Title
Geometric Integrators for Stiff Systems, Lie Groups and Control Systems

Author
Shen, Xuefeng

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Geometric Integrators for Stiff Systems, Lie Groups and Control Systems

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy

in

Mathematics with a Specialization in Computational Science

by

Xuefeng Shen

Committee in charge:

Professor Melvin Leok, Chair
Professor Thomas R. Bewley
Professor Li-Tien Cheng
Professor Petr Krysl
Professor Bo Li

2019

The Dissertation of Xuefeng Shen is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

DEDICATION

To My Father and Mother.

EPIGRAPH

Knowledge as Action.

—YangMing Wang

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank the warm welcome of the University of California San Diego, the patient and high standard of guidance of my advisor Professor Melvin Leok. He showed me an exciting and fruitful research area, and his rigorous attitude and encouragement of new ideas helped me build nice taste and correct understanding of research. Without his help, this work would not be possible.

Besides that, I want to thank my parents for their patience to my pursuit of academic achievements, their support and care accompanied me for all these years.

Last but not the least, I would like to thank the sunshine and beautiful beaches of San Diego, they alleviated my pressure in the hardest time; thanks to my friends in the math department for their support and company; thanks to my primary care doctor and the staff at the department front desk for their support in my life and TA work.

Chapter 2, in full, is a reprint of the material that has been accepted for publication by Journal of Computational Physics, 2019. Xuefeng Shen; Melvin Leok, Elsevier Press, 2019. The dissertation author was the primary investigator and author of this material.

Chapter 3, in full, is currently being prepared for submission for publication of the material. Xuefeng Shen; Melvin Leok. The dissertation author was the primary investigator and author of this material.

Chapter 4, in full, is a reprint of the material that has been submitted for publication to Journal of Computational Dynamics, 2018. Xuefeng Shen; Melvin Leok, American Institute of Mathematical Sciences , 2018. The dissertation author was the primary investigator and author of this material.

Chapter 5, in full, is a reprint of the material that has been submitted for publication to International Journal of Robotics Research, 2019. Xuefeng Shen; Melvin Leok, SAGE Publishing , 2019. The dissertation author was the primary investigator and author of this material.

VITA

2011	B. S. in Mathematics , University of Science and Technology of China, Hefei
2014	M. S. in Computational Mathematics, Chinese Academy of Sciences, Beijing
2019	Doctor of Philosophy in Mathematics with a Specialization in Computational Science, University of California San Diego

PUBLICATIONS

Xuefeng Shen, Melvin Leok, "Geometric Exponential Integrators," *Journal of Computational Physics*, accepted
Xuefeng Shen, Melvin Leok, "Lie group variational integrators for rigid body problems using quaternions"
Xuefeng Shen, Melvin Leok, "High-Order Symplectic Lie Group Methods on $SO(n)$", submitted to *Journal of Computational Dynamics*
Xuefeng Shen, Melvin Leok, "Geometric Symmetry Reduction of the Nonobservable Subspace for Kalman Filtering", submitted to *International Journal of Robotics Research*

ABSTRACT OF THE DISSERTATION

Geometric Integrators for Stiff Systems, Lie Groups
and Control Systems

by

Xuefeng Shen

Doctor of Philosophy in Mathematics with a Specialization in Computational Science

University of California San Diego, 2019

Professor Melvin Leok, Chair

The main idea of a geometric integrator is to adopt a geometric viewpoint of the problem and to construct integrators that preserve the geometric properties of the continuous dynamical system. For classical mechanics, both the Lagrangian and the Hamiltonian formulations can be described using the language of geometry. Due to the rich conservation properties of mechanics, it is natural to study the construction of numerical integrators that preserve some geometric properties, such as the symplectic structure, energy, and momentum maps. Such geometric structure-preserving numerical integrators exhibit nice properties compared to traditional numerical methods. This is especially true in galaxy simulations and molecular dynamics, where

long time simulations are required to answer the corresponding scientific questions. Variational integrators have attracted interest in the geometric integration community as it discretizes Hamilton's principle, as opposed to the corresponding differential equation, to obtain a numerical integrator that is automatically symplectic, and which exhibits a discrete Noether's theorem. Besides classical mechanics, such an approach has also been applied to other fields, such as optimal control [31, 40], partial differential equations [44], stochastic differential equations [9], and so on.

In this thesis, we consider generalizations of geometric integrators that are adapted to three special settings. One is the case of stiff systems of the form, $\dot{q} = Aq + f(q)$, where the coefficient matrix A has a large spectral radius that is responsible for the stiffness of the system, while the nonlinear term $f(q)$ is relatively smooth. Traditionally, exponential integrators have been used to address the issue of stiffness. In Chapter 2, we consider a special semilinear problem with $A = JD$, $f(q) = J\nabla V(q)$, where $J^T = -J, D^T = D$, and $JD = DJ$. Then, the system is described by $\dot{q} = J(Dq + \nabla V(q))$, which naturally arises from the discretization of Hamiltonian partial differential equations. It is a constant Poisson system with Poisson structure $J_{ij}\frac{\partial}{\partial x_i} \otimes \frac{\partial}{\partial x_j}$, and Hamiltonian $H(q) = \frac{1}{2}q^T Dq + V(q)$. Two types of exponential integrators are constructed, one preserves the Poisson structure, and the other preserves energy. Numerical experiments for semilinear Possion systems obtained by semi-discretizing Hamiltonian PDEs are presented. These geometric exponential integrators exhibit better long time stability properties as compared to non-geometric integrators, and are computationally more efficient than traditional symplectic integrators and energy-preserving methods based on the discrete gradient method.

The other generalization is to Lie groups. When configuration manifold is a Lie group, we would like to utilize the group structure rather than simply regard it as embedded submanifold. This is particularly useful when codimension of the embedding is large. For the rigid body problem, the configuration space is $\mathbb{R}^3 \rtimes SO(3)$, which is a Lie group. Lee et al. [35] were the first to directly use the Lie group structure of the rotation group to construct a Lie group variational integrator. In contrast, most prior approaches used the unit quaternion representation

of the rotation group and applied symplectic integrators for constrained systems with the unit length constraint. In Chapter 3, we adopt the approach used in constructing Lie group variational integrators for rigid body dynamics on the rotation group and applied it to the unit quaternion representation. A Lie group variational integrator in the unit quaternion representation is derived, and it can be shown that our method is related to the RATTLE method applied to the rotation representation by the projection from unit quaternions to rotation matrices. The numerical results for our Lie group quaternion variational integrator are presented. The integrators constructed in Chapter 3 are only second-order, and in Chapter 4, variational integrators of arbitrarily high-order on special orthogonal group $SO(n)$ are constructed by using the polar decomposition. It avoids the second-order derivative of the exponential map that arises in the traditional Lie group variational integrator method. Also, a reduced Lie–Poisson integrator is constructed. The resulting algorithms can naturally be implemented using fixed-point iteration. Numerical results are given for the case of $SO(3)$.

The last generalization is to control systems. We studied the problem of uncertainty propagation and measurement update for systems that are partially unobservable. We construct a method that satisfies the chain property that the unobservable subspace remains perpendicular to the measurement dh during propagation. We characterize the unobservable subspace in terms of the group-invariance of the control system, and obtain a reduced control system on the observable variables. By decomposing the system explicitly into unobservable and observable parts (x_N, x_O), the chain property can be naturally satisfied. Also, we propose a reduced Bayesian framework, where the update from the measurement is only applied to the observable variables x_O. In Chapter 5, we consider a planar robot model, which has one odometry sensor and one camera. Odometry is used for propagation and the camera is used for measurement. In this model, the two-dimensional position as well as the orientation are all unobservable. We applied our technique to this model and performed numerical simulations. We tested this on straight line, circle, and general trajectories and found that the reduced Kalman filter that we proposed outperforms the classical Kalman filter and modifications that were proposed in the literature. In

particular, it estimates the angle quite well, and as a result, yields a better estimate of the position as well.

Chapter 1

Introduction and Background

1.1 Geometric setting for classical mechanics

1.1.1 Lagrangian mechanics and Hamiltonian mechanics

There are three equivalent descriptions of classical mechanics:

1. Newton's second law of motion,

$$F = ma;$$

2. Lagrangian mechanics,

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) = 0; \tag{1.1}$$

3. Hamiltonian mechanics,

$$\begin{cases} \dot{q} = \dfrac{\partial H}{\partial p}(q,p), \\ \dot{p} = -\dfrac{\partial H}{\partial q}(q,p). \end{cases} \tag{1.2}$$

It turns out that the notion of a manifold is a natural setting for describing the configuration spaces for Lagrangian mechanics and Hamiltonian mechanics, especially in the case of problems with constraints. Consider a planar pendulum, its trajectory is constrained to lie on a circle, which is a one-dimensional manifold; Two balls connected by a light stick of length l are constrained to be a fixed distance apart, so their positions are not independent. Thus, the configuration space is $\{x_1, x_2 \in \mathbb{R}^3 | \ |x_1 - x_2| = l\}$, which is a five-dimensional submanifold of $\mathbb{R}^6$. For the

rigid body problem, the configuration of the rigid body is uniquely determined by a translation and rotation, thus the configuration space is the Euclidean group $SE(3) = \mathbb{R}^3 \rtimes SO(3)$, which a six-dimensional manifold, and in particular, a Lie group. Using the language of manifolds, generalized coordinates becomes coordinate charts on a manifold, and generalized velocities becomes tangent vectors, while generalized momenta can be represented by cotangent vectors. The intrinsic viewpoint of dynamics on a manifold allows us to focus on the global space and global properties of flow maps, independent of the choice of coordinates, which is essential in deriving symplectic integrators for nonlinear configuration spaces.

In the geometric setting, a Lagrangian mechanical system is denoted by (TQ, L), where Q is the configuration manifold, TQ is the tangent bundle, which is usually called the state space, and $L : TQ \to \mathbb{R}$ is the Lagrangian of the system. In physics, the Lagrangian L is given by the difference of the kinetic energy T and the potential energy V, i.e., $L = T - V$. The mechanics on (TQ, L) is described by the principle of least action (or Hamilton's principle), which states that

$$\delta \int_{t_0}^{t_1} L(q(t), \dot{q}(t))dt = 0,$$

for variations with fixed endpoints $q(t_0) = q_0, q(t_1) = q_1$. Given a local coordinate chart, Hamilton's principle yield the Euler–Lagrange equations (1.1). This variational principle serves as the basis for constructing variational integrators.

In contrast, a Hamiltonian mechanical system is denoted by (T^*Q, H), where T^*Q is the cotangent bundle, which is usually called the phase space, and $H : T^*Q \to \mathbb{R}$ is the Hamiltonian of the system. In the case of quadratic kinetic energies, the Hamiltonian H is the sum of the kinetic energy and the potential energy, i.e., $H = T + V$. The cotangent bundle T^*Q possesses a canonical symplectic structure ω, which is given by

$$\omega = \sum_i dq^i \wedge dp_i, \tag{1.3}$$

in local coordinates $(q^1, \ldots, q^n, p_1, \ldots p_n)$ on T^*Q. Given a symplectic structure ω, and a Hamiltonian H, there is an associated Hamiltonian vector field X_H on T^*Q, that is defined by,

$$i_{X_H}\omega = dH.$$

The Hamiltonian vector field X_H defines a well-posed first-order differential equation on T^*Q, and yields a flow map $\phi_t : T^*Q \to T^*Q$. Given a Darboux coordinate chart, where the symplectic structure has the form given in (1.3), then the Hamiltonian vector field X_H gives Hamilton's equations (1.2). The flow map ϕ_t of the Hamiltonian vector field X_H has a very special geometric property, that is, it preserves the symplectic structure ω, which can be elegantly verified by applying Cartan's magic formula,

$$\begin{aligned} \pounds_{X_H}\omega &= i_{X_H}(d\omega) + d(i_{X_H}\omega), \\ &= i_{X_H}(0) + d^2H, \\ &= 0. \end{aligned}$$

Any one-step integrator applied to X_H with timestep h can be regarded as an approximation of the flow map ϕ_h. It is desirable to construct integrators that preserve the symplectic structure, and such methods are called symplectic integrators.

Lagrangian mechanics and Hamiltonian mechanics are connected by the Legendre transform,

$$(TQ, L) \underset{\mathbb{F}H}{\overset{\mathbb{F}L}{\rightleftarrows}} (T^*Q, H), \tag{1.4}$$

where $\mathbb{F}L$ and $\mathbb{F}H$ are defined as

$$\langle \mathbb{F}L(q,\dot{q}), \delta\dot{q}\rangle = \left.\frac{d}{dt}\right|_{t=0} L(q,\dot{q}+t\delta\dot{q}),$$
$$\langle \mathbb{F}H(q,p), \delta p\rangle = \left.\frac{d}{dt}\right|_{t=0} H(q,p+t\delta p).$$

In local coordinates, $\mathbb{F}L(q,\dot{q}) = (q, \frac{\partial L}{\partial \dot{q}})$, and $\mathbb{F}H(q,p) = (q, \frac{\partial H}{\partial p})$. The Lagrangian L and Hamiltonian H are connected by

$$H(q,p) = p\cdot\dot{q} - L(q,\dot{q})|_{(q,p)=\mathbb{F}L(q,\dot{q})},$$
$$L(q,\dot{q}) = p\cdot\dot{q} - H(q,p)|_{(q,\dot{q})=\mathbb{F}H(q,p)}.$$

When L is hyperregular, i.e., $\mathbb{F}L$ is a global diffeomorphism between TQ and T^*Q, then Lagrangian mechanics (TQ,L) and Hamiltonian mechanics (T^*Q,H) are equivalent, and the Euler–Lagrange equations (1.1) and Hamilton's equations (1.2) are related by the Legendre transforms. We can transfer everything from phase space T^*Q to state space TQ, the pullback of the Hamiltonian H to TQ yields the energy function,

$$E_L(q,\dot{q}) = \frac{\partial L}{\partial \dot{q}}\cdot\dot{q} - L(q,\dot{q}),$$

which is preserved by Euler–Lagrange flow. The pullback of the canonical symplectic structure ω to TQ yields

$$\omega_L = \mathbb{F}L^*\omega = \sum_{i,j} \frac{\partial^2 L}{\partial \dot{q}^i \partial q^j} dq^i \wedge dq^j + \frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j} dq^i \wedge d\dot{q}^j,$$

which is also preserved by the flow of the Euler–Lagrange equation.

1.1.2 First integrals and Noether's theorem

In classical mechanics, consider for example, the N-body problem of N point masses moving under mutual gravitational interaction, has a Hamiltonian that is given by

$$H(q_1, q_2 \dots q_N, p_1, p_2 \dots p_N) = \sum_{i=1}^{N} \frac{|p_i|^2}{2m_i} + \sum_{i<j} \varphi_{ij}(|q_i - q_j|), \tag{1.5}$$

where φ represents the gravitational potential. There are many interesting physical quantities that are conserved by flow, including the energy, linear momentum, and angular momentum. Studying the conserved quantities or first integrals of the system is very important as the constrain the dynamics of the system. For any function F, G defined on T^*Q, we define the Poisson bracket as follows,

$$\{F, G\} = \omega(X_F, X_G).$$

This is a bilinear map $\{\cdot,\cdot\} : C^\infty(T^*Q) \times C^\infty(T^*Q) \to C^\infty(T^*Q)$ that satisfies the following properties,

1. Leibniz rule $\quad \{FG, H\} = F\{G, H\} + \{F, H\}G;$
2. Antisymmetry $\quad \{F, G\} = -\{G, F\};$
3. Jacobi identity $\quad \{\{F, G\}, H\} + \{\{G, H\}, F\} + \{\{H, F\}, G\} = 0.$

Later, we will see that this defines a Poisson structure on T^*Q. By the definition of the Hamiltonian vector field,

$$\{F, H\} = \omega(X_F, X_H) = \langle dF, X_H \rangle = X_H(F),$$

from which we obtain the following equation describing the evolution of a scalar function F evaluated along a trajectory of the Hamiltonian vector field X_H,

$$\frac{dF}{dt} = \{F, H\}. \tag{1.6}$$

This implies that a physical quantity is a first integral if and only if its Poisson bracket with the Hamiltonian H vanishes. By combining this with the Jacobi identity property, we can prove Poisson's theorem, which states that,

Theorem 1. *If F, G are first integrals, then their Poisson bracket $\{F, G\}$ is also a first integral.*

Another way to identify first integrals is by Noether's theorem, which relates symmetries to conservation laws. For the N-body problem (1.5), the system is invariant under translations, which leads to the conservation of linear momentum. Similarly, the system is invariant under rotations, which leads to the conservation of angular momentum. More precisely, Noether's theorem states that,

Theorem 2. *Given a Lie group G that acts on the configuration space Q, there is a momentum map $J : T^*Q \to \mathfrak{g}^*$, which is given by*

$$\langle J(\alpha_q), \xi \rangle = \langle \xi_Q(q), \alpha_q \rangle.$$

If the Hamiltonian H is invariant under the cotangent lift of the action of G on Q, then

$$\pounds_{X_H} J = 0.$$

In the N-body problem, (1.5) is invariant under the translation and rotation groups, thus the linear momentum and angular momentum are preserved.

1.1.3 Euler–Poincaré reduction and Lie–Poisson reduction

When the configuration space is a Lie group G, the state space TG and phase space T^*G are diffeomorphic to $G \times \mathfrak{g}$ and $G \times \mathfrak{g}^*$, respectively, by left-trivialization. First, this is because left-trivialized velocity and momentum arise naturally in problems such as rigid body dynamics; second, this provides a global product structure for the state and phase spaces that simplifies

the representation and calculations. For the left-trivialized Lagrangian $L : G \times \mathfrak{g} \to \mathbb{R}$, applying Hamilton's principle yields the left-trivialized Euler–Lagrange equation,

$$g^{-1}\frac{\partial L}{\partial g} - \frac{d}{dt}\left(\frac{\partial L}{\partial \xi}\right) + \mathrm{ad}^*_\xi\left(\frac{\partial L}{\partial \xi}\right) = 0. \tag{1.7}$$

If the Lagrangian L is G-invariant, i.e., $L(g,\xi) = L(e,\xi)$, then we can define a reduced Lagrangian $l(\xi) = L(e,\xi)$, and (1.7) reduces to

$$\frac{d}{dt}\left(\frac{\partial l}{\partial \xi}\right) = \mathrm{ad}^*_\xi\left(\frac{\partial l}{\partial \xi}\right), \tag{1.8}$$

which evolves on the Lie algebra $\mathfrak{g}$. This kind of symmetry reduction is called Euler–Poincaré reduction, and (1.8) is the Euler–Poincaré equation. Equation (1.8) together with the obvious reconstruction equation $\dot{g} = g\xi$, describe the dynamics of the left-invariant Lagrangian system (TG, l):

$$\begin{cases} \dfrac{d}{dt}\left(\dfrac{\partial l}{\partial \xi}\right) = \mathrm{ad}^*_\xi\left(\dfrac{\partial l}{\partial \xi}\right), \\ \qquad \dot{g} = g\xi. \end{cases} \tag{1.9}$$

The Legendre transform (1.4) can be specialized to the left-trivialized case,

$$(G \times \mathfrak{g}, L) \underset{\mathbb{F}H}{\overset{\mathbb{F}L}{\rightleftarrows}} (G \times \mathfrak{g}^*, H),$$

where $\mathbb{F}L$ and $\mathbb{F}H$ are expressed in terms of the usual partial derivatives on $\mathfrak{g}$ and $\mathfrak{g}^*$, and the following relations hold,

$$\frac{\partial L}{\partial \xi}(g,\xi) = \mu, \quad \frac{\partial H}{\partial \mu}(g,\mu) = \xi, \quad \frac{\partial L}{\partial g}(g,\xi) = -\frac{\partial H}{\partial g}(g,\mu). \tag{1.10}$$

By (1.10), the Euler–Poincaré equations become

$$\frac{d\mu}{dt} = \mathrm{ad}^*_{\frac{\partial h}{\partial \mu}}(\mu), \tag{1.11}$$

which describes the evolution on the dual of the Lie algebra $\mathfrak{g}^*$ for the left-invariant Hamiltonian system (T^*G, h), where $h(\mu) = H(e, \mu)$. This kind of symmetry reduction is called Lie–Poisson reduction, and (1.11) is called the Lie–Poisson equation. And the reconstruction equation becomes $\dot{g} = g\frac{\partial h}{\partial \mu}$. Together, they describe the dynamics of the left-invariant Hamiltonian system (T^*G, h):

$$\begin{cases} \dfrac{d\mu}{dt} = ad^*_{\frac{\partial h}{\partial \mu}}(\mu), \\ \dot{g} = g\dfrac{\partial h}{\partial \mu}. \end{cases} \tag{1.12}$$

1.1.4 Hamiltonian mechanics on symplectic manifolds and Poisson manifolds

Hamiltonian mechanics can be generalized from the cotangent bundle T^*Q to any symplectic manifold. By definition, a symplectic manifold is a smooth manifold M equipped with a nonsingular, closed two-form ω. Together with a Hamiltonian H, the triple (M, ω, H) describe a general Hamiltonian system on a symplectic manifold. Similar to canonical case, we can define a Hamiltonian vector field X_H by requiring that

$$i_{X_H}\omega = dH,$$

and show that the flow of X_H preserves the symplectic structure ω. We can also define a Poisson bracket $\{\cdot,\cdot\} : C^\infty(M) \times C^\infty(M) \to C^\infty(M)$ that describes the evolution of a scalar function along a trajectory of the Hamiltonian vector field (1.6), and obtain Poisson's theorem. Examples

include the Lotka–Volterra model [53],

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & uv \\ -uv & 0 \end{pmatrix} \nabla H(u,v),$$

where $H(u,v) = u - \log u + v - 2\log v$; the generalized harmonic oscillator,

$$\begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} 0 & -(1+p^2+q^2)^2 \\ (1+p^2+q^2)^2 & 0 \end{pmatrix} \nabla H(p,q),$$

where $H(p,q) = p^2/2 - \cos(q)$; and the Ablowitz–Ladik model [1],

$$\begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} 0 & D \\ -D & 0 \end{pmatrix} \nabla H(p,q),$$

which comes from a finite-difference approximation of the nonlinear Schrödinger equation, where $D = \mathrm{diag}(-U_n, \ldots, U_n)$, $U_l = 1 + p_l^2 + q_l^2$, $H(p,q) = \sum_{l=-n}^{n}(p_l p_{l+1} + q_l q_{l+1})$. The problem of constructing a numerical integrator for noncanonical Hamiltonian system that preserves the noncanonical symplectic structure ω is still an open problem. The noncanonical case is more complicated because the canonical symplectic structure is constant with respect to canonical coordinates on T^*Q, whereas on a general symplectic manifold, the symplectic structure is typically nonconstant.

Poisson manifolds are a generalization of symplectic manifolds. A Poisson manifold $(M, \{\cdot,\cdot\})$ is a smooth manifold M equipped with a Poisson bracket $\{\cdot,\cdot\} : C^\infty(M) \times C^\infty(M) \to C^\infty(M)$ which is bilinear, antisymmetric, and satisfies the Leibniz rule and Jacobi identity. The Poisson bracket is equivalent to the existence of a Poisson structure, which a $(2,0)$-tensor

$\tau = J^{ij}\frac{\partial}{\partial x^i} \otimes \frac{\partial}{\partial x^j}$, whose coefficients satisfy

$$\begin{cases} J^{ij}(x) + J^{ji}(x) = 0, \\ \dfrac{\partial J^{ij}}{\partial x^l} J^{lk} + \dfrac{\partial J^{jk}}{\partial x^l} J^{li} + \dfrac{\partial J^{ki}}{\partial x^l} J^{lj} = 0. \end{cases} \tag{1.13}$$

For any Hamiltonian H on M, we can define the corresponding Hamiltonian vector field $X_H = \tau(dH)$, which has the local coordinate representation,

$$\dot{x} = J(x)\nabla H(x).$$

Similar to the case of (M, ω, H), the flow map preserves the Hamiltonian H and the Poisson structure τ. The simplest Poisson structure is constant, i.e. $J(x) \equiv J$, then we get a constant Poisson system

$$\dot{x} = J\nabla H(x).$$

A more complicated Poisson structure is linear, defined by $\tau = (C_{ij}^k x_k)\frac{\partial}{\partial x^i} \otimes \frac{\partial}{\partial x^j}$, where C_{ij}^k are the structure constants from a Lie algebra. It is easy to verify that the linear coefficients $C_{ij}^k x_k$ satisfy (1.13) due to the properties of structure constants, and therefore define a Poisson structure. This linear Poisson system also has a nice geometric representation, defined on the dual of the Lie algebra $\mathfrak{g}^*$, with Poisson bracket,

$$\{f, g\}(\mu) = \langle \mu, [f'(\mu), g'(\mu)] \rangle,$$

which is referred to as a Lie–Poisson bracket.

1.2 Störmer–Verlet, SHAKE and RATTLE

1.2.1 Geometric integrators

As we have seen in Hamiltonian mechanics on (T^*Q, H), noncanonical Hamiltonian mechanics on (M, ω, H), or more generally Poisson systems on $(M, \{\cdot,\cdot\}, H)$, the exact flow map ϕ_t satisfies some special properties. Our numerical integrator ψ_h can be regarded as an approximation of ϕ_t, and it is desirable for it to inherit these properties, and we refer to numerical integrators that preserve such geometric properties as geometric integrators. Let us summarize some of these geometric properties:

1. ϕ_t preserves the symplectic structure ω, or the Poisson structure τ;
2. ϕ_t preserves first integrals, such as energy, momentum and so on;
3. ϕ_t forms a group with respect to the index t;
4. ϕ_t is time-reversible for reflection symmetric Hamiltonians, $H(q,p) = H(q,-p)$.

Property 1 is one of the most important properties that we will construct numerical integrators for, and we refer to such integrators as symplectic integrators. For property 3, we mean that flow map ϕ_t satisfies

$$\phi_t \circ \phi_s = \phi_{t+s},\ \phi_0 = \mathrm{id},\ \phi_{-t} = \phi_t^{-1},$$

i.e., $\{\phi_t\}_{t\in\mathbb{R}}$ forms a commutative group. Precisely maintaining the group property is difficult to achieve in a numerical integrator, and an integrator that satisfies the condition $\psi_{-h} = \psi_h^{-1}$ is called a symmetric integrator. This can be constructed by composing any integrator ψ_h with its adjoint $\psi_h^* = \psi_{-h}^{-1}$, and for such methods, it can be shown that (see page 86 of [37]),

Theorem 3. *The order of a symmetric method is necessarily even.*

Reflection symmetry, $H(q,p) = H(q,-p)$, which arises in property 4, holds for almost all practical mechanical system, and it essentially means that the resulting Hamiltonian system

$$\begin{cases} \dot{q} = \dfrac{\partial H}{\partial p}(q,p), \\ \dot{p} = -\dfrac{\partial H}{\partial q}(q,p), \end{cases}$$

is time-reversible: if $(q(t), p(t))$ is a solution of Hamilton's equations, then so is $(q(-t), -p(-t))$. Let $z = (q,p)^{\mathrm{T}}$, $S = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}$, then ϕ_t satisfies the time-reversibility property if and only if

$$\phi_t(z) = S\phi_{-t}(Sz).$$

Similarly, a numerical integrator ψ_h is called time-reversible if and only if it satisfies

$$\psi_h(z) = S\psi_{-h}(Sz). \tag{1.14}$$

If ψ_h is also symmetric, then (1.14) is equivalent to

$$\psi_h(z) = S\psi_h^{-1}(Sz). \tag{1.15}$$

1.2.2 Störmer–Verlet method

We consider Newton's second law with potential energy $V(q)$,

$$\ddot{q} = -\nabla V(q). \tag{1.16}$$

This is equivalent to the Hamiltonian equation

$$\begin{cases} \dot{q} = p, \\ \dot{p} = -\nabla V(q), \end{cases} \tag{1.17}$$

where $H(q,p) = \frac{|p|^2}{2} + V(q)$. For (1.17), we have two very simple first-order symplectic integrators, the Euler-B method

$$\begin{cases} p_{n+1} = p_n - h \cdot \nabla V(q_n), \\ q_{n+1} = q_n + h \cdot p_{n+1}, \end{cases} \tag{1.18}$$

and the Euler-A method,

$$\begin{cases} q_{n+1} = q_n + h \cdot p_n, \\ p_{n+1} = p_n - h \cdot \nabla V(q_{n+1}). \end{cases} \tag{1.19}$$

It is easy to verify that (1.18) and (1.19) both satisfy $dq_{n+1} \wedge dp_{n+1} = dq_n \wedge dp_n$, i.e., they are symplectic, and that they are adjoints of each other. The composition of (1.18) and (1.19), $\text{Euler}_A\left(\frac{h}{2}\right) \circ \text{Euler}_B\left(\frac{h}{2}\right)$, yields the Störmer–Verlet method,

$$\begin{cases} p_{n+\frac{1}{2}} = p_n - \dfrac{h}{2} \cdot \nabla V(q_n), \\ q_{n+1} = q_n + h \cdot p_{n+\frac{1}{2}}, \\ p_{n+1} = p_{n+\frac{1}{2}} - \dfrac{h}{2} \cdot \nabla V(q_{n+1}). \end{cases} \tag{1.20}$$

Since (1.20) is the composition of two adjoint symplectic methods, it is both symmetric and symplectic. It also satisfies (1.15), and is therefore time-reversible. By Theorem 3, we can see that (1.20) has at least order 2. It can be shown that Euler-B, Euler-A and Störmer–Verlet method all preserve linear first integrals (as do all Runge–Kutta methods), and quadratic first integrals of the form $I(q,p) = p^T Bq$. The Störmer–Verlet method (also known as the leapfrog method) is probably the most widely used low-order symplectic algorithm in practice; numerical experiments and theoretical analysis demonstrate that symplectic integrators have long time stability, and are particularly suitable for the long time simulation of Hamiltonian systems. A typical example is galaxy simulations, which is a N-body problem that is simulated over long timescales to study galaxy formation, and this makes symplectic integrators, in particular the leapfrog method, the methods of choice for collisionless galaxy simulations.

The Störmer–Verlet method can be interpreted in many different ways: besides the

composition of the Euler-A and Euler-B method, it is also equivalent to the following two-step method,

$$q_{n+1} - 2q_n + q_{n-1} = -h^2 \cdot \nabla V(q_n), \tag{1.21}$$

while (1.21) is the most natural central-difference discretization of (1.16); another way is as a splitting method, since $H(q,p) = \frac{|p|^2}{2} + V(q) = T(p) + V(q)$ is separable, their exact flow maps ϕ_h^T and ϕ_h^V have a closed form, and an approximation of the exact flow map $\phi_{H,h}$ by the composition $\phi_h^T \circ \phi_h^V$, which gives the Euler-B method, and $\phi_h^V \circ \phi_h^T$ gives the Euler-A method, while the Störmer–Verlet method is given by the Strang splitting $\phi_{h/2}^V \circ \phi_h^T \circ \phi_{h/2}^V$. Since ϕ_h^T and ϕ_h^V are exact flow maps of Hamiltonian vector fields, they are naturally symplectic, so are their compositions. Later, we will see that it also arises as a variational integrator.

For a nonseparable Hamiltonian system (1.2), we have the generalized Euler-B method,

$$\begin{cases} q_{n+1} = q_n + h \cdot \dfrac{\partial H}{\partial p}(q_n, p_{n+1}), \\ p_{n+1} = p_n - h \cdot \dfrac{\partial H}{\partial q}(q_n, p_{n+1}), \end{cases}$$

the generalized Euler-A method,

$$\begin{cases} q_{n+1} = q_n + h \cdot \dfrac{\partial H}{\partial p}(q_{n+1}, p_n), \\ p_{n+1} = p_n - h \cdot \dfrac{\partial H}{\partial q}(q_{n+1}, p_n), \end{cases}$$

and the generalized Störmer–Verlet method,

$$\begin{cases} p_{n+\frac{1}{2}} = p_n - \dfrac{h}{2} \cdot \dfrac{\partial H}{\partial q}\left(q_n, p_{n+\frac{1}{2}}\right), \\ q_{n+1} = q_n + \dfrac{h}{2} \cdot \left(\dfrac{\partial H}{\partial p}\left(q_n, p_{n+\frac{1}{2}}\right) + \dfrac{\partial H}{\partial p}\left(q_{n+1}, p_{n+\frac{1}{2}}\right)\right), \\ p_{n+1} = p_{n+\frac{1}{2}} - \dfrac{h}{2} \cdot \dfrac{\partial H}{\partial q}\left(q_{n+1}, p_{n+\frac{1}{2}}\right). \end{cases}$$

They are all symplectic, but notice that they are implicit as the Hamiltonian is no longer separable.

1.2.3 SHAKE and RATTLE methods

We consider constrained Hamiltonian mechanics for a Hamiltonian $H(q,p)$ with constraint $g(q)=0$, where $g:\mathbb{R}^n\to\mathbb{R}^k$, the equations of motion are of the form

$$\begin{cases} \dot{q} = \dfrac{\partial H}{\partial p}(q,p),\ 0 = g(q), \\ \dot{p} = -\dfrac{\partial H}{\partial q}(q,p) - G(q)^T\lambda, \end{cases} \tag{1.22}$$

where $G(q)\in\mathbb{R}^{k\times n}$ is the Jacobian matrix of $g(q)$. We adopt a geometric viewpoint, and let the configuration space be $Q=\{q\in\mathbb{R}^n \mid g(q)=0\}$, which is an embedded submanifold $Q\xrightarrow{i}\mathbb{R}^n$. The tangent bundle

$$TQ = \{(q,\dot{q}) \mid q\in\mathbb{R}^n, G(q)\dot{q}=0\},$$

is naturally embedded into $T\mathbb{R}^n$: $TQ\xrightarrow{Ti}T\mathbb{R}^n$. Notice that $\{\dot{q}\mid G(q)\dot{q}=0\}$ defines a subspace of $T_q\mathbb{R}^n$, they provide an extrinsic coordinate representation of the intrinsic tangent space T_qQ. It turns out that the Lagrangian setting is natural for discussing constrained mechanics due to the natural embedding of TQ into $T\mathbb{R}^n$, and the method of Lagrange multipliers for variational problems. Given a Lagrangian $L:\ T\mathbb{R}^n\to\mathbb{R}$, we define a constrained Lagrangian $L^Q:\ TQ\xrightarrow{Ti}T\mathbb{R}^n\to\mathbb{R}$. Then, Hamilton's principle of constrained mechanics on (TQ,L^Q) can be stated using extrinsic coordinates as

$$\delta\int_{t_0}^{t_1} L(q(t),\dot{q}(t))dt = 0, \tag{1.23}$$

for variations $q(t)$ with fixed endpoints $q(t_0)=q_0, q(t_1)=q_1$ that satisfy the constraint $g(q(t))=0$. Performing the variations on (1.23), we obtain the constrained Euler–Lagrange equation,

$$\begin{cases} \dfrac{\partial L}{\partial q} - \dfrac{d}{dt}\left(\dfrac{\partial L}{\partial \dot{q}}\right) = G(q)^T\lambda, \\ \qquad\qquad 0 = g(q). \end{cases} \tag{1.24}$$

It is harder to derive the constrained Hamilton's equations (1.22), since T^*Q does not naturally embedded in $T^*\mathbb{R}^n$. First, notice that the following diagram commutes

$$\begin{tikzcd} T\mathbb{R}^n \arrow[r, "\mathbb{F}L"] & T^*\mathbb{R}^n \arrow[d, "T^*i"] \\ TQ \arrow[u, "Ti"] \arrow[r, "\mathbb{F}L^Q"'] & T^*Q \end{tikzcd}$$

where T^*i is the pullback map. Assume that the Lagrangian is hyperregular, i.e., $\mathbb{F}L$ and $\mathbb{F}L^Q$ are global diffeomorphisms, then we can construct an embedding $\eta : T^*Q \to T^*\mathbb{R}^n$ as follows,

$$\eta = \mathbb{F}L \circ Ti \circ (\mathbb{F}L^Q)^{-1}. \tag{1.25}$$

For any $q \in Q$, we have that

$$T_q^*Q \underset{T^*i}{\overset{\eta}{\rightleftarrows}} T_q^*\mathbb{R}^n,$$

where T^*i is linear, and the preimage of T^*i are parallel affine subspaces, i.e., the translations of $\mathrm{Null}(T^*i)$. While η is nonlinear, it satisfies $T^*i \circ \eta = \mathrm{id}$. By applying the embedding η to the cotangent bundle

$$T^*Q = \left\{ (q,p) \mid g(q) = 0, G(q)\frac{\partial H}{\partial p}(q,p) = 0 \right\},$$

and applying the Legendre transform, (1.24) can be transformed into (1.22). So we can see that the differential–algebraic equation (1.22) describes a Hamilton system (T^*Q, H).

Consider a simpler case of a separable Hamiltonian $H(q,p) = \frac{|p|^2}{2} + V(q)$, with constraint $g(q) = 0$, which have the constrained Euler-Lagrange equations

$$\begin{cases} \ddot{q} = -\nabla V(q) - G(q)^T \lambda, \\ 0 = g(q), \end{cases}$$

and constrained Hamilton's equations,

$$\begin{cases} \dot{q} = p,\ 0 = g(q), \\ \dot{p} = -\nabla V(q) - G(q)^T \lambda. \end{cases} \tag{1.26}$$

Applying the same central-difference discretization that lead to the Störmer–Verlet method, we get the SHAKE method,

$$\begin{cases} q_{n+1} - 2q_n + q_{n-1} = -h^2 \cdot (\nabla V(q_n) + G(q_n)^T \lambda), \\ \qquad\qquad\qquad\qquad 0 = g(q_{n+1}), \end{cases}$$

which is a two-step method that was originally proposed by Ryckaert et al. [51] for molecular dynamics simulations. The Störmer–Verlet method can be adapted to the constrained Hamilton's equations (1.26) to yield the RATTLE method,

$$\begin{cases} p_{n+\frac{1}{2}} = p_n - \dfrac{h}{2} \cdot (\nabla V(q_n) + G(q_n)^T \lambda), \\ q_{n+1} = q_n + h \cdot p_{n+\frac{1}{2}},\ 0 = g(q_{n+1}), \\ p_{n+1} = p_{n+\frac{1}{2}} - \dfrac{h}{2} \cdot (\nabla V(q_{n+1}) + G(q_{n+1})^T \mu),\ 0 = G(q_{n+1})^T p_{n+1}. \end{cases} \tag{1.27}$$

Jay [28] and Reich [50] independently observed that the RATTLE method can be interpreted as a partitioned Runge–Kutta method and this allows it to be extended to general Hamiltonians,

$$\begin{cases} p_{n+\frac{1}{2}} = p_n - \dfrac{h}{2} \cdot \left(\dfrac{\partial H}{\partial q}(q_n, p_{n+\frac{1}{2}}) + G(q_n)^T \lambda \right), \\ q_{n+1} = q_n + \dfrac{h}{2} \cdot \left(\dfrac{\partial H}{\partial p}(q_n, p_{n+\frac{1}{2}}) + \dfrac{\partial H}{\partial p}(q_{n+1}, p_{n+\frac{1}{2}}) \right), \\ \qquad 0 = g(q_{n+1}), \\ p_{n+1} = p_{n+\frac{1}{2}} - \dfrac{h}{2} \cdot \left(\dfrac{\partial H}{\partial q}(q_{n+1}, p_{n+\frac{1}{2}}) + G(q_{n+1})^T \mu \right), \\ \qquad 0 = G(q_{n+1})^T \dfrac{\partial H}{\partial p}(q_{n+1}, p_{n+1}). \end{cases}$$

The restriction $G(q_{n+1})^T \frac{\partial H}{\partial p}(q_{n+1}, p_{n+1}) = 0$ guarantees that (q_{n+1}, p_{n+1}) lies in T^*Q, so the RATTLE method is a map $T^*Q \to T^*Q$. It inherits nice properties of the Störmer–Verlet method, in that it is also symplectic and symmetric.

1.3 High-order symplectic integrators

The Störmer–Verlet, SHAKE, and RATTLE methods are typical geometric integrators with nice properties, but the problem is that they are only second-order accurate. In contrast, Runge–Kutta methods provide a systematic way of constructing high-order integrators. For a general first-order differential equation

$$y' = f(t, y),$$

and a Butcher tableau,

Table 1.1. Butcher Tableau

$$\begin{array}{c|ccccc} c_1 & a_{11} & a_{12} & a_{13} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & a_{23} & \cdots & a_{2s} \\ c_3 & a_{31} & a_{32} & a_{33} & \cdots & a_{3s} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & a_{s3} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & b_3 & \cdots & b_s \end{array}$$

we have the corresponding Runge–Kutta method,

$$\begin{cases} Y_i = y_n + h \cdot \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j), \\ y_{n+1} = y_n + h \cdot \sum_{i=1}^{s} f(t_n + c_i h, Y_i). \end{cases} \tag{1.28}$$

A Runge–Kutta method (1.28) is symplectic when applied to Hamilton's equations (1.2) if its

coefficients satisfy the following property,

Theorem 4 (Sanz-Serna [52])**.** *If $b_i a_{ij} + b_j a_{ji} - b_i b_j = 0$ for $i, j = 1, 2 \cdots s$, then the corresponding Runge–Kutta method* (1.28) *is symplectic.*

In particular, we have that

Theorem 5. *The Gauss–Legendre Runge–Kutta methods are symplectic.*

The partitioned Runge–Kutta method is a generalization of the Runge–Kutta method that is particularly appropriate for Hamilton's equations (1.2), where it may be desirable for the position and momentum to be updated in different ways. Given the Butcher tableau

Table 1.2. Partitioned Butcher Tableau

$$
\begin{array}{c|ccccc}
c_1 & a_{11} & a_{12} & a_{13} & \cdots & a_{1s} \\
c_2 & a_{21} & a_{22} & a_{23} & \cdots & a_{2s} \\
c_3 & a_{31} & a_{32} & a_{33} & \cdots & a_{3s} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s1} & a_{s2} & a_{s3} & \cdots & a_{ss} \\
\hline
 & b_1 & b_2 & b_3 & \cdots & b_s \\
\hline
\overline{c}_1 & \overline{a}_{11} & \overline{a}_{12} & \overline{a}_{13} & \cdots & \overline{a}_{1s} \\
\overline{c}_2 & \overline{a}_{21} & \overline{a}_{22} & \overline{a}_{23} & \cdots & \overline{a}_{2s} \\
\overline{c}_3 & \overline{a}_{31} & \overline{a}_{32} & \overline{a}_{33} & \cdots & \overline{a}_{3s} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\overline{c}_s & \overline{a}_{s1} & \overline{a}_{s2} & \overline{a}_{s3} & \cdots & \overline{a}_{ss} \\
\hline
 & \overline{b}_1 & \overline{b}_2 & \overline{b}_3 & \cdots & \overline{b}_s
\end{array}
$$

the corresponding partitioned Runge–Kutta method for Hamilton's equations (1.2) is

$$\begin{cases} Q_i = q_n + h \cdot \sum_{j=1}^{s} a_{ij} \frac{\partial H}{\partial p}(Q_j, P_j),\ P_i = p_n - h \cdot \sum_{j=1}^{s} \overline{a}_{ij} \frac{\partial H}{\partial q}(Q_j, P_j), \\ q_{n+1} = q_n + h \cdot \sum_{i=1}^{s} b_i \frac{\partial H}{\partial p}(Q_i, P_i),\ \ p_{n+1} = p_n - h \cdot \sum_{i=1}^{s} \overline{b}_i \frac{\partial H}{\partial q}(Q_i, P_i). \end{cases} \tag{1.29}$$

For the partitioned Runge–Kutta method (1.29) to be symplectic, the coefficients have the satisfy the following property,

Theorem 6 (Geng [14])**.** *If $b_i = \overline{b}_i$, $b_i\overline{a}_{ij} + \overline{b}_j a_{ji} - b_i\overline{b}_j = 0$, for $i, j = 1, 2 \cdots s$, then the corresponding partitioned Runge–Kutta method* (1.29) *is symplectic.*

Applying the partitioned Runge–Kutta method to the constrained Hamilton's equations (1.22) yield

$$\begin{cases} \dot{Q}_i = \frac{\partial H}{\partial p}(Q_i, P_i),\ \ \dot{P}_i = -\frac{\partial H}{\partial q}(Q_i, P_i) - G(Q_i)^T \lambda_i, \\ Q_i = q_0 + h \cdot \sum_{j=1}^{s} a_{ij} \dot{Q}_j,\ \ P_i = p_0 + h \cdot \sum_{j=1}^{s} \overline{a}_{ij} \dot{P}_j, \\ 0 = g(Q_i), \\ q_1 = q_0 + h \cdot \sum_{i=1}^{s} b_i \dot{Q}_i,\ \ p_1 = p_0 + h \cdot \sum_{i=1}^{s} \overline{b}_i \dot{P}_i. \end{cases} \tag{1.30}$$

The existence and uniqueness of the solution (q_1, p_1) to (1.30) is not guaranteed without further assumptions. The symplecticity of (1.30) was studied in [28],

Theorem 7. *If $b_i = \overline{b}_i$, $b_i\overline{a}_{ij} + \overline{b}_j a_{ji} - b_i\overline{b}_j = 0$, for $i, j = 1, 2 \cdots s$, and* (1.30) *is well-posed, then the map $(q_0, p_0) \to (q_1, p_1)$ is symplectic.*

One well-posed version of (1.30) that yields a symplectic map from T^*Q to T^*Q is given

as follows

$$\begin{cases} \dot{Q}_i = \dfrac{\partial H}{\partial p}(Q_i, P_i), \;\; \dot{P}_i = -\dfrac{\partial H}{\partial q}(Q_i, P_i) - G(Q_i)^T \lambda_i, \\ Q_i = q_0 + h \cdot \sum_{j=1}^{s} a_{ij} \dot{Q}_j, \;\; P_i = p_0 + h \cdot \sum_{j=1}^{s} \overline{a}_{ij} \dot{P}_j, \\ \;\; 0 = g(Q_i), \;\; 0 = G(q_1)^T \dfrac{\partial H}{\partial p}(q_1, p_1), \\ q_1 = q_0 + h \cdot \sum_{i=1}^{s} b_i \dot{Q}_i, \;\; p_1 = p_0 + h \cdot \sum_{i=1}^{s} \overline{b}_i \dot{P}_i, \end{cases} \tag{1.31}$$

where the coefficients of the Butcher tableau satisfy $b_i = \overline{b}_i$, $b_i \overline{a}_{ij} + \overline{b}_j a_{ji} - b_i \overline{b}_j = 0$. We also require the method to be stiff, i.e., $a_{si} = b_i$, which means that $q_1 = Q_s$. Furthermore, we require that $b_i \neq 0$, which implies that $\overline{a}_{is} = 0$. We also set $a_{1i} = 0$, which means that $q_0 = Q_1$, and the condition that $b_i \neq 0$ now implies that $\overline{a}_{i1} = \overline{b}_i$.

1.4 Variational integrators

Surprisingly, all the symplectic integrators we have introduced so far, including the constrained and unconstrained cases, can be derived using a variational integrator approach. This is a unified framework to derive a large class of geometric integrators, and the resulting methods are automatically symplectic. Given a Lagrangian mechanical system (TQ, L), fix the timestep h, and choose $q_0, q_1 \in Q$. Then, we define the exact discrete Lagrangian,

$$L_d^{\text{exact}}(q_0, q_1) = \underset{\substack{q \in C^2([0,h],Q) \\ q(0)=q_0, q(h)=q_1}}{} \int_0^h L(q(t), \dot{q}(t)) dt, \tag{1.32}$$

which is related to Jacobi's solution of the Hamilton–Jacobi equation. It can equivalently be defined as $L_d^{\text{exact}}(q_0, q_1) = \int_0^h L(q(t), \dot{q}(t)) dt$, where $q(t)$ satisfies the Euler-Lagrange equation, and the boundary conditions $q(0) = q_0$, $q(h) = q_1$. Consider

$$L_d^{\text{exact}}(q_{k-1}, q_k) + L_d^{\text{exact}}(q_k, q_{k-1}), \tag{1.33}$$

where $q_0 = q(t)$, $q_1 = q(t+h)$, $q_2 = q(t+2h)$. By Hamilton's principle, (1.33) should be stationary with respect to variations in q_k, i.e.,

$$D_2 L_d^{\text{exact}}(q_{k-1}, q_k) + D_1 L_d^{\text{exact}}(q_k, q_{k+1}) = 0,$$

which implicitly defines q_{k+1} in terms of q_{k-1}, q_k. However, the exact discrete Lagrangian is a theoretical construct that is not practically computable. Suppose that $L_d(q_k, q_{k+1})$ is an approximation of $L_d^{\text{exact}}(q_k, q_{k+1})$, then consider the discrete action sum,

$$\mathbb{S}_d = \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}).$$

Then, the discrete Hamilton's principle states that

$$\delta \mathbb{S}_d = 0,$$

where the variations at the endpoints vanish, i.e., $\delta q_0 = 0$, $\delta q_N = 0$. From this, we obtain the discrete Euler–Lagrange equations,

$$D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1}) = 0, \tag{1.34}$$

which is a two-step method $(q_{k-1}, q_k) \mapsto (q_k, q_{k+1})$. To construct a one-step method on T^*Q, we need the following observation,

$$p_k = -D_1 L_d^{\text{exact}}(q_k, q_{k+1}), \quad p_{k+1} = D_2 L_d^{\text{exact}}(q_k, q_{k+1}),$$

p_k, p_{k+1} are the exact momenta at the endpoints. This is simply a consequence of the fact that the exact discrete Lagrangian is related to Jacobi's solution of the Hamilton–Jacobi equation, and it is the Type I generating function of the exact time-h flow map of Hamilton's equations.

Thus, by analogy, we can require that the same relationship holds with the approximated discrete Lagrangian $L_d(q_k, q_{k+1})$,

$$\begin{cases} p_k = -D_1 L_d(q_k, q_{k+1}), \\ p_{k+1} = D_2 L_d(q_k, q_{k+1}). \end{cases} \tag{1.35}$$

We refer to these as the implicit discrete Euler–Lagrange equations, and they define a map $(q_0, p_0) \mapsto (q_1, p_1)$ that is automatically symplectic, since $L_d(q_0, q_1)$ serves as the Type I generating function of a symplectic map here. It is also easy to verify that (1.35) implies (1.34) if the momentum variables are eliminated. When $L_d = L_d^{\text{exact}}$, (1.35) implicitly defines the exact time-h flow map of Hamilton's equations. In general, the order of the one-step method (1.35) is determined by the approximation order of L_d to L_d^{exact}, which is a variational error analysis result that was introduced in [43].

Theorem 8. *If $L_d(q_0, q_1) = L_d^{exact}(q_0, q_1) + O(h^{r+1})$, then the one-step method* (1.35) *has order r.*

Given a Lagrangian $L(q, \dot{q})$, one possible discrete Lagrangian is given by,

$$L_d(q_0, q_1) = h \cdot L\left(\frac{q_0 + q_1}{2}, \frac{q_1 - q_0}{h}\right).$$

This is obtained by restricting the curves $q(t)$ to polynomials of degree one or less, and approximating the integral by the midpoint rule. Applying (1.35), we get

$$\begin{cases} -p_0 = h \cdot \left[\frac{\partial L}{\partial q}\left(\frac{q_0 + q_1}{2}, \frac{q_1 - q_0}{h}\right) \cdot \frac{1}{2} + \frac{\partial L}{\partial \dot{q}}\left(\frac{q_0 + q_1}{2}, \frac{q_1 - q_0}{h}\right) \cdot \left(-\frac{1}{h}\right)\right], \\ p_1 = h \cdot \left[\frac{\partial L}{\partial q}\left(\frac{q_0 + q_1}{2}, \frac{q_1 - q_0}{h}\right) \cdot \frac{1}{2} + \frac{\partial L}{\partial \dot{q}}\left(\frac{q_0 + q_1}{2}, \frac{q_1 - q_0}{h}\right) \cdot \frac{1}{h}\right]. \end{cases} \tag{1.36}$$

The Lagrangian $L(q, \dot{q})$ and the Hamiltonian $H(q, p)$ are related by the Legendre transformation, and from this, we can show that

$$\dot{q} = \frac{\partial H}{\partial p}(q, p), \quad p = \frac{\partial L}{\partial \dot{q}}(q, \dot{q}), \quad \frac{\partial L}{\partial q}(q, \dot{q}) = -\frac{\partial H}{\partial q}(q, p).$$

Then, (1.36) can be expressed in terms of the Hamiltonian,

$$\begin{cases} q_1 = q_0 + h \cdot \dfrac{\partial H}{\partial p}\left(\dfrac{q_0+q_1}{2}, \dfrac{p_0+p_1}{2}\right), \\ p_1 = p_0 - h \cdot \dfrac{\partial H}{\partial q}\left(\dfrac{q_0+q_1}{2}, \dfrac{p_0+p_1}{2}\right). \end{cases} \tag{1.37}$$

This is the midpoint rule, which by our construction, is a variational integrator. Similarly, for a separable Lagrangian $L(q,\dot{q}) = \frac{1}{2}\dot{q}^T M\dot{q} - V(q)$, consider the discrete Lagrangian,

$$L_d(q_0,q_1) = \frac{h}{2}\left[L\left(q_0, \frac{q_1-q_0}{h}\right) + L\left(q_1, \frac{q_1-q_0}{h}\right)\right],$$

which is obtained by restricting the curves $q(t)$ to polynomials of degree one or less, and approximating the integral by the trapezoidal rule. The resulting Hamiltonian map is exactly the Störmer–Verlet method (1.20).

To construct higher-order variational integrators, we need higher-order approximations of the exact discrete Lagrangian (1.32). One way to systematically construct such approximations is by using the Galerkin method. Given a quadrature rule $\int_0^h L(q(t),\dot{q}(t))dt \approx h\sum_{i=1}^s b_i L(q(c_i h),\dot{q}(c_i h))$, we construct a Galerkin discrete Lagrangian,

$$L_d(q_0,q_1) = \underset{\substack{q\in\mathbb{P}^k(0,h) \\ q(0)=q_0, q(h)=q_1}}{} h\sum_{i=1}^s b_i L(q(c_i h),\dot{q}(c_i h)),$$

where $\mathbb{P}^k(0,h)$ is the space of polynomials of degree less than or equal to k. So we only consider variation curves that are polynomials and satisfy the prescribed boundary conditions. The resulting variational integrator is a partitioned Runge–Kutta method (1.29), with coefficients given by

$$b_i = \overline{b}_i = \int_0^1 l_i(x)dx,\ a_{ij} = \int_0^{c_i} l_j(x)dx,\ \overline{a}_{ij} = \overline{b}_j(1-\frac{a_{ji}}{b_i}),$$

where $l_i(x)$ are the Lagrange interpolating polynomials for nodes c_i.

Let us derive a variational integrator for constrained mechanics (1.22). For the discrete

Euler–Lagrange equation (1.34), we need to perform variations of q_k that are restricted to the configuration space Q, and this yields the constrained discrete Euler–Lagrange equations,

$$\begin{cases} D_2L_d(q_0,q_1)+D_1L_d(q_1,q_2)=G(q_1)^T\lambda, \\ \qquad\qquad\qquad\qquad 0=g(q_1). \end{cases} \tag{1.38}$$

We can do the same for (1.35), but the momenta need to be represented by extrinsic coordinates on T^*Q embedded into $T^*\mathbb{R}^n$ by η (1.25). Since the preimage of T^*i is a parallel translation of $\mathrm{Null}(T^*i)=\mathrm{span}(G(q)^T)$, (1.35) can equivalently be written as

$$\begin{cases} p_0=-D_1L_d(q_0,q_1)+G(q_0)^T\lambda_0, \\ \ 0=g(q_1), \\ p_1=D_2L_d(q_0,q_1)-G(q_1)^T\lambda_1, \\ \ 0=G(q_1)\dfrac{\partial H}{\partial p}(q_1,p_1). \end{cases} \tag{1.39}$$

From their construction, we expect that (1.39) implies (1.38), and that (1.39) is symplectic. Given a Lagrangian $L(q,\dot{q})=\frac{|\dot{q}|^2}{2}-V(q)$, with the constraint $0=g(q)$, and a discrete Lagrangian of the form,

$$L_d(q_0,q_1)=\frac{h}{2}\left[L\left(q_0,\frac{q_1-q}{h}\right)+L\left(q_1,\frac{q_1-q_0}{h}\right)\right].$$

Then, (1.38) and (1.39) specialized to this case both recover the SHAKE method.

1.5 Kalman Filters and Observability Analysis

1.5.1 Kalman Filters

Uncertainty is ubiquitous in practice, and may arise from incompleteness of the model, inaccuracy of sensor data, and uncontrollable environmental factors. Probabilistic approaches are a nice way to deal with uncertainty. Among these, the Kalman filter is quite popular in state estimation. Suppose we have a robot moving in the environment, with state variable x_t, control

variable u_t, and measurement z_t. In practice, the state consists of the position and orientation of the robot, the controls are data from odometry and IMU, and the measurements are data from the camera and LIDAR. (x_t, u_t, z_t) forms a discrete control system. We assume that the probability distribution satisfies the Markovian property,

$$\begin{cases} P(x_t \mid x_{0:t-1}, z_{0:t-1}, u_{1:t}) = P(x_t \mid x_{t-1}, u_t), \\ P(z_t \mid x_{0:t}, z_{0:t-1}, u_{1:t}) = P(z_t \mid x_t). \end{cases} \tag{1.40}$$

The state transition probability is denoted by $P(x_t \mid x_{t-1}, u_t)$, and the Markovian property implies that we only need the current state and control data to predict the future, independent of the history. The measurement probability is denoted by $P(z_t \mid x_t)$, and it implies that the measurement data only depends on the state. In principle, we would like to use all the information we have, i.e., $(z_{0:t}, u_{1:t})$, to get the best estimate of the state x_t. This gives the posterior belief

$$\mathrm{bel}(x_t) = P(x_t \mid z_{0:t}, u_{1:t}),$$

which uses all the historical sensor data. This is in contrast to the prediction belief

$$\overline{\mathrm{bel}}(x_t) = P(x_t \mid z_{0:t-1}, u_{1:t}),$$

which uses all the historical data except for the current measurement z_t.

We have a very efficient algorithm to estimate $\mathrm{bel}(x_t)$, the Bayes filter,

$$\begin{cases} \overline{\mathrm{bel}}(x_t) = \int P(x_t \mid x_{t-1}, u_t) \cdot \mathrm{bel}(x_{t-1}) dx_{t-1}, \\ \mathrm{bel}(x_t) \propto P(z_t \mid x_t) \cdot \overline{\mathrm{bel}}(x_t). \end{cases} \tag{1.41}$$

Once we know $\mathrm{bel}(x_{t-1})$, we can estimate $\mathrm{bel}(x_t)$ by using (1.41), and it yields the best estimate based on historical data. The first step is propagation of the state distribution via the state

transition model $P(x_t \mid x_{t-1}, u_t)$. The second step is essentially a Baysian inference, where $\overline{\text{bel}}(x_t)$ acts as the prior distribution, and $\text{bel}(x_t)$ is the posterior distribution. So Bayes filter is the composition of propagation and measurement update.

The Kalman filter is a concrete implementation of (1.41) for the linear Gaussian model. Suppose that the probability distributions are all Gaussian, and the state transition probability and measurement probability are given by a linear noisy model,

$$\begin{cases} x_t = A_t x_{t-1} + u_t + \varepsilon_t, \\ z_t = C_t x_t + \delta_t. \end{cases} \tag{1.42}$$

where ε_t is the propagation error with Gaussian distribution $\mathcal{N}(0, R_t)$, and δ_t is the measurement noise with Gaussian distribution $\mathcal{N}(0, Q_t)$. Under the linear Gaussian model (1.42), suppose that $\text{bel}(x_{t-1}) = \mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$, then $\text{bel}(x_t) = \mathcal{N}(\mu_t, \Sigma_t)$ can be calculated as follows [4],

$$\begin{cases} \overline{\mu}_t = A_t \mu_{t-1} + u_t, \\ \overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t, \\ K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}, \\ \mu_t = \overline{\mu}_t + K_t (z_t - C_t \overline{\mu}_t), \\ \Sigma_t = (I - K_t C_t) \overline{\Sigma}_t. \end{cases} \tag{1.43}$$

For Gaussian distributions, besides the mean-covariance representation $\mathcal{N}(\mu, \Sigma)$, we also have an information-precision representation $\mathcal{N}(\xi, \Omega)$, where $\Sigma = \Omega^{-1}$, $\mu = \Omega^{-1}\xi$. The Kalman

filter (1.43) in the information-precision representation has the following form,

$$\begin{cases} \overline{\Omega}_t = (A_t \Omega_{t-1}^{-1} A_t^T + R_t)^{-1}, \\ \overline{\xi}_t = \overline{\Omega}_t (A_t \Omega_{t-1}^{-1} \xi_{t-1} + u_t), \\ \Omega_t = C_t^T Q_t^{-1} C_t + \overline{\Omega}_t, \\ \xi_t = C_t^T Q_t^{-1} z_t + \overline{\xi}_t. \end{cases} \tag{1.44}$$

This is called the information filter.

1.5.2 Linear Time-invariant System

Consider a linear time-invariant control system,

$$\begin{cases} \dot{x} = Ax + Bu, \\ z = Cx + Du. \end{cases} \tag{1.45}$$

where x is the state variable, u is the control variable, and z is the measurement. It has an explicit solution for the initial condition $x(t_0)$ and a given control $u(t)$. The state evolves as

$$x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)} Bu(\tau) d\tau,$$

and the measurement is given by

$$z(t) = Ce^{A(t-t_0)} x(t_0) + C \int_{t_0}^{t} e^{A(t-\tau)} Bu(\tau) d\tau + Du(t).$$

Observability is an important concept for control systems. It quantifies the extent to which we can recover state information directly based on the control u and the measurement z. If the state is observable, then in state estimation, the state error will remain bounded, otherwise, the estimation error will grow with bound. For a linear time-invariant system, we have the following theorem [10],

Theorem 9. *The following statements are equivalent:*

- *A linear time-invariant control system* (1.45) *is observable;*
- *The observability Gramian* $W_o(t) = \int_0^t e^{A^T\tau} C^T C e^{A\tau} d\tau$ *is nonsingular for any* $t > 0$*;*
- *The observability matrix* $O = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix}$ *has full column rank.*

When the observability matrix is rank deficient, i.e., $\text{rank}(O) = n_1 < n$, the the linear time-invariant control system (1.45) contains an unobservable subspace. Construct the transformation matrix P, whose first n_1 rows are a linearly independent basis for O, and remaining $n - n_1$ rows are chosen arbitrarily so that P has full rank. Then, we perform a coordinate transformation to new variables $\bar{x} = Px$. It can be verified that the transformed control system has the following special structure,

$$\begin{cases} \begin{pmatrix} \dot{x}_O \\ \dot{x}_N \end{pmatrix} = \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_O \\ x_N \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u, \\ z = \begin{pmatrix} C_1 & 0 \end{pmatrix} \begin{pmatrix} x_O \\ x_N \end{pmatrix} + Du. \end{cases} \tag{1.46}$$

From (1.46), we can see that the measurement z only depends on observable part x_O, and that x_O evolves independently of x_N. Thus, the x_N part is unobservable. This step is called a canonical decomposition. Even though the transformation P is constructed with some freedom, the x_N part has a clear geometric meaning, it forms the unobservable subspace, which is the nullspace of the observability matrix, i.e., $x_N = \text{Null}(O)$.

1.5.3 Observability of Nonlinear systems

We require the language of manifolds to deal with nonlinear phenomena. On a differential manifold, a distribution $\triangle$ on a smooth manifold M is a smooth assignment of a subspace $\triangle(p) \subseteq T_pM$ of the tangent space at each point $p \in M$. A codistribution Ω on M is a smooth assignment of a subspace $\Omega(p) \subseteq T_p^*M$ of the cotangent space at each point $p \in M$. Given a codistribution Ω, we can define its orthogonal distribution $\Omega^\perp$, where each $\Omega^\perp(p)$ is the annihilator of $\Omega(p)$,

$$\Omega^\perp(p) = \{v \in T_pM \mid \langle v, \alpha \rangle = 0, \forall \alpha \in \Omega(p)\}.$$

Given a distribution $\triangle$, codistribution Ω, and vector field f on a smooth manifold M, we say that a distribution $\triangle$ is invariant under the vector field f if and only if

$$\forall X \in \triangle, \quad [f, X] \in \triangle.$$

Similarly, we say that a codistribution Ω is invariant under the vector field f if and only if

$$\forall \omega \in \Omega, \quad L_f \omega \in \Omega.$$

Suppose that we have vector fields $f_1, f_2, \ldots f_N$ and a distribution $\triangle$, then the smallest distribution that contains $\triangle$ and is invariant under $f_1, f_2, \ldots f_N$ is denoted by $\langle f_1, f_2, \ldots f_N \mid \triangle \rangle$. This can be constructed by the following generating sequence,

$$\begin{aligned} \triangle_0 &= \triangle, \\ \triangle_k &= \triangle_{k-1} + \sum_{i=1}^{N} [f_i, \triangle_{k-1}]. \end{aligned}$$

Theorem 10. *For all k, $\triangle_k \subseteq \langle f_1, f_2, \ldots f_N \mid \triangle \rangle$, and when at a particular step k^*, $\triangle_{k^*} = \triangle_{k^*+1}$,*

then $\triangle_{k^*} = \langle f_1, f_2, \dots f_N | \triangle \rangle$.

Similarly, for a codistribution Ω, the smallest codistribution that contains Ω and is invariant under $f_1, f_2, \dots f_N$ is denoted by $\langle f_1, f_2, \dots f_N \mid \Omega \rangle$, and it can be generated by

$$\Omega_0 = \Omega,$$
$$\Omega_k = \triangle_{k-1} + \sum_{i=1}^{N} L_{f_i} \Omega_{k-1}.$$

We consider a special class of nonlinear control systems, which are affine systems of the form,

$$\begin{cases} \dot{x} = f(x) + \sum_{i=1}^{m} g_i(x) u_i, \\ y_i = h_i(x), i = 1, 2 \dots p. \end{cases} \tag{1.47}$$

Then, consider

$$Q = \langle f, q_1, g_2 \dots g_m \mid \text{span}\{dh_1, dh_2 \dots dh_p\} \rangle^{\perp},$$

which is the maximal distribution that is contained in the span$\{dh_1, dh_2 \dots dh_p\}$ and invariant under $f, q_1, g_2 \dots g_m$. This distribution corresponds to the decomposition into the observable and unobservable parts of system. We have the following theorem,

Theorem 11. *Suppose Q is nonsingular, then at each point p, there exists a local coordinate*

change $z = \phi(x)$, such that (1.47) *can be transformed into the form,*

$$\begin{cases} \dot{z}_1 = f_1(z_1, z_2 \dots z_n) + \sum_{i=1}^{m} g_{i1}(z_1, \dots z_n) u_i, \\ \quad \vdots \\ \dot{z}_s = f_1(z_1, z_2 \dots z_n) + \sum_{i=1}^{m} g_{is}(z_1, \dots z_n) u_i, \\ \dot{z}_{s+1} = f_{s+1}(z_{s+1} \dots z_n) + \sum_{i=1}^{m} g_{i,s+1}(z_{s+1}, \dots z_n) u_i, \\ \quad \vdots \\ \dot{z}_n = f_n(z_{s+1} \dots z_n) + \sum_{i=1}^{m} g_{in}(z_{s+1} \dots z_n) u_i, \\ y_i = h_i(z_{s+1} \dots z_n), \quad 1 \leq i \leq p. \end{cases} \tag{1.48}$$

It is apparent from the form of this system of equations that $z_1, z_2 \dots z_n$ are unobservable.

Chapter 2

Geometric Exponential Integrators

2.1 Introduction

Hamiltonian partial differential equations are often simulated by semi-discretizing the spatial differential operators, and applying a symplectic or energy-preserving integrator to the resulting system of Hamiltonian ordinary differential equations. The critical challenge associated with such an approach is that the resulting system of differential equations become increasingly stiff as the spatial mesh is refined. This is achieved by combining geometric integrators with exponential integrators [22], which are a class of numerical integrators for stiff systems whose vector field can be decomposed into a linear term and a nonlinear term,

$$\dot{q} = Aq + f(q). \tag{2.1}$$

Usually, the coefficient matrix A has a large spectral radius, and is responsible for the stiffness of the system of differential equations, while the nonlinear term $f(q)$ is relatively smooth. There are various ways to construct an exponential integrator [45]. For example, we can perform a change of variables $\tilde{q}(t) = e^{-At}q(t)$, and transform (2.1) to obtain

$$[e^{-At}q(t)]' = \tilde{q}'(t) = e^{-At}f(e^{At}\tilde{q}(t)). \tag{2.2}$$

Notice that the Jacobi matrix of (2.2) equals $e^{-At}\nabla f e^{At}$, which has a smaller spectral radius than the Jacobi matrix $A+\nabla f$ of (2.1). A natural idea is to apply a classical integrator for the mollified system (2.2) to obtain an approximation of $\tilde{q}(t)$, then invert the change of variables to obtain an approximation of the solution $q(t)$ of (2.1). In Section 2.2, we shall demonstrate how to construct symplectic exponential integrators using this approach.

Another way of constructing exponential integrators starts from the variation-of-constants formula,

$$q(t) = e^{A(t-t_0)}q(t_0) + \int_{t_0}^{t} e^{A(t-\tau)} f(q(\tau))d\tau, \tag{2.3}$$

which is the exact solution for (2.1) with initial condition $q(t_0) = q_0$. Then, a computable approximation can be obtained by approximating the $f(q(\tau))$ term inside the integral. If we approximate $f(q(\tau))$ by $f(q_k)$, we arrive at the exponential Euler method,

$$q_{k+1} = e^{Ah}q_k + \int_0^h e^{A\tau}d\tau \cdot f(q_k). \tag{2.4}$$

An exponential Runge–Kutta method of collocation type [21] could also be constructed by approximating $f(q(\tau))$ with polynomials. In Section 2.3, we shall show how to construct energy-preserving exponential integrators from (2.3).

In this paper, we consider a specific form of (2.1) which is a Poisson system. We assume $A = JD$, $f(q) = J\nabla V(q)$, where $J^{\mathrm{T}} = -J, D^{\mathrm{T}} = D$, and $JD = DJ$, thus the coefficient matrix A is also skew-symmetric. The assumption that J and D commutes turns out to be essential for constructing structure-preserving integrators, and it is naturally satisfied in practical problems. Now, the semilinear system (2.1) can be written as,

$$\dot{q} = J(Dq + \nabla V(q)) = J\nabla H(q), \tag{2.5}$$

with Hamiltonian function $H(q) = \frac{1}{2}q^{\mathrm{T}}Dq + V(q)$. Equation (2.5) describes a constant Poisson system, and there are at least two quantities that are preserved by the flow: the Poisson structure

$J_{ij}\frac{\partial}{\partial x_i}\otimes\frac{\partial}{\partial x_j}$ and Hamiltonian $H(q)$. Geometric integrators that preserve the geometric structure and first integrals of the system typically exhibit superior qualitative properties when compared to non-geometric integrators, and they are an active area of research [17, 37, 43].

In this paper we will introduce the exponential midpoint rule and the energy-preserving exponential integrator. Both of these methods are implicit, and a significant advantage of adopting the exponential integrator approach in the context of geometric integrators is that resulting geometric exponential integrators can be implemented using fixed point iterations as opposed to the more computationally expensive Newton iterations. Recall that classical implicit Runge-Kutta methods

$$\begin{cases} Y_i = y_n + h\cdot\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j), \\ y_{n+1} = y_n + h\cdot\sum_{i=1}^{s} b_i f(t_n + c_i h, Y_i), \end{cases}$$

have a form that naturally lends itself to fixed point iterations. However, when $\frac{\partial f}{\partial y}$ has a large spectral radius, the timestep is forced to be very small in order to guarantee that the fixed point iteration converges. The alternative is to use a Newton type iteration, which is time consuming since we need to perform LU decomposition ($O(n^3)$ complexity) during each iteration. This is the problem we face for the stiff semilinear system (2.1) when the coefficient matrix A has a large spectral radius. In contrast, in both the exponential midpoint rule

$$q_{k+1} = e^{Ah}q_k + h\cdot e^{A\frac{h}{2}} f\Big(\frac{e^{A\frac{h}{2}}q_k + e^{-A\frac{h}{2}}q_{k+1}}{2}\Big),$$

and the energy-preserving exponential integrator

$$q_{k+1} = e^{Ah}q_k + \int_0^h e^{A\tau}d\tau\cdot J\overline{\nabla}V(q_k, q_{k+1}),$$

the matrix A only appears in the exponential term e^{Ah}. Since A is skew-symmetric, this term is an orthogonal matrix, which has spectral radius 1. Thus, fixed point iterations can be used to

implement the exponential integrator, regardless of the stiffness of A.

In summary, the main contributions of this paper involve the derivation of geometric exponential integrators that either preserve the Poisson structure or Hamiltonian of (2.5). They exhibit long time stability, allow for relatively larger timesteps for the stiff problem, and are computationally more efficient as they can be implemented using fixed point iterations as opposed to Newton type iterations. For the rest of the paper, Section 2.2 is devoted to developing symplectic exponential integrators that preserve the Poisson structure; Section 2.3 is devoted to developing energy preserving exponential integrators; numerical methods and experiments are presented in Section 2.4 and Section 2.5, respectively.

2.2 Symplectic Exponential Integrator

For constant Poisson systems (2.5), it was shown in [67] that the midpoint rule and diagonally implicit symplectic Runge–Kutta methods preserve the Poisson structure $J_{ij}\frac{\partial}{\partial x_i}\otimes\frac{\partial}{\partial x_j}$. We first start by constructing an exponential midpoint rule: apply the classical midpoint rule to the transformed system (2.2) to obtain

$$\frac{\tilde{q}_{k+1}-\tilde{q}_k}{h} = e^{-At_{k+1/2}} f\Big(e^{At_{k+1/2}}\frac{\tilde{q}_{k+1}+\tilde{q}_k}{2}\Big), \tag{2.6}$$

where

$$t_{k+1/2} = \frac{t_k+t_{k+1}}{2}, \quad h = t_{k+1}-t_k, \quad \tilde{q}_k = e^{-At_k}q_k, \quad \tilde{q}_{k+1} = e^{-At_{k+1}}q_{k+1}.$$

Transform (2.6) back to q_k and q_{k+1}, and we obtain the exponential midpoint rule,

$$q_{k+1} = e^{Ah}q_k + h\cdot e^{A\frac{h}{2}} f\Big(\frac{e^{A\frac{h}{2}}q_k + e^{-A\frac{h}{2}}q_{k+1}}{2}\Big). \tag{2.7}$$

Theorem 12. *The exponential midpoint rule* (2.7) *preserves the Poisson structure when applied*

to the semilinear Poisson system (2.5).

Proof. Recall that a map ϕ preserves Poisson structure $J_{ij}\frac{\partial}{\partial x_i}\otimes\frac{\partial}{\partial x_j}$ iff

$$(\nabla\phi)J(\nabla\phi)^{\mathrm{T}} = J. \tag{2.8}$$

Differentiating (2.7), we obtain,

$$\begin{aligned} dq_{k+1} &= e^{Ah}dq_k + h\cdot e^{A\frac{h}{2}}\nabla f\Big(\frac{1}{2}e^{A\frac{h}{2}}dq_k + \frac{1}{2}e^{-A\frac{h}{2}}dq_{k+1}\Big), \\ \Big(I - \frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{-A\frac{h}{2}}\Big)dq_{k+1} &= \Big(e^{Ah} + \frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{A\frac{h}{2}}\Big)dq_k. \end{aligned}$$

So the map $\phi(q_k) = q_{k+1}$ has Jacobi matrix $\nabla\phi = M^{-1}N$, where

$$\begin{aligned} M &= I - \frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{-A\frac{h}{2}} = I - \frac{h}{2}\cdot e^{A\frac{h}{2}}J\nabla^2 V e^{-A\frac{h}{2}}, \\ N &= e^{Ah} + \frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{A\frac{h}{2}} = e^{Ah} + \frac{h}{2}\cdot e^{A\frac{h}{2}}J\nabla^2 V e^{A\frac{h}{2}}. \end{aligned}$$

Then, we just need to verify (2.8), which is equivalent to $MJM^{\mathrm{T}} = NJN^{\mathrm{T}}$,

$$\begin{aligned} MJM^{\mathrm{T}} &= \Big(I - \frac{h}{2}\cdot e^{A\frac{h}{2}}J\nabla^2 V e^{-A\frac{h}{2}}\Big)J\Big(I + \frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla^2 V J e^{-A\frac{h}{2}}\Big) \\ &= J - \frac{h^2}{4}e^{A\frac{h}{2}}J\nabla^2 V e^{-A\frac{h}{2}}Je^{A\frac{h}{2}}\nabla^2 V J e^{-A\frac{h}{2}} \\ &= J - \frac{h^2}{4}e^{A\frac{h}{2}}J\nabla^2 V J\nabla^2 V J e^{-A\frac{h}{2}} \\ &= NJN^{\mathrm{T}}. \end{aligned} \tag{2.9}$$

In (2.9), we used the property that $\nabla^2 V$ is symmetric, that A is skew-symmetric which implies that $(e^{Ah})^{\mathrm{T}} = e^{-Ah}$. Furthermore, the assumption that D and J commutes, implies that the matrix exponential $e^{A\frac{h}{2}}$, where $A = JD$, also commutes with J. □

The exponential midpoint rule is a second-order method, and we will now develop higher-order symplectic exponential integrators. Recall that a general diagonally implicit Symplectic

Runge–Kutta method (DISRK) [17, Theorem 4.4] has a Butcher tableau of the form given in Table 2.1:

Table 2.1. DISRK

$$\begin{array}{c|ccccc} c_1 & \frac{b_1}{2} & 0 & 0 & 0 & 0 \\ c_2 & b_1 & \frac{b_2}{2} & 0 & 0 & 0 \\ c_3 & b_1 & b_2 & \frac{b_3}{2} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & b_1 & b_2 & b_3 & \cdots & \frac{b_s}{2} \\ \hline & b_1 & b_2 & b_3 & \cdots & b_s \end{array}$$

If we apply the DISRK method to the transformed system (2.2), and then convert back, we obtain the following diagonally implicit symplectic exponential (DISEX) integrator,

$$\begin{cases} Q_i = e^{Ahc_i} q_k + h \cdot \sum_{j=1}^{i} a_{ij} e^{Ah(c_i - c_j)} f(Q_j), \\ q_{k+1} = e^{Ah} q_k + h \cdot \sum_{i=1}^{s} b_i e^{Ah(1-c_i)} f(Q_i). \end{cases} \tag{2.10}$$

where a_{ij} are the coefficients in Table 2.1. This integrator can be represented in terms of the Butcher tableau given in Table 2.2:

Table 2.2. DISEX

$$\begin{array}{c|ccccc} e^{Ahc_1} & \frac{b_1}{2} & 0 & 0 & 0 & 0 \\ e^{Ahc_2} & b_1 e^{Ah(c_2-c_1)} & \frac{b_2}{2} & 0 & 0 & 0 \\ e^{Ahc_3} & b_1 e^{Ah(c_3-c_1)} & b_2 e^{Ah(c_3-c_2)} & \frac{b_3}{2} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ e^{Ahc_s} & b_1 e^{Ah(c_s-c_1)} & b_2 e^{Ah(c_s-c_2)} & b_3 e^{Ah(c_s-c_3)} & \cdots & \frac{b_s}{2} \\ \hline e^{Ah} & b_1 e^{Ah(1-c_1)} & b_2 e^{Ah(1-c_2)} & b_3 e^{Ah(1-c_3)} & \cdots & b_s e^{Ah(1-c_s)} \end{array}$$

It was shown in [67] that any DISRK method can be regarded as the composition of midpoint rules with timesteps $b_1 h, b_2 h, b_3 h, \cdots b_s h$. As such, it is natural to ask whether DISEX method is also the composition of exponential midpoint rules. Notice that (2.1) is autonomous while the transformed system (2.2) is nonautonomous, and our application of DISRK method

to (2.2) involves time explicitly, so the result does not immediately follow from the result of [67]. However, the same conclusion still holds for the DISEX method, as demonstrated by the following theorem:

Theorem 13. *The DISEX method is equivalent to the composition of exponential midpoint rules with timesteps $b_1h, b_2h, b_3h, \cdots b_sh$.*

Proof. The composition of exponential midpoint rules

$$q_k \xrightarrow{b_1h} Z_1 \xrightarrow{b_2h} Z_2 \cdots \xrightarrow{b_sh} Z_s = q_{k+1}$$

is represented as follows,

$$Z_1 = e^{Ahb_1} q_k + b_1 h \cdot e^{Ah\frac{b_1}{2}} f\Big(\frac{e^{Ah\frac{b_1}{2}} q_k + e^{-Ah\frac{b_1}{2}} Z_1}{2}\Big), \tag{M.1}$$

$$Z_2 = e^{Ahb_2} Z_1 + b_2 h \cdot e^{Ah\frac{b_2}{2}} f\Big(\frac{e^{Ah\frac{b_2}{2}} Z_1 + e^{-Ah\frac{b_2}{2}} Z_2}{2}\Big), \tag{M.2}$$

$$\vdots$$

$$Z_i = e^{Ahb_i} Z_{i-1} + b_i h \cdot e^{Ah\frac{b_i}{2}} f\Big(\frac{e^{Ah\frac{b_i}{2}} Z_{i-1} + e^{-Ah\frac{b_i}{2}} Z_i}{2}\Big), \tag{M.i}$$

$$\vdots$$

$$Z_s = e^{Ahb_s} Z_{s-1} + b_s h \cdot e^{Ah\frac{b_s}{2}} f\Big(\frac{e^{Ah\frac{b_s}{2}} Z_{s-1} + e^{-Ah\frac{b_s}{2}} Z_s}{2}\Big). \tag{M.s}$$

Introduce $Q_1 = \frac{e^{Ah\frac{b_1}{2}} q_k + e^{-Ah\frac{b_1}{2}} Z_1}{2}$ in (M.1) as an intermediate variable. Then, on both sides of (M.1), multiply by $e^{-Ah\frac{b_1}{2}}$, add $e^{Ah\frac{b_1}{2}} q_k$, and divide by two, which yields an equivalent form of (M.1),

$$Q_1 = e^{Ah\frac{b_1}{2}} q_k + \frac{b_1}{2} h \cdot f(Q_1). \tag{S.1}$$

For the Runge–Kutta method represented by the Butcher tableau in Table 2.1 to be consistent,

the coefficients have to satisfy,

$$\begin{cases} c_1 = \dfrac{b_1}{2}, \\ c_2 = b_1 + \dfrac{b_2}{2}, \\ \quad \vdots \\ c_i = b_1 + b_2 + \cdots b_{i-1} + \dfrac{b_i}{2}, \\ \quad \vdots \\ c_s = b_1 + b_2 + b_3 + \cdots + b_{s-1} + \dfrac{b_s}{2}, \\ 1 = b_1 + b_2 + b_3 + \cdots + b_{s-1} + b_s. \end{cases}$$

So equation (S.1) coincides with the first line of the Butcher tableau of the DISEX method. Similarly, introduce $Q_2 = \frac{e^{Ah\frac{b_2}{2}}Z_1 + e^{-Ah\frac{b_2}{2}}Z_2}{2}$. Then, on both sides of (M.2), multiply by $e^{-Ah\frac{b_2}{2}}$, add $e^{Ah\frac{b_2}{2}}Z_1$, then divided by two, which yields an equivalent form of (M.2):

$$\begin{aligned} Q_2 &= e^{Ah\frac{b_2}{2}}Z_1 + \frac{b_2}{2}h \cdot f(Q_2) \\ &= e^{Ah(b_1+\frac{b_2}{2})}q_k + b_1 h \cdot e^{Ah(\frac{b_1}{2}+\frac{b_2}{2})}f(Q_1) + \frac{b_2}{2}h \cdot f(Q_2) \\ &= e^{Ahc_2}q_k + b_1 h \cdot e^{Ah(c_2-c_1)}f(Q_1) + \frac{b_2}{2}h \cdot f(Q_2). \end{aligned} \tag{S.2}$$

So equation (S.2) coincides with the second line of the Butcher tableau of the DISEX method. Then, as before, we introduce $Q_i = \frac{e^{Ah\frac{b_i}{2}}Z_{i-1} + e^{-Ah\frac{b_i}{2}}Z_i}{2}$, and apply the same technique to (M.i), which yields

$$Q_i = e^{Ah\frac{b_i}{2}}Z_{i-1} + \frac{b_i}{2}h \cdot f(Q_i).$$

By induction,

$$\begin{aligned} Z_{i-1} = e^{Ah(b_{i-1}+\cdots+b_2+b_1)}q_k + b_1 h \cdot e^{Ah(b_{i-1}+b_{i-2}+\cdots+\frac{b_1}{2})}f(Q_1) \\ + b_2 h \cdot e^{Ah(b_{i-1}+b_{i-2}+\cdots+\frac{b_2}{2})}f(Q_2) + \cdots b_{i-1}h \cdot e^{Ah(\frac{b_{i-1}}{2})}f(Q_{i-1}), \end{aligned}$$

so

$$
\begin{aligned}
Q_i = \; & e^{Ah(\frac{b_i}{2}+b_{i-1}+\cdots+b_2+b_1)}q_k + b_1 h \cdot e^{Ah(\frac{b_i}{2}+b_{i-1}+b_{i-2}+\cdots+\frac{b_1}{2})} f(Q_1) \\
& + b_2 h \cdot e^{Ah(\frac{b_i}{2}+b_{i-1}+b_{i-2}+\cdots+\frac{b_2}{2})} f(Q_2) + \cdots \\
& + b_{i-1} h \cdot e^{Ah(\frac{b_i}{2}+\frac{b_{i-1}}{2})} f(Q_{i-1}) + \frac{b_i}{2} h \cdot f(Q_i) \qquad \text{(S.i)} \\
= \; & e^{Ahc_i} q_k + b_1 h \cdot e^{Ah(c_i - c_1)} f(Q_1) + \cdots \\
& + b_{i-1} h \cdot e^{Ah(c_i - c_{i-1})} f(Q_{i-1}) + \frac{b_i}{2} h \cdot f(Q_i),
\end{aligned}
$$

which coincides with the i-th row of the Butcher tableau of the DISEX method. Finally, we have

$$
\begin{aligned}
q_{k+1} &= Z_s \\
&= e^{Ahb_s} Z_{s-1} + b_s h \cdot f(Q_s) \\
&= e^{Ah(b_s+\cdots b_2+b_1)} q_k + b_1 h \cdot e^{Ah(b_s+\cdots b_2+\frac{b_1}{2})} f(Q_1) + \cdots \\
&\qquad + b_{s-1} h \cdot e^{Ah(b_s+\frac{b_{s-1}}{2})} f(Q_{s-1}) + b_s h \cdot e^{Ah\frac{b_s}{2}} f(Q_s) \\
&= e^{Ah} q_k + b_1 h \cdot e^{Ah(1-c_1)} f(Q_1) + \cdots b_s h \cdot e^{Ah(1-c_s)} f(Q_s),
\end{aligned}
$$

which coincides with the last row of the Butcher tableau of the DISEX method. So the composition of exponential midpoint rules with timesteps $b_1 h, b_2 h, b_3 h, \cdots b_s h$ is equivalent to the DISEX method of Table 2.2. □

Theorem 13 establishes an equivalent relationship between the DISEX method and exponential midpoint rules on the nonautonomous system (2.2) as was established in [67] for the DISRK method and midpoint rules for the autonomous system (2.1), and it also shows that the DISEX method preserves the Poisson structure.

2.3 Energy-preserving Exponential Integrator

Though classical symplectic methods exhibit superior long time stability, it was observed that symplectic schemes are less competitive for the numerical integration of stiff systems with

high frequency. In sharp contrast, energy-preserving methods perform much better [56]. A general way to construct an energy-preserving method for a Poisson system $\dot{q} = J\nabla H(q)$ is the discrete gradient method [49]. We design a discrete gradient $\overline{\nabla} H(q_k, q_{k+1})$ that satisfies the following property,

$$\overline{\nabla} H(q_k, q_{k+1}) \cdot (q_{k+1} - q_k) = H(q_{k+1}) - H(q_k). \tag{2.11}$$

Then, the resulting discrete gradient method is given by,

$$\frac{q_{k+1} - q_k}{h} = J\overline{\nabla} H(q_k, q_{k+1}). \tag{2.12}$$

Multiplying $\overline{\nabla} H(q_k, q_{k+1})$ on both sides of (2.12), we obtain

$$\begin{aligned} H(q_{k+1}) - H(q_k) &= \overline{\nabla} H(q_k, q_{k+1}) \cdot (q_{k+1} - q_k) \\ &= h \cdot \overline{\nabla} H(q_k, q_{k+1}) J \overline{\nabla} H(q_k, q_{k+1}) \\ &= 0. \end{aligned} \tag{2.13}$$

The last equation of (2.13) holds simply due to the skew-symmetric property of matrix J, which implies that discrete gradient method (2.12) preserves energy. We shall combine exponential integrators with the discrete gradient method to obtain an energy-preserving exponential integrator. This approach was initially proposed in [65] for separable Hamiltonian systems using the extended discrete gradient method, and we generalize this to semilinear Poisson systems. Replace the $f(q_k)$ term in the exponential Euler method (2.4) by the discrete gradient $J\overline{\nabla} V(q_k, q_{k+1})$, which yields

$$q_{k+1} = e^{Ah} q_k + \int_0^h e^{A\tau} d\tau \cdot J\overline{\nabla} V(q_k, q_{k+1}). \tag{2.14}$$

Theorem 14. *Method* (2.14) *preserves the Hamiltonian* $H(q)$.

Proof. Let $S = e^{Ah}$, $T = \int_0^h e^{A\tau} d\tau$, then $q_{k+1} = Sq_k + TJ\overline{\nabla} V(q_k, q_{k+1})$. It is beneficial to observe

that the assumption that J and D commute implies that the matrices J, D, A, S, and T all commute with each other. Also, we will show that the following properties hold:

1. $S^{\mathrm{T}} = S^{-1}$;
2. $AT = S - I$;
3. $AT^{\mathrm{T}} = I - S^{\mathrm{T}}$;
4. $S^{\mathrm{T}}T = T^{\mathrm{T}}$.

Property 1 follows from the fact that $S^{\mathrm{T}} = (e^{Ah})^{\mathrm{T}} = e^{-Ah} = S^{-1}$. Property 2 follows from

$$e^{Ah} - I = e^{A\tau}\Big|_0^h = \int_0^h A \cdot e^{A\tau} d\tau = AT.$$

Taking transposes on both sides of Property 2 and using the fact that A and T commute gives Property 3. Property 4 follow from

$$S^{\mathrm{T}}T = e^{-Ah}\int_0^h e^{A\tau} d\tau = \int_0^h e^{-A(h-\tau)} d\tau = \int_0^h e^{-A\tau} d\tau = T^{\mathrm{T}}.$$

From this, we obtain

$$\begin{aligned}
\frac{1}{2}q_{k+1}^{\mathrm{T}} D q_{k+1} &= \frac{1}{2}q_{k+1}^{\mathrm{T}} D(Sq_k + TJ\overline{\nabla}V) \\
&= \frac{1}{2}(Sq_k + TJ\overline{\nabla}V)^{\mathrm{T}} DSq_k + \frac{1}{2}q_{k+1}^{\mathrm{T}} DTJ\overline{\nabla}V \\
&= \frac{1}{2}q_k^{\mathrm{T}} S^{\mathrm{T}} DSq_k + \frac{1}{2}q_k^{\mathrm{T}}(I - S^{\mathrm{T}})\overline{\nabla}V + \frac{1}{2}q_{k+1}^{\mathrm{T}}(S - I)\overline{\nabla}V \\
&= \frac{1}{2}q_k^{\mathrm{T}} Dq_k - (q_{k+1} - q_k)^{\mathrm{T}}\overline{\nabla}V - \frac{1}{2}q_k^{\mathrm{T}}(S^{\mathrm{T}} + I)\overline{\nabla}V + \frac{1}{2}q_{k+1}^{\mathrm{T}}(S + I)\overline{\nabla}V \\
&= \frac{1}{2}q_k^{\mathrm{T}} Dq_k - V(q_{k+1}) + V(q_k) + \frac{1}{2}\overline{\nabla}V^{\mathrm{T}} J^{\mathrm{T}} T^{\mathrm{T}}(S + I)\overline{\nabla}V \\
&= \frac{1}{2}q_k^{\mathrm{T}} Dq_k - V(q_{k+1}) + V(q_k).
\end{aligned} \tag{2.15}$$

Here, we have used the fact that $DTJ = S - I$, $S^{\mathrm{T}}DS = D$, $S^{\mathrm{T}}DTJ = I - S^{\mathrm{T}}$, and $J^{\mathrm{T}}T^{\mathrm{T}}(S + I) = J^{\mathrm{T}}T - JT^{\mathrm{T}}$ which is skew-symmetric. The above calculation demonstrates that $H(q_{k+1}) = H(q_k)$.

□

2.4 Numerical Methods

We will apply our proposed geometric exponential integrators to the semi-discretization of two Hamiltonian PDEs, as the resulting system of differential equations can be expressed in the semilinear form (2.1), with a coefficient matrix A whose spectral radius increases with the spatial resolution of the discretization. The first is the nonlinear Schrödinger equation,

$$i\psi_t + \psi_{xx} - 2|\psi|^2\psi = 0, \tag{2.16}$$

in which $\psi = u + iv$ is the wave function with real part u and imaginary part v, and has the following equivalent form,

$$\begin{cases} u_t = -v_{xx} + 2(u^2+v^2)v, \\ v_t = u_{xx} - 2(u^2+v^2)u. \end{cases} \tag{2.17}$$

The second is the KdV equation,

$$u_t + uu_x + u_{xxx} = 0. \tag{2.18}$$

To discretize the two PDEs, we impose 2π periodic boundary conditions. Given a smooth 2π periodic function $f(x)$, on the interval $[0, 2\pi]$, choose $2n+1$ equispaced interpolation points $x_j = jh$, $j = 0, 1, 2, \ldots, 2n$, $h = \frac{2\pi}{2n+1}$. Given nodal values $\{v_j\}_{j=0}^{2n}$, there exists a unique trigonometric polynomial $v(x)$ with degree less or equal n, such that, $v(x_j) = v_j$ (see, for example, [3]).

$$v(x) = \sum_{k=-n}^{n} \hat{v}_k e^{ikx}, \quad \hat{v}_k = \frac{1}{2n+1}\sum_{j=0}^{2n} v_j e^{-ikx_j}.$$

By substituting the expression for the coefficients $\hat{v}_k$, we obtain

$$v(x) = \sum_{j=0}^{2n} v_j \phi(x - x_j) = \sum_{j=0}^{2n} v_j \phi_j(x), \tag{2.19}$$

where

$$\phi(x) = \sum_{k=-n}^{n} \frac{1}{2n+1} e^{ikx} = \frac{1}{2n+1} \frac{\sin((n+\frac{1}{2})x)}{\sin(\frac{x}{2})}.$$

From this, we see that $\{e^{ikx}\}_{k=-n}^{n}$ and $\{\phi_j\}_{j=0}^{2n}$ are equivalent orthogonal bases for the trigonometric polynomial function space, and each such function can be parametrized by either the nodal values $\{v_j\}_{j=0}^{2n}$ or Fourier coefficients $\{\hat{v}_k\}_{k=-n}^{n}$. They represent the same function, but with respect to two different bases. The transformation between $\{v_j\}_{j=0}^{2n}$ and $\{\hat{v}_k\}_{k=-n}^{n}$ can be performed using the Fast Fourier transformation (FFT), which has $O(n \log n)$ complexity.

The first and second-order differentiation matrices [59] with respect to the representation in terms of nodal values $\{v_j\}_{j=0}^{2n}$ are given by

$$(D_1)_{kj} = \begin{cases} 0, & k = j, \\ \frac{(-1)^{(k-j)}}{2\sin(\frac{(k-j)h}{2})}, & k \neq j, \end{cases}$$

$$(D_2)_{kj} = \begin{cases} -\frac{n(n+1)}{3}, & k = j, \\ \frac{(-1)^{(k-j+1)}\cos(\frac{(k-j)h}{2})}{2\sin^2(\frac{(k-j)h}{2})}, & k \neq j, \end{cases}$$

respectively. However, with respect to the representation in terms of Fourier coefficients $\{\hat{v}_k\}_{k=-n}^{n}$, they are diagonal,

$$\hat{D}_1 = \mathrm{diag}(ik)_{k=-n}^{n}, \quad \hat{D}_2 = \mathrm{diag}(-k^2)_{k=-n}^{n}.$$

We can also define a third-order differentiation matrix D_3, which has the property $D_3 = D_1 D_2 = D_2 D_1$, and it is diagonal with respect to the Fourier coefficients $\hat{D}_3 = \mathrm{diag}(-ik^3)_{k=-n}^{n}$. The

observation that D_1, D_2 and D_3 are all diagonal in the Fourier representation is critical to a fast implementation of the product of matrix functions with vectors.

Theorem 15. *Suppose that D_1, D_2, D_3 are the first, second and third-order differentiation matrices, respectively, $q = \{q_j\}_{j=0}^{2n}$ is a vector with Fourier transform $F[q] = \hat{q} = \{\hat{q}_k\}_{k=-n}^{n}$, and f is an analytic function. Then*

$$f(D_1)q = F^{-1}[\mathrm{diag}(f(ik))\hat{q}], f(D_2)q = F^{-1}[\mathrm{diag}(f(-k^2))\hat{q}], f(D_3)q = F^{-1}[\mathrm{diag}(f(-ik^3))\hat{q}],$$

where F^{-1} is the inverse Fourier transform.

Proof. Recall that the matrix D_2 is diagonalizable with eigenvalues $\lambda_k = -k^2$, and corresponding eigenvectors $e_k = \{e^{ikx_j}\}_{j=0}^{2n}$,

$$\begin{aligned} f(D_2)q = f(D_2)\left(\sum_{k=-n}^{n} \hat{q}_k \cdot e_k\right) &= \sum_{k=-n}^{n} \hat{q}_k \cdot f(D_2)e_k \\ &= \sum_{k=-n}^{n} \hat{q}_k \cdot f(\lambda_k)e_k = F^{-1}[\mathrm{diag}(f(\lambda_k))\hat{q}] = F^{-1}[\mathrm{diag}(f(-k^2))\hat{q}]. \end{aligned}$$

Notice D_1 is also diagonalizable with eigenvalues $\lambda_k = ik$, and corresponding eigenvectors e_k, D_3 is diagonalizable with eigenvalues $\lambda_k = -ik^3$, and corresponding eigenvectors e_k, so the property that $f(D_1)q = F^{-1}[\mathrm{diag}(f(ik))\hat{q}]$, $f(D_3)q = F^{-1}[\mathrm{diag}(f(-ik^3))\hat{q}]$ can be verified in the same way. □

2.4.1 Nonlinear Schrödinger equation

We perform a semi-discretization of (2.17) by discretizing the solution u, v in space using their corresponding nodal values $\{q_j\}$ and $\{p_j\}$. Applying the pseudospectral method, we obtain

the following system of ODEs,

$$\begin{cases} \dot{q} = -D_2 p + 2(q^2+p^2)p, \\ \dot{p} = D_2 q - 2(q^2+p^2)q, \end{cases} \tag{2.20}$$

where the nonlinear term $(q^2+p^2)p$ is computed elementwise, and represents the vector consisting of $\{(q_j^2+p_j^2)p_j\}$ entries. We adopt this notation throughout the rest of the paper for brevity. Then, (2.20) can be expressed as,

$$\frac{d}{dt}\begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix}\begin{pmatrix} q \\ p \end{pmatrix} + \begin{pmatrix} 2(q^2+p^2)p \\ -2(q^2+p^2)q \end{pmatrix}, \tag{2.21}$$

where

$$A = \begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}\begin{pmatrix} -D_2 & 0 \\ 0 & -D_2 \end{pmatrix} = J \cdot D,$$

$$f(q,p) = \begin{pmatrix} 2(q^2+p^2)p \\ -2(q^2+p^2)q \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}\begin{pmatrix} 2(q^2+p^2)q \\ 2(q^2+p^2)p \end{pmatrix} = J \cdot \nabla V(q,p),$$

and $V(q,p) = \frac{1}{2}(q^2+p^2)^2$. It is easy to verify that J is skew-symmetric, D is symmetric, and $JD = DJ$. Thus, (2.21) is a semilinear Poisson system.

To apply the exponential midpoint rule, we need to compute the product of a matrix function and a vector, which has the form $e^{Ah}\begin{pmatrix} q \\ p \end{pmatrix}$,

$$e^{\left(\begin{smallmatrix} 0 & -D_2 \\ D_2 & 0 \end{smallmatrix}\right)} = \sum_{k=0}^{\infty} \frac{1}{k!}\left(\begin{smallmatrix} 0 & -D_2 \\ D_2 & 0 \end{smallmatrix}\right)^k = \left(\begin{smallmatrix} \cos(D_2) & -\sin(D_2) \\ \sin(D_2) & \cos(D_2) \end{smallmatrix}\right).$$

By Theorem 15, we have that

$$
\begin{aligned}
e^{\left(\begin{smallmatrix} 0 & -D_2 \\ D_2 & 0 \end{smallmatrix}\right)h} \begin{pmatrix} q \\ p \end{pmatrix} &= \begin{pmatrix} \cos(D_2 h) & -\sin(D_2 h) \\ \sin(D_2 h) & \cos(D_2 h) \end{pmatrix} \begin{pmatrix} q \\ p \end{pmatrix} \\
&= \begin{pmatrix} \cos(D_2 h) \cdot q - \sin(D_2 h) \cdot p \\ \sin(D_2 h) \cdot q + \cos(D_2 h) \cdot p \end{pmatrix} \\
&= \begin{pmatrix} F^{-1}[\cos(k^2 h)\hat{q}_k + \sin(k^2 h)\hat{p}_k] \\ F^{-1}[\cos(k^2 h)\hat{p}_k - \sin(k^2 h)\hat{q}_k] \end{pmatrix}.
\end{aligned}
\tag{2.22}
$$

In summary, the exponential midpoint rule for the nonlinear Schrödinger equation is given by

$$
z_{k+1} = e^{Ah} z_k + h \cdot e^{A\frac{h}{2}} f\left(\frac{e^{A\frac{h}{2}} z_k + e^{-A\frac{h}{2}} z_{k+1}}{2} \right),
$$

where $z_k = \begin{pmatrix} q_k \\ p_k \end{pmatrix}$, $z_{k+1} = \begin{pmatrix} q_{k+1} \\ p_{k+1} \end{pmatrix}$, $A = \begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix}$, and $e^{Ah} z_k$ can be efficiently calculated using (2.22).

For the energy-preserving exponential integrator for the nonlinear Schrödinger equation,

$$
z_{k+1} = e^{Ah} z_k + \int_0^h e^{A\tau} d\tau \cdot J\overline{\nabla} V(z_k, z_{k+1}).
$$

Here,

$$
\begin{aligned}
\int_0^h e^{\left(\begin{smallmatrix} 0 & -D_2 \\ D_2 & 0 \end{smallmatrix}\right)\tau} d\tau &= \int_0^h \left(\begin{smallmatrix} \cos(D_2\tau) & -\sin(D_2\tau) \\ \sin(D_2\tau) & \cos(D_2\tau) \end{smallmatrix}\right) d\tau \\
&= h \cdot \begin{pmatrix} \frac{\sin(D_2 h)}{D_2 h} & \frac{\cos(D_2 h) - 1}{D_2 h} \\ \frac{1 - \cos(D_2 h)}{D_2 h} & \frac{\sin(D_2 h)}{D_2 h} \end{pmatrix},
\end{aligned}
$$

and by Theorem 15,

$$\int_0^h e^{\begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix}\tau} d\tau \cdot \begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} \frac{\sin(D_2 h)}{D_2} q + \frac{\cos(D_2 h)-1}{D_2} p \\ \frac{1-\cos(D_2 h)}{D_2} q + \frac{\sin(D_2 h)}{D_2} p \end{pmatrix},$$

$$= \begin{pmatrix} F^{-1}[\frac{\sin(k^2 h)}{k^2} \hat{q}_k - \frac{\cos(k^2 h)-1}{k^2} \hat{p}_k] \\ F^{-1}[-\frac{1-\cos(k^2 h)}{k^2} \hat{q}_k + \frac{\sin(k^2 h)}{k^2} \hat{p}_k] \end{pmatrix}.$$

We can construct the discrete gradient

$$\overline{\nabla} V(z_k, z_{k+1}) = \begin{pmatrix} 2({q^2}_{k+\frac{1}{2}} + {p^2}_{k+\frac{1}{2}}) \cdot q_{k+\frac{1}{2}} \\ 2({q^2}_{k+\frac{1}{2}} + {p^2}_{k+\frac{1}{2}}) \cdot p_{k+\frac{1}{2}} \end{pmatrix},$$

where

$$q_{k+\frac{1}{2}} = \frac{q_k + q_{k+1}}{2}, \quad p_{k+\frac{1}{2}} = \frac{p_k + p_{k+1}}{2}, \tag{2.23}$$

$${q^2}_{k+\frac{1}{2}} = \frac{q_k^2 + q_{k+1}^2}{2}, \quad {p^2}_{k+\frac{1}{2}} = \frac{p_k^2 + p_{k+1}^2}{2}. \tag{2.24}$$

Notice that $\overline{\nabla} V(z_k, z_{k+1})$ is symmetric with respect to z_k and z_{k+1}, and it can be verified that it satisfies (2.11). A classical discrete gradient method can be constructed as follows,

$$z_{k+1} = z_k + hJ\overline{\nabla} H(z_k, z_{k+1}), \tag{2.25}$$

where

$$\overline{\nabla} H(z_k, z_{k+1}) = \begin{pmatrix} -D_2 & 0 \\ 0 & -D_2 \end{pmatrix} \frac{z_k + z_{k+1}}{2} + \overline{\nabla} V(z_k, z_{k+1}).$$

The method described by (2.25) is very similar to classical midpoint rule, the only difference is in $\overline{\nabla} V(z_k, z_{k+1})$, ${q^2}_{k+\frac{1}{2}}$ is used, while $(q_{k+\frac{1}{2}})^2$ is used in the midpoint rule, so (2.25) can be viewed as a modified midpoint rule.

2.4.2 KdV equation

Rewrite (2.18) as

$$u_t = \Big(-\frac{\partial}{\partial x}\Big)\Big(\frac{1}{2}u^2 + u_{xx}\Big),$$

then apply pseudospectral semi-discretization to obtain the following system,

$$\begin{aligned}\dot{q} &= (-D_1)\Big(\frac{1}{2}q^2 + D_2 q\Big)\\ &= (-D_1)D_2 q + (-D_1)\Big(\frac{1}{2}q^2\Big),\end{aligned}$$

which has the form of a semilinear Poisson system (2.5),

$$\dot{q} = J(Dq + \nabla V(q)) = J\nabla H(q),$$

where $J = -D_1, D = D_2, A = JD = -D_3, \nabla V(q) = \frac{1}{2}q^2, H(q) = \frac{1}{2}q^{\mathrm{T}}D_2 q + \frac{1}{6}q^3$. The exponential midpoint rule for KdV reads as follows,

$$q_{k+1} = e^{-D_3 h}q_k + h \cdot e^{-D_3\frac{h}{2}} f\Big(\frac{e^{-D_3\frac{h}{2}}q_k + e^{D_3\frac{h}{2}}q_{k+1}}{2}\Big),$$

and the energy preserving exponential integrator is given by,

$$q_{k+1} = e^{-D_3 h}q_k + \int_0^h e^{-D_3\tau}d\tau \cdot (-D_1)\overline{\nabla}V(q_k, q_{k+1}),$$

with discrete gradient $\overline{\nabla}V(q_k, q_{k+1}) = \frac{1}{6}(q_k^2 + q_k \cdot q_{k+1} + q_{k+1}^2)$. A related classical discrete gradient method can be constructed as follows,

$$\overline{\nabla}H(q_k, q_{k+1}) = (D_2)\frac{q_k + q_{k+1}}{2} + \overline{\nabla}V(q_k, q_{k+1}). \tag{2.26}$$

By Theorem 15, in each iteration, the matrix function and vector product can be imple-

mented as

$$e^{-D_3 h} q = F^{-1}[e^{ik^3 h} \hat{q}_k],$$

and

$$\left(\int_0^h e^{-D_3 \tau} d\tau \right) q = \left(\frac{e^{-D_3 h} - I}{-D_3} \right) q = F^{-1} \left[\frac{e^{ik^3 h} - 1}{ik^3} \hat{q}_k \right].$$

2.4.3 Remarks

We have not analyzed the stiff order and long-time behavior of the methods proposed in this paper. In particular, it should be observed that the standard backward error analysis results for symplectic integration of Hamiltonian systems do not apply to the highly oscillatory systems that we are considering here. A rigorous proof of long-time energy stability will likely involve modulated Fourier expansions, as in [11, 13].

2.5 Numerical Experiments

2.5.1 Nonlinear Schrödinger equation

In Table 2.3 below, n denotes the number of nodes we discretize the spatial domain with, and we tabulate the maximum timestep that the nonlinear solver converges for, and the number of iterations taken to converge when averaged over the first thousand timesteps. We consider the convergence properties of fixed point iteration and Newton type iteration for the midpoint rule, and the fixed point iteration for the exponential midpoint rule, discrete gradient method (2.25), energy-preserving exponential integrator, and diagonally implicit symplectic exponential (DISEX) integrator with six stages [32]:

$$b_1 = 0.5080048194000274 \quad b_2 = 1.360107162294827 \quad b_3 = 2.019293359181722$$

$$b_4 = 0.5685658926458250 \quad b_5 = -1.459852049586439 \quad b_6 = -1.996119183935963.$$

The underlying symplectic diagonally implicit Runge-Kutta method has algebraic order 5, and due to the conjugacy between (2.1) and (2.2), the corresponding DISEX integrator also has order 5. The midpoint exponential integrator, discrete gradient integrator and energy preserving exponential integrator, are all implemented using fixed point iteration, while DISEX is implemented as the composition of midpoint exponential integrators. For the Newton iteration, since each iteration requires the LU factorization of the Jacobian, which is computationally prohibitive, we instead implemented a Quasi-Newton method, i.e., the Jacobian of the initial point is used repeatedly until convergence, so the LU factorization need only be computed once per timestep. Since for numerical integration of ODEs, the initial point is well approximated by a high order explicit method, this technique works well in practice.

Table 2.3. Maximum timestep and average iteration number for convergence for the nonlinear Schrödinger equation, as a function of the numerical integrator, nonlinear solver, and spatial resolution.

	midpoint				midpoint exp		discrete gradient		energy exp		DISEX	
	fixed point		Newton		fixed point		fixed point		fixed point		fixed point	
n	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$
11	0.02	14.2	0.1	8.3	0.1	12.6	0.02	14	0.1	11.9	0.1	101.4
21	0.01	12.9	0.1	8.3	0.08	12.6	0.01	11.8	0.1	12	0.03	47.7
41	4×10^{-3}	7.9	0.1	8.3	0.06	10.9	4×10^{-3}	5.8	0.1	12	0.03	49.2
61	2×10^{-3}	2.1	0.1	8.3	0.04	8.1	2×10^{-3}	4	0.1	12	0.025	44.5
81	10^{-3}	4.1	0.1	8.3	0.04	8.1	10^{-3}	3	0.1	12	0.025	44.6
121	5×10^{-4}	13.9	0.1	8.3	0.04	8.1	5×10^{-4}	3.4	0.1	12	0.01	43.3
161	2×10^{-4}	3.9	0.1	8.3	0.01	5	2×10^{-4}	2	0.1	12	0.01	30.5
201	1×10^{-4}	3.08	0.1	8.3	8×10^{-3}	4.7	1×10^{-4}	2	0.1	12	8×10^{-3}	30
401	4×10^{-5}	2	0.1	8.3	5×10^{-3}	4.9	4×10^{-5}	1	0.1	12	4×10^{-3}	32

If the timestep is too large, the nonlinear solver fails to converge, the number of required iterations diverges, or the numerical accuracy of the solution degrades dramatically. Here, we have chosen the maximum timestep by considering both the long time stability and the number of iterations necessary. Since DISEX is implemented as the composition of exponential midpoint rules, the number of iterations reported is the sum of number of iterations for each component exponential midpoint rule.

We observe that when the midpoint rule is implemented using fixed point iterations, there is a significant decay in the allowable timestep as the spatial resolution is increased, whereas the Newton type iteration allows a relatively large timestep that is independent of the spatial resolution. In contrast, the midpoint exponential method exhibits a slower rate of decrease in allowable timestep when using fixed point iterations. When using fixed point iterations, the discrete gradient method, which is an energy preserving method, the allowable timestep behaves similarly to the midpoint rule, and in contrast, the energy preserving exponential integrator has an allowable timestep that is independent of the spatial resolution.

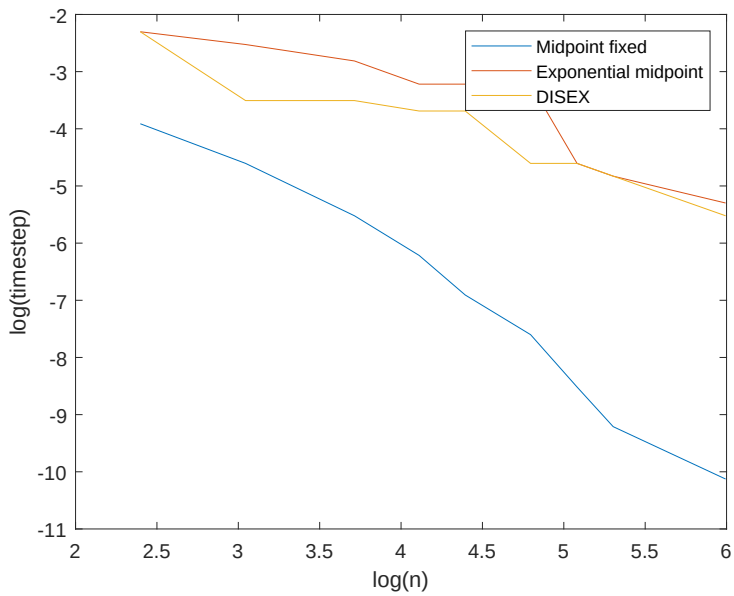
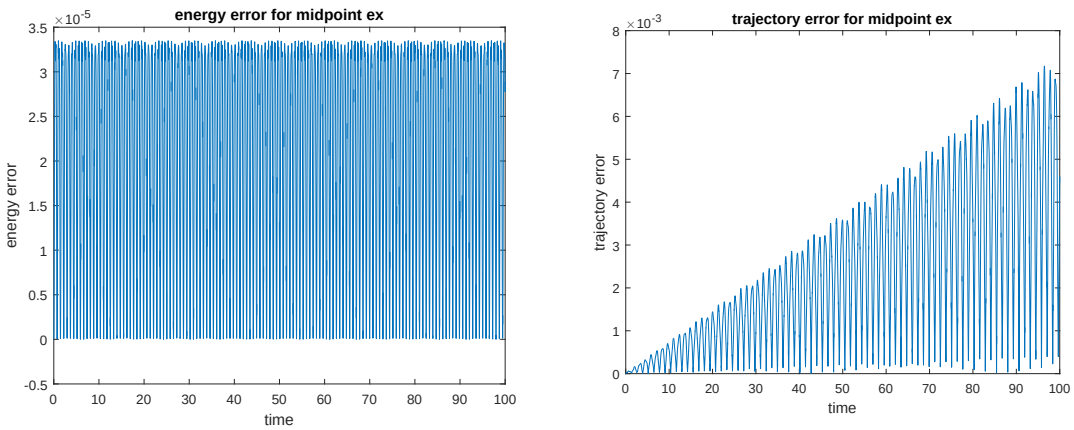


Figure 2.1. Maximum timestep for which fixed point iterations converge as a function of the spatial resolution for the nonlinear Schrödinger equation.

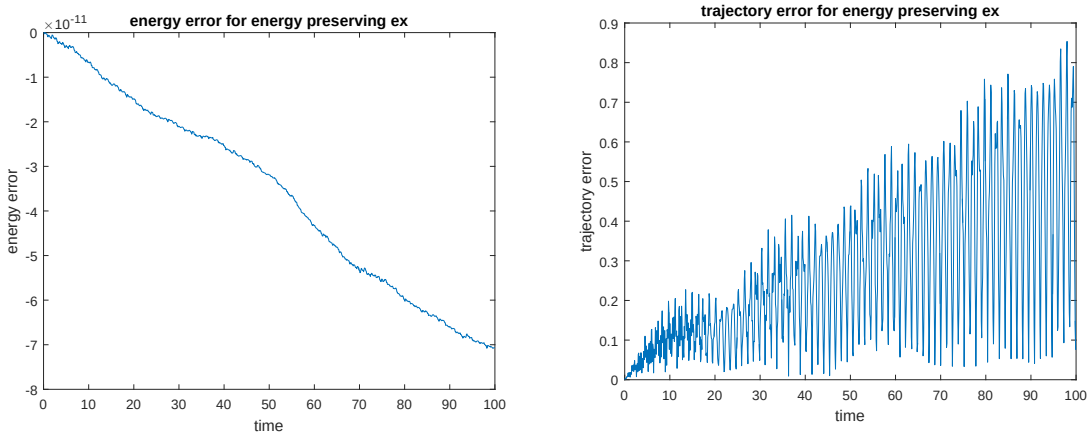
In Figure 2.1, we observe that the allowable timestep when using fixed point iteration scale like n^{-2} for the classical midpoint rule, and n^{-1} for both the midpoint exponential rule and DISEX. As shown in Figure 2.2, the exponential midpoint rule exhibits an energy error that remains small and bounded, which is consistent with it being a symplectic integrator, and the trajectory error grows linearly.

The energy preserving exponential integrator is designed to preserve energy exactly, so



(a) Energy error **(b)** Trajectory error

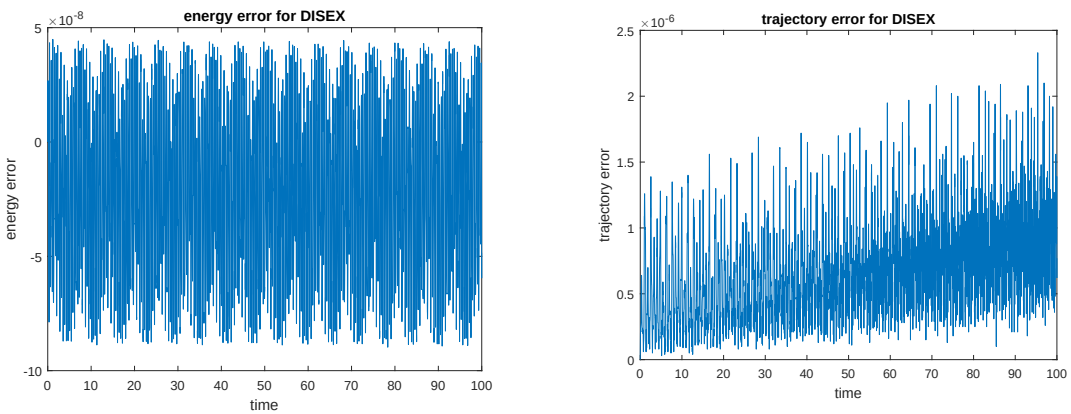
Figure 2.2. Error plots for the exponential midpoint rule applied to the nonlinear Schrödinger equation, $n = 161$, $h = 0.01$.



(a) Energy error **(b)** Trajectory error

Figure 2.3. Error plots for the energy preserving exponential integrator applied to the nonlinear Schrödinger equation, $n = 161$, $h = 0.1$.

even when the timestep is 0.1, we see in Figure 2.3 that the energy is still preserved approximately to within machine error. For DISEX, the energy and trajectory error is shown in Figure 2.4. Observe that the energy error is small and bounded, as expected of a symplectic integrator, and the trajectory error is small as well, as expected of a higher-order numerical integrator.



(a) Energy error **(b)** Trajectory error

Figure 2.4. Error plots for the 6 stage DISEX applied to the nonlinear Schrödinger equation, $n = 161$, $h = 0.01$.

In general, the number of iterations decrease as the timestep decreases, which is not surprising since the initial guess is typically more accurate as the timestep decreases. For a fixed time interval, the total computational cost depends on the number of timesteps, average number of iterations per timestep, and the algorithmic complexity of each iteration. To illustrate the relative computational efficiency of the various methods with different solvers, we set $n = 161$, and plot the trajectory error over the time interval $[0, 1]$ vs. total computational cost in Figure 2.5. As can be seen from the figure, for the same trajectory accuracy, the midpoint rule with Newton iterations is the most computationally expensive, due to the higher complexity of the Newton iterations. However, the minimum computational cost achievable using the midpoint rule with fixed point iterations and the discrete gradient methods is rather high, as they require the use of smaller timesteps in order to converge. This problem becomes more severe for non-exponential integrators as the spatial resolution n increases, as the condition number of the matrix A increases

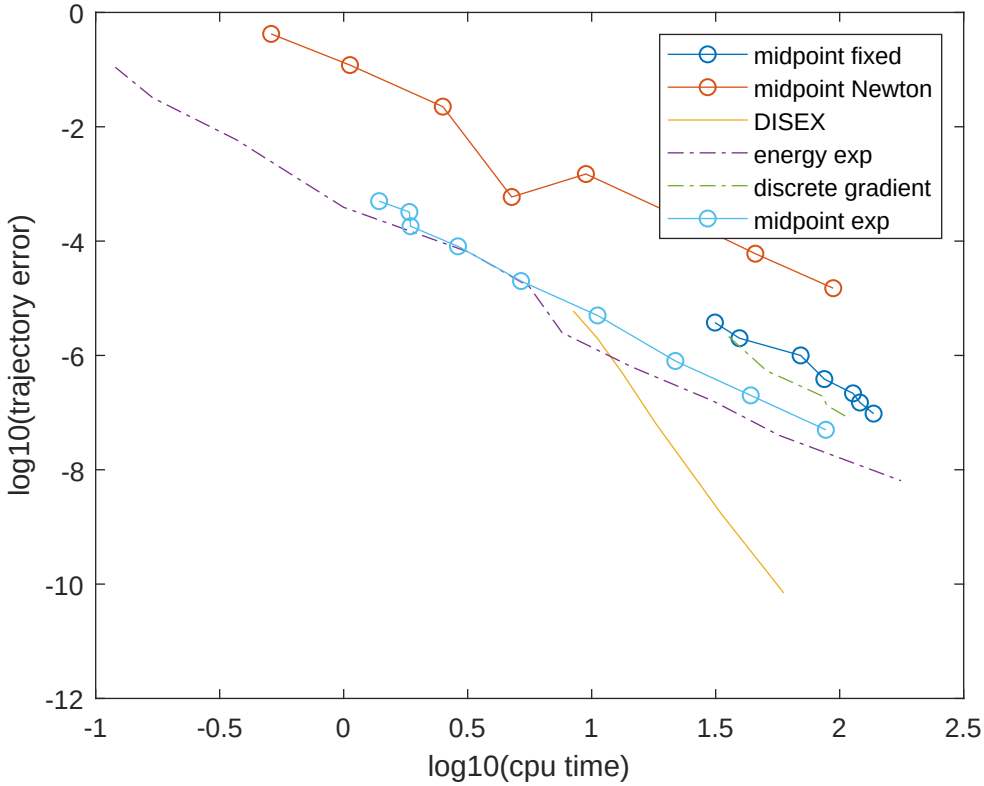


Figure 2.5. Comparison of trajectory error vs. CPU time for the nonlinear Schrödinger equation

with n. In contrast, the minimum computational cost achievable by the midpoint exponential and energy exponential methods is lower, and their trajectory error is also smaller for the same computational cost. For DISEX, the steeper slope indicates that it is a higher-order method, and it is the most economical choice when high accuracy is required.

2.5.2 KdV

We simulate the KdV equation,

$$u_t + uu_x + \nu u_{xxx} = 0,$$

where $\nu = 5 \times 10^{-4}$. In Table 2.4, n denotes the number of nodes used to discretize the spatial domain, and we tabulate the maximum timestep for which the nonlinear solver converges, and

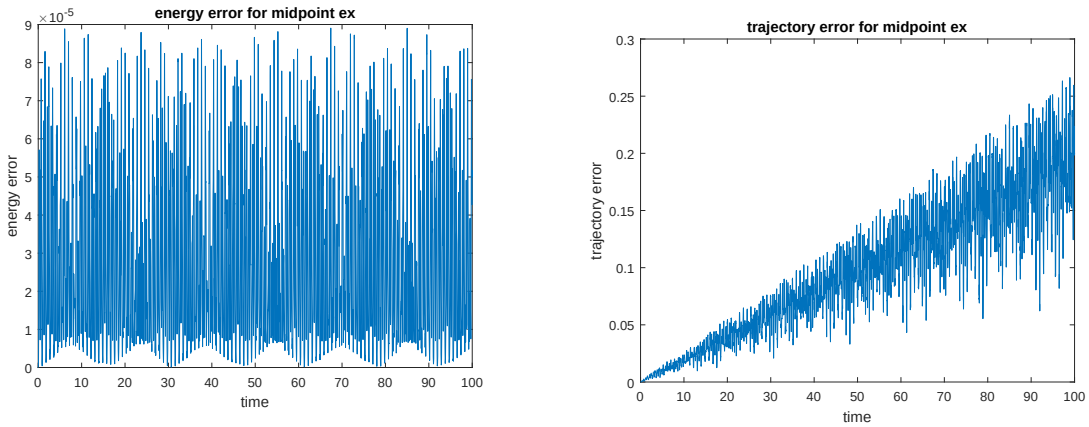
Table 2.4. Maximum timestep and average iteration number for convergence for the KdV equation, as a function of the numerical integrator, nonlinear solver, and spatial resolution.

	midpoint				midpoint exp		discrete gradient		energy exp		DISEX	
	fixed point		Newton		fixed point		fixed point		fixed point		fixed point	
n	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$
401	4×10^{-4}	17.9	4×10^{-4}	25.5	8×10^{-4}	8.4	4×10^{-4}	15.4	0.005	14.9	4×10^{-4}	26.9
601	1×10^{-4}	4.2	1×10^{-4}	7.3	6×10^{-4}	8.6	1×10^{-4}	3.4	0.005	14.8	4×10^{-4}	48.0
801	6×10^{-5}	18.5	5×10^{-5}	4.4	6×10^{-4}	9.4	5×10^{-5}	3.5	0.005	14.9	1×10^{-4}	19
1001	3×10^{-5}	7.7	3×10^{-5}	11.0	6×10^{-4}	10.6	3×10^{-5}	14.9	0.005	14.8	1×10^{-4}	19
1201	1×10^{-5}	2	1×10^{-5}	2	6×10^{-4}	11.4	1×10^{-5}	1	0.005	14.8	1×10^{-4}	22
1401	1×10^{-5}	1	1×10^{-5}	2	6×10^{-4}	16.1	1×10^{-5}	1	0.005	14.8	1×10^{-4}	34

average number of iterations taken to converge. In particular, we explore the effect of the numerical integrator, the nonlinear solver, and the spatial resolution of the semi-discretization, on the convergence properties of the solver. The maximum timestep is chosen by considering both the long time stability and the number of iterations necessary, and the number of iterations is averaged over the first one thousand timesteps. For the midpoint rule, the maximum timestep decreases like n^{-3} for fixed point iterations, and a comparable timestep is required for the Newton iterations to converge, which renders it too costly in practice. It is interesting to compare this with the convergence properties of the NLS problem, where the maximum timestep for the midpoint rule with fixed point iteration decreases like n^{-2}. This difference is due to the fact that for NLS, the second order derivative term ψ_{xx} introduces a D_2 term in the semi-discrete ODE, and its spectral radius increases quadratically; whereas for KdV, u_{xxx} introduces a D_3 term in the semi-discrete ODE, and its spectral radius increases cubicly. The classical discrete gradient method for the KdV equation is given by (2.26), and it exhibits the same timestep restrictions as the midpoint rule. In contrast, both the exponential midpoint and energy preserving exponential integrator allow rather large timesteps that are independent of the spatial resolution. Even though DISEX is implemented as the composition of exponential midpoint rules, it requires smaller timesteps than a single exponential midpoint rule.

In Figure 2.6, we observe that the exponential midpoint rule has an energy error that is small and bounded, as is typical for a symplectic integrator, and the trajectory error grows linearly.

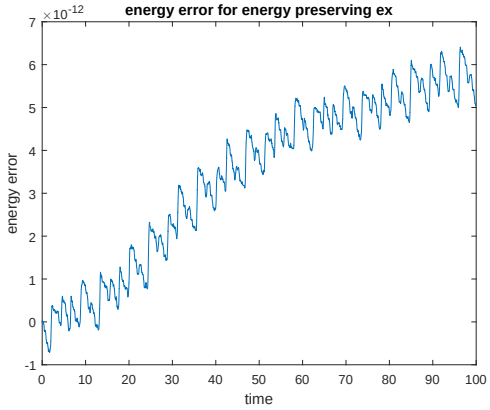
In Figure 2.7, the energy preserving exponential integrator has an energy that is preserved to within machine precision, and the trajectory error grows linearly.



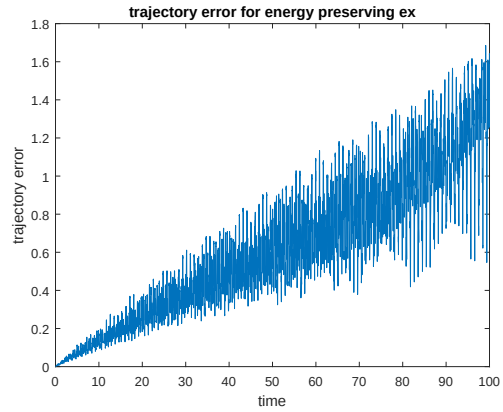
(a) Energy error **(b)** Trajectory error

Figure 2.6. Error plots for the exponential midpoint rule applied to the KdV equation, $n = 401$, $h = 5 \times 10^{-4}$.

As in the NLS case, we explore the relative computational efficiency of the above algorithms by setting $n = 1001$, and plotting the trajectory error over the time interval $[0, 1]$ vs. the total computational cost in Figure 2.8. We did not implement the Newton based solvers here, as we extrapolated that it would require approximately 4×10^4 seconds to compute the trajectory over the time interval $[0, 1]$, so they are not included in the figure. As can be seen from the graph, the minimum computational cost achievable by the energy exponential method is the lowest of the methods considered, as it is capable of taking the largest timestep stably. In addition, it has the highest accuracy for a given computational cost amongst all the low order methods. The method with the next lowest minimum computational cost is the exponential midpoint rule, followed by the midpoint rule with fixed point iteration, and the discrete gradient methods. As before, DISEX still has the steepest slope, which indicates that it is a higher-order method. But, the disadvantage is that the maximum possible timestep is not as large this time, which results in it having the largest minimum computational effort amongst all the methods considered.



(a) Energy error



(b) Trajectory error

Figure 2.7. Error plots for the energy preserving exponential integrator applied to the KdV equation, $n = 401$, $h = 0.005$.

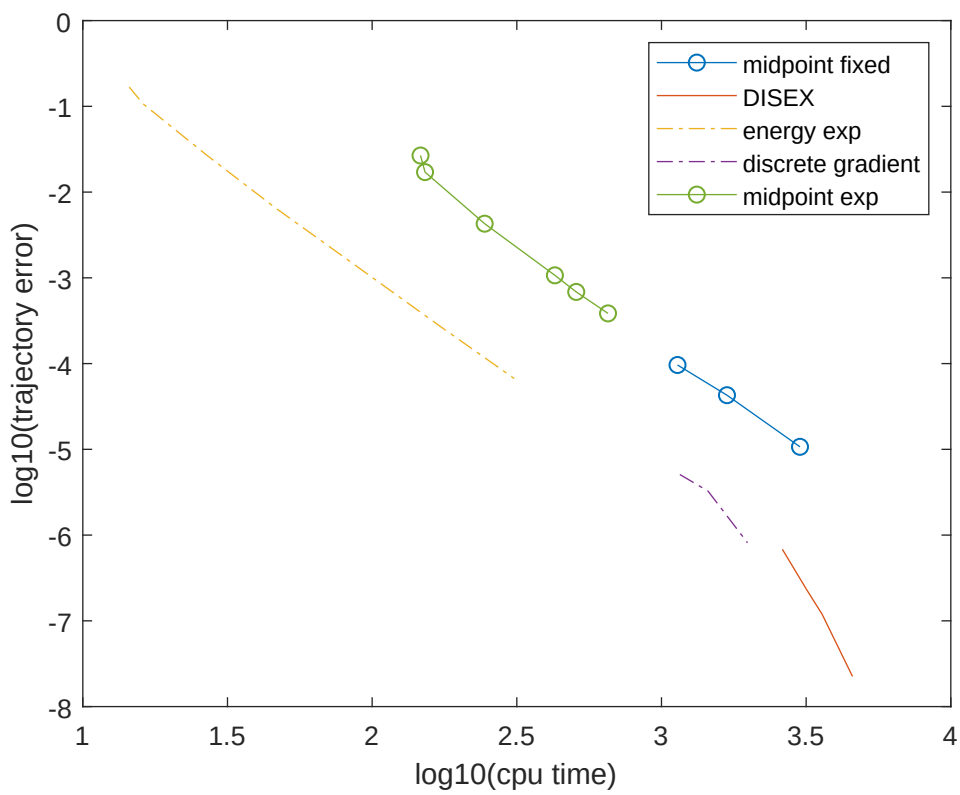


Figure 2.8. Comparison of trajectory error vs. CPU time for the KdV equation

Chapter 2, in full, is a reprint of the material that has been accepted for publication by Journal of Computational Physics, 2019. Xuefeng Shen; Melvin Leok, Elsevier Press, 2019. The dissertation author was the primary investigator and author of this material.

Chapter 3

Lie Group Variational Integrators for Rigid Body Problems using Quaternions

3.1 Introduction

For Lagrangian mechanics on the tangent bundle TQ and the Lagrangian L, the theory of variational integrators is well-established [43], and is based on discretizing Hamilton's principle rather than the Euler–Lagrange equations. The discrete Lagrangian $L_d : Q \times Q \to \mathbb{R}$ approximates the action integral over a small time interval,

$$L_d(q_0, q_1) \approx_{q(0)=q_0, q(h)=q_1} \int_0^h L(q(t), \dot{q}(t))dt,$$

and the discrete Hamilton's principle states that the discrete action sum is stationary with respect to variations in the discrete solution curve that fixes the endpoints,

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}) = 0,$$

where $\delta q_0 = \delta q_N = 0$. This yields the discrete Euler–Lagrange equations,

$$D_2 Ld(q_0, q_1) + D_1(q_1, q_2) = 0. \tag{3.1}$$

By introducing the discrete Legendre transforms, we obtain a variational integrator on the cotangent bundle T^*Q,

$$\begin{cases} p_0 = -D_1 L_d(q_0, q_1), \\ p_1 = D_2 L_d(q_0, q_1). \end{cases} \tag{3.2}$$

It is easy to easy that (3.2) recovers (3.1) when the momentum variables are eliminated, and that (3.2) is the usual characterization of a symplectic map in terms of a Type I generating function. As such, variational integrators are automatically symplectic, and either form of the discrete equations can be easily implemented in a linear space or in local coordinates. However, the configuration manifold Q is not, in general, a Euclidean space. Symplectic integrators do not exactly preserve energy, but instead rely on the existence of an associated modified energy for their long-time energy stability properties.

Constructing a symplectic integrator naïvely on a local coordinate chart typically does not result in a globally well-defined symplectic map on the Lie group. That is to say that if we constructed variational integrators on two local coordinate charts that overlap, they will not generally correspond to the same integrator on the overlapping region. A consequence of this is that changing the coordinate chart at each time step results in the modified energy that is preserved changing from step to step, which destroys the long-time energy stability typically associated with symplectic integrators. In order to overcome this limitation, it is necessary to consider symplectic integrators that is are well-defined globally on the configuration manifold, and which commute with a change of coordinates.

One way to address this issue is to embed Q in Euclidean space. In many cases, the configuration space naturally lives in a higher-dimensional Euclidean space $\mathbb{R}^n$, and is the level set of some constraint function $g : \mathbb{R}^n \to \mathbb{R}^m$. So the position and velocity are represented as vectors in the embedding space, and Lagrange multipliers are introduced to represent the constraint forces.

When the configuration space Q has a high codimension in the Euclidean space, and

the constraint function is complicated, the above method is costly to implement. For example, the rotation group $SO(3)$, which is a three-dimensional Lie group, naturally lives in $\mathbb{R}^{3\times 3}$, a nine-dimensional Euclidean space, with six orthonormality constraints. As opposed to viewing Q extrinsically as an embedded submanifold, when Q is a Lie group, the group structure could be utilized to represent the position and tangent space intrinsically, thus avoiding the introduction of Lagrange multipliers. This idea was used in [35] for the rigid body problem on the configuration space $SO(3)$, and later generalized to the full body problem on $SE(3)$ [36].

Besides the rotation group $SO(3)$, the motion of rigid bodies can also be described by unit quaternions, which is an extremely popular approach in classical mechanics [23], computer graphics [15, 62], virtual reality [33], and engineering applications in control [61, 66] and estimation [54, 55] of aerial and space vehicles. In [64], a variational integrator for the rigid body problem was derived in terms of unit quaternions embedded in $\mathbb{R}^4$, so as to avoid the complicated constraints in $SO(3)$, but where the unit quaternion constraint is enforced using a Lagrange multiplier. We remark that the unit quaternions S^3, is not only the unit sphere in $\mathbb{R}^4$, but also a Lie group under quaternion multiplication.

In this paper, we will derive a Lie group variational integrator for rigid body problems using unit quaternions, but without the use of unit length constraints. Instead, we use the fact that $\mathbb{R}^3$ can be identified with the Lie algebra associated with $S^3 \subset \mathbb{H}$. This will allow us to obtain a variational integrator for rigid body dynamics in the unit quaternion representation that is globally valid, expresses the relative rotation using three degrees-of-freedom, and avoids the need for explicit constraints. While the problem of constructing arbitrarily high-order Lie group variational integrators has been addressed in numerous publications [7, 8, 19], the goal of this paper is more focused, which is to derive a simple, easy to implement, second-order accurate symplectic Lie group method that uses the unit quaternion representation without the need for constraints, local coordinates, or reprojection.

This yields a numerical method in the unit quaternion representation that exhibits the computational advantages of Lie group variational integrators developed for rotation matrices,

thereby making the advantages of geometric integrators accessible to engineering applications which are heavily invested in the unit quaternion representation of the rotation group. In particular, because many existing on board controllers for aerial and space vehicles use the unit quaternion representation, our Lie group variational integrator that natively uses the unit quaternion representation can be easily retrofitted into an existing controller framework without an extensive rewrite of the controller software.

3.2 Background

3.2.1 Rigid body problem

The configuration space of the rigid body is $SE(3) = \mathbb{R}^3 \rtimes SO(3)$, where $(x,R) \in SE(3)$ describes the position and orientation of the body-fixed frame with respect to an inertial frame. Let $\rho \in \mathbb{R}^3$ denotes the position of a mass element in the body-fixed frame. Then, the corresponding inertial position is $x + R\rho$, the velocity is $\dot{x} + \dot{R}\rho$, and the kinetic energy is given by

$$\begin{aligned} T &= \frac{1}{2}\int_{\mathscr{B}} \|\dot{x} + \dot{R}\rho\|^2 dm \\ &= \frac{1}{2}\int_{\mathscr{B}} \|\dot{x}\|^2 dm + \int_{\mathscr{B}} \dot{x}^{\mathrm{T}}\dot{R}\rho dm + \frac{1}{2}\int_{\mathscr{B}} \|\dot{R}\rho\|^2 dm \\ &= \frac{1}{2}m\|\dot{x}\|^2 + \frac{1}{2}\mathrm{tr}[\dot{R}J_d\dot{R}^{\mathrm{T}}], \end{aligned} \tag{3.3}$$

where $J_d = \int_{\mathscr{B}} \rho\rho^{\mathrm{T}} dm$ is the nonstandard moment of inertia matrix. In (3.3), we used the fact that the origin of the body-fixed frame is the center of mass of the body, thus

$$\int_{\mathscr{B}} \rho dm = 0.$$

Since R evolves on $SO(3)$, it follows that $R^{\mathrm{T}}R = I$. By differentiating on both sides, we have $\dot{R}^{\mathrm{T}}R + R^{\mathrm{T}}\dot{R} = 0$, thus $R^{\mathrm{T}}\dot{R}$ is skew symmetric. Define the hat map $\widehat{\cdot}: \mathbb{R}^3 \to R^{3\times 3}$ by the condition

that $\widehat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$. If $x = (x_1, x_2, x_3)$, then $\widehat{x}$ is given by

$$\widehat{x} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix},$$

which is a skew-symmetric matrix. In particular, the hat map defines an isomorphism between $\mathbb{R}^3$ and skew-symmetric matrices. Since $R^{\mathrm{T}}\dot{R}$ is skew-symmetric, there exists a unique vector Ω such that $R^{\mathrm{T}}\dot{R} = \widehat{\Omega}$. In other words,

$$\dot{R} = R\widehat{\Omega}, \tag{3.4}$$

where Ω is the angular velocity in the body-fixed frame. This equation has a natural geometric interpretation: $SO(3)$ is a Lie group under matrix multiplication, and its Lie algebra, $\mathfrak{so}(3)$, consists of all skew-symmetric matrices,

$$\mathfrak{so}(3) = \{\widehat{\Omega} \mid \Omega \in \mathbb{R}^3\}.$$

The tangent space at each R can be represented as the pushforward by the left action of R on the Lie algebra,

$$T_R SO(3) = (L_R)_* \mathfrak{so}(3).$$

In this way, the tangent bundle $TSO(3)$ is diffeomorphic to $SO(3) \times \mathfrak{so}(3)$ by left-trivialization. With this, we can express the rotational term in the kinetic energy as follows,

$$\begin{aligned} \mathrm{tr}[\dot{R} J_d \dot{R}^{\mathrm{T}}] &= \mathrm{tr}[R \widehat{\Omega} J_d \widehat{\Omega}^{\mathrm{T}} R^{\mathrm{T}}] \\ &= \mathrm{tr}[\widehat{\Omega} J_d \widehat{\Omega}^{\mathrm{T}}] \\ &= \Omega^{\mathrm{T}} J \Omega, \end{aligned} \tag{3.5}$$

where $J = \mathrm{tr}[J_d] I_3 - J_d$ is standard moment of inertia matrix. Combining (3.3) and (3.5), we obtain $T = \frac{1}{2} m \|\dot{x}\|^2 + \frac{1}{2} \Omega^{\mathrm{T}} J \Omega$, and the Lagrangian for rigid body problems $L : TSE(3) \to R$ is given by,

$$L(x, R, \dot{x}, \Omega) = T - V = \frac{1}{2} m \|\dot{x}\|^2 + \frac{1}{2} \Omega^{\mathrm{T}} J \Omega - V(x, R).$$

3.2.2 Unit quaternions

Quaternions are a noncommutative division algebra, usually denoted by $\mathbb{H}$, with an element $q = (q_s, \vec{q}_v)$, where q_s is the scalar part and $\vec{q}_v \in \mathbb{R}^3$ is the vector part. $\mathbb{H}$ inherits the vector space structure and differential structure from $\mathbb{R}^4$, and has the following multiplication: given $q = (q_s, \vec{q}_v)$, $p = (p_s, \vec{p}_v)$,

$$q \cdot p = (q_s \cdot p_s - \vec{q}_v \cdot \vec{p}_v, q_s \cdot \vec{p}_v + p_s \cdot \vec{q}_v + \vec{q}_v \times \vec{p}_v).$$

We can also define conjugation $q^* = (q_s, -\vec{q}_v)$, norm $\|q\| = \sqrt{qq^*}$, and inverse under quaternion multiplication $q^{-1} = \frac{q^*}{\|q\|^2}$. The set of unit quaternions

$$S^3 = \{q \in \mathbb{H} \mid \|q\| = 1\}$$

is diffeomorphic to the unit sphere in $\mathbb{R}^4$, and forms a Lie group under quaternion multiplication. The identity element is $e = (1,0,0,0)$, and the Lie algebra

$$T_e S^3 = \{(0,\xi) \mid \xi \in \mathbb{R}^3\}.$$

The corresponding exponential map is

$$\exp(\xi) = \left(\cos(|\xi|), \frac{\xi}{|\xi|}\sin(|\xi|)\right). \tag{3.6}$$

For any $q_0 \in S^3$, the map $\xi \mapsto q_0 \cdot \exp(\xi)$ gives a local diffeomorphism between a neighborhood of the origin of $\mathbb{R}^3$ and a neighborhood of q_0. By identifying $\mathbb{R}^3$ with the Lie algebra, this reduces to canonical coordinates of the first kind on a Lie group [60]. It can be verified that for $q \in S^3, v \in \mathbb{R}^3$, $q(0,v)q^*$ is pure imaginary, i.e., the scalar part vanishes. This defines a linear action on the vector part, which turns out to be a rotation. Thus, for each $q \in S^3$, we have the corresponding rotation $\pi(q) \in SO(3)$,

$$S^3 \xrightarrow{\pi} SO(3), \qquad \pi(q) = (2q_s^2 - 1)I_3 + 2\vec{q}_v\vec{q}_v^{\mathrm{T}} + 2q_s\widehat{\vec{q}_v}.$$

The map π is surjective, locally diffeomorphic, and is also a Lie group homomorphism:

$$\pi(q_1 \cdot q_2) = \pi(q_1) \cdot \pi(q_2),$$

where the operation on left side is quaternion multiplication, and the operation on the right side is matrix multiplication. π is not a global diffeomorphism, for each $R \in SO(3)$, its preimage is always one pair of antipodal points $\pm q$. Actually, S^3 and $SO(3)$ are not diffeomorphic, as they have different homology groups.

We can lift the Lagrangian dynamics on $TSE(3)$ to $T\mathbb{R}^3 \times TS^3$:

$$T\mathbb{R}^3 \times TS^3 \xrightarrow{Id \times T\pi} TSE(3) \xrightarrow{L} \mathbb{R},$$

As manifolds, we have $TSE(3) = T(\mathbb{R}^3 \times SO(3)) = T\mathbb{R}^3 \times TSO(3)$, where Id is the identity map from $T\mathbb{R}^3$ to itself, and $T\pi : TS^3 \to TSO(3)$ is the tangent lift of π. Define the Lagrangian $\hat{L} = L \circ (Id \times T\pi)$ on $T(\mathbb{R}^3 \times S^3)$, and we will develop our algorithm for the Lagrangian mechanics defined by $\hat{L}$ on $T(\mathbb{R}^3 \times S^3)$. Recall that $Id \times T\pi$ is a local diffeomorphism, so $(T(\mathbb{R}^3 \times S^3), \hat{L})$ and $(TSE(3), L)$ are equivalent for initial-value problems, and locally equivalent for two point boundary-value problems. A calculation shows that,

$$(q, q \cdot (0, \xi)) \xrightarrow{T\pi} (\pi(q), \pi(q) \cdot 2\xi),$$

so

$$\begin{aligned} \hat{L}(x, q, \dot{x}, \dot{q}) &= \hat{L}(x, q, \dot{x}, q \cdot (0, \xi)) \\ &= \hat{L}(x, q, \dot{x}, \xi) \\ &= \frac{1}{2} m \|\dot{x}\|^2 + 2\xi^{\mathrm{T}} J \xi - V(x, q). \end{aligned} \tag{3.7}$$

3.3 Lie group variational integrator

3.3.1 Continuous time equation

We derive the Euler–Lagrange equations for the Lagrangian $\hat{L}$ on $T(\mathbb{R}^3 \times S^3)$. By Hamilton's principle, $(x(t), q(t)) \in \mathbb{R}^3 \times S^3$ extremizes the action integral

$$\int_{t_0}^{t_1} \hat{L}(x(t), q(t), \dot{x}(t), \dot{q}(t)) dt,$$

for variations that fix the endpoints $x(t_0) = x_0, x(t_1) = x_1$ and $q(t_0) = q_0, q(t_1) = q_1$. Consider variations of $(x(t), q(t))$ parameterized by λ: Given any $(\delta x(t), \eta(t))$ that vanish at the endpoints,

we construct $(x(t,\lambda), q(t,\lambda))$ as follows,

$$x(t,\lambda) = x(t) + \lambda\,\delta x(t),$$

and $q(t,\lambda) \in S^3$, such that $q(t,0) = q(t)$, satisfies the following equation,

$$\frac{\partial q}{\partial \lambda}(t,0) = q(t)\cdot(0,\eta(t)). \tag{3.8}$$

This implies that $q(t_0,\lambda) = q_0$ and $q(t_1,\lambda) = q_1$, since $\eta(t)$ vanishes at the endpoints. Since $\frac{\partial q}{\partial t}(t,\lambda) \in T_{q(t,\lambda)}S^3$, we have by left-trivialization, $\frac{\partial q}{\partial t}(t,\lambda) = q(t,\lambda)\cdot(0,\xi(t,\lambda))$, for a suitable choice of $\xi(t,\lambda) \in \mathbb{R}^3$. Taking derivatives with respect to λ on both sides, we obtain

$$\frac{\partial^2 q}{\partial\lambda\partial t}(t,\lambda) = \frac{\partial q}{\partial \lambda}(t,\lambda)\cdot(0,\xi(t,\lambda)) + q(t,\lambda)\cdot\left(0,\frac{\partial \xi}{\partial\lambda}(t,\lambda)\right).$$

Evaluating this at $\lambda = 0$ yields,

$$\begin{aligned}\frac{\partial^2 q}{\partial\lambda\partial t}(t,0) &= \frac{\partial q}{\partial \lambda}(t,0)\cdot(0,\xi(t)) + q(t)\cdot\left(0,\frac{\partial \xi}{\partial\lambda}(t,0)\right)\\ &= q(t)\cdot(0,\eta(t))\cdot(0,\xi(t)) + q(t)\cdot\left(0,\frac{\partial \xi}{\partial\lambda}(t,0)\right),\end{aligned} \tag{3.9}$$

where we used (3.8), and we let $\xi(t) = \xi(t,0)$. Taking derivatives with respect to t on both sides of (3.8) yields,

$$\begin{aligned}\frac{\partial^2 q}{\partial t\partial\lambda}(t,0) &= \dot{q}(t)\cdot(0,\eta(t)) + q(t)\cdot(0,\dot{\eta}(t))\\ &= q(t)\cdot(0,\xi(t))\cdot(0,\eta(t)) + q(t)\cdot(0,\dot{\eta}(t)).\end{aligned} \tag{3.10}$$

Equating (3.9) and (3.10) by the equality of mixed partials, we get

$$\begin{aligned}\left(0, \frac{\partial \xi}{\partial \lambda}(t,0)\right) &= (0, \dot{\eta}(t)) + (0, \xi(t)) \cdot (0, \eta(t)) - (0, \eta(t)) \cdot (0, \xi(t)) \\ &= (0, \dot{\eta}(t)) + (0, 2\xi(t) \times \eta(t)).\end{aligned}$$

Thus,

$$\frac{\partial \xi}{\partial \lambda}(t,0) = \dot{\eta}(t) + 2\xi(t) \times \eta(t). \tag{3.11}$$

Using the Lagrangian given in (3.7), Hamilton's principle states that

$$\frac{d}{d\lambda}\bigg|_{\lambda=0} \int_{t_0}^{t_1} \left[\frac{1}{2} m \|\dot{x} + \lambda \dot{\delta x}\|^2 + 2\xi(t,\lambda)^{\mathrm{T}} J \xi(t,\lambda) - V(x(t,\lambda), q(t,\lambda))\right] dt = 0,$$

which means that

$$\int_{t_0}^{t_1} \left[m\dot{x} \cdot \dot{\delta x} + 4\xi(t)^{\mathrm{T}} J \frac{\partial \xi}{\partial \lambda}(t,0) - \left(\frac{\partial V}{\partial x} \cdot \delta x + \left(\frac{\partial V}{\partial q}\right)^{\mathrm{T}} \frac{\partial q}{\partial \lambda}(t,0)\right)\right] dt = 0.$$

The terms involving the infinitesimal variation δx are

$$\int_{t_0}^{t_1} \left[m\dot{x} \cdot \dot{\delta x} - \frac{\partial V}{\partial x} \cdot \delta x\right] dt = \int_{t_0}^{t_1} \left(-m\ddot{x} - \frac{\partial V}{\partial x}\right) \delta x dt, \tag{3.12}$$

and using (3.11), we have that

$$\begin{aligned}\int_{t_0}^{t_1} 4\xi(t)^{\mathrm{T}} J \frac{\partial \xi}{\partial \lambda}(t,0) dt &= \int_{t_0}^{t_1} 4\xi^{\mathrm{T}} J(\dot{\eta} + 2\xi \times \eta) dt \\ &= \int_{t_0}^{t_1} 4\left(-\dot{\xi}^{\mathrm{T}} J + 2\xi^{\mathrm{T}} J \widehat{\xi}\right) \eta dt,\end{aligned} \tag{3.13}$$

where we integrated by parts, and used the fact that η vanishes at the endpoints.

For any $q \in \mathbb{H}, v \in \mathbb{R}^3$, define $F(q) : \mathbb{H} \to \mathbb{R}^{3\times 4}$ by the condition that

$$q \cdot (0, v) = F(q)^{\mathrm{T}} v.$$

It can be easily verified that for $q = (q_s, \vec{q}_v)$, $F(q) = (-\vec{q}_v, q_s \cdot I - \vec{q}_v)$, thus

$$\begin{aligned} \int_{t_0}^{t_1} \left(\frac{\partial V}{\partial q}\right)^{\mathrm{T}} \frac{\partial q}{\partial \lambda}(t,0)dt &= \int_{t_0}^{t_1} \left(\frac{\partial V}{\partial q}\right)^{\mathrm{T}} (q \cdot (0,\eta))dt \\ &= \int_{t_0}^{t_1} \left(\frac{\partial V}{\partial q}\right)^{\mathrm{T}} F(q)^{\mathrm{T}} \eta dt. \end{aligned} \tag{3.14}$$

Combining (3.9), (3.12), (3.14), integrating by parts, and using the fact that δx vanishes at the endpoints, we have

$$\int_{t_0}^{t_1} \left(-m\ddot{x} - \frac{\partial V}{\partial x}\right)\delta x + \left(-4\dot{\xi}^{\mathrm{T}} J + 8\xi^{\mathrm{T}} J \widehat{\xi} + \left(\frac{\partial V}{\partial q}\right)^{\mathrm{T}} F(q)^{\mathrm{T}}\right)\eta dt = 0,$$

for all variations δx and η that vanish at the endpoints. By the fundamental theorem of the calculus of variations, the Euler–Lagrange equations for the Lagrangian $\hat{L}$ on $T(\mathbb{R}^3 \times S^3)$ is given by,

$$\begin{cases} m\ddot{x} = -\dfrac{\partial V}{\partial x}, \\ 4J\dot{\xi} + 8\xi \times (J\varepsilon) = F(q)\dfrac{\partial V}{\partial q}, \\ \dot{q} = q \cdot (0, \xi). \end{cases} \tag{3.15}$$

3.3.2 Variational integrator on the Lagrangian side

The discrete Lagrangian from [64] is used here, which can be viewed as a midpoint rule approximation of the integral, combined with linear interpolation. Given endpoints (x_0, q_0) and (x_1, q_1), since $(0, \xi) = q^* \dot{q} \approx (\frac{q_0+q_1}{2})^* \cdot (\frac{q_1-q_0}{h}) = (0, \frac{1}{h}\mathrm{Im}(q_0^* q_1))$, we have that ξ is approximated

by $\frac{1}{h}\mathrm{Im}(q_0^* q_1)$. We can construct the discrete Lagrangian as

$$\begin{aligned} &L_d(x_0, q_0, x_1, q_1) \\ &\quad = h\left(\frac{1}{2}m\left\|\frac{x_1 - x_0}{h}\right\|^2 + \frac{2}{h^2}(\mathrm{Im}(q_0^* q_1))^{\mathrm{T}} J(\mathrm{Im}(q_0^* q_1)) - \frac{V(x_0, q_0) + V(x_1, q_1)}{2}\right). \end{aligned} \tag{3.16}$$

This discrete Lagrangian is consistent as it converges, as h approaches zero, to h times the continuous Lagrangian. Furthermore, it can be shown to be symmetric, and as such it is at least second-order accurate. More generally, the order of accuracy of the discrete Lagrangian can be obtained by comparing it to the exact discrete Lagrangian, using the variational error analysis developed in [43]. This has been related to the quadrature error and best approximation error of the finite-dimensional approximation space for Galerkin variational integrators in [18, 19].

The discrete Euler Lagrange equations are

$$\begin{cases} D_{x_1}(L_d(x_0, q_0, x_1, q_1) + L_d(x_1, q_1, x_2, q_2)) = 0, & (3.17) \\ D_{q_1}(L_d(x_0, q_0, x_1, q_1) + L_d(x_1, q_1, x_2, q_2)) = 0. & (3.18) \end{cases}$$

By substituting the expression for the discrete Lagrangian (3.16) into (3.17), we obtain

$$m \cdot \frac{x_2 - 2x_1 + x_0}{h^2} = -\frac{\partial V}{\partial x}(x_1, q_1). \tag{3.19}$$

For (3.18), recall that q_1 evolves on S^3, so we consider a variation $q_1(\lambda)$ of q_1, such that $q_1(0) = q_1$ and $\delta q_1 = \frac{dq_1(\lambda)}{d\lambda}\Big|_{\lambda=0} = q_1 \cdot (0, \eta)$, then

$$\begin{aligned} \frac{d}{dt}\bigg|_{\lambda=0} (\mathrm{Im}(q_0^* q_1(\lambda)))^{\mathrm{T}} J(\mathrm{Im}(q_0^* q_1(\lambda))) &= 2(\mathrm{Im}(q_0^* q_1))^{\mathrm{T}} J\left(\frac{d}{d\lambda}\bigg|_{\lambda=0} \mathrm{Im}(q_0^* q_1(\lambda))\right) \\ &= 2(\mathrm{Im}(q_0^* q_1))^{\mathrm{T}} J(\mathrm{Im}(q_0^* q_1(0, \eta))). \end{aligned} \tag{3.20}$$

Define $G(q) : \mathbb{H} \to \mathbb{R}^{3\times 3}$, such that for any $q \in \mathbb{H}$, $v \in \mathbb{R}^3$,

$$\mathrm{Im}(q \cdot (0, v)) = G(q)^{\mathrm{T}} v.$$

It can be verified that

$$G(q) = q_s \cdot I_3 - \vec{q}_v,$$

so

$$2(\mathrm{Im}(q_0^* q_1))^{\mathrm{T}} J(\mathrm{Im}(q_0^* q_1(0, \eta))) = 2(\mathrm{Im}(q_0^* q_1))^{\mathrm{T}} J G(q_0^* q_1)^{\mathrm{T}} \eta.$$

Similarly,

$$\begin{aligned}
\frac{d}{dt}\bigg|_{\lambda=0} (\mathrm{Im}(q_1^*(\lambda) q_2))^{\mathrm{T}} J(\mathrm{Im}(q_1^*(\lambda) q_2)) &= 2(\mathrm{Im}(q_1^* q_2))^{\mathrm{T}} J(\mathrm{Im}((q_1(0, \eta))^* q_2) \\
&= 2(\mathrm{Im}(q_2^* q_1))^{\mathrm{T}} J(\mathrm{Im}(q_2^* q_1(0, \eta))) \\
&= 2(\mathrm{Im}(q_2^* q_1))^{\mathrm{T}} J G(q_2^* q_1)^{\mathrm{T}} \eta.
\end{aligned} \tag{3.21}$$

Substituting (3.20) and (3.21) into (3.18) yields

$$\frac{4}{h^2}(\mathrm{Im}(q_0^* q_1))^{\mathrm{T}} J G(q_0^* q_1)^{\mathrm{T}} \eta + \frac{4}{h^2}(\mathrm{Im}(q_2^* q_1))^{\mathrm{T}} J G(q_2^* q_1)^{\mathrm{T}} \eta - \left(\frac{\partial V}{\partial q}(x_1, q_1)\right)^{\mathrm{T}} F(q_1)^{\mathrm{T}} \eta = 0,$$

which holds for any η, which gives

$$G(q_0^* q_1) J(\mathrm{Im}(q_0^* q_1)) + G(q_2^* q_1) J(\mathrm{Im}(q_2^* q_1)) = \frac{h^2}{4} F(q_1) \frac{\partial V}{\partial q}(x_1, q_1). \tag{3.22}$$

In summary, the discrete Euler–Lagrange equations derived from the discrete Lagrangian (3.16) are given by

$$\begin{cases}
m \cdot \dfrac{x_2 - 2x_1 + x_0}{h^2} = -\dfrac{\partial V}{\partial x}(x_1, q_1), \\
G(q_0^* q_1) J(\mathrm{Im}(q_0^* q_1)) + G(q_2^* q_1) J(\mathrm{Im}(q_2^* q_1)) = \dfrac{h^2}{4} F(q_1) \dfrac{\partial V}{\partial q}(x_1, q_1),
\end{cases}$$

where $G(q) = q_s \cdot I_3 - \vec{q}_v$, $F(q) = (-\vec{q}_v, q_s \cdot I - \vec{q}_v)$.

3.3.3 Variational integrator on the Hamiltonian side

For any Lie group G, its tangent bundle TG is diffeomorphic to $G \times \mathfrak{g}$ by left-trivialization,

$$TG \cong G \times \mathfrak{g}, \qquad (q, v_q) \xrightarrow{\psi} (q, L_{q^{-1}*} v_q),$$

and T^*G is diffeomorphic to $G \times \mathfrak{g}^*$,

$$T^*G \cong G \times \mathfrak{g}^*, \quad (q, p_q) \xrightarrow{\phi} (q, L_q{}^* p_q).$$

Given a Lagrangian $L : TG \to \mathbb{R}$, we can define $\hat{L} = L \circ \psi^{-1}$, and the following diagram commutes,

$$\begin{tikzcd}
TG \arrow[r, "F_L"] \arrow[d, "\psi"'] & T^*G \arrow[d, "\phi"] \\
G \times \mathfrak{g} \arrow[r, "F_{\hat{L}}"'] & G \times \mathfrak{g}^*
\end{tikzcd}$$

where F_L is the Legendre transformation and $F_{\hat{L}}$ denotes the partial derivative of $\hat{L}$ with respect to the Lie algebra element.

For a Lagrangian of the form $\hat{L}(x, q, \dot{x}, \xi) = \frac{1}{2} m \|\dot{x}\|^2 + 2\xi^{\mathrm{T}} J \xi - V(x, q)$, the Legendre transform is given by $(x, q, \dot{x}, \xi) \mapsto (x, q, m\dot{x}, 4J\xi)$.

For a discrete Lagrangian $L_d : G \times G \to \mathbb{R}$, we consider a variation $q_0(\lambda)$ of q_0, such that $q_0(0) = q_0$ and $\delta q_0 = \left.\frac{dq_0(\lambda)}{d\lambda}\right|_{\lambda=0} = L^*_{q_0} \xi$, then an element $\alpha \in \mathfrak{g}^*$ is defined by the condition

$$\left.\frac{d}{d\lambda}\right|_{\lambda=0} L_d(q_0(\lambda), q_1) = \langle \alpha, \xi \rangle.$$

It is easy to verify that $\alpha = L^*_{q_0} D_1 L_d(q_0, q_1)$. The same arguments hold for the derivative with respect to q_1. This allows us to conclude that the the partial derivative of the discrete Lagrangian

with respect to the Lie algebra element in the left-trivialized coordinate system that generates the variation of $q_i \in G$ is related to the partial derivative with respect to q_i by a left translation. So the variational integrator on the Hamiltonian side for the discrete Lagrangian (3.16) is given by

$$\begin{cases} p_0 = m \cdot \dfrac{x_1 - x_0}{h} + \dfrac{h}{2}\dfrac{\partial V}{\partial x}(x_0, q_0), \\ w_0 = -\dfrac{4}{h}G(q_1^* q_0)J(Im(q_1^* q_0)) + \dfrac{h}{2}F(q_0)\dfrac{\partial V}{\partial q}(x_0, q_0), \\ p_1 = m \cdot \dfrac{x_1 - x_0}{h} - \dfrac{h}{2}\dfrac{\partial V}{\partial x}(x_1, q_1), \\ w_1 = \dfrac{4}{h}G(q_0^* q_1)J(Im(q_0^* q_1)) - \dfrac{h}{2}F(q_1)\dfrac{\partial V}{\partial q}(x_1, q_1). \end{cases} \tag{3.23}$$

Here, w_0, w_1 are in the dual space of $\mathfrak{so}(3)$, and the derivation follows the approach adopted in the previous section.

3.4 Implementation of the algorithm

Given initial conditions $(x_0, R_0, \dot{x}_0, \dot{R}_0 = R_0\widehat{\Omega}_0)$ for the rigid body problem, our algorithm is given as follows,

1. lift $(x_0, R_0) \in SE(3)$ to $(x_0, q_0) \in \mathbb{R}^3 \times S^3$, such that $\pi(q_0) = R_0$ (see [42]);
2. start with initial conditions $(x_0, q_0, p_0 = m\dot{x}_0, w_0 = 2J\Omega_0)$ on $T^*(\mathbb{R}^3 \times S^3)$;
3. apply (3.23) repeatedly to generate the discrete sequence (x_k, q_k, p_k, w_k).

At each iteration, x_1 can be directly calculated by

$$x_1 = x_0 + \frac{h}{m}\left(p_0 - \frac{h}{2}\frac{\partial V}{\partial x}(x_0, q_0)\right).$$

There is a trick to calculating the $F(q_0)\frac{\partial V}{\partial q}(x_0,q_0)$ and $F(q_1)\frac{\partial V}{\partial q}(x_1,q_1)$ terms. Notice that $F(q)^T v = q\cdot(0,v)$, thus

$$\begin{aligned}\Big\langle F(q)\frac{\partial V}{\partial q},\varepsilon\Big\rangle &= \Big\langle \frac{\partial V}{\partial q}, F(q)^T\varepsilon\Big\rangle = \Big\langle \frac{\partial V}{\partial q}, q\cdot(0,\varepsilon)\Big\rangle,\\ &= \Big\langle \frac{\partial V}{\partial Q}, Q\cdot 2\varepsilon\Big\rangle = 2\Big\langle Q^T\frac{\partial V}{\partial Q},\varepsilon\Big\rangle.\end{aligned} \tag{3.24}$$

From this, we conclude that $F(q)\frac{\partial V}{\partial q} = 2\mathrm{Vec}(Q^T\frac{\partial V}{\partial Q})$, where Vec is the inverse of the hat map. The reason for calculating $\frac{\partial V}{\partial Q}$ instead of $\frac{\partial V}{\partial q}$ is due to the fact that calculating $\frac{\partial V}{\partial q}$ requires the computation of $\frac{\partial V}{\partial Q}$, so we will use both the quaternion and rotation representations of the orientation at each step, but only store the orientation in the quaternion representation for the numerical solution sequence. The update for q_1 can recast as a novel fixed point problem as opposed to a root finding problem. We have

$$G(q_1^* q_0)J(Im(q_1^* q_0)) = \frac{h}{4}\Big(-w_0 + \frac{h}{2}F(q_0)\frac{\partial V}{\partial q}(x_0,q_0)\Big),$$

and if we denote the updated unit quaternion $q_1^* q_0$ by q, and the right side (which is explicit) by b, we obtain

$$G(q)JIm(q) = b.$$

As such, q can be calculated by the following fixed point iteration,

$$G(q^k)JIm(q^{k+1}) = b. \tag{3.25}$$

During each iteration, the vector part of q^{k+1} is calculated by solving a linear system, and then the scalar part is derived from the unit length constraint; another fixed point iteration is based on the observation that $G(q) = q_s\cdot I - \vec{q}_v$, thus

$$G(q)v = q_s\cdot v - \vec{q}_v\times v,$$

for any vector v. As such, (3.25) can be modified to yield

$$q_s^k JIm(q^{k+1}) = \vec{q}_v^k \times JIm(q^k) + b. \tag{3.26}$$

In practice, (3.26) converges faster than (3.25). Once q_1 is calculated, p_1, w_1 can both be calculated explicitly from the last two equations in the system (3.23). If we wish to recover the discrete solution in the rotation matrix representation, we can project (x_k, q_k, p_k, w_k) to $TSE(3)$ by using

$$R_k = \pi(q_k), \qquad \dot{x}_k = \frac{p_k}{m}, \qquad \Omega_k = \frac{1}{2}J^{-1}w_k.$$

3.5 Analysis of algorithm

Our algorithm inherits some very nice geometric properties of the exact flow. First, (3.23) is by construction symplectic; second, if (3.23) is regarded as a map ϕ_h from (x_0, q_0, p_0, w_0) to (x_1, q_1, p_1, w_1), and we replace h with $-h$, it would yield

$$\begin{cases} p_0 = m \cdot \dfrac{x_0 - x_1}{h} - \dfrac{h}{2}\dfrac{\partial V}{\partial x}(x_0, q_0), \\ w_0 = \dfrac{4}{h}G(q_1^* q_0)J(Im(q_1^* q_0)) - \dfrac{h}{2}F(q_0)\dfrac{\partial V}{\partial q}(x_0, q_0), \\ p_1 = m \cdot \dfrac{x_0 - x_1}{h} + \dfrac{h}{2}\dfrac{\partial V}{\partial x}(x_1, q_1), \\ w_1 = -\dfrac{4}{h}G(q_0^* q_1)J(Im(q_0^* q_1)) + \dfrac{h}{2}F(q_1)\dfrac{\partial V}{\partial q}(x_1, q_1). \end{cases} \tag{3.27}$$

It is easy to check that (3.27) defines the same map ϕ_h from (x_1, q_1, p_1, w_1) to (x_0, q_0, p_0, w_0), thus $\phi_{-h} = \phi_h^{-1}$, and (3.23) is symmetric. The order of a symmetric method is necessarily even [37], so our algorithm has order two. Also the real system is time reversible, and we can verify that

$$\phi_{-h}(x_0, q_0, -p_0, -w_0) = (x_1, q_1, -p_1, -w_1),$$

thus, our method (3.23) is time reversible. Also, we observe that in this application that the Lagrangian (3.7) is invariant under the S^3 group action $S^3 \times (\mathbb{R}^3 \times S^3) \mapsto \mathbb{R}^3 \times S^3$:

$$q_1(x,q) = (\pi(q_1)x, q_1 q).$$

Thus, the corresponding momentum map is preserved, which is the total angular momentum $x \times p + \pi(q)J\Omega$. The discrete Lagrangian (3.16) we designed is also invariant under this group action, so by the discrete Noether's theorem [43], our algorithm will also preserve total angular momentum of the system.

A further investigation shows that discrete flow maps of (3.23) and the classical RATTLE method [17] are related. For the rigid body problem in the rotation representation, the Lagrangian is separable, and the RATTLE method is given by

$$\begin{cases} p_{1/2} = p_0 - \dfrac{h}{2}\dfrac{\partial V}{\partial x}(x_0, Q_0), \qquad x_1 = x_0 + h\dfrac{p_{1/2}}{m}, \\ P_{1/2} = P_0 - \dfrac{h}{2}\dfrac{\partial Q}{\partial x}(x_0, Q_0) - \dfrac{h}{2}Q_0\Lambda_0, \\ Q_1 = Q_0 + hP_{1/2}J_d^{-1}, \qquad Q_1^T Q_1 = I, \\ p_1 = p_{1/2} - \dfrac{h}{2}\dfrac{\partial V}{\partial x}(x_1, Q_1), \\ P_1 = P_{1/2} - \dfrac{h}{2}\dfrac{\partial Q}{\partial x}(x_1, Q_1) - \dfrac{h}{2}Q_1\Lambda_1, \quad Q_1^T P_1 J_d^{-1} + J_d^{-1} P_1^T Q_1 = 0. \end{cases} \tag{3.28}$$

The RATTLE method can be derived using constrained mechanics with the discrete Lagrangian

$$L_d(x_0, Q_0, x_1, Q_1) = h\left(\frac{1}{2}m\left|\frac{x_1 - x_0}{h}\right|^2 + \frac{1}{2}\text{trace}\left[\left(\frac{Q_1 - Q_0}{h}\right)J_d\left(\frac{Q_1 - Q_0}{h}\right)^T\right] - \frac{V(x_0, Q_0) + V(x_1, Q_1)}{2}\right), \tag{3.29}$$

where $Q_0 = \pi(q_0)$ and $Q_1 = \pi(q_1)$. We can prove that (3.29) is equivalent to the discrete Lagrangian (3.16) we used to derive our method, thus the discrete flow maps of the two resulting

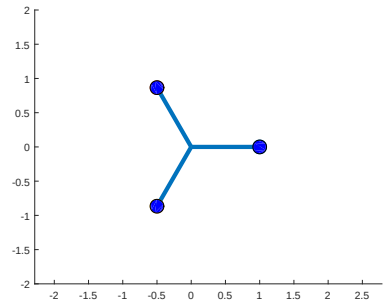
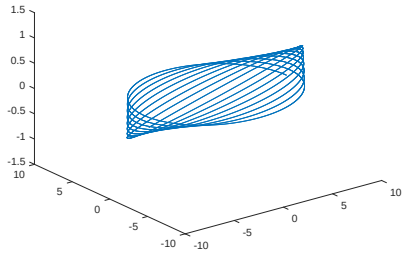


Figure 3.1. Planar rigid body

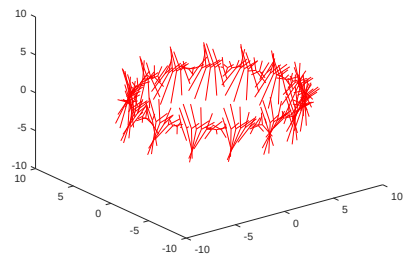
methods (3.23) and (3.28) are related by the projection from unit quaternions to rotation matrices. There are efficient ways of implementing (3.28), where the Lagrangian multipliers Λ_0 and Λ_1 are eliminated, and the quaternion representation can be introduced to obtain a fixed point iteration for Q_1[17], and P_1 can be obtained by solving a linear system. Here in our algorithm (3.23), the angular momentum w_1 is updated explicitly.

3.6 Numerical Experiments

We consider a planar rigid body composed of three uniform balls with unit mass and radius 0.1, connected by massless rods as in Figure 3.1. We numerically simulate the motion of the planar rigid body in a gravitational field centered at the origin with potential energy $-\frac{1}{r}$, initial conditions $x_0 = (8,0,0)$, $q_0 = (1,0,0,0)$, $p_0 = (0,1,0)$, and $w_0 = (1,2,3)$, and timestep $h = 0.01$. The trajectory of the center of mass during the time interval $[0,1000]$ and motion of the rigid body over one orbital period are given in Figure 3.2. The energy error and quaternion error, which is the deviation from the unit quaternion surface, during the time interval $[0,1000]$ are given in Figures 3.3a and 3.3b, and we observe that these invariants are preserved extremely well. It can be seen that energy error remains stable due to the symplecticity of the proposed algorithm, and the quaternion error remains stable due to the use of the intrinsic Lie group composition to update the numerical trajectory. The error in each component of the angular momentum is given in Figure 3.3c. As can be seen, it is preserved quite well.

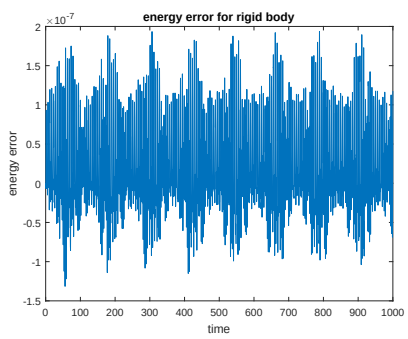


(a) trajectory of the center of mass

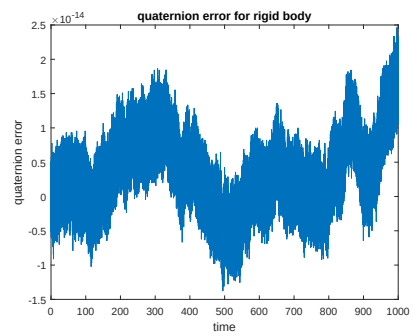


(b) rigid body motion over one orbit

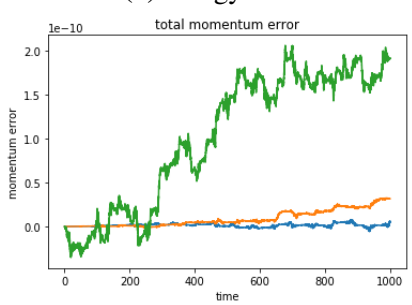
Figure 3.2. Motion of the rigid body in a central gravitational field.



(a) energy error of LGVI

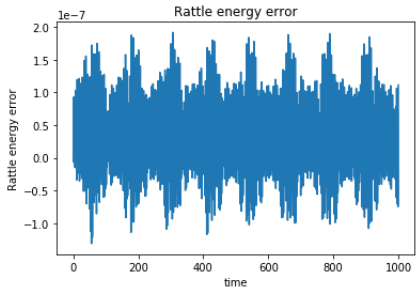


(b) quaternion error of LGVI

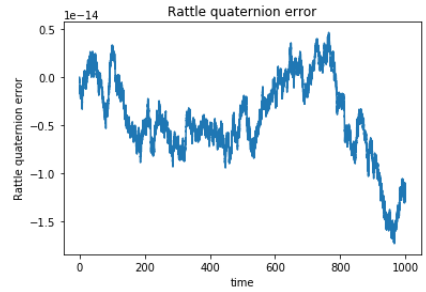


(c) angular momentum error of LGVI

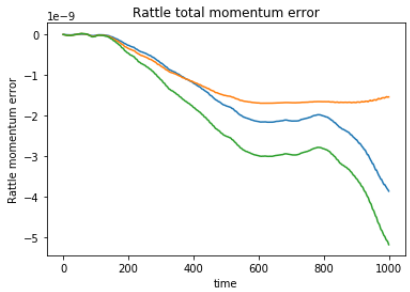
Figure 3.3. The Lie group variational integrator using quaternions preserves the energy and the unit norm condition very well. In addition, the angular momentum is preserved quite well.



(a) energy error of RATTLE



(b) quaternion error of RATTLE



(c) momentum error of RATTLE

Figure 3.4. The RATTLE method performs very similarly to the Lie group variational integrator using quaternions.

Since our method and RATTLE are analytically equivalent if the Lagrange multipliers associated with the unit quaternion constraint are eliminated, it is natural to compare the performance of our method with RATTLE to see if there are differences in the numerical properties in practice. In Figure 3.4, we examine the numerical performance of RATTLE, in particular, the energy error, unit quaternion error and momentum error. The quaternion representation is adopted to update Q_1 by fixed point iteration, and P_1 is solved as a linear system. In our problem, the nonstandard moment of inertia matrix J_d is diagonal, thus this linear system is decoupled into three linear subsystems, and P_1 has a closed form solution. The numerical performance of RATTLE is similar to our method, with almost the same run time, which is demonstrated by the computational efficiency plot in Figure 3.5.

In addition, a runtime comparison is also made between three methods: (i) our Lie group variational integrator using the unit quaternion representation; (ii) the Lie group Störmer–Verlet method [8, Eq. (4.15)-(4.18)], which is a special case of the Lie group variational partitioned

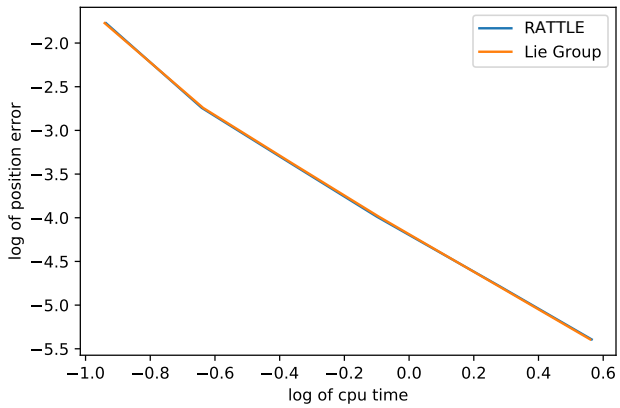


Figure 3.5. The error vs. CPU time for both our proposed Lie group variational integrator using quaternions and RATTLE are virtually indistinguishable.

Table 3.1. Runtime comparison (1000 steps)

Lie group quaternion	Lie group Störmer–Verlet	SHAKE
7.46s	7.20s	11.00s

Runge–Kutta method, where the Lie group is the rotation group $SO(3)$; (iii) the SHAKE method, which can be viewed as a special case of the variational integrator for constrained mechanics [43], where the unit norm for quaternions is used as constraint. All these methods are symplectic by construction. In Table 3.1, the runtime to perform 1000 steps with the same initial condition is reported for each method. It is observed that the SHAKE constrained integrator, costs significantly more than the other two. In practice, the difference in computational cost between our method and the Störmer–Verlet Lie group method is negligible, and the cost depends strongly on the choice of initial conditions, the timestep, and the code implementation. So, for this rigid body example, our method is roughly comparable to the Störmer–Verlet Lie group method in runtime complexity, maintains an advantage in memory cost, and is significantly cheaper than the SHAKE symplectic constrained integrator. As such, our method is competitive with methods using the rotation matrix representation, and cheaper than other methods using the unit quaternion representation.

Chapter 3, in full, is currently being prepared for submission for publication of the material. Xuefeng Shen; Melvin Leok. The dissertation author was the primary investigator and

author of this material.

Chapter 4

High-Order Symplectic Lie Group Methods on $SO(n)$ using the Polar Decomposition

4.1 Introduction

Given a configuration manifold Q, variational integrators provide a useful method of deriving symplectic integrators for Lagrangian mechanics on the tangent bundle TQ in terms of the Lagrangian L, or for Hamiltonian mechanics on the cotangent bundle T^*Q in terms of the Hamiltonian H. It involves discretizing Hamilton's principle or Hamilton's phase space principle rather than the Euler–Lagrange or Hamilton's equations. Discrete Lagrangian variational mechanics is described in terms of a discrete Lagrangian $L_d(q_0, q_1)$, which is an approximation of the exact discrete Lagrangian,

$$L_d^{\text{exact}}(q_0, q_1) = \underset{\substack{q \in C^2([0,h],Q) \\ q(0)=q_0, q(h)=q_1}}{} \int_0^h L(q(t), \dot{q}(t))dt. \tag{4.1}$$

The discrete Hamilton's principle states that the discrete action sum is stationary,

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}) = 0,$$

for variations that vanish at the endpoints, i.e., $\delta q_0 = \delta q_N = 0$. This yields the discrete Euler–Lagrange equation,

$$D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1}) = 0,$$

where D_i denotes the partial derivative with respect to the ith-argument, and this defines an update map on on $Q \times Q$, where $(q_{k-1}, q_k) \mapsto (q_k, q_{k+1})$. This update map can equivalent be described in terms of the discrete Legendre transforms,

$$p_k = -D_1 L_d(q_k, q_{k+1}), \qquad p_{k+1} = D_2 L_d(q_k, q_{k+1}), \tag{4.2}$$

which defines an update map on T^*Q, $(q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, that automatically preserves the canonical symplectic structure on T^*Q. The order of the variational integrator depends on how accurately $L_d(q_0, q_1)$ approximates $L_d^{\text{exact}}(q_0, q_1)$. To derive a high-order discrete Lagrangian $L_d(q_0, q_1)$, a typical approach is the Galerkin method [43]. This involves replacing in (4.1) the function space $C^2([0,h], Q)$ with a finite-dimensional function space, and replacing the integral with a numerical quadrature formula, and when the configuration manifold Q is a linear space, and polynomials of degree less than or equal to s are chosen, the classical symplectic partitioned Runge–Kutta methods are recovered. Later, Leok and Shingel [39] introduced the shooting-based discrete Lagrangian, which allows one to construct a symplectic integrator from an arbitrary one-step method.

When the configuration manifold Q is a Lie group G, the construction of the discrete Lagrangian $L_d(g_0, g_1)$ is more complicated than the linear space case. Leok [38] proposed parametrizing curves on the Lie group using the exponential map, i.e., a curve $g(t)$ connecting g_0 and g_1 is represented by

$$g(t) = g_0 \cdot \exp(\varepsilon(t)),$$

where $\varepsilon(t) \in \mathfrak{g}$ is a curve on the Lie algebra of G, with fixed endpoints $\varepsilon(0) = 0$, $\varepsilon(h) = \log(g_0^{-1} g_1)$. This allows one to replace variations in $g(t)$ by variations in $\varepsilon(t)$ on the Lie algebra

$\mathfrak{g}$, which is a linear space. This yields the following expression for the exact discrete Lagrangian,

$$L_d^{\text{exact}}(g_0, g_1) = \underset{\substack{\varepsilon \in C^2([0,h],\mathfrak{g}) \\ \varepsilon(0)=0, \varepsilon(h)=\log(g_0^{-1} g_1)}}{} \int_0^h L(g_0 \cdot \exp(\varepsilon(t)), g_0 \cdot_{\varepsilon(t)} (\dot{\varepsilon}(t)))dt, \tag{4.3}$$

where ${}_{\varepsilon}(\dot{\varepsilon}) = \exp(\varepsilon) \cdot \frac{1-\mathrm{e}^{-\mathrm{ad}_\varepsilon}}{\mathrm{ad}_\varepsilon}(\dot{\varepsilon})$ is the tangent lift of the exponential map. In (4.3), if $\varepsilon(t)$ is restricted to a finite-dimensional function space, and we replace the integral with a quadrature rule, then we obtain Galerkin Lie group variational integrators. The error analysis and implementation details for such methods can be found in [7, 19]. The above construction can be naturally extended to any retraction [2] τ on G, i.e., a diffeomorphism from a neighborhood of $0 \in \mathfrak{g}$ to neighborhood of $e \in G$ that satisfies a rigidity condition. The main disadvantage of Galerkin Lie group variational integrators is that the discrete Lagrangian (4.3) involves , and the resulting discrete Euler–Lagrange equations will involve $\underset{\cdot}{2}exp$, which cannot be calculated exactly in general, and requires the truncation of a series expansion.

For mechanics on a Lie group, when the Lagrangian L or Hamiltonian H is left-invariant, the mechanical system can be symmetry reduced to evolution on the Lie algebra $\mathfrak{g}$ or its dual space $\mathfrak{g}^*$. On the Lagrangian side, it corresponds to Euler–Poincaré reduction, which is described by the Euler–Poincaré equation,

$$\frac{d}{dt}\left(\frac{\partial l}{\partial \varepsilon}\right) =_{\varepsilon}^{*} \left(\frac{\partial l}{\partial \varepsilon}\right),$$

which is expressed in terms of the reduced Lagrangian $l(g^{-1}g) = L(g, \dot{g})$. This can be described in terms of a reduced variational principle $\delta \int_a^b l(\varepsilon(t))dt = 0$, with respect to variations of form $\delta\varepsilon = \dot{\eta} + [\varepsilon, \eta]$, where $\eta(t)$ is an arbitrary path in the Lie algebra $\mathfrak{g}$ that vanishes at the endpoints, i.e., $\eta(a) = \eta(b) = 0$. On the Hamiltonian side, this corresponds to Lie–Poisson reduction, and the Lie–Poisson structure on $\mathfrak{g}^*$ can be defined as

$$\{F, G\}(\mu) = \left\langle \mu, \left[\frac{\partial F}{\partial \mu}, \frac{\partial G}{\partial \mu}\right] \right\rangle,$$

which together with the reduced Hamiltonian h, gives the Lie–Poisson equation on $\mathfrak{g}^*$,

$$\frac{d\mu}{dt} = \mathrm{ad}^*_{\frac{\partial h}{\partial \mu}}(\mu).$$

If the discrete Lagrangian $L_d(g_0, g_1)$ is also G-invariant, i.e., $L_d(g \cdot g_0, g \cdot g_1) = L_d(g_0, g_1)$, then (4.2) can be reduced to a Lie–Poisson integrator [27],

$$\begin{cases} \mu_0 = l_d^{'}(f_0) f_0^{-1}, \\ \mu_1 = f_0^{-1} \cdot \mu_0 \cdot f_0. \end{cases} \tag{4.4}$$

This algorithm preserves the coadjoint orbits and hence the Poisson structure on $\mathfrak{g}^*$. In (4.4), $l_d(f_0) = L_d(e, f_0)$ is the reduced discrete Lagrangian that arises from G-invariance.

In this paper, we will derive a variational integrator on Lie group $SO(n)$, but instead of parametrizing curves on $SO(n)$ by the exponential map or a retraction, we will embed $SO(n)$ naturally in the space $GL_+(n) = \{A \in \mathbb{R}^{n\times n} \mid \det(A) > 0\}$, which is an open subset of $\mathbb{R}^{n\times n}$. Like with constrained mechanics, given fixed endpoints g_0, g_1, we will construct interpolating polynomials in $GL_+(n)$, while requiring that certain internal points stay on $SO(n)$. However, we do not need to extend the Lagrangian L to $GL_+(n)$, instead, we will project the trajectory onto $SO(n)$. The projection from $GL_+(n)$ to $SO(n)$ can be implemented using the polar decomposition. When L is $SO(n)$-invariant, the constructed discrete Lagrangian is also $SO(n)$-invariant, and we can obtain a reduced symplectic Lie–Poisson integrator by using (4.4).

Our Galerkin variational integrator will involve a discrete Lagrangian that differs from the classical construction in [43]. Traditionally in the linear space setting, (4.1) is approximated with a quadrature rule,

$$L_d(q_0, q_1) = h \sum_{i=1}^{s} b_i L(q(c_i h), \dot{q}(c_i h)) = h \sum_{i=1}^{s} L(Q_i, \dot{Q}_i),$$

and $q(t)$ is approximated by polynomials with degree less than or equal to s with fixed endpoints

q_0, q_1. By choosing interpolation nodes $\{d_0^\nu\}_{\nu=0}^s$ with $d_0^0 = 0, d_0^s = 1$, and interpolation values $\{q_0^\nu\}_{\nu=0}^s$ with $q_0^0 = q_0, q_0^s = q_1$, $q(t)$ can be expressed as $q(t) = \sum_{\nu=0}^{s} q_0^\nu \phi_\nu(\frac{t}{h})$ on $[0,h]$, where $\phi_\nu(t)$ are Lagrange polynomials corresponding to the interpolation nodes $\{d_0^\nu\}_{\nu=0}^s$. So, by taking variations with respect to the interpolation values $\{q_0^\nu\}_{\nu=1}^{s-1}$, $q(t)$ is varied over the finite-dimensional function space,

$$\mathbb{M}^s = \{q(t) \mid q(t) \in P_s[0,h], q(0) = q_0, q(h) = q_1\}.$$

Consider the quadrature approximation of the action integral, viewed as a function of the endpoint and interpolation values,

$$F(q_0, q_1, \{q_0^\nu\}_{\nu=1}^{s-1}) = h \sum_{i=1}^{s} b_i L(q(c_i h), \dot{q}(c_i h)),$$

where $q(t) = \sum_{\nu=0}^{s} q_0^\nu \phi_\nu(\frac{t}{h})$. Then, a variational integrator (4.2) can be obtained as follows,

$$\begin{cases} 0 = \dfrac{\partial F}{\partial q_0^\nu}, & \nu = 1, 2 \ldots s-1, \\ -p_0 = \dfrac{\partial F}{\partial q_0}, \\ p_1 = \dfrac{\partial F}{\partial q_1}. \end{cases} \tag{4.5}$$

However, (4.5) is often impractical due to the complexity of evaluating $q(c_i h)$, $\dot{q}(c_i h)$ because of the Lagrange interpolation form. The other issue is that (4.5) requires computing the root of a system of nonlinear equations, as opposed to a fixed point equation as with the Runge–Kutta method. The latter can be solved by fixed point iteration, which is much more efficient in practice. Notice that the finite-dimensional function space $\mathbb{M}^s$ does not actually depend on the choice of

nodes $\{d_0^\nu\}_{\nu=1}^{s-1}$, and by a tricky elimination of $\phi_\nu(t)$, (4.5) can be simplified to yield,

$$q_1 = q_0 + h\sum_{i=1}^{s} b_i\dot{Q}_i, \qquad p_1 = p_0 + h\sum_{i=1}^{s} b_i\dot{P}_i, \tag{4.6a}$$

$$Q_i = q_0 + h\sum_{j=1}^{s} a_{ij}\dot{Q}_j, \qquad P_i = p_0 + h\sum_{j=1}^{s} \tilde{a}_{ij}\dot{P}_j, \tag{4.6b}$$

$$P_i = \frac{\partial L}{\partial \dot{q}}(Q_i, \dot{Q}_i), \qquad \dot{P}_i = \frac{\partial L}{\partial q}(Q_i, \dot{Q}_i), \tag{4.6c}$$

where $\tilde{a}_{ij} = b_j(1 - \frac{a_{ji}}{b_i})$. When transformed to the Hamiltonian side, (4.6) just recovers the symplectic partitioned Runge–Kutta method. The same variational integrator can be derived in a much simpler way, instead of performing variations on internal points $\{q_0^\nu\}_{\nu=1}^{s-1}$, we will perform variations on the internal derivatives $\{\dot{Q}\}_{i=1}^{s}$, subject to the constraint that $q_1 = q_0 + h\sum_{i=1}^{s} b_i\dot{Q}_i$, and with internal points reconstructed using $Q_i = q_0 + h\sum_{j=1}^{s} a_{ij}\dot{Q}_j$. Consider the quadrature approximation of the action integral, viewed as a function of the endpoint values and the internal velocities,

$$\tilde{F}(q_0, q_1, \{\dot{Q}_i\}_{i=1}^{s}, \lambda) = h\sum_{i=1}^{s} b_i L(Q_i, \dot{Q}_i) + \lambda^T\Big(q_1 - q_0 - h\sum_{i=1}^{s} b_i\dot{Q}_i\Big),$$

where λ is a Lagrange multiplier that enforces the constraint. Then, a variational integrator (4.2) can be obtained as follows,

$$\begin{cases} 0 = \dfrac{\partial \tilde{F}}{\partial \dot{Q}_i}, \qquad i = 1, 2 \ldots s, \\ 0 = \dfrac{\partial \tilde{F}}{\partial \lambda}, \\ Q_i = q_0 + h\displaystyle\sum_{j=1}^{s} a_{ij}\dot{Q}_j, \\ -p_0 = \dfrac{\partial \tilde{F}}{\partial q_0}, \\ p_1 = \dfrac{\partial \tilde{F}}{\partial q_1}. \end{cases} \tag{4.7}$$

Explicitly expanding (4.7) and eliminating the Lagrange multiplier yields (4.6) in a much more straightforward manner. Here, in this paper, the same technique will be adopted on the rotation group $SO(n)$ in order to directly obtain a variational integrator in fixed point form.

4.2 Background

Given $A \in GL(n)$, there exists a unique orthogonal matrix $U \in O(n)$ and symmetric positive-definite matrix P, such that $A = UP$. The corresponding map from $GL(n)$ to $O(n)$, $\mathbb{P}(A) = U$, is called the polar decomposition. Specifically when $A \in GL_+(n)$, we have $U \in SO(n)$. Geometrically, the polar decomposition is the closest point projection from $GL(n)$ to $O(n)$, with respect to the Frobenius norm $\|A\|_F^2 = \mathrm{Tr}(A^T A)$. It satisfies the property

$$\mathbb{P}(Q_1 A Q_2) = Q_1 \mathbb{P}(A) Q_2, \tag{4.8}$$

for $\forall Q_1, Q_2 \in O(n)$. In practice, a fast and efficient algorithm for calculating the poolar decomposition is the Newton iteration,

$$U_{k+1} = \frac{1}{2}(U_k + U_k^{-T}), U_0 = A. \tag{4.9}$$

The polar decomposition can be used to construct a retraction on $SO(n)$. The Lie algebra of $SO(n)$ is $\mathrm{Asym}(n) = \{A \in \mathbb{R}^{n\times n} \mid A^T = -A\}$. Now, consider the retraction from the Lie algebra to $SO(n)$, which maps $\Omega \in \mathrm{Asym}(n)$ to

$$\mathbb{P}(I + \Omega) = U.$$

This provides a diffeomorphism between a neighborhood of $0 \in \mathrm{Asym}(n)$ and a neighborhood of $I \in SO(n)$. To calculate the inverse of the above map, suppose that $I + \Omega = UP$, and take the transpose on both sides to obtain $I - \Omega = PU^T$, which implies that $U^T(I + \Omega) = (I - \Omega)U$.

Thus, we have that

$$U^T\Omega + \Omega U + U^T - U = 0. \tag{4.10}$$

This is a Lyapunov equation, and it is well known that matrix equations of the form $AX + XB + C = 0$ have a unique solution if and only if for $\forall \lambda \in \sigma(A), \mu \in \sigma(B)$, $\lambda + \mu \neq 0$. For U in the neighborhood of identity, its eigenvalues lie in the right open half plane, which ensures that a unique solution to (4.10) exists. In principle, this Lyapunov equation can be solved using classical algorithms [5, 16].

Next we introduce the tangent map and its pullback for the polar decomposition, which is essential for the derivation of the variational integrator. Let $\mathbb{P}(A(t)) = U(t)$, then $A(t) = U(t)P(t)$, where $U(t) \in SO(n)$, $P(t)$ is symmetric positive-definite. Differentiating both sides yields, $\dot{A} = \dot{U}P + U\dot{P}$, and we denote $\dot{U} = U\Omega$, where $\Omega \in \mathrm{Asym}(n)$. Then,

$$U^T\dot{A} - \Omega P = \dot{P} = \dot{A}^T U + P\Omega,$$

from which we obtain,

$$\Omega P + P\Omega = U^T\dot{A} - \dot{A}^T U. \tag{4.11}$$

So the tangent map of the polar decomposition $dP_A(\dot{A}) : \mathbb{R}^{n\times n} \to \mathrm{Asym}(n)$ can be calculated by solving the Lyapunov equation (4.11). To define the pullback of tangent map, we introduce inner products on $\mathbb{R}^{n\times n}$ and $\mathrm{Asym}(n)$, and identify these spaces with their duals using the Riesz representation theorem. For the space $\mathbb{R}^{n\times n}$, $\forall A, B \in \mathbb{R}^{n\times n}$, we define an inner product as follows,

$$A \odot B = \sum_{i,j=1}^{n} a_{ij} \cdot b_{ij},$$

for space $\mathrm{Asym}(n)$, $\forall \Omega, \Omega' \in \mathrm{Asym}(n)$, we define an inner product by

$$\langle \Omega, \Omega' \rangle = \sum_{i<j} \Omega_{ij} \cdot \Omega'_{ij} = \frac{1}{2}\Omega \odot \Omega'.$$

The following properties can be easily verified,

(a) $\forall A, B \in \mathbb{R}^{n\times n}$, $A \odot B = A^T \odot B^T$;

(b) $\forall A, B, P, Q \in \mathbb{R}^{n\times n}$, $A \odot (PBQ) = (P^T A Q^T) \odot B$;

(c) $\forall \Omega \in \mathrm{Asym}(n), A \in \mathbb{R}^{n\times n}$, $\langle \Omega, \mathrm{Asym}(A)\rangle = \sum_{i<j} \Omega_{ij}(A_{ij} - A_{ji}) = \Omega \odot A$.

The Asym operation above is defined to be $\mathrm{Asym}(A) = A - A^T$. Given the choice of inner products, by Riesz representation, $(\mathbb{R}^{n\times n})^*$ can be identified with $\mathbb{R}^{n\times n}$, and $\mathrm{Asym}(n)^*$ can be identified with $\mathrm{Asym}(n)$. Thus, the pullback of dP_A can be defined as

$$dP_A^*(\Omega) \odot W = \langle \Omega, dP_A(W)\rangle, \qquad \forall \Omega \in \mathrm{Asym}(n), W \in \mathbb{R}^{n\times n}.$$

As discussed, $dP_A(W)$ involves solving the Lyapunov equation (4.11). To compute out dP_A^*, we define two maps:

$$\begin{aligned} \phi &: \mathrm{Asym}(n) \to \mathrm{Asym}(n), \qquad && \Omega \mapsto \Omega P + P\Omega, \\ \psi &: \mathbb{R}^{n\times n} \to \mathrm{Asym}(n), && W \mapsto U^T W - W^T U, \end{aligned}$$

and dP_A can be viewed as composition of ψ and ϕ^{-1},

$$dP_A = \phi^{-1} \circ \psi : \mathbb{R}^{n\times n} \to \mathrm{Asym}(n) \to \mathrm{Asym}(n),$$

and $dP_A^*(\Omega) = (\phi^{-1} \circ \psi)^*(\Omega) = \psi^* \circ (\phi^*)^{-1}(\Omega)$. Let us derive the expressions for ϕ^* and ψ^*. For ϕ^*, $\forall \Omega, X \in \mathrm{Asym}(n)$,

$$\begin{aligned} \langle \phi^*(\Omega), X\rangle &= \langle \Omega, \phi(X)\rangle = \langle \Omega, XP + PX\rangle \\ &= \langle \Omega, \mathrm{Asym}(XP)\rangle = \Omega \odot (XP) = (\Omega P) \odot X \\ &= \langle \Omega P + P\Omega, X\rangle. \end{aligned}$$

Thus, $\phi^* = \phi$, and ϕ is Hermitian. For ψ^*, $\forall \Omega \in \mathrm{Asym}(n), W \in \mathbb{R}^{n\times n}$,

$$\begin{aligned}\psi^*(\Omega)\odot W &= \langle \Omega, \psi(W)\rangle = \langle \Omega, U^T W - W^T U\rangle \\ &= \langle \Omega, \mathrm{Asym}(U^T W)\rangle = \Omega \odot (U^T W) \\ &= (U\Omega)\odot W,\end{aligned}$$

and therefore, $\psi^*(\Omega) = U\Omega$. Together, we obtain,

$$dP_A^*(\Omega) = \psi^* \circ (\phi^*)^{-1}(\Omega) = \psi^* \circ (\phi)^{-1}(\Omega) = U\mathrm{Lyap}(P, \Omega^T),$$

where $\mathrm{Lyap}(P, \Omega^T)$ represents the solution of the Lyapunov equation $XP + PX + \Omega^T = 0$.

Finally, we state a lemma that will be used later.

Lemma 1. *$\mathbb{P}(I+S) = I$ iff $S \in Sym(n)$ and $eig(S) > -1$.*

4.3 Lagrangian variational integrators on the rotation group $SO(n)$

Consider the case where the Lie group is the rotation group, i.e., $G = SO(n)$. Given a left-trivialized Lagrangian $L : G \times \mathfrak{g} \to \mathbb{R}$, we construct a discrete Lagrangian following the approach used to derive (4.7). Denote the internal points by $\{U_i\}_{i=1}^s \in G$, and the left-trivialized internal tangent vectors by $\{\Omega_i\}_{i=1}^s \in \mathfrak{g}$. Fixing the endpoints g_0, g_1, we obtain

$$L_d(g_0, g_1) = h\sum_{i=1}^s b_i L(U_i, \Omega_i),$$

subject to the constraint

$$g_1 = \mathbb{P}\Big(g_0 + h\sum_{i=1}^s b_i U_i \Omega_i\Big), \tag{4.12}$$

where the internal points U_i are represented by

$$U_i = \mathbb{P}\Big(g_0 + h\sum_{j=1}^{s} a_{ij}U_j\Omega_j\Big). \tag{4.13}$$

(4.12), (4.13) applies the same interpolation rules as in the linear space case. Since the interpolated points may not lie on the Lie group G, they are projected to G using the polar decomposition.

Observe that (4.12) is equivalent to the condition that $\mathbb{P}\Big(g_1^T(g_0 + h\sum_{i=1}^{s} b_iU_i\Omega_i)\Big) = \mathbb{I}$. Suppose that h is small, and g_0, g_1 are close enough to each other, then $g_1^T\Big(g_0 + h\sum_{i=1}^{s} b_iU_i\Omega_i\Big)$ is in the neighborhood of $\mathbb{I}$. By Lemma 1, (4.12) holds iff $g_1^T\Big(g_0 + h\sum_{i=1}^{s} b_iU_i\Omega_i\Big) \in \mathrm{Sym}(n)$, i.e.,

$$\mathrm{Asym}\Big(g_1^T\Big(g_0 + h\sum_{i=1}^{s} b_iU_i\Omega_i\Big)\Big) = 0.$$

Now we can construct a discrete Lagrangian with the constraint enforced using a Lagrange multiplier term,

$$\tilde{F}(g_0, g_1, \{\Omega_i\}_{i=1}^{s}, \Lambda) = h\sum_{i=1}^{s} b_iL(U_i, \Omega_i) + \Big\langle \Lambda, \mathrm{Asym}\Big(g_1^T\Big(g_0 + h\sum_{i=1}^{s} b_iU_i\Omega_i\Big)\Big)\Big\rangle,$$

where $\Lambda \in \mathrm{Asym}(n)$. The corresponding variational integrator is given by

$$0 = \frac{\partial \tilde{F}}{\partial \Omega_i}, \qquad i = 1, 2 \ldots s, \tag{4.14a}$$

$$0 = \frac{\partial \tilde{F}}{\partial \Lambda}, \tag{4.14b}$$

$$U_i = \mathbb{P}\Big(g_0 + h\sum_{j=1}^{s} a_{ij}U_j\Omega_j\Big), \tag{4.14c}$$

$$-p_0 = \frac{\partial \tilde{F}}{\partial g_0}, \tag{4.14d}$$

$$p_1 = \frac{\partial \tilde{F}}{\partial g_1}. \tag{4.14e}$$

It is easy to see that (4.14b) is equivalent to the constraint (4.12). We now turn to equation (4.14a), where the main difficulty is that the dependence of $\{U_i\}_{i=1}^s$ on $\{\Omega_i\}_{i=1}^s$ is not explicit, rather it involves solving a nonlinear system (4.13). Pick an index $k \in \{1,2\dots s\}$, and vary Ω_k, i.e., $\Omega_k \to \Omega_k(\tau)$, with $\Omega_k(0) = \Omega_k$, and $\dot{\Omega}_k(0) = \delta\Omega_k$, while $\{\Omega_i\}_{i\neq k}$ remain fixed. Then,

$$U_i(\tau) = \mathbb{P}\Big(g_0 + h\sum_{j\neq k} a_{ij}U_j(\tau)\Omega_j + ha_{ik}U_k(\tau)\Omega_k(\tau)\Big).$$

Differentiating both sides, and letting $\dot{U}_i = U_i X_{ik}$, we have that

$$X_{ik} = dP_{A_i}\Big(h\sum_{j=1}^{s} a_{ij}U_jX_{jk}\Omega_j + ha_{ik}U_k\delta\Omega_k\Big), \tag{4.15}$$

where $A_i = g_0 + h\sum\limits_{j=1}^{s} a_{ij}U_j\Omega_j$. Then, (4.15) can be rewritten as

$$X_{ik} - dP_{A_i}\Big(h\sum_{j=1}^{s} a_{ij}U_jX_{jk}\Omega_j\Big) = hdP_{A_i}(a_{ik}U_k\delta\Omega_k), \tag{4.16}$$

In order to represent $\{X_{ik}\}_{i=1}^s$ in terms of $\delta\Omega_k$, we define three maps:

$$\begin{aligned}
\psi_k &: \mathrm{Asym}(n) \to \mathrm{Asym}(n)^s, \quad && \delta\Omega_k \mapsto \{dP_{A_i}(a_{ik}U_k\delta\Omega_k)\}_{i=1}^s,\\
\phi &: \mathrm{Asym}(n)^s \to \mathrm{Asym}(n)^s, \quad && \{X_{ik}\}_{i=1}^s \mapsto \Big\{X_{ik} - dP_{A_i}\Big(h\sum_{j=1}^{s} a_{ij}U_jX_{jk}\Omega_j\Big)\Big\}_{i=1}^s,\\
P_i &: \mathrm{Asym}(n)^s \to \mathrm{Asym}(n), \quad && \{\Omega_i\}_{i=1}^s \mapsto \Omega_i.
\end{aligned}$$

Then, $X_{ik} = P_i \circ \phi^{-1} \circ (h\psi_k)(\delta\Omega_k) = h(P_i \circ \phi^{-1} \circ \psi_k)(\delta\Omega_k)$. Now, we compute $\frac{\partial \tilde{F}}{\partial \Omega_k}$ by evaluating $\frac{d}{d\tau}\big|_{\tau=0}\tilde{F}(\cdots\Omega_k(\tau)\cdots)$, and expressing $\frac{\partial L}{\partial U} : G \times \mathfrak{g} \to \mathfrak{g}^*$ as a left-trivialized cotangent vector.

Then,

$$\frac{d}{d\tau}\bigg|_{\tau=0}\tilde{F}(\cdots\Omega_k(\tau)\cdots) = h\sum_{i=1}^{s} b_i\left\langle \frac{\partial L}{\partial U}(U_i,\Omega_i), X_{ik}\right\rangle + hb_k\left\langle \frac{\partial L}{\partial \Omega}(U_k,\Omega_k), \delta\Omega_k\right\rangle$$
$$+\left\langle \Lambda, \mathrm{Asym}\Big(hg_1^T\sum_{i=1}^{s} b_iU_iX_{ik}\Omega_i + hg_1^T b_kU_k\delta\Omega_k\Big)\right\rangle,$$

where

$$\begin{aligned}\left\langle \Lambda, \mathrm{Asym}\Big(hg_1^T\sum_{i=1}^{s} b_iU_iX_{ik}\Omega_i + hg_1^T b_kU_k\delta\Omega_k\Big)\right\rangle &= \Lambda\odot\Big(hg_1^T\sum_{i=1}^{s} b_iU_iX_{ik}\Omega_i + hg_1^T b_kU_k\delta\Omega_k\Big)\\ &= h\sum_{i=1}^{s} b_i(U_i^Tg_1\Lambda\Omega_i^T)\odot X_{ik} + hb_k(U_k^Tg_1\Lambda)\odot\delta\Omega_k\\ &= h\sum_{i=1}^{s} b_i\langle \mathrm{Asym}(U_i^Tg_1\Lambda\Omega_i^T), X_{ik}\rangle + hb_k\langle \mathrm{Asym}(U_k^Tg_1\Lambda), \delta\Omega_k\rangle,\end{aligned}$$

so that

$$\begin{aligned}\frac{d}{d\tau}\bigg|_{\tau=0}\tilde{F}(\cdots\Omega_k(\tau)\cdots) &= h\sum_{i=1}^{s} b_i\Big\langle \frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^Tg_1\Lambda\Omega_i^T), X_{ik}\Big\rangle\\ &\quad + hb_k\Big\langle \frac{\partial L}{\partial \Omega}(U_k,\Omega_k) + \mathrm{Asym}(U_k^Tg_1\Lambda), \delta\Omega_k\Big\rangle\\ &= h\sum_{i=1}^{s} b_i\Big\langle \frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^Tg_1\Lambda\Omega_i^T), h(P_i\circ\phi^{-1}\circ\psi_k)(\delta\Omega_k)\Big\rangle\\ &\quad + hb_k\Big\langle \frac{\partial L}{\partial \Omega}(U_k,\Omega_k) + \mathrm{Asym}(U_k^Tg_1\Lambda), \delta\Omega_k\Big\rangle\\ &= h^2\sum_{i=1}^{s} b_i\Big\langle (P_i\circ\phi^{-1}\circ\psi_k)^*\Big(\frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^Tg_1\Lambda\Omega_i^T)\Big), \delta\Omega_k\Big\rangle\\ &\quad + hb_k\Big\langle \frac{\partial L}{\partial \Omega}(U_k,\Omega_k) + \mathrm{Asym}(U_k^Tg_1\Lambda), \delta\Omega_k\Big\rangle.\end{aligned}$$

Thus, we can see that

$$\frac{\partial\tilde{F}}{\partial\Omega_k} = h^2\sum_{i=1}^{s} b_i(P_i\circ\phi^{-1}\circ\psi_k)^*\Big(\frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^Tg_1\Lambda\Omega_i^T)\Big) + hb_k\Big(\frac{\partial L}{\partial \Omega}(U_k,\Omega_k) + \mathrm{Asym}(U_k^Tg_1\Lambda)\Big)$$

and (4.14a) reads as

$$h\sum_{i=1}^{s} b_i(P_i\circ\phi^{-1}\circ\psi_k)^*\Big(\frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^Tg_1\Lambda\Omega_i^T)\Big) + b_k\Big(\frac{\partial L}{\partial \Omega}(U_k,\Omega_k) + \mathrm{Asym}(U_k^Tg_1\Lambda)\Big) = 0, \quad (4.17)$$

for $k = 1,2\dots s$. Recall that $P_i \circ \phi^{-1} \circ \psi_k : \mathrm{Asym}(n) \to \mathrm{Asym}(n)^s \to \mathrm{Asym}(n)^s \to \mathrm{Asym}(n)$ is a map from $\mathrm{Asym}(n)$ to $\mathrm{Asym}(n)$, and its dual is given by

$$(P_i \circ \phi^{-1} \circ \psi_k)^*(\Lambda) = \psi_k^* \circ (\phi^*)^{-1} \circ P_i^*(\Lambda).$$

Let us derive an explicit expression for $(P_i \circ \phi^{-1} \circ \psi_k)^*(\Lambda)$. For P_i^*, it is easy to verify that

$$P_i^*(\Lambda) = (0\dots 0, \Lambda, \dots 0), \tag{4.18}$$

where the Λ is in the i-th position. For ϕ^*, $\forall (\Lambda_1, \Lambda_2 \dots \Lambda_s), (X_1, X_2 \dots X_s) \in \mathrm{Asym}(n)$,

$$\begin{aligned}
\langle \phi^*(\Lambda_1, \Lambda_2 \dots \Lambda_s), (X_1, X_2 \dots X_s) \rangle &= \Big\langle (\Lambda_1, \Lambda_2 \dots \Lambda_s), \Big[X_i - dP_{A_i}\Big(h \sum_{j=1}^{s} a_{ij} U_j X_j \Omega_j \Big) \Big]_{i=1}^{s} \Big\rangle \\
&= \sum_{i=1}^{s} \Big\langle \Lambda_i, X_i - dP_{A_i}\Big(h \sum_{j=1}^{s} a_{ij} U_j X_j \Omega_j \Big) \Big\rangle \\
&= \sum_{i=1}^{s} \langle \Lambda_i, X_i \rangle - \sum_{i=1}^{s} dP_{A_i}^*(\Lambda_i) \odot h \sum_{j=1}^{s} a_{ij} U_j X_j \Omega_j \\
&= \sum_{i=1}^{s} \langle \Lambda_i, X_i \rangle - \sum_{i,j=1}^{s} h a_{ij} U_j^T \, dP_{A_i}^*(\Lambda_i) \Omega_j^T \odot X_j \\
&= \sum_{i=1}^{s} \langle \Lambda_i, X_i \rangle - \sum_{j=1}^{s} \Big(h U_j^T \sum_{i=1}^{s} a_{ij} dP_{A_i}^*(\Lambda_i) \Omega_j^T \Big) \odot X_j \\
&= \sum_{j=1}^{s} \Big\langle \Lambda_j - \mathrm{Asym}\Big(h U_j^T \sum_{i=1}^{s} a_{ij} dP_{A_i}^*(\Lambda_i) \Omega_j^T \Big), X_j \Big\rangle,
\end{aligned}$$

so

$$\phi^*(\Lambda_1, \Lambda_2 \dots \Lambda_s) = \Big[\Lambda_j - \mathrm{Asym}(h U_j^T \sum_{i=1}^{s} a_{ij} dP_{A_i}^*(\Lambda_i) \Omega_j^T \Big]_{j=1}^{s} \tag{4.19}$$

For ψ_k^*, $\forall(\Lambda_1,\Lambda_2\dots\Lambda_s),\in \mathrm{Asym}(n), X\in \mathrm{Asym}(n)$,

$$\begin{aligned}\langle \psi_k^*(\Lambda_1,\Lambda_2\dots\Lambda_s),X\rangle &= \langle(\Lambda_1,\Lambda_2\dots\Lambda_s),[dP_{A_i}(a_{ik}U_kX)]_{i=1}^s\rangle\\ &= \sum_{i=1}^s dP_{A_i}^*(\Lambda_i)\odot(a_{ik}U_kX)\\ &= \sum_{i=1}^s a_{ik}U_k^T dP_{A_i}^*(\lambda_i)\odot X\\ &= \Big\langle \mathrm{Asym}\Big(U_k^T\sum_{i=1}^s a_{ik}dP_{A_i}^*(\Lambda_i)\Big),X\Big\rangle,\end{aligned}$$

so

$$\psi_k^*(\Lambda_1,\Lambda_2\dots\Lambda_s) = \mathrm{Asym}\Big(U_k^T\sum_{i=1}^s a_{ik}dP_{A_i}^*(\Lambda_i)\Big). \tag{4.20}$$

Combining (4.18), (4.19), (4.20), then $(P_i\circ\phi^{-1}\circ\psi_k)^*(\Lambda)$ can be implemented as

$$\Lambda_j - \mathrm{Asym}\Big(hU_j^T\sum_{l=1}^s a_{lj}dP_{A_l}^*(\Lambda_l)\Omega_j^T\Big) = \Lambda\cdot\delta_{ij}, \qquad j=1,2\dots s, \tag{4.21a}$$

$$(P_i\circ\phi^{-1}\circ\psi_k)^*(\Lambda) = \mathrm{Asym}\Big(U_k^T\sum_{l=1}^s a_{lk}dP_{A_l}^*(\Lambda_l)\Big). \tag{4.21b}$$

We can first calculate $\{\Lambda_i\}_{i=1}^s$ from (4.21a) by using fixed-point iteration, and then substitute the result into (4.21b) to obtain $(P_i\circ\phi^{-1}\circ\psi_k)^*(\Lambda)$.

Thus far, we have derived explicit formulae for (4.14a) and (4.14b), and now we derive one for (4.14d). Notice that U_i depends on g_0 by the nonlinear system $U_i = \mathbb{P}\Big(g_0 + h\sum_{j=1}^s a_{ij}U_j\Omega_j\Big)$. Consider a variation of g_0 by $g_0(\tau)$, such that $g_0(0)=g_0$, $\dot g_0(0) = g_0\delta g_0$, so that

$$U_i(\tau) = \mathbb{P}\Big(g_0(\tau) + h\sum_{j=1}^s a_{ij}U_j(\tau)\Omega_j\Big).$$

Differentiating on both sides, and letting $\dot U_i = U_iY_i$, where $Y_i\in\mathrm{Asym}(n)$ is a left-trivialized

tangent vector, we obtain

$$Y_i = dP_{A_i}(g_0\delta g_0 + h\sum_{j=1}^{s} a_{ij}U_jY_j\Omega_j),$$

which can be rewritten as

$$Y_i - dP_{A_i}(h\sum_{j=1}^{s} a_{ij}U_jY_j\Omega_j) = dP_{A_i}(g_0\delta g_0).$$

Similar to $X_{ik} = h(P_i \circ \phi^{-1} \circ \psi_k)(\Lambda)$, we introduce a new map

$$\varphi : \mathrm{Asym}(n) \to \mathrm{Asym}(n)^s, \quad \delta g_0 \mapsto \{dP_{A_i}(g_0\delta g_0)\}_{i=1}^s,$$

then $Y_i = P_i \circ \phi^{-1} \circ \varphi(\delta g_0)$. Let us derive φ^*, $\forall(\Lambda_1, \Lambda_2 \dots \Lambda_s) \in \mathrm{Asym}(n)^s, X \in \mathrm{Asym}(n)$,

$$\begin{aligned}
\langle \varphi^*(\Lambda_1, \Lambda_2 \dots \Lambda_s), X\rangle &= \langle (\Lambda_1, \Lambda_2 \dots \Lambda_s), \{dP_{A_i}(g_0X)\}_{i=1}^s\rangle \\
&= \sum_{i=1}^{s} dP^*_{A_i}(\Omega_i) \odot (g_0X) \\
&= \sum_{i=1}^{s} g_0^T dP^*_{A_i}(\Omega_i) \odot X \\
&= \Big\langle \mathrm{Asym}\Big(g_0^T \sum_{i=1}^{s} dP^*_{A_i}(\Lambda_i)\Big), X\Big\rangle.
\end{aligned}$$

Thus, $\varphi^*(\Lambda_1, \Lambda_2 \dots \Lambda_s) = \mathrm{Asym}(g_0^T \sum_{i=1}^{s} dP^*_{A_i}(\Lambda_i))$, and $(P_i \circ \phi^{-1} \circ \varphi)^*(\Lambda)$ can be implemented as

$$\begin{cases}
\Lambda_j - \mathrm{Asym}(hU_j^T \sum_{i=1}^{s} a_{ij} dP^*_{A_i}(\Lambda_i)\Omega_j^T) = \Lambda \cdot \delta_{ij}, \qquad j = 1, 2 \dots s, \\
(P_i \circ \phi^{-1} \circ \varphi)^*(\Lambda) = \mathrm{Asym}(g_0^T \sum_{i=1}^{s} dP^*_{A_i}(\Lambda_i)).
\end{cases}$$

We now compute

$$
\begin{aligned}
\frac{d}{d\tau}\Big|_{\tau=0}\tilde{F}(\cdots g_0(\tau)\cdots) &= h\sum_{i=1}^{s} b_i\Big\langle \frac{\partial L}{\partial U}(U_i,\Omega_i),Y_i\Big\rangle + \Big\langle \Lambda, \mathrm{Asym}\Big(g_1^T\Big(g_0\delta g_0 + h\sum_{i=1}^{s} b_i U_i Y_i \Omega_i\Big)\Big)\Big\rangle \\
&= h\sum_{i=1}^{s} b_i\Big\langle \frac{\partial L}{\partial U}(U_i,\Omega_i),Y_i\Big\rangle + \Lambda\odot\Big(g_1^T\Big(g_0\delta g_0 + h\sum_{i=1}^{s} b_i U_i Y_i \Omega_i\Big)\Big) \\
&= h\sum_{i=1}^{s} b_i\Big\langle \frac{\partial L}{\partial U}(U_i,\Omega_i),Y_i\Big\rangle + (g_0^T g_1\Lambda)\odot\delta g_0 + h\sum_{i=1}^{s} b_i(U_i^T g_1\Lambda\Omega_i^T)\odot Y_i \\
&= h\sum_{i=1}^{s} b_i\Big\langle \frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^T g_1\Lambda\Omega_i^T),Y_i\Big\rangle + \langle \mathrm{Asym}(g_0^T g_1\Lambda),\delta g_0\rangle \\
&= h\sum_{i=1}^{s} b_i\Big\langle \frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^T g_1\Lambda\Omega_i^T),P_i\circ\phi^{-1}\circ\varphi(\delta g_0)\Big\rangle + \langle \mathrm{Asym}(g_0^T g_1\Lambda),\delta g_0\rangle.
\end{aligned}
$$

From the above calculation, we can rewrite (4.14d) as

$$
-p_0 = h\sum_{i=1}^{s} b_i(P_i\circ\phi^{-1}\circ\varphi)^*\Big(\frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^T g_1\Lambda\Omega_i^T)\Big) + \mathrm{Asym}(g_0^T g_1\Lambda), \tag{4.22}
$$

where p_0 is a left-trivialized cotangent vector. For equation (4.14e), it is easy to show that

$$
p_1 = \mathrm{Asym}\Big(g_1^T\Big(g_0 + h\sum_{i=1}^{s} b_i U_i\Omega_i\Big)\Lambda^T\Big). \tag{4.23}
$$

Now, combining (4.17),(4.12),(4.13),(4.22),(4.23), we obtain a Lagrangian variational integrator on the rotation group $SO(n)$:

$$
\begin{cases}
0 = h\sum_{i=1}^{s} b_i(P_i\circ\phi^{-1}\circ\psi_k)^*\Big(\frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^T g_1\Lambda\Omega_i^T)\Big) + b_k\Big(\frac{\partial L}{\partial \Omega}(U_k,\Omega_k) + \mathrm{Asym}(U_k^T g_1\Lambda)\Big), \\
g_1 = \mathbb{P}\Big(g_0 + h\sum_{i=1}^{s} b_i U_i\Omega_i\Big), \\
U_i = \mathbb{P}\Big(g_0 + h\sum_{j=1}^{s} a_{ij} U_j\Omega_j\Big), \\
-p_0 = h\sum_{i=1}^{s} b_i(P_i\circ\phi^{-1}\circ\varphi)^*\Big(\frac{\partial L}{\partial U}(U_i,\Omega_i) + \mathrm{Asym}(U_i^T g_1\Lambda\Omega_i^T)\Big) + \mathrm{Asym}(g_0^T g_1\Lambda), \\
p_1 = \mathrm{Asym}\Big(g_1^T\Big(g_0 + h\sum_{i=1}^{s} b_i U_i\Omega_i\Big)\Lambda^T\Big).
\end{cases}
\tag{4.24}
$$

4.4 Hamiltonian variational integrator on the rotation group $SO(n)$

It is often desirable to transform a numerical method from the Lagrangian side to the Hamiltonian side. The same mechanical system can be represented either on the Lagrangian side or the Hamiltonian side, and they are related by the Legendre transform. In Euclidean space, this gives

$$(TQ,L) \underset{\mathbb{F}H}{\overset{\mathbb{F}L}{\rightleftarrows}} (T^*Q,H)$$

and we have the following relationship,

$$\frac{\partial L}{\partial \dot{q}}(q,\dot{q}) = p, \quad \frac{\partial H}{\partial p}(q,p) = \dot{q}, \quad \frac{\partial L}{\partial q}(q,\dot{q}) = -\frac{\partial H}{\partial q}(q,p).$$

Now on Lie groups, for a left-trivialized Lagrangian $L : G \times \mathfrak{g} \to \mathbb{R}$ and Hamiltonian $H : G \times \mathfrak{g}^* \to \mathbb{R}$, it is easy to verify that similar relations hold

$$\frac{\partial L}{\partial \varepsilon}(g,\varepsilon) = \mu, \quad \frac{\partial H}{\partial \mu}(g,\mu) = \varepsilon, \quad \frac{\partial L}{\partial g}(g,\varepsilon) = -\frac{\partial H}{\partial g}(g,\mu). \tag{4.25}$$

Using (4.25), and denoting the corresponding internal cotangent vectors by $\{\mu_j\}_{j=1}^s$, (4.24) can be transformed to the Hamiltonian side,

$$\begin{cases} \mu_k = -\mathrm{Asym}(U_k^T g_1 \Lambda)) + h\sum_{i=1}^s \frac{b_i}{b_k}(P_i \circ \phi^{-1} \circ \psi_k)^* \Big(\frac{\partial H}{\partial U}(U_i,\mu_i) - \mathrm{Asym}(U_i^T g_1 \Lambda \Omega_i^T)\Big), \\ g_1 = \mathbb{P}\Big(g_0 + h\sum_{i=1}^s b_i U_i \Omega_i\Big), \\ U_i = \mathbb{P}\Big(g_0 + h\sum_{j=1}^s a_{ij} U_j \Omega_j\Big), \\ \mathrm{Asym}(g_0^T g_1 \Lambda) = -p_0 + h\sum_{i=1}^s b_i (P_i \circ \phi^{-1} \circ \varphi)^* \Big(\frac{\partial H}{\partial U}(U_i,\mu_i) - \mathrm{Asym}(U_i^T g_1 \Lambda \Omega_i^T)\Big), \\ p_1 = \mathrm{Asym}\Big(g_1^T \Big(g_0 + h\sum_{i=1}^s b_i U_i \Omega_i\Big)\Lambda^T\Big), \\ \Omega_i = \frac{\partial H}{\partial \mu}(U_i,\mu_i). \end{cases} \tag{4.26}$$

In the above algorithm, Ω_i is given explicitly by $\frac{\partial H}{\partial U}(U_i, \mu_i)$ and only serves to reduce the redundancy in the computations as it is used numerous times in other expressions. The first four equations are solved simultaneously by fixed-point iterations, i.e., the variables $(\{\mu_k\}_{k=1}^s, g_1, \{U_i\}_{i=1}^s, \Lambda)$ are updated concurrently in each iteration. Also observe that the equation for Λ is a Lyapunov equation. For the second to last equation of (4.26), p_1 can be calculated explicitly after solving for $(\{\mu_k\}_{k=1}^s, g_1, \{U_i\}_{i=1}^s, \Lambda)$.

Now we consider a G-invariant Hamiltonian system given by H on the contangent bundle T^*G. In this case, Hamilton's equations can be reduced to a Lie–Poisson system on $\mathfrak{g}^*$. As we know, if the discrete Lagrangian $L_d(g_0, g_1)$ is also G-invariant, (4.2) naturally reduces to yield a Lie–Poisson integrator. For the discrete Lagrangian we have constructed,

$$L_d(g_0, g_1) =_{\{\Omega_i\}_{i=1}^s} \sum_{i=1}^s b_i l(\Omega_i),$$

where

$$U_i = \mathbb{P}\Big(g_0 + h\sum_{j=1}^s a_{ij} U_j \Omega_j\Big),$$
$$g_1 = \mathbb{P}\Big(g_0 + h\sum_{i=1}^s b_i U_i \Omega_i\Big),$$

and where $l : \mathfrak{g} \to \mathbb{R}$ is the reduced Lagrangian, it is easy to verify that it is G-invariant, i.e.,

$$L_d(g_0, g_1) = L_d(g \cdot g_0, g \cdot g_1) =_{\{\Omega_i\}_{i=1}^s} \sum_{i=1}^s b_i l(\Omega_i),$$

where

$$g \cdot U_i = \mathbb{P}\Big(g \cdot g_0 + h\sum_{j=1}^s a_{ij} g \cdot U_j \Omega_j\Big),$$
$$g \cdot g_1 = \mathbb{P}\Big(g \cdot g_0 + h\sum_{i=1}^s b_i g \cdot U_i \Omega_i\Big).$$

Thus, the variational integrator (4.26) can theoretically be reduced to a Lie–Poisson integrator. By letting $g_0^T g_1 = f_0$, $U_i^T g_1 = \Theta_i$, (4.26) can be simplified as follows

$$\begin{cases} \mu_k = -\mathrm{Asym}(\Theta_k \Lambda) - h\sum_{i=1}^{s} \frac{b_i}{b_k}(P_i \circ \phi^{-1} \circ \psi_k)^*(\mathrm{Asym}(\Theta_i \Lambda \Omega_i^T), \\ 0 = \mathrm{Asym}\Big(f_0^T + h\sum_{i=1}^{s} b_i \Theta_i^T \Omega_i\Big), \\ \Theta_i^T = \mathbb{P}\Big(f_0^T + h\sum_{j=1}^{s} a_{ij}\Theta_j^T \Omega_j\Big), \\ \mathrm{Asym}(f_0\Lambda) = -p_0 - h\sum_{i=1}^{s} b_i (P_i \circ \phi^{-1} \circ \varphi)^*(\mathrm{Asym}(\Theta_i \Lambda \Omega_i^T), \\ p_1 = \mathrm{Asym}\Big(\Big(f_0^T + h\sum_{i=1}^{s} b_i \Theta_i^T \Omega_i\Big)\Lambda^T\Big), \\ \Omega_i = \frac{\partial h}{\partial \mu}(\mu_i). \end{cases} \tag{4.27}$$

Multiplying by g_1^T on both sides of $g_1 = \mathbb{P}(g_0 + h\sum_{i=1}^{s} b_i U_i \Omega_i)$ yields

$$\mathbb{I} = \mathbb{P}\Big(f_0^T + h\sum_{i=1}^{s} b_i \Theta_i^T \Omega_i\Big).$$

Suppose that h is small, and g_0, g_1 are close, then $f_0^T + h\sum_{i=1}^{s} b_i \Theta_i^T \Omega_i$ is in the neighborhood of $\mathbb{I}$, and by Lemma 1, this is equivalent to

$$\mathrm{Asym}\Big(f_0^T + h\sum_{i=1}^{s} b_i \Theta_i^T \Omega_i\Big) = 0,$$

which can be regarded as equation for f_0. The first four equations can be solved using fixed-point iteration, with iteration variables $(\{\mu_k\}_{k=1}^s, f_0, \{\Theta_i\}_{i=1}^s, \Lambda)$. From this, p_1 can be calculated explicitly. In the above algorithm, we also need to figure out the reduced version of $(P_i \circ \phi^{-1} \circ \psi_k)^*$ and $(P_i \circ \phi^{-1} \circ \varphi)^*$. Notice that (4.21a), (4.21b) involves $U_j^T dP_{A_i}^*(\Lambda)$. Since $\{g_0, g_1, U_i\}$ and $A_i = g_0 + h\sum_{j=1}^{s} a_{ij} U_j \Omega_j = U_i P_i$ are reduced, we need a reduced version of $U_j^T dP_{A_i}^*(\Lambda)$ as

well. Multiplying A_i on the left by g_1^T, we obtain,

$$g_1^T A_i = f_0^T + h\sum_{j=1}^{s} a_{ij}\Theta_j^T\Omega_j = (g_1^T U_i)P_i.$$

So, $(g_1^T U_i)P_i$ is the polar decomposition of $f_0^T + h\sum_{j=1}^{s} a_{ij}\Theta_j^T\Omega_j$, and

$$\begin{aligned} U_j^T dP_{A_i}^*(\Lambda) &= U_j^T \cdot U_i \mathrm{Lyap}(P_i,\Lambda^T) = \Theta_j g_1^T U_i \mathrm{Lyap}(P_i,\Lambda^T), \\ &= \Theta_j \mathbb{P}\Big(f_0^T + h\sum_{j=1}^{s} a_{ij}\Theta_j^T\Omega_j\Big)\mathrm{Lyap}(P_i,\Lambda^T). \end{aligned}$$

This is the reduced version of $U_j^T dP_{A_i}^*(\Lambda)$ and so $(P_i\circ\phi^{-1}\circ\psi_k)^*(\Lambda)$ can be implemented as

$$\Lambda_j - \mathrm{Asym}\Big(h\Theta_j\sum_{l=1}^{s} a_{lj} dP_{A_l}^*(\Lambda_l)\Omega_j^T\Big) = \Lambda\cdot\delta_{ij}, \qquad j = 1,2\ldots s, \tag{4.28a}$$

$$(P_i\circ\phi^{-1}\circ\psi_k)^*(\Lambda) = \mathrm{Asym}\Big(\Theta_k\sum_{l=1}^{s} a_{lk} dP_{A_l}^*(\Lambda_l)\Big), \tag{4.28b}$$

where $\{A_i\}$ are redefined to be $A_i = f_0^T + h\sum_{j=1}^{s} a_{ij}\Theta_j^T\Omega_j$. For $(P_i\circ\phi^{-1}\circ\varphi)^*$, we need to compute $g_0^T dP_{A_i}^*(\Lambda)$, which is given by

$$g_0^T dP_{A_i}^*(\Lambda) = g_0^T\cdot U_i\mathrm{Lyap}(P_i,\Lambda^T) = f_0 g_1^T U_i\mathrm{Lyap}(P_i,\Lambda^T) = f_0\mathbb{P}\Big(f_0^T + h\sum_{j=1}^{s} a_{ij}\Theta_j^T\Omega_j\Big)\mathrm{Lyap}(P_i,\Lambda^T).$$

4.5 Numerical experiment

Consider a free rigid body problem, with inertia matrix $J = \mathrm{diag}(1,2,3)$, and initial angular momentum $\mu = (0.5,-0.5,0.5)$. The reduced Hamiltonian is given by $h(\mu) = \frac{1}{2}\mu^T J^{-1}\mu$, then the equations of motion in Hamiltonian form [35] is given by,

$$\dot{\mu} + (J^{-1}\mu)\times\mu = 0.$$

The system of equations (4.27) can be applied to solve this problem. Here, we choose the coefficients $\{a_{ij}, b_i\}$ to be the Gauss–Legendre method with two stages, this is a fourth-order method in classical numerical ODE theory. We set the timestep $h = 0.2$, and performed numerical integration over the interval $[0, 100]$. When implementing (4.27), we reduced the complexity of the computation by using the following observation,

$$(P_i \circ \phi^{-1} \circ \psi_k)^*(\mathrm{Asym}(\Theta_i \Lambda \Omega_i^T) = \psi_k^* \circ (P_i \circ \phi^{-1})^* \mathrm{Asym}(\Theta_i \Lambda \Omega_i^T),$$
$$(P_i \circ \phi^{-1} \circ \varphi)^*(\mathrm{Asym}(\Theta_i \Lambda \Omega_i^T) = \varphi^* \circ (P_i \circ \phi^{-1})^* \mathrm{Asym}(\Theta_i \Lambda \Omega_i^T).$$

Notice that $(P_i \circ \phi^{-1})^* \mathrm{Asym}(\Theta_i \Lambda \Omega_i^T)$ can be obtained by solving the fixed-point equation (4.28a), and it need only be calculated once.

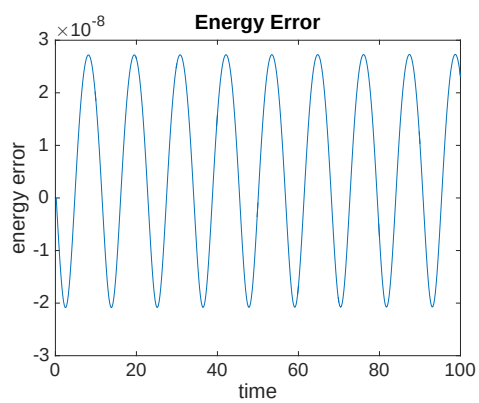


Figure 4.1. Energy Error

Figure 4.1 shows the energy error for algorithm (4.27), and Figure 4.2 shows the trajectory error of algorithm (4.27) when compared to a reference solution obtained by using a high-resolution numerical simulation. We observe that the energy is preserved quite well, which is a common feature of symplectic integrators, and the trajectory error grows linearly in time. Figure 4.3 shows the maximum energy error and trajectory error on the fixed interval $[0, 100]$ for different timesteps. The numerical results indicate that the error decreases with smaller timesteps,

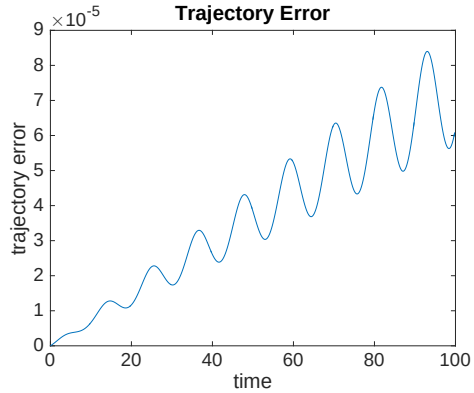


Figure 4.2. Trajectory Error

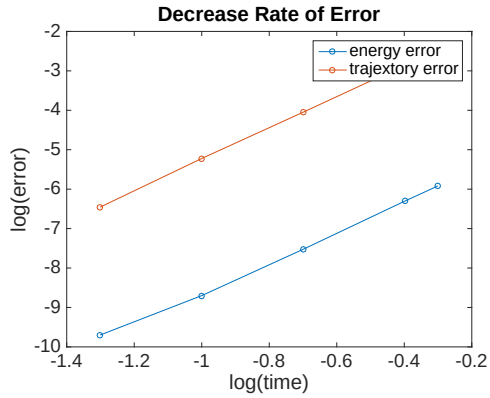


Figure 4.3. Error convergence rate

at approximately a fourth-order rate of convergence, which matches the classical order of the two stage Gauss–Legendre method.

Chapter 4, in full, is a reprint of the material that has been submitted for publication to Journal of Computational Dynamics, 2018. Xuefeng Shen; Melvin Leok, American Institute of Mathematical Sciences , 2018. The dissertation author was the primary investigator and author of this material.

Chapter 5

Geometric Symmetry Reduction of the Unobservable Subspace for Kalman Filtering

5.1 Introduction

Kalman filters are widely used in practical state estimation problems, including SLAM (Simultaneous Localization and Mapping). This is primarily due to its conceptual simplicity and the low computational complexity compared to optimization based methods. Various types of sensors are used for localization tasks, including GPS, odometry, inertial measurement units (IMU), cameras, Light Detection and Ranging (LIDAR), and so on, and the specific choice of sensors used depends on the application. Kalman filters have been successfully applied to Visual Odometry (VO) [12], Visual Inertial Odometry (VIO) [48], and more recently, to autonomous driving cars, where information from multiple sensors (GPS, odometry, IMU, and cameras) and HD maps are integrated together to yield a more robust and accurate result.

A Kalman filter is derived based on the assumptions that the probability distribution is Gaussian and the dynamics are linear. However, practical models generally involve nonlinear dynamics and non-Gaussian distributions, and for such applications, Kalman filters are less accurate than optimization based methods.

There have been ongoing efforts to improve the performance of Kalman filters. The unscented Kalman filter (UKF) [63] is used to reduce the local linearization error of the extended Kalman filer (EKF); inverse depth parametrization [47] is used to represent the three-dimensional

spatial location of feature points recognized by camera, as a Gaussian distribution on Euclidean space about a feature point's position does not model its depth uncertainty well; the multi-state constraint Kalman filter (MSCKF) [48] method only maintains a sliding window of historic camera poses, and removes feature points from the state space. Multiple views of the same feature point gives a constraint on the state variable, and serves as measurement. Such an approach can be viewed as an attempt to further relax the assumption that the uncertainty in the feature point positions are described by Gaussian distributions and thereby improve the algorithm's performance.

There is also an effort to improve the consistency of the Kalman filter by taking into account the issue of observability. The EKF does not respect the observability properties of the underlying continuous control system, which is believed to be a major source of inconsistency in the state estimate. It is hoped that by correctly incorporating the observability property into the construction of the algorithm, the consistency and hence accuracy of the method will be improved. An example of this phenomena is given in [30], where a stationary robot measured the relative position of new landmarks multiple times, and the covariance of robot's pose estimate becomes smaller over time. This is contrary to our intuition, as the the robot is static and therefore new landmarks do not provide new information about the robot's pose, and re-observation of these landmarks should not affect robot's own pose estimation. So, the pose of the robot is unobservable given such measurements, but the algorithm mistakenly treats it as an observable variable.

This phenomena has been studied in depth using observability analysis [24]. It turns out three dimensions of the state space, i.e., the robot's position and its orientation, are unobservable to odometry and camera measurements, while for the linearized discrete model, only two dimensions (the robot's position) remain unobservable. This was addressed by considering two modifications, one is the special transition matrix $\Phi(\check{x}_{k+1}, \hat{x}_k)$ that depends on $\hat{x}_k$ (modified state value at k-th step) and $\check{x}_{k+1}$ (predicted state value at $(k+1)$-th step). If we denote the unobservable subspace at $\hat{x}_k$ by $\mathscr{N}_k$, then after the measurement update, $\Phi(\check{x}_{k+1}, \hat{x}_k)\mathscr{N}_k$ is still perpendicular

to $dh(\check{x}_{k+1})$. More importantly, this transition matrix satisfies $\Phi(x_{k+2},x_{k+1}) \cdot \Phi(x_{k+1},x_k) = \Phi(x_{k+2},x_k)$, thus along the true trajectory, we have

$$\Phi(x_{k+s},x_{k+s-1}) \dots \Phi(x_{k+2},x_{k+1}) \cdot \Phi(x_{k+1},x_k)\mathscr{N}_k = \Phi(x_{k+s},x_k)\mathscr{N}_k \perp dh(x_{k+s}), \tag{5.1}$$

which means the unobservable subspace remains unobservable during the propagation, which is a discrete version of the invariance property satisfied by the continuous control system. However, since the Kalman filter is implemented by alternating propagation and measurement update, the Jacobians are calculated at both $\hat{x}_k$ and $\check{x}_k$, and as a consequence, property (5.1) no longer holds; based on this observation, the other modification is the First Estimate Jacobian (FEJ) framework, where Jacobians are evaluated on the first ever available estimate of all the state variables so to make property (5.1) hold again. Later, a revised technique termed observability constrained (OC)-EKF [25] was proposed to not only guarantee the desired observability property but also to minimize the expected errors of the linearization points. These techniques were combined with the MSCKF method to obtain MSCKF2.0 [41], which is an efficient algorithm that claims to have performance no worse than optimization-based methods.

In this paper, we try to approach the observability problem from a different perspective. Notice that the Kalman filter is composed of alternating propagation and measurement update steps. While the propagation step is simply a discretization of underlying stochastic differential equation, the measurement update step is essentially Bayesian inference. Such a measurement update step does not exist in the continuous control system when we perform observability analysis, so the propagation of system is "interrupted" at each step, so forcing this discrete system to mimic the observability property of the continuous system is somewhat artificial. Since the measurement update step is essentially Bayesian inference, we would try to solve the observability problem at this step from the perspective of probability theory. The main idea is that by viewing the unobservability of the system as an invariance with respect to a group action, the state space can be decomposed into unobservable and observable parts explicitly,

$x = (x_N, x_O)$. As in the case of symmetry reduction in geometric mechanics, we can obtain reduced equations that only involve the reduced (or observable) variables, and the propagation and measurement equations will be expressed only in terms of the observable variables x_O. We claim that in order to keep x_N unobservable during the algorithm, i.e., avoid introducing spurious information from measurements to x_N, Bayesian inference should only be employed on the x_O part, which yields a reduced measurement model. Mathematically, it means, for a probability distribution $p(x_N, x_O)$, which can be factored into

$$p(x_N, x_O) = p(x_O) \cdot p(x_N \mid x_O),$$

the conditional distribution $p(x_N \mid x_O)$ shall remain unchanged during the measurement update, and only $p(x_O)$ can be improved by the reduced measurement. So after the measurement update

$$\overline{p}(x_N, x_O) = \overline{p}(x_O) \cdot p(x_N \mid x_O), \tag{5.2}$$

where $\overline{p}(x_O)$ is the improved distribution for the observable variables. Even though measurement is not applied directly to x_N, improvement in the estimation of x_O helps to improve the estimation of x_N implicitly via the conditional probability $p(x_N \mid x_O)$, which is a consequence of the correlation between x_N and x_O. A geometric picture may help to understand this, the unobservable subspaces are tangent spaces to a collection of submanifolds in state space, and x_N denotes the degrees of freedom on these submanifold. Each point on a given submanifold will yield the same measurement results. The x_O indicates which submanifold the state is in, and the different submanifolds are distinguishable using the measurement data. Suppose we have a probability distribution on state space, if we are more certain about x_O, i.e., which submanifold we are in, it also improves our estimate about x_N as long as they are not independent. By using this framework, the inconsistency counterexample in [30] is naturally avoided. In that case, the unobservable variables are $x_N = (p, \theta)$, the position and orientation of robot, and the observable

variables are $x_O = (z_1, z_2 \ldots, z_N)$, where z_i is the relative position of i-th feature point. When new features are observed, new z_i will be appended to the state variables, and they are independent from the old state variables. As long as the robot remains static, i.e., no propagation of the uncertainty is performed, no matter how often we re-observe these feature points and improve their relative position estimation, it will not affect the pose of the robot itself.

In Section 5.2, we review the planar robot model from [25], together with its observability property. We stress the use of group actions to characterize unobservable subspaces. In Section 5.3, we discuss how to obtain a reduced control system by considering the invariant group action, and how to decompose the planar robot system into the x_N and x_O parts. In Section 5.4, we choose an appropriate Riemannian structure for the new coordinate system, and implement reduced Bayesian inference (5.2) for the Gaussian distribution. Numerical experiments are given in Section 5.5, where straight line, circle and general trajectories are tested, and the reduced EKF method demonstrates superior performance when compared to the classical EKF and FE-EKF method.

5.2 Background

A planar robot is considered, equipped with odometry and cameras, navigating in an unknown environment. The odometry measures local velocity and angular velocity of the robot, while cameras detect feature points and measure their relative position to the robot. To make the notation simple and the numerical simulation easier to implement, we assume that the robot is equipped with binocular cameras, thus at each instant, measurements from two cameras will be combined to give an estimate of the relative position of feature points, and this serves as our measurement. Fix a global frame, and assign an intrinsic frame $\{e_1, e_2\}$ to the robot. The robot pose is denoted by (p, θ), where p is the global position of the robot, and θ represents the relative angle from the global frame to the intrinsic frame. Thus, the intrinsic frame of the robot

is given by

$$\{e_1, e_2\} = R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The global position of feature points are denoted by $(p_1^f, p_2^f, \dots p_N^f)$, and together, they form the classical state space of this problem

$$X = (p, \theta, p_1^f, p_2^f, \dots p_N^f).$$

The total dimension of X is $3+2N$, and this state space has the structure $SE(2) \times \mathbb{R}^{2N}$, where $SE(2)$ is the two-dimensional Euclidean group. Denote the local velocity and angular velocity measured using odometry by v and w, and the relative position of the i-th feature point by z_i. Then, the underlying continuous control system is given by

$$\begin{cases} \dot{p} = R(\theta)v, \\ \dot{\theta} = w, \\ \dot{p}_i^f = 0; \end{cases} \tag{5.3}$$

$$z_i = R(\theta)^T (p_i^f - p), \tag{5.4}$$

where (5.3) is the propagation equation, and (5.4) is the measurement equation.

As we have already mentioned, there are three unobservable dimensions in this model, the robot position p and its orientation θ. There are two main ways to detect unobservable dimensions in a given control system. One is the observability rank criterion [20]. Given a general affine control system

$$\begin{cases} x' = f(x) + \sum_{i=1}^{n} g_i(x) u_i, \\ z = h(x), \end{cases} \tag{5.5}$$

regard $f(x)$, $g_1(x)$, $g_2(x)$, $\dots$, $g_n(x)$ as vector fields on the state space X, then we construct the

minimal dual distribution that is invariant under $f(x)$, $g_1(x)$, $g_2(x)$, $\ldots$, $g_n(x)$, which is given by

$$\Omega = \langle f, g_1, g_2 \ldots, g_n \mid dh \rangle.$$

This step involves repeated Lie derivative calculations, and is analogous to computing the observability matrix for constant linear systems. Its orthogonal distribution $\Omega^{\perp}$, which is analogous to the null space of the observability matrix, defines the unobservable subspace for this system. There exists a corresponding local coordinate transformation $\phi(x) = (x_1, x_2)$ to the distribution $\Omega^{\perp}$, such that the system (5.5) can be decomposed as follows

$$\begin{cases} x_1' = \overline{f}_1(x_1, x_2) + \sum_{i=1}^{n} \overline{g}_{1i}(x_1, x_2) u_i, \\ x_2' = \overline{f}_2(x_2) + \sum_{i=1}^{n} \overline{g}_{2i}(x_2) u_i, \\ z = \overline{h}(x_2). \end{cases} \tag{5.6}$$

It is clear that the measurement only depends on the x_2 part, and x_2 evolves by itself, which is unaffected by x_1. For any solution $(x(t), u(t), z(t))$ that satisfies the system (5.5), if we move the initial state $x(t_0)$ along the unobservable submanifold that passes through it, i.e., by only changing the x_1 part, the new trajectory corresponds to the same control $u(t)$ and measurement $z(t)$, and thus x_1 is unobservable in this system. Any physical property of this system that is observable has to be independent of x_1, in other words, has to be constant along each unobservable submanifold.

It has been verified in [25] that for the system (5.3), (5.4),

$$\Omega^{\perp} = \mathscr{N} = \text{span} \begin{pmatrix} I_2 & Jp \\ \mathbf{0} & 1 \\ I_2 & Jp_1^f \\ I_2 & Jp_2^f \\ \vdots & \vdots \\ I_2 & Jp_N^f \end{pmatrix}, \tag{5.7}$$

where

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

The first two columns indicate that the robot position p is unobservable, and the third column indicates that the robot orientation is also unobservable.

The disadvantage of the observability rank criterion technique is it involves massive amounts of Lie derivative calculations. An easier way to detect unobservable dimensions is by invariant group actions. In classical mechanics, for Hamiltonian systems, we have the famous Noether's theorem, which essentially relates symmetries to conserved quantities. For example, if the system is invariant under translation, then the total linear momentum is preserved by the system; if system is invariant under rotations, then the total angular momentum is also be preserved. A similar idea can be applied to observability analysis. If our control system is invariant under some group action, i.e., corresponds to the same control $u(t)$ and measurement $z(t)$, then we have discovered an unobservable dimension of this system. For the planar robot

model, we can verify that under the following translation and rotation,

$$\begin{cases} p \mapsto p + \Delta p, \\ \theta \mapsto \theta, \\ p_i^f \mapsto p_i^f + \Delta p; \end{cases} \qquad \begin{cases} p \mapsto R(\Delta\theta)p, \\ \theta \mapsto \theta + \Delta\theta, \\ p_i^f \mapsto R(\Delta\theta)p_i^f, \end{cases} \tag{5.8}$$

(5.3) (5.4) remains invariant. For each state $(p, \theta, p_1^f, p_2^f, \dots p_N^f)$, we can calculate the tangent subspace induced by the above group action. In particular, the translation action gives the first two columns of (5.7), and the rotation action gives the third column of (5.7). It is sometimes natural to find these invariant group actions by physical intuition. Odometry measures local velocity and local angular velocity, while the camera measures relative position of feature points, as such, the information that they provide is "local" and cannot give constraints on the global pose. Global position can be measured by using GPS or if a map provides the absolute position of feature points. In the case of VIO, the IMU provides measurements of the local acceleration and angular velocity. Since the IMU measurement can detect the gravity direction, gravity breaks the full rotational symmetry, so only rotations around the gravity direction keep the system invariant, which indicates that the yaw angle is unobservable in VIO applications. The main disadvantage of the invariant group action technique as a method of characterizing unobservable dimensions is that we cannot determine if we have found all the unobservable dimensions.

We now describe the special transition matrix for (5.3) that satisfies the chain rule property (5.1). Let $v = v_m + n_v$, $w = w_m + n_w$, where v_m,w_m are odometry measurements, and n_v, n_w are measurement noise. A simple forward Euler integration of (5.3) gives

$$\begin{cases} p_{k+1} = p_k + \Delta t \cdot R(\theta_k)(v_m^{k+1} + n_v), \\ \theta_{k+1} = \theta_k + \Delta t \cdot (w_m^{k+1} + n_w), \\ p_{i,k+1}^f = p_{i,k}^f, \end{cases} \tag{5.9}$$

so, the nominal values are updated as follows

$$\begin{cases} p_{k+1} = p_k + R(\theta_k)\Delta d_{k+1}, \\ \theta_{k+1} = \theta_k + \Delta\theta_{k+1}, \\ p^f_{i,k+1} = p^f_{i,k}, \end{cases} \tag{5.10}$$

with $\Delta d_{k+1} = v_m^{k+1} \cdot \Delta t$, $\Delta\theta_{k+1} = w_m^{k+1} \cdot \Delta t$. The transition matrix for (5.10) is

$$\Phi = \begin{pmatrix} I_2 & JR(\theta_k)\Delta d_{k+1} & \\ \mathbf{0} & 1 & \\ & & I_{2N} \end{pmatrix} = \begin{pmatrix} I_2 & J(p_{k+1} - p_k) & \\ \mathbf{0} & 1 & \\ & & I_{2N} \end{pmatrix} = \Phi(x_{k+1}, x_k), \tag{5.11}$$

The derivative of the measurement (5.4) is given by

$$\begin{aligned} dh_i(x) &= \begin{pmatrix} -R(\theta)^T, & R(\theta)^T J^T (p_i^f - p), & \mathbf{0}, & \dots & R(\theta)^T \dots \end{pmatrix} \\ &= R(\theta)^T \begin{pmatrix} -I_2, & J^T (p_i^f - p), & \mathbf{0}, & \dots & I_2 \dots \end{pmatrix}, \end{aligned}$$

where I_2 appears at the i-th index of the feature points. We first check that $dh_i \cdot \mathcal{N} = 0$, which is theoretically guaranteed; then the nullspace $\mathcal{N}_k$ at x_k after applying the transition matrix $\Phi(x_{k+1}, x_k)$ is

$$\Phi(x_{k+1}, x_k) \cdot \mathcal{N}_k = \begin{pmatrix} I_2 & J(p_{k+1} - p_k) & \\ \mathbf{0} & 1 & \\ & & I_{2N} \end{pmatrix} \cdot \begin{pmatrix} I_2 & Jp_k \\ \mathbf{0} & 1 \\ I_2 & Jp^f_{1,k} \\ I_2 & Jp^f_{2,k} \\ \vdots & \vdots \\ I_2 & Jp^f_{N,k} \end{pmatrix} = \begin{pmatrix} I_2 & Jp_{k+1} \\ \mathbf{0} & 1 \\ I_2 & Jp^f_{1,k} \\ I_2 & Jp^f_{2,k} \\ \vdots & \vdots \\ I_2 & Jp^f_{N,k} \end{pmatrix},$$

it is easy to verify that $dh_i(x_{k+1}) \cdot \Phi(x_{k+1}, x_k) \cdot \mathcal{N}_k = 0$, since $p^f_{i,k+1} = p^f_{i,k}$. Finally, we can also

verify that transition matrix $\Phi(x_{k+1}, x_k)$ (5.11) satisfies $\Phi(x_{k+2}, x_k) = \Phi(x_{k+2}, x_{k+1}) \cdot \Phi(x_{k+1}, x_k)$, so the chain rule property (5.1) is satisfied along the exact trajectory.

5.3 Geometric reduction

Given the state space X, and a general control system

$$\begin{cases} x' = f(x, u), \\ z = h(x), \end{cases} \tag{5.12}$$

suppose we have a left Lie group action of G on X, i.e., $G \times X \to X$, which keeps the system (5.12) invariant, i.e., for any $(x(t), u(t), z(t))$ that satisfies (5.12), $(g \cdot x(t), u(t), z(t))$ also satisfies (5.12) for $\forall g \in G$, i.e.,

$$\begin{cases} (g \cdot x)' = f(g \cdot x, u), \\ z = h(g \cdot x). \end{cases} \tag{5.13}$$

For each point $x \in X$, $\forall \xi \in \mathfrak{g}$, where $\mathfrak{g}$ is the Lie algebra of G, the infinitesimal generator $\xi_X(x) = \frac{d}{dt}\big|_{t=0} \exp(\xi(t)) \cdot x$ gives one unobservable direction of x, and the orbit of x,

$$\mathrm{Orb}(x) = \{g \cdot x \mid g \in G\},$$

gives the unobservable submanifold that passes through x. This invariant group action on the control system allows us to perform reduction to express the system in terms of reduced variables on the quotient space X/G. This kind of reduction due to the presence of a continuous symmetry group arises in many disciplines. In optimization, when objective function is invariant under a group action, the problem can be rephrased as an optimization problem on the quotient space [2]; also for classical mechanics on a Lie group, when Lagrangian or Hamiltonian is invariant under the group action, the mechanics can be reduced to the Lie algebra $\mathfrak{g}$ or its dual $\mathfrak{g}^*$, which is referred to as Euler–Poincaré reduction and Lie–Poisson reduction [46], respectively. Here,

suppose that the group action $G \times X \to X$ is free and proper, then we obtain a smooth quotient space X/G, with a quotient map[34]

$$\pi : X \to X/G$$

that is a smooth submersion. Then, the control system (5.12) can be reduced to the quotient space X/G. For $[x] \in X/G$,

$$\begin{cases} [x]' = f([x], u), \\ \quad z = h([x]). \end{cases} \tag{5.14}$$

(5.14) is a reduced control system with reduced propagation and measurement equations. Consider (5.3), (5.4) as a concrete example. The state space is $X = SE(2) \times \mathbb{R}^{2N}$, and we already know that the group action $G = SE(2)$ (5.8) on X leaves (5.3) (5.4) invariant, so this induces a reduced control system on X/G. However, X/G is an abstract quotient manifold, so in order to deal with it explicitly, we need a concrete coordinate representation. It turns out that the relative position of feature points z_i provide a natural coordinate representation for X/G. With these coordinates for the reduced space, the measurement equation (5.4) reduces to

$$z_i = \mathbf{Id}(z_i);$$

and the propagation equation (5.3) reduces to

$$\begin{aligned} \frac{d}{dt} z_i &= \frac{d}{dt} R(\theta)^T (p_i^f - p) \\ &= R(\theta)^T (\dot{p}_i^f - \dot{p}) + J^T R(\theta)^T \dot{\theta} (p_i^f - p) \\ &= R(\theta)^T (-R(\theta) v) + J^T R(\theta)^T w (p_i^f - p) \\ &= -v - wJz_i. \end{aligned}$$

Together with the robot position p and its orientation θ, we obtain a new coordinate representation of the state space, $(p, \theta, z_1, z_2 \ldots, z_N)$, and the control system with respect to this new coordinate system is given by

$$\begin{cases} \dot{p} = R(\theta)v, \\ \dot{\theta} = w, \\ \dot{z}_i = -v - wJz_i, \end{cases} \tag{5.15}$$

$$z_i = \mathbf{Id}(z_i). \tag{5.16}$$

The coordinate system given by $(p, \theta, z_1, z_2 \ldots, z_N)$ is global, and can be regarded as the state space being decomposed into the product of $x_N = (p, \theta)$ and $x_O = (z_1, z_2 \ldots, z_N)$, the former is unobservable part, and the latter is observable part, while the group action (5.8) now acts trivially on the x_O part. In this case, the unobservable subspace at each point can be represented as

$$\Omega^{\perp} = \mathscr{N} = \text{span} \begin{pmatrix} I_2 & Jp \\ \mathbf{0} & 1 \\ \mathbf{0} & 0 \\ \mathbf{0} & 0 \\ \vdots & \vdots \\ \mathbf{0} & 0 \end{pmatrix}. \tag{5.17}$$

We improved the linearity of the measurement equation (5.16) by transforming to the relative feature position representation, and as a consequence, the originally trivial propagation equation $\dot{p}_i^f = 0$ now becomes the nontrivial $\dot{z}_i = -v - wJz_i$.

We can construct a similar transition matrix for (5.15) as in (5.11), and apply forward

Euler integration to (5.15), which yields

$$\begin{cases} p_{k+1} = p_k + \Delta t \cdot R(\theta_k)(v_m^{k+1} + n_v), \\ \theta_{k+1} = \theta_k + \Delta t \cdot (w_m^{k+1} + n_w), \\ z_{i,k+1} = z_{i,k} + \Delta t \cdot (-(v_m^{k+1} + n_v) - (w_m^{k+1} + n_w) J z_{i,k}), \end{cases} \tag{5.18}$$

and the corresponding nominal value update step is given by

$$\begin{cases} p_{k+1} = p_k + R(\theta_k)\Delta d_{k+1}, \\ \theta_{k+1} = \theta_k + \Delta\theta_{k+1}, \\ z_{i,k+1} = (I - \Delta\theta_{k+1} J) z_{i,k} - \Delta d_{k+1}, \end{cases} \tag{5.19}$$

and the transition matrix is

$$\Phi_{k+1,k} = \begin{pmatrix} I_2 & J(p_{k+1} - p_k) & & & & & \\ \mathbf{0} & 1 & & & & & \\ & & I - \Delta\theta_{k+1} J & & & & \\ & & & I - \Delta\theta_{k+1} J & & & \\ & & & & \ddots & & \\ & & & & & & I - \Delta\theta_{k+1} J \end{pmatrix}. \tag{5.20}$$

The derivative of the reduced measurement (5.16) is

$$dh_i = \begin{pmatrix} \mathbf{0}_{2\times 2}, & \mathbf{0}_{2\times 1}, & \mathbf{0}_{2\times 2} & \dots & I_2 & \dots \end{pmatrix},$$

and we can see that $dh_i \cdot \mathcal{N} = 0$, which holds simply because the lower $2N \times 3$ part of $\mathcal{N}$ (5.17) is zero. This property still holds after repeated measurement updates

$$\Phi_{k+s,k+s-1} \cdot \ldots \Phi_{k+2,k+1} \cdot \Phi_{k+1,k} \cdot \mathcal{N}_k$$

as the bottom left parts of the transition matrices $\Phi_{k+j,k+j-1}$ vanish.

5.4 Bayesian inference

In Section 5.3, we discussed how to obtain the reduced control system that arises as a consequence of the invariant group action, which lead to the reduced propagation and measurement equations on the reduced quotient space. This motivated the use of a new coordinate representation $(p, \theta.z_1, z_2 \dots z_N)$ for the planar robot system. This decomposes the state space into the product of the unobservable part (p, θ), and the observable part $(z_1, z_2 \dots z_N)$. We also constructed a transition matrix which ensures that the unobservable subspace remains perpendicular to the measurement dh during propagation. But, as we noted, this propagation is "interrupted" at each step during the Kalman update. In order to preserve the unobservability property of the system during the measurement update, we propose a reduced Bayesian inference update(5.2) on the observable part using the reduced measurement. It is natural to apply Bayesian inference on $p(z_1, z_2 \dots z_N)$, and update $p(p, \theta, z_1, z_2 \dots z_N)$ using (5.2). However, before we do that, there are some basic things that we need to make clear. The first question is, what is Bayesian inference? As we all know, Bayesian inference is

$$p(x \mid y) = \frac{p(x) \cdot p(y \mid x)}{p(y)},$$

which arises naturally from the fact that

$$p(x,y) = p(x) \cdot p(y \mid x) = p(y) \cdot p(x \mid y). \tag{5.21}$$

where $p(x)$, $p(y)$, $p(x,y)$ are probability density functions. The next question is, what is a probability density function? As we all know, a probability distribution is a probability measure

on the state space, and the probability density function is a concrete way to represent it,

$$P(x \in A) = \int_A p(x)dx. \tag{5.22}$$

However, in (5.22) we need to specify a measure dx to perform the integration, and the representation of the probability measure as a probability density function depends on that choice of dx. Also, given a probability distribution on state space, when we try to find the most probable point, we find this problem also requires the introduction of a measure on the state space. As such, we need to specify a measure on the state space in order to define the probability density function or to find the most probable point. For a more detailed discussion of such issues, see [29]. After specifying the measure dx, the density function associated with a probability measure μ is just the Radon–Nikodym derivative of μ with respect to dx. A natural way to specify a measure on a smooth manifold is by specifying a Riemannian structure on it. Each Riemannian structure induces a Riemannian volume form, which in turn induces a measure. When we write down the common Gaussian density function, we are actually assuming the standard Riemannian structure on $\mathbb{R}^n$ implicitly. This structure is quite natural, as it is homogeneous and isotropic and does not introduce prior information on the space. More precisely, this structure is invariant under Euclidean transformations.

With the observation that each density function is defined with respect to an underlying base measure, we look at (5.21) again, and find that there is a product measure implicitly defined on (x, y), and that (5.21) is essentially an application of Fubini's theorem.

Now consider the new coordinate system $(p, \theta, z_1, z_2 \ldots z_N)$ we get for the planar robot system, which is diffeomorphic to the original coordinate system by the following transformations

$$(p,\theta,p_1^f,p_2^f\dots p_N^f) \leftrightarrow (p,\theta,z_1,z_2\dots z_N): \begin{cases} p=p, \\ \theta=\theta, \\ p_i^f = p+R(\theta)z_i; \end{cases} \qquad \begin{cases} p=p, \\ \theta=\theta, \\ z_i = R(\theta)^T(p_i^f-p). \end{cases} \tag{5.23}$$

Given the decomposition $p(x_N,x_O) = p(x_O)\cdot p(x_N \mid x_O)$, where $x_N=(p,\theta)$, $x_O=(z_1,z_2,\dots z_N)$, which Riemannian structure should we assign to the state space to obtain a base measure? For the old coordinate system $(p,\theta,p_1^f,p_2^f\dots p_N^f)$, it is natural to use the standard Riemannian structure $dp\otimes dp + d\theta\otimes d\theta + \sum_{i=1}^N dp_i^f\otimes dp_i^f$, as it is invariant under the group action (5.8). Moreover, if a Lie group G acts freely and properly on a Riemannian manifold M, and the action is isometric for $\forall g\in G$, then there is a natural Riemannian structure on M/G induced from M. For our quotient space $(z_1,z_2\dots z_N)$, it can be verified that the induced Riemannian structure is simply the standard Riemannian structure of $\mathbb{R}^{2N}$. However, the Riemannian structure of $(p,\theta,z_1,z_2\dots z_N)$ induced by the diffeomorphism (5.23) is not simply the product of the standard Riemannian structure of (p,θ) with the standard Riemannian structure of $(z_1,z_2\dots z_N)$. This can be verified by checking the Jacobian of (5.23), which is not a unitary matrix. We now have to decide which Riemannian structure to use on $(p,\theta,z_1,z_2\dots z_N)$, either choose the non-product structure induced from (5.23), or choose the product structure. We choose the product structure here, one reason is that it is simpler to apply the reduced Bayesian inference step (5.2), another reason is this product structure is also invariant under the group action (5.8), which ensures that $(z_1,z_2\dots z_N)$ remains fixed.

Finally, we are in a position to discuss how to implement (5.2) for the Gaussian distribution with respect to the standard Riemannian structure. Consider $p(x_N,x_O)$ with a Gaussian

distribution given by

$$p(x_N, x_O) = \mathcal{N}\left(\begin{pmatrix}\mu_N \\ \mu_O\end{pmatrix}, \begin{pmatrix}\Sigma_{NN} & \Sigma_{NO} \\ \Sigma_{ON} & \Sigma_{OO}\end{pmatrix}\right).$$

Then, by the property of conditional Gaussian distributions,

$$\begin{aligned} p(x_N, x_O) &= p(x_O) \cdot p(x_N \mid x_O) \\ &= \mathcal{N}(x_O \mid \mu_O, \Sigma_{OO}) \cdot \mathcal{N}(x_N \mid \mu_{N|O}, \Sigma_{N|O}), \\ &= \mathcal{N}(x_O \mid \mu_O, \Sigma_{OO}) \cdot \mathcal{N}(x_N \mid \mu_N + \Sigma_{NO}\Sigma_{OO}^{-1}(x_O - \mu_O), \Sigma_{NN} - \Sigma_{NO}\Sigma_{OO}^{-1}\Sigma_{ON}). \end{aligned}$$

The next step is to obtain an improved $\overline{p}(x_O) = \mathcal{N}(\overline{\mu}_O, \overline{\Sigma}_{OO})$ by taking a reduced measurement and computing the improved joint distribution $\overline{p}(x_N, x_O)$. In order to compute the joint distribution from the product of two Gaussian distributions efficiently, we choose the precision matrix, information vector representation of the Gaussian distribution,

$$\mathcal{N}(x_O \mid \mu_O, \Sigma_{OO}) = \mathcal{N}(x_O \mid \xi_O, \Omega_{OO}),$$

where $\Omega_{OO} = \Sigma_{OO}^{-1}$, $\xi_O = \Sigma_{OO}^{-1} \cdot \mu_O$, and $\Omega_{N|O} = \Sigma_{N|O}^{-1}$. Suppose that the reduced discrete measurement equation is given by

$$z = C \cdot x_O + \delta,$$

with measurement noise $\delta \sim \mathcal{N}(0, Q)$, where Q is the noise covariance, then by the information filter [58],

$$\begin{cases} \overline{\Omega}_{OO} = \Omega_{OO} + C^T Q^{-1} C, \\ \overline{\xi}_O = \xi_O + C^T Q^{-1} z. \end{cases}$$

Let $\mu_N + \Sigma_{NO}\Sigma_{OO}^{-1}(x_O - \mu_O) = Ax_O + b$ for notational simplicity, where $A = \Sigma_{NO}\Sigma_{OO}^{-1}$, $b =$

$\mu_N - \Sigma_{NO}\Sigma_{OO}^{-1}\mu_O$. We calculate the density function of $\overline{p}(x_N, x_O)$,

$$
\begin{aligned}
\log\overline{p}(x_N, x_O) &= \log\overline{p}(x_O) + \log p(x_N \mid x_O) \\
&= \log\mathcal{N}(x_O \mid \overline{\xi}_O, \overline{\Omega}_{OO}) + \log\mathcal{N}(x_N \mid A \cdot x_O + b, \Omega_{N|O}) \\
&\simeq -\frac{1}{2}x_O^T\overline{\Omega}_{OO}x_O + \overline{\xi}_O^T x_O - \frac{1}{2}(x_N - Ax_O - b)^T\Omega_{N|O}(x_N - Ax_O - b) \\
&\simeq -\frac{1}{2}x_O^T\overline{\Omega}_{OO}x_O + \overline{\xi}_O^T x_O - \frac{1}{2}x_N^T\Omega_{N|O}x_N - \frac{1}{2}x_O A^T\Omega_{N|O}Ax_O \\
&\qquad + x_O^T A^T\Omega_{N|O}x_N + x_N^T\Omega_{N|O}b - x_O^T A^T\Omega_{N|O}b \\
&= -\frac{1}{2}(x_N, x_O)^T \begin{pmatrix} \Omega_{N|O} & -\Omega_{N|O}A \\ -A^T\Omega_{N|O} & \overline{\Omega}_{OO} + A^T\Omega_{N|O}A \end{pmatrix} \begin{pmatrix} x_N \\ x_O \end{pmatrix} + (x_N, x_O)^T \begin{pmatrix} \Omega_{N|O}b \\ \overline{\xi}_O - A^T\Omega_{N|O}b \end{pmatrix}.
\end{aligned}
$$

Thus, the improved joint probability distribution $\overline{p}(x_N, x_O)$ has the following improved precision matrix and information vector,

$$
\overline{\Omega} = \begin{pmatrix} \Omega_{N|O} & -\Omega_{N|O}A \\ -A^T\Omega_{N|O} & \overline{\Omega}_{OO} + A^T\Omega_{N|O}A \end{pmatrix}, \qquad \overline{\xi} = \begin{pmatrix} \Omega_{N|O}b \\ \overline{\xi}_O - A^T\Omega_{N|O}b \end{pmatrix}. \tag{5.24}
$$

We can see from (5.24) that if originally $\Sigma_{NO} = 0$, then $\Sigma_{N|O} = \Sigma_{NN}$ and $A = \Sigma_{NO}\Sigma_{OO}^{-1} = 0$, thus $\overline{\Omega}$ is also block diagonal, and the improvement of $\overline{\Omega}_{OO}$ and $\overline{\xi}_O$ does not affect the distribution of x_N.

5.5 Numerical experiment

We performed numerical experiments for the planar robot system (5.3), (5.4), where the robot is assumed to be round with diameter 0.5 m, and binocular cameras are equipped on the left and right sides of the robot to detect feature points, each with a field angle of 120°. The odometry measures local velocity v (units: m/s) and angular velocity w (units: rad) with Gaussian noise $\mathcal{N}(\mathbf{0}, Q_v)$, $\mathcal{N}(0, Q_w)$, respectively. We set $Q_v = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$ and $Q_w = 0.01$ during the simulation. The camera is assumed to measure the relative direction of the feature point with

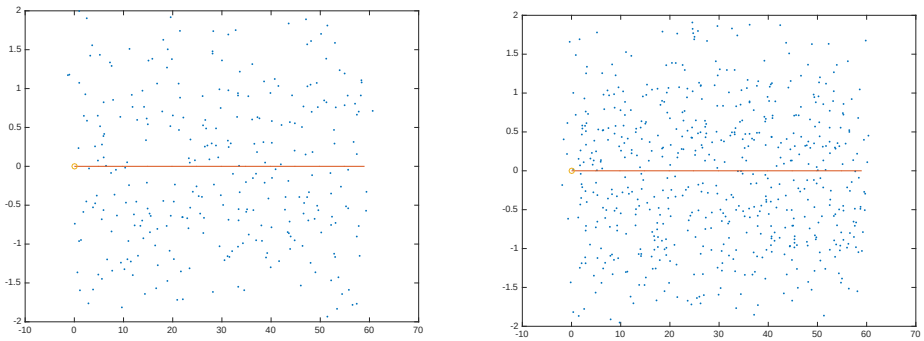
Gaussian noise $\mathcal{N}(0, Q_z)$. As we mentioned in Section 5.2, at each step, the measurement of feature points by the binocular cameras shall be combined together to give an estimate of the relative position of the feature points, and the uncertainty in the relative position is assumed to be described by an approximate Gaussian distribution. The odometry and camera readings are updated at the same frequency.

We tested three different methods, the first is the classical EKF on the state space $(p, \theta, p_1^f, p_2^f \dots p_N^f)$, with the propagation equation (5.9); the second is the First Estimate EKF (FE-EKF) where the First Estimate technique in [24] is used; the third is our reduced EKF on the state space $(p, \theta, z_1, z_2 \dots z_N)$, with propagation equation (5.18) and reduced measurement update equation (5.24). We observed superior performance of the reduced EKF compared to classical EKF and FE-EKF, especially at estimating the robot's orientation θ. Furthermore, the reduced EKF is less sensitive to measurement noise.

We considered three different trajectories: straight line, circle, and a general trajectory. Feature points are generated along the trajectory randomly with a given density. We assume that there are no errors introduced during feature detection and matching. We ignore feature points that are far away (distance $> 5\,\mathrm{m}$) from the robot, since triangulation for such points by binocular cameras are unstable.

5.5.1 Straight line

We tested on a straight line trajectory that is 60 m long, and the robot traveled along it with constant speed $1\,\mathrm{m/s}$. We also compared low and high densities of feature points.



(a) line, low density feature distribution **(b)** line, high density feature distribution

Figure 5.1. Straight line trajectory

In Figure 5.1, the red line is the trajectory, the blue points are features, and yellow circle is the robot. For the low density feature distribution, the average feature points recognized at each step is approximately 20, and for the high density feature distribution, the average recognized feature points at each step is approximately 40. Besides the feature distribution, we also considered update frequencies of 10 Hz and 20 Hz. Moreover, for fixed feature density and update frequency, we adjusted Q_z to observe the sensitivity of the different methods to changes in the measurement noise. Since the feature distribution is generated randomly for fixed feature density, and odometry and camera readings are also generated randomly from the given parameters, the results for a given method will vary between realizations. The three methods (EKF, FE-EKF, reduced EKF) are applied to the same data set of feature distribution and sensor readings, and we generate 20 realizations per combination of update frequency, feature distribution, and measurement noise. We summarize the average position error (units: meters) and the orientation error (units: rad) in Table 5.1 and Table 5.2.

We see that the accuracy of all three methods improve a little bit when we double the update frequency and increase the average feature number per step, which is quite natural. In addition, when we increase the uncertainty of measurement Q_z, the performance of all three methods degrade. There is no evident improvement in accuracy for both FE-EKF and reduced EKF methods over the classical EKF method for the straight line case. A typical error growth in

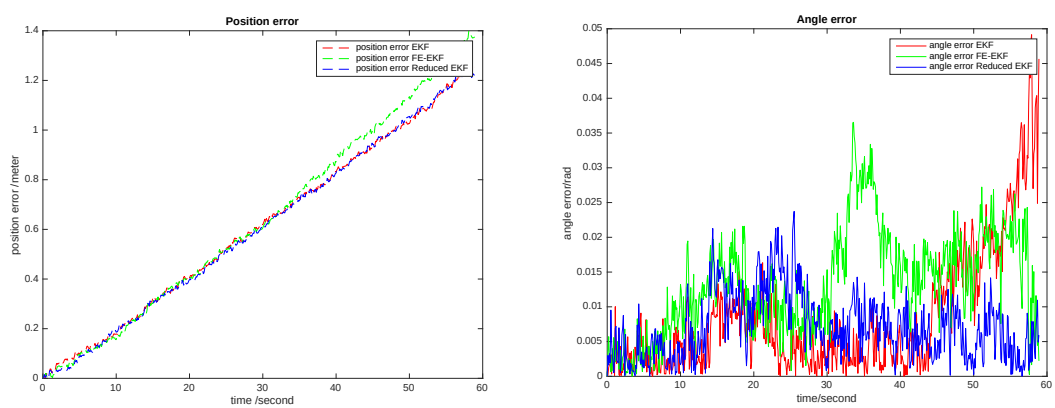
Table 5.1. Average error, straight line trajectory, average feature num = 20

	Straight line trajectory, average feature num = 20											
Frequency	update frequency = 10Hz						update frequency = 20Hz					
Method	EKF		FE-EKF		Reduced EKF		EKF		FE-EKF		Reduced EKF	
Error	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$
$Q_z = 1\times10^{-4}$	1.403	0.0375	1.539	0.0421	1.436	0.0373	1.365	0.0308	1.456	0.0345	1.300	0.0282
$Q_z = 2\times10^{-4}$	2.492	0.0487	2.525	0.0552	2.549	0.0506	2.395	0.0411	2.621	0.057	2.397	0.0432
$Q_z = 4\times10^{-4}$	4.332	0.0752	4.284	0.0758	4.449	0.0773	4.265	0.0578	4.283	0.0677	4.317	0.0644

Table 5.2. Average error, straight line trajectory, average feature num = 40

	Straight line trajectory, average feature num = 40											
Frequency	update frequency = 10Hz						update frequency = 20Hz					
Method	EKF		FE-EKF		Reduced EKF		EKF		FE-EKF		Reduced EKF	
Error	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$
$Q_z = 1\times10^{-4}$	1.303	0.0269	1.306	0.027	1.256	0.0227	1.21	0.0167	1.283	0.0238	1.258	0.0208
$Q_z = 2\times10^{-4}$	2.409	0.0385	2.459	0.0446	2.43	0.0347	2.331	0.0279	2.318	0.0313	2.386	0.0345
$Q_z = 4\times10^{-4}$	4.238	0.047	4.361	0.0628	4.386	0.0613	4.23	0.0445	4.353	0.0558	4.352	0.0577

the position and orientation for the straight line trajectory is given in Figure 5.2.



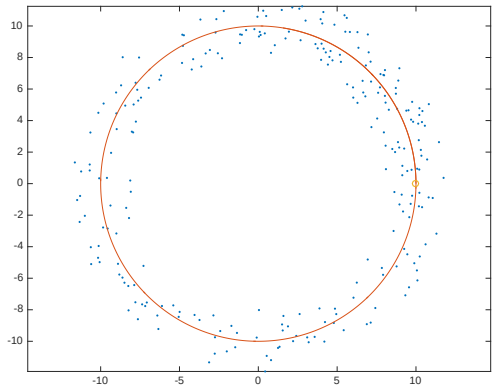
(a) Position error, straight line

(b) Angle error, straight line

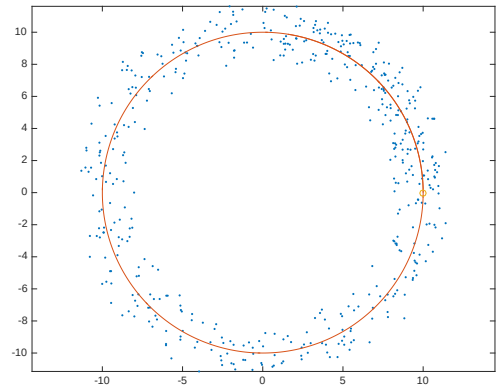
Figure 5.2. Error growth, straight line trajectory

5.5.2 Circle

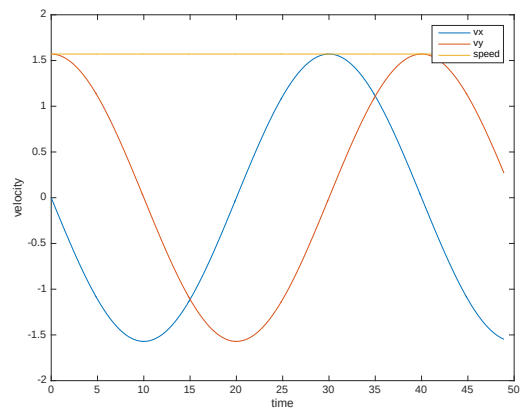
The robot now travels along a circle of radius 10 m, with constant speed that is approximately 1.57 m/s. This is illustrated in Figure 5.3. In contrast to the straight line case, we observed superior performance of the reduced EKF method over both the classical EKF and FE-EKF methods.



(a) Circle, low density feature distribution



(b) Circle, high density feature distribution



(c) Velocity of circle trajectory

Figure 5.3. Circle trajectory and velocity

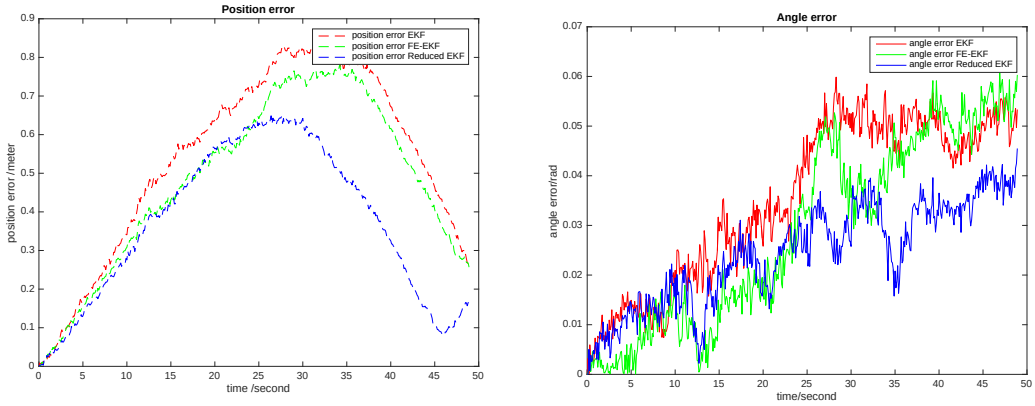
For the circle trajectory with low feature density, the average feature number per step is approximately 20, and for the high feature density, the average feature number per step is approximately 40. The average position error and orientation error of all three methods are summarized in Table 5.3 and Table 5.4. The improvement of reduced EKF method is quite significant compared to the classical EKF and FE-EKF methods. It can reduce the orientation error by approximately 50% in each case, and as a consequence, the estimation of position is also improved, and the position error is reduced by approximately 30% to 50%. A typical error growth for the circle trajectory is given in Figure 5.4.

Table 5.3. Average error, circle trajectory, average feature num = 20

	Average error, circle trajectory, average feature num = 20											
Frequency	update frequency = 10Hz						update frequency = 20Hz					
Method	EKF		FE-EKF		Reduced EKF		EKF		FE-EKF		Reduced EKF	
Error	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$
$Q_z = 1\times10^{-4}$	0.955	0.0875	1.063	0.0936	0.655	0.0443	0.746	0.0657	0.750	0.0576	0.532	0.0342
$Q_z = 2\times10^{-4}$	1.630	0.152	1.496	0.119	0.959	0.0586	1.35	0.128	1.549	0.144	0.982	0.0667
$Q_z = 4\times10^{-4}$	2.893	0.2914	2.721	0.2585	1.512	0.1033	2.295	0.2256	2.066	0.2102	1.469	0.1003

Table 5.4. Average error, circle trajectory, average feature num = 40

	Average error, circle trajectory, average feature num = 40											
Frequency	update frequency = 10Hz						update frequency = 20Hz					
Method	EKF		FE-EKF		Reduced EKF		EKF		FE-EKF		Reduced EKF	
Error	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$
$Q_z = 1\times10^{-4}$	0.816	0.0643	0.7972	0.0627	0.5815	0.0359	0.584	0.0462	0.681	0.0530	0.568	0.0374
$Q_z = 2\times10^{-4}$	1.276	0.1193	1.3061	0.1091	0.9403	0.0591	1.260	0.1164	1.2402	0.1039	0.9178	0.0573
$Q_z = 4\times10^{-4}$	2.593	0.253	2.428	0.236	1.752	0.121	2.4	0.239	2.397	0.216	1.631	0.113



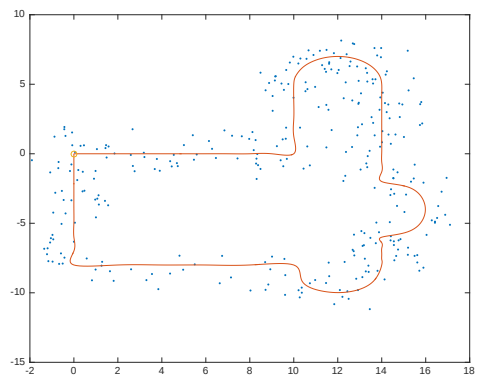
(a) Position error, circle trajectory

(b) Angle error, circle trajectory

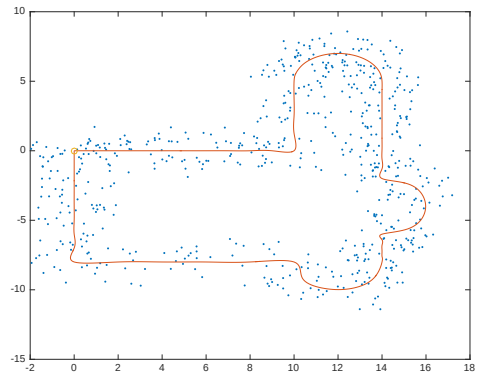
Figure 5.4. Error growth, circle trajectory

5.5.3 General trajectory

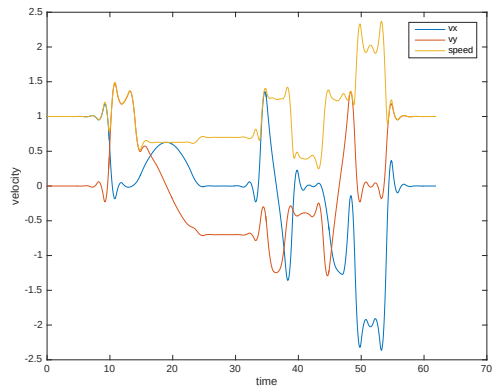
Besides the straight line and circle trajectories, we also consider a general trajectory, see Figure 5.5. The average feature number for the low feature density per step is approximately 25, and the average feature number for the high feature density per step is approximately 50.



(a) General, low density feature distribution



(b) General, high density feature distribution



(c) Velocity of general trajectory

Figure 5.5. General trajectory and velocity

As before, we tested all three methods on this general trajectory, and summarized their average position error and orientation error in Table 5.5 and Table 5.6. When the update frequency is 20 Hz, we observe noticeable improvement in both the position error and orientation error by the reduced EKF method. The FE-EKF method exhibits better performance than the classical EKF, but the reduced EKF method works even better. When the update frequence is 10 Hz, and the measurement noise $Q_z = 1 \times 10^{-4}$, we see that the error of the reduced EKF is actually larger, which is not observed in the circle case. This might possibly be due to the high velocity of the general trajectory, since we construct the trajectory by cubic spline interpolation, and the velocity is very high around corners, which is unrealistic in practice as the actuation bounds in a robot result in bounded velocities as well.

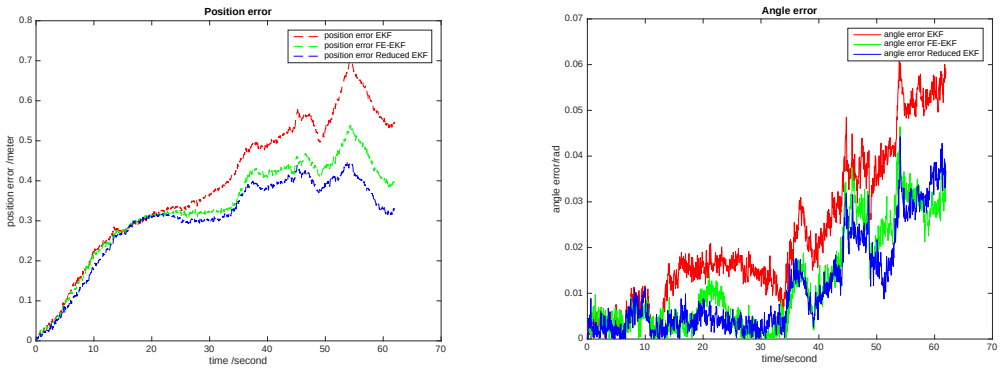
Table 5.5. Average error, general trajectory, average feature num = 25

	Average error, general trajectory, average feature num = 25											
Frequency	update frequency = 10Hz						update frequency = 20Hz					
Method	EKF		FE-EKF		Reduced EKF		EKF		FE-EKF		Reduced EKF	
Error	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$
$Q_z = 1\times10^{-4}$	0.673	0.0789	0.635	0.0803	1.309	0.1062	0.6527	0.0698	0.5620	0.0605	0.4659	0.0371
$Q_z = 2\times10^{-4}$	1.388	0.139	1.143	0.12	1.23	0.107	1.2575	0.1156	0.9985	0.0997	0.8022	0.0654
$Q_z = 4\times10^{-4}$	2.459	0.227	2.111	0.2267	1.592	0.1174	2.3852	0.2039	1.8924	0.1794	1.4595	0.1042

Thus, the propagation error is large when the update frequency is low, and it seems that when the discrete system deviates far away from the true trajectory, forcing it to obey the observability property of the underlying continuous system actually degrades the performance of the algorithm. This is perhaps understandable, as when the timestep is too large to accurately integrate the trajectory, the unobservable subspace becomes poorly approximated as well. This is easily resolved by doubling the update frequency. Also, we see that even when the update frequency is 10 Hz, when we increase the measurement noise to 4×10^{-4}, the reduced EKF method maintains a rather robust orientation error, and its position error is now smaller than the classical EKF and FE-EKF methods, which demonstrates the stability of the reduced EKF method to large measurement noise. This can be understood in terms of the relative error from measurement noise compared to the propagation step. When the measurement noise is larger, the spurious information introduced to the unobservable variables will be significant, which degrades the estimation accuracy. We believe that the reduced EKF method that respects the observability constraint is best suitable for applications where the propagation step is more precise than the measurement step. This is the case in typical VIO systems, where the IMU for propagation has very high update frequency (200–500 Hz), and is very accurate for short time estimation, and the camera for measurement has low update frequency (10–30 Hz). A typical error growth in the general trajectory case is given in Figure 5.6.

Table 5.6. Average error, general trajectory, average feature num = 50

	Average error, general trajectory, average feature num = 50											
Frequency	update frequency = 10Hz						update frequency = 20Hz					
Method	EKF		FE-EKF		Reduced EKF		EKF		FE-EKF		Reduced EKF	
Error	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$	δp	$\delta\theta$
$Q_z = 1\times10^{-4}$	0.6039	0.0609	0.5815	0.0636	1.308	0.1049	0.5518	0.0518	0.4810	0.0447	0.4538	0.0381
$Q_z = 2\times10^{-4}$	1.2201	0.1047	0.9836	0.0945	1.0738	0.0838	1.1676	0.1021	0.9752	0.0869	0.73	0.0611
$Q_z = 4\times10^{-4}$	2.4264	0.2126	2.0421	0.1904	1.4674	0.1121	2.447	0.2141	1.8347	0.145	1.4322	0.1165



(a) Position error, general trajectory **(b)** Angle error, general trajectory

Figure 5.6. Error growth, general trajectory

Chapter 5, in full, is a reprint of the material that has been submitted for publication to International Journal of Robotics Research, 2019. Xuefeng Shen; Melvin Leok, SAGE Publishing , 2019. The dissertation author was the primary investigator and author of this material.

Chapter 6

Conclusions and Future Directions

Geometric integrators are developed that are applicable to stiff systems, Lie groups and control systems. For stiff systems resulting from semi-discretizing Hamiltonian partial differential equations, we designed the midpoint exponential rule that preserves the Poisson structure, and the high-order diagonally implicit symplectic exponential integrator; we also constructed energy-preserving exponential integrators using a discrete gradient method. In general, exponential integrators allow the use of larger timesteps, and are therefore more computationally economical. In addition, geometric exponential integrators also inherit the long time stability properties of geometric integrators. The energy preserving exponential integrator is only second-order accurate, it would be interesting to construct higher-order energy preserving exponential integrators, in particular, by combining methods such as the Hamiltonian Boundary Value Method (HBVM) with exponential integrators.

On the Lie group side, for the rigid body problem, we recognized that the unit quaternions have a Lie group structure, and we constructed a Lie group variational integrator on the space of unit quaternions based on this observation. This technique can be naturally extended to the full body problem on $SE(3)$. On the n-dimensional rotation group $SO(n)$, we constructed high-order symplectic Lie group methods using the polar decomposition, and these methods can be naturally implemented as fixed-point iterations, which yield much more efficient methods than Jacobian based root finding methods when n is large. Also, for Hamiltonian systems with

symmetry where Lie–Poisson reduction applies, this method can be reduced to a Lie–Poisson integrator that preserves the associated Lie–Poisson structure. An interesting application could be to optimization on $SO(n)$, which appears in the solution of the cocktail party problem by independent component analysis [26]. In [57], the relationship between Nesterov's accelerated optimization method and a second-order damped dynamical system is described, while [6] suggests applying a symplectic integrator to a time-dependent Lagrangian formulation of this second-order damped dynamical system as a means of constructing accelerated optimization methods. The resulting problem is a Lagrangian system on $SO(n)$, and we will explore the possibility of applying Lie group variational integrators to this problem in the future.

For control systems, we studied in depth the unobservability properties of the planar robot equipped with odometry sensors and cameras, with an emphasis on the group-invariance of the control system. Due to the Lie group symmetry, the original system can be reduced to the observable part, and the state variable can be decomposed into (x_N, x_O) explicitly. Since the Lie group action on x_N is trivial, the problem of requiring that the propagation step of the Kalman filter satisfies the geometric property that the unobservable subspace remains perpendicular to the measurement dh, becomes quite natural in the reduced representation. Moreover, in order to deal with the problem that this property is destroyed at the measurement update step of the Kalman filter, we propose a reduced Bayesian inference method, i.e., only the observable part x_O is updated by the reduced measurement. This special procedure guarantees that the unobservable variable x_N remains unobservable, and no spurious information is introduced by the measurement update. This reduced Kalman filter method based on observability considerations outperforms the traditional Kalman filter as well as the FEJ-Kalman filter by quite a lot. In the future, we intend to apply this technique to a practical VIO system, where the geometry involved is more complicated. In particular, the rotation group in $\mathbb{R}^3$ is $SO(3)$ which is nonabelian, and only rotations about the gravity direction are unobservable. Also, we would like to update the MSCKF framework, where the state variables include historic poses of the camera, which makes the decomposition into observable and unobservable components more interesting. In principle, all

traditional techniques that are applicable to the Kalman filter, such as UKF, particle filters, and so on, could all be combined with our reduced update method based on the observability property, and we would like to test the efficiency of our method for all related practical applications.

Bibliography

[1] M. J. Ablowitz and J. F. Ladik. Nonlinear differential–difference equations and Fourier analysis. *Journal of Mathematical Physics*, 17(6):1011–1018, 1976.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[3] K. Atkinson and W. Han. *Theoretical numerical analysis*, volume 39 of *Texts in Applied Mathematics*. Springer, Dordrecht, third edition, 2009. A functional analysis framework.

[4] T. D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[5] R. H. Bartels and G. W. Stewart. Solution of the matrix equation $ax + xb = c$ [f4]. *Communications of the ACM*, 15(9):820–826, 1972.

[6] M. Betancourt, M. I. Jordan, and A. C. Wilson. On Symplectic Optimization. *arXiv e-prints*, art. arXiv:1802.03653, February 2018.

[7] G. Bogfjellmo and H. Marthinsen. High-order symplectic partitioned Lie group methods. *Foundations of Computational Mathematics*, 16(2):493–530, 2016.

[8] N. Bou-Rabee and J. E. Marsden. Hamilton–Pontryagin integrators on Lie groups Part I: Introduction and structure-preserving properties. *Foundations of Computational Mathematics*, 9(2):197–219, 2009.

[9] N. Bou-Rabee and H. Owhadi. Stochastic variational integrators. *IMA Journal of Numerical Analysis*, 29(2):421–443, 2009.

[10] C.-T. Chen. *Linear system theory and design*. Oxford University Press, 1998.

[11] D. Cohen, L. Gauckler, E. Hairer, and C. Lubich. Long-term analysis of numerical integrators for oscillatory Hamiltonian systems under minimal non-resonance conditions. *BIT Numerical Mathematics*, 55(3):705–732, 2015.

[12] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single

camera SLAM. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6): 1052–1067, 2007.

[13] L. Gauckler. Numerical long-time energy conservation for the nonlinear Schrödinger equation. *IMA Journal of Numerical Analysis*, 37(4):2067–2090, 2017.

[14] S. Geng. Symplectic partitioned Runge–Kutta methods. *Journal of Computational Mathematics*, pages 365–372, 1993.

[15] R. Goldman. *An Integrated Introduction to Computer Graphics and Geometric Modeling*. CRC Press, Hoboken, NJ, 2009.

[16] G. Golub, S. Nash, and C. Van Loan. A Hessenberg–Schur method for the problem $ax + xb = c$. *IEEE Transactions on Automatic Control*, 24(6):909–913, 1979.

[17] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[18] J. Hall and M. Leok. Spectral variational integrators. *Numerische Mathematik*, 130(4): 681–740, 2015.

[19] J. Hall and M. Leok. Lie group spectral variational integrators. *Foundations of Computational Mathematics*, 17(1):199–257, 2017.

[20] R. Hermann and A. Krener. Nonlinear controllability and observability. *IEEE Transactions on automatic control*, 22(5):728–740, 1977.

[21] M. Hochbruck and A. Ostermann. Exponential Runge–Kutta methods for parabolic problems. *Appl. Numer. Math.*, 53(2-4):323–339, 2005.

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

[23] D. D. Holm. *Geometric mechanics. Part II. Rotating, translating and rolling*. Imperial College Press, London, second edition, 2011.

[24] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Analysis and improvement of the consistency of extended Kalman filter based SLAM. In *2008 IEEE International Conference on Robotics and Automation*, pages 473–479. IEEE, 2008.

[25] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Observability-based rules for designing consistent EKF SLAM estimators. *The International Journal of Robotics Research*, 29(5): 502–528, 2010.

[26] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[27] S. Pekarsky J. E. Marsden and S. Shkoller. Discrete Euler–Poincaré and Lie–Poisson equation. *Nonlinearity*, 12:1647–1662, 1999.

[28] L. Jay. Symplectic partitioned Runge–Kutta methods for constrained Hamiltonian systems. *SIAM Journal on Numerical Analysis*, 33(1):368–387, 1996.

[29] I. H. Jermyn. Invariant Bayesian estimation on manifolds. *The Annals of Statistics*, 33(2): 583–605, 2005.

[30] S. J. Julier and J. K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 4238–4243. IEEE, 2001.

[31] O. Junge, J. E. Marsden, and S. Ober-Blöbaum. Discrete mechanics and optimal control. In *Proceedings of the 16th IFAC World Congress*, volume 16, pages 00310–1, 2005.

[32] Z. Kalogiratou and Th. Monovasilis. Diagonally implicit symplectic Runge–Kutta methods with special properties. *Appl. Math. Inf. Sci.*, 9(1L):11–17, 2015.

[33] J. B. Kuipers. *Quaternions and rotation sequences*. Princeton University Press, Princeton, NJ, 1999. A primer with applications to orbits, aerospace, and virtual reality.

[34] J. M. Lee. Introduction to smooth manifolds, 2001.

[35] T. Lee, N. H. McClamroch, and M. Leok. A Lie group variational integrator for the attitude dynamics of a rigid body with applications to the 3D pendulum. *Proc. IEEE Conf. on Control Applications*, pages 962–967, 2005.

[36] T. Lee, M. Leok, and N. H. McClamroch. Lie group variational integrators for the full body problem. *Comput. Methods Appl. Mech. Engrg.*, 196(29-30):2907–2924, 2007.

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

[38] M. Leok. Generalized Galerkin variational integrators. *arXiv:math/0508360*, 2005.

[39] M. Leok and T. Shingel. General techniques for constructing variational integrators. *Frontiers of Mathematics in China*, 7:273–303, 2012.

[40] S. Leyendecker, S. Ober-Blöbaum, J. E. Marsden, and M. Ortiz. Discrete mechanics and

optimal control for constrained systems. *Optimal Control Applications and Methods*, 31 (6):505–528, 2010.

[41] M. Li and A. I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[42] F. L. Markley. Unit quaternion from rotation matrix. *Journal of Guidance, Control, and Dynamics*, 31(2):440–442, 2008.

[43] J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numer.*, 10:357–514, 2001.

[44] J. E. Marsden, G. W. Patrick, and S. Shkoller. Multisymplectic geometry, variational integrators, and nonlinear pdes. *Communications in Mathematical Physics*, 199(2):351–395, 1998.

[45] B. Minchev. *Exponential integrators for semilinear problems*. PhD thesis, University of Bergen, 2004.

[46] J. Montaldi and T. Ratiu. Geometric mechanics and symmetry. 2005.

[47] J. M. M. Montiel, J. Civera, and A. J.s Davison. Unified inverse depth parametrization for monocular SLAM. Robotics: Science and Systems, 2006.

[48] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Robotics and automation, 2007 IEEE international conference on*, pages 3565–3572. IEEE, 2007.

[49] G. R. W. Quispel and G. S. Turner. Discrete gradient methods for solving ODEs numerically while preserving a first integral. *J. Phys. A*, 29(13):L341–L349, 1996.

[50] S. Reich. Symplectic integration of constrained Hamiltonian systems by Runge–Kutta methods. Technical report, University of British Columbia, Vancouver, BC, Canada, Canada, 1993.

[51] J.-P. Ryckaert, G. Ciccotti, and H. J.C. Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of Computational Physics*, 23(3):327 – 341, 1977.

[52] J. M. Sanz-Serna. Runge–Kutta schemes for Hamiltonian systems. *BIT Numerical Mathematics*, 28(4):877–883, 1988.

[53] J. M. Sanz-Serna. An unconventional symplectic integrator of W. Kahan. *Applied numerical mathematics*, 16(1-2):245–250, 1994.

[54] M. Shuster. Approximate algorithms for fast optimal attitude computation. AIAA Guidance and Control Conference, paper no. 78-1249, 1978.

[55] M. Shuster. The quest for better attitudes. *The Journal of the Astronautical Sciences*, 54(3): 657–683, 2006.

[56] J. C. Simo and O. Gonzalez. Assessment of energy-momentum and symplectic schemes for stiff dynamical systems. *Proc. ASME Winter Annual Meeting, New Orleans, Louisiana.*, 1993.

[57] W. Su, S. Boyd, and E. Candes. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.

[58] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

[59] L. N. Trefethen. *Spectral methods in MATLAB*, volume 10 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[60] V. S. Varadarajan. *Lie groups, Lie algebras, and their representations*, volume 102 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1984. Reprint of the 1974 edition.

[61] R. Vepa. *Nonlinear Control of Robots and Unmanned Aerial Vehicles: An Integrated Approach*. CRC Press, Hoboken, NJ, 2016.

[62] J. Vince. *Quaternions for Computer Graphics*. Springer, 2011.

[63] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[64] J. M. Wendlandt and J. E. Marsden. Mechanical integrators derived from a discrete variational principle. *Physica D*, 106:223–246, 1997.

[65] X. Wu, K. Liu, and W. Shi. An extended discrete gradient formula for multi-frequency oscillatory Hamiltonian systems. In *Structure-Preserving Algorithms for Oscillatory Differential Equations II*, pages 95–115. Springer, 2015.

[66] Y. Yang. Spacecraft attitude determination and control: Quaternion based method. *Annual Reviews in Control*, 36(2):198–219, 2012.

[67] W. J. Zhu and M. Z. Qin. Poisson schemes for Hamiltonian systems on Poisson manifolds. *Comput. Math. Appl.*, 27(12):7–16, 1994.