OVERDAMPED LANGEVIN DYNAMICS ON LIE GROUPS

XINCHANG WANG
(xinchang@umich.edu)
AUG, 2022, ANN ARBOR

Abstract. In recent years, the diffusion process plays a significant role in Machine Learning, and introduces broad applications in areas such as distribution sampling and non-convex optimization. In this report, we focus on degenerate overdamped Langevin dynamics on 3-dimensional matrix Lie groups such as SE(2) and SO(3), which have some nice properties. Under some assumptions on the degenerate overdamped Langevin dynamics, it has been shown that for some feasible function V, the distribution corresponding to the dynamics will converge to a stable distribution $\pi \propto \exp(-V)$. We simulate the degenerate overdamped Langevin dynamics on Lie groups and compare it with other diffusion processes, which implies possible applications in Computer Vision, Machine Learning, Sample Generation, etc.

Contents

1. Introduction

1.1. Langevin Dynamics.

Langevin dynamics is an approach developed by physicist *Paul Langevin*, used to model the dynamics of molecular systems. In recent years, Langevin dynamics has been extensively applied in the fields of optimization theory[10][36], machine learning[12][30][34] and related fields [9]. Below gives the motivation to use Langevin dynamics in non-convex optimization.

Given an unconstrained optimization problem

$$\text{minimize } f(x). \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and twice differentiable. Assume (1.1) is solvable, which means there exists an optimal point x^* such that $\inf_x f(x) = f(x^*)$, and by convexity x^* must be unique [8]. In order to find the optimal point x^*, a basic approach is the **Gradient descent method (GD)**.

Algorithm 1 Gradient descent method

given $x_0 \in \text{dom } f$
set small $\eta > 0$
while $\|\nabla f(x_j)\|_2 > \eta$ **do**
  $\Delta x_j = -\nabla f(x_j)$
  Choose step size s.
  $x_{j+1} = x_j + s\Delta x_j$
end while
return $x \approx x^*$

If in GD we set $s = \Delta t$, where Δt is a small constant, then the algorithm is a discrete-time simulation for the ordinary differential equation $dX_t = -\nabla f(X_t)dt$ with the initial condition $X_0 \in \text{dom } f$ and constraint $t \geq 0$.

One key limitation of the GD is that it performs unsatisfactory in non-convex optimization. Still consider the solvable unconstrained optimization problem (1.1), but this time assume f is non-convex. In this case, the direction of $-\nabla f$ is not

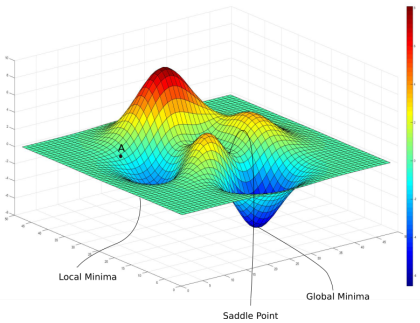


Figure 1. This figure illustrate what *local minimum*, *global minimum* and *saddle point* look like in a non-convex optimization. [Source: Ayoosh, 2018 [23]]

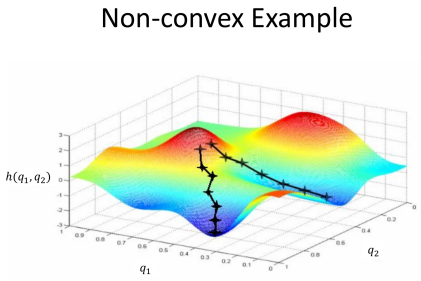


Figure 2. This figure shows how start point of **GD** influence its result. One ends up in the global minima while the other in a local minima.

guaranteed to point to the *global minimum* but only a *local minimum* or even a *saddle point*, and a *local minimum* is not necessarily a *global minimum* for non-convex function. Therefore, the performance of the GD is totally determined by the chosen initial point, and in complicated cases it might be almost impossible to find an initial point such that the GD could reach the *global minimum*.

To deal with this limitation, a method called **Stochastic Gradient Langevin Dynamics (SGLD)** was introduced [10][21], which is an adaption of Langevin dynamics (a type of diffusion process, see Appendix B.2). Instead of taking deterministic descent steps, the **SGLD** adds a random noise on each descent step to help jump out from a *local minimum* or *saddle point*.

Algorithm 2 Stochastic Gradient Langevin Dynamics (SGLD)

given $x_0 \in \text{dom } f$
while stopping criteria not satisfied **do**
  $\Delta x_j = -\nabla f(x_j)$
  Choose step size s.
  Choose noise size S
  $x_{j+1} = x_j + s\Delta x_j + \sqrt{2S}\eta$, $\eta \sim \mathcal{N}(0,1)$
end while
return $x \approx x^*$

If in SGLD we set $s = \Delta t$, $S = \Delta t \cdot M$, where Δt is a small constant and M is a constant, then the SGLD becomes a discrete-time simulation for the *overdamped Langevin dynamics*

$$dX_t = -\nabla f(X_t)dt + \sqrt{2M}dB_t.$$

This report puts special interests in other extended forms of *overdamped Langevin dynamics*, such as *degenerate overdamped Langevin dynamics* on *Lie groups*, and would summarize possible applications.

1.2. Matrix Lie groups.

Matrix Lie groups are important in many engineering fields, including robotic control [11], computer vision [32][33][37] and medical image processing [13][14][20]. Some specific Lie groups can be used to represent engineering structures.

For example, SE(2) group is the group of rigid transformations in the 2D plane, which is a geometric transformation of a Euclidean space that preserves the Euclidean distance between every pair of points [A.2.2]. In [13][14], linear and non-linear left-invariant diffusions on invertible orientation scores are discussed, which are used to deal with left-invariant parabolic evolutions on $SE(2)$ and contour enhancement, an important topic in medical image processing. In [11], the use of information theory on $SE(2)$ is presented, which implies application in mobile robotic control.

SO(3) group, or 3D rotation group, is the group of all rotations about the origin in the three-dimensional Euclidean space $\mathbb{R}^3$ under the operation of composition (or the matrix multiplication) [A.2.3].

For the parameterization of SO(3) group, one naive approach is to use *Euler angles* [C.2], treating the rotations as the composition of several elementary rotations

around the Cartesian axes, which is widely used in traditional engineering fields. Nevertheless, such a representation could ignore the symmetry property of the $SO(3)$ group, which is not desired in SO(3) estimation. To estimate the Gaussian distributions on $SO(3)$, *Fisher matrix* [28] and *Bringham distributions* [22] were introduced in deep learning as probabilistic rotation estimators for orthonormal matrix and quaternion representations of $SO(3)$, respectively. The main problem for the two methods is that they are not closed under convolution, which makes deep learning inefficient. To avoid this defect, [26] adopts the *isotropic Gaussian distribution* on $SO(3)$, denoted $\mathcal{IG}_{SO(3)}(\mu, \varepsilon^2)$, to improve *Denoising Diffusion Probabilistic Models on* $SO(3)$ *for Rotational Alignment*.

In this report we mainly focus on real *matrix Lie groups*, which feature some nice properties. Each real *matrix Lie group* is a closed subgroup of $GL(n, \mathbb{R})$, which means it can be represented by a subset of all invertible $n \times n$ squared matrix including the identity matrix, and closed under group inversion (matrix inverse) and group multiplication (matrix multiplication).

One of the most important property of Lie groups is its connection with Lie algebras, which also builds a connection between the *matrix Lie group* and the Euclidean vector space. Each Lie group induces a corresponding Lie algebra, which is the tangent vector space of the Lie group. This makes *matrix* Lie group parameterizable using coordinates in Euclidean vector space. For example, $SO(3)$ can be parameterized using a 3D vector in $\mathbb{R}^3$, and $SE(2)$ can be parameterized by a 3D vector in $S^1 \times \mathbb{R}^2$ [See appendix A].

1.3. **Overdamped Langevin Dynamics on Lie groups.**

The convergence analysis of *Langevin dynamics* is important in both theory and application [27]. This section gives several forms of *Langevin dynamics*, and discuss their convergence behaviors. For some forms of *Langevin dynamics*, one significant method is the *Gamma calculus* (or *Bakry- Émery iterative calculus*), which provides the *Ricci curvature lower bound* to study the convergence behavior [2].

1.3.1. *Overdamped Langevin dynamics on Euclidean space.*

An *overdamped Langevin dynamics* on *Euclidean space* is defined by:

$$dX_t = -\nabla V dt + \sqrt{2} dB_t, \tag{1.2}$$

where $X_t \in \mathbb{R}^n$, $V \in C^2(\mathbb{R}^n, \mathbb{R})$ and B_t an n-dimensional Brownian motion, with the corresponding *Fokker-Planck Equation*:

$$\begin{aligned} \frac{\partial p(x,t)}{\partial t} &= \sum_{i=1}^{n} \frac{\partial}{\partial x_i}[(\nabla V)_i(x) p(x,t)] + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j} p(x,t) \\ &= \nabla_x (p(x,t) \nabla V(x)) + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j} p(x,t). \end{aligned} \tag{1.3}$$

where $p(x,t)$ is a smooth probability density function of (1.2) with a smooth initial condition:

$$p(0,t) = p_0(t), \ \int p(x,t)dx = 1, \ p_0(t) \geq 0.$$

Assume further that (1.2) has a unique invariant distribution $\pi(x)$, solving the (1.3) such that

$$0 = \nabla_x(\pi(x)\nabla V(x)) + \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2}{\partial x_i \partial x_j}\pi(x). \tag{1.4}$$

The solution of above PDE gives $\pi \propto \exp(-V)$, and the convergence rate is exponentially fast under some *Lyapunov conditions* [4].

1.3.2. *Overdamped langevin dynamics on Lie groups.*

For a *matrix Lie group* $\mathcal{G}$ with degree of freedom $n+m$, the *Itô overdamped Langevin dynamics* on $\mathcal{G}$ is defined as:

$$dX_t = b(X_t)dt + \sqrt{2}a(X_t)dB_t, \tag{1.5}$$

where

$$b(X_t) = -a(X_t)a(X_t)^\intercal \nabla V + (\sum_{j=1}^{n+m}\frac{\partial}{\partial x_j}(a(X_t)a(X_t)^\intercal)_{ij})_{i=1}^{n+m}.$$

Remark. See B.3 for the conversion between *Itô SDE* and *Stratonovich SDE*. For convenience, all SDEs in this report are Itô SDE.

where $X_t \in \mathbb{R}^{n+m}$, $V \in C^2(\mathbb{R}^{n+m}, \mathbb{R})$, $a \in \mathbb{R}^{(n+m)\times n}$ which represents a collection of left invariant fields of $\mathcal{G}$, and B_t a n-dimensional *Brownian motion*. The corresponding *Fokker-Planck Equation* is

$$\frac{\partial}{\partial t}p(x,t) = -\nabla_x(p(x,t)b(x)) + \sum_{i=1}^{n+m}\sum_{j=1}^{n+m}\frac{\partial^2}{\partial x_i \partial x_j}((a(x)a(x)^\intercal)_{ij}p(x,t)), \tag{1.6}$$

where $p(x,t)$ is smooth with initial conditions similar as (1.3). Assume further the existence of unique smooth invariant distribution π solving the PDE, which gives $\pi \propto \exp(-V)$. One can rewrite (1.6) such that [16]

$$\partial_t p(x,t) = \nabla \cdot (p(x,t)a(x)a(x)^\intercal \nabla \log\frac{p(x,t)}{\pi(x)}). \tag{1.7}$$

If $m = 0$, then we call (1.5) the *non-degenerate overdamped Langevin dynamics* on *Lie groups* (For convenience, use *non-degenerate Langevin* in later sections). If $m > 0$, then we call (1.5) the *degenerate overdamped Langevin dynamics* on *Lie groups* (For convenience, use *degenerate Langevin* in later sections).

The convergence analysis of *non-degenerate Langevin* has been well studied using various approaches, including the Entropy method [1]. For the *degenerate Langevin*, [16] [18] analyzed the convergence behavior via a modified entropy dissipation method. Section Section 2 will show the basic ideas.

We are interested in *Langevin dynamics* on *Lie groups* as it might imply applications in many fields, which will be presented in Section Section 4. In fact, (1.2) can be seen as a *Langevin dynamics* with no additional control other than the drift term, while (1.5) follow some *Riemannian* (non-degenerate) or *sub-Riemannian* (degenerate) structural control, respectively. Section 3 will present the simulations of some examples, and compare the difference of results among the various forms of *Langevin dynamics*.

2. Main Theory

2.1. Entropy Dissipation.

2.1.1. *Toy model.*

This section presents the entropy dissipation of an 1-dimensional Langevin dynamics using *Gamma Calculus* method, which shows the exponentially fast convergence rate of the distribution under some assumptions.

Definition 2.1.
Consider an 1-dimensional Itô diffusion process

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t.$$

For any functions $h, g \in C^\infty(\mathbb{R})$, define the following operators [3]:

$$L(g) = b \cdot \partial_x g + \frac{1}{2}\sigma^2 \partial_{xx} g, \qquad (\textit{infinitesimal generator})$$

$$\Gamma_1(h,g) = \frac{1}{2}[L(hg) - h \cdot L(g) - g \cdot L(h)], \qquad (\textit{Carré du champ operator})$$

$$\Gamma_2(h,g) = \frac{1}{2}[L(\Gamma_1(h,g)) - \Gamma_1(h, L(g)) - \Gamma_1(g, L(h))].$$

For the 1-dimensional *overdamped Langevin dynamics* $dX_t = -\nabla V dt + \sqrt{2}dB_t$, let $h = g$, we have

$$\begin{aligned} L(g) &= -\nabla V \cdot \nabla g + \nabla^2 g, \\ \Gamma_1(g,g) &= (\nabla g)^2, \\ \Gamma_2(g,g) &= (\mathfrak{hess}\ g)^2 + \underbrace{(\mathfrak{hess}\ V) \cdot (\nabla g)^2}_{\text{Ricci Curvature } = \mathfrak{Ric}(\nabla g, \nabla g)}. \end{aligned}$$

This coincides with the *Bochner's formula* C.1. In fact, in the convergence analysis for the higher dimensional cases, C.1 is used in the proof.

Definition 2.2.
Let ρ be a distribution and π be the invariant distribution, define the *Kullback-Leibler divergence* (or *KL-divergence*) between ρ and π by

$$\mathcal{D}(\rho) = \int \rho \log \frac{\rho}{\pi} dx =: \mathcal{D}(\rho|\pi),$$

and define the *Fisher-information functional* by

$$\mathcal{I}(\rho) = \int \left\langle \nabla \log \frac{\rho}{\pi}, \nabla \log \frac{\rho}{\pi} \right\rangle \rho(x) dx.$$

Assumption 2.3 (Curvature dimension inequality (**CDI**) [3])**.**
Assume $\exists\ \lambda > 0$ s.t. $\Gamma_2(f,f) \geq \lambda \Gamma_1(f,f)$ for all smooth f.

Assume the CDI holds, then the **Entropy Dissipation** for 1-dimensional *overdamped Langevin dynamics*

$$dX_t = -\nabla V dt + \sqrt{2}dB_t,$$

follows

(1) $\frac{d}{dt}\mathcal{D}(\rho_t) = -\mathcal{I}(\rho_t)$;
(2) $\frac{d}{dt}\mathcal{I}(\rho_t) = -2\int \Gamma_2(\log \frac{\rho_t}{\pi}, \log \frac{\rho_t}{\pi})\rho_t dx$;

(3) $\mathcal{D}(\rho_t) \leq \frac{1}{2\lambda}\mathcal{I}(\rho_t)$; (*Logarithm-Sobolev Inequality*)
and the details are presented in C.1. Then the exponential decay result holds:

$$\mathcal{D}(\rho_t) \leq \frac{1}{2\lambda} e^{-2\lambda t}\mathcal{I}(\rho_0).$$

which illustrates the *KL-divergence* between ρ_t and π is bounded by a term exponentially decayed w.r.t. time t.

Remark. This result can be extended to more general *Langevin dynamics* on higher dimensional euclidean space with similar methods [17].

2.1.2. *Entropy dissipation for degenerate overdamped Langevin dynamics.*

For the *degenerate Langevin* (1.5), the classical *Gamma calculus method* is not valid [2], and [16][18][19] extended the improved *Gamma-z calculus method* for the *Lyapunov exponential convergence analysis*, which was first proposed in [6]. More assumptions are added for the exponential entropy dissipation in this degenerate case. Below summarizes two key assumptions, and for other assumptions check [16][18].

Assumption 2.4.
$\{a_1, \ldots, a_n\}$ in (1.5) satisfies the **weak Hörmander condition**:

$$Span\{a_1(x), \ldots, a_n(x), [a_{i_1}, \ldots, [a_{i_{k-1}}, a_{i_k}]\ldots] : 0 \leq i_1, \ldots, i_k \leq n, k \geq 2\} = \mathbb{R}^{n+m}$$

where $[\cdot, \cdot]$ is the Lie bracket operator (A.1).

Assumption 2.5 (generalized curvature dimension inequality (**GCDI**))**.**
Suppose $\mathfrak{R} \geq k(aa^\intercal + zz^\intercal)$ for $k > 0$. Denote the smooth initial distribution by ρ_0, where $\mathcal{R}$ is the *Ricci matrix* defined in [16], and $z \in \mathbb{R}^{(n+m)\times m}$ is designed to satisfy other assumptions.

Remark. An intuitive understanding of Assumption 2.4 is that with limited directions on a sub-Riemannian manifold, one can generate other directions on the manifold via *Lie bracket* operations. In other words, the vector fields $\{b, a_1, \ldots, a_n\}$ can generate full rank Lie algebras for any $x \in \mathbb{R}^{n+m}$. For GCDI, the matrix z can be viewed as a complement of the matrix a in (1.5), such that the combination of a and z form a basis for the whole manifold, and the inequality is again a guarantee for the *generalized Logarithm-Sobolev inequality.*

2.2. **Examples.**

2.2.1. *Degenerate Langevin on Heisenberg Group.*
The *Heisenberg Group* is the group of matrices of the form

$$\begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3},\ x, y, z \in \mathbb{R}$$

with the following generators (the basis of its Lie algebra):

$$A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},\ A_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and it admits left invariant vector fields

$$X = \frac{\partial}{\partial x} - \frac{1}{2}y\frac{\partial}{\partial z},\ Y = \frac{\partial}{\partial y} + \frac{1}{2}x\frac{\partial}{\partial z},\ z = \frac{\partial}{\partial z}.$$

that form an orthonormal frame of left invariant vector fields for the left invariant metric on $\mathbb{H}^3$ [5]. Note

$$[X, Y] = Z,\ [X, Z] = [Y, Z] = 0.$$

Take $a(X_t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{1}{2}y & \frac{1}{2}x \end{pmatrix}$ into (1.5), where a is the matrix form of A_1 and A_2, and thus Assumption 2.4 is satisfied, we get

$$dX_t = -a(X_t)a(X_t)^\intercal \nabla V dt + \sqrt{2}a(X_t)dB_t,\ a(X_t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{1}{2}y & \frac{1}{2}x \end{pmatrix} \tag{2.6}$$

since

$$\sum_{j=1}^{n+m} \frac{\partial}{\partial x_j}(a(X_t)a(X_t)^\intercal)_{ij} = 0,\ i = 1, \ldots, n+m$$

Explicitly expand all the terms, we get

$$\begin{aligned}
dX_t^{(1)} &= (-\nabla_x V + \frac{1}{2}y\nabla_z V)dt + \sqrt{2}dB_t^{(1)}, \\
dX_t^{(2)} &= (-\nabla_y V - \frac{1}{2}x\nabla_z V)dt + \sqrt{2}dB_t^{(2)}, \\
dX_t^{(3)} &= [\frac{1}{2}y\nabla_x V - \frac{1}{2}x\nabla_y V - \frac{1}{4}(x^2 + y^2)\nabla_z V]dt - \frac{\sqrt{2}}{2}ydB_t^{(1)} + \frac{\sqrt{2}}{2}xdB_t^{(2)}.
\end{aligned}$$

Remark. In [16], a proof of the conditions for the exponential convergence rate of distributions for *degenerate Langevin* on *Heisenberg group* is provided.

2.2.2. *Degenerate Langevin on SE(2) Group.*

For the left-invariant vector fields of SE(2), [14] gives

$$A_1 = \frac{\partial}{\partial_\theta},\ A_2 = \cos\theta\frac{\partial}{\partial_x} + \sin\theta\frac{\partial}{\partial_y},\ A_3 = -\sin\theta\frac{\partial}{\partial_x} + \cos\theta\frac{\partial}{\partial_y},$$

w.r.t the coordinates (θ, x, y), with the lie bracket relationship:

$$[A_1, A_2] = A_3,\ [A_1, A_3] = -A_2,\ [A_2, A_3] = 0.$$

Take $a(X_t) = \begin{pmatrix} 1 & 0 \\ 0 & \cos\theta \\ 0 & \sin\theta \end{pmatrix}$ into (1.5), where a is the matrix form of A_1 and A_2, and thus Assumption 2.4 is satisfied, we get

$$dX_t = -a(X_t)a(X_t)^\intercal \nabla V dt + \sqrt{2}a(X_t)dB_t,\ a(X_t) = \begin{pmatrix} 1 & 0 \\ 0 & \cos\theta \\ 0 & \sin\theta \end{pmatrix}, \tag{2.7}$$

since

$$\sum_{j=1}^{n+m} \frac{\partial}{\partial x_j}(a(X_t)a(X_t)^\intercal)_{ij} = 0,\ i = 1, \ldots, n+m,$$

which implies

$$
\begin{aligned}
dX_t^{(1)} &= -\nabla_\theta V dt + \sqrt{2} dB_t^{(1)}, \\
dX_t^{(2)} &= -(\cos^2\theta \nabla_x V + \sin\theta\cos\theta \nabla_y V)dt + \sqrt{2}\cos\theta dB_t^{(2)}, \\
dX_t^{(3)} &= -(\sin\theta\cos\theta \nabla_x V + \sin^2\theta \nabla_y V)dt + \sqrt{2}\sin\theta dB_t^{(2)}.
\end{aligned}
$$

In fact, we can also choose A_1 and A_3 for the matrix a, as Assumption 2.4 would also be satisfied. Furthermore, if we take all of A_1, A_2, A_3 and consider the *non-degenerate Langevin* case, the corresponding Itô form is

$$
dX_t = -\nabla V dt + \sqrt{2} a(X_t) dB_t, \; a(X_t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}, \tag{2.8}
$$

where a is the matrix form of A_1, A_2 and A_3, B_t is the 3-dimensional *Brownian motion*.

Remark. The exponential convergence rate of distribution for (2.7) and (2.8) haven't been checked yet, but for computational convenience in later *simulations* section we will mainly focus on the SE(2) case. The analysis for SE(2) will be provided in the future, which is expected to follow similarly to *Heisenberg group* and *Displacement group* case in [16].

3. Simulations

This section shows the results for the simulations of *Langevin dynamics* on Lie groups and Euclidean space, and compare their final distributions along with generated paths. All simulations are done by using *Euler-Maruyama Method* [24], which is a commonly used numerical method in simulating Itô SDE.

3.1. **Generating Gaussian distribution.**

Consider the following Multivariate Gaussian distributions:

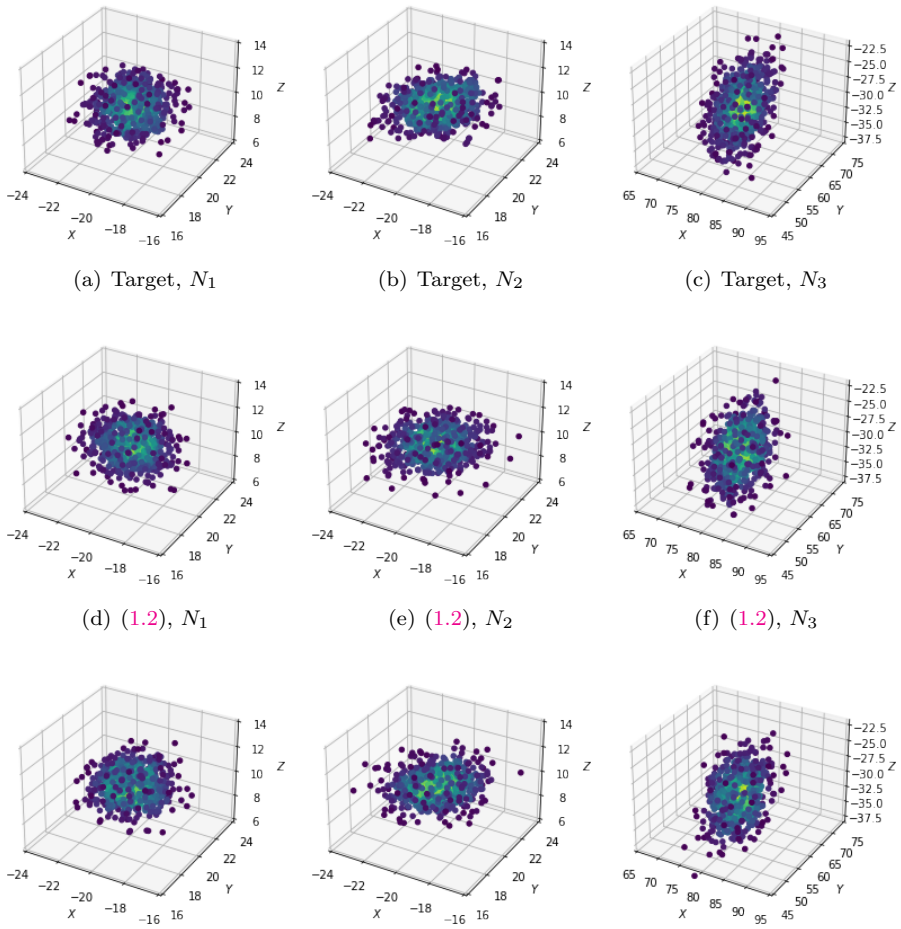
$$
\begin{aligned}
&N_1 \sim \mathcal{N}(\mu_1; \Sigma_1),\ \mu_1 = [-20, 20, -10], && \Sigma_1 = \begin{pmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{pmatrix}; \\
&N_2 \sim \mathcal{N}(\mu_2; \Sigma_2),\ \mu_2 = [-20, 20, -10], && \Sigma_2 = \begin{pmatrix} 1.0 & 0.5 & 0 \\ 0.5 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{pmatrix}; \\
&N_3 \sim \mathcal{N}(\mu_3; \Sigma_3),\ \mu_3 = [80, 60, -30], && \Sigma_3 = \begin{pmatrix} 9.0 & 0 & 0 \\ 0 & 9.0 & 7.0 \\ 0 & 7.0 & 6.0 \end{pmatrix}.
\end{aligned}
$$

In the simulation, we try to generate the desired Gaussian distribution from a random initial distribution. In other word, in (1.2) (2.7) and (2.8), we set

$$
\exp(-V) = \frac{\exp\left(-\frac{1}{2}(x-\mu_i)^\intercal \Sigma_i^{-1}(x-\mu_i)\right)}{(2\pi)^{3/2}\left|\Sigma_i\right|^{1/2}},\ \nabla V = \Sigma_i^{-1}(x-\mu_i),\ i = 1, 2, 3, \tag{3.1}
$$

and generate numerical simulations.

Figure 3 shows the final distributions of the simulations. (a)-(c) are the samples of Gaussian distribution N_1, N_2, N_3 generated by python `numpy` package with 750



(a) Target, N_1 (b) Target, N_2 (c) Target, N_3

(d) (1.2), N_1 (e) (1.2), N_2 (f) (1.2), N_3

(g) (2.7), N_1 (h) (2.7), N_2 (i) (2.7), N_3

Figure 3. The generated samples, sample size $N = 750$, see Sample Mean and Covariance Matrix table for details.

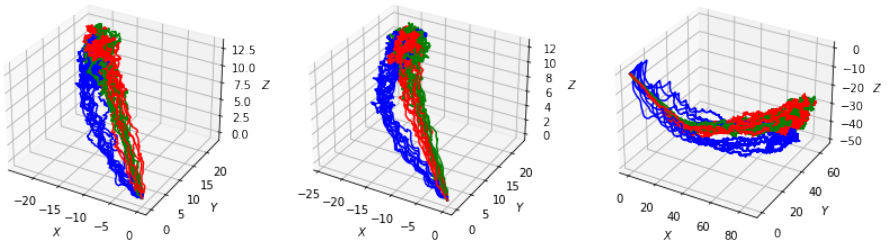
points, respectively, which are the desired distributions. (d)-(f) are the final distributions generated by (1.2), and (g)-(i) are the final distributions generated by (2.7). All initial points are set to be $[0, 0, 0]$, and the parameters (time, step numbers) of simulations are set sufficiently large so that the distributions converge.

Compare figures on each column of Figure 3, it can be easily seen that they all follow the same Gaussian distribution, with small noises generated by sampling, which shows the *overdamped Langevin dynamics* produce the desired results in **Sample Generation**.

In addition to the final distributions, we are also interested in how the distributions converge to the *invariant distribution*, or the paths generated by *overdamped Langevin dynamics*. Figure 4 shows the paths generated by (1.2, green), (2.7, blue) and (2.8, red), where (a) (b) (c) corresponds to the generation of different Gaussian

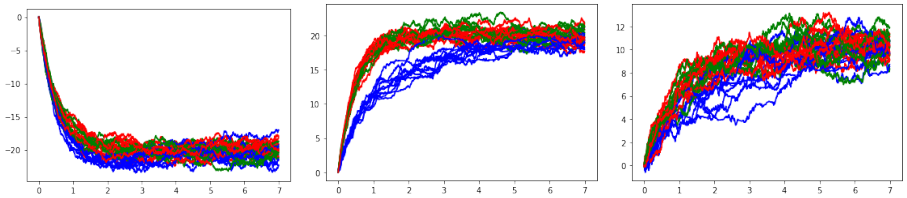
distributions N_1, N_2, N_3, and the starting point for each path is set to be $[0, 0, 0]$. It can be seen even though the final distributions are the same, the paths generated are different.

Remark. Note here the green paths and red paths look similar. This is because in the special case of SE(2), the *drift term* $-a(X_t)a(X_t)^\intercal \nabla V dt$ in (2.8) coincides with (1.2) since $a(X_t)a(X_t)^\intercal = I$.



(a) Paths, N_1 (b) Paths, N_2 (c) Paths, N_3

Figure 4. Generated Paths, N = 10



(a) x-axis (b) y-axis (c) z-axis

Figure 5. Convergence of axis w.r.t. time, N_2, N = 10

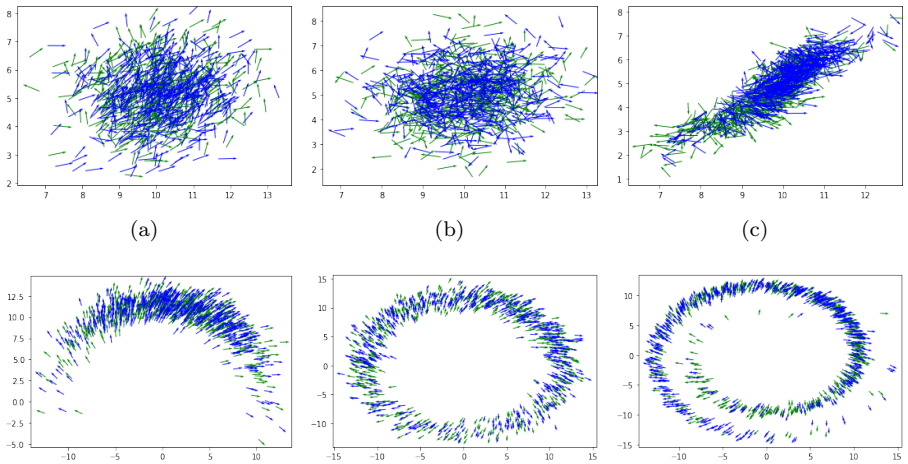
Figure 5 shows the generation of paths in Figure 4 (b) with respect to time. Similarly, blue lines represent the simulation of (2.7), green lines represent (1.2), and red lines represent (2.8). It can be seen that for (1.2) and (2.8), the simulated paths converge exponentially fast to the invariant distribution, while the paths of (2.7) converge slower. One simple reason is that V (3.1) is strictly convex in $\mathbb{R}^3$, and *Curvature Dimension Inequality* 2.3 holds, which guarantees the exponential convergence behavior for (1.2) and (2.8). However, the *Generalized Curvature Dimension Inequality* 2.5 is not trivially hold for this V in (2.7). Further investigations need to be conducted.

3.2. **Generating Partially Wrapped Gaussian.**

In [25], *Partially Wrapped Normal Distributions* are introduced for SE(2) estimation, where the *p.d.f* is simplified to

$$f(x,\mu,\Sigma) = \sum_{k=-\infty}^{\infty} \mathcal{N}(x + \begin{bmatrix} 2\pi k \\ 0 \\ 0 \end{bmatrix}; \mu, \Sigma)$$
$$= \sum_{k=-\infty}^{\infty} \frac{exp(-\frac{1}{2}(\overline{x}-\mu)^{\intercal}\Sigma^{-1}(\overline{x}-\mu))}{(2\pi^{3/2}\left|\Sigma\right|^{1/2})},$$

where $\overline{x} = x + \begin{bmatrix} 2\pi k \\ 0 \\ 0 \end{bmatrix}$, $k \in \mathbb{Z}$, with $x \in [0, 2\pi) \times \mathbb{R}^2$, $\mu \in [0, 2\pi) \times \mathbb{R}^2$, and symmetric positive definite $\Sigma \in \mathbb{R}^{3\times 3}$. Note that *PWND* can be generated by $\mathcal{N}(x; \mu, \Sigma)$ in $\mathbb{R}^3$ for $SE(2)$, since in the (θ, x, y) coordinates $\theta + 2\pi k$ for $k \in \mathbb{Z}$ have no difference [appendix A.2.2]. However, the function $V = -\log f$ in the *overdamped Langevin dynamics* is not necessarily strictly convex (we will check in the future for the convexity) when we adapt the *PWND*, and we are interested in the results of simulations in this case.



(a) (b) (c)

(d) (e) (f)

Figure 6. Partially Wrapped Gaussian distribution, N = 500

Figure 6 shows the results of simulations, and we generate the samples as in [25] using `numpy` package, which is represented by blue quivers, while the green quivers represent the samples generated by the simulations of (2.7). We use $\mu = [1, 10, 5]^{\intercal}$in all cases. In the first row, we apply translation first and rotation later, in the second row we do it vice versa. Each arrow indicates the transformation applied to the

vector $[1,0]^\intercal$, and we set the parameters of $PWND$ for each column as follows:

$$\text{Column 1: } c_{11}=0.3, c_{22}=1.0, c_{33}=1.0, \rho_{12}=0.1, \rho_{13}=0.1, \rho_{23}=0.1$$
$$\text{Column 2: } c_{11}=3.0, c_{22}=1.0, c_{33}=1.0, \rho_{12}=0.1, \rho_{13}=0.1, \rho_{23}=0.1$$
$$\text{Column 3: } c_{11}=3.0, c_{22}=1.0, c_{33}=1.0, \rho_{12}=0.9, \rho_{13}=0.9, \rho_{23}=0.9$$

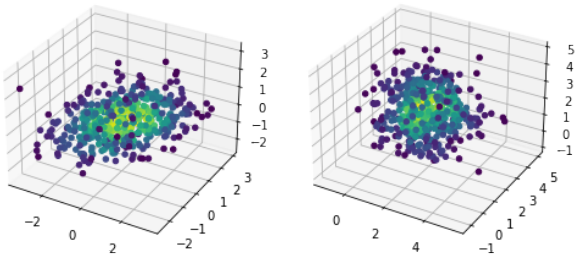
Figure 6 indicates that (2.7) works well even for this more complicated case.

3.3. **Discussions.**

The above simulations show examples that when $V = -\log(\pi)$, where π is a *probability density function*, with some initial sets of points, the distributions in (1.2) (2.7) and (2.8) all converge to the invariant distribution π. In addition, as shown in the previous two subsections, the convergence rate of (1.2) and (2.8) are both exponential w.r.t. time t.

There are still some limitations of the above simulations. First, we only simulated *overdampled Langevin dynamics* on *SE(2)* and $\mathbb{R}^3$. In fact, we tried the simulations on other *Lie groups*, but failed in many cases. For *Heisenberg group*, when the mean of the target distribution and initial points are both close to $[0,0,0]$, then everything performs as desired, while in other cases either the distributions do not converge as expected (or converge too slowly) or numerical errors occur (see Figure 7). One possible explanation is that (2.6) includes 2-degree polynomial terms which makes the simulation numerically unstable, as the *Heisenberg group* is unbounded. For $SO(3)$ and more complicated *Lie groups*, the *left invariant fields* are also unsatisfactory for numerical simulation.

In addition, we only considered Gaussian and modified Gaussian distributions, which only guarantee exponential convergence rate for *overdamped Langevin dynamics* on $\mathbb{R}^3$, and we didn't find good distributions such that the *Generalized Curvature Dimension Inequality* 2.5 holds for the $SE(2)$ case.



(a) (b)

Figure 7. Simulations of (2.6). See Sample Mean and Covariance for Heisenberg Case for details.

4. Future Works

The convergence behavior of *overdamped Langevin dynamics* illustrates that one can generate samples for almost all continuous probability distributions from any

initial sample. Furthermore, the *overdamped Langevin dynamics* on *Lie groups* considers manifold structure, and can also generate the desired sample as *overdamped Langevin dynamics*. The main differences between them are the convergence rate and generating path (or geometric control). Future works could be conducted mainly from the following two perspectives.

From a forward perspective, we want to find a set of satisfactory functions V for (2.7) such that it both satisfy *Generalized Curvature Dimension Inequality* 2.5 and is useful in application. To achieve this, we may either find an equivalent statement of *GCDI* or shrink it to a sufficient condition with more straightforward expression. For example, understanding the convexity in Lie group structure may be helpful as convexity of V is a necessary condition of *CDI* 2.3 for the *overdamped Langevin dynamics* in *Euclidean space*.

In addition, as presented in the Introduction section, *overdamped Langevin dynamics* plays an important role in non-convex optimization. It could be interesting to incorporate the *overdamped Langevin dynamics* on *Lie groups* into SGLD and see how is performs compared to the original method, and this might be introduced to more complicated methods such as *Replica Exchange Langevin Diffusion* [12].

From a backward perspective, one may use *neural networks* to learn and approximate the *overdamped Langevin dynamics*, which is known as *diffusion model*. In this model, an explicit formula of V is not expected. Instead, given a family of data sets, assuming they are sampled from the same distribution $x_0 \sim q(x)$, one could add small Gaussian noises in T steps so that $x_T \sim \mathcal{N}(0; I)$, with the conditional probability $q(x_t|x_{t-1})$ on each step determined by a prefix noise size, which constructs the forward diffusion process. If the forward process can be reversed, then one can generate new samples following the original distribution from an initial Gaussian distribution.

However, it is technically impossible to directly compute the explicit form of the reverse process (or in each step compute the $p(x_{t-1}|x_t)$). Therefore, neural networks are used to learn the conditional probability $p_\theta(x_{t-1}|x_t) \approx p(x_{t-1}|x_t)$, which is an approximation of the reverse diffusion process [35].

Currently *diffusion model* mainly focuses on the approximation of *reversible Langevin dynamics* on *Euclidean space*, and no work has been done for *Lanegvin dynamics* on *Lie groups*. It is very promising that introducing *overdamped Langevin dynamics* on *Lie groups* in *diffusion model* might improve the effectiveness of the model on data in medical images and computer vision (pose estimation, object tracking, etc), where a lot of Lie group structures are used. With some additional geometric control, one could restrict the neural networks to follow Lie group structure. Further investigations need to be conducted.

Acknowledgments

This research was supported by the REU program at the Department of Mathematics, University of Michigan. I'd like to thank my mentor, Dr. Qi Feng, for his support and guidance throughout the project. I'd also like to thank the organizers of REU and the other participants, from whom I have learned a great deal.

References

[1] Anton Arnold et al. "On convex Sobolev inequalities and the rate of convergence to equilibrium for Fokker-Planck type equations". In: *Communications in Partial Differential Equations* (2001).

[2] Dominique Bakry and Michel Émery. "Diffusions hypercontractives". In: *Seminaire de probabilités XIX 1983/84*. Springer, 1985, pp. 177–206.

[3] Dominique Bakry, Ivan Gentil, Michel Ledoux, et al. *Analysis and geometry of Markov diffusion operators*. Vol. 103. Springer, 2014.

[4] Dominique Bakry et al. "A simple proof of the Poincaré inequality for a large class of probability measures". In: *Electronic Communications in Probability* 13 (2008), pp. 60–66.

[5] Fabrice Baudoin. "Sub-Laplacians and hypoelliptic operators on totally geodesic Riemannian foliations". In: *arXiv preprint arXiv:1410.3268* (2014).

[6] Fabrice Baudoin and Nicola Garofalo. "Curvature-dimension inequalities and Ricci lower bounds for sub-Riemannian manifolds with transverse symmetries". In: *Journal of the European Mathematical Society* 19.1 (2016), pp. 151–219.

[7] James Blowey, John P Coleman, and Alan W Craig. *Theory and numerics of differential equations: Durham 2000*. Springer Science & Business Media, 2001.

[8] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[9] Ngoc Huy Chau et al. "On stochastic gradient langevin dynamics with dependent data streams: The fully nonconvex case". In: *SIAM Journal on Mathematics of Data Science* 3.3 (2021), pp. 959–986.

[10] Tzuu-Shuh Chiang, Chii-Ruey Hwang, and Shuenn Jyi Sheu. "Diffusion for global optimization in R^n". In: *SIAM Journal on Control and Optimization* 25.3 (1987), pp. 737–753.

[11] Gregory S Chirikjian. "Information theory on lie groups and mobile robotics applications". In: *2010 IEEE International Conference on Robotics and Automation*. IEEE. 2010, pp. 2751–2757.

[12] Wei Deng et al. "Non-convex learning via replica exchange stochastic gradient mcmc". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2474–2483.

[13] Remco Duits and Erik Franken. "Left-invariant parabolic evolutions on SE(2) and contour enhancement via invertible orientation scores Part I: Linear left-invariant diffusion equations on SE(2)". In: *Quarterly of Applied Mathematics* 68.2 (2010), pp. 255–292.

[14] Remco Duits and Erik Franken. "Left-invariant parabolic evolutions on SE(2) and contour enhancement via invertible orientation scores Part II: Nonlinear left-invariant diffusions on invertible orientation scores". In: *Quarterly of applied mathematics* 68.2 (2010), pp. 293–331.

[15] Jacques Faraut. *Analysis on Lie Groups*. Cambridge University Press, 2008. URL: https://www.cambridge.org/core/books/analysis-on-lie-groups/8E34C57AFA7E0FBCDD38B9D2106C71BF.

[16] Qi Feng and Wuchen Li. "Entropy dissipation for degenerate stochastic differential equations via sub-Riemannian density manifold". In: *arXiv preprint arXiv:1910.07480* (2019).

[17] Qi Feng and Wuchen Li. "Entropy dissipation via Information Gamma calculus: Non-reversible stochastic differential equations". In: *arXiv preprint arXiv:2011.08058* (2020).

[18] Qi Feng and Wuchen Li. "Hypoelliptic entropy dissipation for stochastic differential equations". In: *arXiv preprint arXiv:2102.00544* (2021).

[19] Qi Feng and Wuchen Li. "Sub-Riemannian Ricci curvature via generalized Gamma z calculus". In: *arXiv preprint arXiv:2004.01863* (2020).

[20] P Thomas Fletcher, Conglin Lu, and Sarang Joshi. "Statistics of shape via principal geodesic analysis on Lie groups". In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 1. IEEE. 2003, pp. I–I.

[21] Saul B Gelfand and Sanjoy K Mitter. "Recursive stochastic algorithms for global optimization in R^d". In: *SIAM Journal on Control and Optimization* 29.5 (1991), pp. 999–1018.

[22] Igor Gilitschenski et al. "Deep orientation uncertainty learning based on a bingham loss". In: *International Conference on Learning Representations*. 2019.

[23] Ayoosh Kathuria. *Intro to optimization in deep learning: Gradient Descent*. 2018. URL: https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/.

[24] Peter E Kloeden and Eckhard Platen. "Stochastic differential equations". In: *Numerical solution of stochastic differential equations*. Springer, 1992, pp. 103–160.

[25] Gerhard Kurz, Igor Gilitschenski, and Uwe D Hanebeck. "The partially wrapped normal distribution for SE (2) estimation". In: *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*. IEEE. 2014, pp. 1–8.

[26] Adam Leach et al. "Denoising Diffusion Probabilistic Models on SO(3) for Rotational Alignment". In: *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*. 2022. URL: https://openreview.net/forum?id=BY88eBbkpe5.

[27] Peter A Markowich and Cédric Villani. "On the trend to equilibrium for the Fokker-Planck equation: an interplay between physics and functional analysis". In: *Mat. Contemp* 19 (2000), pp. 1–29.

[28] David Mohlin, Josephine Sullivan, and Gérald Bianchi. "Probabilistic orientation estimation with matrix fisher distributions". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4884–4893.

[29] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[30] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. "Non-convex learning via Stochastic Gradient Langevin Dynamics: a nonasymptotic analysis". In: *Proceedings of the 2017 Conference on Learning Theory*. Ed. by Satyen Kale and Ohad Shamir. Vol. 65. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1674–1703. URL: https://proceedings.mlr.press/v65/raginsky17a.html.

[31] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. "A micro Lie theory for state estimation in robotics". In: *arXiv preprint arXiv:1812.01537* (2018).

[32] Oncel Tuzel, Fatih Porikli, and Peter Meer. "Learning on lie groups for invariant detection and tracking". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[33] Oncel Tuzel, Raghav Subbarao, and Peter Meer. "Simultaneous multiple 3D motion estimation via mode finding on Lie groups". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 1. IEEE. 2005, pp. 18–25.

[34] Max Welling and Yee W Teh. "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer. 2011, pp. 681–688.

[35] Lilian Weng. "What are diffusion models?" In: *lilianweng.github.io* (2021). URL: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.

[36] Pan Xu et al. "Global convergence of Langevin dynamics based algorithms for nonconvex optimization". In: *Advances in Neural Information Processing Systems* 31 (2018).

[37] Qiang Xu and Dengwu Ma. "Applications of Lie groups and Lie algebra to computer vision: A brief survey". In: *2012 International Conference on Systems and Informatics (ICSAI2012)*. IEEE. 2012, pp. 2024–2029.

Appendix A. Matrix Lie Groups

A.1. **Lie group and Lie algebra.**

The *exponential of a matrix* $X \in M(n, \mathbb{F})$ is defined by Taylor series:

$$exp(X) = I + \sum_{k=1}^{\infty} \frac{X^k}{k!},$$

where $M(n, \mathbb{F})$ denotes the set of all $n \times n$ matrices. Similarly, the *Logarithm of a matrix* $g \in M(n, \mathbb{F})$ is defined:

$$log(g) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} (g - I)^k, \text{ for } \|g - I\| < 1.$$

Definition A.1 (**Lie bracket**)**.**
A **Lie bracket** or **commutator** on a *Lie algebra* is a binary operation:

$$\mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}, \ (X, Y) \to [X, Y] = XY - YX,$$

with properties:

$$[X, Y] = -[Y, X],$$
$$[X, [Y, Z]] = [[X, Y], Z] + [Y, [X, Z]]. \quad (Jordan\ Identity)$$

Definition A.2 (**Lie algebra**)**.**
A Lie algebra is a vector space $\mathfrak{g}$ over $\mathbb{F}$ equipped with the Lie bracket operator satisfying the following axioms ($\forall\ x, y, z \in \mathfrak{g},\ a, b \in \mathbb{F}$):

- *Bilinearity:* $[ax + by, z] = a[x, z] + b[y, z], [z, ax + by] = a[z, x] + b[z, y]$.
- *Alternativity:* $[x, x] = 0$.
- *Jacobi identity:* [x,[y,z]]+[y,[z,x]]+[z,[x,y]]=0.

Definition A.3 (**Matrix Lie Group and Lie algebra**)**.**
A **Lie group** $\mathcal{G}$ is a smooth manifold whose elements satisfy the group axioms. A **Matrix (Linear) Lie group** is a *closed subgroup (submanifold)* of $\mathcal{G} \subseteq GL(n, \mathbb{F})$, and the **Lie algebra of** $\mathcal{G}$ is defined as:

$$\mathfrak{g} = Lie(\mathcal{G}) = \{X \in M(n, \mathbb{R}) |\ \forall\ t \in \mathbb{R},\ exp(tX) \in \mathcal{G}\}.$$

which is also the **tangent vector space of** $\mathcal{G}$ **at** I (the identity matrix), denoted by $T_e(G)$. In addition, the dimension of $\mathcal{G} = \dim \mathfrak{g}$ [15].

Definition A.4 (**generators**)**.**
Note that $\dim G = \dim \mathfrak{g}$ is finite, we can find a basis $\{G_1, \ldots, G_k\}$ of the Lie algebra $\mathfrak{g}$, which is called the **generators**. $\forall\ g \in \mathfrak{g}$, we can write $g = \sum_{i=1}^{k} c_i G_i$, and define the map:

$$alg : \mathbb{R}^k \to \mathfrak{g} \subset \mathbb{R}^{n \times n},\ \mathbf{c} \to \sum_{i=1}^{k} c_i G_i.$$

Some papers also use $\wedge$ instead of alg such that $\mathbf{c}^\wedge = alg(\mathbf{c})$, and use $\vee$ to represent the inverse map such that $(\mathbf{c}^\wedge)^\vee = \mathbf{c}$.

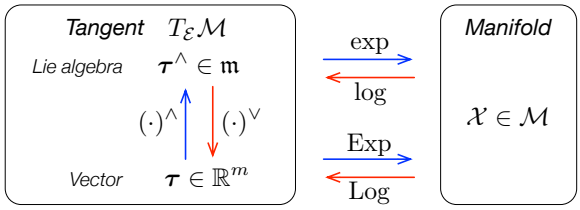


Figure 8. Mappings between the Lie group $\mathcal{G} = \mathcal{M}$ and its Lie algebra (set ε to be the identity matrix). Note that $\mathfrak{g} = \mathfrak{m}$ is a vector space (assume $\dim \mathfrak{g} = n$), then we may define *isomorphisms* $alg(\cdot) = (\cdot)^{\vee} : \mathbb{R}^n \to \mathfrak{g}$ and $alg^{-1}(\cdot) = (\cdot)^{\wedge} : \mathfrak{g} \to \mathbb{R}^n$. See A.2 for examples. [Source: [31]]

A.2. **Examples.**

A.2.1. *SO(2) group.*

SO(2) is the group of rotations in 2D space., with generators

$$G = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

and any element of its *Lie algebra* $\mathfrak{so}(2)$ can be represented by

$$\theta \in \mathbb{R},\ \theta_{\times} = \theta G \in \mathfrak{so}(2).$$

and its exponential map

$$exp(\theta_{\times}) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \in SO(2).$$

The action of an element of $SO(2)$ on a 2D object is to rotate it with θ degrees w.r.t. the origin counter-clockwisely.

A.2.2. *SE(2) group.*

$SE(2)$ is the group of rigid transformations in the 2D plane, which is represented by linear transformations on homogeneous 3-vectors:

$$R \in SO(2),\ t \in \mathbb{R}^2,$$
$$C = \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right) \in SE(2) \subset \mathbb{R}^{3\times 3}.$$

The matrix representation is:

$$x = (x\ y\ \omega)^{\intercal} \in \mathbb{R}^3,$$
$$C \cdot x = \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right) \cdot x = \begin{pmatrix} R(x,y)^{\intercal} + \omega t \\ \omega \end{pmatrix}.$$

The action of an element of $SE(2)$ on a 2D object is equivalent to rotate it by θ degrees counter-clockwisely w.r.t. the origin and then do translation. In the above equations (x, y) denotes the current position of the object, and t denotes the

direction of translation, R is a $SO(2)$ rotation matrix, and ω is the translation step size. The Lie algebra $\mathfrak{se}(2)$ has 3 degrees of freedom, with generators:

$$G_1 = \left(\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right),\ G_2 = \left(\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \end{array}\right),\ G_3 = \left(\begin{array}{cc|c} 0 & -1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right).$$

Every element in $\mathfrak{se}(2)$ can be represented by:

$$(u_1\ u_2\ \theta)^\intercal \in \mathbb{R}^3,$$
$$u_1 G_1 + u_2 G_2 + \theta G_3 \in \mathfrak{se}(2).$$

As before, we can also derive a closed form for the exponential map:

$$\begin{aligned}
v = (x\ y\ \theta)^\intercal &= (u\ \theta)^\intercal \in \mathbb{R}^3 \\
alg(v) &= \left(\begin{array}{cc|c} 0 & -\theta & x \\ \theta & 0 & y \\ \hline 0 & 0 & 0 \end{array}\right), \\
exp(alg(v)) &= exp\left(\begin{array}{c|c} \theta_\times & u \\ \hline 0 & 0 \end{array}\right) \\
&= \left(\begin{array}{c|c} exp(\theta_\times) & Vu \\ \hline 0 & 1 \end{array}\right) \\
&= \left(\begin{array}{c|c} R & Vu \\ \hline 0 & 1 \end{array}\right), \\
R = exp(\theta_\times) &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \\
V &= (\sum_{i=0}^{\infty} \frac{(-1)^i \theta^{2i}}{(2i+1)!}) \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + (\sum_{i=0}^{\infty} \frac{(-1)^i \theta^{2i+1}}{(2i+2)!}) \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\
&= \begin{pmatrix} \frac{\sin\theta}{\theta} & -\frac{1-\cos\theta}{\theta} \\ \frac{1-\cos\theta}{\theta} & \frac{\sin\theta}{\theta} \end{pmatrix}.
\end{aligned}$$

Consider the $log(\cdot)$ function in the other direction:

$$\begin{aligned}
A &= \frac{\sin\theta}{\theta}, \\
B &= \frac{1-\cos\theta}{\theta}, \\
V^{-1} &= \frac{1}{A^2+B^2} \begin{pmatrix} A & B \\ -B & A \end{pmatrix}, \\
log\left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right) &= alg\begin{pmatrix} V^{-1} \cdot t \\ \theta \end{pmatrix} \in \mathfrak{se}(2).
\end{aligned}$$

A.2.3. *SO(3) group.*
$SO(3)$ is the group of rotations in 3D space, represented by 3×3 orthogonal matrices with unit determinant. One can denote:

$$SO(3) := \{X \in GL(3,\mathbb{R}) :\ R^\intercal = R^{-1},\ \det R = 1\}.$$

The Lie algebra $\mathfrak{so}(3)$ is the set of skew-symmetric 3×3 matrices, with generators:

$$G_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \; G_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \; G_1 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The mapping $alg : \mathbb{R}^3 \to \mathfrak{so}(3)$ sends $3 - vectors$ to their skew matrix:

$$\omega = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \in \mathbb{R}^3,$$

$$alg(\omega) = \omega_\times = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} = aG_1 + bG_2 + cG_3 \in \mathfrak{so}(3).$$

We can use a interesting property of skew-symmetric matrices, for $\omega \in \mathbb{R}^3$:

$$\omega_\times^3 = -(\omega^\intercal \omega) \cdot \omega_\times,$$

so that

$$\omega_\times^{2i+1} = (-1)^i \theta^{2i} \omega_\times,$$
$$\omega_\times^{2i+2} = (-1)^i \theta^{2i} \omega_\times^2.$$

The tangent vector ω camn be interpreted as an axis-angle representation of rotation: its exponential is the rotation around the axis $\omega / \|\omega\|$ by $\|\omega\|$ radians (denote $\theta = \sqrt{\omega^\intercal \omega}$):

$$\begin{aligned} exp(alg(\omega)) &= exp(\omega_\times) \\ &= I + \sum_{i=0}^{\infty} [\frac{\omega_\times^{2i+1}}{(2i+1)!} + \frac{\omega_\times^{2i+2}}{(2i+2)!}] \\ &= I + (\frac{\sin\theta}{\theta})\omega_\times + (\frac{1-\cos\theta}{\theta^2})\omega_\times^2 \\ &= \begin{bmatrix} 1-(b^2+c^2)(\frac{1-\cos\theta}{\theta^2}) & -c\frac{\sin\theta}{\theta} + ab(\frac{1-\cos\theta}{\theta^2}) & b\frac{\sin\theta}{\theta} + ac(\frac{1-\cos\theta}{\theta^2}) \\ c\frac{\sin\theta}{\theta} + ab(\frac{1-\cos\theta}{\theta^2}) & 1-(a^2+c^2)(\frac{1-\cos\theta}{\theta^2}) & -a\frac{\sin\theta}{\theta} + bc(\frac{1-\cos\theta}{\theta^2}) \\ -b\frac{\sin\theta}{\theta} + ac(\frac{1-\cos\theta}{\theta^2}) & a\frac{\sin\theta}{\theta} + bc(\frac{1-\cos\theta}{\theta^2}) & 1-(a^2+b^2)(\frac{1-\cos\theta}{\theta^2}) \end{bmatrix}. \end{aligned}$$

Finally we note that $\forall \; R \in SO(3)$, one can express

$$R = I - aG_1 + bG_2 - cG_3.$$

Thus the exponential map can be inverted to give a backward logarithm from $SO(3)$ to $\mathfrak{so}(3)$ and also θ:

$$log(R) = \frac{\theta}{2\sin\theta} \cdot (R - R^\intercal),$$
$$\omega = Log(R) = \frac{\theta(R-R^\intercal)^\vee}{2\sin\theta},$$
$$\theta = \cos^{-1}(\frac{\mathrm{tr}\,(R)-1}{2}).$$

In addition to the above *Lie algebra* parameterization, there are also some other ways to parameterize $SO(3)$ in engineering, such as *Euler angles* and *Quaternions* [See C.2].

Appendix B. Langevin Dynamics

B.1. Itô Process.

Theorem B.1 (**The 1-dimensional Itô Formula**).
Let X_t be an 1-dimensional Itô process given by

$$dX_t = u(x,t)dt + v(x,t)dB_t.$$

Let $g(t,x) \in C^2([0,\infty)\times\mathbb{R})$ (i.e. g is twice continuously differentiable on $[0,\infty)\times\mathbb{R}$). Then

$$Y_t = g(t, X_t)$$

is again an Itô process, and

$$dY_t = \frac{\partial g}{\partial t}(t,X_t)dt + \frac{\partial g}{\partial x}(t,X_t)dX_t + \frac{1}{2}\frac{\partial^2 g}{\partial x^2}(t,X_t)\cdot(dX_t)^2,$$

where $(dX_t)^2 = (dX_t)\cdot(dX_t)$ is computed according to the rules

$$dt\cdot dt = dt\cdot dB_t = dB_t\cdot dt = 0,\ dB_t\cdot dB_t = dt.$$

Remark. For the definition of Itô process and proof of Itô formula, check the details in [29] (Øksendal, 2014).

B.2. Diffusion process.

Definition B.2 (**Itô diffusion**).
A (time-homogeneous) **Itô diffusion** is a stochastic process

$$X_t(\omega) = X(t,\omega) : [s,\infty)\times\Omega \to \mathbb{R}^n$$

satisfying a stochastic differential equation of the form

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t,\ t \geq s;\ X_s = x,$$

where B_t is *m-dimensional Brownian motion* and $b : \mathbb{R}^n \to \mathbb{R}^n,\ \sigma : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ satisfy the *Lipschitz condition*

$$|b(x) - b(y)| + |\sigma(x) - \sigma(y)| \leq D\,|x-y|\,;\ \forall\ x,y \in \mathbb{R}^n,$$

for some constant $D > 0$. The vector field b is called the *drift coefficient* of X; the matrix field σ is called the *diffusion coefficient* of X. Note that b and σ do not depend upon time; otherwise, X would be referred to only as an Itô process.

Lemma B.3 (**Itô and Stratonovich conversion**).
Given an N-dimensional Stratonovich SDE:

$$dX_t = \underline{a}(t,X_t)dt + \sum_{j=1}^{M} b_j(t,X_t)\circ dB_t^{(j)}$$

and an Itô SDE with the same solution:

$$dX_t = a(t,X_t)dt + \sum_{j=1}^{M} b_j(t,X_t)dB_t^{(j)}$$

where $X_t \in \mathbb{R}^N$, $a, \underline{a} \in \mathbb{R}^N$, $b_j \in \mathbb{R}^N$, $j = 1,\ldots,M$, $B_t^{(j)}$ be an 1-dimensional Brownian motion for $j = 1,\ldots,M$. The conversion formula is [7]:

$$\underline{a}_i(t,X_t) = a_i(t,X_t) + \frac{1}{2}\sum_{k=1}^{N}\sum_{j=1}^{M} b_{kj}(t,X_t)\frac{\partial}{\partial x_k}b_{ij}(t,X_t),\ i = 1,\ldots,N.$$

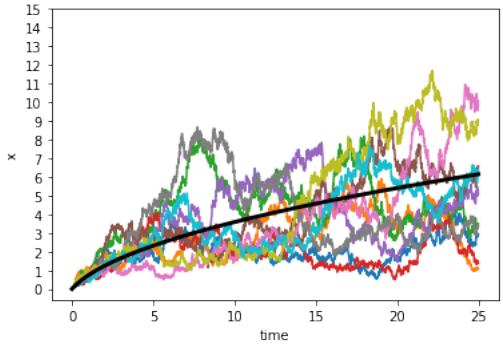


Figure 9. This figure shows an example of diffusion process. The black line is the path of 1-dimensional ODE $dX_t = \frac{1}{X_t+1}dt$ with $X_0 = 0$, while the colorful lines are the paths of 1-dimensional diffusion process $dX_t = \frac{1}{X_t+1}dt + \frac{\sqrt{X_t}}{2}dB_t$.

Appendix C. Supplement

Lemma C.1 (Bochner's formula).
Consider Riemannian Manifold $(\mathcal{M}, \mathfrak{g})$, $\mathfrak{g} : \mathcal{T}_{\mathcal{M}} \to \mathbb{R}^n$,

$$\Gamma_2(f,f) = (\mathfrak{hess} f)^2 + \underbrace{\mathfrak{Ric}(\nabla f, \nabla f)}_{Ricci\ Curvature}. \tag{C.2}$$

for smooth $f : \mathcal{M} \to \mathbb{R}$ *[3].*

C.1. Sketch of proof for 2.1.1.

The *Fokker-Planck Equation* is given by:

$$\begin{aligned}
\partial_t p(x,t) &= \nabla \cdot (\nabla V(x) \cdot p(x,t)) + \nabla^2 p(x,t) \\
&= -\nabla \cdot (\nabla \log \pi(x) \cdot p(x,t)) + \nabla(p(t,x)\nabla \log p(x,t)) \\
&= \nabla(p(t,x)\nabla \log \frac{p(x,t)}{\pi(x)}).
\end{aligned}$$

For convenience, we use ∂_t for $\frac{\partial}{\partial t}$, ∇ for $\frac{\partial}{\partial x}$.

(a)

$$\begin{aligned}
\frac{d}{dt}\mathcal{D}(p(x,t)) &= \frac{d}{dt}\int p(x,t)\log\frac{p(x,t)}{\pi(x)}dx \\
&= \int (\partial_t p(x,t))\log\frac{p(x,t)}{\pi(x)}dx + \int \partial_t p(x,t)dx \\
&= \int \partial_t p(x,t)(\log p(x,t) - \log\pi(x))dx + \partial_t \int p(x,t)dx \\
&= \int \partial_t p(x,t)\log\frac{p(x,t)}{\pi(x)}dx \\
&= \int \nabla\cdot(p(x,t)\nabla\log\frac{p(x,t)}{\pi(x)})\log\frac{p(x,t)}{\pi(x)}dx \\
&= -\mathcal{I}(p(x,t)).
\end{aligned}$$

(b)

$$\begin{aligned}
\frac{d}{dt}\mathcal{I}(p(x,t)) &= \int \Gamma_1(\log\frac{p(x,t)}{\pi(x)}, \log\frac{p(x,t)}{\pi(x)})\partial_t p(x,t)dx + 2\int(\nabla\log\frac{p(x,t)}{\pi(x)}, \partial_t\nabla\log\frac{p(x,t)}{\pi(x)})p(x,t)dx \\
&= \int \Gamma_1(\log\frac{p(x,t)}{\pi(x)}, \log\frac{p(x,t)}{\pi(x)})\partial_t p(x,t)dx + 2\int(\nabla\log\frac{p(x,t)}{\pi(x)}, \nabla\log\frac{\partial_t p(x,t)}{\pi(x)})p(x,t)dx \\
&= \int \Gamma_1(\log\frac{p(x,t)}{\pi(x)}, \log\frac{p(x,t)}{\pi(x)})\partial_t p(x,t)dx + 2\int(\nabla\log\frac{p(x,t)}{\pi(x)}, \nabla\log\frac{\partial_t p(x,t)}{\pi(x)})p(x,t)dx \\
&= \int \Gamma_1(\log\frac{p(x,t)}{\pi(x)}, \log\frac{p(x,t)}{\pi(x)})\partial_t p(x,t)dx - 2\int\frac{1}{p(x,t)}\nabla\cdot(p(x,t)\nabla\log\frac{p(x,t)}{\pi(x)})\partial_t p(x,t)dx \\
&= -2\int \Gamma_2(\log\frac{p(x,t)}{\pi(x)}, \log\frac{p(x,t)}{\pi(x)})p(x,t)dx.
\end{aligned}$$

Remark. For the last equality, check [17] for details by taking $\gamma = 0$ to change the non-reversible case to reversible case.

(c) With (b) and assumption 2.3, one has

$$\begin{aligned}
\frac{d}{dt}\mathcal{I}(p(x,t)) &= -2\int \Gamma_2(\log\frac{p(x,t)}{\pi(x)}, \log\frac{p(x,t)}{\pi(x)})p(x,t)dx \\
&\le -2\lambda\int \Gamma_1(\log\frac{p(x,t)}{\pi(x)}, \log\frac{p(x,t)}{\pi(x)})p(x,t)dx \\
&= -2\lambda\mathcal{I}(p(x,t)|\pi(x)) \\
&= 2\lambda\frac{d}{dt}\mathcal{D}(p(x,t)|\pi(x)).
\end{aligned}$$

Note that

$$\begin{aligned}
-\mathcal{I}(p(x,t)) &= \int_t^\infty \frac{d}{ds}\mathcal{I}(p(x,s))ds \\
&\le -2\lambda\int_t^\infty \mathcal{I}(p(x,s))ds \\
&= -2\lambda\int_t^\infty -\frac{d}{ds}\mathcal{D}(p(x,s))ds \\
&= -2\lambda\mathcal{D}(p(x,t)).
\end{aligned}$$

Then it follows that by solving

$$\frac{d}{dt}\mathcal{I}(p(x,t)) \le -2\lambda\mathcal{I}(p(x,t)),$$

one gets

$$\mathcal{I}(p(x,t)) \le e^{-2\lambda}\mathcal{I}(p(x,0)).$$

The entropy dissipation follows

$$\begin{aligned}
\mathcal{D}(p(x,t)) &\le \frac{1}{2\lambda}\mathcal{I}(p(x,t)) \\
&\le \frac{1}{2\lambda}e^{-2\lambda}\mathcal{I}(p(x,0)).
\end{aligned}$$

Following the inequality between *KL-divergence* and L_1 distance, we have

$$\int \|p(x,t) - \pi(x)\|\,dx \le \sqrt{2\mathcal{D}(p(x,t))} \le \sqrt{\frac{1}{\lambda}\mathcal{I}(p(x,0))}e^{-\lambda t}.$$

C.2. **Euler angles and Quaternions of $SO(3)$.**

Euler angles is a method to parameterize the elements of $SO(3)$. Using coordinates (ψ, θ, ϕ), $\forall\ R \in SO(3)$, we could write

$$\begin{aligned} R &= R_z(\phi)R_y(\theta)R_x(\psi) \\ &= \begin{pmatrix} \cos\theta\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi \\ -\sin\theta & \sin\psi\cos\theta & \cos\psi\cos\theta \end{pmatrix}, \end{aligned}$$

where we have:
A rotation of ψ radians about the x-axis is defined as

$$R_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}.$$

Similarly, the rotation of θ radians about the y-axis is defined as

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}.$$

Lastly, the rotation of ϕ radians about the z-axis is defined

$$R_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Quaternion can encode the *axis-angle* representation of rotation matrix in four numbers. In 3-D space, according to *Euler's rotation theorem*, any sequence of rotations about a fixed point is equivalent to a single rotation by a given angle θ about a fixed axis (*Euler axis*).

In fact, the *Quaternion representation* shows that $P : S^3 \to SO(3)$ is a double covering and $P' : S^3/\{\pm 1\} \to SO(3)$ is a homeomorphism. For a rotation of angle θ about the normalized rotation axis $\vec{r}$, the quaternion representation is given by $q = \cos\frac{\theta}{2} + i\sin\frac{\theta}{2}x_r + j\sin\frac{\theta}{2}y_r + k\sin\frac{\theta}{2}z_r$, and we define its inverse by $q^{-1} = \cos\frac{\theta}{2} - (x_r i + y_r j + z_r k)\sin\frac{\theta}{2}$. For a general quaternion $q = q_r + q_i i + q_j j + q_k k$, and consider point $p \in \mathbb{R}^3$, we have the quaternion rotation $p' = qpq^{-1} = Rp$, where

$$R = \begin{bmatrix} 1 - 2(q_j^2 + q_k^2) & 2(q_i q_j - q_k q_r) & 2(q_i q_k + q_j q_r) \\ 2(q_i q_j + q_k q_r) & 1 - 2(q_i^2 + q_k^2) & 2(q_j q_k - q_i q_r) \\ 2(q_i q_k - q_j q_r) & 2(q_j q_k + q_i q_r) & 1 - 2(q_i^2 + q_j^2) \end{bmatrix}.$$

Note that all terms of R are of second-order w.r.t. q, and thus it is invariant up to the sign.

C.3.

Sample Mean and Covariance		
Sample	Mean	Covariance Matrix
(a)	$[-19.98, 20.02, 9.99]$	$\begin{pmatrix} 1.023 & -0.019 & 0.035 \\ -0.019 & 1.061 & 0.042 \\ 0.035 & 0.042 & 0.969 \end{pmatrix}$
(b)	$[-19.99, 20.06, 10.02]$	$\begin{pmatrix} 0.972 & 0.490 & 0.004 \\ 0.490 & 1.026 & -0.021 \\ 0.004 & -0.021 & 0.960 \end{pmatrix}$
(c)	$[79.96, 59.84, -30.08]$	$\begin{pmatrix} 8.999 & 0.063 & 0.006 \\ 0.063 & 9.958 & 7.761 \\ 0.006 & 7.761 & 6.603 \end{pmatrix}$
(d)	$[-20.02, 20.03, 10.07]$	$\begin{pmatrix} 1.120 & 0.007 & -0.073 \\ 0.007 & 0.984 & -0.008 \\ -0.073 & -0.008 & 0.983 \end{pmatrix}$
(e)	$[-20.05, 20.03, 10.03]$	$\begin{pmatrix} 1.064 & 0.515 & -0.056 \\ 0.515 & 1.007 & -0.029 \\ -0.056 & -0.029 & 1.009 \end{pmatrix}$
(f)	$[79.81, 59.16, -30.73]$	$\begin{pmatrix} 9.580 & 0.116 & -0.041 \\ 0.116 & 8.864 & 6.936 \\ -0.041 & 6.936 & 6.130 \end{pmatrix}$
(g)	$[-19.97, 20.00, 10.01]$	$\begin{pmatrix} 1.039 & -0.021 & 0.022 \\ -0.021 & 1.034 & -0.048 \\ 0.022 & -0.048 & 0.978 \end{pmatrix}$
(h)	$[-19.97, 19.99, 9.99]$	$\begin{pmatrix} 0.998 & 0.488 & -0.020 \\ 0.488 & 1.007 & -0.036 \\ -0.020 & -0.036 & 0.918 \end{pmatrix}$
(i)	$[80.03, 58.51, -31.19]$	$\begin{pmatrix} 9.111 & -0.038 & -0.065 \\ -0.038 & 8.739 & 6.810 \\ -0.065 & 6.810 & 5.868 \end{pmatrix}$

C.4.

Sample Mean and Covariance for Heisenberg Case		
Sample	Mean	Covariance Matrix
Target for (a)	$[0, 0, 0]$	$\begin{pmatrix} 1.0 & 0.9 & 0 \\ 0.9 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{pmatrix}$
(a)	$[0.04, 0.02, 0.05]$	$\begin{pmatrix} 1.051 & 0.925 & -0.012 \\ 0.925 & 1.008 & -0.020 \\ -0.012 & -0.020 & 0.928 \end{pmatrix}$
Target for (b)	$[2, 2, 2]$	$\begin{pmatrix} 1.0 & 0. & 0. \\ 0. & 1.0 & 0. \\ 0. & 0. & 1.0 \end{pmatrix}$
(b)	$[1.94, 1.95, 2.03]$	$\begin{pmatrix} 1.088 & -0.038 & -0.034 \\ -0.038 & 0.980 & -0.024 \\ -0.034 & -0.024 & 1.028 \end{pmatrix}$