Optimization on the Special Orthogonal Group

Milder, Jordy G.

Award date:
2023

Master Thesis

Optimization on the Special Orthogonal Group

June 12, 2023

Full Name	Student ID	Study
J.G. Milder	1022273	Industrial Applied Mathematics

TU/e Supervisor 1:	Olga Mula Hernández
TU/e Supervisor 2:	Jan ten Thije Boonkkamp

Eindhoven, June 12, 2023

Contents

1 Introduction

One fundamental challenge in robotics is finding a configuration for a robotic arm such that the end effector of the arm reaches a certain position $y \in \mathbb{R}^3$. An example of a robotic arm is given in Figure 1. This particular robotic arm has 6 axes of rotation, which are called joints. These joints are connected by links. To reach the position y, one needs to find the corresponding rotation angles of every joint of the robotic arm. This is called the inverse kinematics problem and is extensively studied within the field of robotics (see [10] and [16]).

Figure 1: The Stäubli robot. [15]

As an illustration, let us consider a robotic arm which only consists of one freely rotating joint and one link. Let the initial position of the end effector be described by the vector $x \in \mathbb{R}^3$. To reach the end position $y \in \mathbb{R}^n$ we need to find a $n \times n$ rotation matrix R, such that $Rx = y$. Here n denotes the dimensionality of the problem. Since R is a rotation matrix it belongs to the special orthogonal group $\mathrm{SO}(n)$, which is defined by

$$\mathrm{SO}(n) = \{R \in \mathrm{M}_n(\mathbb{R}) : R^T R = I_n, \det(R) = 1\}. \tag{1}$$

We can transform this inverse kinematics problem into an optimization problem by considering the objective function $f(R) = \|Rx - y\|_2$, which needs to be minimized for R over $\mathrm{SO}(n)$.

In this thesis, we dive into the subject of Riemannian geometry and Lie groups to develop an iterative algorithm which solves this optimization problem. This provides a powerful mathematical framework for understanding the geometry and structure of $\mathrm{SO}(n)$. One can opt to solve the problem by using classical gradient descent over the space of all real $n \times n$ matrices. However, at iteration step t the classical gradient descent method will not give an $R_t \in \mathrm{SO}(n)$ necessarily. Therefore, we will exploit the manifold structure of $\mathrm{SO}(n)$ to obtain a Riemannian optimization method, which gives after every iteration step t a feasible $R_t \in \mathrm{SO}(n)$. This allows for real-time updating of the robotic arm.

Furthermore, we present an innovative iterative algorithm that finds the configuration of a robotic arm such that it accurately follows a given curve $\gamma : \mathbb{R} \to \mathbb{R}^n$. This algorithm is also suitable for real-time updating of the robotic arm. Moreover, this algorithm is not restricted to finding a robotic arm configuration in $\mathbb{R}^2$ or $\mathbb{R}^3$ but can be used to find a configuration in a general multi-dimensional space $\mathbb{R}^n$, where n is finite.

In summary, this thesis dives into the field of Riemannian geometry to develop algorithms that can allow for real-time updating of the robotic arm. The main result of this thesis is an algorithm that can find the configuration of a robotic arm which accurately follows a given curve in n-dimensional space. This algorithm can find use in the field of robotics, where the robotic arm is restricted to follow a certain path, due to certain obstacles within the range of the arm.

1.1 Structure of the Report

This report has the following structure. Chapter 2 contains the theoretical background for the remainder of this report. This includes subjects from topology, Lie group theory and Riemannian geometry. In Chapter 3, we apply this theory to $\mathrm{SO}(n)$ to adjust the classical gradient descent method to the Riemannian setting.

The main result of this section is the constant line search method given by Algorithm 2. In Chapter 4, we test Algorithm 2 extensively and show numerical convergence. Furthermore, we develop a mathematical model for a robotic arm in $\mathbb{R}^n$ and use Algorithms 3 and 4 to find the configuration of the robotic arm which accurately follows a given curve. Lastly, Chapter 5 contains the conclusion and future research directions.

2 Manifolds, Tangent Spaces and Lie Algebra

In this chapter, we give the mathematical framework which will be useful when deriving the iterative optimization method on the special orthogonal group $\mathrm{SO}(n)$. The first three sections, Section 2.1, Section 2.2 and Section 2.3 introduce the concepts of matrix groups, topological spaces, manifolds and the differential map. Then, in Section 2.4, Section 2.5, Section 2.6 and Section 2.7, we look further into manifolds and their tangents space. We add a differential structure to a manifold to obtain a smooth manifold and we add a Riemannian metric to a smooth manifold to obtain a Riemannian manifold. Here, we will interpret $\mathrm{SO}(n)$ as a Riemannian submanifold of $\mathrm{M}_n(\mathbb{R})$. In Section 2.9 the matrix exponential is defined, which plays an important role when defining the Lie algebra of a matrix Lie group in the next section, Section 2.10. The final section, Section 2.11, is about differential equations on submanifolds. Here, we obtain a nice description of the tangent space of $\mathrm{SO}(n)$ in terms of its Lie algebra. This description of the tangent space is very useful for finding a good update matrix in the iterative optimization algorithm introduced in the next chapter.

Information in this chapter about topologies, manifolds and tangent spaces is from [2], [17], [18], [22]. Information about matrix groups, Lie groups and Lie algebras is from [6], [12], [13], [21].

2.1 Matrix groups and $\mathrm{SO}(n)$

This section is an introduction to matrix groups of which the matrix group $\mathrm{SO}(n)$ is an important example. First, let us give the general definition of a group and a subgroup,

Definition 2.1 (Group)**.** A group G is a set with a binary operator. Let us denote this binary operator as $*$. This operator combines two elements, say $a, b \in G$, such that $a * b \in G$. Moreover, the following group axioms must be satisfied

- associativity, i.e., $(a * b) * c = a * (b * c)$ for all $a, b, c \in G$.
- There exist a (unique) identity element $e \in G$, such that $e * a = a$ and $a * e = a$ for all $a \in G$.
- For every $a \in G$, there exists an inverse $a^{-1} \in G$ such that $a * a^{-1} = e$ and $a^{-1} * a = e$.

Definition 2.2 (Subgroup)**.** A subgroup H of a group G is a subset of G, such that

- $a * b \in H$ for all $a, b \in H$, so H is closed under its binary operator.
- for every $a \in H$, its inverse a^{-1} is also in H.

Note that the identity element e is always an element of the subgroup H since $a * a^{-1} = e$ must be in H. Moreover, note that a subgroup is also a group. Examples of some simple groups and subgroups are given below.

Example 2.1. (Groups)

- A simple example of a group is the set of all integers $\mathbb{Z}$ with the addition operator $+$. Here 0 is the identity element and an integer a has inverse $-a$. A subgroup of all the integers is the group of all even integers.
- The set of integers modulo a prime together with the multiplication operator $*$ is a group. This group can be denoted as $(\mathbb{Z}/p\mathbb{Z})^*$. Let $a \in (\mathbb{Z}/p\mathbb{Z})^*$, then the inverse a^{-1} of a can be calculated by using the extended Euclidean algorithm or by calculating a multiplication table. As an example of a modulo multiplicative group, let us consider $(\mathbb{Z}/7\mathbb{Z})^* = \{1, 2, 3, 4, 5, 6\}$. 1 is the identity element. To find the inverse pairs, we use the extended Euclidean algorithm.

$$\begin{aligned}
\gcd(7,2) = \gcd(2,1) = 1; \quad & 7 - 3 * 2 = 1 \mod 7, \\
& 4 * 2 = 1 \mod 7, \\
\gcd(7,3) = \gcd(3,1) = 1; \quad & 7 - 2 * 3 = 1 \mod 7, \\
& 5 * 3 = 1 \mod 7, \\
\gcd(7,6) = \gcd(6,1) = 1; \quad & 7 - 1 * 6 = 1 \mod 7 \\
& 6 * 6 = 1 \mod 7.
\end{aligned}$$

This results in table 1.

a	a^{-1}
2	4
3	5
4	2
5	3
6	6

Table 1: The inverse a^{-1} for each $a \in (\mathbb{Z}/7\mathbb{Z})^*$.

- The set of all invertible $n \times n$ matrices, which is denoted as $\mathrm{GL}(n, \mathbb{R})$, is a group under matrix multiplication. Associativity holds since $(AB)C = A(BC)$ for all $A, B, C \in \mathrm{GL}(n, \mathbb{R})$. Moreover, for every $A \in \mathrm{GL}(n, \mathbb{R})$, there exists an $A^{-1} \in \mathrm{GL}(n, \mathbb{R})$, for which $AA^{-1} = I_n$, where I_n is the $n \times n$ identity matrix, which is the identity element of the group $\mathrm{GL}(n, \mathbb{R})$.

Now, we are ready to give the definition of a matrix group.

Definition 2.3 (Matrix group)**.** Let G be any subgroup of $\mathrm{GL}(n, \mathbb{R})$. If a sequence of matrices $(A_i)_{i=0}^{\infty} \subseteq G$ converges to some matrix A (i.e., every entry of A_i converges to the corresponding entry of A as $i \to \infty$), and A is either an element of the subgroup G or A is not invertible, then G is called a matrix group.

In literature, matrix groups are sometimes called matrix Lie groups. Below, some examples of matrix groups are given of which one is $\mathrm{SO}(n)$.

Example 2.2. (Matrix Groups)

- The orthogonal group is defined as the set of all $n \times n$ orthogonal matrices, i.e.,

$$\mathrm{O}(n) = \{X \in \mathrm{M}_n(\mathbb{R}) : X^T X = I_n\}. \tag{2}$$

  Here $\mathrm{M}_n(\mathbb{R})$ denotes the set of all $n \times n$ matrices. $\mathrm{O}(n)$ is a subgroup of $\mathrm{GL}(n, \mathbb{R})$, since for every $X \in \mathrm{O}(n)$ there is an inverse $X^{-1} = X^T$, which is also an element of $\mathrm{O}(n)$. Moreover, the matrix product of two orthogonal matrices $Z = XY$, where $X, Y \in \mathrm{O}(n)$, is also orthogonal since

$$\begin{aligned} Z^T Z &= (XY)^T XY, \\ &= Y^T X^T XY, \\ &= Y^T Y = I_n. \end{aligned}$$

  Moreover the relation $X^T X = I_n$ is preserved under taking limits, therefore $O(n)$ is a matrix group.

- The special orthogonal group $\mathrm{SO}(n)$ is defined as the group of all $n \times n$ orthogonal matrices with determinant equal to 1. In other words, $\mathrm{SO}(n)$ is the group of all matrices which represent rotations in $\mathbb{R}^n$. The formal definition is given by

$$\mathrm{SO}(n) = \{X \in \mathrm{M}_n(\mathbb{R}) : X^T X = I_n, \det(X) = 1\}. \tag{3}$$

  $\mathrm{SO}(n)$ is also a subgroup of $\mathrm{O}(n)$ and therefore also a subgroup of $\mathrm{GL}(n, \mathbb{R})$. Like the property that $X^T X = I_n$ is preserved under taking limits, the property that $\det(X) = 1$ is too. Therefore, $\mathrm{SO}(n)$ is a matrix group.

- To show that not every subgroup of $\mathrm{GL}(n, \mathbb{R})$ is a matrix group, let us consider the group of all invertible $n \times n$ matrices with rational entries, denoted as $\mathrm{GL}(n, \mathbb{Q})$. Consider the following sequence

$(A_i)_{i=0}^{\infty}$ of matrices in GL$(2, (\mathbb{Q})$:

$$\begin{aligned}
A_0 &= \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \\
A_1 &= \begin{pmatrix} 3,1 & 0 \\ 0 & 1 \end{pmatrix}, \\
A_2 &= \begin{pmatrix} 3,14 & 0 \\ 0 & 1 \end{pmatrix}, \\
&\vdots \\
\lim_{i\to\infty} A_i &= \begin{pmatrix} \pi & 0 \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

For each element A_i of the sequence $(A_i)_{i=0}^{\infty}$, the upper-left entry is equal to an approximation of π up to i digits. For $i \to \infty$ this entry becomes equal to π. Note that $\lim_{i\to\infty} A_i$ is invertible. Therefore, for GL$(n, \mathbb{Q})$ to be a matrix group, we require that $\lim_{i\to\infty} A_i \in$ GL$(n, \mathbb{Q})$. However, this is clearly not the case.

2.2 Topological Spaces and Manifolds

In this section, the definition of a manifold is given. Before we arrive at this definition we need some basic concepts from the field of topology. First, consider the definition of a topological space below.

Definition 2.4 (Topological space)**.** Given is a set X. A topology $\mathcal{T}$ on X is a collection $\mathcal{T}$ of subsets of X satisfying the following properties:

- $\emptyset, X \in \mathcal{T}$.
- $\mathcal{T}$ is closed under finite intersections: if $U_1, ..., U_n$ are elements of $\mathcal{T}$, then their intersection $U_1 \cap \cdots \cap U_n$ is an element of $\mathcal{T}$.
- $\mathcal{T}$ is closed under arbitrary unions: if $(U_\alpha)_{\alpha\in A}$ is any (finite or infinite) family of elements of $\mathcal{T}$, then their union $\bigcup_{\alpha\in A} U_\alpha$ is an element of $\mathcal{T}$.

The elements of $\mathcal{T}$ are called open sets or open neighbourhoods. A set X together with a topology $\mathcal{T}$ on X is called a topological space, denoted as $(X, \mathcal{T})$ (see [17] and [23]).

Example 2.3. (Topology)

- $\mathcal{T} = \{\emptyset, X\}$ is a topology on the set X. This specific topology is also called the indiscrete topology of X.
- $\mathcal{T} = \mathcal{P}(X)$, where $\mathcal{P}(X)$ denotes the power set of X, is a topology on X. This specific topology is also called the discrete topology of X.
- Let $X = \{1, 2\}$, $\mathcal{T} = \{\emptyset, \{2\}, \{1, 2\}\}$ is a topology on X.
- Let $X = \mathbb{R}$, $\mathcal{T} = \{\emptyset, \mathbb{R}\} \cup \{\langle b, \rightarrow\rangle : b \in \mathbb{R}\}$ is a topology on X.
- Let $X \subseteq \mathbb{R}^n$. Let us define the Euclidean distance between two points $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$ as

  $$\begin{aligned}
  \text{dist}(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\| \\
  &= \sqrt{(x_1 - y_1)^2 + ... + (x_n - y_n)^2}.
  \end{aligned}$$

  Here, $\|\cdot\|$ denotes the 2-norm and x_i and y_i, $i \in \{1, ..., n\}$, are the coordinate elements of the vectors $\mathbf{x}$ and $\mathbf{y}$ respectively, when using the standard basis for $\mathbb{R}^n$. We define the standard topology $\mathcal{T}$ of X as the set of all open sets U for which we have that for all $\mathbf{x} \in U$, there exists an $\epsilon > 0$, such that all points in the set

  $$B_\epsilon(\mathbf{x}) = \{\mathbf{y} : \text{dist}(\mathbf{x}, \mathbf{y}) < \epsilon\}, \tag{4}$$

  are also in U, i.e., $B_\epsilon(\mathbf{x}) \subseteq U \in \mathcal{T}$. Here, $B_\epsilon(\mathbf{x})$ is called an open epsilon ball around $\mathbf{x}$.

Recall that we defined an open set U of a topological space $(X, \mathcal{T})$ as an element of a topology $\mathcal{T}$, i.e., $U \in \mathcal{T}$. In metric spaces, one can also define an open set by using open epsilon balls as we have seen in the last example. There, we gave the definition of the standard topology of a subset of $\mathbb{R}^n$. Note that this topology contains open sets U, where we can recognize these open sets in the way they are defined in metric spaces. Therefore, $X \subset \mathbb{R}^n$ can be regarded as an metric space with the Euclidean distance as a metric. In general, open sets when considering X as a metric space are also called open sets when considereing X as an topological space. The reverse argument is not necessarily true, since a metric on a topological space is not always defined.

Next, let us consider maps between topological spaces. This will give us the following notion of continuity.

Definition 2.5 (Continuous maps)**.** Consider two topological spaces $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$. A map $f : X \to Y$ is called continuous if for all $U \in \mathcal{T}_Y$: $f^{-1}[U] \in \mathcal{T}_X$. Here, $f^{-1}[U]$ denotes the pre-image of U, i.e., $f^{-1}[U] = \{x \in X : f(x) \in U\}$.

A special type of continuous map is a homeomorphism.

Definition 2.6 (Homeomorphism)**.** Consider two topological spaces $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$. A map $f : X \to Y$ is called a homeomorphism if f is bijective (one-to-one), continuous and if its inverse $f^{-1} : Y \to X$ is also continuous.

Now we arrive at the notion of locally Euclidean topological spaces.

Definition 2.7 (Locally Euclidean and charts)**.** A topological space $(X, \mathcal{T})$ is locally Euclidean of dimension n if, for all $a \in X$, there exists an open set (also called an open neighbourhood) $U_a \in \mathcal{T}$, for which $a \in U_a$ and there exists a homeomorphism $\varphi_a : U_a \to U'_a$, with $U'_a \subseteq \mathbb{R}^n$ open. φ_a is called a chart of $(X, \mathcal{T})$, which one can denote as (U_a, φ_a).

So, for a topological space $(X, \mathcal{T})$, which is locally Euclidean, we have for every point $a \in X$ an open neighbourhood U_a, such that (U_a, φ_a) forms a chart. A collection of charts, of which their corresponding open neighbourhoods covers X is called an atlas, i.e.,

Definition 2.8 (Atlas)**.** A collection of charts $\mathcal{A} = \cup_i \{(U_i, \varphi_i)\}$, $i \in \{1, 2, ...\}$, of the set X is called an atlas of X if $\cup_i U_i = X$.

Now, we have all the mathematical tools to give the definition of a manifold, which generalizes the concept of surfaces in space.

Definition 2.9 (n-manifold)**.** The topological space $(\mathcal{M}, \mathcal{T})$, or simply $\mathcal{M}$, is a manifold of dimension n if it has the following properties:

- M is a Hausdorff space: for every pair of distinct points $p, q \in \mathcal{M}$, there are disjoint open neighbourhoods $U, V \in \mathcal{T}$, such that $p \in U$ and $q \in V$.
- M is second-countable: there exists a countable basis for the topology $\mathcal{T}$ of M.
- M is locally Euclidean of dimension n. (See Definition 2.7).

A manifold can be denoted as $(\mathcal{M}, \mathcal{T})$ or simply as the set $\mathcal{M}$. Recall that a topological space can be denoted as a couple $(X, \mathcal{T})$, where X is a set and $\mathcal{T}$ is a topology on X. Denoting a manifold as $(\mathcal{M}, \mathcal{T})$ corresponds to the notion of a manifold as a topological space. For simplicity, we will denote a manifold as the set $\mathcal{M}$ and not state the topology explicitly, except if stated otherwise.

Example 2.4. (Manifolds) Consider the following examples of a manifold.

- Consider a finite number of points $\mathcal{M}$ and its discrete topology $\mathcal{T} = \mathcal{P}(\mathcal{M})$. One can easily verify that $(\mathcal{M}, \mathcal{T})$ is Hausdorff and that the set of the individual points forms a basis for $\mathcal{T}$. Therefore, $(\mathcal{M}, \mathcal{T})$ is second-countable. Moreover, for every point, let us consider a map φ which maps the point to 0. These charts are homeomorphisms and therefore $(\mathcal{M}, \mathcal{T})$ is a 0-dimensional manifold.
- Consider a subset $\mathcal{M}$ of $\mathbb{R}^n$, i.e., $\mathcal{M} \subseteq \mathbb{R}^n$. $\mathbb{R}^n$ with the standard topology (see Example 2.3) has the property that it is Hausdorff and second-countable [18]. Since $\mathcal{M}$ is a subspace of $\mathbb{R}^n$, $\mathcal{M}$ is also

Hausdorff and second-countable. As a chart, consider the identity map, which is a homeomorphism. Therefore, $\mathcal{M}$ is an n-dimensional manifold.

- Any n-dimensional vector space $\mathcal{V}$ is a manifold [2]. Given a basis $(\mathbf{e}_i)_{i=1}^n$ of $\mathcal{V}$. The map $\psi : \mathcal{V} \to \mathbb{R}^n$,

$$\psi(\mathbf{x}) = (x^1, x^2, ..., x^n)^T,$$

such that $\mathbf{x} = \sum_{i=1}^n x^i \mathbf{e}_i$, is a chart of $\mathcal{V}$.

- Consider the n-dimensional sphere, which is defined as

$$S^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \|\mathbf{x}\| = 1\},$$

where $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$. For simplicity, let us take $n = 2$, S^2 is a sphere in $\mathbb{R}^3$ with radius 1. We will show that S^2 is a two-dimensional manifold. S^2 is Hausdorff and second-countable since it is a topological subspace of $\mathbb{R}^3$. Now, let us define the following open sets on S^2.

$$\begin{aligned} U_{i,+} &= \{\mathbf{x} \in S^2 : x_i > 0\}, \quad \text{for } i \in \{1,2,3\}, \\ U_{i,-} &= \{\mathbf{x} \in S^2 : x_i < 0\}, \quad \text{for } i \in \{1,2,3\}, \end{aligned}$$

where $\mathbf{x} = (x_1, x_2, x_3)^T$. These $U_{i,\pm}$'s are hemispheres. For example, $U_{3,+}$ is the northern hemisphere and $U_{3,-}$ is the southern hemisphere. Next, let us introduce an open subset of $\mathbb{R}^2$.

$$\widetilde{U} = \{\widetilde{\mathbf{x}} \in \mathbb{R}^2 : \|\widetilde{\mathbf{x}}\| < 1\}.$$

Here, $\widetilde{\mathbf{x}} = (\widetilde{x}_1, \widetilde{x}_2)^T$. Note that $U_{i,\pm} \subseteq S^2$ for all $i \in \{1,2,3\}$ and $\widetilde{U} \subseteq \mathbb{R}^2$. Let us introduce the maps $h_{i,+} : U_{i,+} \to \widetilde{U}$ and $h_{i,-} : U_{i,-} \to \widetilde{U}$ for all $i \in \{1,2,3\}$. For $i = 1$, these maps and their inverses are given by

$$\begin{aligned} h_{1,+}(\mathbf{x}) &= (x_2, x_3), \\ h_{1,+}^{-1}(\widetilde{\mathbf{x}}) &= (\sqrt{1 - \|\widetilde{\mathbf{x}}\|}, \widetilde{x}_1, \widetilde{x}_2)^T, \\ h_{1,-}(\mathbf{x}) &= (x_2, x_3), \\ h_{1,-}^{-1}(\widetilde{\mathbf{x}}) &= (-\sqrt{1 - \|\widetilde{\mathbf{x}}\|}, \widetilde{x}_1, \widetilde{x}_2)^T, \end{aligned}$$

and likewise for $i = 2, 3$. First note that the $h_{i,\pm}$'s and the $h_{i,\pm}^{-1}$'s are continuous and bijective. Therefore the $h_{i,\pm}$'s are homeomorphisms, so $(U_{i,+}, h_{i,+})$ and $(U_{i,-}, h_{i,-})$ for all $i \in \{1,2,3\}$ are charts of S^2. Furthermore, note that $\mathcal{A} = \cup_{i \in \{1,2,3\}} \{(U_{i,+}, h_{i,+}), (U_{i,-}, h_{i,-})\}$ is an atlas of S^2. This shows that S^2 is locally Euclidean of dimension two and therefore a two-dimensional manifold. In general, one can show that any n-dimensional sphere S^n in $\mathbb{R}^{n+1}$ is an n-dimensional manifold, following the same steps as given above (see [18]).

In this last example of S^2, we constructed charts for S^2, where a chart is a double containing an open set $U \subseteq S^2$ and a homeomorphism h. Then we find a collection of charts of which its open sets cover S^2. So, this collection of charts is an atlas for S^2. Therefore, S^2 is locally Euclidean and together with the Hausdorff and second-countable property we can conclude that S^2 is a manifold.

In the remainder of this chapter, we broaden our understanding of manifolds by introducing manifolds with special properties, like smooth manifolds and Riemannian manifolds.

2.3 The Differential

Definition 2.10 (Differential). Let $\mathcal{E}$ and $\mathcal{F}$ be two finite-dimensional normed vector spaces over $\mathbb{R}$. A function $F : \mathcal{E} \to \mathcal{F}$ is (Fréchet)-differentiable at a point $x \in \mathcal{E}$ if there exists a linear operator

$$\mathrm{D}F(\mathbf{x}) : \mathcal{E} \to \mathcal{F},$$

which maps $\mathbf{h} \in \mathcal{E}$ to $\mathrm{D}F(\mathbf{x})[\mathbf{h}] \in \mathcal{F}$, such that

$$F(\mathbf{x}+\mathbf{h}) = F(\mathbf{x}) + \mathrm{D}F(\mathbf{x})[\mathbf{h}] + o(\|\mathbf{h}\|), \tag{5}$$

in other words,

$$\lim_{\mathbf{h}\to\mathbf{0}} \frac{\|F(\mathbf{x}+\mathbf{h}) - F(\mathbf{x}) - \mathrm{D}F(\mathbf{x})[\mathbf{h}]\|}{\|\mathbf{h}\|} = 0.$$

The linear operator $\mathrm{D}F(\mathbf{x})$ is called the differential and the element $\mathrm{D}F(\mathbf{x})[\mathbf{h}]$ is called the directional derivative of F at $\mathbf{x}$ along $\mathbf{h}$. The function $F : \mathcal{E} \to \mathcal{F}$ is said to be differentiable on an open domain $\Omega \subseteq \mathcal{E}$ if F is differentiable at every point $\mathbf{x} \in \Omega$. [2]

The differential has some nice properties. We have the chain rule

$$\mathrm{D}(F \circ G)(\mathbf{x}) = \mathrm{D}F(G(\mathbf{x})) \circ \mathrm{D}G(\mathbf{x}),$$

in other words,

$$\mathrm{D}(F \circ G)(\mathbf{x})[\mathbf{h}] = \mathrm{D}F(G(\mathbf{x}))[\mathrm{D}G(\mathbf{x})[\mathbf{h}]].$$

The differential $\mathrm{D}F(\mathbf{x})$ belongs to the vector space $\mathcal{L}(\mathcal{E};\mathcal{F})$ of all linear operators from $\mathcal{E}$ to $\mathcal{F}$. So, we can see $\mathrm{D}F$ as a map $\mathrm{D}F : \mathcal{E} \to \mathcal{L}(\mathcal{E};\mathcal{F})$. If $\mathrm{D}F$ is continuous on an open domain Ω, then we say that F is continuously differentiable on Ω.

Let $(\mathbf{e}_1, ..., \mathbf{e}_m)$ and $(\mathbf{e}'_1, ..., \mathbf{e}'_n)$ be bases for $\mathcal{E}$ and $\mathcal{F}$ respectively. Let $\hat{F} : \mathbb{R}^m \to \mathbb{R}^n$ be the expression for F in terms of these bases, in other words,

$$F\left(\sum_{i=1}^{m} x^i \mathbf{e}_i\right) = \sum_{j=1}^{n} \hat{F}^j(\mathbf{x})\mathbf{e}'_j,$$

where the x^i's are the coordinates of $\mathbf{x}$, with respect to the the basis $(\mathbf{e}_1, ..., \mathbf{e}_m)$ and $\hat{F}^j(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ are the coordinates of $F(\mathbf{x})$ with respect to the basis $(\mathbf{e}'_1, ..., \mathbf{e}'_n)$. Then, we have the following theorem.

Theorem 2.1 (Continuous differentiability via partial derivatives)**.** *A function $F : \mathcal{E} \to \mathcal{F}$ is continuously differentiable if and only if the partial derivatives $\partial_i \hat{F}^j$ of $\hat{F}$ exist and are continuous. Then we have*

$$\mathrm{D}F(\mathbf{x})[\mathbf{h}] = \sum_{j=1}^{n}\sum_{i=1}^{m} \partial_i \hat{F}^j(\mathbf{x}) h^i \mathbf{e}'_j. \tag{6}$$

Here, the h^i's are the coordinates of $\mathbf{h}$, with respect to the basis $(\mathbf{e}_1, ..., \mathbf{e}_m)$. It can be shown that this expression does not depend on the chosen bases (see appendix of [2]).

Example 2.5. (Differential)

- Let $f : \mathbb{R}^2 \to \mathbb{R}$ and let $\{\mathbf{e}_1, \mathbf{e}_2\}$ be the canonical basis of $\mathbb{R}^2$

$$f(x_1, x_2) = x_1^2 + \frac{x_2^2}{2},$$

  Using Definition 2.10, we obtain

$$\begin{aligned} f(x_1+h_1, x_2+h_2) &= (x_1+h_1)^2 + \frac{1}{2}(x_2+h_2)^2 \\ &= (x_1^2 + 2x_1h_1) + \left(\frac{x_2^2}{2} + x_2h_2\right) + o(h_1) + o(h_2). \end{aligned}$$

  Therefore,

$$\mathrm{D}f(x)[h] = 2x_1h_1 + x_2h_2$$

Alternatively, one can use Equation (6), to obtain

$$\mathrm{D}f(\mathbf{x})[\mathbf{h}] = \langle \nabla f(\mathbf{x}), \mathbf{h} \rangle_2,$$

where $\nabla f(\mathbf{x})$ denotes the Euclidean gradient of f at $\mathbf{x}$, $\mathbf{h} = (h_1, h_2)^T$ and $\langle \cdot, \cdot \rangle_2$ denotes the standard inner product. Note that the differential map $\mathrm{D}f : \mathbb{R}^2 \to \mathcal{L}(\mathbb{R}^2, \mathbb{R})$ can be expressed as

$$\mathrm{D}f = (\nabla f(\mathbf{x}))^T.$$

- Let $F : \mathbb{R}^2 \to \mathbb{R}^3$, be defined as

$$F(x_1, x_2) = x_1^2 \mathbf{e}_1 + 2x_1x_2\mathbf{e}_2 + x_2^2\mathbf{e}_3,$$

where $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ is the canonical basis of $\mathbb{R}^3$. Using the standard bases for $\mathbb{R}^2$ and $\mathbb{R}^3$ and Equation (6) we obtain

$$\mathrm{D}F(x_1, x_2)[\mathbf{h}] = \begin{pmatrix} 2x_1 & 0 \\ 2x_2 & 2x_1 \\ 0 & 2x_2 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}.$$

Note that the differential map $\mathrm{D}F : \mathbb{R}^2 \to \mathcal{L}(\mathbb{R}^2; \mathbb{R}^3)$ is given by

$$\mathrm{D}F = \begin{pmatrix} 2x_1 & 0 \\ 2x_2 & 2x_1 \\ 0 & 2x_2 \end{pmatrix}$$

Note that $\mathrm{D}F$ is equal to the Jacobian matrix.

- Let $F(X) = \mathrm{tr}(X)$, where $X \in \mathrm{M}_n(\mathbb{R})$. Note that

$$\begin{aligned} F(X+H) &= \mathrm{tr}(X+H), \\ &= \mathrm{tr}(X) + \mathrm{tr}(H), \\ &= F(X) + \mathrm{tr}(H). \end{aligned}$$

Therefore, $\mathrm{D}F(X)[H] = \mathrm{tr}(H)$.

- Let $F(X) = X^{-1}$, where X is an invertible matrix. Let H be a bounded matrix, which is small in the operator norm $|\cdot|$, such that $|H| < \frac{1}{|X^{-1}|}$. Then,

$$\begin{aligned} F(X+H) &= (X+H)^{-1} \\ &= (X(I + X^{-1}H))^{-1} \\ &= (I + X^{-1}H)^{-1}X^{-1} \\ &= \sum_{k=0}^{\infty} (-1)^k (X^{-1}H)^k X^{-1} \\ &= X^{-1} - X^{-1}HX^{-1} + ... \\ &= F(X) - X^{-1}HX^{-1} + ... \end{aligned}$$

Here, I is the identity matrix. Note that we used the Neumann series

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k,$$

which holds for $|A| < 1$. We left out the terms which contains H to the power of 2 or higher. We can conclude that $\mathrm{D}F(X)[H] = -X^{-1}HX^{-1}$.

2.4 Smooth Maps and Smooth Manifolds

The definition of a manifold in the previous section is sufficient for studying the topological properties of manifolds, like compactness, connectedness and simple connectivity [18]. Unfortunately, this is not enough for doing optimization over manifolds. In the next chapter, we will introduce a real-valued loss function over SO(n), which is a map between two manifolds, SO(n) and $\mathbb{R}$. We will minimize this loss function over SO(n). Therefore, we need to make sense of derivatives of real-valued functions between manifolds. For this, we need a special kind of manifold called a smooth manifold. This will help us decide if a map to or from a manifold is smooth.

In the previous section, we constructed an atlas for the two-dimensional manifold S^2. The six charts in this atlas contain the six open sets $U_{i,\pm}$ ($i \in \{1,2,3\}$). Note that some open sets partly overlap. For example, $U_{1,+} \cap U_{3,+} = \{\mathbf{x} \in S^2 : x_1 > 0, x_3 > 0\} \neq \emptyset$. So we have two (or even more) homeomorphisms $h_{1,+}$ and $h_{3,+}$ that map the points in $U_{1,+} \cap U_{3,+}$ to a subset of $\mathbb{R}^2$ ($\widetilde{U}$). Therefore, we can construct a so-called transition map $h_{3,+} \circ h_{1,+}^{-1} : \widetilde{U} \to \widetilde{U}$. In general, a transition map is defined as follows.

Definition 2.11 (Transition Map)**.** Let $\mathcal{M}$ be an n-dimensional manifold. If (U,φ), (V,ψ) are two charts such that $U \cap V \neq \emptyset$, the composite map $\psi \circ \varphi^{-1} : \varphi(U \cap V) \to \psi(U \cap V)$ is called the transition map from φ to ψ.

Note that the transition map is a composition of homeomorphisms and therefore is itself a homeomorphism. The transition map $\psi \circ \varphi^{-1}$ is illustrated in Figure 2.

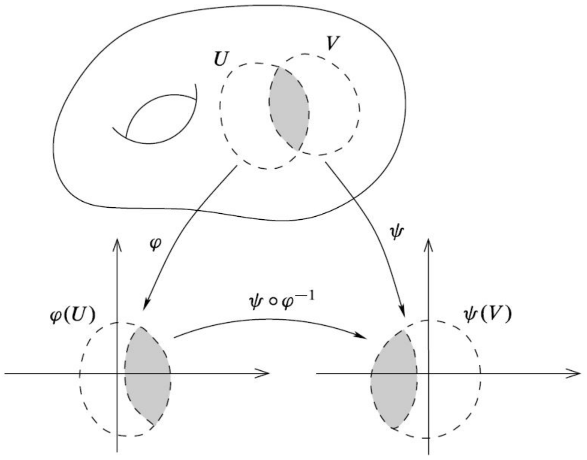


Figure 2: Illustration of the transition map from φ to ψ. Here, (U,φ) and (V,ψ) are charts. φ maps the open set U onto $\mathbb{R}^n$ and ψ maps the open set V onto $\mathbb{R}^n$. The transition map $\psi \circ \varphi^{-1}$ is a map from the image of φ ($\varphi(U)$) to the image of ψ ($\psi(V)$) (see [18]
).

We will say that the two charts (U,φ) and (V,ψ) are smoothly compatible if the transition map $\psi \circ \varphi^{-1}$ is a diffeomorphism, where a diffeomorphism is defined as follows:

Definition 2.12 (diffeomorphism)**.** A smooth map, which is bijective and has a smooth inverse, is called a diffeomorphism (see [18]).

Note that a diffeomorphism is also a homeomorphism.

Recall that an atlas of $\mathcal{M}$ is a collection of charts whose domains cover $\mathcal{M}$. Now consider the definitions of a smooth atlas and a maximal smooth atlas.

Definition 2.13 (Smooth Atlas)**.** An atlas $\mathcal{A}$ for a manifold $\mathcal{M}$ is called a smooth atlas if any two charts in $\mathcal{A}$ are smoothly compatible with each other (see [18]).

Definition 2.14 (Maximal Smooth Atlas or Smooth Structure)**.** Let $\mathcal{A}^+$ be a smooth atlas of the manifold $\mathcal{M}$. $\mathcal{A}^+$ is called a maximal smooth atlas or smooth structure on $\mathcal{M}$ if any chart that is smoothly compatible with every chart in $\mathcal{A}^+$ is already in $\mathcal{A}^+$ (see [18]), [2]

Now we have everything to give the definition of a smooth manifold.

Definition 2.15 (n-dimensional Smooth Manifold)**.** An n-dimensional manifold $\mathcal{M}$ is called an n-dimensional smooth manifold if $\mathcal{M}$ has a maximal smooth atlas.

A smooth manifold is often denoted as $(\mathcal{M}, \mathcal{A}^+)$, where $\mathcal{M}$ is the smooth manifold and $\mathcal{A}^+$ is its corresponding smooth structure. In Ref. [18] it is noted that a smooth manifold could have many different smooth structures. We will not dive into further details here, since for us it is only interesting if such a smooth structure does exist.

In general, it is very inconvenient to define a smooth structure by explicitly describing a maximal smooth atlas on a given manifold $\mathcal{M}$. Fortunately, the following lemma helps us a lot.

Lemma 2.2. *Let $\mathcal{M}$ be a manifold. Every smooth atlas $\mathcal{A}$ for $\mathcal{M}$ is contained in a unique maximal smooth atlas (see [18]).*

So, to prove that a manifold $\mathcal{M}$ is also a smooth manifold, we only have to find a smooth atlas $\mathcal{A}$ for $\mathcal{M}$. Then, by using Lemma 2.2, we can conclude that there exists a maximal smooth atlas for $\mathcal{M}$ containing $\mathcal{A}$. Therefore, we can conclude that $\mathcal{M}$ has a smooth structure, i.e., $\mathcal{M}$ is a smooth manifold.

Next, let us consider smooth maps between smooth manifolds. The definition of smooth maps between smooth manifolds is based on the definition of smooth maps between Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^m$.

Definition 2.16 (Smooth maps between Euclidean spaces)**.** Let U and V be open subsets of the Euclidean spaces $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, respectively. A function $f : U \to V$ is called infinitely differentiable, C^∞, or smooth if each of its component functions has continuous partial derivatives of all orders (see [18]).

Now, let us jump back to manifolds. Let F be a function from a smooth manifold $\mathcal{M}_1$ of dimension d_1 to a smooth manifold $\mathcal{M}_2$ of dimension d_2. Consider a point x on $\mathcal{M}_1$, which is mapped by F to a point $F(x)$ on $\mathcal{M}_2$. We can choose a chart φ_1 around x and a chart φ_2 around $F(x)$. We can define a smooth function between two manifolds in a similar way as we did for a function between Euclidean spaces. Given these two charts φ_1 and φ_2, the function F can be read through the charts, i.e., we can define a new function $\hat{F} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$

$$\hat{F} = \varphi_2 \circ F \circ \varphi_1^{-1}.$$

$\hat{F}$ is called the coordinate representation of F. The coordinate representation $\hat{F}$ is schematically illustrated in Figure 3.

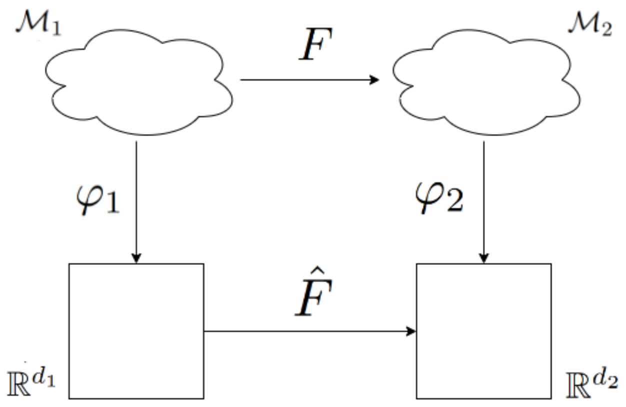


Figure 3: Illustration of the coordinate representation $\hat{F} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ of a map $F : \mathcal{M}_1 \to \mathcal{M}_2$. Here, $\mathcal{M}_1$ and $\mathcal{M}_2$ are manifolds of dimension d_1 and d_2, respectively. $\varphi_1 : \mathcal{M}_1 \to \mathbb{R}^{d_1}$ is a chart of $\mathcal{M}_1$ and $\varphi_2 : \mathcal{M}_2 \to \mathbb{R}^{d_2}$ is a chart of $\mathcal{M}_2$.

This gives us the following definitions.

Definition 2.17 (Smooth maps between manifolds)**.** Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be smooth manifolds. A function $F : \mathcal{M}_1 \to \mathcal{M}_2$ is infinitely differentiable or smooth at x if $\hat{F} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ is smooth at $\varphi_1(x)$. A function $F : \mathcal{M}_1 \to \mathcal{M}_2$ is said to be smooth if it is smooth at every point of its domain.

Definition 2.18 (Diffeomorphisms between manifolds)**.** $F : \mathcal{M}_1 \to \mathcal{M}_2$ is a diffeomorphism if and only if it is a bijection such that F and its inverse F^{-1} are both smooth.

Note that we can take the differential of $\hat{F}$, since $\hat{F}$ is function between two vector spaces, namely $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$. Therefore, we can define the differentiability of F in terms of the differentiability of $\hat{F}$. The following definition of the rank of a function F will become useful in the next section when determining the dimension of manifolds.

Definition 2.19. (Rank) Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be smooth manifolds of dimension d_1 and d_2, respectively, and let $F : \mathcal{M}_1 \to \mathcal{M}_2$ be a smooth map. The rank of F at $x \in \mathcal{M}_1$ is the dimension of the range of $\mathrm{D}\hat{F}(\varphi_1(x))[\cdot] : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, where $\hat{F}$ is the coordinate representation of F.

Next, let us consider some examples of smooth manifolds.

Example 2.6. (Smooth manifolds)

- $\mathbb{R}^n$ is a smooth manifold since the one chart $(\mathbb{R}^n, \mathrm{Id})$ already forms a smooth atlas for $\mathbb{R}^n$. Here Id is the identity map, which is clearly a diffeomorphism. By using Lemma 2.2, we can conclude that there exists a maximal smooth atlas for $\mathbb{R}^n$. Therefore, $\mathbb{R}^n$ is a smooth manifold.
- In this example we prove that S^2 is a smooth manifold by showing that every pair of charts from the atlas $\mathcal{A} = \cup_{i\in\{1,2,3\}}\{(U_{i,+}, h_{i,+}), (U_{i,-}, h_{i,-})\}$ is smoothly compatible. These charts and this atlas were already defined in Example 2.4.

  Proof. Consider an arbitrary pair of charts from the atlas $\mathcal{A}$: $(U_{i,\pm}, h_{i,\pm})$ and $(U_{j,\pm}, h_{j,\pm})$ for arbitrary $i, j \in \{1, 2, 3\}$. The transition map $w_{i\pm,j\pm} : h_{i,\pm}(U_{i,\pm} \cap U_{j,\pm}) \to h_{j,\pm}(U_{i,\pm} \cap U_{j,\pm})$ defined as $w_{i\pm,j\pm} = h_{j,\pm} \circ h^{-1}_{i,\pm}$, is a smooth map. To see this, consider for example the transition map $w^* = w_{3+,1+} = h_{1,+} \circ h^{-1}_{3,+}$ and let $\widetilde{\mathbf{x}} = (\widetilde{x}_1, \widetilde{x}_2)^T$ be an arbitrary point in $h_{3,+}(U_{3,+} \cap U_{1,+}) \subseteq \widetilde{U}$. Then, we define w^* as

$$w^* : \begin{pmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{pmatrix} \xrightarrow{h^{-1}_{3,+}} \begin{pmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \\ \sqrt{1 - \|\widetilde{\mathbf{x}}\|^2} \end{pmatrix} \xrightarrow{h_{1,+}} \begin{pmatrix} \widetilde{x}_2 \\ \sqrt{1 - \|\widetilde{\mathbf{x}}\|^2} \end{pmatrix}.$$

Note that $\|\widetilde{\mathbf{x}}\| < 1$, since $\widetilde{\mathbf{x}}$ lies somewhere in $\widetilde{U}$ (see example 2.4). Therefore, w^* is a smooth map. This holds in general for every transition map $w_{i\pm,j\pm}$. Since,

$$\begin{aligned}(w_{i\pm,j\pm})^{-1} &= \left(h_{j,\pm} \circ h_{i,\pm}^{-1}\right)^{-1} \\ &= h_{i,\pm} \circ h_{j,\pm}^{-1} = w_{j\pm,i\pm},\end{aligned}$$

we can conclude that all the inverses of $w_{i\pm,j\pm}$ are smooth maps too. Therefore all the transition maps are diffeomorphisms and therefore $\mathcal{A}$ is a smooth atlas for S^2. By using Lemma 2.2 we can conclude that there exists a maximal smooth atlas for S^2. Therefore S^2 is a smooth manifold. □

- Consider an n-dimensional vector space $\mathcal{V}$. In Example 2.4, we have seen that $\mathcal{V}$ is a manifold. Let $(\mathbf{e}_i)_{i=1,...,n}$ be a basis for $\mathcal{V}$. Consider the map $E : \mathbb{R}^n \to \mathcal{V}$

$$E(\mathbf{x}) = \sum_{i=1}^{n} x^i \mathbf{e}_i.$$

This map is a homeomorphism, so $(\mathcal{V}, E^{-1})$ is a chart. This chart covers the whole vector space. So we have already found a smooth atlas for $\mathcal{V}$ and can conclude that $\mathcal{V}$ is a smooth manifold by using Lemma 2.2. There could be other bases for $\mathcal{V}$, let $(\widetilde{\mathbf{e}_j})_{j=1,...,n}$ be such a basis. Then, there is a corresponding homeomorphism $\widetilde{E} : \mathbb{R}^n \to \mathcal{V}$

$$\widetilde{E}(\mathbf{x}) = \sum_{j=1}^{n} \widetilde{x}^j \widetilde{\mathbf{e}_j}.$$

If such a basis exists, there exists some invertible $n \times n$ matrix (A_i^j) such that

$$\mathbf{e}_i = \sum_{j=1}^{n} A_i^j \widetilde{\mathbf{e}_j},$$

for each $i \in \{1, ..., n\}$. This gives us another chart $(\mathcal{V}, \widetilde{E}^{-1})$. Let us check if this chart is indeed smoothly compatible with the chart $(\mathcal{V}, E^{-1})$. The transition map $w : \mathbb{R}^n \to \mathbb{R}^n$ between the two charts is $w : \widetilde{E}^{-1} \circ E$, i.e.

$$w : \mathbf{x} \xrightarrow{E} \sum_{i=1}^{n} x^i \mathbf{e}_i \xrightarrow{\widetilde{E}^{-1}} \widetilde{\mathbf{x}}.$$

Here,

$$\begin{aligned}\sum_{j=1}^{n} \widetilde{x}^j \widetilde{\mathbf{e}_j} &= \sum_{i=1}^{n} x^i \mathbf{e}_i \\ &= \sum_{i=1}^{n} x^i \sum_{j=1}^{n} A_i^j \widetilde{\mathbf{e}_j} \\ &= \sum_{j=1}^{n} \sum_{i=1}^{n} x^i A_i^j \widetilde{\mathbf{e}_j}.\end{aligned}$$

From this, we can conclude that

$$\widetilde{x}^j = \sum_{i=1}^{n} x^i A_i^j.$$

So the transition map w, sending $\mathbf{x}$ to $\widetilde{\mathbf{x}}$, is an invertible linear map and therefore a diffeomorphism. So if there exist such a pair of charts, $(\mathcal{V}, \widetilde{E}^{-1})$ and $(\mathcal{V}, E^{-1})$, these charts are indeed smoothly compatible.

The space of all real $m \times n$ matrices is a finite-dimensional real vector space under matrix addition and scalar multiplication. From Example 2.6, we can conclude that this vector space is a smooth manifold. In particular, the space of all square real matrices $\mathrm{M}_n(\mathbb{R})$ is an $n \times n$-dimensional real vector space and therefore an n^2-dimensional smooth manifold. Starting from the next section, we will assume that every manifold is a smooth manifold unless stated otherwise.

2.5 Submanifolds

In this section, we introduce the concept of embedded submanifolds. We prove that $\mathrm{SO}(n)$ is a submanifold of $\mathrm{M}_n(\mathbb{R})$, from which we can conclude that $\mathrm{SO}(n)$ is a (smooth) manifold. First, let us consider the definition of an (embedded) submanifold (see [2]), [12].

Definition 2.20 ((embedded) submanifold)**.** A subset $\mathcal{M}$ of an n-dimensional manifold $\overline{\mathcal{M}}$ is a d-dimensional embedded submanifold of $\overline{\mathcal{M}}$ ($d \leq n$) if and only if, around each point $x \in \mathcal{M}$, there exists a chart (U, φ) of $\overline{\mathcal{M}}$ such that $\mathcal{M} \cap \overline{\mathcal{M}} = \mathcal{M}$ is a φ-coordinate slice of U, i.e.,

$$\mathcal{M} = \{x \in U : \varphi(x) \in \mathbb{R}^d \times \{0\}_{n-d}\}.$$

The chart $(\mathcal{M} \cap \overline{\mathcal{M}}, \varphi)$, where φ is seen as a mapping into $\mathbb{R}^d$, is a chart of the embedded submanifold $\mathcal{M}$. When $\mathcal{M}$ is a submanifold of $\overline{\mathcal{M}}$, $\overline{\mathcal{M}}$ is called the embedding space.

There also exists the notion of immersed submanifold, which is not necessarily an embedded manifold. More details can be found in Ref. [18]. In this report, we will simply call an embedded submanifold a submanifold.

Example 2.7. (S^2 as a submanifold of $\mathbb{R}^3$)

- Consider the manifold $\mathbb{R}^3$ and the chart $\psi : \mathbb{R}^3 \to \mathbb{R}^3$ defined as $\psi(x_1, x_2, x_3) \to (r - 1, \theta, \varphi)$, where

$$r = \sqrt{x_1^2 + x_2^2 + x_3^2},$$
$$\theta = \arccos\left(\frac{x_3}{r}\right),$$
$$\varphi = \operatorname{sgn}(x_2) \arccos\left(\frac{x_1}{\sqrt{x_1^2 + x_2^2}}\right)$$

are polar coordinates, which are illustrated in Figure 4.

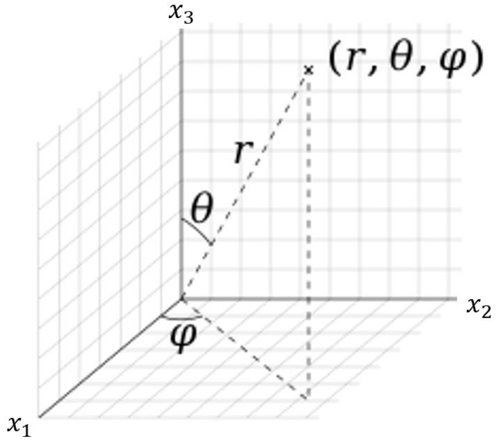


Figure 4: Spherical coordinates r, θ, φ in $\mathbb{R}^3$.

For S^2, we have $r = 1$, since S^2 is a sphere of radius 1 in $\mathbb{R}^3$. So, for the image of S^2, denoted as $\psi[S^2]$, we have

$$\psi[\mathbb{R}^3 \cap S^2] = \psi[S^2] = [0, \pi] \times [-\pi, \pi] \times \{0\}_1.$$

Therefore, S^2 is a two-dimensional submanifold of $\mathbb{R}^3$.

Functions defined on embedded submanifolds pose no particular difficulty regarding smoothness. A submanifold is itself also a (smooth) manifold. Furthermore, let $\mathcal{M}$ be a submanifold of $\overline{\mathcal{M}}$ and let F be a smooth function on $\overline{\mathcal{M}}$. Then, the restriction of F on $\mathcal{M}$, $F|_{\mathcal{M}}$, is a smooth function on $\mathcal{M}$. The following lemma is very useful when identifying submanifolds, without explicitly finding the φ-coordinate slices.

Lemma 2.3. *(Subsets of manifolds) Let $\mathcal{M}$ be a subset of a manifold $\overline{\mathcal{M}}$. Then $\mathcal{M}$ admits at most one smooth structure that makes it an embedded submanifold of $\overline{\mathcal{M}}$ (see [2]).*

We have seen that $\mathrm{M}_n(\mathbb{R})$ is a manifold. Since $\mathrm{SO}(n) \subset \mathrm{M}_n(\mathbb{R})$, by using Lemma 2.3 we can conclude that $\mathrm{SO}(n)$ is a submanifold of $\mathrm{M}_n(\mathbb{R})$. One can use the following theorem to obtain the dimension of a submanifold, without using φ-coordinate slices (see [2]).

Theorem 2.4 (submersion theorem)**.** *Let $F : \mathcal{M}_1 \to \mathcal{M}_2$ be a smooth mapping between two manifolds of dimension d_1 and d_2, respectively, where $d_1 > d_2$. Moreover, let y be a point of $\mathcal{M}_2$. If y is a regular value of F, i.e., the rank of $\mathrm{D}F(y)$ is equal to d_2 at every point of $F^{-1}(y)$, then $F^{-1}(y)$ is a closed embedded submanifold of $\mathcal{M}_1$, with dimension $d_1 - d_2$.*

The submersion theorem is schematically illustrated in Figure 5.

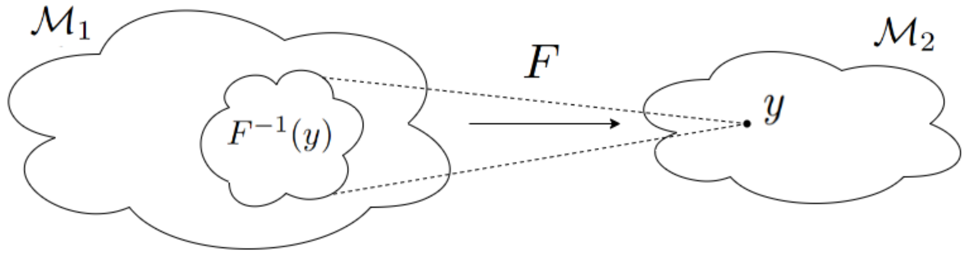


Figure 5: Illustration of the submersion theorem, which states that $F^{-1}(y)$ is an embedded submanifold of $\mathcal{M}_1$ if the rank of F is equal to the dimension of $\mathcal{M}_2$ for all point in $F^{-1}(y)$.

Example 2.8. (Submersion theorem)

- Consider the set of $n \times n$ orthogonal matrices $\mathrm{O}(n)$,

$$\mathrm{O}(n) = \{X \in \mathrm{M}_n(\mathbb{R}) : X^T X = I_n\},$$

  where I_n denotes the $n \times n$ identity matrix. Clearly, $\mathrm{O}(n)$ is a subset of $\mathrm{M}_n(\mathbb{R})$. In the previous section, we found that $\mathbb{R}^m$ is a smooth manifold for some arbitrary finite value of m. Since there is a one-to-one correspondence of $\mathrm{M}_n(\mathbb{R})$ with $\mathbb{R}^{n^2}$ (one can simply rewrite the elements of an $n \times n$ matrix into one vector of dimension n^2), we can conclude that $\mathrm{M}_n(\mathbb{R})$ is also a smooth manifold. Therefore, by Lemma 2.3 $\mathrm{O}(n)$ is submanifold of $\mathrm{M}_n(\mathbb{R})$.
  We can also prove that $\mathrm{O}(n)$ is a submanifold of $\mathrm{M}_n(\mathbb{R})$ by using the submersion theorem. Then, we also obtain the dimension of $\mathrm{O}(n)$. Consider the smooth map $F : \mathrm{M}_n(\mathbb{R}) \to \mathcal{S}_{\mathrm{sym}}(n)$; $F(X) = X^T X - I_n$, where $\mathcal{S}_{\mathrm{sym}}(n)$ denotes the set of all $n \times n$ symmetric matrices. Note $\mathrm{M}_n(\mathbb{R})$ and $\mathcal{S}_{\mathrm{sym}}(n)$ are both vector spaces. Therefore, there is no need to read F through the charts. Also note that $\mathrm{O}(n) = F^{-1}(0_n)$, where $0_n \in \mathcal{S}_{\mathrm{sym}}(n)$ is the $n \times n$ zero matrix. Next, let us calculate the differential of F.

$$\begin{aligned} F(X + H) &= (X + H)^T (X + H) - I_n, \\ &= X^T X - I_n + X^T H + H^T X + ..., \\ &= F(X) + X^T H + H^T X + ... \end{aligned}$$

  Therefore, $\mathrm{D}F(X)[H] = X^T H + H^T X$. It remains to show that for all $\hat{H} \in \mathcal{S}_{\mathrm{sym}}(n)$, there exists a

$H \in \mathrm{M}_n(\mathbb{R})$, such that $\mathrm{D}F(X)[H] = \hat{H}$, where $X \in F^{-1}(0_n)$. Let $H = \frac{1}{2}X\hat{H}$, then

$$\begin{aligned}\mathrm{D}F(X)[H] &= \mathrm{D}F(X)[\frac{1}{2}X\hat{H}]\\ &= \frac{1}{2}X^T X\hat{H} + (\frac{1}{2}X\hat{H})^T X\\ &= \frac{1}{2}\hat{H} + \frac{1}{2}\hat{H}^T X^T X\\ &= \frac{1}{2}\hat{H} + \frac{1}{2}\hat{H} = \hat{H}.\end{aligned}$$

Here, we used that $X^T X = I_n$, since $X \in \mathrm{O}(n)$ and that $\hat{H} = \hat{H}^T$, since $\hat{H} \in \mathcal{S}_{\mathrm{sym}}(n)$. So we can conclude that $\mathrm{O}(n)$ is a submanifold of $\mathrm{M}_n(\mathbb{R})$. The dimension of $\mathrm{O}(n)$ is equal to the dimension of $\mathrm{M}_n(\mathbb{R})$ minus the dimension of $\mathcal{S}_{\mathrm{sym}}(n)$. Therefore,

$$\begin{aligned}\dim(\mathrm{O}(n)) &= \dim(\mathrm{M}_n(\mathbb{R})) - \dim(\mathcal{S}_{\mathrm{sym}}(n)),\\ &= n^2 - \frac{1}{2}n(n+1),\\ &= \frac{1}{2}n(n-1).\end{aligned}$$

- $\mathrm{O}(n)$ consist of two components, matrices X for which $\det(X) = 1$ and for which $\det(X) = -1$. The first component is $\mathrm{SO}(n)$, which is also a subgroup of $\mathrm{O}(n)$. $\mathrm{SO}(n)$ has the same dimension as $\mathrm{O}(n)$, i.e., the dimension of $\mathrm{SO}(n)$ is equal to $\frac{1}{2}n(n-1)$.

2.6 Tangent Spaces

In this section, the definition of the tangent space of an embedded submanifold is given, where the embedding space is a vector space. An example of these embedded submanifolds is S^2, which is an embedded submanifold of the vector space R^3. Other examples are matrix manifolds like $\mathrm{SO}(n)$ and $\mathrm{O}(n)$, which are submanifolds of the vector space $\mathrm{M}_n(\mathbb{R})$. It is important to note that there also exists a general definition of the tangent space of a manifold (which is not necessarily an embedded submanifold of a vector space). This general definition is given in Ref. [2] and Ref. [18].

Let us consider a one-dimensional manifold in $\mathbb{R}^n$, which is a regular parametric curve $\gamma : I \to \mathbb{R}^n$. The tangent at $a = \gamma(0)$ is a straight line given by $\tau : I \to \mathbb{R}^n$, $\tau(t) = a + tv$, where $v = \gamma'(0) \neq 0$. Shifting the origin to the point a, the tangent space at a is the linear space

$$T_a\mathcal{M} = \{tv : t \in \mathbb{R}\}.$$

As an example, let $\mathcal{M}$ be described by $\gamma : [-\pi, \pi] \to \mathbb{R}^3$, where γ is the unit circle in the x_1x_2-plane, i.e.,

$$\gamma(\theta) = \begin{pmatrix}\cos(\theta)\\ \sin(\theta)\\ 0\end{pmatrix}.$$

So, $\gamma(0) = a = (1,0,0)^T$, then the tangent vector is $\gamma'(0) = (0,1,0)^T$, therefore

$$T_a\mathcal{M} = \{(0,t,0) : t \in \mathbb{R}\}.$$

Note that in the original variables, the tangent space is given by the affine space $a + T_a\mathcal{M}$. For multi-dimensional manifolds, the tangent space should also become multi-dimensional. As an example consider the tangent space of the unit sphere $\mathcal{M}$ in $\mathbb{R}^3$ at $a = (1,0,0)^T$. Now, there are infinitely many continuously differentiable paths γ over $\mathcal{M}$ for which $\gamma(0) = a$, with different values for $v = \gamma'(0)$. One can choose again γ_1 to be the unit circle in the x_1x_2-plane, then $\gamma_1'(0) = (0,1,0)^T$. However, if one chooses the path γ_2 to be the unit circle in the x_1x_3-plane, then $\gamma_2'(0) = (0,0,1)^T$. These two tangent vectors span the tangent space, so

$$T_a\mathcal{M} = \{(0,t_1,t_2) : t_1, t_2 \in \mathbb{R}\}. \qquad (7)$$

The following definition defines the tangent space of a submanifold in terms of these continuously differentiable paths through a.

Definition 2.21 (tangent space)**.** Let $\mathcal{M} \subset \overline{\mathcal{M}}$ be a submanifold of the vector space $\overline{\mathcal{M}}$ and let $a \in \mathcal{M}$. The tangent space of $\mathcal{M}$ at a is the space given by

$$T_a\mathcal{M} = \left\{ v \in \overline{\mathcal{M}} : \begin{array}{c} \exists \epsilon > 0, \exists \gamma : (-\epsilon, \epsilon) \to \overline{\mathcal{M}} \text{ continuoulsy differentiable, such that} \\ \gamma(s) \in \mathcal{M} \text{ for } s \in (-\epsilon, \epsilon),\, \gamma(0) = a \text{ and } \gamma'(0) = v. \end{array} \right\}.$$

Elements of $T_a\,\mathcal{M}$ are called tangent vectors of $\mathcal{M}$ at x. $T_a\,\mathcal{M}$ has the important property that it is a vector space and that its dimension is equal to the dimension of $\mathcal{M}$ (see [2]).

The tangent bundle $T\mathcal{M}$ is defined as the collection of all tangent spaces, i.e.,

Definition 2.22 (tangent bundle)**.** Let $\mathcal{M}$ be a submanifold of the embedding vector space $\overline{\mathcal{M}}$. The tangent bundle $T\mathcal{M}$ of the submanifold $\mathcal{M}$ is given by

$$T\,\mathcal{M} = \{v \in \overline{\mathcal{M}} : \exists x \in \mathcal{M}, \text{ such that } v \in T_x\,\mathcal{M}\}.$$

When the manifold $\mathcal{M}$ is (globally or locally) defined as the level set of a constant-rank function $F : \overline{\mathcal{M}} \to \mathbb{R}^n$, where $\overline{\mathcal{M}}$ is the embedding vector space, we have

Theorem 2.5 (tangent space through the level set representation of a submanifold)**.** *Consider a submanifold $\mathcal{M} \subset \overline{\mathcal{M}}$ of dimension k and let $a \in \mathcal{M}$. Furthermore, let $\mathcal{M}$ be locally given by $U \cap \mathcal{M} = \{x \in U : F(x) = \mathbf{0}\}$, where $F : U \to \mathbb{R}^n$ is a constant-rank function, then*

$$T_a\mathcal{M} = \ker\left(\mathrm{D}F(a)[\cdot]\right) = \{v \in \overline{\mathcal{M}} : \mathrm{D}F(a)[v] = 0\}. \tag{8}$$

The proof of this theorem can be found in Ref. [2] and Ref. [12].

Example 2.9. (Tangent spaces)

- Consider the unit sphere S^{n-1}, which is a submanifold of the vector space $\mathbb{R}^n$. Consider the function $F : \mathbb{R}^n \to \mathbb{R}$ Defined as

  $$F(x) = x^T x - 1.$$

  $\mathrm{D}F(x)[h] = x^T h + h^T x$. Note that $F(x)$ is of constant-rank equal to 1, since for an arbitrary $\hat{h} \in \mathbb{R}$, we have $h = \frac{1}{2}x\hat{h}$, such that

  $$\begin{aligned} \mathrm{D}F(x)[\frac{1}{2}x\hat{h}] &= \frac{1}{2}x^T x\hat{h} + \frac{1}{2}(x\hat{h})^T x \\ &= \hat{h}. \end{aligned}$$

  Therefore, we can use Theorem 2.5 to obtain

  $$\begin{aligned} T_x S^{n-1} &= \ker(\mathrm{D}F(x)[\cdot]) \\ &= \{v \in \mathbb{R}^n : x^T v + v^T x = 0\} \\ &= \{v \in \mathbb{R}^n : x^T v = 0\} \end{aligned} \tag{9}$$

  Earlier, we concluded that the tangent space of S^2 at $(1, 0, 0)^T$ can be described by Equation (7). Note that this expression coincides with Equation (9), when we let $n = 3$ and $x = (1, 0, 0)^T$.

- Next, lets us consider the orthogonal group $\mathrm{O}(n)$. In Example 2.8, we have seen that $\mathrm{O}(n)$ can be described by

  $$\mathrm{O}(n) = \{X : F(X) = 0_n\},$$

where $F : \mathrm{M}_n(\mathbb{R}) \to \mathcal{S}_{sym}(\mathbb{R})$, defined as $F(X) = X^T X - I_n$. $F(X)$ is of constant rank equal to $\frac{1}{2}n(n+1)$ and

$$DF(X)[H] = X^T H + H^T X,$$

so

$$\begin{aligned} T_X \mathrm{O}(n) &= \ker(DF(X)[\cdot]) \\ &= \{V \in \mathrm{M}_n(\mathbb{R}) : X^T V + V^T X = 0_{n\times n}\}. \end{aligned} \tag{10}$$

In particular the tangent space of $\mathrm{O}(n)$ at the identity element is defined as

$$T_I \mathrm{O}(n) = \{\Omega \in \mathrm{M}_n(\mathbb{R}) : \Omega + \Omega^T = 0_{n\times n}\}. \tag{11}$$

Note that $T_I\mathrm{O}(n)$ is the space of all $n \times n$ skew-symmetric matrices $\mathcal{S}_{\text{skew}}(n)$.

Next, let us propose an alternative characterization of $T_X\mathrm{SO}(n)$, which is also given in Example 3.5.2 and 3.5.3 of Ref. [2]. Let X_0 be an element of $\mathrm{O}(n)$ and let $t \to X(t)$ be a curve in $\mathrm{O}(n)$, through X_0 at $t = 0$, i.e., $X(0) = X_0$. Since $X(t) \in \mathrm{O}(n)$, we have

$$\begin{aligned} X^T(t)X(t) &= I_{n\times n} \\ \dot{X}^T(t)X(t) + X^T(t)\dot{X}(t) &= 0_{n\times n} \end{aligned} \tag{12}$$

Since $X(t) \in \mathrm{O}(n)$, $X(t)$ is of full rank, therefore $\dot{X}(t)$ can be written as

$$\dot{X}(t) = X(t)\Omega. \tag{13}$$

Substitution of Equation (13) into Equation (12) gives

$$\begin{aligned} \Omega^T X^T(t)X(t) + X^T(t)X(t)\Omega &= 0_{n\times n} \\ \Omega^T + \Omega &= 0_{n\times n} \end{aligned}$$

Therefore, $\dot{X}(t) = X(t)\Omega$, where $\Omega \in \mathcal{S}_{\text{skew}}(n)$ is a skew-symmetric matrix. In example 2.8 we found that the dimension of $\mathrm{O}(n)$ is equal to $\frac{1}{2}n(n-1)$. Since the dimension of the tangent space is equal to the dimension of the manifold and X is of full rank and the space of all skew-symmetric matrices $\mathcal{S}_{\text{skew}}(n)$ is also an $\frac{1}{2}n(n-1)$ dimensional space, we can conclude that the tangent space of $\mathrm{O}(n)$ at a point X can be written as

$$T_X \mathrm{O}(n) = \{X\Omega : \Omega^T + \Omega = 0_{n\times n}\}, \tag{14}$$

Which is an alternative way to describe the tangent space of $\mathrm{O}(n)$ as the expression given in Equation (10). Note that for any matrix $Y \in \mathrm{M}_n(\mathbb{R})$ and $\Omega \in \mathcal{S}_{\text{skew}}(n)$ we have

$$Y\Omega = (Y^T\Omega^T)^T = (-Y^T\Omega)^T = \Omega Y.$$

In particular, we have $X\Omega = \Omega X$ for $X \in \mathrm{O}(n)$. Therefore, we can also write

$$T_X \mathrm{O}(n) = \{\Omega X : \Omega^T + \Omega = 0_{n\times n}\}.$$

- As we have already discussed in the previous section, $\mathrm{SO}(n)$ is one of the two connected components of $\mathrm{O}(n)$, therefore the tangent space $T_R\,\mathrm{SO}(n)$ at a point $R \in \mathrm{SO}(n)$ can also be described by Equation (14), i.e.,

$$\boxed{T_R\,\mathrm{SO}(n) = \{R\Omega : \Omega \in \mathcal{S}_{\text{skew}}(n)\} = \{\Omega R : \Omega \in \mathcal{S}_{\text{skew}}(n)\}.} \tag{15}$$

2.7 Riemannian Manifolds

In the previous two sections, we described ways to find and describe the tangent space $T_x\,\mathcal{M}$ at a point x of submanifolds $\mathcal{M}$. It is important to have a good representation of the tangent space of a submanifold since tangent vectors generalize the notion of direction derivatives. (See next chapter). Therefore these tangent vectors will be very useful when optimizing a function $F : \overline{\mathcal{M}} \to \mathbb{R}$ over an embedded submanifold $\mathcal{M}$ of the embedding vector space $\overline{\mathcal{M}}$. To characterize which direction of motion from x produces the steepest increase or decrease in the function value, we need a notion of length that applies to tangent vectors. This can be done by endowing the tangent space with an inner product $\langle\cdot,\cdot\rangle_x$, which is a bilinear, symmetric positive-definite form. This inner product induces a norm

$$\|v\| = \sqrt{\langle v, v\rangle_x},$$

where $v \in T_x\,\mathcal{M}$. Note that one can define various inner products for various tangent spaces. This dependency of the inner product $\langle\cdot,\cdot\rangle_x$ on the tangent space $T_x\,\mathcal{M}$ is denoted by using a subscript x.

Definition 2.23 (Riemannian manifold and Riemannian metric)**.** Consider a manifold $\mathcal{M}$, whose tangent spaces are endowed with a smoothly varying inner product $\langle\cdot,\cdot\rangle_x$. Then $\mathcal{M}$ is called a Riemannian manifold and $\langle\cdot,\cdot\rangle_x$ is called the Riemannian metric of $\mathcal{M}$ (see [2]).

Here, smoothly varying means that the inner product smoothly changes with respect to the chosen tangent spaces, i.e., $\langle\cdot,\cdot\rangle_x$ smoothly changes with respect to x.

Example 2.10. (Riemannian manifold).

- Let us consider the space of all $n \times n$ matrices $\mathrm{M}_n(\mathbb{R})$. The tangent space of $\mathrm{M}_n(\mathbb{R})$ is simply $\mathrm{M}_n(\mathbb{R})$ itself. Let us endow the tangent space of $\mathrm{M}_n(\mathbb{R})$, with the Frobenius inner product $\langle\cdot,\cdot\rangle_{\mathrm{F}} : \mathrm{M}_n(\mathbb{R}) \times \mathrm{M}_n(\mathbb{R}) \to \mathbb{R}$ defined as

$$\langle X, Y\rangle_{\mathrm{F}} = \mathrm{tr}(X^T Y). \tag{16}$$

  Note that this inner product does not depend on which tangent space $T_X\,\mathrm{M}_n(\mathbb{R})$ is chosen. $\mathrm{M}_n(\mathbb{R})$ together with the Frobinius inner products forms a Riemannian manifold.

2.8 Riemannian submanifolds

$\mathrm{SO}(n)$ can be considered as an embedded submanifold of the Riemannian manifold $\mathrm{M}_n(\mathbb{R})$. In general, let us denote the submanifold as $\mathcal{M}$ and the embedding Riemannian manifold as $\overline{\mathcal{M}}$. One would expect that $\mathcal{M}$ can inherit a Riemannian metric from $\overline{\mathcal{M}}$ in a natural way. Let us denote the Riemannian metric in the embedding space $\overline{\mathcal{M}}$ as $\overline{g}(X,Y)$. Since every tangent space $T_R\mathcal{M}$ can be regarded as a subspace of $T_R\,\overline{\mathcal{M}}$, the Riemannian metric $\bar{g}$ of $\overline{\mathcal{M}}$ induces a Riemannian metric g on $\mathcal{M}$ according to

$$\boxed{g_R(X,Y) = \overline{g}(X,Y), \quad \forall X, Y \in T_R\mathcal{M},} \tag{17}$$

where X and Y are viewed as elements of $T_R\overline{\mathcal{M}}$. Endowed with this Riemannian metric, the embedded submanifold $\mathcal{M}$ becomes a Riemannian manifold. $\mathcal{M}$ is called the Riemannian submanifold of $\overline{\mathcal{M}}$.

Next, we take a closer look at elements in the tangent space $T_R\,\overline{\mathcal{M}}$ and how they can be related to elements in $T_R\,\mathcal{M}$. Therefore, we need the concept of an orthogonal projection.

Definition 2.24 (Orthogonal projection and orthogonal complement)**.** Let $\overline{\mathcal{V}}$ be a vector space endowed with an inner product $\langle\cdot,\cdot\rangle$ and let $\mathcal{V}$ be a linear subspace of $\overline{\mathcal{V}}$. An orthogonal projection $\mathrm{P}_{\mathcal{V}}x$ of a vector $x \in \overline{\mathcal{V}}$ onto $\mathcal{V}$ is a vector that satisfy two conditions:

1. $\mathrm{P}_{\mathcal{V}}x \in \mathcal{V}$
2. $\langle x - \mathrm{P}_{\mathcal{V}}x, y\rangle = 0$, for all $y \in \mathcal{V}$.

The orthogonal complement $\mathcal{V}^\perp$ of $\mathcal{V}$ is defined as

$$\mathcal{V}^\perp = \{x \in \overline{\mathcal{V}} : \langle x, y\rangle = 0, \quad \forall y \in \mathcal{V}\}. \tag{18}$$

Every element in $x \in \overline{\mathcal{V}}$ can be uniquely decomposed into the sum of an element in $\mathcal{V}$ and an element in $\mathcal{V}^{\perp}$, i.e.,

$$x = \mathrm{P}_{\mathcal{V}} x + \mathrm{P}_{\mathcal{V}^{\perp}} x.$$

[20]

Since $T_R \mathcal{M}$ is a linear subspace of the vector space $T_R \overline{\mathcal{M}}$, we can uniquely decompose any $X \in T_R \overline{\mathcal{M}}$ into

$$X = \mathrm{P}_{T_R \mathcal{M}} X + \mathrm{P}_{(T_R \mathcal{M})^{\perp}} X, \tag{19}$$

and we can define the orthogonal complement $(T_R \mathcal{M})^{\perp}$ of $T_R \mathcal{M}$ as

$$(T_R \mathcal{M})^{\perp} = \{X \in T_R \overline{\mathcal{M}} : \overline{g}(X, Y) = 0, \quad \forall Y \in T_R \mathcal{M}\}.$$

$(T_R \mathcal{M})^{\perp}$ is also called the normal space to $\mathcal{M}$ at R.

Example 2.11. (The normal space of SO(n))

- In section 2.5, we have seen that SO(n) is a submanifold of the embedding space $\mathrm{M}_n(\mathbb{R})$. We choose the Frobenius norm (see Equation (16)) as our Riemannian metric for $\mathrm{M}_n(\mathbb{R})$. According to Equation (17), SO(n) inherits the Riemannian metric from the embedding space $\mathrm{M}_n(\mathbb{R})$. Since $\mathrm{tr}(S^T \Omega) = 0$ for all $S \in S_{\mathrm{sym}}(n), \Omega \in \mathcal{S}_{\mathrm{skew}}(n)$, the normal space of SO$(n)$ at R is given by

$$(T_R \,\mathrm{SO}(n))^{\perp} = N_R \,\mathrm{SO}(n) = \{XS : S \in S_{\mathrm{sym}}(n)\}, \tag{20}$$

  where $S_{\mathrm{sym}}(n)$ denotes the space of all $n \times n$ symmetric matrices.

To conclude this section, let us consider the following theorem, which characterizes the tangent space of the product space of two Riemannian submanifolds of $\mathbb{R}^n$.

Theorem 2.6. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be submanifolds of $\mathbb{R}^n$. Let $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ be the product spade of $\mathcal{M}_1$ and $\mathcal{M}_2$. Then,*

$$T_{(x_1,x_2)} \mathcal{M} = T_{x_1} \mathcal{M}_1 \times T_{x_2} \mathcal{M}.$$

Proof. This proof consists of two parts. First we will show that $T_{x_1} \mathcal{M}_1 \times T_{x_2} \mathcal{M}_2 \subseteq T_{(x_1,x_2)} \mathcal{M}$. Then we will show that $T_{(x_1,x_2)} \mathcal{M} \subseteq T_{x_1} \mathcal{M}_1 \times T_{x_2} \mathcal{M}_2 \mathcal{M}$.
Part 1.
Let $v_1 \in T_{x_1} \mathcal{M}$ and $v_2 \in T_{x_2} \mathcal{M}$ be chosen arbitrarily. From Definition 2.21, we have

$$v_1 \in T_{x_1} \mathcal{M} \Leftrightarrow \exists \epsilon_1, \exists \gamma_1(s_1), s_1 \in (-\epsilon_1, \epsilon_1) \text{ such that } \gamma_1(0) = x_1 \text{ and } \gamma_1'(0) = v_1$$

and

$$v_2 \in T_{x_2} \mathcal{M} \Leftrightarrow \exists \epsilon_2, \exists \gamma_2(s_2), s_2 \in (-\epsilon_2, \epsilon_2) \text{ such that } \gamma_2(0) = x_2 \text{ and } \gamma_2'(0) = v_2.$$

Let us define $\epsilon = \min\{\epsilon_1, \epsilon_2\}$ and let us construct the path $\gamma(s)$ over $\mathcal{M}$ as follows

$$\gamma(s) = (\gamma_1(s), \gamma_2(s)), \quad \text{for } s \in (-\epsilon, \epsilon).$$

Note that $\gamma(s) \in \mathcal{M}$ and that

$$\gamma(0) = (\gamma_1(0), \gamma_2(0)) = (x_1, x_2)$$

and

$$\gamma'(0) = (\gamma'(0), \gamma'(0)) = (v_1, v_2).$$

Therefore, $(v_1, v_2) \in T_{(x_1,x_2)}\mathcal{M}$, so we can conclude that $T_{x_1}\mathcal{M}_1 \times T_{x_2}\mathcal{M}_2 \subseteq T_{(x_1,x_2)}\mathcal{M}$.
Part 2.
Let $v \in T_{(x_1,x_2)}\mathcal{M}$ be chosen arbitrarily. From Definition 2.21, we have

$$v \in T_{(x_1,x_2)}\mathcal{M} \Leftrightarrow \exists \epsilon, \exists \gamma(s), s \in (-\epsilon, \epsilon), \text{ such that } \gamma(0) = (x_1, x_2) \text{ and } \gamma'(0) = v.$$

Note that $\gamma(s)$ can be writen as

$$\gamma(s) = (\mathrm{P}_{\mathcal{M}_1}\gamma(s), \mathrm{P}_{\mathcal{M}_2}\gamma(s)),$$

where $\mathrm{P}_{\mathcal{M}_1}$ and $\mathrm{P}_{\mathcal{M}_2}$ denotes the orthognal projection onto $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. We define $\gamma_1(s) = \mathrm{P}_{\mathcal{M}_1}\gamma(s)$ and $\gamma_2(s) = \mathrm{P}_{\mathcal{M}_2}\gamma(s)$. So,

$$\gamma(s) = (\gamma_1(s), \gamma_2(s)).$$

Therefore, we have

$$\gamma(0) = (\gamma_1(0), \gamma_2(0)) = (x_1, x_2)$$

and

$$\gamma'(0) = (\gamma_1'(0), \gamma_1'(0)) = v.$$

Note that $\gamma_1'(0) \in T_{x_1}\mathcal{M}$ and $\gamma_2'(0) \in T_{x_2}\mathcal{M}$, therefore $v \in T_{x_1}\mathcal{M}_1 \times T_{x_2}\mathcal{M}_2$. So we can conclude that $T_{(x_1,x_2)}\mathcal{M} \subseteq T_{x_1}\mathcal{M}_1 \times T_{x_2}\mathcal{M}$. This result together with the result from part 1 proves this theorem. □

2.9 The Matrix Exponential

This section is about the matrix exponential, which is useful when we discuss Lie algebras that describe tangent spaces, in the next section. Moreover, calculating the matrix exponential will be an important step in our optimization algorithms.

The definition of the matrix exponential is given below.

Definition 2.25 (matrix exponential)**.** Let $X \in \mathrm{M}_n(\mathbb{R})$. The matrix exponential is a map $\exp : \mathrm{M}_n(\mathbb{R}) \to \mathrm{M}_n(\mathbb{R})$ given by

$$\exp(X) = \sum_{m=0}^{\infty} \frac{X^m}{m!}. \tag{21}$$

Example 2.12. (Matrix exponential)

- As an example, let us calculate the matrix exponential of the following matrix

$$X = \begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix}, \tag{22}$$

$$X = \theta J, \tag{23}$$

  where $\theta \in \mathbb{R}$ and J is the symplectic matrix given by

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

  Computing the powers of X gives

$$\begin{aligned} X^2 &= \theta^2 J^2 \\ &= -\theta^2 I_2, \\ X^3 &= \theta^3 J^3 \\ &= -\theta^3 J, \\ X^4 &= \theta^4 J^4 \\ &= \theta^4 I_2, \\ &\vdots \end{aligned}$$

So,

$$\begin{aligned}
\exp(X) &= I_2 + X + \frac{1}{2!}X^2 + \frac{1}{3!}X^3 + \frac{1}{4!}X^4 + ... \\
&= I_2 + \theta J - \frac{1}{2!}\theta^2 I_2 - \frac{1}{3!}\theta^3 J^3 + \frac{1}{4!}\theta^4 I_2 + ... \\
&= (1 - \frac{1}{2!}\theta^2 + \frac{1}{4!}\theta^4 - ...)I_2 + (\theta - \frac{1}{3!}\theta^3 + ...)J \\
&= \cos(\theta) I_2 + \sin(\theta) J \\
&= \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.
\end{aligned}$$

In conclusion, the matrix exponential maps a real 2×2 skew-symmetric matrix to the space of SO(2). Below some properties of the matrix exponential are given.

Lemma 2.7. *Let $X, Y \in \mathrm{M}_n(\mathbb{R})$, the matrix exponential has the following properties:*

1. *The matrix exponential $\exp(X)$ is a continuous function of X.*
2. $\exp(0_n) = I_n$, *where I_n is the $n \times n$ identity matrix and 0_n is the $n \times n$ zero matrix.*
3. $(\exp(X))^T = \exp(X^T)$.
4. $\exp(X)$ *is invertible and* $(\exp(X))^{-1} = \exp(-X)$. *Therefore,* $\exp$ *is a map from* $\mathrm{M}_n(\mathbb{R})$ *to* $\mathrm{GL}(n, \mathbb{R})$.
5. $\exp((\alpha + \beta)X) = \exp(\alpha X)\exp(\beta X)$ *for all* $\alpha, \beta \in \mathbb{R}$.
6. *If $XY = YX$, then* $\exp(X)\exp(Y) = \exp(Y)\exp(X) = \exp(X + Y)$.
7. *If C is invertible, then* $\exp(CXC^{-1}) = C\exp(X)C^{-1}$.
8. $\|\exp(X)\| \leq \exp(\|X\|)$, *where $\|X\|$ is the Frobenius norm defined as* $\sqrt{\sum_{k,l=1}^{n} |X_{kl}|^2}$.
9. $\det(\exp(X)) = e^{\mathrm{tr}(X)}$, *where* $\mathrm{tr}(X)$ *denotes the trace of the matrix X.*
10. $\frac{\mathrm{d}}{\mathrm{d}t}\exp(tX) = X\exp(tX) = \exp(tX)X$. *In particular,* $\frac{\mathrm{d}}{\mathrm{d}t}\exp(tX)|_{t=0} = X$.
11. $\exp(X + Y) = \lim_{m\to\infty} \left(\exp(\frac{X}{m})\exp(\frac{Y}{m})\right)^m$. *This is known as the Lie Product formula [13].*

All properties above are proven in [13]. These properties also hold when considering the matrix exponential as a map from $\mathrm{M}_n(\mathbb{C})$ to $\mathrm{M}_n(\mathbb{C})$. Here $\mathrm{M}_n(\mathbb{C})$ denotes the space of all $n \times n$ matrices with complex entries. However, in property 3, one should replace the transpose with the conjugate transpose.

Most properties can directly be shown using Definition 21 and writing out the power series. As an example, the proof of property 6 is given below.

Proof of property 6.

$$\begin{aligned}
\exp(X)\exp(Y) &= \left(I + X + \frac{X^2}{2!} + ...\right)\left(I + Y + \frac{Y^2}{2!} + ...\right) \\
&= \sum_{m=0}^{\infty}\sum_{k=0}^{m} \frac{X^k}{k!}\frac{Y^{m-k}}{(m-k)!} \\
&= \sum_{m=0}^{\infty} \frac{1}{m!}\sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} X^k Y^{m-k}.
\end{aligned}$$

Only because X and Y commute, one can use the binomial theorem

$$(X + Y)^m = \sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} X^k Y^{m-k}.$$

Therefore,

$$\begin{aligned}\exp(X)\exp(Y) &= \sum_{m=0}^{\infty} \frac{1}{m!}(X+Y)^m \\ &= \exp(X+Y).\end{aligned}$$

Similarly, one can prove that $\exp(Y)\exp(X) = \exp(Y+X)$. Therefore,

$$\exp(X)\exp(Y) = \exp(Y)\exp(X) = \exp(X+Y).$$

□

Note that $XY = YX$ is crucial for property 6 to hold. If $XY \neq YX$, i.e., X and Y do not commute, $\exp(X)\exp(Y) = \exp(Z) \neq \exp(X+Y)$, where Z is given by the Baker-Campbell-Hausdorff formula [13].

Now, let us consider methods for calculating the matrix exponential $\exp(X)$, where $X \in \mathrm{M}_n(\mathbb{R})$) without calculating all the powers of X. These methods are also discussed in [13]. We will distinguish three cases: the matrix X is diagonalizable, the matrix X is nilpotent and the matrix X is arbitrary.

2.9.1 Case 1: X is diagonalizable

When $X \in \mathrm{M}_n(\mathbb{R})$ is diagonalizable, there exists an invertible complex matrix $C \in \mathrm{M}_n(\mathbb{C})$, of which the columns are the ordered eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_n$ of X, and a complex diagonal matrix D containing the corresponding eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$ of X, such that $X = CDC^{-1}$. Then, $\exp(X)$ can be written as

$$\begin{aligned}\exp(X) &= \exp(CDC^{-1}) \\ &= C\exp(D)C^{-1} \\ &= C\begin{pmatrix} e^{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_n}\end{pmatrix}C^{-1}.\end{aligned}$$

Here, property 7 of Lemma 2.7 is used.

2.9.2 Case 2: X is nilpotent

An $n \times n$ matrix X is nilpotent if and only if $X^m = 0$ for some positive integer m. Consequently, for all integers $l \geq m$, $X^l = 0$. Therefore,

$$\exp(X) = \sum_{k=0}^{m-1} \frac{X^k}{k!}.$$

can be computed explicitly.

2.9.3 Case 3: X is arbitrary

A matrix can be neither diagonalizable nor nilpotent. However, every matrix X can uniquely be written as $X = S + N$, where S is diagonalizable, N is nilpotent, and $SN = NS$ [13]. Since S and N commute, we can use property 6 of Lemma 2.7 to obtain

$$\begin{aligned}\exp(X) &= \exp(S+N) \\ &= \exp(S)\exp(N).\end{aligned}$$

Now, $\exp(S)$ and $\exp(N)$ can be calculated individually by considering them as case-1 and case-2 matrix exponential, respectively.

Example 2.13. (Calculation of the matrix exponential).

- Consider the 2×2 skew-symmetric matrix X defined as in Equation (22). This matrix is diagonalizable. The eigenvectors of X are $(1,\mathrm{i})^T$ and $(\mathrm{i},1)^T$, with corresponding eigenvalues $-\mathrm{i}\theta$ and $\mathrm{i}\theta$, respectively. Therefore,

$$\begin{aligned}\exp(X) &= \begin{pmatrix} 1 & \mathrm{i} \\ \mathrm{i} & 1 \end{pmatrix}\begin{pmatrix} e^{-\mathrm{i}\theta} & 0 \\ 0 & e^{\mathrm{i}\theta} \end{pmatrix}\frac{1}{2}\begin{pmatrix} 1 & -\mathrm{i} \\ -\mathrm{i} & 1 \end{pmatrix}\\ &= \frac{1}{2}\begin{pmatrix} e^{-\mathrm{i}\theta}+e^{\mathrm{i}\theta} & -\mathrm{i}(e^{-\mathrm{i}\theta}-e^{\mathrm{i}\theta}), \\ \mathrm{i}(e^{-\mathrm{i}\theta}-e^{\mathrm{i}\theta}) & e^{-\mathrm{i}\theta}+e^{\mathrm{i}\theta} \end{pmatrix}\\ &= \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.\end{aligned}$$

- Consider the following matrix

$$X = \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}.$$

Note that X is nilpotent since $X^2 = 0_2$. Therefore,

$$\begin{aligned}\exp(X) &= I + X\\ &= \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}.\end{aligned}$$

- Consider the following matrix

$$X = \begin{pmatrix} a & b \\ 0 & a \end{pmatrix}.$$

X can be rewritten as $X = S + N$, where S is a diagonalizable matrix and N a nilpotent matrix, in the following way

$$X = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} + \begin{pmatrix} 0 & b \\ 0 & 0 \end{pmatrix}.$$

Therefore,

$$\begin{aligned}\exp(X) &= \begin{pmatrix} e^a & 0 \\ 0 & e^a \end{pmatrix}\begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix}\\ &= \begin{pmatrix} e^a & e^a b \\ 0 & e^a \end{pmatrix}.\end{aligned}$$

2.10 Lie Groups and Lie Algebras

In this section, the concepts of a Lie group and a Lie Algebra are given. And we relate the previously introduced concepts of the matrix exponential and tangent spaces of matrix manifolds to each other via Lie Algebras.

Definition 2.26 (Lie Group)**.** A Lie group is a smooth manifold G, which is also a group and such that the group product $* : G \times G \to G$ and the inverse map $g \to g^{-1}$ are smooth maps for all $g \in G$.

In Section 2.1, we concluded that $\mathrm{SO}(n)$ together with the matrix product is a matrix group. In section 2.5 have seen that $\mathrm{SO}(n)$ is a smooth submanifold. Therefore, $\mathrm{SO}(n)$ is a Lie group. For matrix groups in general the following theorem holds. [13]

Theorem 2.8. *Every matrix group is a smoothly embedded submanifold of* $\mathrm{M}_n(\mathbb{R})$ *and is thus a Lie group.*

Therefore, matrix groups are also called matrix Lie groups. The proof of Theorem 2.8 can be found in Ref. [13]. The reverse does not hold, i.e., not every Lie group is a matrix (Lie) group. Some counterexamples to prove this statement are also given in Ref. [13]. One should keep this difference between matrix (Lie) groups and Lie groups in mind.

The main reason why we want to see $\mathrm{SO}(n)$ as a Lie group is because a Lie group has a corresponding matrix Lie algebra, which gives us a nice way to describe the tangent space of $\mathrm{SO}(n)$. The definition of a general Lie algebra is given below.

Definition 2.27 (Lie Algebra)**.** A Lie algebra $\mathfrak{g}$ over some field F is a vector space equipped with a binary operator, called the bracket operator

$$[\cdot,\cdot]:\mathfrak{g}\times\mathfrak{g}\to\mathfrak{g},$$

such that the following properties hold

- bilinearity, i.e.,

$$\begin{aligned}[aX+bY,Z]&=a[X,Z]+b[Y,Z],\\ [X,aY+bZ]&=a[X,Y]+b[X,Z],\end{aligned}$$

  for all $a,b\in F$ and $X,Y,Z\in\mathfrak{g}$.

- alternativity, i.e.,

$$[X,X]=0,$$

  for all $X\in\mathfrak{g}$.

- Jacobi identity, i.e.,

$$[X,[Y,Z]]+[Y,[Z,X]]+[Z,[X,Y]]=0,$$

  for all $X,Y,Z\in\mathfrak{g}$.

By expanding $[X+Y,X+Y]$ and using the bilinearity and the alternativity properties, we obtain

$$[X,Y]=-[Y,X]. \tag{24}$$

This property is called anticommutativity.

Now let us consider a matrix group G, and let us define the following set $\mathfrak{g}$ as

$$\mathfrak{g}=\{X:\exp(tX)\in G,\forall t\in\mathbb{R}\}. \tag{25}$$

We claim that the following theorem holds.

Theorem 2.9. *Let G be a matrix group. The set $\mathfrak{g}=\{X:\exp(tX)\in G,\forall t\in\mathbb{R}\}$ equipped with the bracket operator $[X,Y]=XY-YX$ is a Lie algebra.*

Proof. This proof consists of two parts. First, we will prove that the bracket operator $[\cdot,\cdot]$ fulfils all the properties, i.e., $[\cdot,\cdot]$ must fulfil the bilinearity, alternativity and Jacobi identity conditions. Then, we will prove that $\mathfrak{g}$ is a vector space, which is closed under the bracket operator.
Checking all conditions for the bracket operator.
Bilinearity,

$$\begin{aligned}[X,aY+bZ]&=X(aY+bZ)-(aY+bZ)X,\\ &=aXY+bXZ-aYX-bZX,\\ &=a[X,Y]+b[X,Z].\\ [aX+bY,Z]&=(aX+bY)Z-Z(aX+bY),\\ &=aXZ+bYZ-aZX-bZY,\\ &=a[X,Z]+b[Y,Z].\end{aligned}$$

Alternativity,

$$[X, X] = XX - XX = 0.$$

Jacobi identity,

$$\begin{aligned}
[X,[Y,Z]] + [Y,[Z,X]] + [Z,[X,Y]] &= [X, YZ - ZY] + [Y, ZX - XZ] + [Z, XY - YX], \\
&= X(YZ - ZY) - (YZ - ZY)X \\
&\quad + Y(ZX - XZ) - (ZX - XZ)Y \\
&\quad + Z(XY - YX) - (XY - YX)Z, \\
&= XYZ - XZY - YZX + ZYX \\
&\quad + YZX - YXZ - ZXY + XZY \\
&\quad + ZXY - ZYX - XYZ + YXZ = 0.
\end{aligned}$$

So, all conditions for the bracket operator are fulfilled. Next, we will show that $\mathfrak{g}$ is a vector space which is closed under the bracket operator. Let $X, Y \in \mathfrak{g}$ be arbitrary, we require that the common properties of a vector space are satisfied and that $\mathfrak{g}$ is closed under the bracket operator, i.e., we require

1. $sX \in \mathfrak{g}$, for all $s \in \mathbb{R}$,
2. $X + Y \in \mathfrak{g}$,
3. $[X, Y] = XY - YX \in \mathfrak{g}$.

To prove the first property, note that $\exp(tsX) \in G$. Therefore $sX \in \mathfrak{g}$.

Next, let us look at the second property. Using the Lie Product Formula (property 11 of Lemma 2.7) gives

$$\exp(t(X+Y)) = \lim_{m\to\infty} \left(\exp\left(t\frac{X}{m}\right) \exp\left(t\frac{Y}{m}\right)\right)^m.$$

Note that $(\exp(t\frac{X}{m})\exp(t\frac{Y}{m}))^m \in G$ for all $m \in \mathbb{N}$. Since G is a matrix group (see Definition 2.3) and since the limit is invertible, the limit must be again in G. Therefore, $X + Y$ is in $\mathfrak{g}$.

To prove the third property, let us take an arbitrary $A \in G$. By property 7 of Lemma 2.7, we have

$$\exp(tAYA^{-1}) = A\exp(tY)A^{-1} \in G.$$

Note that $A\exp(tY)A^{-1}$ is an element of G, since $A \in G$ and $\exp(tY) \in G$. Therefore, $AYA^{-1} \in \mathfrak{g}$. Now, let $A = \exp(tX)$. Then, $\exp(tX)Y\exp(-tX) \in \mathfrak{g}$ for all $t \in \mathbb{R}$. From properties 1 and 2, one can conclude that $\mathfrak{g}$ is a subspace of $\mathrm{M}_n(\mathbb{R})$. This means, in particular, that $\mathfrak{g}$ is a topologically closed subset of $\mathrm{M}_n(\mathbb{R})$ [13], i.e., for any convergent sequence $(Z_m)_{m=0}^{\infty}$ of elements $Z_m \in \mathfrak{g}$ the limit is also in $\mathfrak{g}$. Now let

$$Z_m = m\left(\exp\left(\frac{X}{m}\right) Y \exp\left(-\frac{X}{m}\right) - Y\right).$$

Note that $Z_m \in \mathfrak{g}$. Therefore, $\lim_{m\to\infty} Z_m \in \mathfrak{g}$. Using the definition of the differential, property 10 of Lemma 2.7 and the product rule, we obtain

$$\begin{aligned}
\lim_{m\to\infty} Z_m &= \lim_{m\to\infty} m\left(\exp\left(\frac{X}{m}\right) Y \exp\left(-\frac{X}{m}\right) - Y\right) \\
&= \lim_{h\to 0} \frac{\exp(hX)Y\exp(-hX) - Y}{h} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\exp(tX)Y\exp(-tX)\right)|_{t=0} \\
&= \left(X\exp(tX)Y\exp(-tX) + \exp(tX)Y(-X)\exp(-tX)\right)|_{t=0} \\
&= XY - YX.
\end{aligned}$$

Therefore, $XY - YX \in \mathfrak{g}$. So, $\mathfrak{g}$ is a vector space which is closed under the bracket operator. This completes the proof. $\square$

Due to Theorem 2.9 we can say that the set $\mathfrak{g}$ defined by Equation (25) is the Lie algebra of the matrix group G. In particular, for $\mathrm{SO}(n)$ we will denote the corresponding Lie algebra as $\mathfrak{so}(n)$, i.e.,

$$\mathfrak{so}(n) = \{X : \exp(tX) \in \mathrm{SO}(n), \forall t \in \mathbb{R}\}. \tag{26}$$

To further characterize the Lie algebra of $\mathrm{SO}(n)$. Consider the following Lemma.

Lemma 2.10. *A matrix $X \in \mathrm{M}_n(\mathbb{R})$ is in the Lie algebra of $\mathrm{SO}(n)$ if and only if the matrix X is a skew-symmetric matrix, i.e. $X^T = -X$.*

Proof $\Rightarrow$. Let $\mathfrak{so}(n)$ denote the Lie Algebra of the rotation group $\mathrm{SO}(n)$, i.e.,

$$\mathfrak{so}(n) = \{X : \exp(tX) \in \mathrm{SO}(n), \forall t \in \mathbb{R}\}.$$

Let $X \in \mathfrak{so}(n)$ be chosen arbitrarily, then we have

$$(\exp(tX))^T \exp(tX) = I_n, \quad \forall t.$$

Differentiating this equation with respect to t and substuting $t = 0$ gives

$$X^T + X = 0_{n\times n},$$

so $X^T = -X$. Therefore X is skew-symmetric. $\square$

Proof $\Leftarrow$. Let X be an arbitrary skew-symmetric matrix, i.e., $X^T = -X$, and let $t \in \mathbb{R}$ be arbitrary. We have

$$\exp(tX)^T = \exp(tX^T) = \exp(-tX) = (\exp(tX))^{-1}.$$

Therefore,

$$\exp(tX)^T \exp(tX) = I_n.$$

Furthermore, we have

$$\det(\exp(tX)) = e^{(\mathrm{tr}(tX))} = e^0 = 1$$

So, $X \in \mathfrak{so}(n)$. $\square$

So, the Lie algebra of $\mathrm{SO}(n)$ is the set of all $n \times n$ skew-symmetric matrices. There are other matrix groups of which there are clear descriptions of their corresponding Lie Algebra. Some of them are given in Table 2. There, J_n is defined as the $2n \times 2n$ symplectic matrix

$$J_n = \begin{pmatrix} 0_{n\times n} & I_{n\times n} \\ -I_{n\times n} & 0_{n\times n} \end{pmatrix}.$$

Lie Group	Lie Algebra
$\mathrm{GL}(n) = \{X \in \mathrm{M}_n(\mathbb{R}) : \det(X) \neq 0\}$ general linear group	$\mathfrak{gl}(n) = \mathrm{M}_n(\mathbb{R})$ $n \times n$ matrices
$\mathrm{SL}(n) = \{X \in \mathrm{M}_n(\mathbb{R}) : \det(X) = 1\}$ special linear group	$\mathfrak{sl}(n) = \{Z \in \mathrm{M}_n(\mathbb{R}) : \mathrm{tr}(Z) = 0\}$ trace zero matrices
$\mathrm{O}(n) = \{X \in \mathrm{M}_n(\mathbb{R}) : X^T X = I_{n\times n}\}$ orthogonal group	$\mathfrak{o}(n) = \{Z \in \mathrm{M}_n(\mathbb{R}) : Z^T + Z = 0_{n\times n}\}$ skew-symmetric matrices
$\mathrm{SO}(n) = \{X \in \mathrm{O}(n) : \det(X) = 1\}$ special orthogonal group	$\mathfrak{so}(n) = \{Z \in \mathrm{M}_n(\mathbb{R}) : Z^T + Z = 0_{n\times n}\}$ skew-symmetric matrices
$\mathrm{Sp}(2n) = \{X \in \mathrm{M}_{2n}(\mathbb{R}) : X^T J_n X = J_n\}$ symplectic group	$\mathfrak{sp}(2n) = \{Z \in \mathrm{M}_{2n}(\mathbb{R}) : JZ + Z^T J = 0_{2n\times 2n}\}$

Table 2: Lie algebras of some matrix groups. Note that $\mathfrak{o}(n) = \mathfrak{so}(n)$. This table can also be found in Ref. [12].

The following theorem describes the connection between the Lie algebra of a matrix group and the tangent space of this matrix group.

Theorem 2.11. *The matrix Lie algebra $\mathfrak{g} = \{X : \exp(tX) \in G, \forall t \in \mathbb{R}\}$ of a matrix Lie group G completely describes the tangent space of the matrix Lie group G at the identity element I, i.e.,*

$$\mathfrak{g} = T_I G$$

To prove this theorem, we need the definition of the matrix logarithm, which is given below.

Definition 2.28 (matrix logarithm)**.** Let $X \in \mathrm{M}_n(\mathbb{R})$, the matrix logarithm $\log : \mathrm{M}_n(\mathbb{R}) \to \mathrm{M}_n(\mathbb{R})$ is defined by

$$\log(X) = \sum_{m=1}^{\infty} (-1)^{m+1} \frac{(X - I_{n\times n})^m}{m}, \tag{27}$$

whenever the series converges.

The sum in Equation (27) might not converge. However, it does converge for matrices in a neighbourhood around the identity matrix $I_{n\times n}$. In particular, the series converges for X, if $\|X - I_{n\times n}\|_{\mathrm{F}} < 1$, where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm defined as

$$\|A\|_{\mathrm{F}} = \sqrt{\langle A, A\rangle_{\mathrm{F}}} = \sqrt{\mathrm{tr}(A^T A)}. \tag{28}$$

The matrix logarithm is the inverse map of the matrix exponential, in other words,

$$\log(\exp(X)) = X,$$
$$\exp(\log(X)) = X.$$

For proof, see Theorem 2.8 of Ref. [18]. Now, let us prove Theorem 2.11.

Proof of Theorem 2.11. The proof consists of two parts. First, we will show that $\mathfrak{g} \subseteq T_I G$. Secondly, we will show that $T_I G \subseteq \mathfrak{g}$.
Part 1.
Let $X \in \mathfrak{g}$ be arbitrary, then define the path $\gamma : [-1, 1] \to G$ as

$$\gamma(s) = \exp(sX).$$

Note that by definition of $\mathfrak{g}$ we indeed have that $\gamma(s) \in G$ for all $s \in [-1, 1]$. Furthermore,

$$\gamma(0) = \exp(0) = I,$$
$$\frac{\mathrm{d}\gamma(0)}{\mathrm{d}s} = X \exp(0) = X.$$

So, $X \in T_I G$.
Part 2.
Let $Y \in T_I G$ be chosen arbitrarily. Then, there exist an $\epsilon > 0$ and a path $\gamma(s) : [-\epsilon, \epsilon] \to G$, such that $\gamma(0) = I$ and $Y = \frac{\mathrm{d}\gamma(0)}{\mathrm{d}s}$. So, we need to prove that $\exp\left(t\frac{\mathrm{d}\gamma(0)}{\mathrm{d}s}\right) \in G$. It is sufficient to prove that $\exp\left(\frac{\mathrm{d}\gamma(0)}{\mathrm{d}s}\right) \in G$, since we can simply create another path $\gamma^*(s)$ that satisfies the same properties as $\gamma(s)$ by re-scaling the s parameter, using t, by choosing

$$\gamma^*(s) = \gamma(ts),$$
$$\frac{\mathrm{d}\gamma^*(s)}{\mathrm{d}s} = t\frac{\mathrm{d}\gamma(s)}{\mathrm{d}s}.$$

Next, let us define $\psi(s) = \log(\gamma(s))$ for small values of $|s| < \epsilon$. Here log is the matrix logarithm as defined in Definition 2.28. The matrix logarithm is uniquely defined in a neighbourhood around the identity element $I_{n\times n}$. Since we choose $|s|$ to be small, $\gamma(s)$ will be in this neighbourhood. Therefore, $\gamma(s) = \exp(\psi(s)) \in G$ for small s. Using the chain rule we obtain

$$\frac{\mathrm{d}\psi(s)}{\mathrm{d}s} = \mathrm{Dlog}(\gamma(s))\frac{\mathrm{d}\gamma(s)}{\mathrm{d}s}.$$

Subsitution of $s = 0$ gives

$$\frac{\mathrm{d}\psi(0)}{\mathrm{d}s} = \mathrm{Dlog}(I_{n\times n})\frac{\mathrm{d}\gamma(0)}{\mathrm{d}s}.$$

Note that

$$\mathrm{Dexp}(0_{n\times n}) = I_{n\times n}.$$

By the inverse function theorem we have

$$\mathrm{Dlog}(I_{n\times n}) = (\mathrm{Dexp}(0_{n\times n}))^{-1} = I_{n\times n}.$$

Therefore,

$$\frac{\mathrm{d}\psi(0)}{\mathrm{d}s} = \frac{\mathrm{d}\gamma(0)}{\mathrm{d}s}.$$

So, it remains to prove that $\exp(\frac{\mathrm{d}\psi(0)}{\mathrm{d}s}) \in G$. Using the definition of the derivative we obtain

$$\frac{\mathrm{d}\psi(0)}{\mathrm{d}s} = \lim_{h\to 0}\frac{\psi(h) - \psi(0)}{h}.$$

Note that $\psi(0) = \log(\gamma(0)) = \log(I_{n\times n}) = 0_{n\times n}$. Take $h = \frac{1}{n}$, then

$$\frac{\mathrm{d}\psi(0)}{\mathrm{d}s} = \lim_{n\to\infty} n\psi\left(\frac{1}{n}\right).$$

Note that $\psi(\frac{1}{n})$ is defined for sufficiently large n. Recall that we have $\gamma(s) = \exp(\psi(s)) \in G$ for small $|s|$. Therefore, $\gamma(s)^n = \exp(n\psi(s))$ is also in G for small values of s. Now take n sufficiently large, then $\exp(n\psi(\frac{1}{n})) \in G$. Since G is a matrix group and since a matrix exponential is always invertible, we have that $\exp(\lim_{n\to\infty} n\psi(\frac{1}{n})) = \exp(\frac{\mathrm{d}\psi(0)}{\mathrm{d}s}) \in G$. Therefore, $\exp(\frac{\mathrm{d}\gamma(0)}{\mathrm{d}s}) \in G$, $\exp(tY) \in G$ and $Y \in \mathfrak{g}$. This completes the proof. □

We can combine Theorem 2.9 and Theorem 2.11 to conclude that one can use the Lie Algebra of a matrix group G as a way to describe the tangent space of G at the identity element I.

To conclude this section, let us look at what this all means for the matrix group $\mathrm{SO}(n)$. In Section 2.6, we found Equation (15), which describes the tangent space of $\mathrm{SO}(n)$ at $R \in \mathrm{SO}(n)$. In particular, the tangent space at the identity element I can be described by

$$T_I \,\mathrm{SO}(n) = \{\Omega : \Omega \in \mathcal{S}_{\mathrm{skew}}(n)\}. \tag{29}$$

In this section, we found an alternative way to describe $T_I \,\mathrm{SO}(n)$, i.e.,

$$\boxed{T_I \,\mathrm{SO}(n) = \mathfrak{so}(n) = \{X : \exp(tX) \in \mathrm{SO}(n)\}.} \tag{30}$$

This observation explains the result from example 2.12. There we calculated the matrix exponential of an arbitrary 2×2 skew-symmetric matrix and found that this results in an element of $\mathrm{SO}(2)$.

2.11 Differential Equations on Submanifolds and their Tangent Spaces

In this section, differential equations on submanifolds will be discussed. These differential equations give another perspective on concepts like tangent spaces, Lie algebras and the matrix exponential. This perspective will give us a new way to describe the tangent space $T_X \,\mathrm{SO}(n)$ at an arbitrary element $X \in \mathrm{SO}(n)$. Whereas, in the previous section we only found a way to describe the tangens space $T_I \,\mathrm{SO}(n)$ at the identity element I_n. A lot of the results given in this section can also be found in Ref. [12].

The definition of a differential equation on a submanifold is given below.

Definition 2.29 (vector field, differential equation and integral curves on a manifold)**.** Let $\mathcal{M}$ be a submanifold. A vector field on $\mathcal{M}$ is a continuously differentiable mapping $f : \mathcal{M} \to \mathbb{R}^d$ such that

$$f(Y) \in T_Y \mathcal{M} \text{ for all } y \in \mathcal{M}.$$

For such a vector field,

$$\dot{Y} = f(Y) \tag{31}$$

is called a differential equation on the submanifold $\mathcal{M}$, and a function $Y : I \to \mathcal{M}$, where $I \subset \mathbb{R}$, satisfying $\dot{Y}(t) = f(Y(t))$ for all $t \in I$ is called an integral curve or simply a solution of the differential equation.

The following theorem describes the existence and uniqueness of an initial value problem on a manifold as described by Equation (31). The proof of this theorem will not be given here but can be found in Ref. [12]. More information on how the solution of the differential equation depends on the initial conditions and how perturbations in the system propagate is also given in Ref. [12].

Theorem 2.12 (Existence an uniqueness of differential equations on a submanifold)**.** *Consider a differential equation $\dot{y} = f(y)$ on a submanifold $\mathcal{M} \subset \mathbb{R}^n$. Here, $f : \mathcal{M} \to \mathbb{R}^n$ is a continuous differentiable vector field. Then, for every $y_0 \in \mathcal{M}$, there exists a maximal open interval $I = I(y_0)$ and a twice continuous differentiable function $y : I \to \mathcal{M}$ satisfying*

- *$y(t)$ is a solution of $\dot{y} = f(y)$ on I satisfying $y(0) = y_0$.*
- *If $\hat{y} : J \to \mathcal{M}$ is a solution of $\dot{y} = f(y)$, with initial condition $y(0) = y_0$, on the interval J, then $J \subset I$ and $\hat{y}(t) = y(t)$ for $t \in J$.*

Example 2.14. (vector field on S^2).

- As an example of a differential equation on a submanifold, we will consider Euler's equations of motion, which describe the rotation of a rigid body [12]. Let I_1, I_2, I_3 be the principal moments of inertia of the rigid body and let $\mathbf{y}(t) = (y_1(t), y_2(t), y_3(t))^T$ be the angular momentum vector and let $\mathbf{y}_0 = y(0)$ be the initial condition. Here $t \in \mathbb{R}_{\geq 0}$ represents time. Then the Euler equations state

$$\begin{aligned}
\dot{y}_1 &= (I_3^{-1} - I_2^{-1}) y_3 y_2, \\
\dot{y}_2 &= (I_1^{-1} - I_3^{-1}) y_1 y_3, \\
\dot{y}_3 &= (I_2^{-1} - I_1^{-1}) y_2 y_1.
\end{aligned}$$

We will show that Euler's equations of motion are a system of equations which can also be described as a differential equation on a submanifold of the form $\dot{\mathbf{y}} = f(\mathbf{y})$, where

$$f(\mathbf{y}) = \begin{pmatrix} (I_3^{-1} - I_2^{-1})y_3y_2 \\ (I_1^{-1} - I_3^{-1})y_1y_3 \\ (I_2^{-1} - I_1^{-1})y_2y_1 \end{pmatrix}.$$

First, note that the vector $\mathbf{y}$ lies on a submanifold of $\mathbb{R}^3$. This can be seen by considering the function $C : \mathbb{R}^3 \to \mathbb{R}$, which is defined as

$$C(\mathbf{y}) = \frac{1}{2}\left(y_1^2 + y_2^2 + y_3^2\right). \tag{32}$$

Taking the derivate with respect to time gives

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}(y_1^2 + y_2^2 + y_3^2)\right) &= y_1\dot{y}_1 + y_2\dot{y}_2 + y_3\dot{y}_3, \\ &= (I_3^{-1} - I_2^{-1})y_1y_2y_3 + (I_1^{-1} - I_3^{-1})y_1y_2y_3 + (I_2^{-1} - I_1^{-1})y_1y_2y_3, \\ &= 0.\end{aligned}$$

So, $C(\mathbf{y})$ is a preserved quantity. Therefore,

$$\mathcal{M} = \{\mathbf{y} \in \mathbb{R}^3 : \frac{1}{2}\|\mathbf{y}\|_2 - C(\mathbf{y}_0) = 0\} \tag{33}$$

describes a submanifold on which the solution $\mathbf{y}(t)$ lives for all $t \in \mathbb{R}_{\geq 0}$. Let $g : \mathbb{R}^3 \to \mathbb{R}$ be defined as

$$\begin{aligned} g(\mathbf{y}) &= C(\mathbf{y}) - C(\mathbf{y}_0) \\ &= \frac{1}{2}\|\mathbf{y}\|_2 - C(\mathbf{y}_0).\end{aligned}$$

Note that $g(\mathbf{y}) = 0$ for all $\mathbf{y}$ on the manifold. From this, one can conclude that the manifold described by Equation (33) is a two-dimensional manifold. More precisely, $\mathcal{M}$ represents a sphere in $\mathbb{R}^3$ with radius $r = \sqrt{2C(\mathbf{y}_0)}$. Note that

$$\mathrm{D}g(\mathbf{y}) = (y_1, y_2, y_3).$$

Therefore,

$$\begin{aligned}\mathrm{D}g(\mathbf{y})[f(\mathbf{y})] &= (y_1, y_2, y_3)\begin{pmatrix} (I_3^{-1} - I_2^{-1})y_3y_2 \\ (I_1^{-1} - I_3^{-1})y_1y_3, \\ (I_2^{-1} - I_1^{-1})y_2y_1 \end{pmatrix} \\ &= (I_3^{-1} - I_2^{-1})y_1y_2y_3 + (I_1^{-1} - I_3^{-1})y_1y_2y_3 + (I_2^{-1} - I_1^{-1})y_1y_2y_3, \\ &= 0.\end{aligned}$$

Comparing the equality above with the description of the tangent space through the level set representation of the submanifold (see Equation (8)) one can conclude that $f(\mathbf{y}) \in T_{\mathbf{y}}\mathcal{M}$. So, $\dot{\mathbf{y}} = f(\mathbf{y})$ is indeed a differential equation on the submanifold $\mathcal{M}$.

An important result from the previous section was the equality $\mathfrak{g} = T_IG$, where $\mathfrak{g}$ is the matrix Lie algebra of a matrix group G and T_IG is the tangent space of the matrix Lie group at the identity element I. The following theorem gives a description of the tangent space at an arbitrary element of G (see [2], [12]).

Theorem 2.13. *The tangent space T_YG of a matrix Lie group G at an element $Y \in G$ can be given as*

$$T_YG = \{AY : A \in \mathfrak{g}\} = \{YA : A \in \mathfrak{g}\}, \tag{34}$$

where $\mathfrak{g}$ is the Lie algebra of the matrix Lie group G.

Proof. (Proof of Theorem 2.13) Consider an arbitrary element $A \in \mathfrak{g}$. Since $\mathfrak{g} = T_I G$, there exists a differentiable path $\alpha(t)$ in G satisfying $\alpha(0) = I$ and $\alpha'(0) = A$. For a fixed $Y \in G$, the path $\gamma_1(t) = \alpha(t)Y$ is in G and satisfies $\gamma_1(0) = Y$ and $\dot{\gamma_1}(0) = AY$. Consequently, $AY \in T_Y G$. Alternatively, the path $\gamma_2(t) = Y\alpha(t)$ is in G and satisfies $\gamma_2(0) = Y$ and $\dot{\gamma_2}(0) = YA$. Consequently, $YA \in T_Y G$. $\square$

$\dot{Y} = AY$ defines a differential equation on the submanifold G. The solution $Y(t) = \exp(tA)$ is therefore in G for all $t \in \mathbb{R}$. This again shows that the matrix exponential maps elements from the Lie algebra to the Lie group, i.e. $\exp : \mathfrak{g} \to G$. This results in the following theorem.

Theorem 2.14. *Let G be a matrix Lie group and $\mathfrak{g}$ its Lie algebra. If $A(Y) \in \mathfrak{g}$ for all $Y \in G$ and if the initial condition $Y_0 \in G$ then the solution of $\dot{Y} = YA(Y)$ satisfies $Y(t) \in G$ for all $t \in \mathbb{R}$ (see [12]).*

Now, let us consider the case where $G = \mathrm{SO}(n)$. Applying Theorem 2.13 gives

$$\boxed{T_R\,\mathrm{SO}(n) = \{R\Omega : \Omega \in \mathfrak{so}(n)\}.} \tag{35}$$

Note that this expression is equivalent to Equation (15), since $\mathfrak{so}(n) = S_{\text{skew}}(n)$.

To conclude this section, let us apply Theorem 2.14 on the following initial value problem defined on $\mathrm{SO}(n)$.

- $Y(0) = R \in \mathrm{SO}(n)$,
- $Y'(t) = \Omega Y(t)$, where $\Omega \in \mathcal{S}_{\text{skew}}(n)$

The solution is given by

$$Y(t) = R\exp(t\Omega). \tag{36}$$

Note that $\Omega \in \mathfrak{so}(n)$, therefore Theorem 2.14 applies. So, $Y(t) \in \mathrm{SO}(n)$ for all $t \in \mathbb{R}$

3 Optimization on SO(n)

Given is a smooth function $f : M_n(\mathbb{R}) \to \mathbb{R}$. This chapter aims to obtain an iterative method to solve the following minimization problem.

$$\min_{R \in \mathrm{SO}(n)} f(R). \tag{37}$$

Consider the following example of such a problem: Let $\mathbf{x}, \mathbf{y}$ be given vectors in $\mathbb{R}^n$. Suppose one wants to find a rotation matrix $R \in \mathrm{SO}(n)$, such that $R\mathbf{x} = \mathbf{y}$. One can consider this as a minimization problem by defining the function

$$\begin{aligned} f(R) &= \|R\mathbf{x} - \mathbf{y}\|_2, \\ &= (R\mathbf{x} - \mathbf{y})^T (R\mathbf{x} - \mathbf{y}). \end{aligned}$$

which needs to be minimized for all $R \in \mathrm{SO}(n)$. In general, the problem described by Equation (37) is not easy, due to the restriction $R \in \mathrm{SO}(n)$. If we ignore this restriction we could use the classical gradient descent algorithm over $M_n(\mathbb{R})$ which is an iterative algorithm given by

$$R_{t+1} = R_t - \eta \mathrm{Grad} f(R_t). \tag{38}$$

Here, $\eta \geq 0$ is the learning rate (which may depend on the number of prior iteration steps t) and $\mathrm{Grad} f(R)$ is the Euclidean matrix gradient as defined below.

Definition 3.1 (Euclidean (matrix) gradient)**.** Given a real-valued matrix function $f : M_n(\mathbb{R}) \to \mathbb{R}$, the Euclidean gradient of $f(R)$ at $R = (r_{ij}) \in M_n(\mathbb{R})$ is defined to be the matrix,

$$\mathrm{Grad} f(R) = \begin{pmatrix} \frac{\partial f(R)}{\partial r_{11}} & \cdots & \frac{\partial f(R)}{\partial r_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(R)}{\partial r_{n1}} & \cdots & \frac{\partial f(R)}{\partial r_{nn}} \end{pmatrix}. \tag{39}$$

Note, that there is a one-to-one correspondence between the definition of the Euclidean matrix gradient and the definition of the classical Euclidean gradient for a function $\hat{f} : \mathbb{R}^{n^2} \to \mathbb{R}$. To see this, let us define a mapping $\mathrm{vec} : \mathrm{M}_n(\mathbb{R}) \to \mathbb{R}^{n^2}$, which rearranges the elements of a matrix $R \in \mathrm{M}_n(\mathbb{R})$ into a vector $\mathrm{vec}(R) \in \mathbb{R}^{n^2}$. Moreover, consider the function $\hat{f} : \mathbb{R}^{n^2} \to \mathbb{R}$, which maps the vector $\mathrm{vec}(R) \in \mathbb{R}^{n^2}$ to the scalar $f(R)$, i.e., $\hat{f}(\mathrm{vec}(R)) = f(R)$ for all $R \in M_n(\mathbb{R})$. The classical Euclidean gradient of $\hat{f}$ is denoted as $\nabla \hat{f}(\mathrm{vec}(R)) \in \mathbb{R}^{n^2}$. When one rearranges this classical Euclidean gradient $\nabla \hat{f}(\mathbf{r})$ into an $n \times n$ matrix by using the inverse vec^{-1} of the mapping vec, one obtains the Euclidean matrix gradient $\mathrm{Grad} f(R) = \mathrm{vec}^{-1}(\nabla \hat{f}(\mathrm{vec}(R)))$. In the remainder of this report, we will call the Euclidean matrix gradient simply the Euclidean gradient.

When optimizing over $\mathrm{SO}(n)$ the problem which arises when using the classical gradient descent algorithm is that the updated matrix R_{t+1} in Equation (38) is not necessarily an element of $\mathrm{SO}(n)$ even if $R_t \in \mathrm{SO}(n)$. Moreover, $\mathrm{Grad} f(R_t)$ is not necessarily an element of the tangent space of $\mathrm{SO}(n)$ at R_t. In this chapter, we adjust the classical gradient descent algorithm to obtain an iterative method where the updated matrix R_{t+1} stays in $\mathrm{SO}(n)$. Therefore, we give the definition of the Riemannian gradient, which can be regarded as an extension of the concept of a Euclidean gradient from $\mathrm{M}_n(\mathbb{R})$ to Riemannian manifolds. Secondly, we look into retractions, which are maps that map a matrix in the tangent space back onto the submanifold $(\mathrm{SO}(n))$ in a meaningful way. Then, we will talk about line-search methods and finally, we will look into the convergence properties of these line-search methods.

3.1 Riemannian gradient

Let us first consider the space of $\mathbb{R}^n$ equipped with the standard inner product $\langle \cdot, \cdot \rangle_2 : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, i.e., $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth scalar function. Let us look at how we use the Euclidean gradient $\nabla f(\mathbf{r}_t)$ to obtain the steepest-descent direction $\mathbf{x}^*$ of f at $\mathbf{r}_t$. Using this direction we can update $\mathbf{r}_t$ as follows.

$$\mathbf{r}_{t+1} = \mathbf{r}_t + \eta \mathbf{x}^*, \quad \eta \geq 0,$$

where η is the learning rate and $\mathbf{x}^*$ is the direction of steepest-descent, which is defined as

$$\mathbf{x}^* = \underset{x \in \mathbb{R}^n : \|x\|_2 = 1}{\operatorname{argmin}} \mathrm{D}f(\mathbf{r}_t)[\mathbf{x}]. \tag{40}$$

Here, $\|x\|_2$ denotes the vector 2-norm, i.e., $\|x\|_2 = \sqrt{\langle x, x \rangle_2}$ and $\mathrm{D}f(\mathbf{r}_t)[\mathbf{x}]$ denotes the differential or directional derivative of f at $\mathbf{r}_t$ along $\mathbf{x}$ (see Definition 2.10) which in this case can also be written as

$$\mathrm{D}f(\mathbf{r}_t)[\mathbf{x}] = \langle \nabla f(\mathbf{r}_t)), \mathbf{x} \rangle_2 = (\nabla f(\mathbf{r}_t))^T \mathbf{x}. \tag{41}$$

Now, consider the following lemma.

Lemma 3.1 (Steepest-descent direction $\mathbb{R}^n$)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth mapping. Then*

$$\underset{\|x\|_2 = 1}{\operatorname{argmin}} \mathrm{D}f(r)[x] = x^* = -\frac{\nabla f(r)}{\|\nabla f(r)\|_2}.$$

Proof. Let $\mathbf{x} \in \mathbb{R}^n$, with $\|\mathbf{x}\|_2 = 1$. By the Cauchy-Schwarz inequality

$$-\|\nabla f(\mathbf{r}_t)\|_2 \leq \mathrm{D}f(\mathbf{r}_t)[\mathbf{x}] = \langle \nabla f(\mathbf{r}_t), \mathbf{x} \rangle_2 \leq \|\nabla f(\mathbf{r}_t)\|_2. \tag{42}$$

These bounds are achieved for $\mathbf{x} = \pm \frac{\nabla f(\mathbf{r}_t)}{\|\nabla f(\mathbf{r}_t)\|_2}$. Therefore, the direction of greatest descent is

$$\mathbf{x}^* = -\frac{\nabla f(\mathbf{r}_t)}{\|\nabla f(\mathbf{r}_t)\|_2}.$$

□

Note that it also follows from the proof of Lemma 3.1 that the direction of greatest ascent is $-\mathbf{x}^*$.

Now let us consider an embedded submanifold $\mathcal{M}$ of a Riemannian manifold $\overline{\mathcal{M}}$, a smooth scalar field $f : \mathcal{M} \to \mathbb{R}$ and the iteration point $\mathbf{r}_t$. We will drop the subscript t for notation purposes. We can extend the idea of minimising the directional derivative from $\mathbb{R}^n$ to the manifold. Analogous to Equation (40), we have that the direction of greatest descent x^* is given by

$$x^* = \underset{x \in T_r\mathcal{M} : \|x\| = 1}{\operatorname{argmin}} \mathrm{D}f(r)[x]. \tag{43}$$

Here, $\|x\|$ is the norm of x induced by the Riemannian metric at $r \in \mathcal{M}$. $\mathrm{D}f(r)[x]$ is again called the directional derivative of f at r along x. Note the restriction of x onto $T_r\mathcal{M}$. This restriction is important. To see this, suppose x has a component $y \in (T_r\mathcal{M})^\perp$ such that $x = \mathrm{P}_{T_r\mathcal{M}}x + y$ (see Section 2.8), this component gives us no meaningful direction of movement over the submanifold $\mathcal{M}$ since this component is orthogonal with respect to the submanifold at point r.

Similar to the case of $\mathbb{R}^n$ (see Equation (41)), we have a way to express $\mathrm{D}f(r)[x]$ in terms of a gradient. This gradient is called the Riemannian gradient or simply gradient.

Definition 3.2 (Riemannian gradient)**.** Let $\mathcal{M}$ be a Riemannian manifold and let $\langle \cdot, \cdot \rangle_r$ be the Riemannian metric of $\mathcal{M}$. Given is also a smooth map $f : \mathcal{M} \to \mathbb{R}$. Then, the Riemannian gradient $\mathrm{grad}f(r)$ of f at a point $r \in \mathcal{M}$ is an element of $T_r\mathcal{M}$, which is defined in the following sense

$$\mathrm{D}f(r)[x] = \langle \mathrm{grad}f(r), x \rangle_r, \quad \forall x \in T_r\mathcal{M}. \tag{44}$$

In other words, the Riemannian gradient $\mathrm{grad}\, f(r)$ is the Riesz representer of the linear functional $\mathrm{D}f(r) \in \mathcal{L}(T_r\,\mathcal{M})$ that takes values in the space $T_r\,\mathcal{M}$ with inner product $\langle \cdot, \cdot \rangle_r$. The Riesz represented (thus the Riemannian gradient) is an element of $T_r\,\mathcal{M}$. [5]

The Riemannian gradient has the following properties (see [2]):

- The direction of grad$f(r)$ is the steepest-ascent direction of f at r

$$\frac{\mathrm{grad}f(r)}{\|\mathrm{grad}f(r)\|} = \underset{x \in T_R\mathcal{M}: \|x\|=1}{\mathrm{argmax}} \ \mathrm{D}f(r)[x]. \tag{45}$$

- The norm of grad$f(R)$ gives the steepest slope of f at r:

$$\|\mathrm{grad}f(r)\| = \mathrm{D}f(r)\left[\frac{\mathrm{grad}f(x)}{\|\mathrm{grad}f(x)\|}\right]. \tag{46}$$

One can prove these properties by using the Cauchy-Schwartz inequality in a similar way as was done for $\mathbb{R}^n$ (see Equation (42)), but now replacing the standard inner product with the Riemannian metric. Similar one can show that $-\mathrm{grad}f(r)$ is the steepest-descent direction of f at r.

Note that for $\mathcal{M} = \mathbb{R}^n$, where $\mathcal{M}$ is equipped with the standard inner product $\langle\cdot,\cdot\rangle_2$, we have that the Riemannian gradient is equal to the orthogonal projection of the Euclidean gradient into the tangent space (see Example 2.5). In general, this holds for all vector fields where one uses the standard inner product as a Riemannian metric [2].

3.2 Calculation of the Riemannian Gradient

In the previous section, we have given the definition of the Riemannian gradient and we have seen that this gradient gives us the direction of greatest ascent and descent. In this section, we develop a method to find the Riemannian gradient at a point $R \in \mathrm{SO}(n)$. Recall that $\mathrm{SO}(n)$ can be regarded as a submanifold of the Riemannian manifold $\mathrm{M}_n(\mathbb{R})$.

Let the Riemannian metric of $\mathrm{M}_n(\mathbb{R})$ be the Frobenius inner product (see Equation (16)). Since the Frobenius inner product is defined on $M_n(\mathbb{R})$, it is also defined on the tangent space of $\mathrm{SO}(n)$. Therefore, $\mathrm{SO}(n)$ together with this metric form a Riemannian manifold. Now let $f : M_n(\mathbb{R}) \to \mathbb{R}$ be a smooth function which has to be minimized (see Equation (37)). Since f is defined over $M_n(\mathbb{R})$ it is also defined over the tangent space of $\mathrm{SO}(n)$. For notation let us denote $\mathrm{SO}(n)$ as $\mathcal{M}$. The Riemannian gradient of f at R is given in the following sense

$$\mathrm{D}f(R)(V) = \langle \mathrm{grad}\, f(R), V\rangle_F, \quad \forall V \in T_R\,\mathcal{M}\,. \tag{47}$$

Note that we write V and R as uppercase letters to emphasise that these are matrices. Since $\mathrm{M}_n(\mathbb{R})$ is a vector space of which the Frobenius inner product is its standard inner product, we have

$$\mathrm{D}f(R)(V) = \langle \mathrm{Grad}\, f(R), V\rangle_F, \quad \forall V \in T_R\,\mathrm{M}_n(\mathbb{R}) = \mathrm{M}_n(\mathbb{R})\,. \tag{48}$$

Here $\mathrm{Grad}\, f(R)$ is the Euclidean matrix gradient as defined in definition 3.1. Since $T_R\,\mathcal{M} \subset M_n(\mathbb{R})$, we can use Equation (17). We have for all $V \in T_R\,\mathcal{M}$

$$\begin{aligned}\langle \mathrm{grad}\, f(R), V\rangle_F =& \langle \mathrm{Grad}\, f(R), V\rangle_F \\ =& \langle \mathrm{P}_{T_R\,\mathcal{M}}\,\mathrm{Grad}\, f(R) + \mathrm{P}_{(T_R\,\mathcal{M})^\perp}\,\mathrm{Grad}\, f(R), V\rangle_F \\ =& \langle \mathrm{P}_{T_R\,\mathcal{M}}\,\mathrm{Grad}\, f(R), V\rangle_F.\end{aligned}$$

Therefore,

$$\boxed{\mathrm{grad}\, f(R) = \mathrm{P}_{T_R\,\mathcal{M}}\,\mathrm{Grad}\, f(R).} \tag{49}$$

Here, $\mathrm{P}_{T_R\,\mathcal{M}}\,\mathrm{Grad}\, f(R)$ denotes the orthogonal projection of $\mathrm{Grad}\, f(R)$ onto $T_R\,\mathcal{M}$ and $\mathrm{P}_{(T_R\,\mathcal{M})^\perp}\,\mathrm{Grad}\, f(R)$ denotes the orthogonal projection of $\mathrm{Grad}\, f(R)$ onto $(T_R\,\mathcal{M})^\perp$. Equation (49) gives us a way to calculate the Riemannian gradient of f at $R \in \mathrm{SO}(n)$.

3.3 Projecting the Euclidean gradient

In this section, we discuss a way to project the Euclidean gradient Grad $f(R)$ onto the tangent space $T_R\mathrm{SO}(n)$ of $\mathrm{SO}(n)$ and find the Riemannian gradient at a point $R \in \mathrm{SO}(n)$. For notation, purposes denote $\mathrm{SO}(n)$ as G and denote the Lie algebra $\mathfrak{so}(n)$ of $\mathrm{SO}(n)$ as $\mathfrak{g}$.

According to Theorem 2.11, we have that the matrix Lie algebra $\mathfrak{g}$ of G completely describes the tangent space T_IG at the identity element I. According to Lemma 2.10, the Lie algebra $\mathfrak{g}$ of G and the tangent space T_IG is the vector space of all skew-symmetric matrices $\mathcal{S}_{\text{skew}}(n)$. This can be denoted as

$$T_IG = \operatorname{span}\{U_1, ..., U_k\}. \tag{50}$$

Here $k = \frac{1}{2}n(n-1)$ and the sequence $(U_i)_{i=1}^k$, $U_i \in \mathrm{M}_n(\mathbb{R})$ forms a basis for the vector space of all $n \times n$ skew-symmetric matrices. We choose the following basis

$$U_1 = \frac{1}{2}\sqrt{2}\begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \tag{51}$$

$$U_2 = \frac{1}{2}\sqrt{2}\begin{pmatrix} 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \tag{52}$$

$$\vdots \tag{53}$$

$$U_n = \frac{1}{2}\sqrt{2}\begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \tag{54}$$

$$\vdots \tag{55}$$

$$U_k = \frac{1}{2}\sqrt{2}\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 \\ 0 & \cdots & 0 & 0 & -1 & 0 \end{pmatrix}. \tag{56}$$

Note that for all $i, j \in \{1, ..., k\}$, we have

$$\langle U_i, U_j \rangle_F = \delta_{i,j}.$$

Therefore $\{U_1, ..., U_k\}$ forms an orthonormal basis for T_IG.

In general, we want to describe the tangent space T_RG at an arbitrary element $R \in G$. According to Theorem 2.13,

$$T_RG = \{RA : A \in \mathfrak{g}\}.$$

Therefore, $\{RU_1, ..., RU_k\}$ forms a basis for T_RG and we have

$$T_RG = \text{span}\{RU_1, ..., RU_k\}.$$

According to Equation (49), we need to project the Euclidean gradient Grad $f(R)$ onto this tangent space to obtain the Riemannian gradient grad $f(R)$. Before we discuss how to project onto this space, let us first consider the following example of a projection of a vector in $\mathbb{R}^n$ onto a linear subspace.

Example 3.1. (Projection onto a linear subspace of $\mathbb{R}^n$)

- Let us consider a vector $\mathbf{x} \in \mathbb{R}^n$ and the vector space $\mathcal{V} \subseteq \mathbb{R}^n$ of dimension $k \leq n$, which is given by

  $$\mathcal{V} = \text{span}\{\mathbf{u}_1, ..., \mathbf{u_k}\}.$$

  Here, $\mathbf{u}_i \in \mathbb{R}^n$ for $i \in \{1, ..., k\}$ and $\{\mathbf{u}_1, ..., \mathbf{u}_k\}$ forms an orthonormal basis for $\mathcal{V}$. Let us define a matrix $U \in \mathbb{R}^{n\times k}$, the columns of which are the basisvectors of $\mathcal{V}$, i.e.,

  $$U = \begin{pmatrix} \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_k \\ \vdots & & \vdots \end{pmatrix}.$$

  We want to project the vector $\mathbf{x}$ onto $\mathcal{V}$. The projected vector is denoted as $\mathrm{P}_{\mathcal{V}}\mathbf{x}$. We can write $\mathbf{x}$ as

  $$\mathbf{x} = \mathrm{P}_{\mathcal{V}}\mathbf{x} + \mathrm{P}_{\mathcal{V}}^T\mathbf{x},$$

  where $\mathrm{P}_{\mathcal{V}^\perp}\mathbf{x}$ denotes the projection onto the orthogonal complement $\mathcal{V}^\perp$ of $\mathcal{V}$, where $\mathcal{V}^\perp$ is a $n-k$ dimensional vector space.

  $\mathrm{P}_{\mathcal{V}}\mathbf{x}$ can be written as

  $$\mathrm{P}_{\mathcal{V}}\mathbf{x} = U\mathbf{c},$$

  where $\mathbf{c}$ represents the coordinate vector of $\mathrm{P}_{\mathcal{V}}\mathbf{x}$ with respect to the basis $\{\mathbf{u}_1, ..., \mathbf{u}_k\}$. Due to orthonormality, we have

  $$\begin{aligned} U^T\mathrm{P}_{\mathcal{V}^\perp}\mathbf{x} &= 0_k \\ U^T(\mathbf{x} - U\mathbf{c}) &= 0_k \\ \mathbf{c} &= (U^TU)^{-1}U^T\mathbf{x} \\ \mathrm{P}_{\mathcal{V}}\mathbf{x} &= UU^T\mathbf{x}, \end{aligned} \tag{57}$$

  where we used that $U^TU = I_{k\times k}$. Equation (57) gives us a way to project x onto $\mathcal{V}$.

Next, let us define

$$U_R = \begin{pmatrix} \vdots & & \vdots \\ \text{vec}\left(\frac{RU_1}{\|RU_1\|_F}\right) & \cdots & \text{vec}\left(\frac{RU_k}{\|RU_k\|_F}\right) \\ \vdots & & \vdots \end{pmatrix}, \tag{58}$$

where vec : $\mathrm{M}_n(\mathbb{R}) \to \mathbb{R}^{n^2}$ is an operator which orders the elements of a $n \times n$ matrix into a n^2 dimensional vector. The way of ordering can be chosen arbitrarily. We will use row-wise ordering in our final algorithm. Furthermore, let $\text{vec}^{-1} : \mathbb{R}^{n^2} \to \mathrm{M}_n(\mathbb{R})$ denote the inverse of vec. U_R is an orthogonal matrix and note that

$$U_I = \begin{pmatrix} \vdots & & \vdots \\ \text{vec}(U_1) & \cdots & \text{vec}(U_k) \\ \vdots & & \vdots \end{pmatrix}. \tag{59}$$

Since $T_R G$ can be seen as a linear subspace of $\mathbb{R}^{n^2}$, we proceed similarly as in Example 3.1. Therefore, the Riemannian gradient in vector form $\mathrm{vec}(\mathrm{grad}\, f(R))$ is given by

$$\mathrm{grad}\, f(R)) = \mathrm{vec}^{-1}(U_R U_R^T \,\mathrm{vec}(\mathrm{Grad}\, f(R))), \tag{60}$$

In Equation (60), we transform the Euclidean gradient into a vector of $\mathbb{R}^{n^2}$. Then, we project onto the linear subspace described by the columns of U_R. The resulting vector is then transformed back into a matrix of $\mathrm{M}_n(\mathbb{R})$. This matrix is the Riemannian gradient of $f(R)$.

Example 3.2. (Calculating the Riemannian gradient)

- Suppose $\mathbf{x}, \mathbf{y} \in R^n$ are given and we want to minimize the loss function

  $$f(R) = \|R\mathbf{x} - \mathbf{y}\|_2$$

  for $R \in \mathrm{SO}(n)$. Note that if $\mathbf{x}, \mathbf{y} \in \mathrm{S}^{n-1}$, then there exists an R^*, such that $R^*\mathbf{x} = \mathbf{y}$, therefore $f(R^*) = 0$. The Euclidean gradient $\mathrm{Grad}\, f(R)$ of $f(R)$ is given by

  $$\mathrm{Grad}\, f(R) = 2(R\mathbf{x} - \mathbf{y})\mathbf{x}^T.$$

  Using Equation (60) and U_R as defined by Equation (58) gives

  $$\mathrm{grad}\, f(R) = 2\,\mathrm{vec}^{-1}(U_R U_R^T \,\mathrm{vec}((R\mathbf{x} - \mathbf{y})\mathbf{x}^T)).$$

The Riemannian gradient needs to be computed during every iteration step. Moreover, using the method described above, we need to compute a basis for $T_R G$ and obtain U_R. Then, U_R needs to be multiplied with its transpose. U_R is a big $n^2 \times k$ matrix, so constructing this matrix and calculating $U_R U_R^T$ can be computationally expensive and we need to redo all these calculations during every iteration step. Therefore, consider the following lemma.

Lemma 3.2. *Let $M = \mathrm{SO}(n)$, $V \in \mathrm{M}_n(\mathbb{R})$ and $R \in \mathrm{SO}(n)$, then*

$$\boxed{\mathrm{P}_{T_R G}\, V = R\, \mathrm{P}_{T_I G}(R^T V),} \tag{61}$$

where $\mathrm{P}_{T_R G}$ denotes the projection onto $T_R G$ and $\mathrm{P}_{T_I G}$ denotes the projection onto $T_I G = \mathfrak{g}$.

Proof. From the definition of a projection, we have

$$\langle \mathrm{P}_{T_R G}\, V, R\Omega \rangle_F = \langle V, R\Omega \rangle_F, \quad \forall \Omega \in T_I G$$

and

$$\langle \mathrm{P}_{T_I G}(R^T V), \Omega \rangle_F = \langle R^T V, \Omega \rangle_F, \quad \forall \Omega \in T_I G.$$

Using the equalities above and the property $\langle AB, C \rangle_F = \langle B, A^T C \rangle_F$, we obtain

$$\langle \mathrm{P}_{T_I G}(R^T V), \Omega \rangle_F = \langle R^T V, \Omega \rangle_F = \langle V, R\Omega \rangle_F = \langle \mathrm{P}_{T_R G}\, V, R\Omega \rangle_F = \langle R^T\, \mathrm{P}_{T_R G}\, V, \Omega \rangle_F, \quad \forall \Omega \in T_I G.$$

Therefore,

$$\begin{aligned} R^T\, \mathrm{P}_{T_R G}\, V &= \mathrm{P}_{T_I G}(R^T V), \\ \mathrm{P}_{T_R G}\, V &= R\, \mathrm{P}_{T_I G}(R^T V). \end{aligned}$$

□

From Lemma 3.2 and Equation (49) , we obtain

$$\operatorname{grad} f(R) = R\,\mathrm{P}_{T_I G}(R^T \operatorname{Grad} f(R)). \tag{62}$$

Projecting, like in Equation (60), gives

$$\boxed{\operatorname{grad} f(R) = R\,\mathrm{vec}^{-1}(U_I U_I^T\,\mathrm{vec}(R^T \operatorname{Grad} f(R))).} \tag{63}$$

We will use Equation (63) rather than Equation (60) to calculate the Riemannian gradient since this is computationally more efficient. Because, by using Equation (63), we no longer have to construct a new basis for the tangent space $T_R G$ during every iteration step. We can simply use $\{U_1, ..., U_k\}$ as a basis for $T_I G$, obtain U_I and then precompute $U_I U_I^T$.

3.4 Retractions

In the previous section, we found a way to calculate the Riemannian gradient $\operatorname{grad} f(R) \in T_R \operatorname{SO}(n)$ of the loss function $f : \operatorname{SO}(n) \to \mathbb{R}$ at a point R. Since $-\operatorname{grad} f(R)$ is the steepest descent direction (see Equation (45)), we suggest the following update method

$$R_{t+1} = R_t - \eta \operatorname{grad} f(R_t), \tag{64}$$

where t is the iteration step and η is the learning rate. This update method is already an improvement with respect to classical gradient descent, given by Equation (38). In this new method, we move in the direction of a tangent vector since $\operatorname{grad} f(R_t) \in T_{R_t} \operatorname{SO}(n)$, whereas $\operatorname{Grad} f(R_t)$ does not lie in the tangent space $T_{R_t} \operatorname{SO}(n)$ necessarily. However, we still have the problem that the update R_{t+1} does not necessarily lie in $\operatorname{SO}(n)$. In our final update method, we want to move in the direction of a tangent vector, while staying on the manifold. To ensure the latter, we will introduce the notion of a retraction mapping. Let $x \in \mathcal{M}$ be a point on the manifold $\mathcal{M}$. A retraction ρ at x, denoted by ρ_x, is a mapping from $T_x \mathcal{M}$ to $\mathcal{M}$ that preserves gradient at x. A more formal definition is given below (see [2]).

Definition 3.3 (retraction)**.** Consider a manifold $\mathcal{M}$ and a smooth mapping ρ from the tangent bundle $T\mathcal{M}$ onto M. Let ρ_x denote the restriction of ρ on $T_x \mathcal{M}$. ρ is called a retraction of $\mathcal{M}$ if for all $x \in \mathcal{M}$

$$\rho_x(0_x) = x. \quad \text{and} \tag{65}$$

$$\mathrm{D}\rho_x(0_x)[v] = v, \quad \forall v \in T_x \mathcal{M}, \tag{66}$$

where 0_x denotes the zero element of the tangent space $T_x \mathcal{M}$.

In general, we assume that the domain of ρ is the whole tangent bundle $T\mathcal{M}$. This property will hold for all the retractions we will discuss in this report.

Now we suggest the following update method, which we will call the Riemannian gradient descent method.

$$R_{t+1} = \rho_{R_t}(-\eta \operatorname{grad} f(R_t)), \tag{67}$$

where t is the iteration step, $\rho_{R_t} : T_{R_t} \mathcal{M} \to \mathcal{M}$ is a retraction and η is the learning rate. Suppose the Riemannian gradient $\operatorname{grad} f(R_t)$ is equal to the zero element $0_{R_t} \in T_{R_t} \mathcal{M}$. (In case of optimization over $\operatorname{SO}(n)$, we have $0_{R_t} = 0_{n\times n}$, where $0_{n\times n}$ is the $n \times n$ zero matrix). The first property, Equation (65), guarantees that 0_{R_t} is sent to R_t. This is desirable since if $\operatorname{grad} f(R_t) = 0_{R_t}$, then we are at a possible minimum of f. During each iteration step, we want to move in the direction of $-\operatorname{grad} f(R_t)$ over the manifold to get closer to a minimum of the loss function f. The second property, Equation (66), secures that we indeed move in the direction of $-\operatorname{grad} f(R_t)$ over the manifold, at least for an infinitesimal step size.

A special kind of retraction is a retraction which is based on a decomposition. Examples of such decompositions are the QR-decomposition or the polar decomposition (see [2]).

Theorem 3.3 (retractions based on decompositions)**.** *Let $\mathcal{M}$ be an embedded submanifold of a vector space $\overline{\mathcal{M}}$ and let $\mathcal{N}$ be a manifold such that* $\dim(\mathcal{M}) + \dim(\mathcal{N}) = \dim(\overline{\mathcal{M}})$*. Assume that there is a diffeomorphism*

$$\begin{aligned} \phi : \mathcal{M} \times \mathcal{N} &\to \overline{\mathcal{M}}_* \\ (X, Y) &\to \phi(X, Y), \end{aligned}$$

where $\overline{\mathcal{M}}_*$ is an open subset of $\overline{\mathcal{M}}$, with a neutral element $I \in \mathcal{N}$ satisfying

$$\phi(X, I) = X, \quad \forall X \in \mathcal{M}\,.$$

Let $V \in T_X\,\mathcal{M}$. The mapping

$$\rho_X(V) = \pi_1(\phi^{-1}(X+V)), \tag{68}$$

defines a retraction on $\mathcal{M}$. Here

$$\begin{aligned} \pi_1 : \mathcal{M} \times \mathcal{N} &\to \mathcal{M} \\ (X, Y) &\to X \end{aligned}$$

denotes the projection onto the first component.

The proof of this theorem can be found in Section 4.1 of Ref. [2]. Next, we will consider some examples of retractions based on decompositions.

Example 3.3. Retractions and decompositions

- Consider the unit sphere $\mathcal{M} = S^{n-1}$ in $\overline{\mathcal{M}} = \mathbb{R}^n$, let $\mathcal{N} = \{r \in \mathbb{R} : r > 0\}$, and consider the mapping

$$\begin{aligned} \phi : \mathcal{M} \times \mathcal{N} &\to \mathbb{R}^n_* \\ (x, r) &\to xr, \end{aligned}$$

  where $\mathbb{R}_* = \{x \in \mathbb{R}^n : x \neq 0_n\}$ is open. One can easily verify that ϕ is a diffeomorphism. Using Theorem 3.3, we can conclude that

$$\rho_x(v) = \frac{x+v}{\|x+v\|_2}$$

  is a retraction, where $v \in T_x\,\mathcal{M}$. This retraction maps the point $x+v$, to the point $\rho_x(v) \in S^{n-1}$ which lies closest to $x+v$. For S^1 this retraction method is visualized in Figure 6.

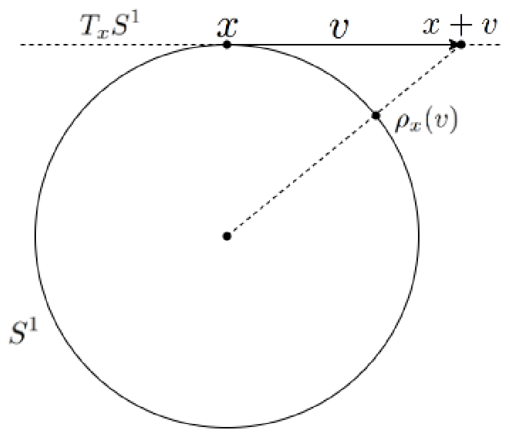


Figure 6: Retraction onto the unit sphere S^1, which is a submanifold of $\mathbb{R}^2$. Here $x \in S^1$ and $v \in T_xS^1$. Suppose we want to minimize a smooth function f over S^1 using the Riemannian gradient descent method and we start at x. Then, $v = -\eta\,\mathrm{grad}\,f(R)$ and $\rho_x(v)$ is the next iteration point.

- Consider the manifold of orthogonal matrices O_n. A QR-decomposition of an $n \times n$ real matrix $A \in \mathrm{M}_n(\mathbb{R})$ is the decomposition $A = QR$, where $Q \in \mathrm{O}(n)$ is an orthogonal matrix and $R \in \mathcal{S}_{\text{upp+}}(n)$ is an upper triangular matrix with strictly positive diagonal elements. Note that $\dim(\mathrm{O}(n)) = \frac{1}{2}n(n-1)$ (see Example 2.8), $\dim(\mathcal{S}_{\text{upp+}}(n)) = \frac{1}{2}n(n+1)$ and $\dim(\mathrm{M}_n(\mathbb{R})) = n^2 = \dim(\mathrm{O}(n)) + \dim(\mathcal{S}_{\text{upp+}}(n))$.

The inverse of a QR decomposition is the mapping

$$\begin{aligned}\phi :& \mathrm{O}_n \times \mathcal{S}_{\text{upp}+}(n) \to \mathrm{M}_n(\mathbb{R}) \,. \\ & (Q, R) \to QR.\end{aligned}$$

ϕ is a diffeomorphism and has the property that $\phi(Q, I) = Q$ for all $Q \in \mathrm{O}(n)$. Let

$$\begin{aligned}\pi_1 :& \mathrm{O}(n) \times \mathcal{S}_{\text{text}+}(n) \to \mathrm{O}(n) \\ & (Q, R) \to Q\end{aligned}$$

be the projection onto the first component. According to Theorem 3.3 the map

$$\mathrm{qf} = \pi_1 \circ \phi^{-1}$$

is a retraction onto $\mathrm{O}(n)$.

3.5 Geodesics and the Riemannian Retraction

Let us reconsider the classical gradient descent algorithm in $\mathrm{M}_n(\mathbb{R})$ with no restrictions on R, i.e., $R \in \mathrm{M}_n(\mathbb{R})$ (see Equation (38)). Here, we move in a straight line from R_t towards R_{t+1} in the direction of $-\mathrm{Grad} f(R_t)$. In Section 3.1, we have given the definition of the Riemannian gradient grad $f(R_t)$ of f at R_t and we argued that on a manifold $\mathcal{M} = \mathrm{SO}(n)$ we should move into the direction of $-\mathrm{grad} f(R_t)$, since this is the direction of greatest descent and we have $\mathrm{grad} f(R_t) \in T_{R_t}\mathcal{M}$, whereas $\mathrm{Grad} f(R_t)$ is not necessarily an element of the tangent space $T_{R_t}\mathcal{M}$. Like in the classical gradient method on $\mathrm{M}_n(\mathbb{R})$, we would like to move in a straight line over the manifold G in the direction of $-\mathrm{grad} f(R_t)$. This raises the question of how to move in a "straight line" over a curved space like a manifold. "Straight lines" over a manifold are called geodesics. Geodesics are one of the fundamental concepts of differential geometry.

Since $\mathrm{SO}(n)$ is a Riemannian submanifold of $\mathrm{M}_n(\mathbb{R})$ and there is a one-to-one correspondence between $\mathrm{M}_n(\mathbb{R})$ and $\mathbb{R}^{n^2}$, we can consider $\mathrm{SO}(n)$ as a submanifold of $\mathbb{R}^{n^2}$. Then we have the following definition of a geodesic.

Definition 3.4. (geodesic) Let $\mathcal{M}$ be a Riemannian submanifold of $\mathbb{R}^n$ were the Riemannian metric is induced by the usual metric on $\mathbb{R}^n$. Furthermore, let $\gamma : I \to \mathcal{M}$ be a curve on the manifold, where I is some interval of $\mathbb{R}$. $\gamma'' \in \mathbb{R}^n$, i.e. the second derivative of γ, is called the acceleration vector, which can be decomposed into a tangential and a normal component with respect to the tangent space $T_\gamma\mathcal{M}$, i.e.,

$$\gamma'' = \gamma''^{\parallel} + \gamma''^{\perp},$$

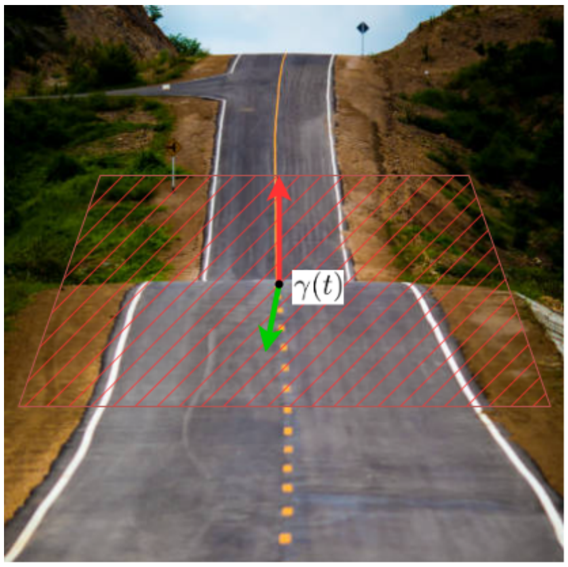
where $\gamma''^{\parallel} \in T_\gamma\mathcal{M}$ and $\gamma''^{\perp} \in N_\gamma\mathcal{M}$. $N_\gamma\mathcal{M}$ denotes the normal space of $\mathcal{M}$ at γ, which is the orthogonal complement of the tangent space $T_\gamma\mathcal{M}$.

The curve γ is called a geodesic if and only if

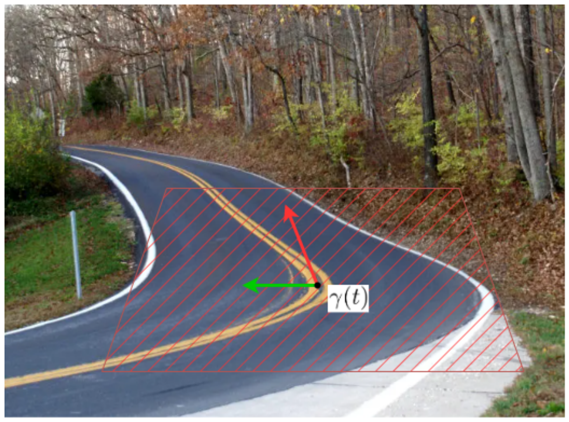
$$\gamma''^{\parallel}(t) = 0_n, \quad \forall t \in I.$$

As an example, consider the roads in Figure 7 as curved surfaces in $\mathbb{R}^3$. Let γ be the curve on the road which is marked by the orange road markings at the centre of the road. The tangent vector $\gamma'(t)$ is denoted as a red arrow and the acceleration vector $\gamma''(t)$ is denoted as a green arrow. The tangent plane $T_{\gamma(t)}\mathcal{M}$ is depicted as a red plane. In Figure **??**, γ is a geodesic since γ'' has no tangential component. Note that it does have a normal component, which is pointing straight into the ground. In Figure 7b, γ is not a geodesic, since γ'' does have a tangential component.

For general Riemannian manifolds, geodesics can be defined in terms of an affine connection. Let $\mathfrak{X}(\mathcal{M})$ denote the space of vector fields of the manifold $\mathcal{M}$ (see Definition 2.29). An affine connection $\nabla : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M})$ is a map which can map one tangent space to another (see [9]). One can define infinitely many affine connections on a Riemannian manifold. However, for every Riemannian manifold there exists one unique affine connection, called the Levi-Civita or Riemannian connection, which is both torsion-free



(a) γ is a geodesic of this curved surface in $\mathbb{R}^3$.



(b) γ is not a geodesic of this curved surface in $\mathbb{R}^3$.

Figure 7: On the left-hand side, the road markings are forming a geodesic. On the right-hand-side, they are not.

and compatible with the Riemannian metric. The Koszul formula characterizes this connection (see [9] and [2]).

The idea behind affine connections is illustrated in Figure 8. We can see an affine connection as the operation of rolling one tangent plane over a curve on the manifold to get to another tangent plane. Intuitively, it makes sense to define geodesics by using affine connections, since when we are moving away from a point $x_0 \in M$ in the direction of the tangent vector $v_0 \in T_{x_0}\mathcal{M}$ by an infinitesimal step, we arrive at a new point $x_\delta \in \mathcal{M}$, were we want to keep moving in the direction of v_0. However, v_0 is not necessarily an element of the tangent space of $T_{x_\delta}\mathcal{M}$ anymore. An affine connection gives us a way to transform the tangent vector v_0 into a new tangent vector $v_\delta \in T_{x_\delta}\mathcal{M}$. Now, we can continue moving in a "straight line" by using this new tangent vector v_δ.

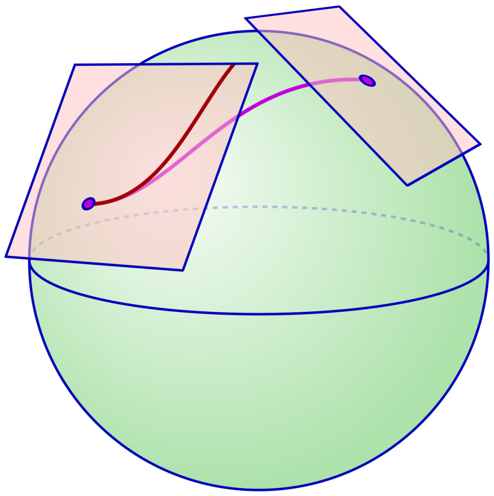


Figure 8: Illustration of an affine connection. Intuitively speaking, an affine connection transforms one tangent space into another one, by rolling the tangent space over a curve on the manifold.

We will not dive into further details about affine connections and geodesics, since our case study of $\mathrm{SO}(n)$, where $\mathrm{SO}(n)$ is endowed with the Frobenius norm as a Riemannian metric, allows us for a rather simple analysis of geodesics with respect to the general case where we need to define geodesics in terms of an affine connection. For further reading see Ref. [9] or other introductory books about differential geometry.

For $\mathrm{SO}(n)$ we claim the following.

Theorem 3.4. *γ is a geodesic of $\mathrm{SO}(n)$ if and only if*

$$\gamma^T \gamma'' = \gamma''^T \gamma$$

Proof. Equation (14) from Section 2.6 gives us an expression for the tangent space $T_R\,\mathrm{SO}(n)$ of $\mathrm{SO}(n)$ at $R \in \mathrm{SO}(n)$. Equation (20) from Section 2.8 gives us an expression for the normal space $N_R\,\mathrm{SO}(n)$. Furthermore, note that $\gamma \in \mathrm{SO}(n)$, therefore $\gamma^T = \gamma^{-1}$. We have

$$\gamma \text{ is a geodesic} \iff \gamma''^{\parallel} = 0 \iff \gamma'' \in N_\gamma\,\mathrm{SO}(n) \iff \gamma^T\gamma'' \in \mathcal{S}_{\mathrm{sym}}(n) \iff \gamma^T\gamma'' = \gamma''^T\gamma.$$

□

When we are at R_t we would like to move in a direction $v = -\mathrm{grad} f(R_t)$ in a "straight line". We have seen that geodesics on a manifold are like straight lines in $\mathbb{R}^n$. Therefore, let us consider the following theorem. (For more details we refer to Section 5.4 of [2]).

Theorem 3.5 (existence and uniqueness of geodesics)**.** *Let $\mathcal{M}$ be a Riemannian manifold. For every point $x \in \mathcal{M}$ and every tangent vector $v \in T_x\,\mathcal{M}$, there is some interval $(-\beta, \beta)$ and a unique geodesic*

$$\gamma(t; x, v) : (-\beta, \beta) \to \mathcal{M},$$

satisfying the conditions

$$\gamma(0; x, v) = x, \quad \gamma'(0; x, v) = v.$$

The following theorem gives an expression for geodesics on $\mathrm{SO}(n)$.

Theorem 3.6. *Let $R \in \mathrm{SO}(n)$ and let $\Omega \in \mathcal{S}_{skew}(n)$, then*

$$\begin{aligned}\gamma :&\mathbb{R} \to \mathcal{M} :\\ &t \to \gamma(t) = R\exp(t\Omega),\end{aligned}$$

is the unique geodesic satisfying the conditions

$$\gamma(0; R, R\Omega) = R, \quad \gamma'(0; R, R\Omega) = R\Omega.$$

Proof. The conditions are easily verified. What is left to prove is that γ is a geodesic. Therefore, by using Theorem 3.4, we need to prove that $\gamma^T\gamma'' = \gamma''^T\gamma$. We have

$$\gamma''(t) = R\Omega^2\exp(t(\Omega)).$$

Since $\Omega \in \mathcal{S}_{\text{skew}}(n)$, we have

$$(\Omega^2)^T = (\Omega^T)^2 = (-\Omega)^2 = \Omega^2$$

and since $R \in \mathrm{SO}(n)$, we have $R^TR = I_{n\times n}$ Rewriting $\gamma^T\gamma''$ gives

$$\begin{aligned}\gamma^T\gamma'' &= (R\exp(t\Omega))^TR\Omega^2\exp(t\Omega) = (\exp(t\Omega))^TR^TR\Omega^2\exp(s\Omega) = (\exp(t\Omega))^T(\Omega^2)^TR^TR\exp(t\Omega)\\ &= (R\Omega^2\exp(t\Omega))^TR\exp(t\Omega) = \gamma''^T\gamma.\end{aligned}$$

□

Now, we arrive at the definition of the Riemannian exponential (not to be confused with the matrix exponential defined in Definition 21).

Definition 3.5. (Riemannian exponential and logarithm) Let $\mathcal{M}$ be a smooth manifold on which we can define geodesics, i.e., $\mathcal{M}$ is equipped with an arbitrary connection. Furthermore, let $x \in \mathcal{M}$ and

$$\mathcal{D}_x = \{v \in T_x\mathcal{M} : \gamma(1, x, v) \text{ is defined}\}$$

The mapping

$$\begin{aligned}\mathrm{Exp}_x :&\mathcal{D}_x \to \mathcal{M} :\\ &v \to \mathrm{Exp}_x(v) = \gamma(1, x, v),\end{aligned}$$

where $\gamma(t, x, v)$ denotes the unique geodesic trough x for which $\gamma'(0; x, v) = v$. The inverse of Exp_x is called the Riemannian logarithm, which is denoted as Log_x, which maps from the manifold $\mathcal{M}$ to the tangent space $T_x\mathcal{M}$.

Note that the Riemannian exponential map Exp_x is a retraction (see Definition 3.3), called the Riemannian retraction, with domain $\mathcal{D}_x$. If $\mathcal{D}_x = T_x\mathcal{M}$ for all $x \in \mathcal{M}$, we say that the Riemannian manifold $\mathcal{M}$ is geodesically complete. Let $R \in \mathrm{SO}(n)$ and $V = R\Omega \in T_x\mathrm{SO}(n)$, where $\Omega \in \mathcal{S}_{skew}(n) = \mathfrak{so}(n)$ then, from Theorem 3.6, we can conclude that the Riemannian retraction for $\mathrm{SO}(n)$ is given by

$$\mathrm{Exp}_R(\Omega R) = R\exp(\Omega), \quad \text{i.e.,} \tag{69}$$

$$\mathrm{Exp}_R(V) = R\exp(R^TV). \tag{70}$$

Note that $\mathrm{Exp}_R(V)$ is defined for all $V \in T_R\mathrm{SO}(n)$ and $R \in \mathrm{SO}(n)$. Therefore, $\mathrm{SO}(n)$ is geodesically complete. The Riemannian logarithm is given by

$$\mathrm{Log}_R(Y) = R\log(R^TY),$$

where log denotes the matrix logarithm, defined in Definition 2.28.

Equation (69) should not come as a surprise. At the end of Section 2.11 we discussed a particular initial value problem where $Y(0) = R$ and $Y'(t) = \Omega Y(t)$. We found the solution $Y(t) = R\exp(t\Omega)$ and concluded that $Y(t) \in \mathrm{SO}(n)$ for all t. Now, we know that $Y(t)$ describes the unique geodesic starting at R, which moves in the direction of the tangent vector $R\Omega$.

Substitution of the Riemannian retraction method into Equation (67) gives us the following update method

$$R_{t+1} = R_t \exp(-\eta R_t^T \mathrm{grad} f(R_t)). \tag{71}$$

$\eta > 0$ is the learning rate or step size. Substitution of Equation (49) and using Lemma 3.2 gives

$$\boxed{R_{t+1} = R_t \exp\left(-\eta \mathrm{P}_{T_I\mathcal{M}} R_t^T \mathrm{Grad} f(R_t)\right),} \tag{72}$$

where $\mathcal{M} = \mathrm{SO}(n)$. The projection of $\mathrm{Grad}f(R_t)$ onto $T_I\mathcal{M}$ can be done like in Equation (63). In this update method, at every iteration, we move from R_t over the geodesic in the steepest descent direction $-\mathrm{grad}f(R_t)$. This method will be numerically tested in the next chapter.

3.6 Line Search Methods

The constant line-search (CLS) method for optimization on Riemannian manifolds (see Chapter 4 of Ref. [2]) is given by Algorithm 1. CLS can be regarded as the Riemannian equivalent of the classical gradient descent method and is therefore also called the Riemannian gradient descent method.

Algorithm 1 Constant Line Search (CLS)

```
Require: Riemannian manifold $\mathcal{M}$; continuous differentiable $f : \mathcal{M} \to \mathbb{R}$; retraction $\rho : T\mathcal{M} \to \mathcal{M}$; a learning rate or step size $\eta > 0$.
Input: Initial iterate $x_0 \in \mathcal{M}$.
Output: Sequence of iterates $\{x_t\}$.
1: for $t = 0, 1, 2, ...$ do
2:     $x_{t+1} \leftarrow \rho_{x_t}(-\eta \mathrm{grad} f(x_t))$
3: end for
```

For $\mathrm{SO}(n)$, we can further specify the steps given in Algorithm 1 by implementing the Riemannian retraction (see Equation (72)) and by giving the expression for the Riemannian gradient. This results in Algorithm 2.

Algorithm 2 Constant Line Search on $\mathrm{SO}(n)$

```
Require: Continuous differentiable $f : \mathrm{SO}(n) \to \mathbb{R}$; a learning rate or step size $\eta > 0$; an operator $\mathrm{vec} : \mathrm{M}_n(\mathbb{R}) \to \mathbb{R}^{n^2}$, which orders the elements of a matrix into a vector (see Section 3.3).
Input: Initial iterate $R_0 \in \mathrm{SO}(n)$.
Output: Sequence of iterates $\{R_t\}$.
1: Construct a basis $\{U_1, U_2, ..., U_k\}$ for $\mathcal{S}_{skew}(n)$, where $k = \frac{1}{2}n(n-1)$.
2: Construct the matrix $U_I = (\mathrm{vec}(U_1), \mathrm{vec}(U_2), ..., \mathrm{vec}(U_k)) \in \mathbb{R}^{n^2 \times k}$ (see Equation (59)).
3: $V \leftarrow U_I U_I^T$
4: for $t = 0, 1, 2, ...$ do
5:     $R_{t+1} \leftarrow R_t \exp(-\eta \mathrm{vec}^{-1}(V \mathrm{vec}(R_t^T \mathrm{Grad} f(R_t))))$
6: end for
```

Note that the update method is based on Equation (63). Furthermore note that we precompute V to speed up the algorithm (see section 3.3). We test Algorithm 2 extensively in Chapter 4, where we vary the dimensionality n and the learning rates η.

3.7 Convergence in Continuous Time

In this section, we prove convergence of the CLS algorithm (see Section 3.6) in the continuous-time limit. The analyses in this section is based on information from [3], [24] and [19]. For the convergence properties of the ALS algorithm, we refer to Chapter 4 of [2].

Let us consider the following differential equation, which we refer to as the gradient flow problem.

$$x'(t) + \mathrm{grad} f((x(t)) = 0; \quad x(0) = x_0 \tag{73}$$

where $x : [0, \infty) \to \mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$ is smooth. Now, we have the following theorem from [19].

Theorem 3.7 (Existance and Uniqueness of the gradient flow solution)**.** *Let $\mathcal{M}$ be a Riemannian manifold which is geodesically complete. Let $f : \mathcal{M} \to \mathbb{R}$ be smooth and bounded from below. For every starting point $x_0 \in \mathcal{M}$, the problem described by Equation* (73) *has a unique solution $x(t)$ defined on $[0, \infty)$, which is also continuously differentiable.*

In particular, $\mathrm{SO}(n)$ is geodesically complete. Therefore there exists a unique solution $x(t)$ defined on $\mathrm{SO}(n)$ for every smooth f which is bounded from below.

The discrete Riemannian retraction method (Equation (71)) is based on the following equation

$$R(t + \mathrm{d}t) = \mathrm{Exp}_{R(t)}(-\mathrm{d}t\, \mathrm{grad}\, f(R(t)), \tag{74}$$

where dt is infinitesimal. From this equalition we obtain $R'(t) = -\mathrm{grad} f(R(t))$. Therefore, the continuous version $R(t)$ fulfils the differential equation described by Equation (73). This is also what we would expect since Equation (73) generalizes to a Riemannian setting the continuous version of the steepest descent equation for $\mathbb{R}^n$. In Ref. [19] it is stated that $R(t)$ decreases and converges for a geodesically complete manifold like $\mathrm{SO}(n)$.

We can prove additional convergence properties when $f : \mathcal{M} \to \mathbb{R}$ is μ-strongly convex (see Definition 3.6).

Definition 3.6 (μ-strongly convex)**.** Let $\mathcal{M}$ be a Riemannian manifold which is geodesically complete on $A \subset \mathcal{M}$. A smooth function $f : A \to \mathcal{M}$ is called (geodesically) μ-strongly convex, $\mu > 0$, if for all $x, y \in A$

$$f(x) - f(y) \geq \langle \mathrm{grad} f(y), \mathrm{Log}_y(x) \rangle_y + \frac{\mu}{2} \|\mathrm{Log}_y(x)\|^2. \tag{75}$$

Here, $\mathrm{Log}_y(x)$ is the Riemannian logarithm, which maps x onto the tangent space $T_y\, \mathcal{M}$ (see Definition 3.5).

Now, consider the following theorem about the rate of convergence of the continuous-times solution.

Theorem 3.8 (convergence rate in the continuous-time limit)**.** *Let $\mathcal{M}$ be a Riemannian manifold which is geodesically complete. Let $f : \mathcal{M} \to \mathbb{R}$ be μ-stronlgy convex and let $x(t)$ be the solution to Equation* (73)*. $x(t)$ minimizes $f(x)$ with rate*

$$f(x(t)) - f^* \leq e^{-2\mu t}(f(x(0)) - f^*), \tag{76}$$

where $f^* = \min_{x \in \mathcal{M}} f(x)$.

Before we prove this theorem, let us consider two other theorems, which we will use in the proof of Theorem 3.8.

Theorem 3.9 (Polyak-Lojasiewicz)**.** *Let f be μ-strongly convex on a geodesically complete Riemannian manifold $\mathcal{M}$. Then,*

$$f^* \geq f(x) - \frac{1}{2\mu} \|\mathrm{grad} f(x)\|^2, \tag{77}$$

where $f^* = \min_{x \in \mathcal{M}} f(x)$.

Proof. Let $x, y \in \mathcal{M}$. From μ-stronlgy convexity we obtain

$$f(y) \geq f(x) + \langle \mathrm{grad} f(x), \mathrm{Log}_x(y) \rangle_x + \frac{\mu}{2} \|\mathrm{Log}_x(y)\|^2, \tag{78}$$

where $\langle \cdot, \cdot \rangle_x$ denotes the Riemannian metric at $x \in \mathcal{M}$. Minimizing on both sides with respect to y gives.

$$f^* \geq f(x) + \min_{y \in \mathcal{M}} \left(\langle \mathrm{grad} f(x), \mathrm{Log}_x(y) \rangle_x + \frac{\mu}{2} \|\mathrm{Log}_x(y)\|^2 \right).$$

In this case, minimizing over $y \in \mathcal{M}$ is equivalent to minimizing over $v = \mathrm{Log}_x(y) \in T_x\mathcal{M}$. Therefore,

$$f^* \geq f(x) + \min_{v \in T_x\mathcal{M}} \left(\langle \mathrm{grad} f(x), v \rangle_x + \frac{\mu}{2} \|v\|_x^2 \right).$$

Cauchy-Schwarz implies that v is of the form $v = -c \,\mathrm{grad}\, f(x)$, $c \geq 0$. Substitution of $v = -c \,\mathrm{grad}\, f(x)$ gives

$$f^* \geq f(x) + \min_{c \geq 0} g(c) \tag{79}$$

$$g(c) = -c \|\,\mathrm{grad}\, f(x)\|^2 + c^2 \frac{\mu}{2} \|\,\mathrm{grad}\, f(x)\|^2 \tag{80}$$

Differentiation of $g(c)$ and setting the result equal to zero gives

$$-\|\,\mathrm{grad}\, f(x)\|^2 + c\mu \|\,\mathrm{grad}\, f(x)\|^2 = 0.$$
$$c = \frac{1}{\mu}.$$

Note that indeed $c > 0$ since $\mu > 0$. We obtain

$$\min_{c \geq 0} g(c) = -\frac{1}{2\mu} \|\,\mathrm{grad}\, f(x)\|^2 \tag{81}$$

Substituion of Equation (81) into (79) results into Equation (77). □

The following theorem we state without proof (see Ref. [11]).

Theorem 3.10 (Grönwall's inequality)**.** *Let $I \subset \mathbb{R}$ denote an interval on the real line of the form $[a, \infty)$, $[a, b]$ or $[a, b)$ with $a < b$. Let $\beta(t)$ be real-valued continuous and let $u(t)$ be real-valued continuous differentiable on I. If*

$$u'(t) \leq \beta(t) u(t), \quad \forall t \in I,$$

then

$$u(t) \leq u(a) e^{\int_0^t \beta(s) \mathrm{d}s}, \quad \forall t \in I.$$

Now we are ready to prove Theorem 3.8.

Proof of Theorem 3.8. Let us differentiate $f(x(t)) - f^*$ with respect to t.

$$\frac{\mathrm{d}}{\mathrm{d}t}(f(x(t)) - f^*) = \mathrm{D}f(x(t))[x'(t)] = \langle \mathrm{grad} f(x(t)), -\,\mathrm{grad}\, f(x(t)) \rangle_{x(t)} = -\|\mathrm{grad} f(x(t))\|^2$$

Using Polyak-Lojasiewicz, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}(f(x(t)) - f^*) = -\|\mathrm{grad} f(x(t))\|^2 \leq -2\mu(f(x(t)) - f^*).$$

Using Grönwall's inequility on the interval $[0, \infty)$ with $\beta(t) = -2\mu$ and $u(t) = f(x(t)) - f^*$, we obtain

$$f(x(t)) - f^* \leq e^{-2\mu t}(f(x(0)) - f^*).$$

□

4 Numerical Simulations

At the start of this chapter, we discuss the performance of Algorithm 2 by applying it to numerous problems. In Section 4.5, we will introduce a new algorithm which is developed to fit a robotic arm on a given curve. This algorithm is then further tested and improved in Section 4.6 and Section 4.7. The algorithms discussed in this chapter are implemented in Python. The code can be found on my GitHub page: https://github.com/DeMilder/Thesis-Final-Code.git.

4.1 Rotation of vectors in $\mathbb{R}^2$

Consider the problem of rotating a vector $x \in \mathbb{R}^n$ towards a vector $y \in \mathbb{R}^n$, where x and y are not the zero vector. Without loss of generality, let us assume that $\|x\|_2 = \|y\|_2$. We want to find $R \in \mathrm{SO}(n)$ such that $Rx = y$. For the case that $\|x\|_2 \neq \|y\|_2$, we can rescale the vectors x and y by defining $\hat{x} = \frac{x}{\|x\|_2}$ and $\hat{y} = \frac{y}{\|y\|_2}$, such that $\|\hat{x}\|_2 = 1 = \|\hat{y}\|_2$. Then, we want to find $R \in \mathrm{SO}(n)$, such that $R\hat{x} = \hat{y}$. We will refer to this problem as the vector-rotation problem.

To solve this problem we can use Algorithm 2. Therefore, let the loss function f be defined as

$$\boxed{f(R) = \|Rx - y\|_2^2.} \tag{82}$$

Then the problem can be described as finding $R^* \in \mathrm{SO}(n)$ such that

$$R^* = \underset{R \in \mathrm{SO}(n)}{\operatorname{argmin}} f(R). \tag{83}$$

The minimum of f at R^* is a unique minimum and R^* fulfils $R^*x = y$.

To perform Algorithm 2, we need to find the Euclidean gradient of $f(R)$. We claim the following.

$$\operatorname{Grad} \|Rx - y\|_2^2 = 2(Rx - y)x^T. \tag{84}$$

Proof.

$$\begin{aligned}
\operatorname{Grad} \|Rx - y\|_2^2 &= \operatorname{Grad}(Rx - y)^T(Rx - y) \\
&= \operatorname{Grad}\left(x^T R^T R x - y^T R x - x^T R^T y + y^T y\right) \\
&= \operatorname{Grad}(x^T R^T R x) - \operatorname{Grad}(y^T R x) - \operatorname{Grad}(x^T R^T y)
\end{aligned}$$

Considering each term separately by differentiating element-wise gives (see Definition 3.1) the following. For the first term, we have

$$x^T R^T R x = \sum_{m=1}^n \sum_{k=1}^n R_{mk} x_k \sum_{l=1}^n R_{ml} x_l,$$

therefore

$$\begin{aligned}
\frac{\partial}{\partial R_{ij}} x^T R^T R x &= x_j \left(\sum_{l=1}^n R_{il} x_l\right) + \left(\sum_{k=1}^n R_{ik} x_k\right) x_j, \\
\frac{\partial}{\partial R_{ij}} x^T R^T R x &= 2\left(\sum_{k=1}^n R_{ik} x_k\right) x_j,
\end{aligned}$$

therefore

$$\operatorname{Grad}_R(x^T R^T R x) = 2Rxx^T.$$

For the second term, we have

$$\begin{aligned}
y^T R x &= \sum_{k=1}^n y_k \sum_{l=1}^n R_{kl} x_l, \\
y^T R x &= \sum_{k=1}^n \sum_{l=1}^n y_k R_{kl} x_l,
\end{aligned}$$

therefore

$$\frac{\partial}{\partial R_{ij}} y^T R x = y_i x_j,$$

therefore

$$\mathrm{Grad}_R(y^T R x) = y x^T.$$

For the third term, we have

$$x^T R^T y = \sum_{k=1}^{n} x_k \sum_{l=1}^{n} R_{lk} y_l,$$
$$x^T R^T y = \sum_{k=1}^{n} \sum_{l=1}^{n} x_k R_{lk} y_l,$$

therefore

$$\frac{\partial}{\partial R_{ij}} x^T R^T y = x_j y_i,$$

therefore

$$\mathrm{Grad}_R(x^T R^T y) = y x^T.$$

Substitution gives

$$\mathrm{Grad}\, \|Rx - y\|_2^2 = 2(Rx - y)x^T.$$

□

As a matrix-to-vector operator $\mathrm{vec} : \mathrm{M}_n(\mathbb{R}) \to \mathbb{R}^{n^2}$, we choose to order the matrix element row-wise into a vector in $\mathbb{R}^{n^2}$. Now, we have all the ingredients to perform Algorithm 2.

Algorithm 2 is implemented in Python, where we used the Numpy and Scipy libraries. The vectors are represented as Numpy arrays and we calculate the matrix exponential by using the scipy.linalg.expm() function. In this section, we only consider vector-rotation problems in $\mathbb{R}^2$. Higher dimensional vector-rotation problems are discussed in Section 4.3.

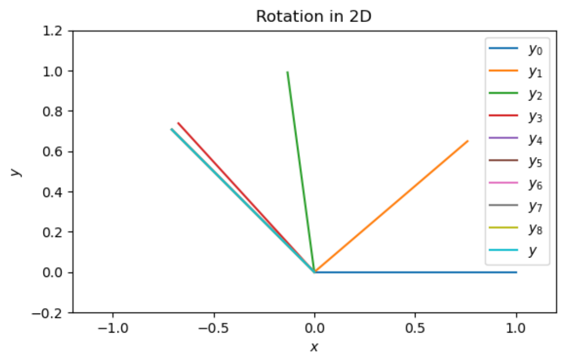
4.1.1 Testcase 1: $y = \frac{1}{2}\sqrt{2}(-1, 1)^T$

Let $R_0 = I_{2\times 2}$, $x = (1, 0)^T$ and $y = \frac{1}{2}\sqrt{2}(-1, 1)^T$. We set the learning rate $\eta = 1$ and use our Python implementation of Algorithm 2 to solve this problem. The results are given in Figure 9. In the left figure, the $y_t = R_t x$ vectors, where t denotes the iteration index, are illustrated. Note that $y_0 = Ix = x$. We see that the $y_t = R_t x$ rotates in the positive direction toward y. In the right figure, the value of the loss function $f(R_t)$ for every iteration step t is given. After 6 iterations, we converge to computer precision. Note that we used double-precision floating point numbers (float64). Similar results are obtained when considering $y = -\frac{1}{2}\sqrt{2}(-1, 1)^T$. However, in that case, $y_t = R_t x$ rotates in the negative direction towards y.

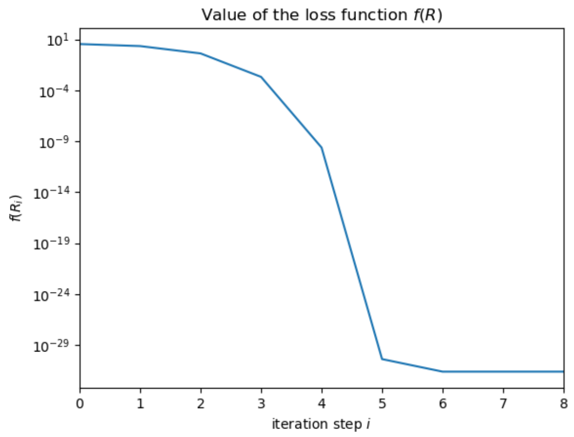
4.1.2 Testcase 2: $y = (-1, 0)^T$

Now, suppose $R_0 = I_{2\times 2}$, $x = (1, 0)^T$ and $y = -x$. Does the algorithm for this case rotate in the positive or the negative direction towards y? Calculating the Euclidean gradient gives

$$\mathrm{Grad}\, f(R_0) = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix}.$$



(a) Graphical representation of the $y_t = R_t x$ vector, where t is the iteration index.



(b) Value of the loss function $f(R)$, as given in Equation (82), at every iteration step.

Figure 9: Solving the vector-rotation-problem in $\mathbb{R}^2$. Here $R_0 = I_{2\times 2}$, $x = (1,0)^T$, $y = \frac{1}{2}\sqrt{2}(-1,1)^T$ and $\eta = 1$. We see that after 6 iteration steps, we reach computer precision.

Calculating the Riemannian gradient according to Equation (62) gives

$$\operatorname{grad} f(R_0) = \mathrm{P}_{T_I \mathrm{SO}(2)} \operatorname{Grad} f(R_0) = 0_{2\times 2}.$$

Note that $\operatorname{Grad} f(R_0)$ is a symmetric matrix. Therefore, projecting this matrix onto $T_I \mathrm{SO}(2) = \mathcal{S}_{\text{skew}}(2)$ gives the zero matrix $0_{2\times 2}$. So the Riemannian gradient $\operatorname{grad} f(R_0)$ is equal to the zero matrix. Retracting the zero matrix onto the tangents space gives R_0 (see the first property of Definition 3.3). So in this case, the algorithm keeps returning R_0 and does not converge to the solution at all.

R_0 is a critical point, i.e., $\operatorname{grad} f(R_0)$ is equal to the zeros matrix, but R_0 is not a solution to the problem. Furthermore, note that R_0 is an unstable critical point because when we slightly change R_0, the algorithm moves away from R_0 and converges to the solution. Therefore in general, when we encounter such an unstable critical point, say $\hat{R}$, we can slightly perturb $\operatorname{grad} f(\hat{R})$, such that it is not equal to the zero matrix anymore. By doing so, We observed that the algorithm moves away from $\hat{R}$ and eventually converges to the solution R^*.

4.1.3 Testcase 3: $y \in S^1$

Let $R_0 = I_{n\times n}$, $x_0 = (1,0)^T$, $\eta = 1$ and $y = (\cos(\theta), \sin(\theta))^T$ for $\theta \in (-\pi, \pi]$. We solve this problem for multiple values of θ, using Algorithm 2. Note that the exact solution to the problem is the rotation matrix

$$R^* = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

We say that the algorithm converged at iteration t^* if t^* is the smallest integer such that the inequality $f(R_{t^*}) < \epsilon$ is true, where $\epsilon \in \mathbb{R}_{>0}$ is small. Let $\epsilon = 10^{-15}$, the number of iterations needed until convergence t^* for different values of θ is given in Figure 10. We see that for values of θ close to 0, the algorithm converges rapidly to the solution since we only need to rotate the vector x over a small angle θ to obtain y. For values of θ close to π or $-\pi$, the number of iteration steps needed before convergence is larger, since we need to rotate x over a larger angle θ and the Riemannian gradient will be approximately equal to the zero matrix during the first iteration steps.

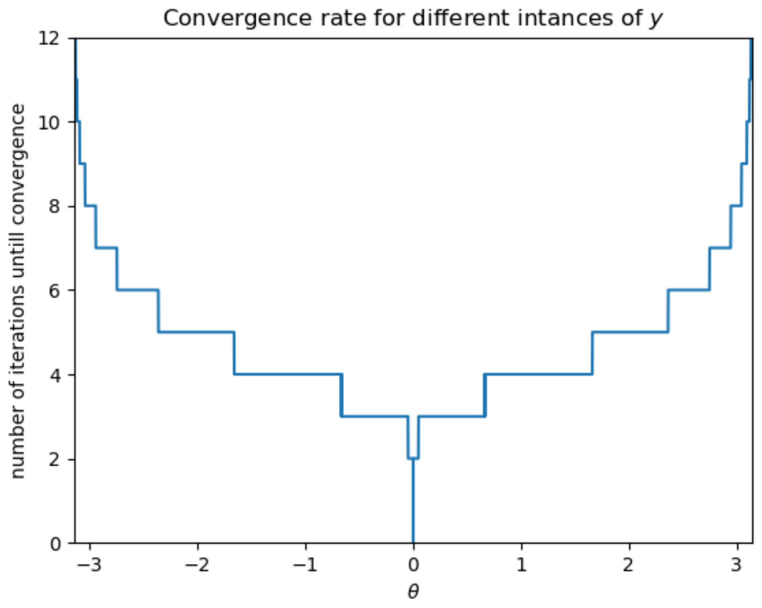


Figure 10: Number of iterations until convergence. Here, we say that this number is equal to the smallest t such that $f(R_t) < 10^{-15}$, where f denotes the loss function.

4.2 Rotating towards a perturbed figure

Suppose that we have a number of points $x^i \in \mathbb{R}^n$, where $i \in \{1, 2, ..., p\}$. Let us call the collection of these points a figure, which is denoted as x. Suppose we want to rotate this figure x so that it best fits the points $\widehat{y}^i \in \mathbb{R}^n, i \in \{1, 2, ..., p\}$. The $\widehat{y}^i$'s can be expressed as

$$\widehat{y}^i = y^i + \epsilon^i, \quad i \in \{1, 2, ..., p\}.$$

Here $y^i = R^* x^i$, where $R^* \in \mathrm{SO}(n)$ is the true rotation matrix and ϵ^i is some perturbation vector. Let us assume that the ϵ^i's are sampled from a multivariate normal distribution $\mathcal{N}(0_n, \sigma^2 I_{n\times n})$ with a mean equal to the zero vector and with a diagonal covariance matrix $\sigma^2 I_{n\times n}$. For this problem consider the following loss function $f(R)$

$$\boxed{f(R) = \frac{1}{p}\sum_{i=1}^{p} \|Rx^i - \widehat{y}^i\|_2^2}, \tag{85}$$

and its Euclidean gradient Grad $f(R)$

$$\mathrm{Grad}\, f(R) = \frac{1}{p}\sum_{i=1}^{p} (Rx^i - \widehat{y}^i)(x^i)^T. \tag{86}$$

This problem can be described as a minimization problem, where we want to minimize f for $R \in \mathrm{SO}(n)$. Therefore we can apply Algorithm 2.

As an example consider the parametric curve $\gamma : [0, 1) \to \mathbb{R}^2$, defined as

$$\gamma(t) = \begin{pmatrix} 5\sin(4\pi t) \\ 10\sin(2\pi t) \end{pmatrix}, \quad t \in [0, 1).$$

Let the figure $x = \{x^i : i \in \{1, 2, ..., p\}\}$ be a collection of $p = 10,000$ points sampled from $\gamma(t)$, such that

$$x^i = \gamma\left(\frac{i}{p}\right), \quad i \in \{1, 2, ..., p\}.$$

Let the true rotation R^* be a 90 degree rotation around the origin in the positive direction, i.e.,

$$R^* = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

We have a number of p perturbated fitting points $\widehat{y}^i = R^* x^i + \epsilon^i$, $i \in \{1, 2, ..., p\}$. The ϵ^i's are samples from the multivariate normal distribution $\mathcal{N}(0_2, \sigma^2 I_{2\times 2})$, with $\sigma = 1$. Suppose R^* is unknown, then we can recover R^* from the $\widehat{y}^i$'s using Algorithm 2. After 10 iterations, with a learning rate $\eta = 0.01$ and initial guess $R_0 = I_{2\times 2}$, we obtained the matrix

$$R_{10} = \begin{pmatrix} 0.00053 & -1.00000 \\ 1.00000 & 0.00053 \end{pmatrix}.$$

We indeed see that R_{10} accurately approximates R^*.

More results are visualized in Figures 11 and 12. In Figure 11, the red dots represent the perturbed data points $\widehat{y}^i$ for $i \in \{1, ..., p\}$. The collection of points $y_t^i = R_t x^i$, $i \in \{1, ..., p\}$, at iteration steps t are plotted as curves and labeled as y_t. Here, y_0 corresponds to the initial collection of points $y_0^i = x^i$, $i \in \{1, ..., p\}$. We see that after iteration step $t = 6$, y_t overlaps with the unperturbed figure y.

In Figure 12, the loss function f (see Equation (85)) decreases rapidly during the first 3 iteration steps and reaches a value of $f = 1.25$ after the sixth iteration step. By the law of large numbers, we have that

$$\frac{1}{p}\sum_{i=1}^{p} \|R^* x^i - \widehat{y}^i\|_2 = \frac{1}{p}\sum_{i=1}^{p} \|\epsilon^i\|_2 \xrightarrow{p\to+\infty} \mathbb{E}[\|\epsilon\|_2] = \sqrt{\frac{\pi}{2}},$$

with rate $O(\frac{1}{p})$. Here, we recognize that $\|\epsilon\|_2$ is distributed according to the Rayleigh distribution with scale parameter $\sigma = 1$, which has a mean of $\sqrt{\pi/2} \approx 1.25$ (see Ref. [1]). This also validates that Algorithm 2 gives the correct solution.

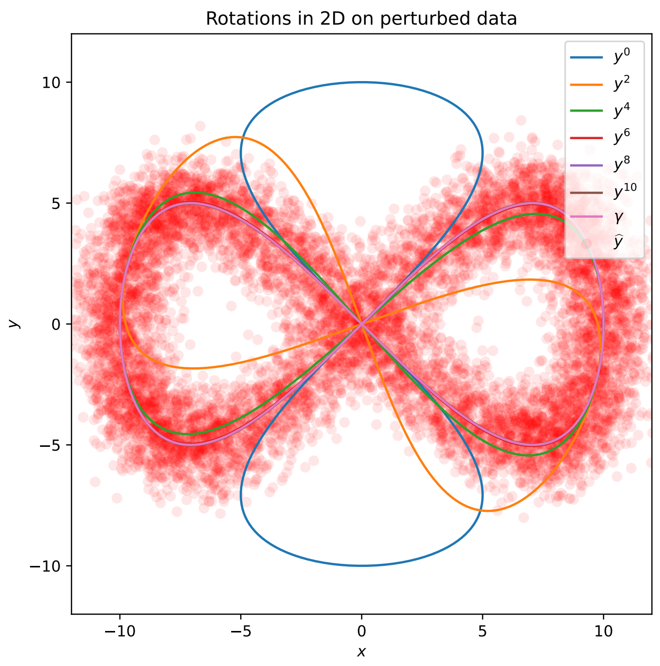


Figure 11: Graphical representation of the y_t's at different iteration steps t. y is the desired figure. $\hat{y}$ is the perturbated version of y.

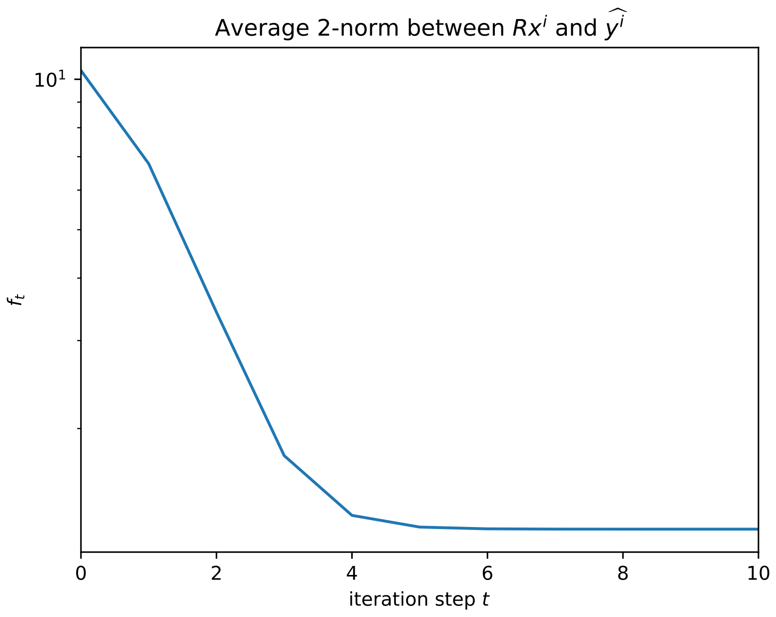


Figure 12: f as given in Equation (85).

4.3 Rotation of vectors in $\mathbb{R}^n$

In the previous sections, we looked at the rotations of vectors and figures in $\mathbb{R}^2$. We have minimized a loss function $f(R)$ for $R \in \mathrm{SO}(n)$. In this section, we again look at problem (83), where we use Equation (82) as our loss function $f(R)$, and we check the performance of the algorithm for different dimensions n.

Let $R_0 = I_{n\times n}$ and let $x = (1, 0, 0, ...)^T$ be the unit basis vector. We set the learning rate $\eta = 1$. We can choose an arbitrary y on the sphere centred at the origin. In general, one can pick a random point on a hypersphere of arbitrary dimension n by generating n Gaussian random variables $g_1, g_2, ..., g_n$. Then the distribution of the vectors

$$\frac{1}{\sqrt{g_1^2 + g_2^2 + ...g_n^2}}(g_1, g_2, ..., g_n)^T \tag{87}$$

is uniform over the surface S^{n-1}.[14] We perform the algorithm $m = 10,000$ times in $\mathbb{R}^n$ where $n = 2, 3, 5, 10, 20$. So, in total, we perform the algorithm $50,000$ times. For every instance, we will use a different end vector y which is sampled from the uniform distribution over the surface S^{n-1} (see Equation (87)). After every iteration step t, the value of f, as given in Equation (82), is calculated. These values are then averaged over the m instances. The results are given in Figure 13.

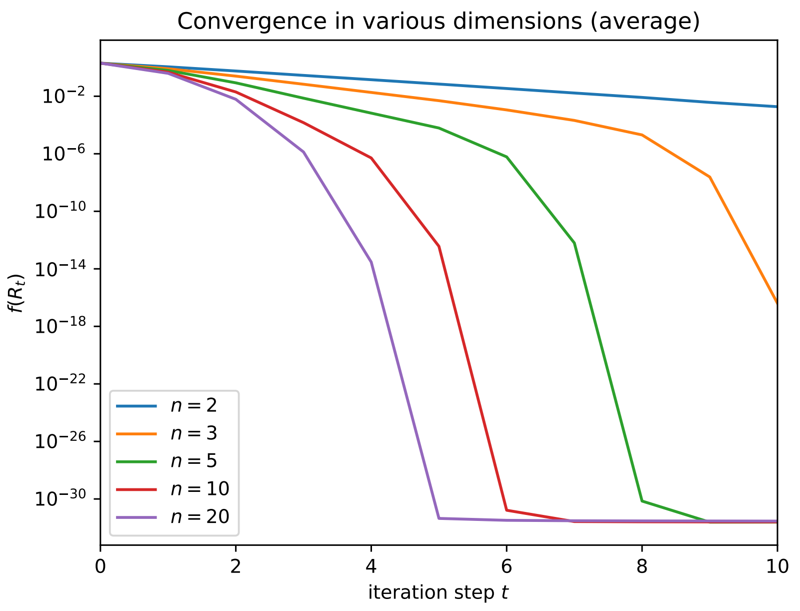


Figure 13: The average value of the loss function $f(R_t)$ after every iteration step t is plotted for different dimensions. The results are averaged over $m = 10,000$ instances.

Figure 13 gives some unexpected results. In Section 4.1, we have seen that in $\mathbb{R}^2$ the algorithm often already converges after 6 iteration steps. However looking at Figure 13, it looks like that, on average, we are still far from convergence after 6 iteration steps. It also looks like we need fewer iteration steps to reach convergence if the dimensionality of the problem is larger. For $n = 10$ and $n = 20$, we see the same convergence rates as the $n = 2$ instance illustrated in Figure 9.

In Section 4.1, we have seen that for $n = 2$ the convergence rate of the loss function can depend on the angle θ over which we have to rotate x to obtain y. It can take a lot of iteration steps before the value of the loss function converges especially when $\theta \approx \pi$ or $\theta \approx -\pi$. Therefore, the average value of the loss function $f(R_t)$ at t gives a misleading image of the performance of the algorithm.

Let us take a look at Figure 14. There we see a histogram of the number of iteration steps needed until the convergence of $f(R_t)$. Again, we say that the algorithm has converged to the solution at iteration t^* if t^* is the smallest integer such that $f(R_t^*) < \epsilon$ for some small $\epsilon \in \mathbb{R}_{>0}$. Let us choose $\epsilon = 10^{-15}$. We see that the variation in the value for t^* is larger for $n = 2$ than for the higher value of n. For $n = 20$, t^* seems to be equal to 4 for almost every instance of y. This seems reasonable, since the probability that we select a point y on the unit sphere which lies very close to x or $-x$ decreases if we increase the dimensionality of the sphere.

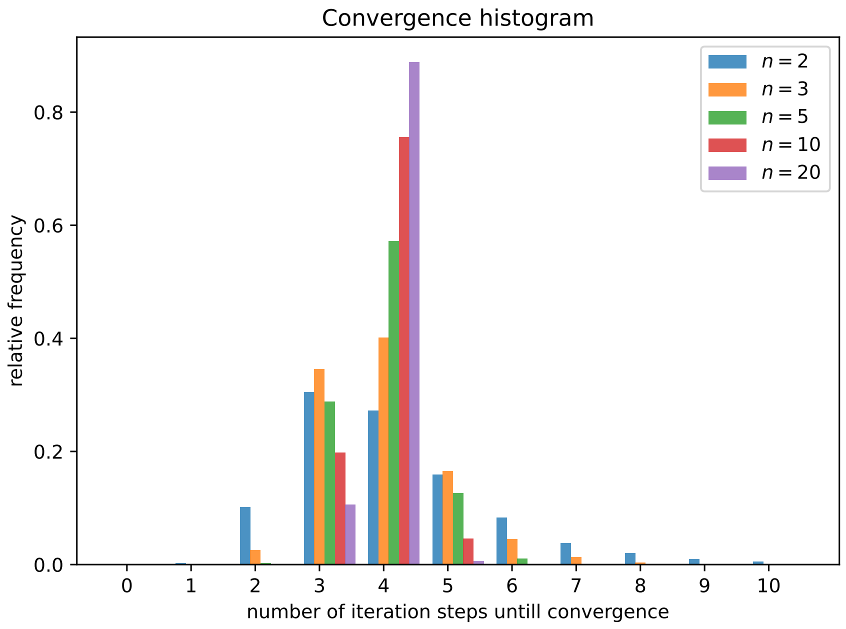


Figure 14: Histogram of the number of iteration steps t^* needed until $f^{t^*} < 10^{-15}$.

In Figure 15 we have plotted the median of the loss function $f(R_t)$ of the $m = 10,000$ iterations for every dimension. When we look at the median, we see that the convergence behaviour is the same in every dimension. The edges of the coloured area mark the minimum and the maximum observed value for $f(R_t)$. Again we see that for $n = 2$ the value for $f(R_t)$ can vary significantly between different instances of y, whereas for $n = 20$ this difference in $f(R_t)$ is small.

In short, when looking at the median, we see that the number of iterations needed before convergences t^* is often equal to 4 and does not depend on the dimensionality of the problem. When the dimensionality of the problem is low $n < 5$, we see that t^* can vary significantly between different instances for y. Since it is more likely for the algorithm to get stuck in an unstable critical point (see Subsection 4.1.2).

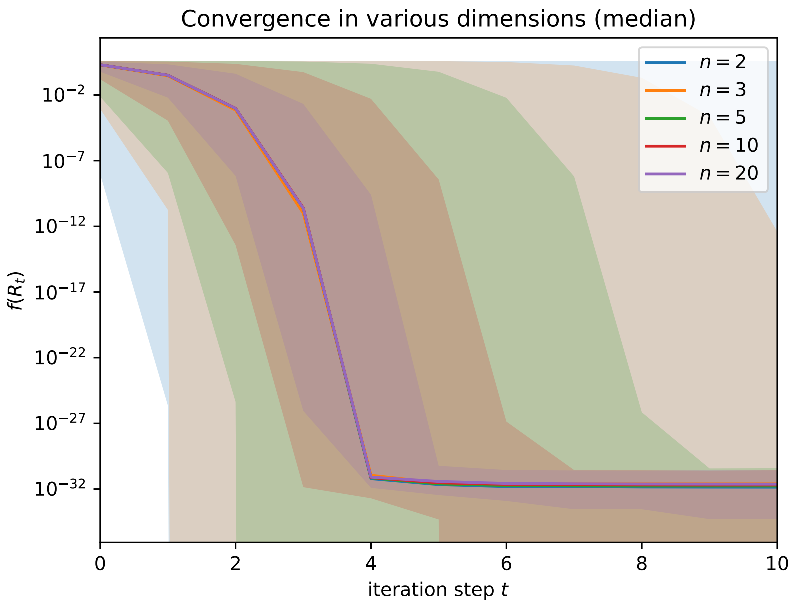


Figure 15: Median value of the loss function $f(R_t)$ at iteration t for different dimensions. The edges of the coloured area mark the minimum and maximum observed value for $f(R_t)$ ($m = 10,000$)).

4.4 Different Learning rates

In this section, we will solve the vector-rotation problem (see Equations (82) and (83)) using different values for the learning rate η. In the previous section, we discovered that the convergence rate of Algorithm 2 does not heavily depend on the dimensionality of the problem. Furthermore, we have seen that when the dimensionality n of the vector-rotation problem is high, the variance in the number of iteration steps needed until convergence between different given instances of y is low because we are less likely to select y such that R_t gets stuck in an unstable critical point. However, the computational cost to perform one iteration step increases as n increases. Therefore, let us take $n = 5$ to test the performance of the algorithm for different learning rates η because the variance in the convergence rate between multiple instances of y is low with respect to smaller values of n and the computational cost of performing one iteration step is small with respect to higher values of n.

Let us test the performance of the algorithm for $\eta = 0.1, 0.5, 0.9, 1.0, 1.1, 1.5, 2.0$. As in the previous section, we perform the algorithm for $m = 10,000$ instances for y. In Figure 16, the median value of the loss function $f(R_t)$ at iteration step t is plotted. The upper edges of the coloured areas indicate the maximum observed value for $f(R_t)$ and the lower edges indicate the minimum value for $f(R_t)$. We see that we get the fasted convergence rate for $\eta = 1$. When choosing lower values for η, the convergence rate becomes gradually worse as we move further away from $\eta = 1$. When choosing a higher value for η, the convergence rate also becomes worse and for $\eta = 2$ we even see that algorithm becomes unstable.

We have seen that in general we only need 4 or 5 iteration steps to converge to computer precision. Therefore let us take a closer look at these first 5 iteration steps. In Figure 17, we see that the reduction in the loss function after the first iteration step is larger for $\eta = 1.5$ than for $\eta = 1.1$ and $\eta = 1.0$. After the second iteration step $\eta = 1.1$ has the fastest convergence rate. We have to note that Figure 17 is based on the median and we can see that the value of $f(R_t)$ can vary highly between different instances for y.

Let us compare the value of the loss function f for different values of the learning rate η to the case when $\eta = 1$. Note that we used the same $m = 10,000$ instances for y to test the different learning rates. Let us denote these instances as y_k for $k \in \{1, 2, ..., m\}$. Let $f(R_t, \eta, y_k)$ be the value of the loss function f at

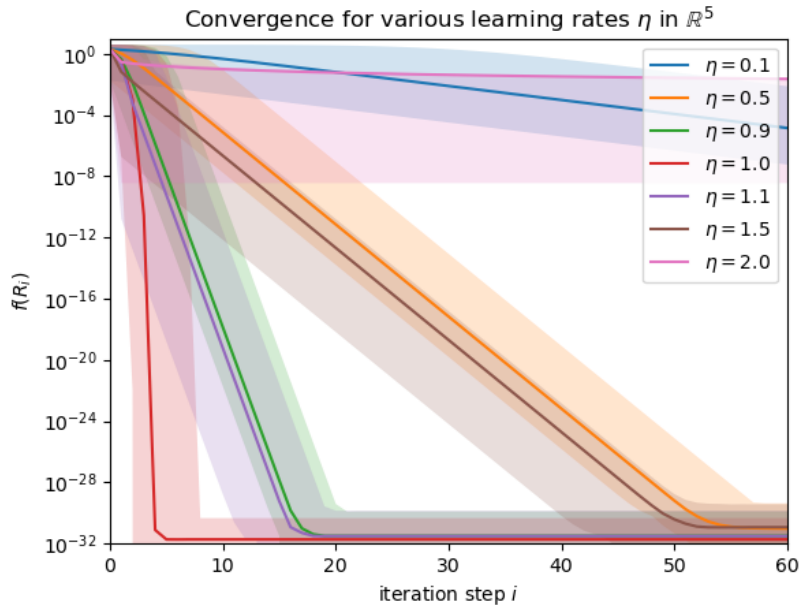


Figure 16: Loss function f for learning rates $\eta = 0.1, 0.5, 0.9, 1.0, 1.1, 1.5, 2.0$. $n = 5$, $m = 10,000$.

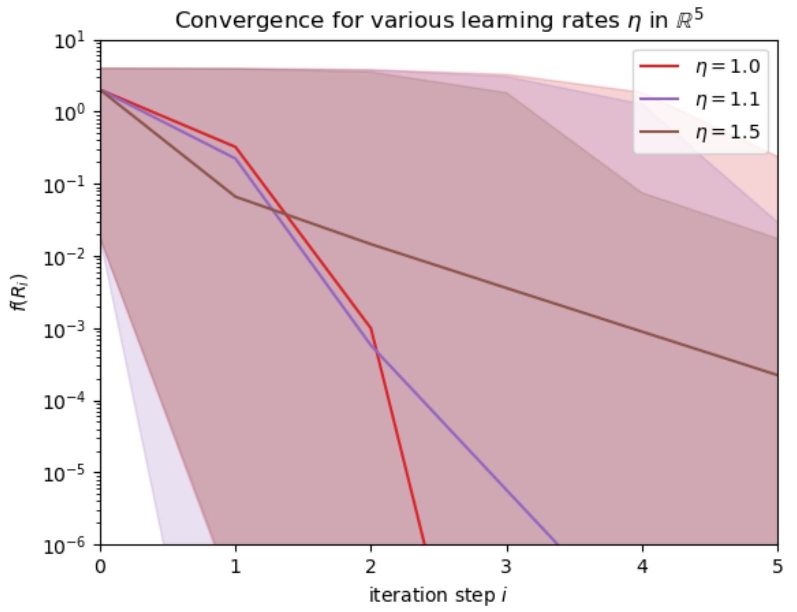


Figure 17: Caption

iterations step t where we used a learning rate of η. So we want to look at the difference between $f(R_t, \eta, y_k)$ and $f(R_t, \eta = 1, y_k)$ for every instance y_k. This difference for iteration step $t = 1$ is calculated for the $m = 10,000$ instances for y_k and these data points are visualized as boxplots in Figure 18 for the different values of η.

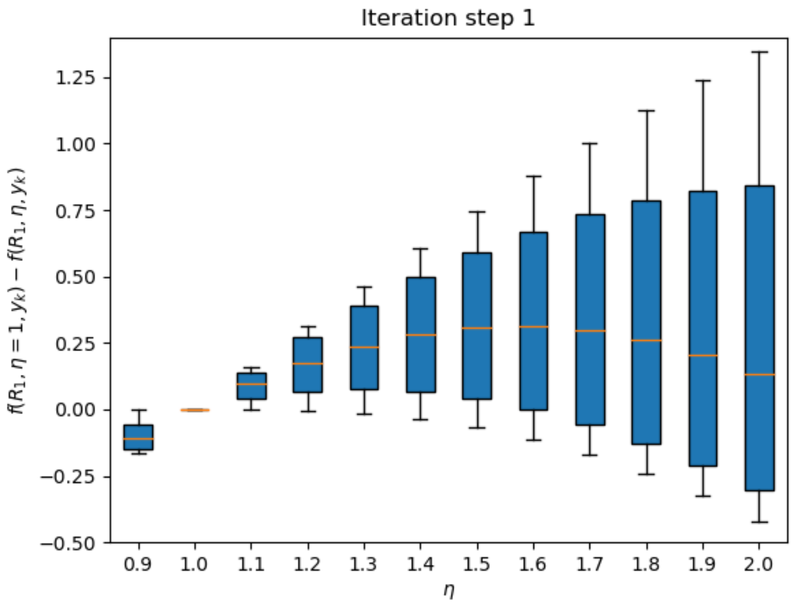


Figure 18: Boxplot where we compare the value of the loss function $f(R_t, \eta, y_k)$ at iteration step $t = 1$ for different learning rates to the value of the loss function when using a learning rate of $\eta = 1$. On the y-axis the quantity $f(R_1, \eta = 1, y_k) - f(R_1, \eta, y_k)$ is plotted.

First note that all the data points lie at 0 for $\eta = 1$, because we look at the difference between $f(R_t, \eta, y_k)$ and $f(R_t, \eta = 1, y_k)$. In Figure 18, we can see that choosing a learning rate between 1 and 1.5 gives a greater decrease in the loss function during the first iteration step than for the $\eta = 1$ case. When using a learning rate larger than 1.5, the variance in the data points increases. So the convergence could be better or worse, which highly depends on the given y_k instance. This can be explained by the fact that some instances for y_k lie very close to $-x$. Therefore, the algorithm can get stuck in an unstable critical point. A larger learning rate η can then be beneficial to escape this unstable critical point faster, especially during the first iteration step. (see also subsection 4.1.2). At higher iteration steps, i.e., $t > 1$, we generally see that $\eta = 1$ gives the greatest decrease in the value of the loss function f (see Figure 16).

To conclude this section, we have found that choosing $\eta = 1$ gives the fastest convergence rate. However, one can opt to choose a slightly larger value for η during the first iteration step, which can help the algorithm escape from an unstable critical point.

4.5 Fitting a Robotic Arm

In the previous sections, we have mainly discussed the vector-rotation problem. We have looked at this problem in multiple dimensions and applied Algorithm 2 for different values of the learning rate η. In this section, we will extend our view and consider another problem, which is inspired by the inverse kinematics problem in robotics [8]. Our goal is to find a configuration of a robotic arm that flows a given curve $\gamma : \mathbb{R} \to \mathbb{R}^n$. We will call this problem the robotic-arm problem. Here n denotes the dimensionality of this problem. The robotic arm is made out of k links. These links are connected by a point of rotation, which are called joints. We assume that these joints can be freely rotated, like a ball-and-socket joint. This stands in contrast to a hinge joint, which only allows movement in a two-dimensional plane. Furthermore, we assume that the curve γ is continuously differentiable, i.e., $\gamma \in C^1$. Note that we look at the robotic arm problem in $\mathbb{R}^n$ so we do not restrict ourselves to $n = 2$ or $n = 3$.

To describe the robotic-arm problem in further detail, consider a piecewise linear function $\alpha : [0, k] \to \mathbb{R}^n$, where $k \in \mathbb{N}$ which is equal to the number of links. This function is piecewise linear over the intervals $[0, 1], [1, 2], ..., [k - 1, k]$. Let α_i be the linear function which describes α on the interval $[i - 1, i]$, for $i \in \{1, 2, ..., k\}$, i.e.,

$$\alpha(t) = \begin{cases} \alpha_1(t), & \text{for } t \in [0,1]. \\ \alpha_2(t), & \text{for } t \in [1,2]. \\ \vdots & \\ \alpha_k(t), & \text{for } t \in [k-1,k]. \end{cases} \tag{88}$$

The function $\alpha(t)$ is the curve which describes the robotic arm. $\alpha(0), \alpha(1), ..., \alpha(k-1)$ describe the positions of the joints. Between these joints are the links, where the ith link is described by $\alpha_i(t)$ for $t \in [i - 1, i]$. Let $x_1, x_2, ..., x_k \in R^n$ be given vectors, where x_i represents the initial orientation of the ith link with respect to the ith joint $\alpha(i)$. Then the α_i's can be defined as

$$\begin{aligned} \alpha_1(t) = & \quad tR_1x_1, & \text{for } t \in [0,1], \\ \alpha_2(t) = & \quad \alpha_1(1) + (t-1)R_2x_2, & \text{for } t \in [1,2], \\ \vdots & & \\ \alpha_i(t) = & \quad \alpha_{i-1}(i-1) + (t-(i-1))R_ix_i, & \text{for } t \in [i-1,i], \\ \vdots & & \\ \alpha_k(t) = & \quad \alpha_{k-1}(k-1) + (t-(k-1))R_kx_k, & \text{for } t \in [k-1,k], \end{aligned}$$

where we take $\alpha_0(0) = 0_n$.

The path on which we want to fit the robotic arm describe by α is given as a continuously differentiable curve

$$\begin{aligned} \gamma : \mathbb{R} &\to \mathbb{R}^n, \\ \tau &\to \gamma(\tau), \end{aligned}$$

for which $\gamma(0) = 0$.

Next, let there be a continuous injective mapping

$$\begin{aligned} m : &[0,k] \to \mathbb{R}, \\ &t \to \tau, \end{aligned}$$

The robotic-arm problem can be described as follows: find the continuous (injective) mapping $m : [0, k] \to \mathbb{R}$ and the matrices $R_1, R_2, ...R_k \in \mathrm{SO}(n)$, such that

$$\frac{1}{k}\int_0^k \|\alpha(t) - \gamma(m(t))\|^2 \mathrm{d}t \tag{89}$$

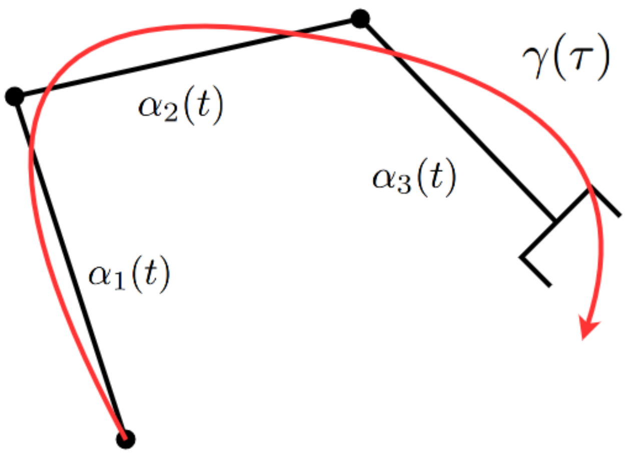


Figure 19: Illustration of a robotic arm described by $\alpha(t)$, where the number of links is $k = 3$. $\gamma(\tau)$ is the desired path.

is minimal. Note that this is a minimization problem over $\mathrm{SO}(n)^k \times C^0$, where C^0 denotes the set of continuous functions.

Our goal is to find or obtain a very good approximation for these matrices $R_1, ..., R_k$, because these matrices tell us how to rotate each joint of the robotic arm. We are less interested in an exact expression for the mapping m. Furthermore, minimizing Equation (89) for m is rather difficult. Therefore, let us make an assumption for the shape of m. We assume that m is piecewise linear over the intervals $[0, 1]$, $[1, 2]$, ..., $[k-1, k]$, i.e.

$$m(t) = \begin{cases} t\tau_1, & \text{for } t \in [0,1], \\ \tau_1 + (t-1)(\tau_2 - \tau_1), & \text{for } t \in [1,2], \\ \vdots \\ \tau_{i-1} + (t-i)(\tau_i - \tau_{i-1}), & \text{for } t \in [i-1,i], \\ \vdots \\ \tau_{k-1} + (t-(k-1))(\tau_k - \tau_{k-1}), & \text{for } t \in [k-1,k], \end{cases} \tag{90}$$

where $\tau_0 = m(0) = 0$, $\tau_1 = m(1)$, ..., $\tau_k = m(k)$. Note that $m(t)$ rescales the τ parameter of the $\gamma(\tau)$ function over the intervals $[\tau_0, \tau_1]$,$[\tau_1, \tau_2]$, ..., $[\tau_{k-1}, \tau_k]$ linearly, but with a different scale parameter for each interval. Assuming this form, m is completely defined by a finite set of value $\tau_1, \tau_2, ..., \tau_k$. So now we can minimize Equation (89) for $R_1, R_2, ..., R_k$ and $\tau_1, \tau_2, ..., \tau_k$ instead. For notation purposes let us define $g = \gamma(m(t))$ and let us write

$$g(t) = \begin{cases} g_1(t), & \text{for } t \in [0,1], \\ g_2(t), & \text{for } t \in [1,2], \\ \vdots \\ g_i(t), & \text{for } t \in [i-1,i], \\ \vdots \\ g_k(t), & \text{for } t \in [k-1,k]. \end{cases} \tag{91}$$

By substituion of $m(t)$ one can define the g_i's as

$$g_i(t) = \gamma\left(\tau_{i-1} + (t-(i-1))(\tau_i - \tau_{i-1})\right), \quad \text{for } i \in \{1, ..., k\}.$$

Splitting the integral of Equation (89) and writing it terms of the g_i's gives

$$\frac{1}{k}\int_0^k \|\alpha(t)-\gamma(m(t))\|^2 \mathrm{d}t = \frac{1}{k}\sum_{i=1}^{k}\int_{i-1}^{i} \|\alpha_i(t)-g_i(t)\|^2. \tag{92}$$

Note that we transformed the minimization problem (89) over $\mathrm{SO}(n)^k \times C^0$ into a minimization problem over $\mathrm{SO}(n)^k \times \mathbb{R}^k$. Next, let us discretize each integral by considering p subintervals of equal width. This gives

$$\frac{1}{k}\int_0^k \|\alpha(t)-\gamma(m(t))\|^2 \mathrm{d}t \approx \frac{1}{kp}\sum_{i=1}^{k}\sum_{j=1}^{p} \left\|\alpha_i\left((i-1)+\frac{j}{p}\right) - g_i\left((i-1)+\frac{j}{p}\right)\right\|^2.$$

Elaborating this expression in terms of the R_i's and τ_i's gives

$$\frac{1}{k}\int_0^k \|\alpha(t)-\gamma(m(t))\|^2 \mathrm{d}t \approx \frac{1}{kp}\sum_{i=1}^{k}\sum_{j=1}^{p} \left\|\alpha(i-1)+\frac{j}{p}R_i x_i - \gamma\left(\tau_{i-1}+\frac{j}{p}(\tau_i-\tau_{i-1})\right)\right\|^2.$$

So we found the following discrete minimization problem: Let k be the number of links and joints of the robotic arm. Let p be the number of integration points per link so that we have a total of kp integration points. Then find $R_1, R_2, ..., R_k \in \mathrm{SO}(n)$ and $\tau_1, \tau_2, ..., \tau_k \in \mathbb{R}$ such that

$$\boxed{f(R_1,...,R_k,\tau_1,...,\tau_k) = \frac{1}{kp}\sum_{i=1}^{k}\sum_{j=1}^{p} \left\|\alpha(i-1)+\frac{j}{p}R_i x_i - \gamma\left(\tau_{i-1}+\frac{j}{p}(\tau_i-\tau_{i-1})\right)\right\|^2} \tag{93}$$

is minimal. For notation let us define α_{ij} as

$$\alpha_{ij} = \alpha(i-1)+\frac{j}{p}R_i x_i \tag{94}$$

and g_{ij} as

$$g_{ij} = g_i\left((i-1)+\frac{j}{p}\right). \tag{95}$$

such that

$$\boxed{f(R_1,...,R_k,\tau_1,...,\tau_k) = \frac{1}{kp}\sum_{i=1}^{k}\sum_{j=1}^{p} \|\alpha_{ij}-g_{ij}\|^2.} \tag{96}$$

We can use the Riemannian gradient descent algorithm to calculate this minimum. Therefore, we need to specify Algorithm 1 further, such that it optimizes f over $\mathcal{M} = \mathrm{SO}(n)^k \times R^k$. To use Algorithm 1 we need to obtain the Riemannian gradient $\mathrm{grad}_x f$ for $x \in \mathcal{M}$. Therefore let us first calculate the Euclidean gradient of f with respect to R_i

$$\mathrm{Grad}_{R_i} f = \frac{1}{kp}\sum_{j=1}^{p}\frac{j}{p}\left(\alpha_{ij}-g_{ij}\right)x_i^T. \tag{97}$$

The Euclidean gradient of f with respect to τ_i is given by

$$\mathrm{Grad}_{\tau_i} f = -\frac{2}{kp}\left(\frac{j}{p}(\alpha_{ij}-g_{ij})^T g'_{ij} + \mathbb{1}_{i<k}\left(\frac{p-j}{p}(\alpha_{(i+1)j}-g_{(i+1)j})^T g'_{(i+1)j}\right)\right), \tag{98}$$

where

$$g'_{ij} = \gamma'\left(\tau_{i-1}+\frac{j}{p}(\tau_i-\tau_{i-1})\right).$$

For notation purposes, let us denote $\operatorname{Grad}_x f$ to be the Euclidean gradient $\operatorname{Grad}_x f$ at a point

$$x = (R_1, ..., R_k, \tau_1, ..., \tau_k) \in \mathcal{M},$$

i.e.,

$$\operatorname{Grad}_x f = (\operatorname{Grad}_{R_1} f, ..., \operatorname{Grad}_{R_k} f, \operatorname{Grad}_{\tau_1} f, ..., \operatorname{Grad}_{\tau_k} f).$$

Note that $\mathcal{M}$ can be seen as a submanifold of $\mathrm{M}_n(\mathbb{R})^k \times \mathbb{R}^n$. Therefore, to obtain the Riemannian gradient $\operatorname{grad}_x f$ we need to project the Euclidean gradient $\operatorname{Grad}_x f$ onto the tangent space $T_x \mathcal{M}$ of $\mathcal{M}$. Using Theorem 2.6, we can conclude that the tangent space $T_x \mathcal{M}$ of $\mathcal{M}$ at a point $x = (R_1, ..., R_2, \tau_1, ..., \tau_k)$ can be written as

$$\begin{aligned} T_x \left(\mathrm{SO}(n)^k \times R^k \right) &= T_{R_1} \mathrm{SO}(n) \times ... \times T_{R_k} \mathrm{SO}(n) \times T_{\tau_1} \mathbb{R} \times ... \times T_{\tau_k} \mathbb{R}, && (99) \\ &= T_{R_1} \mathrm{SO}(n) \times ... \times T_{R_k} \mathrm{SO}(n) \times \mathbb{R}^k. && (100) \end{aligned}$$

Therefore, we can write

$$\begin{aligned} \operatorname{grad}_x f &= (\mathrm{P}_{T_{R_1} \mathrm{SO}(n)} \operatorname{Grad}_{R_1} f, ..., \mathrm{P}_{T_{R_k} \mathrm{SO}(n)} \operatorname{Grad}_{R_k} f, \operatorname{Grad}_{\tau_1} f, ..., \operatorname{Grad}_{\tau_k} f), && (101) \\ &= (\operatorname{grad}_{R_1} f, ..., \operatorname{grad}_{R_k} f, \operatorname{grad}_{\tau_1} f, ..., \operatorname{grad}_{\tau_k} f). && (102) \end{aligned}$$

Note that $\operatorname{grad}_{\tau_i} f = \operatorname{Grad}_{\tau_i} f$ for $i \in \{1, ..., k\}$, since for these element we are working in $\mathbb{R}$. Next, we need to define a retraction method $\rho_x : T_x \mathcal{M} \to \mathcal{M}$, which maps an element of the tangent space $T_x \mathcal{M}$, in our case $-\eta \operatorname{grad}_x f$, onto the manifold $\mathcal{M}$. Here, $\eta \in \mathbb{R}_{>0}$ is the learning rate for the R_i's and $\sigma \in \mathbb{R}_{>0}$ is the learning rate for the τ_i's. We can do this by retracting element-wise, i.e.,

$$\rho_x(-\eta \operatorname{grad}_x f) = \left(\rho_{R_1}\left(-\eta \operatorname{grad}_{R_1} f\right), ..., \rho_{R_1}\left(-\eta \operatorname{grad}_{R_1} f\right), \tau_1 - \eta \operatorname{Grad}_{\tau_1} f, ..., \tau_k - \eta \operatorname{Grad}_{\tau_k} f \right).$$

Here, we define $\rho_{R_1} : T_{R_1} \mathrm{SO}(n) \to \mathrm{SO}(n)$ to be the Riemannian retraction which is defined in Section 3.5. Therefore, we can use Equation (72) as our update method for the R_i's. Note that for the last elements containing τ_i's the Riemannian retraction method reduces to the standard gradient descent algorithm in $\mathbb{R}$, since the tangent space of $\mathbb{R}$ is equal to $\mathbb{R}$.

Our analysis results in Algorithm 3, where we denote the iteration index as a superscript, i.e., R_i^t is the matrix R_i duration the t'th iteration step. The first three steps of Algorithm 3 are similar to Algorithm 2. These steps construct a basis for $T_I \mathrm{SO}(n) = \mathcal{S}_{\text{skew}}(n)$ and calculate the projection matrix, which is used to project the Euclidean gradient of f with respect to the R_i's onto the tangent space. In the fourth step, we calculate the initial value of the loss function f^0 by substituting in the R_i^0's and the τ_i^0's into Equation (93). The loop over $t = \{0, 1, 2, ...\}$ is over the iteration steps. One can specify when to stop this loop giving a maximum number of iterations t_{max} or by using a while-loop and a conditional statement. A possible conditional statement can be $f^t > \epsilon$ for some $\epsilon \in \mathbb{R}_{\geq 0}$. In this way, one can make sure that the value of the loss function f is below some maximal allowable value ϵ. One can also add a condition based on the values of the gradients since each gradient with respect to one of the R_i's should converge to the $n \times n$ zero matrix and each gradient with respect to one of the τ_i's should converge to 0. We will use a maximum number of iteration steps t_{max} to terminate this loop.

The first loop over $i = \{1, 2, ..., k\}$ is a loop over the number of links and joints k of the robotic arm. This loop updates the R_i^t's by using the Riemannian retraction. Then, we get an intermediate update $f^{t+\frac{1}{2}}$ for the loss function f by substituting the R_i^{t+1}'s and τ_i^t's into Equation (93). Then the second loop over $i = \{1, 2, ..., k\}$ updates the τ_i^t's by already using the previously obtained R_i^{t+1}'s and $f^{t+\frac{1}{2}}$. One can also opt to update the R_i^t's and the τ_i^t's in one loop over $i = \{1, ..., k\}$ and not calculate an intermediate value $f^{t+\frac{1}{2}}$ of the loss function. This will give a slower convergence rate, but this makes the algorithm simpler. We have implemented Algorithm 3 in Python, where we used the Numpy and Scipy libraries. The matrix exponential is calculated using the scipy.linalg.expm() function.

Algorithm 3 Riemannian Gradient Descent for the Robotic Arm

```
    Require: Piecewise linear function $\alpha(t) : [0,k] \to \mathbb{R}^n$ which describes the robotic arm (see Equation (88)) including the $x_1, ..., x_k$ which describe the initial orientation and length of each of the links, here $k$ is the number of links and joints; the minimization function $f$ (see Equation (93)); a learning rate or step size $\eta > 0$ for the $R_i$'s; a learning rate or step size $\sigma$ for the $\tau_i$'s; an operator vec : $\mathrm{M}_n(\mathbb{R}) \to \mathbb{R}^{n^2}$, which orders the elements of a matrix into a vector (see Section 3.3).
    Input: Initial iterate $R_1^0, ..., R_k^0 \in \mathrm{SO}(n)$ and $\tau_1^0, ..., \tau_k^0$.
    Output: Sequences of iterates $\{R_1^t\}, ..., \{R_k^t\}_{t=0}, \{\tau_1^t\}, ..., \{\tau_k^t\}$
 1: Construct a basis $\{U_1, U_2, ..., U_k\}$ for $\mathcal{S}_{skew}(n)$, where $k = \frac{1}{2}n(n-1)$.
 2: Construct the matrix $U_I = (\mathrm{vec}(U_1), \mathrm{vec}(U_2), ..., \mathrm{vec}(U_k)) \in \mathbb{R}^{n^2 \times k}$ (see Equation (59)).
 3: $V \leftarrow U_I U_I^T$.
 4: Calculate the initial loss $f^0$ by substituting the $R_i^0$'s and the $\tau_i^0$'s in $f$ (see Equation (93)).
 5: for $t = 0, 1, 2, ...$ do
 6:     Calculate $\mathrm{Grad}_{R_i^t} f^t$ for $i \in \{1, ..., k\}$.
 7:     for $i = 1, 2, .., k$ do
 8:         $R_i^{t+1} \leftarrow R_i^t \exp(-\eta \mathrm{vec}^{-1}(V \mathrm{vec}((R_i^t)^T \mathrm{Grad}_{R_i^t} f^t))$
 9:     end for
10:     Calculate $f^{t+\frac{1}{2}}$ by substituting the $R_i^{t+1}$'s and the $\tau_i^t$'s in $f$ (see Equation (93)).
11:     Calculate $\mathrm{Grad}_{\tau_i^t} f^{t+\frac{1}{2}}$ for $i \in \{1, ..., k\}$.
12:     for $i = 1, 2, ..., k$ do
13:         $\tau_i^{t+1} \leftarrow \tau_i^t - \sigma \, \mathrm{Grad}_{\tau_i^t} f^{t+\frac{1}{2}}$
14:     end for
15:     Calculate $f^{t+1}$ by substituting the $R_i^{t+1}$'s and the $\tau_i^{t+1}$'s in $f$ (see Equation (93)).
16: end for
```

As an initial test of Algorithm 3, let us consider the robotic-arm problem in $\mathbb{R}^2$. We take a robotic arm with three links and joints, i.e., $k = 3$. We want to fit the robotic arm on the curve $\gamma : \mathbb{R} \to \mathbb{R}^2$ given by

$$\gamma(\tau) = \begin{pmatrix} \tau \\ -(\tau-1)^2 + 1 \end{pmatrix}. \tag{103}$$

We set the number of integration points per link to $p = 1$. Therefore, only the endpoints of each link will be fitted on γ. We set $x_1 = x_2 = x_3 = x = (0.8, 0)$, therefore $\|x_i\| = 0.8$. We use $R_1^0 = R_2^0 = R_3^0 = I_{2\times 2}$ as our initial guesses for the R_i's. We use $\tau_1 = 0.4$, $\tau_2 = 0.8$ and $\tau_3 = 1.2$ as our initial guesses for the τ_i's.

In Section 4.4, we learned that a learning rate of $\eta = 1$ for the vector-rotation problem gives the best overall convergence. To solve the vector-rotation problem we used Algorithm 2. Note Algorithm 2 also uses the Riemannian retraction to update R. Therefore it is reasonable to assume that a learning rate of $\eta = 1$ is also a good choice for the robotic arm case. However, since $\|\mathrm{Grad}_{R_i} f\|_F$, for all $i \in \{1, 2, ..., k\}$, scales with $\|x\|^2$, we take $\eta = \frac{1}{\|x\|^2}$ to compensate for this scaling factor. Therefore, we choose the learning rate $\eta = \frac{1}{0.8^2} \approx 1.56$. Testing some learning rates σ gives that $\sigma = 1$ gives a faster convergence rate than smaller values of σ. Using a much higher value of σ will result in oscillatory behaviour in the value of the loss function f between iterations.

In Figure 20, illustrations of the robotic arm during the first ten iteration steps are given. Here, α^t denotes the robotic arm at iteration step t, which is defined by substituting $R_1 = R_1^t$, ..., $R_k = R_k^t$ in the expression for α (see Equation (88)). We see that the first link of the arm rapidly fits on γ. This is also visible in Figure 21. Here, f_i represents the loss induced by the ith link, i.e.,

$$f_i = \frac{1}{p} \sum_{j=1}^{p} \|\alpha_{ij} - g_{ij}\|^2, \tag{104}$$

such that

$$f = \frac{1}{k}\sum_{i=1}^{k} f_i. \tag{105}$$

f_i gives an indication of how well the ith link is fitted on top of γ. During the first four iterations, f_1 is smaller than f_2 and f_3, which indicates that the first link is better fitted than the other two links. In the next iteration steps, we see that f_1 and f_2 increase whereas f_3 and f still decrease. This is also visible in Figure 20 where the first two links are moved away from γ so that the third link fits better. The points $\gamma(\tau_1^{20})$, $\gamma(\tau_2^{20})$ and $\gamma(\tau_3^{20})$ are represented by the blue dots on the γ curve.

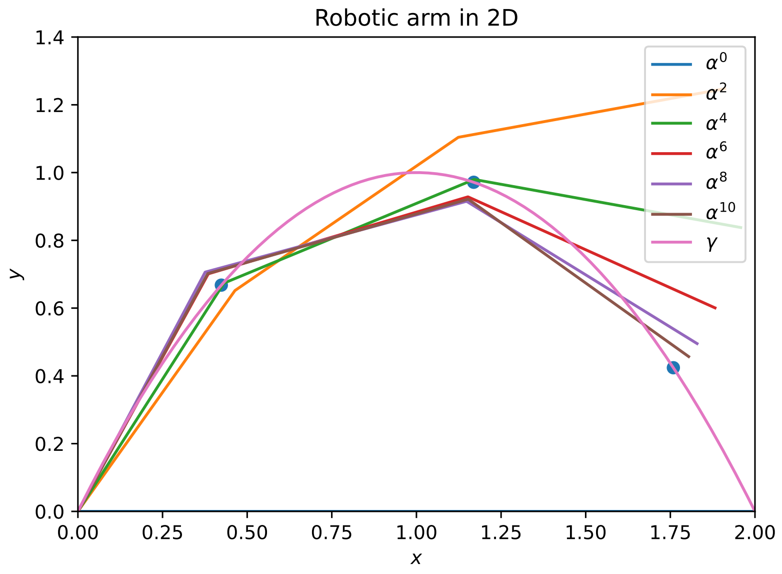


Figure 20: The robotic arm α^t. Here, $k = 3$, $p = 1$, $\eta = 1.56$, $\sigma = 1$. $\gamma(\tau_1^{10})$, $\gamma(\tau_2^{10})$ and $\gamma(\tau_3^{10})$ are represented by the blue dots.

The convergence behaviour over 110 iteration steps is given in Figure 23. We see that f converges to computer precision after these 110 iterations. The final orientation of the robotic arm after 110 iterations is given in Figure 22. The blue dots represent $\gamma(\tau_1^{110})$, $\gamma(\tau_2^{110})$ and $\gamma(\tau_3^{110})$. We see that the endpoints of each link are plotted on top of the blue dots and on top of the gamma curve, which is indeed optimal for $p = 1$.

Next, let us increase the number of integration points per link to $p = 10$. A graphical representation of the robotic arm for the first 20 iteration steps is given in Figure 24. We see that the first and second links are fitted on top of γ first. Performing more iterations also brings the third link on top of the γ curve. The same phenomenon is also visible in Figure 25, where we see that f_1 and f_2 rapidly decrease during the first 2 iteration steps. After these first 2 iterations the value of f_2 increases, which seems suboptimal. However, in this way the overall loss value f decreases, since f_1 and f_3 decrease. The same phenomenon occurs after iteration step 9, where now f_1 increases. So the algorithm allows for increasing f_i's if this results in a nett decrease of the loss function f.

In Figure 27 the value of the loss function f and the losses per link f_1, f_2, f_3 are plotted. We see that f_3 is larger than f_1 and f_2 until iteration step 41, then f_3 becomes smaller than f_1 and f_2, which indicates that the third link is fitted best. After 100 iterations the value of f stays around $3.9 \cdot 10^{-3}$ and the values of f_1, f_2 and f_3 do not change significantly anymore, indicating that we have reached a minimum of f. Note that the value of f does not go to zero, since for $p > 1$ it is not possible to fit all the integration points (the α_{ij}'s) on top of the γ curve. To further validate that we are approaching a minimum of f, we can also

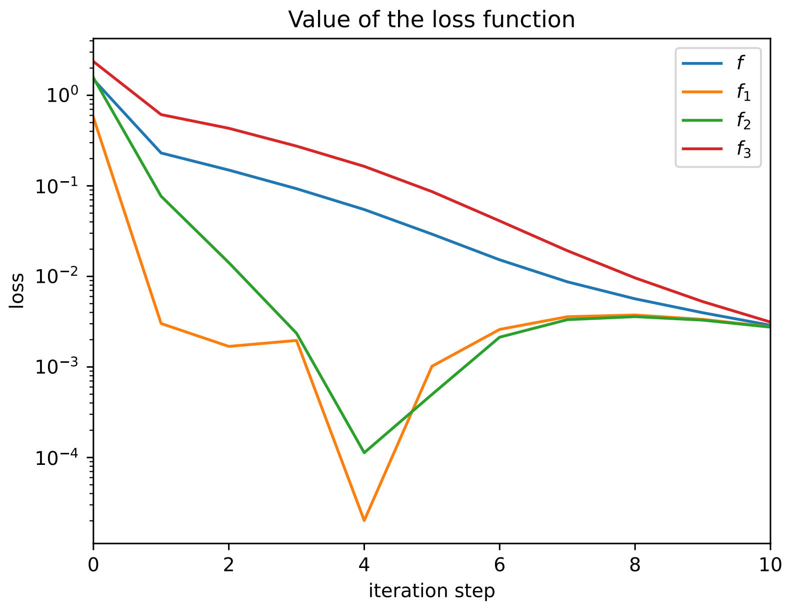


Figure 21: The loss function. f_1, f_2 and f_3 are also plotted (see Equation (104)).

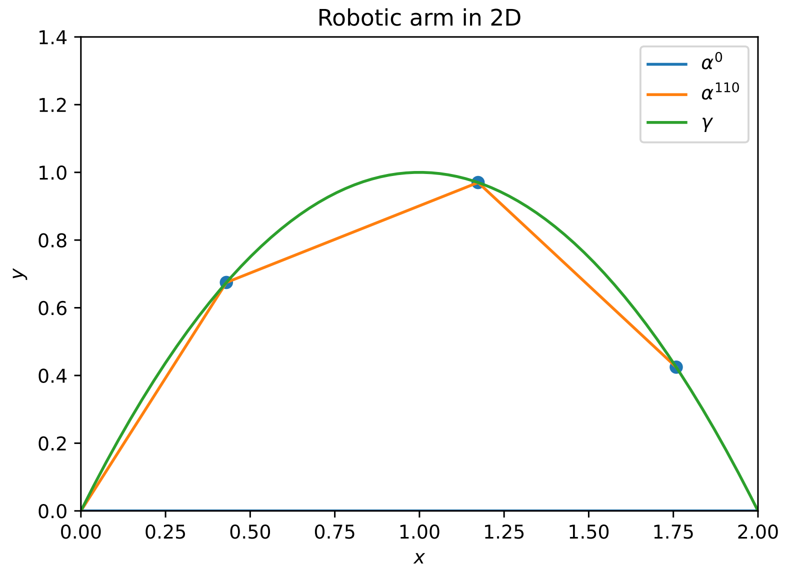


Figure 22: The robotic arm α^{110}. Here, $k = 3$, $p = 1$, $\eta = 1.56$, $\sigma = 1$. $\gamma(\tau_1^{110})$, $\gamma(\tau_2^{110})$ and $\gamma(\tau_3^{110})$ are represented by the blue dots.

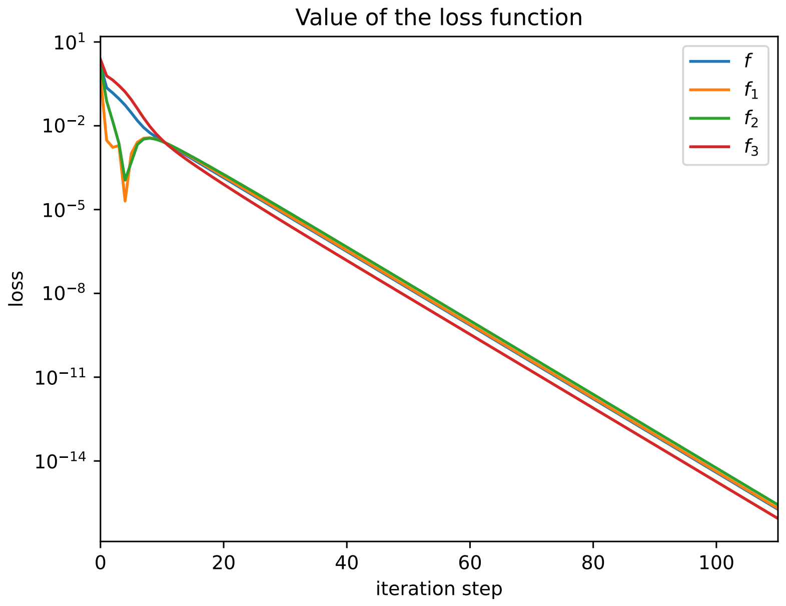


Figure 23: The loss function. Here, $k = 3$, $p = 1$, $\eta = 1.56$, $\sigma = 1$.

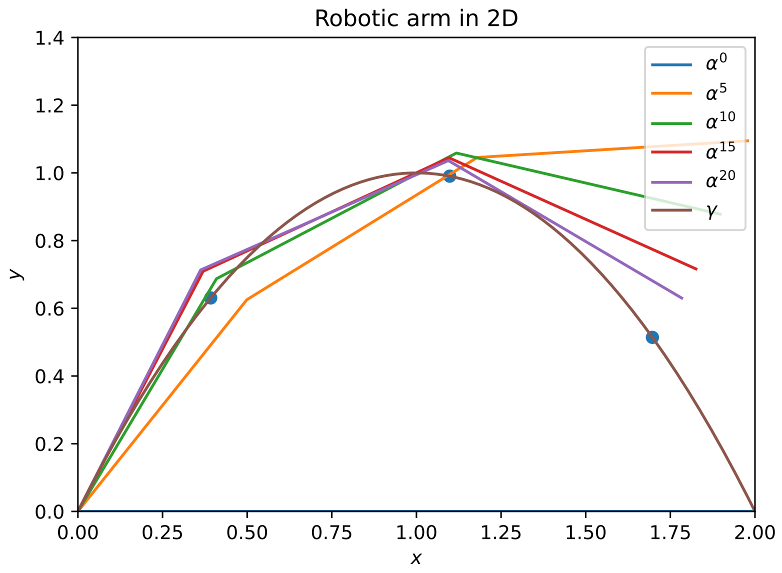


Figure 24: The robotic arm α^t. Here, $k = 3$, $p = 10$, $\eta = 1.56$, $\sigma = 1$. $\gamma(\tau_1^{20})$, $\gamma(\tau_2^{20})$ and $\gamma(\tau_3^{20})$ are represented by the blue dots.

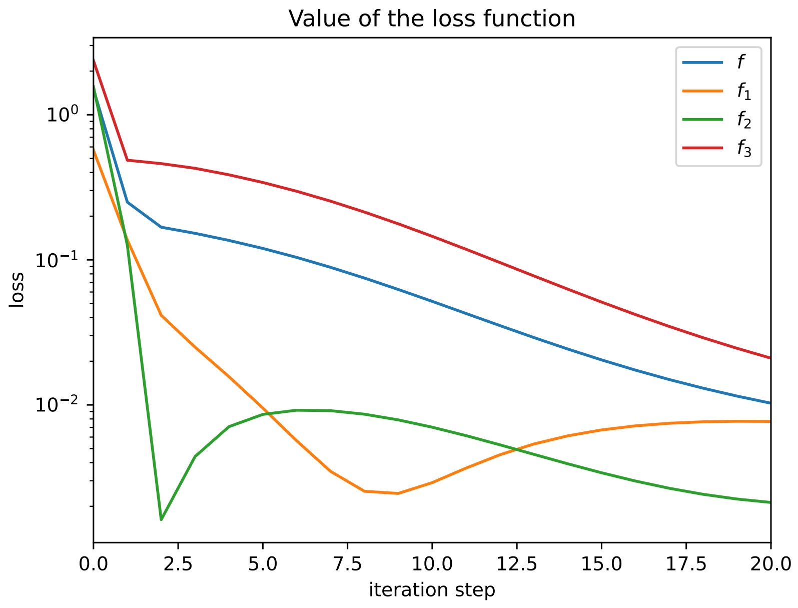


Figure 25: Loss during the first 20 iteration steps. Here, $\eta = 1.56$, $k = 3$ and $p = 10$.

look at the Riemannian gradients $\operatorname{grad}_{R_1} f$, $\operatorname{grad}_{R_2} f$ and $\operatorname{grad}_{R_3} f$. For convergence, these gradients should approach $0_{n\times n}$, therefore the Frobenius norms of these matrices should become zero. To investigate this, we have plotted $\|\operatorname{grad}_{R_1} f\|_F$, $\|\operatorname{grad}_{R_2} f\|_F$ and $\|\operatorname{grad}_{R_3} f\|_F$ in Figure 28. Furthermore, let us define

$$\operatorname{grad}_\tau f = \left(\operatorname{grad}_{\tau_1} f, ..., \operatorname{grad}_{tau_k} f\right)^T,$$

where $\operatorname{grad}_\tau f \in \mathbb{R}^n$. The Frobenius norm of $\operatorname{grad}_\tau f$ is given by

$$\|\operatorname{grad}_\tau f\|_F = \sqrt{\sum_{i=1}^{k} \left(\operatorname{grad}_{\tau_i} f\right)^2},$$

which is also given in Figure 28. We indeed see that these norms go to zero, which indicates that we are approaching a minimum for f. The final configuration of the robotic arm α^{200} after 200 iterations is given in Figure 26, which nicely fits the γ curve.

Next, we double the number of links k to $k = 6$. Furthermore, let $x_i = (0.4, 0)$ and $R_i^0 = I_{2\times 2}$, for all $i \in \{1, ..., 6\}$. We use $\tau_1^0 = 0.2$, $\tau_2^0 = 0.4$, ..., $\tau_6^0 = 1.2$ as our initial guesses for the τ_i's. We keep the learning rate for the τ_i's at $\sigma = 1$ and we choose a learning rate for the R_i's of $\eta = \frac{1}{\|x\|^2} \approx 6.25$. The number of integration points per link is $p = 10$. The end result after 200 iterations is given in Figure 29. We see that the robotic arm matches the γ curve. In Figure 30 the value of the loss function is plotted over 200 iterations. Again we see that the value of the loss function stabilizes around 100 iterations where $f \approx 2.5 \cdot 10^{-4}$. Note that for the $k = 3$ case, we had $f = 3.9 \cdot 10^{-3}$ at iteration step 100. When $k = 6$, we have more links. Therefore, we can fit the robotic arm better on γ than for the $k = 3$ case. This explains why f is smaller for the $k = 6$ case than for the $k = 3$ case. Furthermore, we see that f_6 convergences very slowly. This is the loss associated with the sixth and final link. We can explain this slow convergence by the fact that there are a lot of ways to change the position of the sixth link since this position depends on all the R_i's, whereas the position of the first link only depends on R_1. Therefore we need more iteration steps to fit the sixth link than we need to fit only the first link on γ.

In short, we developed Algorithm 3 which can fit a robotic arm, which is described by Equation (88), on a given curve $\gamma : \mathbb{R} \to \mathbb{R}^n$ in general dimensions n. We have tested the algorithm in 2 dimensions where each

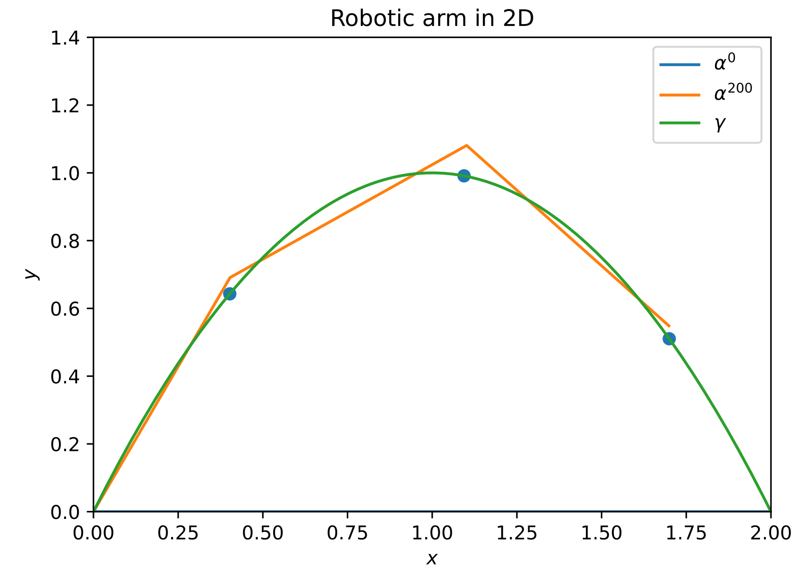


Figure 26: The robotic arm α^t. Here, $k = 3$, $p = 10$, $\eta = 1.56$, $\sigma = 1$. $\gamma(\tau_1^{200})$, $\gamma(\tau_2^{200})$ and $\gamma(\tau_3^{200})$ are represented by the blue dots.

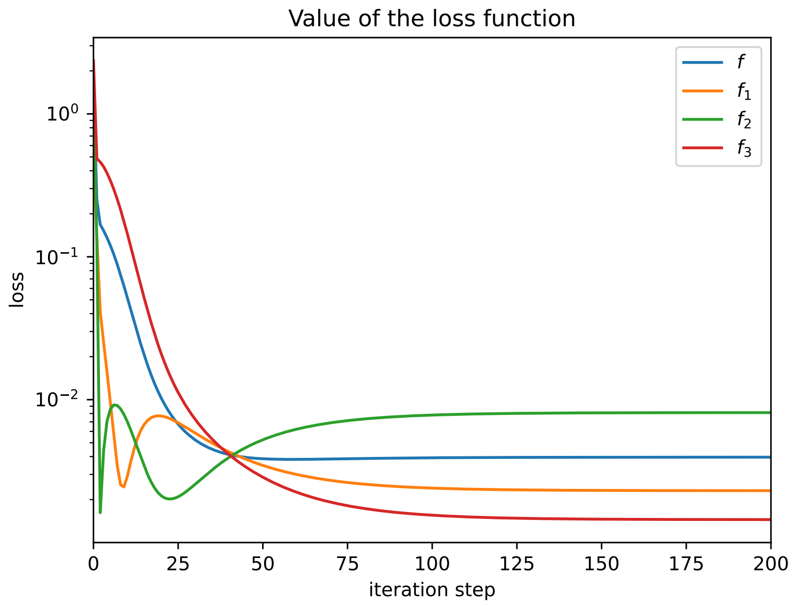


Figure 27: The loss function. Here, $k = 3$, $p = 10$, $\eta = 1.56$, $\sigma = 1$.

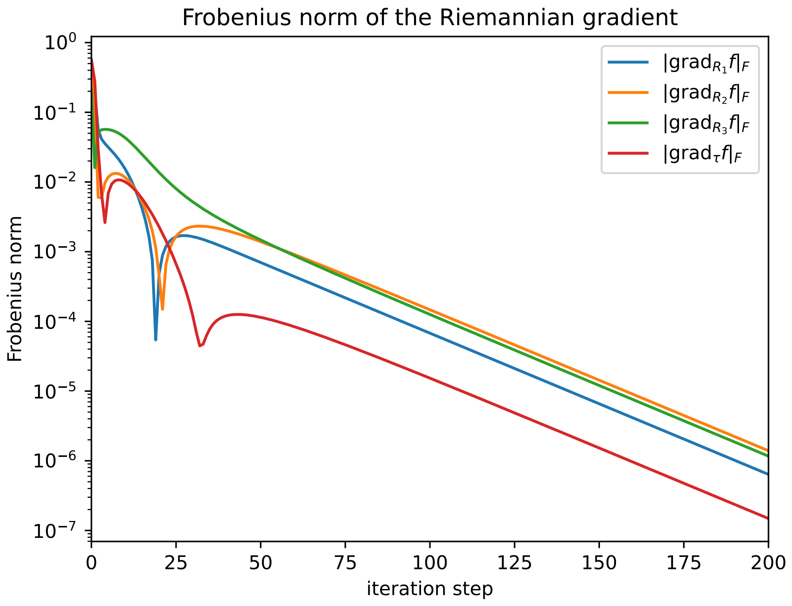


Figure 28: Frobenius norm of the Riemannian gradient. Here, $k = 3$, $p = 10$, $\eta = 1.56$, $\sigma = 1$.

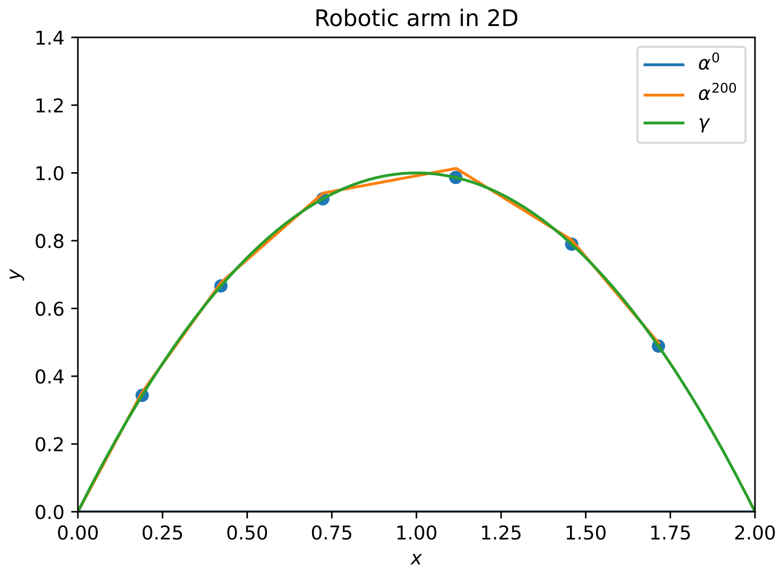


Figure 29: The robotic arm α^{200}. Here, $k = 6$, $p = 10$, $\eta = 6.25$, $\sigma = 1$.

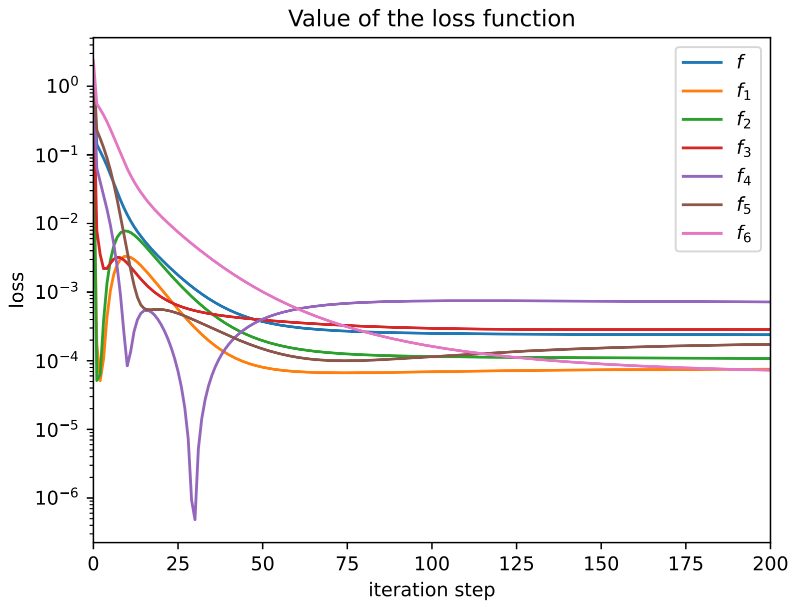


Figure 30: The loss function f. Here, $k = 6, p = 10, \eta = 6.25, \sigma = 1$.

link has an equal length. When choosing the learning rate $\eta = \frac{1}{\|x\|^2}$ and $\sigma = 1$ gives a proper solution after 100 iterations. We advise doing more iteration steps when k is large so that the final links are also fitted on top of γ.

In the next section, Section 4.6, we will discuss a flaw of Algorithm 3, which occurs when the initial value of the τ_i's are chosen poorly. This flaw is fixed by introducing a method which finds proper initial values for the τ_i's. In Section 4.7, we will test the algorithm in higher dimensions.

4.6 Arc Length Initialization

In this section, we discuss a flaw of Algorithm 3. The flaw is that the final solution of Equation (93) obtained by the algorithm may depend on the initial values R_i^0 and τ_i^0, where $i \in \{1, 2, ..., k\}$.

To illustrate this problem let us consider the case where the number of links is $k = 3$. Let the number of integration points per link be $p = 10$ and let γ be described by Equation (103). We take $\|x_i\| = 0.8$ for $i \in \{1, 2, 3\}$. Note that we already considered this problem in the previous section. However, let $R_1^0 = R_3^0 = I_{2\times2}$ and $R_2^0 = -I_{n\times n}$ be the initial values for the R_i's and let $\tau_1^0 = \tau_3^0 = 0.4$ and $\tau_2^0 = 0$ be the initial values for the τ_i's. We choose the learning rates $\eta = \frac{1}{\|x_i\|^2} \approx 1.56$ and $\sigma = 0.5$. The results are given in Figures 31, 32 and 33.

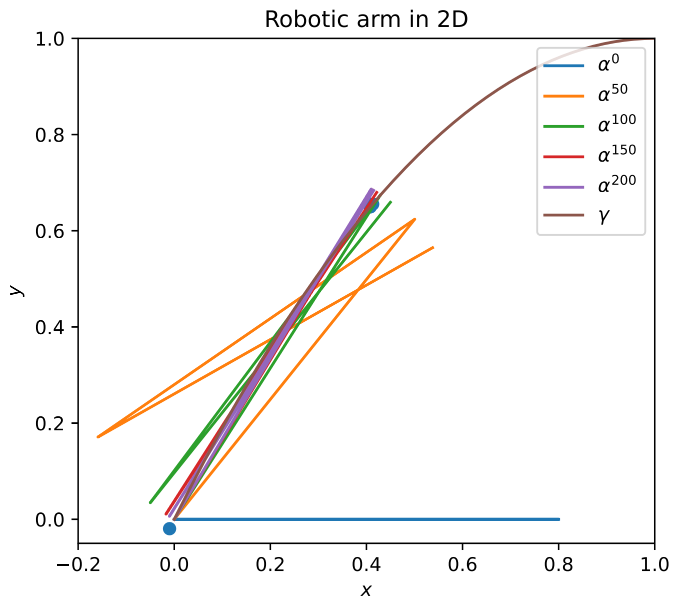


Figure 31: The robotic arm α^t. Here, $k = 3, p = 10, \eta = 1.56, \sigma = 1$. The initial values are $R_1^0 = R_3^0 = I_{2\times2}$, $R_2^0 = -I_{2\times2}$ and $\tau_1^0 = \tau_3^0 = 0.4$, $\tau_2^0 = 0$.

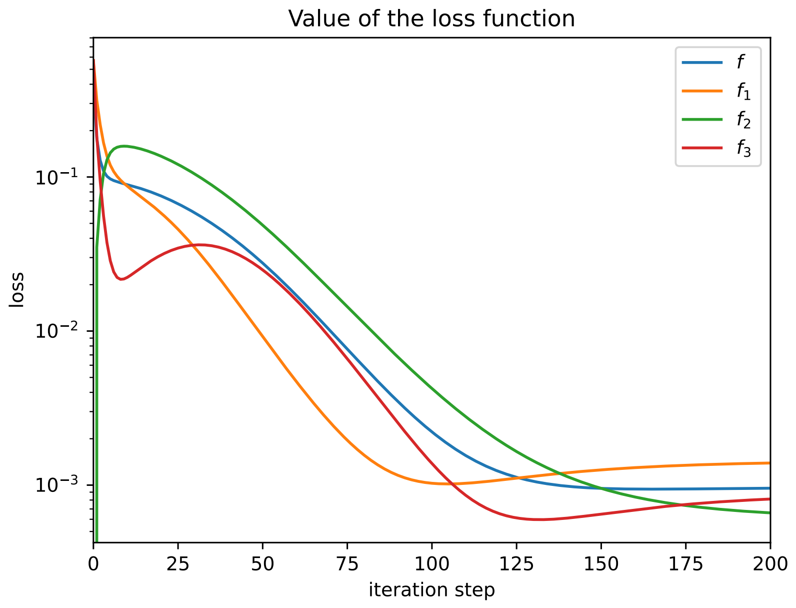


Figure 32: The loss function f. Here, $k = 3, p = 10, \eta = 1.56, \sigma = 1$. The initial values are $R_1^0 = R_3^0 = I_{2\times 2}$, $R_2^0 = -I_{2\times 2}$ and $\tau_1^0 = \tau_3^0 = 0.4$, $\tau_2^0 = 0$.

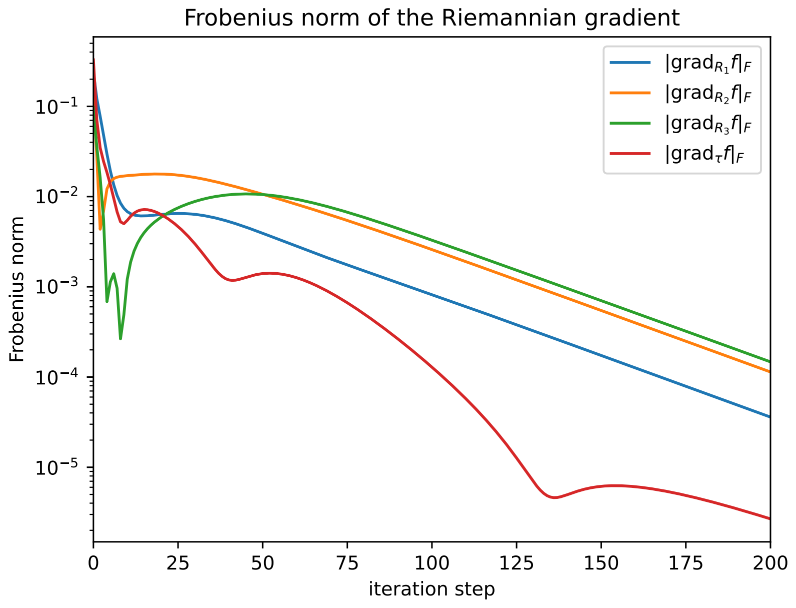


Figure 33: The Frobenius norm of the gradients. Here, $k = 3, p = 10, \eta = 1.56, \sigma = 1$. The initial values are $R_1^0 = R_3^0 = I_{2\times 2}$, $R_2^0 = -I_{2\times 2}$ and $\tau_1^0 = \tau_3^0 = 0.4$, $\tau_2^0 = 0$.

In Figure 31, we see that the end configuration of the robotic arm α^{200} differs from the configuration given in Figure 26. This is because of the difference in the initial values between the two instances. In Figure 32 we see that after 200 iterations the value of f stays around $9.5 \cdot 10^{-4}$ and in Figure 33 we see that Frobenius norms of the Riemannian gradients go to zero, which both indicate that we have reached a minimum of the loss function f. However, we prefer the solution given in Figure 26 over the solution given in Figure 31. We want that the robotic arm travels along γ as far as possible and that its links do not fold into eachother. So, we require

$$R_i \neq -R_{i+1}, \quad \text{for all } i \in \{1, ..., k\}.$$

One can impose such a constraint by adding an extra term to the loss function f. However, we developed another technique which solves this problem and does not further complicate the loss function f. Our idea is to give a good guess for the initial values τ_i^0, for $i \in \{1, ..., k\}$. To determine this guess we first need to consider the definition of the arc length. [4]

Definition 4.1 (Arc Length)**.** Given is a curve $\gamma : \mathbb{R} \to \mathbb{R}^n$. Let $\gamma(a)$ and $\gamma(b)$ be two points on γ The arc length $s(a, b)$ between these points is defined as the length along the curve $\gamma(\tau)$ from $\gamma(a)$ to $\gamma(b)$, which is given by

$$s(a, b) = \int_a^b \|\gamma'(\tau)\| \, \mathrm{d}\tau. \tag{106}$$

We choose our initial values τ_i^0 such that

$$s(\tau_{i-1}^0, \tau_i^0) = \int_{\tau_{i-1}^0}^{\tau_i^0} \|\gamma'(\tau)\| \mathrm{d}\tau = \|x_i\|, \quad \text{for all } i \in \{1, ..., k\}, \tag{107}$$

where $\tau_0^0 = 0$. Moreover, we fix $\tau_i = \tau_i^0$, $i \in \{1, ..., k\}$, for the first $\widehat{t}$ iteration steps and optimize f only for the R_i's. One can choose to do this until convergence. Then, we let go of the restriction on the τ_i's and optimize f for the R_i's and the τ_i's simutanuously. This idea results in Algorithm 4. Note that during the first for-loop of Algorithm 4 we fix $\tau_i^t = \tau_i^0$ from iteration step 1 upto $\widehat{t}$. One can opt to change this for-loop into a while loop, which terminates if f^t has converged. One can check for convergence by looking at the difference $f^t - f^{t-1}$ or by looking at the Riemannian gradients and checking if they approach the zero matrices.

Let us test Algorithm 4 by considering the path $\gamma : [0, 1] \to \mathbb{R}^n$, which is defined as

$$\gamma(\tau) = \begin{pmatrix} \sin(2\pi\tau) \\ \cos(4\pi\tau) \end{pmatrix}.$$

We fit a robotic arm with $k = 18$ links and $p = 10$ integration points per link on this γ curve. Let us first use Algorithm 3 from the previous section, which does not initialize τ_i^0 based on the arc lengths equations, to solve this case. Let us take $\tau_i^0 = 0$ instead, for all $i \in \{1, ..., 18\}$. Furthermore, let $x_i = (0.5, 0)$, $R_i^0 = I_{2\times 2}$. We choose $\eta = 2$ and $\sigma = 0.1$ as our learning rates. We perform 400 iteration steps. The results are given in Figures 34 and 35.

Algorithm 4 Riemannian gradient descent with arc length initialization

htbp

Require: Piecewise linear function $\alpha(t) : [0, k] \to \mathbb{R}^n$ which describes the robotic arm (see Equation (88)) including the $x_1, ..., x_k$ which describe the initial orientation and length of each of the links, here k is the number of links and joints; the minimization function f (see Equation (93)); a learning rate or step size $\eta > 0$ for the R_i's; a learning rate or step size σ for the τ_i's; an operator vec : $\mathrm{M}_n(\mathbb{R}) \to \mathbb{R}^{n^2}$, which orders the elements of a matrix into a vector (see Section 3.3); Number of iteration $\widehat{t}$ for which we want to fix $\tau_i = \tau_i^0$, $i \in \{1, ..., k\}$.

Input: Initial iterate $R_1^0, ..., R_k^0 \in \mathrm{SO}(n)$.

Output: Sequences of iterates $\{R_1^t\}, ..., \{R_k^t\}_{t=0}, \{\tau_1^t\}, ..., \{\tau_k^t\}$

```
1: Initialize the $\tau_i$'s by solving Equation (106) for $\tau_i^0$, $i \in \{1, ..., k\}$.
2: Construct a basis $\{U_1, U_2, ..., U_k\}$ for $\mathcal{S}_{skew}(n)$, where $k = \frac{1}{2}n(n-1)$.
3: Construct the matrix $U_I = (\mathrm{vec}(U_1), \mathrm{vec}(U_2), ..., \mathrm{vec}(U_k)) \in \mathbb{R}^{n^2 \times k}$ (see Equation (59)).
4: $V \leftarrow U_I U_I^T$.
5: Calculate the initial loss $f^0$ by substituting the $R_i^0$'s and the $\tau_i^0$'s in $f$ (see Equation (93)).
6: for $t = 0, 1, 2, ..., \widehat{t}$ do
7:     Calculate $\mathrm{Grad}_{R_i^t} f^t$ for $i \in \{1, ..., k\}$.
8:     for $i = 1, 2, .., k$ do
9:         $R_i^{t+1} \leftarrow R_i^t \exp(-\eta \mathrm{vec}^{-1}(V \mathrm{vec}((R_i^t)^T \,\mathrm{Grad}_{R_i^t} f^t))$
10:        $\tau_i^t \leftarrow \tau_i^0$
11:    end for
12: end for
13: for $t = \widehat{t}, (\widehat{t} + 1), ...$ do
14:    Calculate $\mathrm{Grad}_{R_i^t} f^t$ for $i \in \{1, ..., k\}$.
15:    for $i = 1, 2, .., k$ do
16:        $R_i^{t+1} \leftarrow R_i^t \exp(-\eta \mathrm{vec}^{-1}(V \mathrm{vec}((R_i^t)^T \,\mathrm{Grad}_{R_i^t} f^t))$
17:    end for
18:    Calculate $f^{t+\frac{1}{2}}$ by substituting the $R_i^{t+1}$'s and the $\tau_i^t$'s in $f$ (see Equation (93)).
19:    Calculate $\mathrm{Grad}_{\tau_i^t} f^{t+\frac{1}{2}}$ for $i \in \{1, ..., k\}$.
20:    for $i = 1, 2, ..., k$ do
21:        $\tau_i^{t+1} \leftarrow \tau_i^t - \sigma \,\mathrm{Grad}_{\tau_i^t} f^{t+\frac{1}{2}}$
22:    end for
23:    Calculate $f^{t+1}$ by substituting the $R_i^{t+1}$'s and the $\tau_i^{t+1}$'s in $f$ (see Equation (93)).
24: end for
```

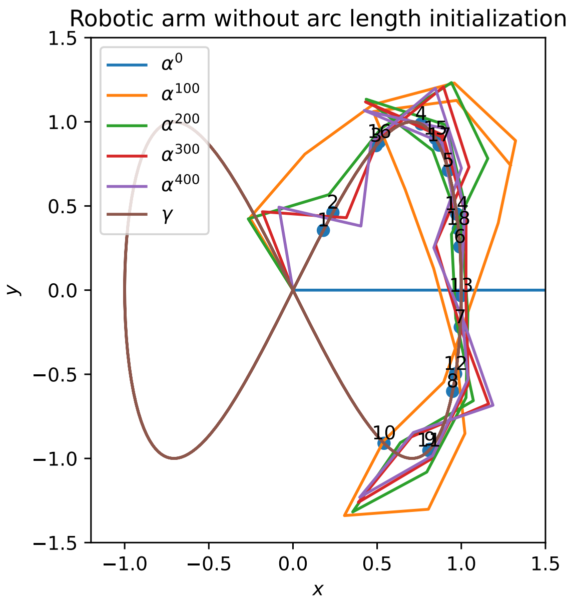


Figure 34: The robotic arm α^t with $\tau_i^0 = 0$ for $i \in \{1, ..., k\}$. Here, $k = 18, p = 10, \eta = 2, \sigma = 0.1$.

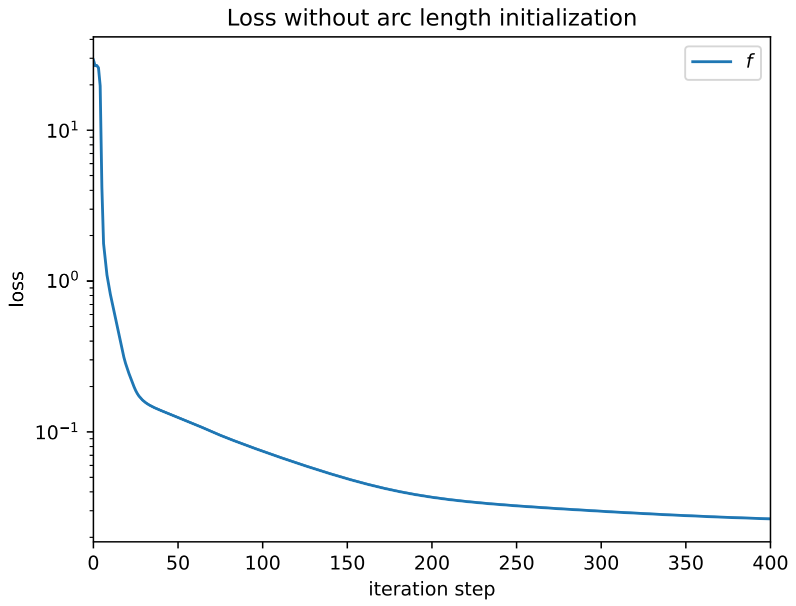


Figure 35: The loss function f with $\tau_i^0 = 0$ for $i \in \{1, ..., k\}$. Here, $k = 18, p = 10, \eta = 2, \sigma = 0.1$.

In Figure 34, the blue dots label $\gamma(\tau_i^{400})$ for $i \in \{1, ..., 18\}$. We see that $\gamma(\tau_9^{400})$ and γ_{11}^{400} lay on top of each other and so are $\gamma(\tau_{15}^{400})$ and $\gamma(\tau_{17}^{400})$. This indicates that $R_9 \approx -R_{10}$ and $R_{15} \approx -R_{16}$. This phenomenon occurs at a place where the γ curve makes a sharp turn. Looking at Figure 35, we see that we approach a local minimum for f. The algorithm is not able to escape this minimum. After 400 iterations, we obtain

$f^{400} = 2.65 \cdot 10^{-2}$.

Next, let us consider the same problem with the same initial R_i^0 and the same learning rates $\eta = 2$ and $\sigma = 0.1$. However, now we use Algorithm 4. So we first calculate τ_i^0 based on Equations (106). Then we fix $\tau_i^t = \tau_i^0$ during the first $\widehat{t} = 100$ iterations. Then we perform 300 more iterations, where we unfix τ_i^t. Algorithm 4 is implemented in Python. We used the scipy.optimize.root$_\text{s}$calar() and scipy.integrate.quad() functions to solve Equation (106) numerically to obtain τ_i^0 for $i \in \{1, ..., 18\}$. If the primitive function of $\|\gamma'(\tau)\|$ is known, one can also opt to solve Equation (106) exactly. The results of our code are given in Figures 36 and 37.

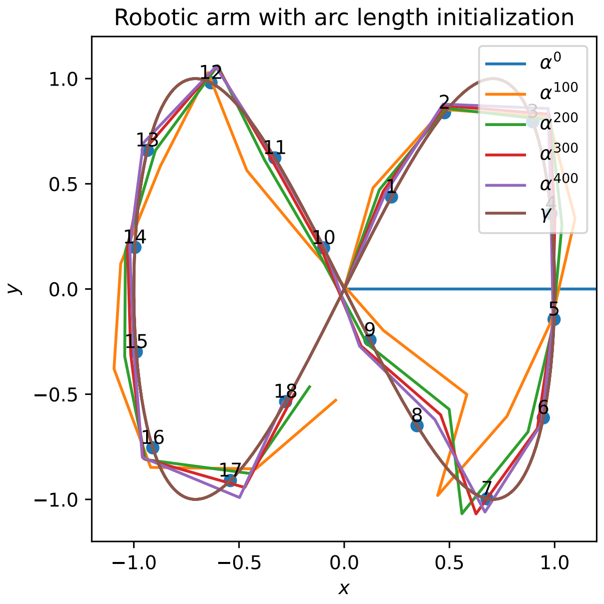


Figure 36: The robotic arm α^t, where the τ_i^t's are fixed during the first $\widehat{t} = 100$ iterations. Here, $k = 18, p = 10, \eta = 2, \sigma = 0.1$.

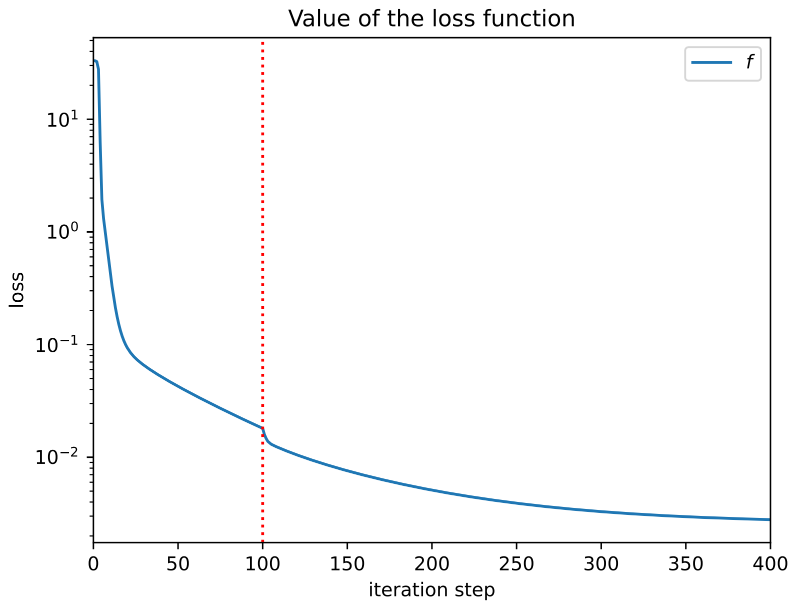


Figure 37: The loss function f, where the τ_i^t's are fixed during the first $\widehat{t} = 100$ iterations. Here, $k = 18, p = 10, \eta = 2, \sigma = 0.1$.

In Figure 36, we see that the robotic arm is stretched out over γ. There are no joints for which $R_{i+1} \approx -R_i$. In Figure 37, we see a plot of the value of the loss function f over the iteration steps. For the first $\widehat{t} = 100$ iterations, the shape of the graph of f is similar to the one given in Figure 34. During these iteration steps we fix $\tau_i^t = \tau_i^0$, so we are minimimzing over $\mathrm{SO}(2)^{18}$. In Figure 36, the decrease in the loss function is larger than in Figure 34, indicating that we reach a better minimum. After iteration step $\widehat{t} = 100$, we see a sudden decrease in f, because we unfix the τ_i's at iteration $\widehat{t} = 100$. This gives more freedom to minimize f, since we are now mimimizing over $\mathrm{SO}(2)^{18} \times \mathbb{R}^{18}$ instead. After a total of 400 iterations, we obtain $f^{400} = 2.79 \cdot 10^{-3}$. This is a significant improvement with respect to the results from Algorithm 3, where we obtained $f^{400} = 2.65 \cdot 10^{-2}$.

In terms of computational complexity Algorithm 4 is slightly more expensive than Algorithm 3, since we first need to solve the arc length equations (given by Equation (106)) numerically before we start the iteration steps. However, this extra computational cost is neglectable in comparison to the cost of performing the rest of the Algorithm 4. Moreover, if a primitive function of $\|\gamma'(\tau)\|$ is known, one can solve the arc length equations exactly and very efficiently.

In short, we developed Algorithm 4 as a way to improve Algorithm 3. Algorithm 3 can get stuck in a local minimum for which for some links i, we have $R_{i+1} \approx -R_i$, which is undesired. This can especially occur when solving more complicated problems where the number of links k is large or the path γ contains multiple curves. Algorithm 4 initializes the τ_i's based on the arc lenght equations (106) and keeps these τ_i's fixed for the first couple of iterations $\widehat{t}$, to guarantee that we end up in the desired minimum of f. A test case, where $k = 18$, shows that Algorithm 4 indeed ends up in this desired minimum.

4.7 The Robotic Arm in Higher Dimensions

In this final section about the numerical simulations, we lay emphasis on the fact that Algorithm 4 is suitable for fitting a robotic arm on a curve γ in any multiple dimensional space $\mathbb{R}^n$. We will look at $\mathbb{R}^3$ since for this case we still have tools to visualize the robotic arm.

Let us difine the path $\gamma : \mathbb{R} \to \mathbb{R}^3$ as follows

$$\gamma(\tau) = \begin{pmatrix} 1 - \cos(2\pi\tau) \\ \sin(2\pi\tau) \\ \tau \end{pmatrix}.$$

Let the number of links be $k = 18$ and let the number of integration points per link be $p = 10$. Furthermore, let $R_i^0 = I_{3\times 3}$ for $i \in \{1, 2, ..., 18\}$. We use Algorithm 4 with learning rates $\eta = 2$ and $\sigma = 0.1$, where we fix $\tau_i^t = \tau_i^0$ for the first $\widehat{t} = 50$ iterations. The results are given in Figures 38 and 39. Here, the labeled blue dots represent $\gamma(\tau_i^{200})$ for $i \in \{1, ..., 18\}$.

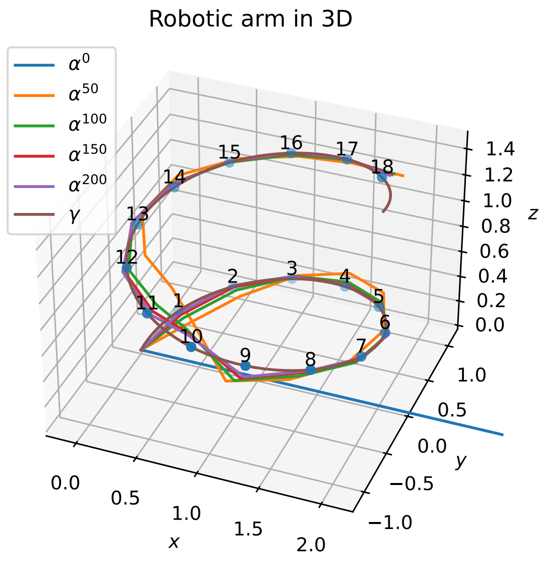


Figure 38: The robotic arm α^t in $\mathbb{R}^3$, where the τ_i^t's are fixed during the first $\widehat{t} = 50$ iterations. Here, $k = 18, p = 10, \eta = 2, \sigma = 0.1$.

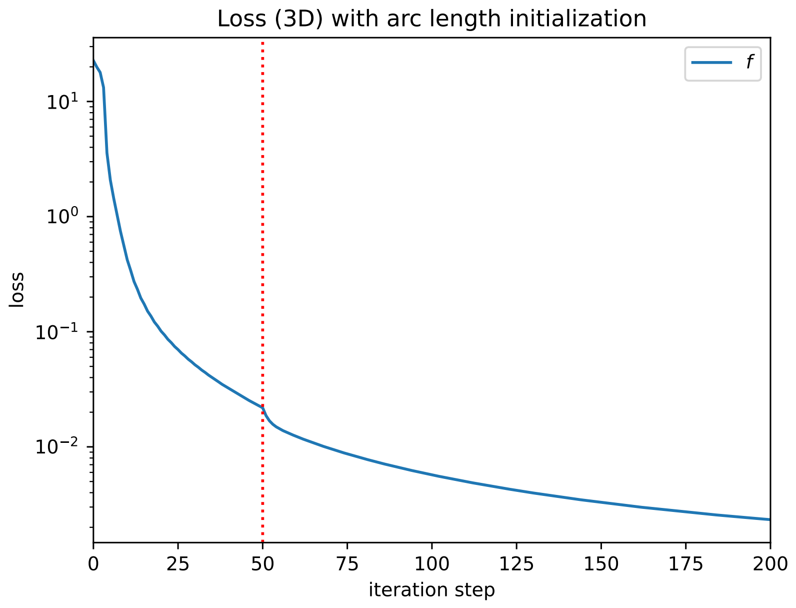


Figure 39: The loss function f, where the τ_i^t's are fixed during the first $\widehat{t} = 50$ iterations. Here, $n = 3$, $k = 18, p = 10, \eta = 2, \sigma = 0.1$.

In figure 38, we see that after 200 iterations the robotic arm fits γ. The loss function is plotted in Figure 39. We need slightly fewer iteration steps to converge to the solution in this case than for the 2-dimensional case from the previous section (see Figures 36 and 37). This could be due to multiple factors. For example, the γ curve differs from the γ curve considered in the previous section. It can also be due to the phenomenon which we discussed in Section 4.3. There we found that increasing the dimensionality of the vector-rotation problem can in some cases lead to a faster convergence because it becomes less likely for the algorithm to get stuck in an unstable critical point.

In general, the computational cost for each iteration increases as we increase the dimensionality n of the problem, because of the increasing dimensions of the R_i matrices. Therefore, the total computational cost increases as n increases. Based on our observation in Section 4.3, we also expect that for the robotic-arm problem the number of iterations needed until convergence is independent of n.

5 Conclusion

In this report, we looked at the classical gradient descent method and adapted it to a Riemannian setting. Our analyses of the tangent spaces of Riemannian manifolds, the Riemannian metric and the Riemannian gradient resulted in the constant line search method (CLS) given by Algorithm 1. Performing this algorithm includes calculating the Riemannian gradient and using a retraction method.

We have further specified the CLS method for the Riemannian manifold $\mathrm{SO}(n)$. $\mathrm{SO}(n)$ is a Riemannian submanifold of $\mathrm{M}_n(\mathbb{R})$ and therefore inherits the Riemannian metric from $\mathrm{M}_n(\mathbb{R})$, which is the Frobenius inner product. Using this Riemannian metric we were able to calculate the Riemannian gradient $\operatorname{grad} f(R)$ of a loss function f at a point $R \in \mathrm{SO}(n)$ by projecting the Euclidean gradient onto the tangent space $T_R\,\mathrm{SO}(n)$ (see Equation (63)). As a retraction method, we used the Riemannian exponential map, which is defined in terms of geodesics (see Definition 3.5). This gave us all the tools to specify the CLS method for $\mathrm{SO}(n)$. This resulted in Algorithm 2.

We tested Algorithm 2 on the vector-rotation problem. We concluded that a learning rate of $\eta = 1$ gives the fasted convergence rate and that the number of iteration steps needed until convergence does not change as we increase the dimensionality of the problem. This shows that Algorithm 2 is very suitable for solving higher dimensional problems.

Lastly, we considered the robotic-arm problem, where we try to find a robotic arm configuration which accurately approximates a given curve $\gamma : \mathbb{R} \to \mathbb{R}^n$. We developed Algorithm 3, which gave good results if the number of links k is small. We adapted Algorithm 3 by initializing the τ_i ($i \in \{1, 2, ..., k\}$) variables based on the arc length equations given by Equation (106). This resulted in Algorithm 4, which also gives good results for high values for k. We lay special emphasis on the fact that Algorithm 4 can be used to do real-time updating of the robotic arm. Moreover, Algorithm 4 is very suitable for solving such the robotic-arm problem in a higher dimensional space, since the number of iteration steps until convergence does not increase as n increases. However, the computational cost of each iteration step does increase as n increases, due to the increasing dimensions of the R_i matrices.

5.1 Future Directions

The last thing we want to discuss is some future research directions related to this thesis. First of all, it could be interesting to adopt the CLS method for other matrix groups, like the special linear group or the symplectic group. Secondly, we could compare our algorithm to other methods, for example, the stochastic gradient descent method (see Ref. [7]) and the Newton method (see chapter 6 of Ref. [2]). Finally, there is a lot of literature about methods which can optimize over the orthogonal group $\mathrm{O}(n)$ for training deep neural networks. These methods are also based on the theory of Lie groups and Riemannian geometry. It can be interesting to further explore the possibilities within this field of research.

References

[1] *5.14: The Rayleigh Distribution - Statistics LibreTexts.* URL: https://stats.libretexts.org/Bookshelves/Probability_Theory/Probability_Mathematical_Statistics_and_Stochastic_Processes_(Siegrist)/05:_Special_Distributions/5.14:_The_Rayleigh_Distribution.

[2] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds.* 2009. ISBN: 9780691132983. DOI: 10.1515/9781400830244.

[3] Foivos Alimisis et al. "A Continuous-time Perspective for Modeling Acceleration in Riemannian Optimization". In: 108 (2019). URL: http://arxiv.org/abs/1910.10782.

[4] *Arc Length – from Wolfram MathWorld.* URL: https://mathworld.wolfram.com/ArcLength.html.

[5] George Bachman and Lawrence Narici. *Functional Analysis.* Second. Mineola, New York: Dover Publications, 2000. ISBN: 978-0486402512.

[6] Andrew Baker. *Matrix groups : an introduction to Lie group theory.* London ; Springer, 2002, xi, 330 pages : ISBN: 1852334703.

[7] S Bonnabel - IEEE Transactions on Automatic Control and undefined 2013. "Stochastic gradient descent on Riemannian manifolds". In: *ieeexplore.ieee.org* (). URL: https://ieeexplore.ieee.org/abstract/document/6487381/?casa_token=u19G8CcQ8WgAAAAA:ZepvodzvrHl0YrsSBOOlgFgFygaPVW5r52-gycPXjYQ_IWsk7RXORxjFk7lqOMGS6oNQ-eNa.

[8] Aaron D'Souza, Sethu Vijayakumar, and Stefan Schaal. "Learning inverse kinematics". In: *IEEE International Conference on Intelligent Robots and Systems.* Vol. 1. 2001, pp. 298–303. DOI: 10.1109/iros.2001.973374.

[9] Jean Gallier and Jocelyn Quaintance. *Differential Geometry and Lie Groups.* Vol. 12. Geometry and Computing. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-46039-6. DOI: 10.1007/978-3-030-46040-2. URL: http://link.springer.com/10.1007/978-3-030-46040-2.

[10] Andrew A. Goldenberg, B. Benhabib, and Robert G. Fenton. "A Complete Generalized Solution to the Inverse Kinematics of Robots". In: *IEEE Journal on Robotics and Automation* 1.1 (1985), pp. 14–20. ISSN: 08824967. DOI: 10.1109/JRA.1985.1086995.

[11] *Grönwall's inequality - Wikipedia.* URL: https://en.wikipedia.org/wiki/Gr%C3%B6nwall%27s_inequality.

[12] Ernst Hairer. "Solving Differential Equations on Manifolds". In: June (2011), pp. 1–55. URL: http://www.unige.ch/~hairer/poly-sde-mani.pdf.

[13] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations.* Vol. 222. Graduate Texts in Mathematics. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-13466-6. DOI: 10.1007/978-3-319-13467-3. URL: https://link.springer.com/10.1007/978-3-319-13467-3.

[14] *Hypersphere Point Picking – from Wolfram MathWorld.* URL: https://mathworld.wolfram.com/HyperspherePointPicking.html.

[15] *Industrial robots — Stäubli.* URL: https://www.staubli.com/tw/en/robotics/products/industrial-robots.html.

[16] Jamshed Iqbal et al. *Modeling and analysis of a 6 DOF robotic arm manipulator Towards autonomous cleaning of photovoltaic modules: Design and realization of a robotic cleaner View project Industrial Robotics Simulation Design Planning and Optimization Platform View project.* Tech. rep. 6. 2012, p. 300. URL: https://www.researchgate.net/publication/280643085.

[17] Claude LeBrun and John M. Lee. "Introduction to Topological Manifolds". In: *The American Mathematical Monthly.* Graduate Texts in Mathematics 109.6 (2002), p. 577. ISSN: 00029890. DOI: 10.2307/2695461. URL: https://link.springer.com/10.1007/978-1-4419-7940-7.

[18] John M. Lee. "Introduction to smooth manifolds". In: *Choice Reviews Online.* Graduate Texts in Mathematics 40.08 (2003), pp. 40–4655. ISSN: 0009-4978. DOI: 10.5860/choice.40-4655. URL: http://link.springer.com/10.1007/978-1-4419-9982-5.

[19] Julien Munier. "Steepest descent method on a Riemannian manifold: The convex case". In: *Balkan Journal of Geometry and its Applications* 12.2 (2007), pp. 98–106. ISSN: 12242780.

[20] *Orthogonal Projection.* URL: https://textbooks.math.gatech.edu/ila/projections.html.

[21] Rajendra Bhatia. *Graduate Texts in Mathematics 169.* 1997. ISBN: 9781441979391.

[22] Clifford Henry Taubes. "Smooth manifolds". In: *Differential Geometry.* Compact Textbooks in Mathematics. Cham: Springer International Publishing, 2013, pp. 1–13. ISBN: 978-3-030-49774-3. DOI: 10.

1093/acprof:oso/9780199605880.003.0001. URL: http://link.springer.com/10.1007/978-3-030-49775-0.

[23] *Topology – from Wolfram MathWorld.* URL: https://mathworld.wolfram.com/Topology.html.

[24] Hongyi Zhang and Suvrit Sra. "First-order methods for geodesically convex optimization". In: *Journal of Machine Learning Research* 49.June (2016), pp. 1617–1638. ISSN: 15337928.