \leq L\|a - b\|_2$.
2. Stability and bounded energy: For a fixed T and $t < T$, denote by $x(t) = (q(t), p(t))$ the exact solution to (5), and by $x_t = (q_t, p_t)$ the discrete numerical trajectory given by Integrator 1, then $\|x(t)\|_2^2 \leq C$, $\|x_t\|_2^2 \leq C$, $|\mathcal{H}(q(t), p(t))| \leq C$ and $|\mathcal{H}(q_t, p_t)| \leq C$ for some constant C independent of ϵ^{-1} but dependent on initial condition $\| \begin{bmatrix} q_0 \\ p_0 \end{bmatrix} \|_2^2$ and possibly T as well. \square

Condition 3.2 (Slowly varying frequencies). Consider the solution $q(s), p(s)$ up to time $s \leq H$ to the system

$$\begin{aligned}
 dq^{\text{fast}} &= p^{\text{fast}} dt, \\
 dq^{\text{slow}} &= p^{\text{slow}} dt, \\
 dp^{\text{fast}} &= -\partial V / \partial q^{\text{fast}}(q^{\text{fast}}, q^{\text{slow}}) dt - \epsilon^{-1} K(q^{\text{slow}}) q^{\text{fast}} dt, \\
 dp^{\text{slow}} &= -\partial V / \partial q^{\text{slow}}(q^{\text{fast}}, q^{\text{slow}}) dt - \epsilon^{-1} \frac{1}{2} [q^{\text{fast}}]^T \nabla K(q^{\text{slow}}) q^{\text{fast}} dt,
 \end{aligned} \tag{45}$$

with initial condition $q(0), p(0)$ in the domain of interest that satisfies bounded energy. Assume that q^{fast} can be written as

$$\mathbf{Q}(t) \sum_{i=1}^{d_f} \vec{e}_i \sqrt{\epsilon} a_i(t) \cos[\sqrt{\epsilon^{-1}} \theta_i(t) + \phi_i] \tag{46}$$

where $\mathbf{Q}(t)$ is a slowly varying matrix (i.e., $Q_{ij}(t) \in C^1([0, H])$) and there exists a C independent of ϵ^{-1} such that $\|\mathbf{Q}(t)\| \leq C$ and $\|\dot{\mathbf{Q}}(t)\| \leq C$ for all $t \in [0, H]$, indicating a slowly varying diagonalization frame, d_f is the dimension of the fast variable, \vec{e}_i are standard vectorial basis of \mathbb{R}^{d_f} , $a_i(t)$'s are slowly varying amplitudes (in the same sense as for $\mathbf{Q}(t)$), $\theta_i(t)$'s are non-decreasing and slowly varying in the sense that $\theta_i(t) \in C^2([0, H])$,

$|\ddot{\theta}_i(t)| \leq C$, $|\theta_i(t)| \leq C$, and $C_1 \leq \dot{\theta}_i(t) \leq C_2$ for some $C > 0, C_1 > 0, C_2 > 0$ independent of ϵ^{-1} , and ϕ_i 's are such that $\theta_i(0) = 0$. \square

Remark 3.1. In the case of constant frequencies ($K(\cdot)$ being a constant) and no slow drift ($V(\cdot)$ being a constant), we have $q^{\text{fast}} = \mathbf{Q} \sum_{i=1}^{d_f} \bar{\mathbf{e}}_i \sqrt{\epsilon} a_i \cos[\sqrt{\epsilon^{-1}} \omega_i t + \phi_i]$ (the amplitude is $\mathcal{O}(\sqrt{\epsilon})$ because of bounded energy). When K is not a constant, Condition 3.2 is supported by an asymptotic expansion of q^{fast} . In particular, to the leading order in ϵ , we have $\dot{\theta}_i(t) = \omega_i(t)$ where the $\omega_i^2(t)$ are the eigenvalues of $K(q_s^{\text{slow}})$. The rigorous justification of this asymptotic expansion for $d_f > 1$ is beyond the scope of this paper. \square

Lemma 3.8. If Condition 3.2 holds, there exists $C_1 > 0, C_2 > 0$ independent of ϵ^{-1} such that

$$\left\| \int_0^H f(t) q^{\text{fast}}(t) dt \right\| \leq \epsilon \left(C_1 \max_{0 \leq s \leq H} \|f(s)\| + C_2 H \max_{0 \leq s \leq H} \|\dot{f}(s)\| + \mathcal{O}(H^2) \right) \quad (47)$$

for arbitrary matrix valued function $f \in C^1([0, H])$ that satisfies $f(0) = 0$. \square

Proof. Recall the form of q^{fast} in Condition 3.2. It is sufficient to prove that for all i 's the i th component of q^{fast} satisfies (47), whereas the i th component writes as:

$$\sqrt{\epsilon} \sum_{j=1}^{d_f} Q_{ij}(t) a_j(t) \cos[\sqrt{\epsilon^{-1}} \theta_i(t) + \phi_i] \quad (48)$$

Furthermore, since summation commutes with integral and therefore will only introduce a factor of d_f on the bound, it is sufficient to prove (47) for $q^{\text{fast}} = \sqrt{\epsilon} Q_{ij}(t) a_j(t) \cos[\sqrt{\epsilon^{-1}} \theta_i(t) + \phi_i]$. On this token, we could assume that we are in the 1D case and absorb $Q(t)$ into $a_j(t)$.

Similarly, slowly varying $a_i(t)$ can be absorbed into the test function $f(t)$, and doing so will only change the constants on the right-hand side. Therefore, it will be sufficient to prove that

$$\left| \int_0^H \sqrt{\epsilon} \cos[\sqrt{\epsilon^{-1}} \theta(t) + \phi] f(t) dt \right| \leq \epsilon \left(C_1 \max_{0 \leq s \leq H} |f(s)| + C_2 H \max_{0 \leq s \leq H} |f'(s)| + \mathcal{O}(H^2) \right) \quad (49)$$

for a scalar-valued function $f \in C^1([0, H])$ that satisfies $f(0) = 0$.

By Condition 3.2, θ is strictly increasing. If we write $\tau = \theta(t)$, there will be a θ^{-1} such that $t = \theta^{-1}(\tau)$. With time transformed to the new variable τ , the integral on the left-hand side of (49) is equal to

$$\int_0^{\theta(H)} \sqrt{\epsilon} \cos[\sqrt{\epsilon^{-1}} \tau + \phi] f(\theta^{-1}(\tau)) \frac{d\theta^{-1}}{d\tau}(\tau) d\tau \quad (50)$$

By integration by parts, this is (since $f(0) = 0$)

$$-\epsilon \sin[\sqrt{\epsilon^{-1}}H + \phi]f(H) \frac{1}{\dot{\theta}(H)} + \epsilon \int_0^{\theta(H)} \sin[\sqrt{\epsilon^{-1}}\tau + \phi] \left[\frac{df}{d\tau} \left(\frac{d\theta^{-1}}{d\tau} \right)^2 + f(\theta^{-1}(\tau)) \frac{d^2\theta^{-1}}{d\tau^2}(\tau) \right] d\tau \quad (51)$$

Because $\ddot{\theta} \leq C$, $\omega - CH \leq \dot{\theta} \leq \omega + CH$, where $\omega := \dot{\theta}(0) \geq C_1 > 0$. Together with $\frac{d\theta^{-1}}{d\tau} = \frac{1}{\dot{\theta}}$, we have $\frac{d^2\theta^{-1}}{d\tau^2} = 1/\omega + \mathcal{O}(H)$. Similarly, we also have

$$\frac{d^2\theta^{-1}}{d\tau^2} = \frac{d}{d\tau} \frac{1}{\dot{\theta}(t)} = \frac{dt}{d\tau} \frac{d}{dt} \frac{1}{\dot{\theta}(t)} = -\frac{1}{\dot{\theta}(t)^3} \ddot{\theta}(t) = \mathcal{O}(1) \quad (52)$$

It is easy to show that $\theta(H) = \mathcal{O}(H)$. Together with $\sin(\cdot)$ being $\mathcal{O}(1)$, the left-hand side in (49) is bounded by

$$\begin{aligned} & \epsilon f(H)\mathcal{O}(1) + \epsilon \mathcal{O}(H) \left(\mathcal{O}(1) \max_{0 \leq s \leq H} |\dot{f}(s)| + \mathcal{O}(1) \max_{0 \leq s \leq H} |f(s)| \right) \\ & \leq \epsilon \left(\mathcal{O}(1) \max_{0 \leq s \leq H} |f(s)| + \mathcal{O}(H) \max_{0 \leq s \leq H} |\dot{f}(s)| \right) \end{aligned} \quad (53)$$

■

Theorem 3.9. If Conditions 3.1 and 3.2 hold, the proposed method (Integrator 1) for system (5) has a uniform global error of $\mathcal{O}(H)$ in q , given a fixed total simulation time $T = NH$:

$$\|q(T) - q_T\|_2 \leq CH \quad (54)$$

where $q(T)$, $p(T)$ are the exact solution and q_T , p_T are the numerical solution; C is a positive constant independent of ϵ^{-1} but dependent on simulation time T , scaleless elasticity matrix K , slow potential energy $V(\cdot)$, and initial condition $\| \begin{bmatrix} q_0 \\ p_0 \end{bmatrix} \|_2$. □

Proof. Let \tilde{K} be a constant matrix and consider the following system:

$$\begin{aligned} d\tilde{q}^{\text{fast}} &= \tilde{p}^{\text{fast}} dt, \\ d\tilde{q}^{\text{slow}} &= \tilde{p}^{\text{slow}} dt, \\ d\tilde{p}^{\text{fast}} &= -\partial V / \partial q^{\text{fast}}(\tilde{q}^{\text{fast}}, \tilde{q}^{\text{slow}}) dt - \epsilon^{-1} \tilde{K} \tilde{q}^{\text{fast}} dt, \\ d\tilde{p}^{\text{slow}} &= -\partial V / \partial q^{\text{slow}}(\tilde{q}^{\text{fast}}, \tilde{q}^{\text{slow}}) dt, \end{aligned} \quad (55)$$

Integrator 1, applied to the system (55) under Condition 3.1, has been shown in [32] to be uniformly convergent in “energy norm,” i.e., with local error $\|[\tilde{q}(H), \tilde{p}(H)] - [\tilde{q}_H, \tilde{p}_H]\|_E \leq C_1 H^2$ and global error $\|[\tilde{q}(T), \tilde{p}(T)] - [\tilde{q}_T, \tilde{p}_T]\|_E \leq C_2 H$, where C_1 and C_2 are constants that do not depend ϵ^{-1} (C_2 depends on T). Recall that the “energy norm” was defined in

[32] to be

$$\|[\tilde{q}, \tilde{p}]\|_E = \sqrt{\tilde{q}^T \tilde{q} + \epsilon \tilde{p}^T \tilde{K}^{-1} \tilde{p}}, \tag{56}$$

but in fact \tilde{K}^{-1} is not important because it is just $\mathcal{O}(1)$, and the following definition would also work for the proof there:

$$\|[\tilde{q}, \tilde{p}]\|_E = \sqrt{\tilde{q}^T \tilde{q} + \epsilon \tilde{p}^T \tilde{p}} \tag{57}$$

Observe that (56) is proportional to the square root of the physical energy, and this is why the name. It can be seen that uniform convergence in energy norm means uniform convergence on position but nonuniform convergence on momentum.

However, the system considered here is (45). To prove uniform convergence for (45), it is sufficient to show that (i) a δ difference between two trajectories of (55) in energy norm leads to a difference of $\delta(1 + CH)$ in energy norm after a time step H (ii) trajectories of (55) and (45) starting at the same point remain at a distance at most $\mathcal{O}(H^2)$ in energy norm after time H , i.e., a second-order uniform local error. (i) was shown by [32, Lemma 6.5], and we will now prove (ii).

We can assume without loss of generality that we start at time 0, and let $\tilde{K} = K(q^{\text{slow}}(0))$, $q^{\text{fast,slow}}(0) = \tilde{q}^{\text{fast,slow}}(0)$ (where $q^{\text{fast,slow}} = (q^{\text{fast}}, q^{\text{slow}})$) and $p^{\text{fast,slow}}(0) = \tilde{p}^{\text{fast,slow}}(0)$. We first let $x = \tilde{q}^{\text{fast}} - q^{\text{fast}}$ and $y = \tilde{p}^{\text{fast}} - p^{\text{fast}}$, and proceed to bound x and y .

The evolutions of x and y follow from

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= - \left(\frac{\partial V}{\partial q^{\text{fast}}}(\tilde{q}) - \frac{\partial V}{\partial q^{\text{fast}}}(q) \right) - \epsilon^{-1} (\tilde{K} \tilde{q}^f - K(q^{\text{slow}}) q^{\text{fast}}) \end{aligned} \tag{58}$$

Writing

$$f_1 = - \left(\frac{\partial V}{\partial q^{\text{fast}}}(\tilde{q}) - \frac{\partial V}{\partial q^{\text{fast}}}(q) \right) \quad \text{and} \quad f_2 = (\tilde{K} - K(q^{\text{slow}})) q^{\text{fast}},$$

we have

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= f_1 - \epsilon^{-1} \tilde{K} x - \epsilon^{-1} f_2 \end{aligned} \tag{59}$$

If we let

$$B(t) = \exp \left(\begin{bmatrix} 0 & I \\ -\epsilon^{-1} \tilde{K} & 0 \end{bmatrix} t \right),$$

we will have

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = B(t) \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + \int_0^t B(t-s) \begin{bmatrix} 0 \\ f_1 - \epsilon^{-1} f_2 \end{bmatrix} ds \tag{60}$$

The first term on the right-hand side drops off because $x(0) = 0$ and $y(0) = 0$ by definition.

Since \tilde{K} is a constant matrix, it is sufficient to diagonalize it and treat each diagonal element individually. Hence, assume without loss of generality that we are in the one-dimensional case. Then

$$B(s) = \begin{bmatrix} \cos(\sqrt{\epsilon^{-1}\tilde{K}}s) & \sin(\sqrt{\epsilon^{-1}\tilde{K}}s)/\sqrt{\epsilon^{-1}\tilde{K}} \\ -\sqrt{\epsilon^{-1}\tilde{K}}\sin(\sqrt{\epsilon^{-1}\tilde{K}}s) & \cos(\sqrt{\epsilon^{-1}\tilde{K}}s) \end{bmatrix}.$$

As a consequence,

$$y(t) = \int_0^t \cos[\sqrt{\epsilon^{-1}\tilde{K}}(t-s)] [f_1 - \epsilon^{-1}(\tilde{K} - K(q^{\text{slow}}))q^{\text{fast}}] ds \quad (61)$$

By Lipschitz continuity of ∇V (Item 1 of Condition 3.1), we will have

$$|f_1(t)| \leq L|x(t)| = L \left| \int_0^t y(s) ds \right| = \mathcal{O}(t) \quad (62)$$

The first inequality holds because f_1 is the difference between partial derivatives of V , which could be bounded by the difference between full derivatives. The last equality holds because $y = p - \tilde{p}$ is bounded due to the fact that $[q(s), p(s)]$ and $[\tilde{q}(s), \tilde{p}(s)]$ are bounded (Item 2 of Condition 3.1). Consequently, we have

$$\left| \int_0^t \cos[\sqrt{\epsilon^{-1}\tilde{K}}(t-s)] f_1 ds \right| \leq \int_0^t |f_1| = \mathcal{O}(t^2) \quad (63)$$

In order to bound $\int_0^t \cos[\sqrt{\epsilon^{-1}\tilde{K}}(t-s)] [\epsilon^{-1}(\tilde{K} - K(q^{\text{slow}}))q^{\text{fast}}] ds$, we use Lemma 3.8 (with the choice of $f = \tilde{K} - K(q^{\text{slow}})$). Indeed, $\cos[\sqrt{\epsilon^{-1}\tilde{K}}(t-s)]$ can be absorbed into $q^{\text{fast}}(s) = \sqrt{\epsilon} \cos[\sqrt{\epsilon^{-1}\tilde{K}}\theta(s) + \phi]$: due to an equality $2\cos(A)\cos(B) = \cos(A+B) + \cos(A-B)$, θ will be just added by $\pm\sqrt{\tilde{K}}$ and ϕ will have a new constant value, neither of which will violate Condition 3.2.

For f , we clearly have $f = 0$ at $s = 0$. By mean value theorem, there is a ξ_s such that $f(s) = K \circ q^{\text{slow}}(0) - K \circ q^{\text{slow}}(s) = \frac{dK \circ q^{\text{slow}}}{dt}(\xi_s) \cdot s$, and therefore $f(s) = \mathcal{O}(s)$. Similarly, $\dot{f}(s) = \mathcal{O}(1)$. Plotting these two bounds in Lemma 3.8, we obtain

$$\left| \int_0^t \cos[\sqrt{\epsilon^{-1}\tilde{K}}(t-s)] [\epsilon^{-1}(\tilde{K} - K(q^{\text{slow}}))q^{\text{fast}}] ds \right| = \mathcal{O}(t) \quad (64)$$

Putting this together with (63), we arrive in $y(t) = \mathcal{O}(t)$, and $x(t) = \int_0^t y(s) ds = \mathcal{O}(t^2)$ follows.

Next, we bound y : since

$$\begin{aligned} & \left| \int_0^t \cos[\sqrt{\epsilon^{-1}\tilde{K}}(t-s)][\epsilon^{-1}(\tilde{K} - K(q^{\text{slow}}))q^{\text{fast}}] ds \right| \\ &= \left| \int_0^t \cos[\dots]\epsilon^{-1}\mathcal{O}(s)\sqrt{\epsilon}\mathcal{O}(1)\cos[\dots] ds \right| = \epsilon^{-1/2}\mathcal{O}(t^2) \end{aligned} \tag{65}$$

we have $y(t) = \epsilon^{-1/2}\mathcal{O}(t^2)$. Together with $x(t) = \mathcal{O}(t^2)$, this is equivalent to $\|[x, y]\|_E = \mathcal{O}(t^2)$.

Similarly, we can bound $q^{\text{slow}} - \tilde{q}^{\text{slow}}$ and $p^{\text{slow}} - \tilde{p}^{\text{slow}}$. Let $x^s = q^{\text{slow}} - \tilde{q}^{\text{slow}}$ and $y^s = p^{\text{slow}} - \tilde{p}^{\text{slow}}$, then we have

$$\begin{aligned} \dot{x}^s &= y^s \\ \dot{y}^s &= - \left(\frac{\partial V}{\partial q^{\text{slow}}}(q) - \frac{\partial V}{\partial q^{\text{slow}}}(\tilde{q}) \right) - \epsilon^{-1} \frac{1}{2} [q^{\text{fast}}]^T \nabla K(q^{\text{slow}}) q^{\text{fast}} \end{aligned} \tag{66}$$

Analogous to before, the first term on the right-hand side of the y^s dynamics is $\mathcal{O}(t)$. Since $q^{\text{fast}} = \mathcal{O}(\epsilon^{1/2})$, the second term on the right-hand side is $\mathcal{O}(1)$. Therefore, $\dot{y}^s = \mathcal{O}(1)$, $y^s(t) = y^s(0) + \mathcal{O}(t) = \mathcal{O}(t)$, and $x^s(t) = x^s(0) + \int_0^t y^s(s) ds = \mathcal{O}(t^2)$. For our purpose of fast integration, we use a big timestep $H \geq \sqrt{\epsilon}$, and hence $y^s(H) = \mathcal{O}(H) \leq \epsilon^{-1/2}\mathcal{O}(H^2)$ (notice that if $H < \sqrt{\epsilon}$, we do not even need to prove uniform convergence, because the nonuniform error bound that is guaranteed by Lie–Trotter splitting theory is already very small).

$\mathcal{O}(H^2)$ and $\epsilon^{-1/2}\mathcal{O}(H^2)$ bounds on separations of slow position and slow momentum imply a $\mathcal{O}(t^2)$ uniform bound in energy norm (analogous to that of the fast degrees of freedom). This demonstrates a second-order uniform local error on all variables in energy norm, and therefore concludes the proof. ■

Remark 3.2. Unlike (54), a global bound on the error of momentum will not be uniform. The error propagation is quantified in energy norm, and in 2-norm we will only have $\epsilon^{-1/2}\mathcal{O}(H^2)$ local error and $\epsilon^{-1/2}\mathcal{O}(H)$ global error on momentum. In fact, Integrator 1 applied to the constant frequency system (55) is nonuniformly convergent on momentum [32]. □

4 Numerical Examples

4.1 The case of a diagonal frequency matrix

Consider the Hamiltonian example introduced in [22]:

$$\mathcal{H} = \frac{1}{2} p_x^2 + \frac{1}{2} p_y^2 + (x^2 + y^2 - 1)^2 + \frac{1}{2}(1 + x^2)\omega^2 y^2 \tag{67}$$

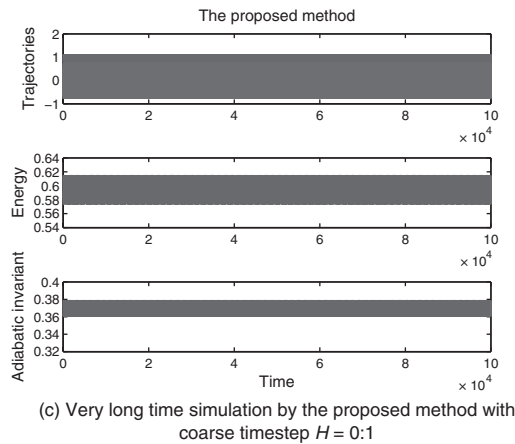
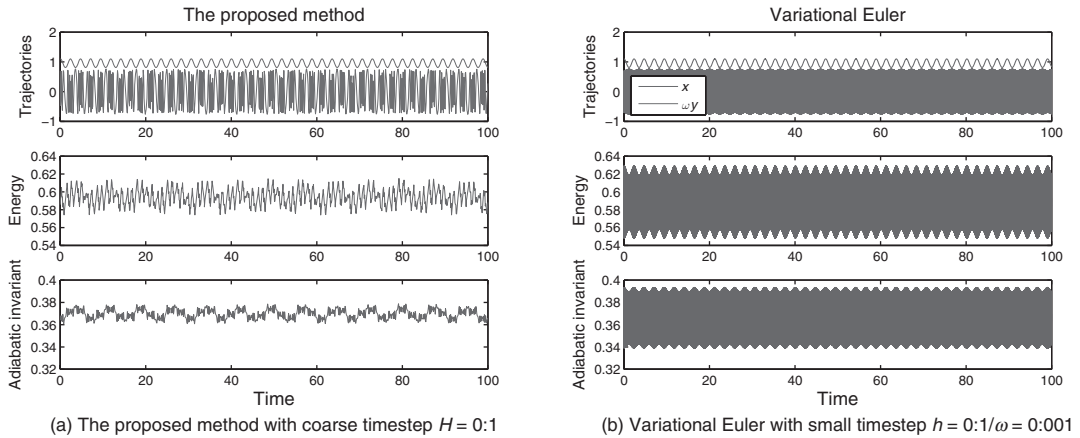


Fig. 1. Simulations of a diagonal fast frequency example (67) by the proposed method and variational Euler. $\omega = 100$; $x(0) = 1.1$, $y(0) = 0.7/\omega$.

When $\omega = \epsilon^{-1/2} \gg 1$, bounded energy translates to initial conditions $x(0) \sim \omega y(0)$, which satisfy separation of timescales: x is the slow variable, and y is the fast. $K(x) = 1 + x^2$ is trivially diagonal. In addition to conservation of total energy, $I = \frac{p_y^2}{2\sqrt{1+x^2}} + \frac{\sqrt{1+x^2}\omega^2 y^2}{2}$ is an adiabatic invariant.

A comparison between variational Euler (VE) and the proposed method is shown in Figure 1. In the figure, it can be seen that preservations of energy and adiabatic invariant are numerically captured at least to a very large timescale. Since there is no overhead spent on matrix exponentiation here, an accurate $100\times$ speed up is achieved by the proposed method (because $H/h = 100$).

It is known that the impulse method and its derivatives (such as mollified impulse methods) are not stable if the integration step falls in resonance intervals

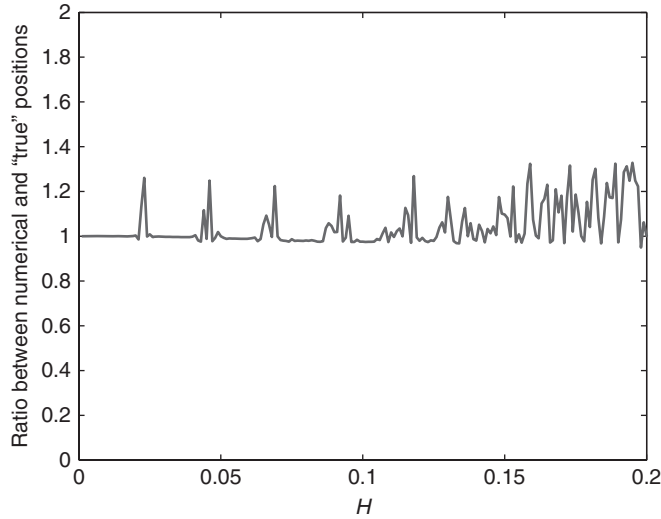


Fig. 2. Investigation on resonance frequencies of the proposed method on example (67). The ratio between $x(T)|_{T=100}$ integrated by the proposed method integration and benchmark provides the ruler: a ratio closer to 1 means a more accurate integration, and deviations from 1 stand for step lengths that correspond to resonance frequencies. Time step H samples from 0.001 to 0.2 with an increment of 0.001. $\omega = 100$; $x(0) = 1.1$, $y(0) = 0.7/\omega$. Benchmark is obtained by fine VE integration with $h = 0.01/\omega$.

(mollified impulse methods have much narrower resonance intervals, which however still exist) [3, 12]. Similarly, it will be very unnatural if the proposed method does not have resonance, because it reduces to a first-order version of impulse methods when there is no slow variable (Remark 2.1). In fact, in our numerical investigation (Figure 2), we clearly observe resonance frequencies before the integration step reaches the unstable limit (around $H \approx 0.5$), and widths of resonant intervals increase as H grows for this particular example; however, we will not carry out a systematic analysis on resonance due to limitation of the length of a short communication.

4.2 The case of a non-diagonal frequency matrix

Extend the previous example to a toy example of three degrees of freedom:

$$\mathcal{H} = \frac{1}{2}p_x^2 + \frac{1}{2}p_y^2 + \frac{1}{2}p_z^2 + (x^2 + y^2 + z^2 - 1)^2 + \frac{1}{2}\omega^2 \begin{bmatrix} y \\ z \end{bmatrix}^T \begin{bmatrix} 1 + x^2 & x^2 - 1 \\ x^2 - 1 & 3x^2 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad (68)$$

It is easy to check that eigenvalues of $K(x) = \begin{bmatrix} 1+x^2 & x^2-1 \\ x^2-1 & 3x^2 \end{bmatrix}$ are both positive when $x > 0.44$, which will always be true if the initial condition of x stays close to 1 and ω is big enough. In this case, bounded energy again implies $x(0) \sim \omega y(0) \sim \omega z(0)$ and gives well separation of timescales: x is the slow variable and y and z are the fast. $K(x)$ has its orthogonal frame for diagonalization as well as its eigenvalues slowly varying with time.

Figure 3 shows a comparison between VE, the proposed method with the matrix exponentiations computed by diagonalization and analytical integration (Equation (9); diagonalization implemented by MATLAB command 'diag'), and the proposed methods based on exponentiations (Equations (10) and (15)) via MATLAB command 'expm' [17] and via the fast matrix exponentiation method (Integrator 2). The default MATLAB matrix multiplication operation is used. All implementations of the proposed method are accurate, except that numerical errors in repetitive diagonalizations contaminated the symplecticity of the corresponding implementation over a long-time simulation (as suggested by drifted energy), whereas other two implementations, respectively, based on

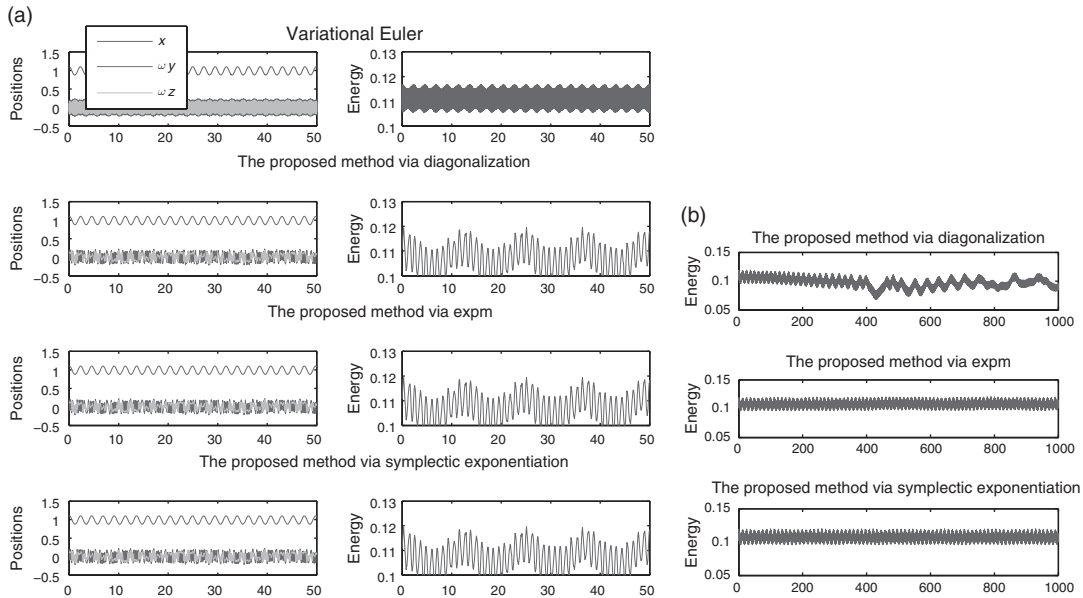


Fig. 3. Simulations of a nondiagonal fast frequency example (68) by VE, the proposed method with different implementations of matrix exponentiations. $\omega = 100$, VE uses $h = 0.1/\omega = 0.001$ and the proposed method uses $H = 0.1$ and $n = 10$; $x(0) = 1.1$, $y(0) = 0.2/\omega$, $z(0) = 0.1/\omega$, and initial momenta are zero.

accurate but slow ‘expm’ and fast symplectic exponentiations, do not have this issue. In a typical notebook run with MATLAB R2008b, the above four methods, respectively, spent 11.12, 0.23, 0.29 and 0.24 s on the same integration (till time 50), while 0, 0.14, 0.18, and 0.14 s were spent on matrix exponentiations. Computational gain by the symplectic exponentiation algorithm will be much more significant as the fast dimension becomes higher. Notice also that the computational gain by the proposed method over VE will go to infinity as $\epsilon \rightarrow 0$, even if the fast matrix exponentiation method is not employed.

4.3 The case of a high-dimensional nondiagonal frequency matrix

Consider an arbitrarily high-dimensional example:

$$\mathcal{H} = \frac{1}{2}p^2 + \frac{1}{2}y^T y + (x^T x + q^2 - 1)^2 + \frac{1}{2}\omega^2 x^T T(q)x \quad (69)$$

where $q, p \in \mathbb{R}$ correspond to the slow variable, $x, y \in \mathbb{R}^{d_f}$ correspond to fast variables, and $T(q)$ is the following Toeplitz matrix valued function:

$$T(q) = \begin{bmatrix} 1 & \hat{q}^1 & \hat{q}^2 & \dots & \hat{q}^{d_f-1} \\ \hat{q}^1 & 1 & \hat{q}^1 & \dots & \hat{q}^{d_f-2} \\ \hat{q}^2 & \hat{q}^1 & 1 & \dots & \hat{q}^{d_f-3} \\ & & \vdots & & \\ \hat{q}^{d_f-1} & \hat{q}^{d_f-2} & \hat{q}^{d_f-3} & \dots & 1 \end{bmatrix} \quad (70)$$

where $\hat{q} = q/2$ so that eigenvectors and eigenvalues vary slowly with q given an initial condition of $q(0) \approx 1$. Note that the expression of $T(\cdot)$ is highly nonlinear.

We present in Figure 4 a comparison between VE and the proposed methods with the matrix exponentials computed by MATLAB command ‘expm’ and by the fast matrix exponentiation method (Integrator 2) on a high-dimensional example with $d_f = 100$. Accuracy-wise, the proposed method simulations yield results similar to VE (note that fast variables are not fully resolved due to a coarse time step that is larger than their periods). Speed-wise, VE, the proposed methods via ‘expm’ and via symplectic exponentiation, respectively, spent 136.7, 66.0 and 12.0 s on the same integration, while 65.7 and 11.7 s were spent on matrix exponentiation operations in the latter two. Notice that if Coppersmith–Winograd [6] is used to replace MATLAB matrix multiplication, the number 11.7 should be further reduced. In spite of that, the proposed method with the proposed matrix exponentiation scheme already holds a dominant speed advantage, and this advantage will be even more significant if ω and/or d_f is further increased (results not shown).

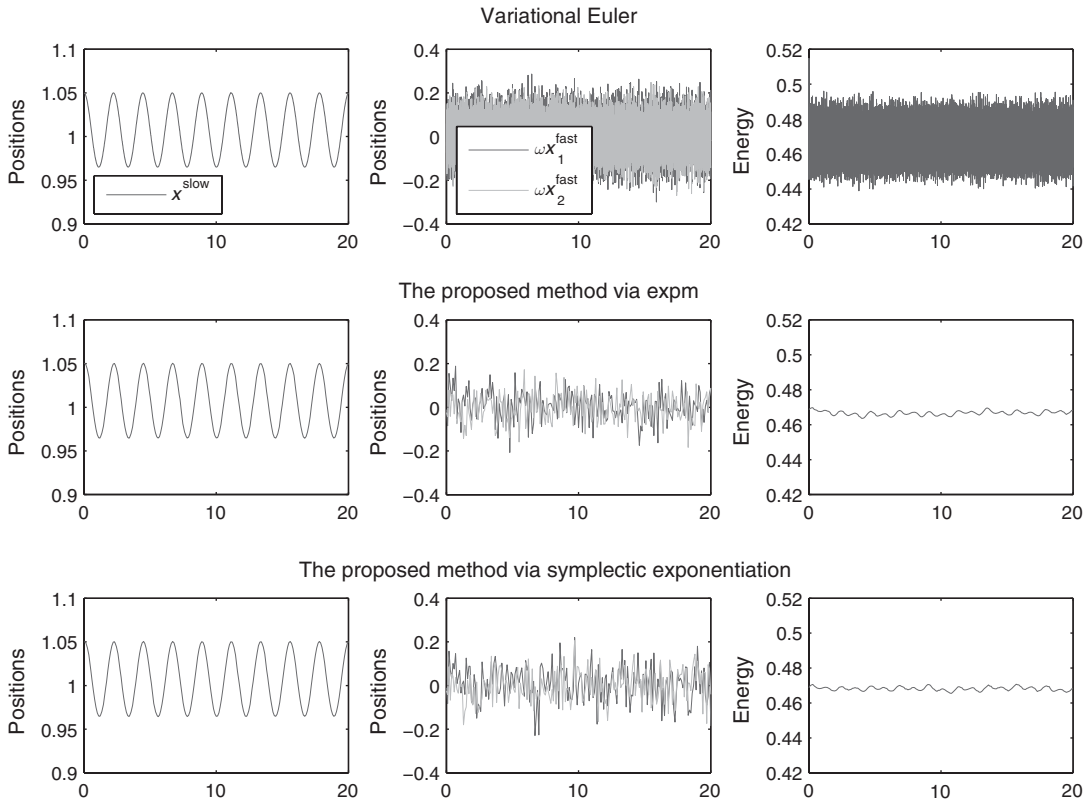


Fig. 4. Simulations of a nondiagonal fast frequency high-dimensional example (70) by VE, the proposed method via MATLAB matrix exponentiation “expm,” and the proposed method via fast matrix exponentiations ($n=10$). Fast variable dimensionality is $d_f=100$. $\omega=1000$. VE uses $h=0.1/\omega$ and the proposed method uses $H=0.1$, $q(0)=1.05$, $x(0)$ is a d_f+1 -dimensional vector with independent and identically distributed components that are normal random variables with zero mean and variance of $1/\omega/\sqrt{d_f}$ (so that energy is bounded), and initial momenta are zero. Only trajectories of the first two fast variables were drawn for clarity.

5 Related Work

5.1 Stiff integration

Many elegant methods have been proposed in the area of stiff Hamiltonian integration, and some are closely related to this work. An incomplete list will be discussed here.

Impulse methods [15, 35] admit uniform error bounds on positions and can be categorized as splitting methods [32]. In their abstract form, impulse methods are not limited to quadratic stiff potentials; however, their practical implementation requires

an approximation of the flow associated with the stiff potential. Our method is based on a generalization of the impulse method to (possibly high-dimensional) situations where the stiff potential contains a slowly varying component. Although simple in its abstract expression, the practical implementation of this generalization (for high-dimensional systems) has required the introduction of a nontrivial symplectic matrix exponentiations scheme.

Impulse methods have been mollified [12, 27] to gain extra stability and accuracy. However, mollified impulse methods and other members of the exponential integrator family [14], for instance Gautschi-type integrators [18], are not based on splitting, and hence the splitting approach in this paper does not immediately generalize them.

The reversible averaging integrator proposed in [23] averages the force on slow variables and avoids resonant instabilities. It treats the dynamics of slow and fast variables separately and assumes piecewise linear trajectories of the slow variables, both in the same spirit as in our proposed method; it is, however, not symplectic, although reversible.

Implicit methods, for example LIN [37], work for generic stiff Hamiltonian systems, but *implicit methods in general fail to capture the effective dynamics of the slow time scale because they cannot correctly capture non-Dirac invariant distributions* [24], and they are generally slower than explicit methods if comparable step lengths are employed.

IMEX is a variational integrator for stiff Hamiltonian systems [30]. It works by introducing a discrete Lagrangian via trapezoidal approximation of the soft potential and midpoint approximation of the stiff potential. It is explicit in the case of quadratic fast potential, but is implicit in the case of quasi-quadratic fast potentials.

A Hamilton–Jacobi approach is used to derive a homogenization method for multiscale Hamiltonian systems [22], which works for quasi-quadratic fast potentials with scalar frequency and yields a symplectic method. We also refer to [7] for a generalization of this method to systems that have either one varying fast frequency or several constant frequencies. The difficulty with this elegant analytical approach would be to deal with high-dimensional systems.

Other generic multiscale methods that integrate the slow dynamics by averaging the effective contribution of the fast dynamics include: heterogeneous multiscale methods (HMM) [1, 4, 8, 10, 11], the equation-free method [13, 19, 20], and FLOW AVeraging integratORS (FLAVORS) [31]. Those methods can be applied to a much broader spectrum of problems than considered here. However, they all essentially use a mesoscopic timestep, which is usually one or two orders of magnitude smaller than the coarse step

employed here. Moreover, symplecticity is a big concern. In their original form, both HMM and equation-free method are based on the averaging of the instantaneous drifts of slow variables, which breaks symplecticity in all variables. Reversible and symmetric HMM generalizations have been proposed [2, 28]. FLAVORS [31] are based on averaging instantaneous flows by turning on and off stiff coefficients in legacy integrators used as black boxes. In particular, they do not require the identification of slow variables and inherit the symplecticity and reversibility of the legacy integrators that they are derived from.

5.2 Matrix exponentiation

In the case of quasi-quadratic stiff potentials, the proposed algorithm exponentiates a slowly varying matrix at each time step. When the elasticity matrix K is not diagonalizable by a constant orthogonal transformation, a numerical algebra algorithm is employed for that calculation at the expense of $\mathcal{O}(n)$ operations of d_f -by- d_f matrix multiplications per time step, where d_f is the dimension of fast variable (and hence K), and n is a preset constant that is at most $\log(\epsilon^{-1})$.

There are various approaches to exponentiate a matrix, including diagonalization, series methods, scaling and squaring, ODE solving, polynomial methods, matrix decomposition methods, and splitting, etc., as comprehensively reviewed in [25]. Many of these methods, however, differ from our approach here in that they do not guarantee that the resulting implementation of the proposed method to be symplectic as it analytically should be, unless high precision (hence slow computation) is required; most of them could not even guarantee a symplectic approximation to F_2 and F_3 .

The proposed approach (Integrator 2) obtains its efficiency by a trick of self-multiplication, which was previously used in the method of scaling and squaring [17]. However, the Padé approximation used in scaling and squaring is replaced by a symplectic and reversible approximation based on the Verlet integrator. Consequently, symplecticity and better efficiency are obtained, and accuracy is kept. Improvements by this numerical exponentiation over ‘expm’ and ‘diag,’ in terms of both accuracy and speed, are observed numerically in Sections 4.2 and 4.3.

For our alternative approach (see (31) for the general strategy and Appendix for implementational details for the specific purpose of multiscale integration), it uses the slowly varying property of the matrix to repetitively modify the exponential from the previous step by a small symplectic change to get a new exponential. Regarding updating matrix exponentials, since there are results such as [9] on relationships between

perturbed eigenvalues and perturbation in the matrix, a natural thought is to use eigenstructures that were explored in the previous step as initial conditions in iterative algorithms (such as Jacobi–Davidson for eigenvalues [29] or Rayleigh quotient for extreme eigenvalues [33]). This idea, however, did not significantly accelerate the computation as we explored in numerical experiments with an incomplete pool of methods. Other matrix decompositions methods (QR, for instance) did not gain much from previous decompositions either in our numerical investigations. Our way of exponential updating is essentially an operator splitting approach, which is analogous to our main vector field splitting strategy that yields the proposed multiscale integrator.

Acknowledgement

We sincerely thank Charles Van Loan for a stimulating discussion and Sydney Garstang for proof-reading the manuscript. We are also grateful to two anonymous referees for precise and detailed comments and suggestions.

Funding

This work was supported by the National Science Foundation [CMMI-092600].

Appendix: an Alternative Matrix Exponentiation Scheme

We will present in Integrator A.1 an alternative (symplectic) way for computing $F_{3,k}$ and $G_{2,k,i}$. This alternative is based on iteratively updating the matrix exponential from the computation at the previous step. We will first demonstrate its full version, and then provide a simple approximation which is not exactly symplectic on all variables but symplectic on the fast variables (in the sense of a symplectic submanifold) and exhibits satisfactory long-time performance in numerical experiments.

Lemma A.1. Define

$$\begin{bmatrix} \alpha(t) & \beta(t) & \gamma(t) \\ 0 & F_2(t) & G_2(t) \\ 0 & 0 & F_3(t) \end{bmatrix} := \exp \left(\begin{bmatrix} -N^T & MJ & 0 \\ 0 & -N^T & M \\ 0 & 0 & N \end{bmatrix} t \right) \quad (\text{A.1})$$

Then for any H , we have $-F_3(H)^T \gamma(H) = \int_0^H F_3^T(s) M (-JG_2(s)) ds$. \square

Proof. Differentiating (A.1) with respect to t and equating each matrix component on left- and right-hand sides, we obtain

$$\begin{aligned}
\dot{\alpha} &= -N^T \alpha \\
\dot{F}_2 &= -N^T F_2 \\
\dot{F}_3 &= N F_3 \\
\dot{\beta} &= -N^T \beta + M J F_2 \\
\dot{G}_2 &= -N^T G_2 + M F_3 \\
\dot{\gamma} &= -N^T \gamma + M J G_2
\end{aligned} \tag{A.2}$$

where the initial conditions obviously are $\alpha(0) = I, F_2(0) = I, F_3(0) = I, \beta(0) = 0, G_2(0) = 0, \gamma(0) = 0$.

Solving these inhomogeneous linear equations leads to known results including $F_2(t) = \exp(-N^T t)$, $F_3(t) = \exp(Nt)$ and $G_2(t) = \int_0^t \exp(-N^T(t-s)) M \exp(Ns) ds$, as well as new results such as

$$\gamma(t) = \int_0^t \exp(-N^T(t-s)) M J G_2(s) ds, \tag{A.3}$$

which is equivalent to

$$-F_3(H)^T \gamma(H) = \int_0^H F_3(s)^T M (-J G_2(s)) ds \tag{A.4}$$

■

Lemma A.2. If $M = M^T, F_2^T F_3 = I$ and $\partial F_3 = -J G_2$, such as those derived from N and M defined in Section 2.3, then

$$\begin{aligned}
\partial G_2(H) &= F_2(H) \left(- (F_3(H)^T \gamma(H))^T - F_3(H)^T \gamma(H) \right. \\
&\quad \left. + \int_0^H F_3(s)^T \partial M F_3(s) ds - (-J G_2(H))^T G_2(H) \right)
\end{aligned} \tag{A.5}$$

□

Proof. By Leibniz's rule

$$\partial G_2(H) = [F_3(H)^T]^{-1} (\partial (F_3(H)^T G_2(H)) - \partial F_3(H)^T G_2(H)) \tag{A.6}$$

By the definition of F_3 and G_2 , this is

$$\partial G_2(H) = F_2(H) \left(\partial \left(\int_0^H F_3(s)^T M F_3(s) ds \right) - (-J G_2(H))^T G_2(H) \right), \tag{A.7}$$

in which

$$\begin{aligned}
& \partial \left(\int_0^H F_3(s)^T M F_3(s) \, ds \right) \\
&= \int_0^H \partial F_3(s)^T M F_3(s) \, ds + \int_0^H F_3(s)^T M \partial F_3(s) \, ds + \int_0^H F_3(s)^T \partial M F_3(s) \, ds \\
&= \int_0^H (-J G_2(s))^T M F_3(s) \, ds + \int_0^H F_3(s)^T M (-J G_2(s)) \, ds + \int_0^H F_3(s)^T \partial M F_3(s) \, ds \\
&= -(F_3(H)^T \gamma(H))^T - F_3(H)^T \gamma(H) + \int_0^H F_3(s)^T \partial M F_3(s) \, ds
\end{aligned} \tag{A.8}$$

for $\gamma(H)$ defined in Lemma A.1. ■

Remark A.1. $\int_0^H F_3(s)^T \partial M F_3(s) \, ds = \tilde{F}_3(H)^T \tilde{G}_2(H)$ can be computed by again using the trick of:

$$\begin{bmatrix} \tilde{F}_2(t) & \tilde{G}_2(t) \\ 0 & \tilde{F}_3(t) \end{bmatrix} := \exp \left(\begin{bmatrix} -N^T & \partial M \\ 0 & N \end{bmatrix} t \right) \tag{A.9}$$

Of course, to get $B'_{0,i,j} = \partial_j B_{0,i}$, we use the fact that $B_{0,i} = G_{2,0,i}(H/2^n)$. □

Lemma A.3. Suppose $q_{(k+1)',i}^{\text{fast}}, p_{(k+1)',j}^{\text{fast}}, q_{(k+1)',i}^{\text{slow}}, p_{(k+1)',j}^{\text{slow}}$ are obtained from $q_k^{\text{fast}}, p_k^{\text{fast}}, q_k^{\text{slow}}, p_k^{\text{slow}}$ by Integrator 1 with $F_{3,k}$ and $G_{2,k,i}$ satisfying $F_{3,k}^T J F_{3,k} = J$ and $G_{2,k,i} = -J \frac{\partial}{\partial q_{k',i}^{\text{slow}}} F_{3,k}$, then

$$\frac{\partial q_{(k+1)',i}^{\text{slow}}}{\partial q_{k',j}^{\text{slow}}} = I + \frac{H}{2} \begin{bmatrix} q_k^{\text{fast}} \\ p_k^{\text{fast}} \end{bmatrix} \left(G_{2,k,j}(H)^T J G_{2,k,i}(H) + F_{3,k}(H)^T \frac{\partial}{\partial q_{k',j}^{\text{slow}}} G_{2,k}(H) \right) \begin{bmatrix} q_k^{\text{fast}} \\ p_k^{\text{fast}} \end{bmatrix} \tag{A.10}$$

□

Proof. Using chain rule, we have

$$\frac{\partial q_{(k+1)',i}^{\text{slow}}}{\partial q_{k',j}^{\text{slow}}} = I + \frac{H}{2} \begin{bmatrix} q_k^{\text{fast}} \\ p_k^{\text{fast}} \end{bmatrix} \left(\frac{\partial}{\partial q_{k',j}^{\text{slow}}} F_{3,k}(H)^T G_{2,k,i}(H) + F_{3,k}(H)^T \frac{\partial}{\partial q_{k',j}^{\text{slow}}} G_{2,k,i}(H) \right) \begin{bmatrix} q_k^{\text{fast}} \\ p_k^{\text{fast}} \end{bmatrix} \tag{A.11}$$

This simplifies to (A.10) because $G_{2,k,i} = -J \frac{\partial}{\partial q_{k',i}^{\text{slow}}} F_{3,k}$ and $-J^T = J$. ■

Integrator A.1. Iterative matrix exponentiation scheme (alternative to Integrator 2) that obtains $F_{3,k}$ and $G_{2,k,i}$ via symplectic updates. k is the same index as the one used in Integrator 1. $n \geq 1$ is an integer controlling the accuracy of matrix exponential approximations.

1. At the beginning of simulation, let $q_0^{\text{slow}} = q_0^{\text{slow}} + H P_0^{\text{slow}}$ and evaluate $K_0 := K(q_0^{\text{slow}})$ and $\partial_i K_0 := \frac{\partial}{\partial q_0^{\text{slow},i}} K(q_0^{\text{slow}})$ ($i = 1, \dots, d_s$). Calculate

$$\begin{bmatrix} A_0 & B_{0,i} \\ 0 & C_0 \end{bmatrix} := \exp \left(\begin{bmatrix} -N_0^T & M_{0,i} \\ 0 & N_0 \end{bmatrix} H/2^n \right)$$

by any favorite matrix exponentiation method (e.g., by the symplectic method introduced in Section 2.3), where

$$N_0 = \begin{bmatrix} 0 & I \\ -\epsilon^{-1} K_0 & 0 \end{bmatrix} \quad \text{and} \quad M_{0,i} = \begin{bmatrix} \epsilon^{-1} \partial_i K_0 & 0 \\ 0 & 0 \end{bmatrix}.$$

2. Compute $B'_{0,i,j} = \frac{\partial}{\partial q_0^{\text{slow},j}} B_{0,i}$. One inexpensive way to do so is to use Lemma A.2 with Remark A.1.
3. Start the updating loop, with the step count indicated by k starting from 1; let $q_1^{\text{slow,fast}} = q_0^{\text{slow,fast}}$, $p_1^{\text{slow,fast}} = p_0^{\text{slow,fast}}$, and $\frac{q_1^{\text{slow}}}{q_0^{\text{slow}}} = I$;
4. Carry out the $q_k, p_k \mapsto q'_k, p'_k$ half-step (in Integrator 1). Evaluate $K_k := K(q_k^{\text{slow}})$, and let

$$D_k := \begin{bmatrix} 0 & \epsilon^{-1} (K_k^T - K_{k-1}^T) H/2^n \\ 0 & 0 \end{bmatrix}.$$

Define $A_k := A_{k-1} \exp(D_k)$ and use the equality $\exp(D_k) = I + D_k$ (since D_k is nilpotent); similarly, define $C_k := C_{k-1} \exp(-D_k^T) = C_{k-1} - C_{k-1} D_k^T$;

5. Let $B_{k,i} = -J \frac{\partial C_k}{\partial q_k^{\text{slow},i}}$, which can be computed from known values using chain rule:

$$\begin{aligned} B_{k,i} &= -J \frac{\partial (C_{k-1} (I + D_k))}{\partial q_k^{\text{slow},i}} = -J \left(\sum_{j=1}^{d_s} \frac{\partial q_{(k-1)',j}^{\text{slow}}}{\partial q_k^{\text{slow},i}} \frac{\partial C_{k-1}}{\partial q_{k-1}',j} (I + D_k) + C_{k-1} \frac{\partial D_k}{\partial q_k^{\text{slow},i}} \right) \\ &= \sum_{j=1}^{d_s} \frac{\partial q_{(k-1)',j}^{\text{slow}}}{\partial q_k^{\text{slow},i}} B_{k-1,j} (I + D_k) + C_{k-1} \frac{\partial D_k}{\partial q_k^{\text{slow},i}} \end{aligned} \quad (\text{A.12})$$

To compute $\frac{\partial D_k}{\partial q_k^{\text{slow},i}}$, we need the derivatives of K_k and K_{k-1} with respect to $q_k^{\text{slow},i}$; the former is trivial, and the latter again can be computed by chain rule:

$$\frac{\partial K_{k-1}^T}{\partial q_k^{\text{slow},i}} = \sum_{j=1}^{d_s} \frac{\partial q_{(k-1)',j}^{\text{slow}}}{\partial q_k^{\text{slow},i}} \frac{\partial K_{k-1}^T}{\partial q_{k-1}',j} \quad (\text{A.13})$$

6. $B'_{k,i,j}$ can be similarly computed from $B'_{k-1,i,j}$, $B_{k-1,i}$, C_{k-1} and D_k by repetitively applying chain rule. The detail is lengthy and hence omitted.

7. Let $F_{2,k}^1 := A_k$, $G_{2,k,i}^1 := B_{k,i}$, $F_{3,k}^1 := C_k$, then repetitively apply

$$\begin{aligned} \begin{bmatrix} F_{2,k}^{j+1} & G_{2,k,i}^{j+1} \\ 0 & F_{3,k}^{j+1} \end{bmatrix} &:= \begin{bmatrix} F_{2,k}^j & G_{2,k,i}^j \\ 0 & F_{3,k}^j \end{bmatrix}^2 \\ &= \begin{bmatrix} F_{2,k}^j F_{2,k}^j & F_{2,k}^j G_{2,k,i}^j + G_{2,k,i}^j F_{3,k}^j \\ 0 & F_{3,k}^j F_{3,k}^j \end{bmatrix} \quad \text{for } j = 1, \dots, n, \end{aligned}$$

and finally define $F_{2,k} := F_{2,k}^{n+1}$, $G_{2,k,i} := G_{2,k,i}^{n+1}$, $F_{3,k} = F_{3,k}^{n+1}$.

8. Compute $\frac{\partial q_{(k+1)',i}^{\text{slow}}}{\partial q_{k',j}^{\text{slow}}}$ by using Lemma A.3, so that it could be used by Step 5 for the next k . $\frac{\partial}{\partial q_{k',j}^{\text{slow}}} G_{2,k,i}(H)$ is computed based on the following:

$$\frac{\partial}{\partial q_{k',j}^{\text{slow}}} (A_k B_{k,i} + B_{k,i} C_k) = -A_k B_{k,j}^T J A_k B_{k,i} + A_k B'_{k,i,j} + B'_{k,i,j} C_k - B_{k,i} J B_{k,j} \tag{A.14}$$

where the first term is due to $\frac{\partial A_k}{\partial q_{k',j}^{\text{slow}}} = -A_k B_{k,j}^T J A_k$, which is because $\partial A^T C + A^T \partial C = \partial(A^T C) = \partial I = 0$ and therefore $\partial A^T = -A^T \partial C C^{-1} = A^T J B C^{-1} = A^T J B A^T$. A similar trick of self multiplication applies to get the derivative of the 2^n -times product.

9. Carry out the $q_k, p_k \mapsto q_{k+1}, p_{k+1}$ half-step update of numerical integration using $F_{2,k}$, $F_{3,k}$ and $G_{2,k,i}$, and then increase k by 1 and go to Step 4 until integration time is reached. □

$F_{3,k}$ and $G_{2,k,i}$ computed in this way (Integrator A.1) will also satisfy Lemma 3.1 and render the integration symplectic on all variables. Proofs are omitted, but they are analogous to those in Section 3.2, and all structures, such as reversibility, symplecticity of F_2 and F_3 (illustrated by corresponding lemmas), and the relation between F_3 and G_2 will be preserved as long as they are satisfied by $A_0, B_{0,i}, C_0$ (i.e., the initial matrix exponentiation is accurate).

In terms of efficiency, this method only uses one single matrix exponentiation operation and then keeps on updating it. Nevertheless, it is not easy to implement, and its speed advantage is not dominant. However, if the requirement on symplecticity is not that strict and a small numerical error in the matrix exponential is allowed (recall an analogous case of the famous implicit mid-point integrator, in which implicit solves are in fact not done perfectly and continuously polluting the symplecticity), we could use the approximation of $\frac{\partial q_{(k+1)',i}^{\text{slow}}}{\partial q_{k',j}^{\text{slow}}} = I$. This will introduce a local error of $\mathcal{O}(Hn/2^n)$ in $G_{2,k,i}$ at each timestep (details omitted), but the local error in $F_{2,k}$ and $F_{3,k}$ is 0, and the method

is symplectic on the submanifold of the fast variables (although not symplectic on all variables). The approximating method is as follows.

Integrator A.2. An efficient approximation of Integrator A.1:

1. At the beginning of simulation, let $q_0^{\text{slow}} = q_0^{\text{slow}} + Hp_0^{\text{slow}}$ and evaluate $K_0 := K(q_0^{\text{slow}})$ and $\partial_i K_0 := \partial_i K(q_0^{\text{slow}})$ ($i = 1, \dots, d_s$) and calculate

$$\begin{bmatrix} A_0 & B_{0,i} \\ 0 & C_0 \end{bmatrix} := \exp \left(\begin{bmatrix} -N_0^T & M_{0,i} \\ 0 & N_0 \end{bmatrix} H/2^n \right)$$

by any favorite matrix exponentiation method, where

$$N_0 = \begin{bmatrix} 0 & I \\ -\epsilon^{-1}K_0 & 0 \end{bmatrix} \quad \text{and} \quad M_{0,i} = \begin{bmatrix} \epsilon^{-1}\partial_i K_0 & 0 \\ 0 & 0 \end{bmatrix};$$

let $q_1^{\text{slow,fast}} = q_0^{\text{slow,fast}}$ and $p_1^{\text{slow,fast}} = p_0^{\text{slow,fast}}$.

2. Start the updating loop, with the step count indicated by k starting from 1;
3. Carry out the $q_k, p_k \mapsto q_{k'}, p_{k'}$ half-step. Evaluate $K_k := K(q_k^{\text{slow}})$ and $\partial_i K_k := \partial_i K(q_k^{\text{slow}})$, let

$$D_k := \begin{bmatrix} 0 & \epsilon^{-1}(K_k^T - K_{k-1}^T)H/2^n \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad E_{k,i} := \begin{bmatrix} \epsilon^{-1}(\partial_i K_k - \partial_i K_{k-1})H/2^n & 0 \\ 0 & 0 \end{bmatrix}.$$

Define

$$\begin{bmatrix} A_k & B_{k,i} \\ 0 & C_k \end{bmatrix} := \begin{bmatrix} A_{k-1} & B_{k-1,i} \\ 0 & C_{k-1} \end{bmatrix} \times \exp \begin{bmatrix} D_k & E_{k,i} \\ 0 & -D_k^T \end{bmatrix}$$

and use the equality

$$\exp \begin{bmatrix} D_k & E_{k,i} \\ 0 & -D_k^T \end{bmatrix} = \begin{bmatrix} I + D_k & E_{k,i} \\ 0 & I - D_k^T \end{bmatrix}$$

(because $D_k E_{k,i} = 0$ and $E_{k,i} D_k^T = 0$) to evaluate $A_k = A_{k-1} + A_{k-1} D_k$, $B_{k,i} = B_{k-1,i} + A_{k-1} E_{k,i} - B_{k-1,i} D_k^T$, and $C_k = C_{k-1} - C_{k-1} D_k^T$;

4. Let $F_{2,k}^1 := A_k$, $G_{2,k,i}^1 := B_{k,i}$, $F_{3,k}^1 := C_k$, then repetitively apply

$$\begin{aligned} \begin{bmatrix} F_{2,k}^{j+1} & G_{2,k,i}^{j+1} \\ 0 & F_{3,k}^{j+1} \end{bmatrix} &:= \begin{bmatrix} F_{2,k}^j & G_{2,k,i}^j \\ 0 & F_{3,k}^j \end{bmatrix}^2 \\ &= \begin{bmatrix} F_{2,k}^j F_{2,k}^j & F_{2,k}^j G_{2,k,i}^j + G_{2,k,i}^j F_{3,k}^j \\ 0 & F_{3,k}^j F_{3,k}^j \end{bmatrix} \quad \text{for } j = 1, \dots, n, \end{aligned}$$

and finally define $F_{2,k} := F_{2,k}^{n+1}$, $G_{2,k,i} := G_{2,k,i}^{n+1}$, $F_{3,k} = F_{3,k}^{n+1}$.

5. Carry out the $q_k, p_k \mapsto q_{k+1}, p_{k+1}$ half-step update of numerical integration using $F_{2,k}$, $F_{3,k}$ and $G_{2,k,i}$, and then increase k by 1 and go to Step 3 until integration time is reached. \square

Numerical experiments presented in Section 4 are repeated using this approximating integrator. Energy preservations are as good as before, and slow trajectories show no significant deviation, suggesting no significant effect of the approximated symplecticity (detailed results omitted). This approximation, on the other hand, allows a choice of an even smaller n , such as $n=5$ for the previous examples, which results in a further speed-up.

References

- [1] Ariel, G., B. Engquist, and Y.-H. R. Tsai. "A multiscale method for highly oscillatory ordinary differential equations with resonance." *Mathematics of Computation* 78 (2009): 929.
- [2] Ariel, G., B. Engquist, and R. Tsai. "A reversible multiscale integration method." *Communications in Mathematical Sciences* 7, no. 3 (2009): 595–610.
- [3] Calvo, M. P. and J. M. Sanz-Serna. "Instabilities and inaccuracies in the integration of highly oscillatory problems." *SIAM Journal on Scientific Computing* 31, no. 3 (2009): 1653–77.
- [4] Calvo, M. P. and J. M. Sanz-Serna. "Heterogeneous multiscale methods for mechanical systems with vibrations." *SIAM Journal on Scientific Computing* 32, no. 4 (2010): 2029–46.
- [5] Cohen, D., E. Hairer, and C. Lubich. "Modulated fourier expansions of highly oscillatory differential equations." *Foundations of Computational Mathematics* 3 (2003): 327–45. 10.1007/s10208-002-0062-x.
- [6] Coppersmith, D. and S. Winograd. "Matrix multiplication via arithmetic progressions." *Journal of Symbolic Computation* 9, no. 3 (1990): 251–80. Computational algebraic complexity editorial.
- [7] Dobson, M., C. Le Bris, and F. Legoll. "Symplectic schemes for highly oscillatory Hamiltonian systems: the homogenization approach beyond the constant frequency case." (2010): preprint. arXiv:1008.1030.
- [8] E, W., W. Ren, and E. Vanden-Eijnden. "A general strategy for designing seamless multiscale methods." *Journal of Computational Physics* 228 (2009): 5437–53.
- [9] Eisenstat, S. C. and I. C. F. Ipsen. "Relative perturbation techniques for singular value problems." *SIAM Journal on Numerical Analysis* 32 (1995): 1972.
- [10] Engquist, W. E. B., X. Li, W. Ren, and E. Vanden-Eijnden. "Heterogeneous multiscale methods: a review." *Communications in Computational Physics* 2, no. 3 (2007): 367–450.
- [11] Engquist, B. and Y.-H. R. Tsai. "Heterogeneous multiscale methods for stiff ordinary differential equations." *Mathematics of Computation* 74, no. 252 (2005): 1707–42 (electronic).
- [12] García-Archilla, B., J. M. Sanz-Serna, and R. D. Skeel. "Long-time-step methods for oscillatory differential equations." *SIAM Journal on Scientific Computing* 20, no. 3 (1999): 930–63.

- [13] Givon, D., I. G. Kevrekidis, and R. Kupferman. "Strong convergence of projective integration schemes for singularly perturbed stochastic differential systems." *Communications in Mathematical Sciences* 4, no. 4 (2006): 707–29.
- [14] Grimm, V. and M. Hochbruck. "Error analysis of exponential integrators for oscillatory second-order differential equations." *Journal of Physics A—Mathematical and General* 39 (2006): 5495–507.
- [15] Grubmuller, H., H. Heller, A. Windemuth, and K. Schulten. "Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions." *Molecular Simulation* 6 (1991): 121–42.
- [16] Hairer, E., C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd edn. Heidelberg, Germany: Springer, 2004.
- [17] Higham, N. J. "The scaling and squaring method for the matrix exponential revisited." *SIAM Journal on Matrix Analysis and Applications* 26 (2005): 1179–93.
- [18] Hochbruck, M. and C. Lubich. "A gautschi-type method for oscillatory second-order differential equations." *Numerische Mathematik* 83 (1999): 403–26.
- [19] Kevrekidis, I. G., C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. "Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis." *Communications in Mathematical Sciences* 1, no. 4 (2003): 715–62.
- [20] Kevrekidis, I. and G. Samaey. "Equation-free multiscale computation: algorithms and applications." *Annual Review of Physical Chemistry* 60, no. 1 (2009): 321–44.
- [21] Lax, P. D. and R. D. Richtmyer. "Survey of the stability of linear finite difference equations." *Communications on Pure and Applied Mathematics* 9 (1956): 267–93.
- [22] Le Bris, C. and F. Legoll. "Integrators for highly oscillatory hamiltonian systems: an homogenization approach." *Discrete and Continuous Dynamical Systems. Series B* 13 (2010): 347–73.
- [23] Leimkuhler, B. and S. Reich. "A reversible averaging integrator for multiple time-scale dynamics." *Journal of Computational Physics* 171 (2001): 95–114.
- [24] Li, T., A. Abdulle, and W. E. "Effectiveness of implicit methods for stiff stochastic differential equations." *Communications in Computational Physics* 3, no. 2 (2008): 295–307.
- [25] Moler, C. and C. Van Loan. "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later." *SIAM Review* 45 (2003): 3–49.
- [26] Neri, F. "Lie algebras and canonical integration." Technical Report, Department of Physics, University of Maryland, 1988.
- [27] Sanz-Serna, J. M. "Mollified impulse methods for highly oscillatory differential equations." *SIAM Journal on Numerical Analysis* 46, no. 2 (2008): 1040–59.
- [28] Sanz-Serna, J. M., G. Ariel, and Y.-H. R. Tsai. "Multiscale methods for stiff and constrained mechanical systems." Preprint, 2009.
- [29] Sleijpen, G. L. G. and H. A. Van der Vorsty. "A Jacobi–Davidson iteration method for linear eigenvalue problems." *SIAM Review* 42 (2000): 267–93.

- [30] Stern, A. and E. Grinspun. "Implicit–explicit variational integration of highly oscillatory problems." *Multiscale Modeling and Simulation* 7 (2009): 1779–94.
- [31] Tao, M., H. Owhadi, and J. E. Marsden. "Non-intrusive and structure preserving multiscale integration of stiff ODEs, SDEs and Hamiltonian systems with hidden slow dynamics via flow averaging." *Multiscale Modeling and Simulation* 8 (2010): 1269–324.
- [32] Tao, M., H. Owhadi, and J. E. Marsden. "Structure preserving Stochastic Impulse Method for stiff langevin systems with a uniform global error of order 1 or 1/2 on position." (2010): preprint. arXiv:1006.4657.
- [33] Trefethen, L. N. and D. Bau III. *Numerical Linear Algebra*. Philadelphia: Society for Industrial and Applied Mathematics, 1997.
- [34] Trotter, H. F. "Product of semigroups of operators." *Proceedings of American Mathematical Society* 10 (1959): 545–51.
- [35] Tuckerman, M., B. J. Berne, and G. J. Martyna. "Reversible multiple time scale molecular dynamics." *Journal of Chemical Physics* 97 (1992): 1990–2001.
- [36] Van Loan, C. "Computing integrals involving the matrix exponential." *IEEE Transactions on Automatic Control* 23, no. 3 (1978): 395–404.
- [37] Zhang, G. and T. Schlick. "LIN: A new algorithm to simulate the dynamics of biomolecules by combining implicit-integration and normal mode techniques." *Journal of Computational Chemistry* 14 (1993): 1212–33.