

# Étude du groupe de Lie $SO(n)$ dans le cadre de l'analyse de formes

Rapport de stage recherche

Antonin Lecocq

Sous la direction de Rayane Mouhli

Licence 2 Mathématiques-Informatique

*Année universitaire 2025–2026*

# Table des matières

|                                                                                           |           |
|-------------------------------------------------------------------------------------------|-----------|
| <b>Introduction</b>                                                                       | <b>1</b>  |
| <b>1 Le modèle de recalage LDDMM</b>                                                      | <b>2</b>  |
| 1.1 Déformations et Difféomorphismes . . . . .                                            | 2         |
| 1.2 Espace des formes, tangent et cotangent . . . . .                                     | 3         |
| 1.3 Problème variationnel LDDMM . . . . .                                                 | 4         |
| 1.4 Formulation Hamiltonienne . . . . .                                                   | 5         |
| 1.4.1 Hamiltonien réduit . . . . .                                                        | 5         |
| 1.4.2 Équations géodésiques . . . . .                                                     | 5         |
| <b>2 Le groupe <math>SO(n)</math> et son algèbre de Lie <math>\mathfrak{so}(n)</math></b> | <b>7</b>  |
| 2.1 Le groupe de Lie $SO(n)$ . . . . .                                                    | 7         |
| 2.1.1 Propriétés de l'action du groupe $SO(n)$ sur $\mathbb{R}^n$ . . . . .               | 10        |
| 2.1.2 Propriétés topologiques de $SO(n)$ . . . . .                                        | 12        |
| 2.2 L'algèbre de Lie $\mathfrak{so}(n)$ . . . . .                                         | 14        |
| 2.3 L'application exponentielle . . . . .                                                 | 17        |
| <b>3 Étude de l'équation différentielle linéaire <math>\dot{x}(t) = Ax(t)</math></b>      | <b>20</b> |
| 3.1 Obtenir l'équation différentielle $\dot{x}(t) = Ax(t)$ . . . . .                      | 20        |
| 3.2 Solution de $\dot{x}(t) = Ax(t)$ . . . . .                                            | 21        |
| <b>4 Formulation Hamiltonienne du recalage sur <math>SO(n)</math></b>                     | <b>23</b> |
| 4.1 Espaces tangent et cotangent des nuages de points . . . . .                           | 23        |
| 4.2 Hamiltonien sur $SO(n)$ . . . . .                                                     | 24        |
| 4.3 Équations géodésiques . . . . .                                                       | 24        |
| <b>5 Cas particuliers de <math>SO(2)</math> et <math>SO(3)</math></b>                     | <b>26</b> |
| 5.1 $SO(2)$ , Les rotations en dimension 2 . . . . .                                      | 26        |
| 5.2 $SO(3)$ , Les rotations en dimension 3 . . . . .                                      | 29        |
| <b>6 Implémentation algorithmique</b>                                                     | <b>33</b> |
| 6.1 Approche directe - Discrétiser puis Optimiser . . . . .                               | 33        |
| 6.1.1 Recalage rigide sur $SO(2)$ . . . . .                                               | 33        |
| 6.2 Approche hamiltonienne - Tir géodésique . . . . .                                     | 37        |
| 6.2.1 Recalage rigide sur $SO(2)$ . . . . .                                               | 38        |
| <b>Conclusion</b>                                                                         | <b>40</b> |
| <b>Références</b>                                                                         | <b>41</b> |

|          |                                |           |
|----------|--------------------------------|-----------|
| <b>A</b> | <b>Code Python</b>             | <b>42</b> |
| <b>B</b> | <b>Figures supplémentaires</b> | <b>44</b> |

# Introduction

Ce rapport contient le travail réalisé pendant un stage consacré à l'étude mathématique et numérique d'un problème de recalage de formes. Le recalage consiste à déterminer une transformation permettant d'aligner une forme dite source sur une forme dite cible. Ce type de problème apparaît dans de nombreux domaines, par exemple en imagerie médicale ou encore en analyse morphologique.

Le cadre général étudié durant ce stage est celui du *Large Deformation Diffeomorphic Metric Mapping*, ou LDDMM [1][7]. Cette approche sera détaillée dans le **Chapitre 1**. L'intérêt principal de cette méthode est de garantir que les transformations obtenues préservent la structure topologique des objets étudiés : il n'y a ni déchirure ni fusion de formes.

Le modèle LDDMM complet est mathématiquement et numériquement riche. Dans ce stage, nous avons choisi d'en étudier une restriction plus simple : le cas où les transformations admissibles considérées sont uniquement des rotations [2]. Cette restriction conduit naturellement à l'étude du groupe spécial orthogonal  $SO(n)$  et de son algèbre de Lie  $\mathfrak{so}(n)$ . On s'intéressera donc, dans un cadre en dimension finie, à comment on transporte des formes via une action de groupe.

L'objectif du rapport est de présenter les outils théoriques : groupes de Lie, algèbres de Lie, exponentielle de matrice et équations différentielles linéaires. Aussi, il s'agit d'illustrer ces notions par une implémentation numérique en Python, appliquée à des exemples simples de recalage par rotation dans le plan. Ainsi, le rapport suit naturellement la progression adoptée pendant le stage. Le premier chapitre rappelle les grandes idées du modèle *Large Deformation Diffeomorphic Metric Mapping* et introduit le problème variationnel de recalage. Le deuxième chapitre est consacré à l'étude du groupe  $SO(n)$ , à ses propriétés géométriques et topologiques, ainsi qu'à son algèbre de Lie. Le troisième chapitre montre comment l'équation différentielle  $\dot{x}(t) = Ax(t)$  apparaît naturellement dans le contexte du recalage sur des landmarks et comment elle se résout à l'aide de l'exponentielle de matrice. Le quatrième chapitre présente une formulation Hamiltonienne du problème de recalage pour les rotations. Le cinquième chapitre détaille le cas particulier de  $SO(2)$  et  $SO(3)$ , qui permet d'obtenir une description explicite des rotations. Enfin, le dernier chapitre présente l'implémentation algorithmique et discute les résultats obtenus suite à l'exécution de ces algorithmes.

# 1 Le modèle de recalage LDDMM

Parmi les nombreuses approches de recalage, nous nous concentrerons sur le *Large Deformation Diffeomorphic Metric Mapping*<sup>1</sup>. Sa force réside dans sa capacité à modéliser des déformations tout en garantissant que la transformation obtenue est un *difféomorphisme* – c’est-à-dire une application bijective, lisse et d’inverse lisse. Cette propriété est cruciale pour préserver la topologie des structures (anatomiques par exemple) : des ensembles connexes restent connexes, des surfaces lisses restent lisses, et il n’y a pas de nouvelles pliures ou de déchirures. Le cadre de *Large Deformation Diffeomorphic Metric Mapping* fournit une formulation variationnelle du problème de recalage, où la transformation recherchée est obtenue comme solution d’un problème de minimisation d’énergie.

L’objectif de ce chapitre est de présenter le modèle LDDMM.

## 1.1 Déformations et Difféomorphismes

En analyse de formes, on cherche à transformer une forme source  $q_S$  vers une forme cible  $q_T$ . Pour pouvoir effectuer cette transformation, on veut comparer ces deux formes en prenant en compte leurs propriétés géométriques. Et pour préserver la topologie de la forme, on ne déforme pas l’objet directement, mais l’espace ambiant. En effet, si on considère un champ de vitesse  $v_t$ , alors la solution de l’équation différentielle (que nous verrons par la suite) est un flot de difféomorphismes appelé flot engendré par  $v$ , qui envoie la forme source sur la forme cible [6]. Cette approche permet d’obtenir naturellement un difféomorphisme de tout l’espace.

**Définition 1.1** (Flot de difféomorphismes).

On considère une famille de difféomorphismes

$$\varphi_t : \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad t \in [0, 1],$$

engendrée par un champ de vitesse dépendant du temps  $v_t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ .

Le flot est défini comme la solution de l’équation différentielle :

$$\begin{cases} \dot{\varphi}_t = v_t \circ \varphi_t, \\ \varphi_0 = Id. \end{cases}$$

Cette équation décrit le transport des points de l’espace sous l’action du champ de vitesse  $v_t$ .

L’ensemble des difféomorphismes est un groupe, et la déformation à l’instant  $t$  est donc donnée par l’action de groupe des difféomorphismes sur les formes :  $q_T = \varphi_t \cdot q_S$ . Cette action dépend de la nature géométrique de la forme étudiée [9].

---

1. Que l’on appellera LDDMM par la suite

- **Nuage de points (Landmarks)** : Si  $q_S = (x_1, \dots, x_N) \in (\mathbb{R}^d)^N$ , l'action est l'évaluation ponctuelle :  $\varphi \cdot q_S := \varphi(q_S)$ .
- **Courbes paramétrées** : Si  $q \in \mathcal{C}^1(I; \mathbb{R}^d)$ , l'action est la composition à gauche :  $\varphi \cdot q := \varphi \circ q$ .
- **Images** : Si  $q_S \in \mathcal{L}^2(\Omega, \mathbb{R})$ , l'action est la composition avec l'inverse (pour préserver l'intensité) :  $\varphi \cdot q_S := q_S \circ \varphi^{-1}$ .

**Exemple 1.2.** Voyons un exemple de recalage rigide avec une forme source  $x_S$  et une autre forme cible  $x_T$  séparées par une rotation (cet exemple sera détaillé plus loin (6.2)). La forme source, à laquelle on a appliqué une rotation, recouvre parfaitement la forme cible :



FIGURE 1.1 – Recalage sur les rotations  $SO(2)$  d'un coq.

## 1.2 Espace des formes, tangent et cotangent

Dans la formulation géométrique du recalage, on considère un espace de champs de vecteurs  $V \hookrightarrow C_0^2(\mathbb{R}^d, \mathbb{R}^d)$  et un espace de formes  $Q$ . Cet espace peut être, selon le problème étudié, un espace d'images, de courbes, de surfaces ou encore de nuages de points. Dans le cas des nuages de points, l'espace des formes est

$$Q = (\mathbb{R}^d)^N.$$

Cet espace s'identifie naturellement à  $\mathbb{R}^{dN}$ . C'est donc un espace vectoriel normé de dimension finie où  $N$  désigne le nombre de points et  $d$  la dimension de l'espace ambiant. Comme tout espace vectoriel normé de dimension finie est complet,  $\mathbb{R}^{dN}$  est un espace de Banach. Ainsi,  $Q$  peut être vu comme une variété de Banach<sup>2</sup>.

Pour décrire l'évolution des formes, on a besoin d'introduire les espaces tangents et cotangents.

**Définition 1.3** (Espace tangent de  $Q$ ).

Puisque  $Q$  est une variété (de Banach), on peut lui associer en tout point  $q$  un espace tangent  $T_q Q$ , au sens de la définition générale (2.2). On note

$$TQ = \bigcup_{q \in Q} T_q Q = \{(q, v) \mid q \in Q, v \in T_q Q\}.$$

---

2. Variété de Banach : Espace localement semblable à un espace de Banach

**Définition 1.4** (Espace cotangent de  $Q$ ).

On définit son dual, l'espace cotangent  $T_q^*Q$ , qui est l'ensemble des formes linéaires sur  $T_qQ$ . On le définit par

$$T^*Q = \bigcup_{q \in Q} T_q^*Q = \{(q, p) \mid q \in Q, p \in T_q^*Q\}.$$

L'élément  $p \in T_q^*Q$  est appelé un covecteur. Dans le cadre hamiltonien, on appelle ça un moment.

## 1.3 Problème variationnel LDDMM

Le problème de recalage qui consiste à trouver la meilleure déformation peut s'écrire comme un problème variationnel où l'on doit déterminer le champ de vitesse  $v_t$  qui minimise une énergie composée de deux termes :

- Un terme de régularité contrôlant le “coût” de la déformation (à gauche de l'addition),
- Un terme d'attache aux données mesurant la similarité entre l'image transformée et la cible (à droite).

Le premier terme favorise des déformations douces, tandis que le second impose que la forme finale reste proche de la cible.

Le problème s'écrit :

$$\inf_{v \in L^2([0,1], V)} J(v) = \frac{1}{2} \int_0^1 \|v_t\|_V^2 dt + \|\varphi_1 \cdot q_S - q_T\|_{L^2}$$

$$\text{tel que } \begin{cases} \dot{\varphi}_t = v_t \circ \varphi_t \\ \varphi_0 = Id \end{cases}$$

**Théorème 1.5.** *Notre problème ci-dessus admet un minimum.*

*Preuve.* Nous admettons le théorème et ne démontrons donc pas ce résultat ici. Piste de preuve : On utilise une méthode de calcul variationnel [5], “Direct method in the calculus of variations” pour montrer que l'équation de minimisation admet un minimum, et en plus, le minimum est global.  $\square$

Réécrivons le problème en utilisant le *Théorème 1.5*. On note  $q_t = \varphi \cdot q_S$  la forme à l'instant  $t$ . L'action du champ de vecteurs  $v_t$  sur la forme est notée  $v_t \circ q_t$ .

La dynamique du problème de recalage s'écrit alors

$$\dot{q}_t = v_t \circ q_t.$$

Le problème variationnel peut donc s'écrire sous la forme :

$$\min_{v \in L^2([0,1], V)} \int_0^1 \|v_t\|_V^2 dt + \mathcal{D}(q_1, q_T)$$

$$\text{tel que } \begin{cases} \dot{q}_t = v_t \circ q_t \\ q_0 = q_S \end{cases}$$

avec  $\mathcal{D}$  une métrique adaptée aux données.

Ici,  $q_S$  désigne encore la forme source et  $q_T$  la forme cible.

*Remarque 1.6.* Pour appliquer la méthode LDDMM, on n'a besoin que de l'action du groupe des difféomorphismes sur l'espace de formes. La formulation du problème de minimisation repose uniquement sur l'application du groupe à cet espace via  $q_S \mapsto \varphi_1 \cdot q_S$ .

## 1.4 Formulation Hamiltonienne

On peut interpréter le problème LDDMM comme un problème de contrôle optimal [8]. Dans ce cadre, le champ de vecteurs  $v_t$  est le contrôle, tandis que la forme  $q_t$  représente l'état du système. Son évolution est donnée par

$$\dot{q}_t = v_t \circ q_t.$$

Pour traiter ce problème contraint, on introduit un covecteur  $p_t \in T_{q_t}^*Q$ , appelé moment. Le Hamiltonien associé au problème de contrôle optimal noté

$$H : T^*Q \times V \longrightarrow \mathbb{R}$$

est défini par

$$H(q, p, v) = (p \mid v \circ q) - \frac{1}{2} \|v\|_V^2.$$

Le premier terme mesure l'interaction entre le moment  $p$  et la vitesse de déformation  $v \circ q$ , tandis que le second représente le coût énergétique du champ de vecteurs  $v$ .

### 1.4.1 Hamiltonien réduit

Le principe du maximum de Pontryagin [10] impose que le contrôle optimal maximise ce Hamiltonien par rapport à  $v$ . Pour obtenir cet Hamiltonien réduit en fonction de  $q$  et  $p$ , on cherche d'abord le champ de vecteurs optimal  $v^*$ . Celui-ci est obtenu en annulant la dérivée de  $H$  par rapport à la variable de contrôle  $v$  :

$$\partial_v H(q, p, v) = 0.$$

On obtient ainsi un champ de vecteurs optimal

$$v^* = v^*(q, p).$$

On remplace ensuite cette expression dans le hamiltonien initial pour définir le hamiltonien réduit

$$H_r(q, p) = H(q, p, v^*(q, p)).$$

Cette réduction permet de transformer le problème de contrôle optimal qui est en dimension infinie en un système d'équations différentielles sur  $T^*Q$  qui est un problème en dimension finie.

### 1.4.2 Équations géodésiques

Les candidats à l'optimalité sont les géodésiques, c'est-à-dire des trajectoires satisfaisant les équations hamiltoniennes suivantes :

$$\dot{q}_t = \partial_p H_r(q_t, p_t),$$

$$\dot{p}_t = -\partial_q H_r(q_t, p_t).$$

La première équation décrit l'évolution de la forme. La seconde décrit l'évolution du moment associé. Dans le cadre LDDMM, la première équation redonne la dynamique

$$\dot{q}_t = v_t \circ q_t.$$

*Remarque 1.7.* L'intérêt principal de la formulation hamiltonienne est qu'elle permet de remplacer l'optimisation sur le champ de vitesse  $v_t$ , qui appartient à un espace de dimension infinie, par l'étude d'un système différentiel portant sur les variables  $(q_t, p_t)$ .

Ce chapitre a posé les fondements nécessaires à l'étude des cas particuliers, comme celui des rotations  $SO(2)$  et  $SO(3)$ , qui seront abordés dans les chapitres suivants. En effet, dans ce rapport nous ne traitons pas le modèle LDDMM complet. Nous étudions une restriction de ce cadre dans le cas où les transformations considérées sont uniquement des rotations. Cette restriction permet d'obtenir un modèle plus simple que le modèle LDDMM complet qui repose sur l'action du groupe des difféomorphismes en dimension infinie.

## 2 Le groupe $SO(n)$ et son algèbre de Lie $\mathfrak{so}(n)$

Dans ce chapitre, nous nous intéresserons aux particularités du groupe  $SO(n)$  et de son algèbre de Lie  $\mathfrak{so}(n)$  pour  $n \geq 2$ . Voici quelques définitions pour commencer.

**Définition 2.1** (Variété différentielle).

Une **variété différentielle** est un espace qui est localement assimilable<sup>1</sup> à un espace euclidien [3].

Considérons  $G$  une variété différentielle.

**Définition 2.2** (Espace tangent).

On appelle **espace tangent** en  $g$  à  $G$ , que l'on note  $T_g G$ , l'ensemble des vecteurs tangents en  $g$ .

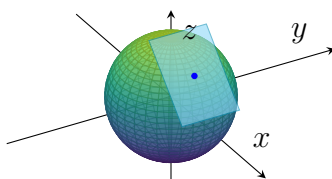


FIGURE 2.1 – Figure du plan tangent à la sphère en un point.

### 2.1 Le groupe de Lie $SO(n)$

Le **groupe spécial orthogonal**  $SO(n)$  est un sous-groupe du **groupe orthogonal**  $O(n)$ . C'est le groupe des matrices de rotation à  $n$  dimensions. On peut l'écrire :

$$SO(n) = \{A \in \mathcal{M}_n(\mathbb{R}) \mid \det A = 1 ; A^\top A = I_n\}$$

**Définition 2.3** (Groupe de Lie).

Un **groupe de Lie** est une variété différentielle qui est également un groupe, de telle sorte que l'opération de composition du groupe et l'opération d'inversion sont des applications lisses<sup>2</sup> [3].

**Théorème 2.4** (Théorème de submersion).

Soient  $f \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^m)$  et  $G := f^{-1}(e)$  où  $e \in \mathbb{R}^m$ .

Si  $f$  est une submersion (i.e. si  $Df(x)$  est surjective en tout point  $x$  de  $G$ ), alors  $G$  est une sous-variété de l'espace  $\mathbb{R}^n$  de dimension  $n - m$ .

1. On pourrait aussi dire : localement homéomorphe à  $\mathbb{R}^n$

2. Lisse : de classe  $\mathcal{C}^\infty$

*Preuve.* Admis

□

**Proposition 2.5.**  $SO(n)$  est un groupe de Lie

*Preuve.* Nous montrons d'abord que  $SO(n)$  est un groupe (1), puis qu'il possède une structure de variété différentielle compatible avec les opérations de groupe (2).

(1)  $SO(n)$  est un groupe pour le produit matriciel.

Soient  $A, B \in SO(n)$ , montrons que le produit est stable :

$$(AB)^\top (AB) = B^\top A^\top AB = B^\top B = I_n \quad \text{et} \quad \det(AB) = \det(A) \det(B) = 1.$$

Ainsi  $AB \in SO(n)$ .

L'associativité est celle du produit matriciel.

L'élément neutre est la matrice identité  $I_n$ , qui appartient à  $SO(n)$  puisque

$$I_n^\top I_n = I_n \quad \text{et} \quad \det(I_n) = 1.$$

Enfin, si  $A \in SO(n)$ , alors

$$A^\top A = I_n,$$

d'où

$$A^{-1} = A^\top.$$

Comme

$$\det(A^{-1}) = \frac{1}{\det(A)} = 1,$$

on obtient  $A^{-1} \in SO(n)$ .

Ainsi  $SO(n)$  est un groupe et le produit matriciel est bien une loi de composition interne sur l'ensemble  $SO(n)$ .

(2)  $SO(n)$  est une variété différentielle.

Montrons que  $SO(n)$  est muni d'une structure de variété différentielle, comme sous-variété de l'espace  $\mathcal{M}_n(\mathbb{R})$ .

Considérons l'application

$$\begin{aligned} f: \mathcal{M}_n(\mathbb{R}) &\longrightarrow \mathcal{S}_n(\mathbb{R}) \\ X &\longmapsto X^\top X \end{aligned}$$

où  $\mathcal{S}_n(\mathbb{R})$  désigne l'espace des matrices symétriques.

On remarque que

$$O(n) = f^{-1}(I_n).$$

Calculons la différentielle de  $f$ .

Soient  $X \in O(n)$ , et  $H \in \mathcal{M}_n(\mathbb{R})$  :

$$\begin{aligned} f(X + H) &= (X + H)^\top (X + H) \\ &= X^\top X + X^\top H + H^\top X + H^\top H \\ &= f(X) + X^\top H + H^\top X + H^\top H \end{aligned}$$

Comme la norme subordonnée est sous-multiplicative

$$\|H^\top H\| \leq \|H^\top\| \|H\| = \|H\|^2,$$

on a

$$\begin{aligned} f(X + H) &= f(X) + X^\top H + H^\top X + O(\|H\|^2) \\ &= f(X) + Df(X)H + O(\|H\|^2) \end{aligned}$$

La différentielle de  $f$  en  $X$  est donc donnée par la partie linéaire en  $H$ ,

$$Df(X)H = X^\top H + H^\top X.$$

Il reste à montrer que  $f$  est une submersion en tout point de  $O(n)$ , donc que  $Df(X)$  est surjective :

Soit  $Y \in \mathcal{S}_n(\mathbb{R})$ , déterminons  $H$

$$Df(X)H = Y \Leftrightarrow X^\top H + H^\top X = Y.$$

Posons  $H = \frac{1}{2}XY$  de telle sorte que pour  $X \in O(n)$

$$\begin{aligned} Df(X)\frac{1}{2}XY &= Y \Leftrightarrow X^\top \frac{1}{2}XY + (\frac{1}{2}XY)^\top X = Y \\ &\Leftrightarrow \frac{1}{2}X^\top XY + \frac{1}{2}Y^\top X^\top X = Y \end{aligned}$$

or  $X^\top X = I_n$  et  $Y^\top = Y$ , donc

$$Df(X)\frac{1}{2}XY = \frac{1}{2}I_n Y + \frac{1}{2}Y I_n = Y.$$

On a trouvé pour n'importe quel  $Y$ , qu'il y a un antécédent  $H$ . Ainsi,  $Df$  est surjective et donc  $O(n)$  est bien une sous-variété de l'espace  $\mathcal{M}_n(\mathbb{R})$  (par le *Théorème de submersion 2.4*).

Posons  $U = \{A \in \mathcal{M}_n(\mathbb{R}) \mid \det(A) > 0\}$ . Comme le déterminant est continu,  $U$  est un ouvert de  $\mathcal{M}_n(\mathbb{R})$ . Or l'intersection d'une sous-variété avec un ouvert de l'espace ambiant est encore une sous-variété. Puisque

$$SO(n) = O(n) \cap U,$$

on en déduit que  $SO(n)$  est une sous-variété de  $\mathcal{M}_n(\mathbb{R})$ .

Enfin, la structure de groupe de  $SO(n)$  est compatible avec sa structure différentielle. En effet, l'application

$$\begin{aligned} \mu: SO(n) \times SO(n) &\longrightarrow SO(n), \\ (M, N) &\longmapsto MN^{-1} \end{aligned}$$

est lisse. Pour tout  $N \in SO(n)$ , on a  $N^{-1} = N^\top$ . Ainsi,

$$\mu(M, N) = MN^\top.$$

La transposition étant linéaire et la multiplication matricielle étant bilinéaire, ces deux opérations sont lisses. Par conséquent,  $\mu$  est lisse. On a  $SO(n)$  est un groupe de Lie.  $\square$

*Remarque 2.6.* Comme  $SO(n) = O(n) \cap \{A \in \mathcal{M}_n(\mathbb{R}) \mid \det(A) > 0\}$ , le groupe  $SO(n)$  est un ouvert de la variété  $O(n)$ . Il possède donc la même dimension que  $O(n)$  et, en chaque point  $X \in SO(n)$ , le même espace tangent.

Par ailleurs,  $O(n) = f^{-1}(I_n)$ , où

$$\begin{aligned} f: \mathcal{M}_n(\mathbb{R}) &\longrightarrow \mathcal{S}_n(\mathbb{R}) \\ X &\longmapsto X^\top X \end{aligned}$$

Comme  $I_n$  est une valeur régulière de  $f$ , le théorème des valeurs régulières donne

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

### 2.1.1 Propriétés de l'action du groupe $SO(n)$ sur $\mathbb{R}^n$

Soient  $G$  un groupe et  $X$  un ensemble.

**Définition 2.7** (Action de groupe).

On dit que  $G$  agit à gauche sur  $X$  s'il existe une application  $u: G \times X \longrightarrow X$  vérifiant

- $u(e, x) = x$  pour tout  $x \in X$  avec  $e$  l'élément neutre de  $G$ ,
- pour tout  $g, g' \in G$  et  $x \in X$

$$u(g, u(g', x)) = u(gg', x).$$

*Remarque 2.8.* On définit de même l'action à droite par la propriété  $v(g, v(g', x)) = v(g'g, x)$ .

**Proposition 2.9.** *L'action naturelle du groupe  $SO(n)$  sur  $\mathbb{R}^n$  est donnée par la multiplication matricielle :*

$$\begin{aligned}u: SO(n) \times \mathbb{R}^n &\longrightarrow \mathbb{R}^n \\ (R, x) &\longmapsto Rx\end{aligned}$$

*Preuve.* Soient  $R, R' \in SO(n)$  et  $x \in \mathbb{R}^n$ ,

$$u(I_n, x) = I_n x = x.$$

Et

$$u(R, u(R', x)) = u(R, R'x) = RR'x = u(RR', x).$$

La multiplication matricielle  $u$  est une action de groupe. □

**Proposition 2.10.** *L'action naturelle de  $SO(n)$  sur  $\mathbb{R}^n$  est fidèle, mais elle n'est ni libre ni transitive*

*Preuve.* L'action de  $SO(n)$  dans  $\mathbb{R}^n$  est **fidèle** :

Si l'action est fidèle, alors si  $R \in SO(n)$  tel que  $u(R, x) = x, \forall x \in \mathbb{R}^n$  alors  $R = I_n$ .

Supposons que  $R \in SO(n)$  vérifie  $Rx = x$  pour tout  $x \in \mathbb{R}^n$ . Alors, pour tout vecteur de la base canonique de  $\mathbb{R}^n : (e_1, \dots, e_n)$ , on a

$$Re_i = e_i \quad \forall 1 \leq i \leq n.$$

Ainsi, les colonnes de  $R$  sont celles de l'identité, donc  $R = I_n$ . L'action est fidèle.

L'action de  $SO(n)$  dans  $\mathbb{R}^n$  n'est **pas libre** :

Si l'action est libre, alors pour tout  $x \in \mathbb{R}^n$ , si  $x \neq 0$  et  $R \neq I_n$  alors  $Rx \neq x$ .

Elle n'est pas libre sur  $\mathbb{R}^n$ , puisque  $R0 = 0$  pour tout  $R \in SO(n)$ .

Sur  $\mathbb{R}^2 \setminus \{0\}$ , si une rotation fixe un vecteur non nul, son angle est nul modulo  $2\pi$ , donc elle est égale à l'identité.

En revanche, pour  $n \geq 3$ , choisissons

$$R = \begin{pmatrix} 1 & 0 \\ 0 & S \end{pmatrix}, \quad S \in SO(n-1), \quad S \neq I_{n-1}.$$

Alors  $R \neq I_n$  mais  $Re_1 = e_1$ . Donc l'action de  $SO(n)$  dans  $\mathbb{R}^n$  n'est pas libre.

L'action de  $SO(n)$  dans  $\mathbb{R}^n$  n'est **pas transitive** :

Si l'action est transitive, alors  $\forall x, y \in \mathbb{R}^n, \exists R \in SO(n)$  tel que  $u(R, x) = y$ .

Prenons  $x = 0$  et  $y = (1, 0, \dots, 0)$ .

Alors  $\forall R \in SO(n), u(R, x) = Rx = R \cdot 0 = 0 \neq y$ .

Donc l'action de  $SO(n)$  dans  $\mathbb{R}^n$  n'est pas transitive. □

**Proposition 2.11.** *L'action de  $SO(n)$  sur  $\mathbb{R}^n$  est isométrique. C'est-à-dire :*

$$\forall x, y \in \mathbb{R}^n, \forall R \in SO(n), \langle Rx, Ry \rangle = \langle x, y \rangle.$$

*Preuve.* Soient  $x, y \in \mathbb{R}^n$  et  $R \in SO(n)$

$$\langle Rx, Ry \rangle = \langle x, R^\top Ry \rangle.$$

Or  $R^\top R = I_n$  car  $R \in SO(n)$ , donc

$$\begin{aligned} \langle Rx, Ry \rangle &= \langle x, I_n y \rangle \\ &= \langle x, y \rangle. \end{aligned}$$

L'action de  $SO(n)$  sur  $\mathbb{R}^n$  est isométrique, elle préserve bien la distance. □

**Définition 2.12** (Orbite d'une action).

On appelle **orbite de  $x$**  la partie de l'ensemble  $X$  définie par

$$Orb(x) = \{u(g, x) \mid g \in G\}.$$

**Proposition 2.13.** *Les orbites de l'action naturelle de  $SO(n)$  sur  $\mathbb{R}^n$  sont les sphères centrées en l'origine. Plus précisément, pour tout  $x \in \mathbb{R}^n$ ,*

$$Orb(x) = \{Rx \mid R \in SO(n)\} = \{y \in \mathbb{R}^n \mid \|y\| = \|x\|\}.$$

*En particulier, si  $x = 0$ , alors*

$$Orb(0) = \{0\}.$$

*Preuve.* Soit  $x \in \mathbb{R}^n$ . Montrons par double inclusion :

$Orb(x) \subset \{z \in \mathbb{R}^n \mid \|z\| = \|x\|\}$  : Soit  $y \in Orb(x)$ . Alors il existe  $R \in SO(n)$  tel que

$$y = Rx.$$

Comme l'action de  $SO(n)$  sur  $\mathbb{R}^n$  est isométrique, on a

$$\|y\| = \|Rx\| = \|x\|.$$

Donc

$$y \in \{z \in \mathbb{R}^n \mid \|z\| = \|x\|\}.$$

Ainsi,

$$\text{Orb}(x) \subset \{z \in \mathbb{R}^n \mid \|z\| = \|x\|\}.$$

$\text{Orb}(x) \supset \{z \in \mathbb{R}^n \mid \|z\| = \|x\|\}$  : Soit maintenant  $y \in \mathbb{R}^n$  tel que

$$\|y\| = \|x\|.$$

Si  $x = 0$ , alors  $\|y\| = 0$ , donc  $y = 0$ . On a donc bien

$$\text{Orb}(0) = \{0\}.$$

Supposons maintenant que  $x \neq 0$ . Alors  $y \neq 0$  également. Les vecteurs

$$\frac{x}{\|x\|} \quad \text{et} \quad \frac{y}{\|y\|}$$

sont deux vecteurs unitaires de  $\mathbb{R}^n$ . On peut compléter chacun d'eux en une base ortho-normée directe de  $\mathbb{R}^n$  :

$$\mathcal{B}_x = \left( \frac{x}{\|x\|}, e_2, \dots, e_n \right), \quad \mathcal{B}_y = \left( \frac{y}{\|y\|}, f_2, \dots, f_n \right).$$

Il existe alors une matrice  $R \in SO(n)$  envoyant la base  $\mathcal{B}_x$  sur la base  $\mathcal{B}_y$ . En particulier,

$$R \left( \frac{x}{\|x\|} \right) = \frac{y}{\|y\|}.$$

Comme  $\|x\| = \|y\|$ , on obtient

$$Rx = y.$$

Donc  $y \in \text{Orb}(x)$ .

Ainsi,

$$\{z \in \mathbb{R}^n \mid \|z\| = \|x\|\} \subset \text{Orb}(x).$$

Finalement, les deux inclusions donnent  $\text{Orb}(x) = \{y \in \mathbb{R}^n \mid \|y\| = \|x\|\}$ . □

## 2.1.2 Propriétés topologiques de $SO(n)$

**Théorème 2.14** (Compacité en dimension finie).

*Soient  $(X, \|\cdot\|)$  un espace vectoriel normé de dimension finie et  $E \subset X$  une partie de  $X$ . Alors  $E$  est compacte si et seulement si  $E$  est fermée et bornée.*

*Preuve.*  $\Rightarrow$  Une partie compacte est aussi fermée et bornée. Cette implication est donc vraie dans n'importe quel espace.

$\Leftarrow$  C'est une conséquence du Théorème de Bolzano-Weierstrass. □

**Proposition 2.15.**  *$SO(n)$  est un groupe compact.*

*Preuve.* On a  $SO(n) \subset \mathcal{M}_n(\mathbb{R})$  qui est en dimension finie.

Montrons que  $SO(n)$  est fermé (1) et borné (2) :

(1)  $SO(n)$  **fermé**.

Soient  $f$  et  $g$  deux applications continues.

$$\begin{array}{ll} f: \mathcal{M}_n(\mathbb{R}) \longrightarrow \mathbb{R} & g: \mathcal{M}_n(\mathbb{R}) \longrightarrow \mathcal{M}_n(\mathbb{R}) \\ X \longmapsto \det(X) & X \longmapsto X^\top X \end{array}$$

On pose  $\{1\} \subset \mathbb{R}$  et  $\{I_n\} \subset \mathcal{M}_n(\mathbb{R})$ , deux fermés (car ce sont des singletons). Alors  $SO(n) = f^{-1}(\{1\}) \cap g^{-1}(\{I_n\})$ , comme l'intersection de deux fermés (images réciproques de fermés par des applications continues). Ainsi  $SO(n)$  est fermé.

(2)  $SO(n)$  **borné.**

Soit  $A \in SO(n)$ , alors  $\|A\|_F^2 = \text{Tr}(A^\top A) = \text{Tr}(I_n) = n$ .

On a donc :  $\forall A \in SO(n), \exists M = \sqrt{n}$  tel que  $\|A\| \leq M$  (toutes les normes sont équivalentes en dimension finie).

On retrouve que  $SO(n)$  est borné car  $SO(n) \subset B(0, M)$ .

Finalement,  $SO(n)$  fermé et borné donc (par le *Théorème de compacité en dimension finie 2.14*)  $SO(n)$  est compact.  $\square$

**Proposition 2.16.**  *$SO(n)$  est un groupe connexe par arcs.*

*Preuve.* Le groupe  $SO(n)$  est compact (2.15) et c'est aussi une variété différentielle (2.5). Donc, il est donc localement connexe par arcs et possède un nombre finie de composantes connexes par arcs.

Il ne reste plus qu'à prouver qu'il n'y a qu'une unique composante :

Notons  $C$  la composante connexe par arcs de l'identité  $I_n$  dans  $SO(n)$ . Puisque  $SO(n)$  est une variété différentielle, il existe  $U$  un voisinage ouvert de  $I_n$  qui est homéomorphe à un ouvert de  $\mathbb{R}^{n(n-1)/2}$ . On peut choisir  $U$  connexe par arcs, de sorte que  $U \subset C$ .

Par ailleurs, pour toute matrice  $A \in SO(n)$ , la translation à gauche

$$\begin{aligned} L_A: SO(n) &\longrightarrow SO(n) \\ M &\longmapsto AM \end{aligned}$$

est un homéomorphisme. Elle envoie donc la composante  $C$  de  $I_n$  sur la composante  $AC$  de  $A$ . En particulier, toutes les composantes connexes par arcs de  $SO(n)$  sont ouvertes. Pour montrer qu'il n'existe qu'une seule composante, considérons  $A \in SO(n)$ . Puisque  $A$  est une matrice orthogonale, il existe une matrice  $P \in O(n)$  et des réels  $\theta_1, \dots, \theta_r$  tels que

$$P^\top A P = \text{diag}(R(\theta_1), \dots, R(\theta_r), 1, \dots, 1),$$

où

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

En effet, les éventuels blocs  $(-1)$  apparaissent en nombre pair, car  $\det(A) = 1$ , et peuvent donc être regroupés deux à deux sous la forme

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = R(\pi).$$

Définissons alors l'application

$$\begin{aligned} \gamma: [0, 1] &\longrightarrow SO(n) \\ t &\longmapsto P \text{diag}(R(t\theta_1), \dots, R(t\theta_r), 1, \dots, 1) P^\top \end{aligned}$$

Cette application est continue. De plus, chaque matrice  $\gamma(t)$  est orthogonale et vérifie

$$\det(\gamma(t)) = 1,$$

donc  $\gamma(t) \in SO(n)$  pour tout  $t \in [0, 1]$ . Enfin,

$$\gamma(0) = I_n \quad \text{et} \quad \gamma(1) = A.$$

Ainsi, toute matrice  $A \in SO(n)$  appartient à la composante connexe par arcs  $C$  de l'identité. On en déduit que

$$C = SO(n).$$

Par conséquent,  $SO(n)$  ne possède qu'une seule composante connexe par arcs.

Et  $SO(n)$  est connexe par arcs. □

## 2.2 L'algèbre de Lie $\mathfrak{so}(n)$

L'outil principal qui nous permettra d'étudier les groupes de Lie est leur algèbre de Lie. Définissons cette structure pour les rotations et intéressons-nous aux propriétés différentielles des groupes de Lie. Prenons  $G$  un groupe de Lie et  $e \in G$  son élément neutre.

**Définition 2.17** (Algèbre de Lie).

Une **algèbre de Lie**, notée  $\mathfrak{g} = T_e G$ , est un espace vectoriel muni d'un produit appelé crochet de Lie, qui est une application bilinéaire antisymétrique [3]

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

et qui satisfait l'identité de Jacobi :

$$[\xi, [\eta, \zeta]] + [\eta, [\zeta, \xi]] + [\zeta, [\xi, \eta]] = 0, \quad \text{pour } (\xi, \eta, \zeta) \in \mathfrak{g}^3.$$

**Proposition 2.18.** *Les algèbres de Lie des groupes  $SO(n)$  et  $O(n)$  coïncident :*

$$\mathfrak{so}(n) = \mathfrak{o}(n)$$

*Preuve.* Montrons par double inclusion :

$\mathfrak{so}(n) \subset \mathfrak{o}(n)$  : Comme  $SO(n) \subset O(n)$ , tout vecteur tangent à  $SO(n)$  en l'identité est aussi tangent à  $O(n)$ .

$$T_{I_n} SO(n) \subset T_{I_n} O(n), \text{ donc } \mathfrak{so}(n) \subset \mathfrak{o}(n).$$

$\mathfrak{so}(n) \supset \mathfrak{o}(n)$  : On utilise le fait que  $SO(n)$  est la composante connexe de l'identité dans  $O(n)$  (2.16).

Au voisinage de  $I_n$ , la condition  $\det X = 1$  dans  $O(n)$  est automatiquement satisfaite sur la composante où  $\det X > 0$ . Comme le déterminant est continu et vaut 1 en  $I_n$ , il existe un voisinage de  $I_n$  dans lequel tout élément de  $O(n)$  a un déterminant positif, donc égal à 1 puisqu'un élément de  $O(n)$  a un déterminant  $\pm 1$ . Ainsi,  $O(n)$  et  $SO(n)$  coïncident localement au voisinage de  $I_n$ . Ils ont donc le même espace tangent en  $I_n$ .

Finalement, on a  $\mathfrak{so}(n) = \mathfrak{o}(n)$ . □

**Proposition 2.19.**  $T_{I_n} SO(n) = \{H \in \mathcal{M}_n(\mathbb{R}) \mid H^\top = -H\}$  est l'algèbre de Lie de  $SO(n)$ , plus précisément :  $\mathfrak{so}(n) = T_{I_n} SO(n)$ .

*Preuve.*  $SO(n)$  est un groupe de Lie (2.5). L'algèbre de Lie d'un groupe de Lie est définie comme son espace tangent en l'identité (ici  $Id$ ), que l'on trouve en différenciant  $f^3$  en l'identité avec

$$\begin{aligned} f: \mathcal{M}_n(\mathbb{R}) &\longrightarrow \mathcal{S}_n(\mathbb{R}) \\ X &\longmapsto X^\top X \end{aligned}$$

Par la preuve de la *proposition* (2.5), on a

$$Df(X)H = X^\top H + H^\top X.$$

On l'évalue en l'identité

$$Df(I_n)H = I_n^\top H + H^\top I_n = H + H^\top.$$

Et comme

$$O(n) = f^{-1}(I_n)$$

et que  $I_n$  est une valeur régulière de  $f$ , le théorème des valeurs régulières donne, pour tout  $X \in O(n)$ ,

$$T_X O(n) = \ker(Df(X)).$$

Par (2.6), on a

$$T_{I_n} SO(n) = T_{I_n} O(n) = \ker Df(I_n).$$

Enfin,

$$T_{I_n} SO(n) = \ker(Df(I_n)) = \{H \in \mathcal{M}_n(\mathbb{R}) \mid H^\top = -H\}$$

Par définition, l'algèbre de Lie de  $SO(n)$  est l'espace tangent en l'identité.

Ainsi, l'algèbre de Lie du groupe des rotations de dimension  $n$  est l'ensemble des matrices antisymétriques.

$$\mathfrak{so}(n) = \{H \in \mathcal{M}_n(\mathbb{R}) \mid H^\top = -H\}$$

□

**Définition 2.20** (Crochet de Lie dans  $\mathfrak{so}(n)$ ).

Le commutateur de  $\mathfrak{so}(n)$  est, pour  $A, B \in \mathfrak{so}(n)$  :

$$\begin{aligned} ([\cdot, \cdot]) : \mathfrak{g} \times \mathfrak{g} &\rightarrow \mathfrak{g} \\ (A, B) &\mapsto AB - BA \end{aligned}$$

**Proposition 2.21.** Si  $A, B \in \mathfrak{so}(n)$ , Alors  $[A, B] \in \mathfrak{so}(n)$

*Preuve.* Soient  $A, B \in \mathfrak{so}(n)$ , calculons la transposée de  $[A, B]$

$$\begin{aligned} [A, B]^\top &= (AB - BA)^\top \\ &= (AB)^\top - (BA)^\top \\ &= B^\top A^\top - A^\top B^\top. \end{aligned}$$

On utilise l'antisymétrie de  $A$  et  $B$

$$\begin{aligned} [A, B]^\top &= (-B)(-A) - (-A)(-B) \\ &= -(AB - BA) = -[A, B] \end{aligned}$$

On a bien  $[A, B]$  antisymétrique, donc  $[A, B] \in \mathfrak{so}(n)$ . □

---

3. En effet, on peut ignorer la condition sur le déterminant car on a déjà montré que les algèbres de Lie de  $SO(n)$  et  $O(n)$  coïncident (2.18)

**Définition 2.22** (Translation à droite).

Soit  $g \in G$ . On note **translation à droite** l'application :

$$\begin{aligned} R_g: G &\longrightarrow G \\ h &\longmapsto hg \end{aligned}$$

**Proposition 2.23.** *La translation à droite sur le groupe  $SO(n)$  est une application lisse.*

*Preuve.* Soit  $R_M$  une translation à droite sur  $SO(n)$ . On peut définir cette translation comme :

$$\begin{aligned} R_M: SO(n) &\longrightarrow SO(n) \\ X &\longmapsto u \times v = XM \end{aligned}$$

Par produit lisse (car c'est une application polynomiale en les coefficients), on a que  $R_M$  est une application lisse sur  $SO(n)$ .  $\square$

**Proposition 2.24.** *L'espace  $T_M SO(n)$  est isomorphe à  $T_{I_n} SO(n)$  pour tout  $M \in SO(n)$*

*Preuve.* Soit  $M \in SO(n)$ . On considère la translation à droite (2.22) sur  $SO(n)$  définie par

$$\begin{aligned} R_M: SO(n) &\longrightarrow SO(n) \\ X &\longmapsto XM \end{aligned}$$

Cette application est lisse (2.23), et son inverse  $R_{M^{-1}}$  est lisse. Ainsi,  $R_M$  est un difféomorphisme de  $SO(n)$ .

En particulier, sa différentielle en tout point est un isomorphisme linéaire. En évaluant en l'identité, on obtient

$$(dR_M)_{I_n}: T_{I_n} SO(n) \longrightarrow T_M SO(n),$$

qui est un isomorphisme. Son inverse est donné par

$$(dR_{M^{-1}})_M: T_M SO(n) \longrightarrow T_{I_n} SO(n),$$

et par théorème de dérivation des fonctions composées, on a

$$\begin{aligned} (dR_{M^{-1}})_M \circ (dR_M)_{I_n} &= d(R_{M^{-1}} \circ R_M)_{I_n} \\ &= d(\text{Id}_{SO(n)})_{I_n} \\ &= \text{Id}_{T_{I_n} SO(n)} \end{aligned}$$

On en déduit que  $T_M SO(n)$  et  $T_{I_n} SO(n)$  sont isomorphes.  $\square$

*Remarque 2.25.* La translation à droite permet de “passer”<sup>4</sup> d'un espace tangent à un autre, reliant l'espace tangent en un point quelconque et l'espace tangent à l'identité (2.24).

**Exemple 2.26.** *La figure ci-dessous **Figure (2.2)** illustre ce principe sur la surface  $z = x^2 - y^2$  (qui joue ici le rôle d'une variété différentielle). Le long de la courbe  $y = 1$ , on a représenté les plans tangents  $T_{g_0} G$  et  $T_{g_t} G$  en  $g_0 = (-2, 1, 3)$  et  $g_t = (0, 1, -1)$ , ainsi que les vecteurs tangents correspondants. Le passage de l'un à l'autre peut être réalisé par une translation à droite convenable.*

---

4. Passer : transporter de manière isomorphe les structures

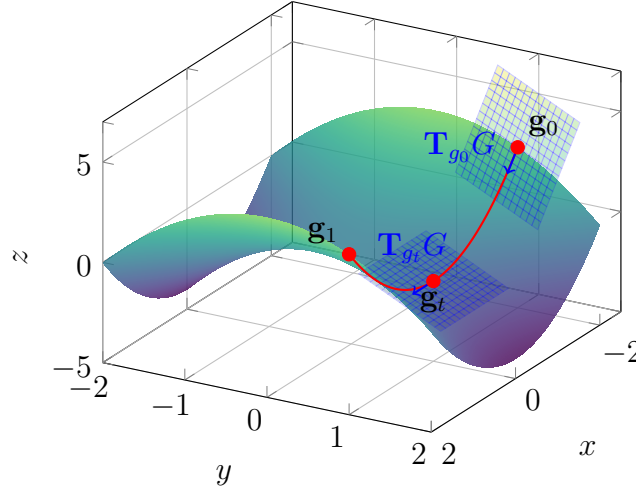


FIGURE 2.2 – Figure des plans tangents à une surface en  $g_0$  et  $g_t$ .

## 2.3 L'application exponentielle

L'application exponentielle joue un rôle fondamental dans l'étude des groupes de Lie car elle établit un lien entre l'algèbre de Lie  $\mathfrak{g} = T_e G$  et le groupe de Lie  $G$ .

**Définition 2.27** (Exponentielle de matrice).

Soit  $A \in \mathcal{M}_n(\mathbb{K})$  avec  $\mathbb{K} = \mathbb{R}$  ou  $\mathbb{C}$ . On appelle exponentielle de la matrice  $A$ , notée  $e^A$ , la somme de la série :

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}, \quad \exp(0) = I_n.$$

**Proposition 2.28.** Si  $A \in \mathfrak{so}(n)$ , Alors  $e^A \in SO(n)$ .

*Preuve.* Montrons que  $\det(e^A) = 1$  et  $(e^A)^\top e^A = I_n$ , et donc que  $e^A \in SO(n)$ .

Soit  $A \in \mathfrak{so}(n)$ ,  $A$  est une matrice antisymétrique.

On sait que pour tout  $M \in \mathcal{M}_n(\mathbb{R})$

$$\det(e^M) = e^{\text{Tr}(M)}.$$

Par antisymétrie de  $A$ ,  $\text{Tr}(A) = 0$ .

Et donc

$$\det(e^A) = e^{\text{Tr}(A)} = e^0 = 1.$$

Maintenant, montrons que  $(e^A)^\top e^A = I_n$ .

Pour la même matrice  $A$ ,

$$\begin{aligned} (e^A)^\top &= \left( \sum_{k=0}^{\infty} \frac{A^k}{k!} \right)^\top = \sum_{k=0}^{\infty} \frac{(A^k)^\top}{k!} \\ &= \sum_{k=0}^{\infty} \frac{(A^\top)^k}{k!} = e^{(A^\top)}. \end{aligned}$$

Or  $A^\top = -A$ ,

$$(e^A)^\top (e^A) = (e^{-A})(e^A) = e^0 = I_n.$$

Ainsi,  $e^A$  vérifie  $\det(e^A) = 1$  et  $(e^A)^\top e^A = I_n$ .

On a bien  $e^A \in SO(n)$ . □

Nous avons vu que, étant donné un vecteur  $A \in \mathfrak{g}$ , l'exponentielle permet de construire un élément  $\exp(A) \in G$  (au moins dans le cas où  $G = SO(n)$ ). L'exponentielle permet donc de passer d'une information infinitésimale, contenue dans l'espace tangent à l'identité, à une transformation appartenant au groupe. Dans le cas des rotations, elle associe à une matrice antisymétrique  $A \in \mathfrak{so}(n)$  une matrice de rotation  $e^A \in SO(n)$ . Intéressons-nous au cas où  $n = 2$ .

**Exemple 2.29.** *Exemple d'exponentielle d'une matrice  $2 \times 2$  antisymétrique :*

Soient  $\theta \in \mathbb{R}$  et  $A = \begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix} \in \mathfrak{so}(2)$ . Calculons  $e^A$  :

On observe que

$$A^2 = \begin{pmatrix} -\theta^2 & 0 \\ 0 & -\theta^2 \end{pmatrix} = -\theta^2 I_2,$$

et par récurrence,

$$A^{2k} = (-1)^k \theta^{2k} I_2, \quad A^{2k+1} = (-1)^k \theta^{2k+1} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Alors

$$\begin{aligned} e^A &= \sum_{k=0}^{\infty} \frac{A^k}{k!} = \sum_{k=0}^{\infty} \frac{A^{2k}}{(2k)!} + \sum_{k=0}^{\infty} \frac{A^{2k+1}}{(2k+1)!} \\ &= \left( \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k}}{(2k)!} \right) I_2 + \left( \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k+1}}{(2k+1)!} \right) \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \end{aligned}$$

On reconnaît les séries du cosinus et du sinus :

$$\cos \theta = \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k}}{(2k)!}, \quad \sin \theta = \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k+1}}{(2k+1)!}.$$

Ainsi,

$$e^A = \cos \theta I_2 + \sin \theta \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix},$$

qui est bien une matrice de rotation, donc  $e^A \in SO(2)$ .

**Proposition 2.30.** *L'application exponentielle  $\exp : \mathfrak{so}(n) \longrightarrow SO(n)$  est surjective.*

*Preuve.* Soit  $R \in SO(n)$ . Par le théorème spectral, il existe une base orthonormée dans laquelle la matrice de  $R$  est diagonale par blocs, les blocs diagonaux étant soit des blocs de rotation :

$$R(\theta_k) = \begin{pmatrix} \cos \theta_k & -\sin \theta_k \\ \sin \theta_k & \cos \theta_k \end{pmatrix},$$

soit des blocs  $(1)$  et  $(-1)$ .

Comme  $\det R = 1$ , les valeurs propres  $(-1)$  apparaissent en nombre pair ; on les regroupe donc deux à deux pour former des blocs

$$-I_2 = R(\pi).$$

On obtient ainsi une décomposition

$$R = P \operatorname{diag}(R(\theta_1), \dots, R(\theta_r), 1, \dots, 1) P^{-1},$$

où  $P$  est orthogonale.

Pour chaque bloc de rotation d'angle  $\theta_k$ , on construit explicitement

$$A_k = \begin{pmatrix} 0 & -\theta_k \\ \theta_k & 0 \end{pmatrix} \in \mathfrak{so}(2),$$

de sorte que

$$e^{A_k} = R(\theta_k).$$

On pose alors

$$A = P \operatorname{diag}(A_1, \dots, A_r, 0, \dots, 0) P^{-1}.$$

Comme  $\mathfrak{so}(n)$  est stable par conjugaison orthogonale, on a  $A \in \mathfrak{so}(n)$ .

Finalement,

$$e^A = P \operatorname{diag}(e^{A_1}, \dots, e^{A_r}, 1, \dots, 1) P^{-1} = R.$$

Ainsi, on retrouve que  $e^A = R$ , et donc que pour tout  $R \in SO(n)$ , il existe  $A \in \mathfrak{so}(n)$  tel que  $e^A = R$ . L'exponentielle est surjective de l'algèbre de Lie de  $SO(n)$  dans son groupe de Lie.  $\square$

# 3 Étude de l'équation différentielle linéaire $\dot{x}(t) = Ax(t)$

Dans ce chapitre, nous verrons comment cette équation différentielle linéaire apparaît naturellement par l'action du groupe spécial orthogonal  $SO(n)$  sur des landmarks de  $\mathbb{R}^n$ , l'importance de ce résultat, ainsi que sa résolution. Commençons avec le groupe de Lie  $SO(n)$ .

## 3.1 Obtenir l'équation différentielle $\dot{x}(t) = Ax(t)$

Considérons des landmarks  $x_i \in \mathbb{R}^n$  et l'action du groupe  $SO(n)$  sur  $\mathbb{R}^n$ . Remarquons une égalité que l'on utilisera juste après.

*Remarque 3.1.* On sait que  $\forall X \in SO(n): X^\top X = I_n$ , et donc  $X^{-1} = X^\top$ .

**Trouvons l'équation.**

Considérons une courbe différentiable à valeurs matricielles  $M_t \in SO(n)$  et définissons la déformation d'un landmark  $x_i$  par

$$x_i(t) = M_t x_i.$$

Différencions cette expression par rapport au temps,

$$\dot{x}_i(t) = \dot{M}_t x_i.$$

En posant  $A(t) = \dot{M}_t M_t^{-1}$ , on a  $\dot{M}_t = A(t) M_t$ . Montrons que  $A(t)$  est antisymétrique : en dérivant l'identité (3.1)  $M_t^\top M_t = I_n$  par rapport au temps,

$$\dot{M}_t^\top M_t + M_t^\top \dot{M}_t = 0.$$

Comme  $M_t^{-1} = M_t^\top$ , on a  $A(t) = \dot{M}_t M_t^\top$ , et l'égalité ci-dessus donne  $A(t)^\top + A(t) = 0$ . Ainsi  $A(t) \in T_{I_n} SO(n) = \mathfrak{so}(n)$  (l'algèbre de Lie de  $SO(n)$ ).

On obtient alors

$$\begin{aligned} \dot{x}_i(t) &= A(t) M_t x_i \\ \dot{x}_i(t) &= A(t) x_i(t). \end{aligned}$$

Donc si on suppose que ce vecteur est constant en fonction du temps  $A(t) = A$ , on retrouve

$$\dot{x}_i(t) = Ax_i(t).$$

*Remarque 3.2.* Cette construction peut également s'interpréter à l'aide de la translation à droite sur  $SO(n)$ . Pour  $M \in SO(n)$  :

$$\begin{aligned} R_M: SO(n) &\longrightarrow SO(n) \\ X &\longmapsto XM \end{aligned}$$

Sa différentielle en  $X \in SO(n)$  est

$$\begin{aligned} D(R_M)_X: T_X SO(n) &\longrightarrow T_{XM} SO(n) \\ H &\longmapsto HM. \end{aligned}$$

En effet, pour  $H \in \mathcal{M}_n(\mathbb{R})$ ,

$$\begin{aligned} R_M(X + H) &= (X + H)M \\ &= XM + HM \\ &= R_M(X) + DR_M(X)H + o(\|H\|). \end{aligned}$$

Ainsi

$$DR_M: SO(n) \longrightarrow \mathcal{L}(T_X SO(n), T_{XM} SO(n))$$

est à valeurs dans les applications linéaires d'un espace tangent dans un autre, qui sont isomorphes (2.24).

*Remarque 3.3.* La solution de l'équation différentielle linéaire ci-dessus permet, à partir d'un seul élément  $A$  de l'algèbre de Lie  $\mathfrak{so}(n)$ , de créer un mouvement sur tout le groupe  $SO(n)$  et de déformer les landmarks par l'action de ce mouvement.

## 3.2 Solution de $\dot{x}(t) = Ax(t)$

Pour garantir l'existence et l'unicité des trajectoires sur l'espace de forme, revenons au cas général avec le théorème suivant. Soit  $I$  un intervalle compact.

**Théorème 3.4** (Théorème de Cauchy-Lipschitz).

*Soient  $A: I \rightarrow \mathcal{M}_n(\mathbb{R})$  et  $B: I \rightarrow \mathbb{R}^n$  deux fonctions continues.*

*Alors  $\forall t_0 \in I$ , le problème de Cauchy*

$$\begin{cases} Y' = A(t)Y + B(t) \\ Y(t_0) = y_0 \end{cases}$$

*admet une unique solution définie sur  $I$  tout entier.*

*Preuve.* Admis □

**Proposition 3.5.** *La solution de l'équation différentielle linéaire  $\dot{x}(t) = Ax(t)$  est :*

$$x(t) = e^{tA}x_0$$

*Preuve.* Dans notre cas, en associant  $\mathbb{R}^d$  à  $\mathcal{M}_n(\mathbb{R})$ , l'application  $F(t, X) = AX$  est linéaire par rapport à  $X$ , donc globalement lipschitzienne.

Le *Théorème de Cauchy-Lipschitz 3.4* s'applique et assure l'existence d'une unique solution maximale définie sur  $\mathbb{R}$  tout entier, dont la forme explicite fait intervenir l'exponentielle de matrice.

Vérifions que notre unique solution est  $x(t) = e^{tA}x_0$  :  
 $x$  étant lisse, on peut écrire :

$$\begin{aligned}x(t) &= e^{tA}x_0 \\ \dot{x}(t) &= (e^{tA})x_0\end{aligned}$$

On dérive en fonction de  $t$ .

$$\begin{aligned}&= Ae^{tA}x_0 \\ \dot{x}(t) &= Ax(t)\end{aligned}$$

Avec

$$x(0) = I_n x_0 = x_0$$

Ainsi, l'unique solution de l'équation différentielle est  $x(t) = e^{tA}x_0$ .

□

## 4 Formulation Hamiltonienne du recalage sur $SO(n)$

Avant de nous restreindre aux cas explicites  $SO(2)$  et  $SO(3)$ , reprenons le cadre hamiltonien introduit au **Chapitre 1**. Étudions le cas où le groupe de transformations est  $SO(n)$  et que la forme étudiée est un nuage de points.

### 4.1 Espaces tangent et cotangent des nuages de points

Dans ce cadre, on distingue le champ de vecteurs  $v_t$  du flot de difféomorphismes  $\varphi_t$  qu'il engendre. En effet,  $v_t \in C_0^2(\mathbb{R}^n, \mathbb{R}^n)$  est le champ de vecteurs dépendant du temps. Dans le cadre général, on considère. On travaille donc dans l'espace

$$V = Skew_n = \mathfrak{so}(n),$$

où  $Skew_n$  désigne l'espace des matrices antisymétriques de taille  $n \geq 2$ . La forme étudiée est un ensemble de  $N$  landmarks. L'espace des formes est

$$Q = (\mathbb{R}^n)^N,$$

qui est un espace vectoriel de dimension finie. Un espace vectoriel étant son propre tangent en chacun de ses points, on a donc

$$T_q Q = (\mathbb{R}^n)^N = Q \quad \text{pour tout } q \in Q.$$

Pour définir l'espace cotangent  $T_q^* Q$ , on cherche l'ensemble des formes linéaires sur  $T_q Q$ , c'est-à-dire les applications qui prennent un élément de  $T_q Q = (\mathbb{R}^n)^N$  et renvoient un scalaire dans  $\mathbb{R}$ .

Soit  $p \in T_q^* Q = (\mathbb{R}^n)^N$  et  $\delta q \in T_q Q = (\mathbb{R}^n)^N$ . On cherche une application telle que  $p(\delta q) = (p \mid \delta q) \in \mathbb{R}$ . On pose  $Q = (\mathbb{R}^{nN}, \langle \cdot, \cdot \rangle)$  muni de son produit scalaire canonique, et donc on obtient une application bilinéaire de  $T_q^* Q$  dans les réels qui convient :

$$(p \mid \delta q) = \langle p, \delta q \rangle = p^\top \delta q.$$

*Remarque 4.1.* Cette identification de  $T_q^* Q$  avec  $T_q Q$  est spécifique au cas des nuages de points.

## 4.2 Hamiltonien sur $SO(n)$

Reprenons le Hamiltonien introduit au **Chapitre 1**. Le contrôle est ici la matrice antisymétrique  $A \in V = Skew_n$ , et l'action du champ de vecteurs sur la forme est donnée par la multiplication matricielle  $A \cdot q = Aq$ . Le Hamiltonien s'écrit

$$H: T^*Q \times V \longrightarrow \mathbb{R}$$

$$(q, p, A) \longmapsto (p \mid Aq) - \frac{1}{2}\|A\|^2.$$

En utilisant l'expression vue juste au-dessus et la norme de Frobenius  $\|A\|_F^2 = \text{tr}(A^\top A)$ , on obtient :

$$H(q, p, A) = (p \mid Aq) - \frac{1}{2}\|A\|^2$$

$$= p^\top(Aq) - \frac{1}{2}\text{tr}(A^\top A)$$

Le premier terme mesure l'interaction entre le moment  $p$  et la vitesse de déformation  $Aq$ , tandis que le second représente le coût énergétique du champ de vecteurs.

Le problème variationnel s'écrit alors, avec  $q_S$  la forme source et  $q_T$  la forme cible :

$$\min_{A \in L^2([0,1], V)} \frac{1}{2} \int_0^1 \|A_t\|_V^2 dt + \|q_1 - q_T\|$$

$$\text{tel que } \begin{cases} \dot{q}_t = A_t q_t \\ q_0 = q_S \end{cases}$$

## 4.3 Équations géodésiques

Le principe du maximum de Pontryagin [10] impose que le contrôle optimal maximise le Hamiltonien par rapport à  $A$ . Les trajectoires optimales sont alors solutions des équations hamiltoniennes

$$\dot{q}_t = \partial_p H(q_t, p_t, A_t),$$

$$\dot{p}_t = -\partial_q H(q_t, p_t, A_t),$$

$$0 = \partial_A H(q, p, A).$$

La condition pour optimiser  $A$  et obtenir le Hamiltonien réduit se trouve en annulant la dérivée du Hamiltonien par rapport à  $A$ . En effet,  $H$  étant strictement concave en  $v$ , l'annulation de la dérivée caractérise le maximum.

### Calculons cette différentielle

Le premier terme  $p^\top(Aq)$  est linéaire en  $A$ , il s'écrit

$$p^\top(Aq) = \text{Tr}(p^\top Aq) = \text{Tr}(qp^\top A),$$

d'où,

$$\nabla(p^\top(Aq)) = pq^\top.$$

Le second terme :  $-\frac{1}{2}\text{Tr}(A^\top A)$  a pour différentielle  $-A$ . En effet, soit

$$f(A) = -\frac{1}{2} \text{Tr}(A^\top A).$$

Pour toute variation  $H$ ,

$$\begin{aligned} f(A+H) &= -\frac{1}{2} \operatorname{Tr}((A+H)^\top (A+H)) \\ &= f(A) - \frac{1}{2} \operatorname{Tr}(A^\top H + H^\top A) + o(\|H\|) \\ &= f(A) - \operatorname{Tr}(A^\top H) + o(\|H\|), \end{aligned}$$

car  $\operatorname{Tr}(H^\top A) = \operatorname{Tr}(A^\top H)$ .

Ainsi,

$$Df(A)H = -\operatorname{Tr}(A^\top H) = \langle -A, H \rangle,$$

où  $\langle X, Y \rangle = \operatorname{Tr}(X^\top Y)$  est le produit scalaire de Frobenius. Par conséquent,

$$\nabla f(A) = -A.$$

On obtient donc

$$\nabla H(q, p, A) = pq^\top - A.$$

Par définition, le contrôle  $A$  est contraint d'appartenir à l'algèbre de Lie  $\mathfrak{so}(n) = \operatorname{Skew}_n$ . Les variations admissibles  $K$  appartiennent donc également à  $\operatorname{Skew}_n$ . La condition pour avoir le Hamiltonien stationnaire par rapport à  $A$  s'écrit alors

$$\langle pq^\top - A, K \rangle = 0, \quad \forall K \in \operatorname{Skew}_n.$$

c'est-à-dire

$$A^* = \operatorname{Proj}_{\operatorname{Skew}_n}(pq^\top).$$

Comme la projection orthogonale sur  $\operatorname{Skew}_n$  est donnée par

$$\operatorname{Proj}_{\operatorname{Skew}_n}(M) = \frac{1}{2}(M - M^\top),$$

on obtient

$$A^* = \frac{1}{2}(pq^\top - (pq^\top)^\top) = \frac{1}{2}(pq^\top - qp^\top).$$

donc  $A^* \in \operatorname{Skew}_n = \mathfrak{so}(n)$ .

On définit alors l'Hamiltonien réduit en remplaçant le contrôle optimal dans le Hamiltonien initial :

$$H_r(q, p) = H(q, p, A^*).$$

Comme le Hamiltonien réduit  $H_r$  ne dépend pas explicitement du temps, il est conservé le long des solutions des équations hamiltoniennes et le problème de contrôle optimal se ramène ainsi à un système d'équations différentielles sur  $T^*Q$ , avec pour seule condition la donnée du moment initial  $p_0$  :

$$\min_{p_0 \in T_{q_0}^*Q} \frac{1}{2} \int_0^1 \|A_t\|_V^2 dt + \|q_1 - q_T\|$$

tel que les équations hamiltoniennes sont satisfaites.

*Remarque 4.2.* Sur le groupe des difféomorphismes, on aurait  $G = \operatorname{Diff}(\mathbb{R}^n)$  et  $\mathfrak{g} = C_0^k(\mathbb{R}^n, \mathbb{R}^n)$ , un espace de dimension infinie. Ici, dans le cas des nuages de points, on a  $p \in T^*Q$  avec  $\dim(T^*Q) = \dim(Q) = nN$  fini. L'optimisation ne porte plus sur le champ de vitesse en dimension infinie, mais uniquement sur le moment initial  $p_0$ , un objet de dimension finie.

## 5 Cas particuliers de $SO(2)$ et $SO(3)$

Dans ce chapitre, nous sortirons un peu de la théorie en nous intéressant à des cas simples : les groupes de rotations  $SO(2)$  et  $SO(3)$ .

Ces deux groupes vont nous donner un aperçu plus concret du cadre *LDDMM*.

### 5.1 $SO(2)$ , Les rotations en dimension 2

Le groupe  $SO(2)$  est très simple, il décrit les rotations du plan autour de l'origine en dimension 2.

**Proposition 5.1.** *Soit  $R$  une matrice de  $SO(2)$ , alors il existe  $\theta \in \mathbb{R}$  tel que*

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

*Preuve.* Soit

$$R = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SO(2).$$

Comme  $R^\top R = I_2$ , les colonnes de  $R$  forment une base orthonormée de  $\mathbb{R}^2$ . On a donc

$$a^2 + c^2 = 1.$$

Il existe alors  $\theta \in \mathbb{R}$  tel que

$$a = \cos \theta, \quad c = \sin \theta.$$

La seconde colonne doit être un vecteur unitaire orthogonal à  $(a, c)^\top$ . Elle est égale à

$$(-\sin \theta, \cos \theta)^\top,$$

ou à

$$(\sin \theta, -\cos \theta)^\top.$$

Or  $\det(R) = 1$ , ce qui élimine la seconde possibilité. Ainsi

$$b = -\sin \theta, \quad d = \cos \theta.$$

Ainsi,

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

□

### Étudions l'équation $\dot{R}(t) = AR(t)$ dans $SO(2)$ .

L'inconnue est une rotation que l'on va nommer ici  $R$ . On remarque que  $A \in \mathfrak{so}(2)$  pour que la solution appartienne à  $SO(2)$ .

Alors on peut écrire  $A$  sous la forme

$$A = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}, \text{ avec } \omega \in \mathbb{R}.$$

On a vu que la solution d'une équation différentielle linéaire de ce type est  $e^{tA}x_0$  (3.5). Dans notre cas, on a donc

$$R(t) = e^{tA}R_0.$$

Or l'exponentielle d'une matrice de  $\mathfrak{so}(n)$  est à valeur dans  $SO(n)$  (2.28).

Sachant que  $tA \in \mathfrak{so}(2)$ , donc  $e^{tA} \in SO(2)$  et  $e^{tA}$  peut s'écrire (5.1) comme

$$e^{tA} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \text{ avec } \theta \in \mathbb{R} \text{ dépendant de } t.$$

Posons

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

Alors  $A = \omega J$  et  $J^2 = -I_2$ .

Par définition de l'exponentielle matricielle (2.27),

$$e^{tA} = \sum_{k=0}^{+\infty} \frac{(tA)^k}{k!} = \sum_{k=0}^{+\infty} \frac{(\omega t)^k J^k}{k!}.$$

Comme

$$J^{2k} = (-1)^k I_2 \quad \text{et} \quad J^{2k+1} = (-1)^k J,$$

on sépare les puissances paires et impaires :

$$\begin{aligned} e^{tA} &= \sum_{k=0}^{+\infty} \frac{(-1)^k (\omega t)^{2k}}{(2k)!} I_2 + \sum_{k=0}^{+\infty} \frac{(-1)^k (\omega t)^{2k+1}}{(2k+1)!} J \\ &= \cos(\omega t) I_2 + \sin(\omega t) J. \end{aligned}$$

Après résolution, on trouve

$$e^{tA} = \begin{pmatrix} \cos(\omega t) & -\sin(\omega t) \\ \sin(\omega t) & \cos(\omega t) \end{pmatrix}.$$

Par conséquent,

$$R(t) = \begin{pmatrix} \cos(\omega t) & -\sin(\omega t) \\ \sin(\omega t) & \cos(\omega t) \end{pmatrix} R_0.$$

Comme toute matrice de  $SO(2)$  est une rotation, il existe  $\theta_0 \in \mathbb{R}$  tel que

$$R_0 = \begin{pmatrix} \cos \theta_0 & -\sin \theta_0 \\ \sin \theta_0 & \cos \theta_0 \end{pmatrix}.$$

On obtient alors

$$R(t) = \begin{pmatrix} \cos(\omega t) & -\sin(\omega t) \\ \sin(\omega t) & \cos(\omega t) \end{pmatrix} \begin{pmatrix} \cos(\theta_0) & -\sin(\theta_0) \\ \sin(\theta_0) & \cos(\theta_0) \end{pmatrix}.$$

En effectuant le produit matriciel et en utilisant les formules d'addition du sinus et du cosinus, on trouve

$$R(t) = \begin{pmatrix} \cos(\theta_0 + \omega t) & -\sin(\theta_0 + \omega t) \\ \sin(\theta_0 + \omega t) & \cos(\theta_0 + \omega t) \end{pmatrix} \cdot (*)$$

Ainsi, la solution est une rotation dont l'angle évolue linéairement avec le temps si  $\theta_0 = 0$ .

### Application au modèle de recalage.

Soit  $x_S$  notre forme source et  $x_T$  notre forme cible. Rappelons notre équation de minimisation dans le cas général [7] pour  $q_S$  la forme source et  $q_T$  la forme cible :

$$\begin{aligned} \min_{v \in L^2([0,1], V)} J(v) &= \frac{1}{2} \int_0^1 \|v_t\|_V^2 dt + \|\varphi_1 \cdot q_S - q_T\| \\ \text{tel que } \begin{cases} \dot{\varphi}_t &= v_t \circ \varphi_t \\ \varphi_0 &= Id \end{cases} \end{aligned}$$

Dans  $SO(2)$ , l'espace des transformations est réduit aux rotations du plan. On cherche la meilleure rotation, où  $x_1$  désigne la forme transformée, vérifiant

$$\begin{cases} \dot{x}(t) = A_t x(t) \\ x_0 = x_S \end{cases}$$

Alors, on résout l'équation différentielle ci-dessus en utilisant la solution (\*)

$$x(t) = R(t)x_S.$$

On obtient ainsi une description explicite de l'évolution de la forme source sous l'action d'une rotation. Le résultat  $x_1$  apparaît quand la rotation est totalement appliquée à notre forme source.

Le champ de vecteurs associé à la rotation est donné par

$$v_t(x) = A_t x, \quad A_t \in \mathfrak{so}(2).$$

En identifiant un champ de vecteurs linéaire à sa représentation matricielle, on notera par la suite

$$v_t = A_t, \quad A_t \in \mathfrak{so}(2).$$

Ainsi, notre minimisation ressemble maintenant à ça (où  $Skew_2$  désigne l'espace des matrices antisymétriques) :

$$\begin{aligned} \min_{A \in L^2([0,1], Skew_2)} \frac{1}{2} \int_0^1 \|A_t\|_{Skew_2}^2 dt + \|x_1 - x_T\| \\ \text{tel que } \begin{cases} \dot{x}(t) = A_t x(t) \\ x_0 = x_S \end{cases} \end{aligned}$$

Or  $A$  est antisymétrique, donc sous la forme  $A_t = \begin{pmatrix} 0 & -\omega_t \\ \omega_t & 0 \end{pmatrix}$  pour un certain  $\omega_t \in \mathbb{R}$ .

On peut simplifier notre minimisation, et utiliser la valeur absolue au lieu d'une norme qui convient :

$$\begin{aligned} \min_{\omega_t \in \mathbb{R}} \frac{1}{2} \int_0^1 |\omega_t|^2 dt + \|x_1 - x_T\| \\ \text{tel que } \begin{cases} \dot{x}(t) = A_t x(t) \\ x_0 = x_S \end{cases} \end{aligned}$$

Si l'on cherche une géodésique<sup>1</sup>, on pose  $\omega_t = \omega$  pour tout  $t$ . Le problème se réduit à l'optimisation d'un seul scalaire paramétrique  $\omega \in \mathbb{R}$  :

$$\begin{aligned} \min_{\omega \in \mathbb{R}} J(\omega) &= \frac{1}{2}|\omega|^2 + \|x_1(\omega) - x_T\| \\ \text{tel que } &\begin{cases} \dot{x}(t) = A_t x(t) \\ x_0 = x_S \end{cases} \end{aligned}$$

Autrement dit, dans le cas de  $SO(2)$ , le problème de recalage consiste à rechercher une vitesse d'angle<sup>2</sup> optimale permettant de faire coïncider au mieux la forme source avec la forme cible tout en "optimisant"<sup>3</sup> la rotation.

## 5.2 $SO(3)$ , Les rotations en dimension 3

Le cas de la dimension 3 introduit une géométrie supérieure à celle de la dimension 2, notamment en raison de la *non-commutativité* du groupe  $SO(3)$ .

**Proposition 5.2.** *Soit  $R$  une matrice de  $SO(3)$ , alors cette matrice est semblable (par conjugaison orthogonale) à une matrice de cette forme :*

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}, \quad \text{avec } \theta \in \mathbb{R}$$

*c'est-à-dire, il existe une matrice  $P$  orthogonale et un réel  $\theta$  vérifiant*

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

*Preuve.* Soit  $R \in SO(3)$  et soit  $u$  l'endomorphisme orthogonal associé. Considérons le polynôme caractéristique

$$\chi(t) = \det(R - tI_3).$$

C'est un polynôme de degré 3, donc il admet au moins une racine réelle  $\lambda$ . Comme  $R$  est une matrice orthogonale, toutes ses valeurs propres sont de module 1, on obtient donc nécessairement  $\lambda \in \{-1, 1\}$ .

Montrons que 1 est nécessairement une valeur propre de  $R$  :

Supposons, par l'absurde, que 1 ne soit pas valeur propre. La valeur propre réelle  $\lambda$  est alors égale à  $-1$ . Soit  $e_1$  un vecteur propre unitaire associé à  $\lambda = -1$ . Puisque  $u$  est orthogonal, le sous-espace  $H = e_1^\perp$  est stable par  $u$ .

Dans une base orthonormale adaptée à la décomposition  $\mathbb{R}^3 = \mathbb{R}e_1 \oplus H$ , la matrice de  $u$  est de la forme

$$\begin{pmatrix} -1 & 0 \\ 0 & A \end{pmatrix}, \quad \text{avec } A \in O(2).$$

---

1. La rotation "la plus naturelle", sans accélération, à vitesse angulaire constante

2. Vitesse d'angle ou vitesse angulaire : un angle qui évolue en fonction du temps

3. Optimiser : minimiser le coût énergétique

Comme  $\det(R) = 1$ , on obtient

$$1 = \det(R) = -\det(A),$$

donc  $\det(A) = -1$ . Or toute matrice de  $O(2)$  de déterminant  $-1$  admet  $1$  comme valeur propre. Ainsi,  $u$  admet également  $1$  comme valeur propre, ce qui est contradictoire.

Soit  $e_1$  un vecteur unitaire associé à la valeur propre  $1$ . Le sous-espace

$$H = e_1^\perp$$

est stable par  $u$ . En choisissant une base orthonormale  $(e_2, e_3)$  de  $H$ , la base

$$\mathcal{B} = (e_1, e_2, e_3)$$

est orthonormale et

$$\text{Mat}_{\mathcal{B}}(u) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & a & b \\ 0 & c & d \end{pmatrix},$$

où

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in O(2).$$

Comme

$$\det(A) = \det(R) = 1,$$

on a en réalité  $A \in SO(2)$ . Or, par (5.1), toute matrice de  $SO(2)$  est de la forme

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \quad \theta \in \mathbb{R}.$$

Ainsi,

$$\text{Mat}_{\mathcal{B}}(u) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}.$$

Si  $P$  désigne la matrice de passage de la base canonique à la base orthonormale  $\mathcal{B}$ , alors  $P$  est orthogonale et on a bien

$$R = P \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} P^{-1}.$$

□

**Équation  $\dot{R}(t) = AR(t)$  dans  $SO(3)$ .**

L'inconnue est une rotation que l'on va nommer ici  $R$ . On remarque que  $A \in \mathfrak{so}(3)$  pour que la solution appartienne à  $SO(3)$ .

Alors on peut écrire  $A$  sous la forme

$$A = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \text{ avec } \omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3.$$

On a vu que la solution d'une équation différentielle linéaire de ce type est  $e^{tA}x_0$  (3.5). Dans notre cas, on a donc

$$R(t) = e^{tA}R_0.$$

Or l'exponentielle d'une matrice de  $\mathfrak{so}(n)$  est à valeur dans  $SO(n)$  (2.28).

Contrairement au cas de  $SO(2)$ , on n'utilisera pas une écriture simple de  $e^{tA}$ . On va plutôt calculer avec les séries entières.

Notons

$$\theta = \|\omega\| = \sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2},$$

la vitesse angulaire (la norme du vecteur  $\omega$ ), et, si  $\theta \neq 0$ ,  $n = \omega/\theta$  le vecteur unitaire correspondant à l'axe de rotation.

**Lemme 5.3.** *La matrice  $A$  vérifie*

$$A^3 = -\theta^2 A.$$

*Preuve.* Pour tout  $x \in \mathbb{R}^3$ , on a  $Ax = \omega \times x$ . Donc

$$A^2x = \omega \times (\omega \times x) = (\omega \cdot x)\omega - \|\omega\|^2x = (\omega \cdot x)\omega - \theta^2x.$$

Ainsi,  $A^2 = \omega\omega^\top - \theta^2 I_3$ . On en déduit

$$A^3 = A(\omega\omega^\top - \theta^2 I_3) = (A\omega)\omega^\top - \theta^2 A.$$

Or  $A\omega = \omega \times \omega = 0$ , d'où  $A^3 = -\theta^2 A$ . □

Cette relation (5.3) permet de calculer, par récurrence, toutes les puissances de  $A$  à partir de  $A$  et  $A^2$  seulement :

$$A^{2k+1} = (-1)^k \theta^{2k} A, \quad A^{2k+2} = (-1)^k \theta^{2k} A^2, \quad k \in \mathbb{N}.$$

On peut alors regrouper les termes pairs et impairs de la série exponentielle :

$$e^{tA} = \sum_{k=0}^{\infty} \frac{(tA)^k}{k!} = I_3 + \left( \sum_{k=0}^{\infty} \frac{(-1)^k (\theta t)^{2k+1}}{(2k+1)! \theta} \right) A + \left( \sum_{k=0}^{\infty} \frac{(-1)^k (\theta t)^{2k+2}}{(2k+2)! \theta^2} \right) A^2.$$

On reconnaît dans ces deux sommes les développements en série entière du sinus et du cosinus :

$$\sum_{k=0}^{\infty} \frac{(-1)^k (\theta t)^{2k+1}}{(2k+1)!} = \sin(\theta t), \quad \sum_{k=0}^{\infty} \frac{(-1)^k (\theta t)^{2k+2}}{(2k+2)!} = 1 - \cos(\theta t).$$

On obtient ainsi la formule de Rodrigues :

$$e^{tA} = I_3 + \frac{\sin(\theta t)}{\theta} A + \frac{1 - \cos(\theta t)}{\theta^2} A^2.$$

(Si  $\theta = 0$ , c'est-à-dire  $\omega = 0$ , on a simplement  $e^{tA} = I_3$  : l'absence de vitesse angulaire correspond à l'absence de rotation.)

Ainsi,

$$R(t) = \left( I_3 + \frac{\sin(\theta t)}{\theta} A + \frac{1 - \cos(\theta t)}{\theta^2} A^2 \right) R_0.$$

On retrouve que  $R(t)$  est la rotation d'angle  $\theta t$  autour de l'axe donné par  $n = \omega/\theta$ , appliquée à  $R_0$ . L'angle de rotation croît donc linéairement en  $t$ , exactement comme dans le cas de  $SO(2)$ .

Il faut remarquer une différence importante avec le cas de  $SO(2)$  : le groupe  $SO(3)$  n'est pas commutatif. En général, pour deux rotations  $R_1, R_2 \in SO(3)$ ,

$$R_1 R_2 \neq R_2 R_1.$$

L'ordre dans lequel on applique deux rotations de l'espace a donc une importance.

### Application au modèle de recalage.

Dans le cas des rotations en 3 dimensions, le problème se réduit aussi à l'optimisation d'un seul vecteur paramétrique  $\omega \in \mathbb{R}^3$  :

$$\begin{aligned} \min_{\omega \in \mathbb{R}^3} J(\omega) &= \frac{1}{2} \|\omega\|^2 + \|x_1(\omega) - x_T\| \\ \text{tel que } &\begin{cases} \dot{x}(t) = A_t x(t) \\ x_0 = x_S \end{cases} \end{aligned}$$

avec  $A \in \mathfrak{so}(3)$ .

Sauf que la quantité  $\omega$  contient alors deux informations :

- sa direction donne l'axe de rotation ;
- sa norme donne l'angle de rotation appliqué entre  $t = 0$  et  $t = 1$ .

Ainsi, dans le cas de  $SO(3)$ , le recalage consiste à chercher l'axe et l'angle de rotation qui permettent de rapprocher au mieux la forme source de la forme cible, tout en minimisant le coût énergétique associé à cette rotation.

Cependant, contrairement au cas de  $SO(2)$ , l'optimisation n'est plus scalaire. Elle porte maintenant sur un vecteur de  $\mathbb{R}^3$ , et la non-commutativité de  $SO(3)$  rend la géométrie du problème plus riche.

## 6 Implémentation algorithmique

Nous détaillons ici deux stratégies de résolution numérique du problème de recalage restreint aux rotations, ainsi que l'analyse des résultats expérimentaux obtenus sur différentes formes tests.

### 6.1 Approche directe - Discrétiser puis Optimiser

Pour résoudre ce problème numériquement, on utilise un algorithme itératif :

— **Initialisation** : On choisit une vitesse angulaire de départ, par exemple

$$\omega^{(0)} = 0.$$

— **Boucle d'optimisation** (pour chaque itération  $k$ ) :

1. *Intégration du flot* (méthode d'Euler explicite) : On discrétise le temps physique  $t \in [0, 1]$  en  $N$  sous-intervalles de pas  $\Delta t$ . En partant de la position initiale  $x_{t=0} = x_S$ , on intègre la dynamique pas à pas :

$$x_{t+\Delta t} = x_t + \Delta t A(\omega^{(k)}) x_t.$$

On obtient ainsi la position déformée finale  $x_{\text{finale}}$ , correspondant à l'instant  $t = 1$ .

2. *Évaluation de l'énergie* : On calcule le coût global

$$J(\omega^{(k)}).$$

3. *Mise à jour* (descente de gradient) : On calcule le gradient de l'énergie  $\nabla J(\omega^{(k)})$  et on actualise la vitesse angulaire avec un taux d'apprentissage  $\alpha$  :

$$\omega^{(k+1)} = \omega^{(k)} - \alpha \nabla J(\omega^{(k)}).$$

La descente de gradient peut être réalisée à l'aide de l'algorithme L-BFGS de PyTorch.

#### 6.1.1 Recalage rigide sur $SO(2)$

On souhaite toujours minimiser le problème suivant pour les landmarks :

$$\begin{aligned} \min_{v \in L^2([0,1], V)} J(v) &= \frac{1}{2} \int_0^1 \|v_t\|_V^2 dt + \|\varphi_1 \cdot q_S - q_T\| \\ \text{tel que } &\begin{cases} \dot{\varphi}_t = v_t \circ \varphi_t \\ \varphi_0 = Id \end{cases} \end{aligned}$$

On remplace le champ de vitesse LDDMM par un champ rigide

$$\dot{x} = A(\omega)x,$$

avec

$$A(\omega) = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \in \mathfrak{so}(2).$$

Le paramètre  $\omega \in \mathbb{R}$  est la vitesse angulaire.

On retrouve alors le problème présenté dans le chapitre **5 Cas particuliers de  $SO(2)$  et  $SO(3)$** , avec les landmarks,  $x_S$  notre forme source et  $x_T$  notre forme cible :

$$\begin{aligned} \min_{\omega \in \mathbb{R}} J(\omega) &= \frac{1}{2}|\omega|^2 + \|x_1(\omega) - x_T\| \\ \text{tel que } &\begin{cases} \dot{x} = A(\omega)x \\ x_{t=0} = x_S \end{cases} \end{aligned}$$

**Méthode pour le recalage rigide sur  $SO(2)$ .**

**Étape 1 : Intégration du flot**

La fonction `Euler` (A.1) intègre le flot rigide par la méthode d'Euler explicite sur  $t \in [0, 1]$ .

**Étape 2 : Évaluation de l'énergie**

La fonction `rotation_loss` (A.2) calcule l'énergie totale  $J(\omega) = \frac{1}{2}|\omega|^2 + \|x_1(\omega) - x_T\|$ .

**Étape 3 : Mise à jour (descente de gradient)**

La fonction `opti_rotation` (A.3) optimise la vitesse angulaire  $\omega$  par descente de gradient.

**Exemple 6.1** (Application de la méthode et calcul de la première itération).

*On initialise le paramètre de rotation à :*

$$\omega^{(0)} = 0$$

*Cette valeur correspond à une rotation nulle. On évalue alors la fonction coût associée au recalage :*

$$J(\omega^{(0)}) = J(0) = 9.488581$$

*Cette valeur mesure l'écart entre la forme de référence et la forme transformée lorsque aucune rotation n'est appliquée. La valeur ici provient de l'exemple **Figure (6.1)**.*

*Calcul du gradient*

*Le gradient de la fonction coût en  $\omega^{(0)}$  est :*

$$\nabla J(\omega^{(0)}) = 23.035329$$

*Ce gradient est positif, ce qui signifie que la fonction coût augmente lorsque  $\omega$  augmente localement. Pour diminuer la fonction coût, il faut donc se déplacer dans la direction opposée au gradient.*

*Application de la descente de gradient*

*La formule de mise à jour est :*

$$\omega^{(k+1)} = \omega^{(k)} - \alpha \nabla J(\omega^{(k)})$$

avec :

$$\alpha = 10^{-3}$$

En remplaçant par les valeurs numériques :

$$\omega^{(1)} = 0 - 10^{-3} \times 23.035329$$

On calcule d'abord le déplacement

$$10^{-3} \times 23.035329 = 0.023035329$$

Puis

$$\omega^{(1)} = 0 - 0.023035329$$

d'où

$$\boxed{\omega^{(1)} = -0.023035329}$$

Vérification de l'amélioration

Après cette mise à jour, on réévalue la fonction coût :

$$J(\omega^{(1)}) = 8.950083$$

Comme

$$8.950083 < 9.488581,$$

la fonction coût a diminué. La première étape de la descente de gradient améliore donc le recalage en rapprochant les deux formes.

Résumé de l'itération :

$$\omega^{(0)} = 0$$

$$\nabla J(\omega^{(0)}) = 23.035329$$

$$\Delta\omega = -\alpha \nabla J(\omega^{(0)}) = -0.023035329$$

$$\omega^{(1)} = -0.023035329$$

$$J(\omega^{(0)}) = 9.488581 \quad \rightarrow \quad J(\omega^{(1)}) = 8.950083$$

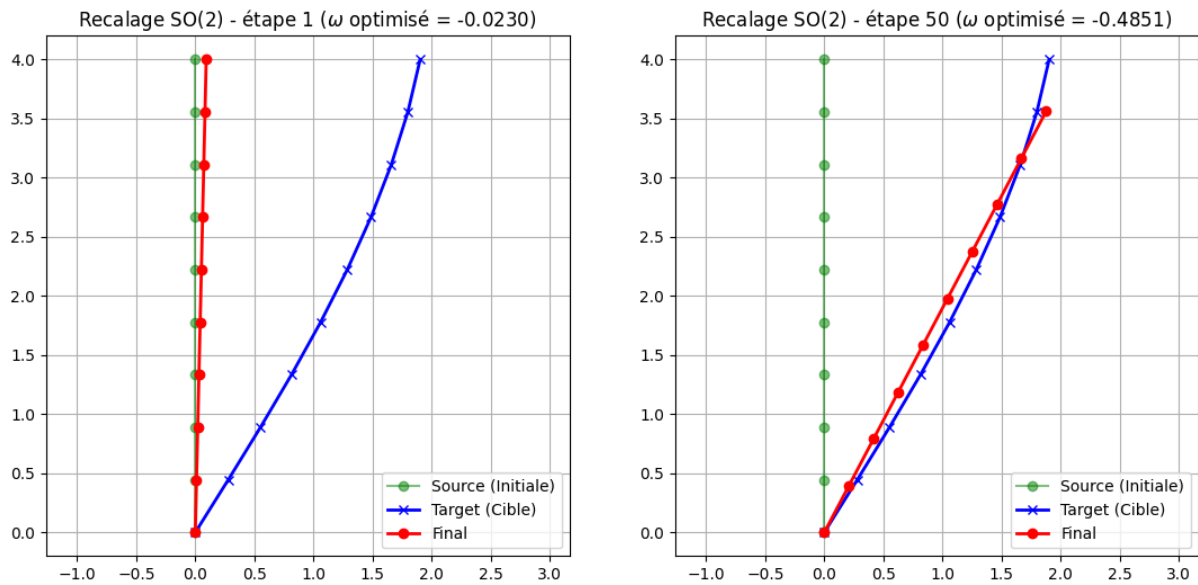


FIGURE 6.1 – Première et 50<sup>e</sup> étape du recalage sur les rotations  $SO(2)$ .

**Exemple 6.2** (Recalons Lecocq).

Voyons des exemples de recalage par rotation sur  $SO(2)$  avec des formes plus complexes.

Dans la **Figure (6.2)**, on applique une rotation à un coq  $x_S$  pour le faire “matcher” avec la forme cible  $x_T$  qui est sensiblement le même coq légèrement tourné. On voit que la figure source et cible se superposent assez bien. La géométrie du coq est préservée tout au long de la déformation, le cadre LDDMM restreint aux rotations est efficace lorsque la transformation est seulement une rotation.

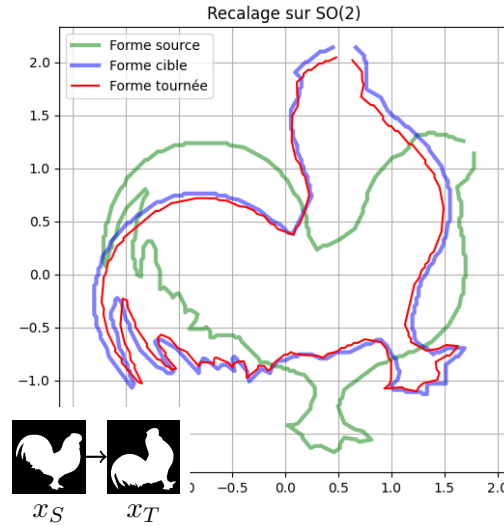


FIGURE 6.2 – Recalage sur les rotations  $SO(2)$  d'un coq.

La **Figure (6.3)** présente deux cas où la forme cible  $x_T$  diffère de la forme source  $x_S$  non seulement par une rotation, mais aussi par une déformation (des coqs légèrement différents séparés de  $x_S$  par une transformation non rigide). Dans ces situations, le modèle restreint à  $SO(2)$  montre ses limites : en cherchant uniquement une rotation optimale, l'algorithme ne peut pas capturer les déformations non rigides. La forme transformée  $x_1$  ne coïncide alors que partiellement voire même pas du tout avec la cible  $x_T$ , et le terme d'attache aux données  $\|x_1 - x_T\|$  reste élevé malgré la convergence de l'optimisation.

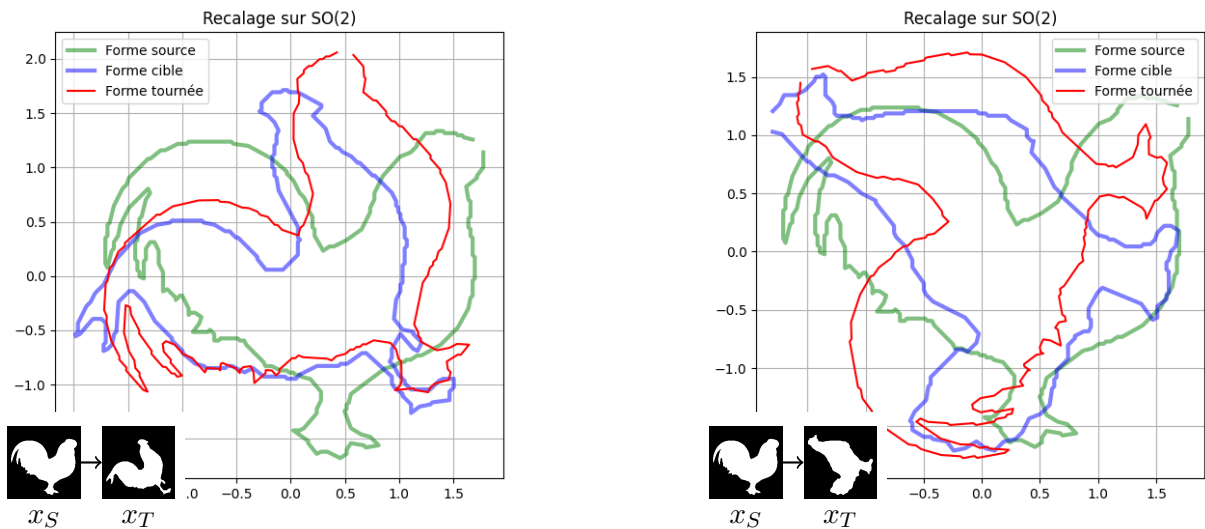


FIGURE 6.3 – Recalage sur les rotations  $SO(2)$  de différents coqs.

## 6.2 Approche hamiltonienne - Tir géodésique

On reprend maintenant la formulation hamiltonienne établie au **Chapitre 4** dans le cas de  $N$  landmarks pour faire un algorithme de résolution par *tir géodésique* (*geodesic shooting*) [4], où la seule variable d'optimisation est le moment initial.

**Proposition 6.3** (Équations géodésiques pour les nuages de points sous  $SO(n)$ ).

Les trajectoires optimales  $(q_t, p_t) \in T^*Q$  du problème de recalage restreint à  $SO(n)$  vues précédemment dans le **Chapitre 4** sont solutions de

$$\begin{aligned}\dot{q}_t &= A_t^* q_t \\ \dot{p}_t &= A_t^* p_t\end{aligned}$$

avec

$$A_t^* = \frac{1}{2} (p_t q_t^\top - q_t p_t^\top) \in \mathfrak{so}(n).$$

*Preuve.* L'équation sur  $q$  s'obtient directement en substituant le contrôle optimal  $A^*$  dans la dynamique  $\dot{q}_t = A_t q_t$ .

Pour l'équation sur  $p$ , on dérive le Hamiltonien réduit  $H_r(q, p) = H(q, p, A^*(q, p))$  par rapport à  $q$ . Comme  $A^*(q, p)$  maximise  $H(q, p, \cdot)$  sur  $\mathfrak{so}(n)$ , la contribution de la dérivée de  $A^*$  par rapport à  $q$  disparaît et il reste donc seulement  $H$  exprimé en fonction de  $q$  :

$$\partial_q H_r(q, p) = \partial_q H(q, p, A^*) = (A^*)^\top p,$$

puisque  $H(q, p, A) = p^\top(Aq) - \frac{1}{2}\|A\|^2$  est linéaire en  $q$ . Les équations hamiltoniennes donnent alors

$$\dot{p}_t = -\partial_q H_r(q_t, p_t) = -(A_t^*)^\top p_t = A_t^* p_t,$$

où l'on a utilisé l'antisymétrie de  $A_t^* \in \mathfrak{so}(n)$ . □

Comme on disait juste au-dessus et donc contrairement à l'approche directe, on n'optimise plus le contrôle  $\omega_t$  (ou  $A_t$ ) le long de la trajectoire, mais uniquement la condition initiale  $p_0 \in (\mathbb{R}^n)^N$  du moment. L'algorithme est le suivant :

— **Initialisation** : On choisit un moment initial :

$$p_0^{(0)} = 0.$$

— **Boucle d'optimisation** (pour chaque itération  $k$ ) :

1. *Tir géodésique* (intégration d'Euler explicite des équations de la *Proposition 6.3*) :  
En partant de  $x_0 = x_S$  et  $p_0 = p_0^{(k)}$ , on intègre simultanément les deux dynamiques :

$$x_{t+\Delta t} = x_t + \Delta t A_t^* x_t, \quad p_{t+\Delta t} = p_t + \Delta t A_t^* p_t,$$

où le champ optimal  $A_t^*$  est recalculé à chaque pas à partir de  $(x_t, p_t)$  courants.

On obtient ainsi  $x_1 = x_1(p_0^{(k)})$  la déformation finale.

2. *Évaluation de l'énergie* : On calcule le coût global

$$J(p_0^{(k)}).$$

3. *Mise à jour* (descente de gradient sur  $p_0$ ) : On calcule

$$p_0^{(k+1)} = p_0^{(k)} - \alpha \nabla J(p_0^{(k)}).$$

*Remarque 6.4.* La taille de la variable d'optimisation  $p_0$  est fixée par la dimension de l'espace des formes ( $nN$ ), indépendamment du nombre de pas de temps utilisés pour l'intégration. C'est précisément l'intérêt qu'on a mentionné au **Chapitre 4** : le problème de contrôle optimal, initialement posé sur un espace de contrôles  $v_t$  de dimension infinie, se ramène à l'optimisation d'un unique moment initial  $p_0$ .

### 6.2.1 Recalage rigide sur $SO(2)$

#### Étape 1 : Calcul du générateur optimal

La fonction `hamiltonian_A` (A.4) calcule  $A_t^* = \frac{1}{2} (p_t q_t^\top - q_t p_t^\top)$  à partir des tenseurs  $p_t$  et  $x_t$ .

#### Étape 2 : Tir géodésique

La fonction `Euler_Hamilton` (A.5) intègre simultanément  $x_t$  et  $p_t$  par la méthode d'Euler explicite sur  $t \in [0, 1]$ , en réévaluant  $A_t^*$  à chaque pas.

#### Étape 3 : Évaluation de l'énergie

La fonction `hamiltonian_loss` (A.6) calcule l'énergie totale

$$J(p_0) = \frac{1}{2} \|p_0\|^2 + \|x_1(p_0) - x_T\|.$$

#### Étape 4 : Mise à jour (descente de gradient sur $p_0$ )

La fonction `opti_hamiltonien` (A.7) optimise le moment initial  $p_0$  par descente de gradient.

On n'illustrera pas cette partie par des exemples imagés car le changement de méthode n'impacte pas le résultat. Mais voici un exemple comparatif sur le temps d'exécution des deux méthodes.

**Exemple 6.5.** On compare le temps d'exécution des deux méthodes d'optimisation pour la rotation présentée en **Figure (6.1)** (100 itérations de *L-BFGS*, mêmes points source et cible, même intégrateur d'Euler à 20 pas de temps) :

$$t_{\text{Hamiltonien}} \approx 0.91 \text{ s}, \quad t_{\text{Scalaire}} \approx 0.56 \text{ s}.$$

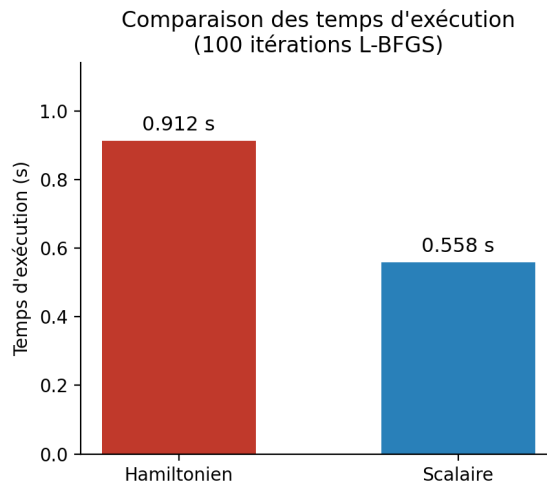


FIGURE 6.4 – Comparaison des temps d'exécution sur la rotation de la **Figure 6.1**

*L'approche scalaire est ici environ 1,6 fois plus rapide que l'approche hamiltonienne sur cette expérience particulière. Cet écart s'explique par le nombre de variables intégrées à chaque pas de temps : l'approche directe ne propage que la position  $x_t$ , tandis que l'approche hamiltonienne propage simultanément la position  $x_t$  et le moment  $p_t$ , ce qui double la quantité de calcul à chaque itération d'Euler tout en recalculant  $A_t^*$  à partir des deux tenseurs courants. En contrepartie, rappelons que l'approche hamiltonienne optimise un vecteur  $p_0$  de même dimension que les nuages de points, ce qui la rend directement généralisable à des dynamiques bien plus riches (comme  $SO(3)$  ou le cadre LDDMM général avec les difféomorphismes en dimension infinie), là où l'approche scalaire directe repose spécifiquement sur la paramétrisation par un angle  $\omega$  propre à  $SO(2)$ .*

# Conclusion

Ce stage a permis d'étudier un cas particulier du recalage de formes dans lequel les transformations sont restreintes aux rotations. Après avoir rappelé le cadre général LDDMM, nous avons décrit le groupe de Lie  $SO(n)$ , son algèbre de Lie  $\mathfrak{so}(n)$  et le rôle de l'application exponentielle pour les rotations.

La formulation hamiltonienne a constitué l'une des notions les plus challengeantes de ce travail. Le fait de comprendre que la recherche d'un champ de vitesse pouvait se ramener à l'intégration d'équations géodésiques dont la seule inconnue est le moment initial a permis d'appréhender concrètement l'intérêt du principe du maximum de Pontryagin. Les études des cas  $SO(2)$  et  $SO(3)$  ont rendu ces concepts plus clairs, notamment à travers les formulations simples du problème de minimisation en fonction d'un unique scalaire ou vecteur.

D'un point de vue algorithmique : les deux implémentations, l'approche directe et le tir géodésique, ont confirmé l'efficacité du modèle lorsque la transformation est une rotation rigide, en mettant aussi en évidence ses limites quand on se restreint à ces transformations "simples". Dès qu'interviennent des déformations non rigides, la restriction à  $SO(n)$  ne permet plus de représenter correctement les transformations. Ce travail met ainsi en lumière un compromis entre la rapidité de l'approche directe et la généralité de l'approche hamiltonienne.

Plusieurs prolongements naturels sont possibles, notamment l'abandon de la limitation aux rotations afin de se rapprocher du modèle LDDMM complet, ou encore l'extension aux formes plus riches que les simples landmarks. Au-delà de ces perspectives, ce stage a surtout constitué une occasion de mettre en pratique les outils d'algèbre et d'équa-diff, vus en cours théorique, dans un contexte appliqué. Il a aussi ouvert des portes sur des domaines des mathématiques pas étudiés en 2<sup>e</sup> année de licence.

# Références

- [1] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2) :139–157, 2005.
- [2] Darryl D Holm. *Geometric mechanics-Part I : Dynamics and symmetry*. World Scientific Publishing Company, 2011.
- [3] Darryl D Holm. 31 lectures on geometric mechanics. *arXiv preprint arXiv :2408.09564*, 2024.
- [4] Michael I Miller, Alain Trouvé, and Laurent Younes. Geodesic shooting for computational anatomy. *Journal of mathematical imaging and vision*, 24(2) :209–228, 2006.
- [5] Rayane Mouhli and Thomas Pierron. Decoupling actions of finite-dimensional lie groups and of groups of diffeomorphisms in the large deformation framework. *arXiv preprint arXiv :2511.14151*, 2025.
- [6] Thomas Pierron. A right-invariant sub-riemannian setting for large deformation models. *ESAIM : Proceedings and Surveys*, 79 :69–82, 2025.
- [7] Thomas Polzin. *Large deformation diffeomorphic metric mappings : Theory, numerics, and applications*. PhD thesis, Zentrale Hochschulbibliothek Lübeck, 2018.
- [8] Emmanuel Trélat and Fanny Bastien. Emmanuel Trélat - Théorie du contrôle optimal et applications aux missions spatiales, February 2016.
- [9] Laurent Younes. *Shapes and diffeomorphisms*, volume 171. Springer, 2010.
- [10] Irigo Edouard Zibo. *Contrôle optimal : méthodes géométriques et numériques et applications au modèle de Bloch, et intégrabilité des géodésiques sous-riemanniennes*. Theses, Normandie Université, November 2021.

# A Code Python

Tout le code sera disponible sur [ce dépôt](#) GitHub.

```
1 def Euler(q0, omega, nt):
2     dt = 1.0 / (nt - 1)
3     q = q0.clone()
4     A = skew_matrix(omega)
5
6     Q = torch.zeros((q0.shape[0], q0.shape[1], nt), dtype=q0.dtype,
7 device=q0.device)
8     Q[:, :, 0] = q
9
10    for t in range(1, nt):
11        q = q + dt * (q @ A.T)
12        Q[:, :, t] = q
13
14    return q, Q
```

Code A.1 – Fonction Euler

```
1 def rotation_loss(data_loss_fnc, nt):
2     def loss(q0, omega):
3         q_final, _ = Euler(q0, omega, nt)
4         dloss = data_loss_fnc(q_final)
5         reg = 0.5 * (omega ** 2)
6         return reg + dloss
7     return loss
```

Code A.2 – Fonction rotation\_loss

```
1 def opti_rotation(q0, loss_fnc, n_iter, eta):
2     omega = torch.zeros(1, dtype=q0.dtype, device=q0.device,
3 requires_grad=True)
4     optimizer = torch.optim.LBFGS([omega], lr=eta, line_search_fn='
5 strong_wolfe')
6     losses = []
7
8     def closure():
9         optimizer.zero_grad()
10        L = loss_fnc(q0, omega)
11        L.backward()
12        return L
13
14    for i in range(n_iter):
15        L = optimizer.step(closure)
16        losses.append(L.item())
17
18    return omega.detach(), losses
```

Code A.3 – Fonction opti\_rotation

```

1 def hamiltonian_A(p, q):
2     return 0.5 * (p.T @ q - q.T @ p)

```

Code A.4 – Fonction hamiltonian\_A

```

1 def Euler_Hamilton(q0, p0, nt):
2     dt = 1.0 / (nt - 1)
3     q = q0.clone()
4     p = p0.clone()
5     Q = torch.zeros((q0.shape[0], q0.shape[1], nt), dtype=q0.dtype,
6                     device=q0.device)
7     P = torch.zeros((p0.shape[0], p0.shape[1], nt), dtype=p0.dtype,
8                     device=p0.device)
9     Q[:, :, 0] = q
10    P[:, :, 0] = p
11
12    for t in range(1, nt):
13        A = hamiltonian_A(p, q)
14        q = q + dt * (q @ A.T)
15        p = p + dt * (p @ A.T)
16        Q[:, :, t] = q
17        P[:, :, t] = p
18
19    return q, p, Q, P

```

Code A.5 – Fonction Euler\_Hamilton

```

1 def hamiltonian_loss(data_loss_fnc, nt):
2     def loss(q0, p0):
3         q_final, p_final, _, _ = Euler_Hamilton(q0, p0, nt)
4         dloss = data_loss_fnc(q_final)
5         reg = 0.5 * (p0 ** 2).sum()
6         return reg + dloss
7     return loss

```

Code A.6 – Fonction hamiltonian\_loss

```

1 def opti_hamiltonian(q0, loss_fnc, n_iter, eta):
2     p0 = torch.zeros_like(q0, requires_grad=True)
3     optimizer = torch.optim.LBFGS([p0], lr=eta, line_search_fn='
4     strong_wolfe')
5     losses = []
6
7     def closure():
8         optimizer.zero_grad()
9         L = loss_fnc(q0, p0)
10        L.backward()
11        return L
12
13    for i in range(n_iter):
14        L = optimizer.step(closure)
15        losses.append(L.item())
16
17    print(f"Iteration {i+1:3d} - Loss: {L.item():.6f} | |p0|: {p0.
18    norm().item():.4f}")
19    print(f"")
20
21    return p0.detach(), losses

```

Code A.7 – Fonction opti\_hamiltonian

## B Figures supplémentaires

Quelques figures supplémentaires.

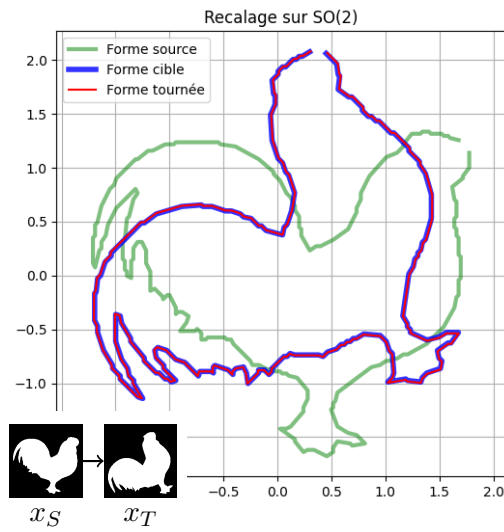


FIGURE B.1 – Complète l'exemple (1.2)

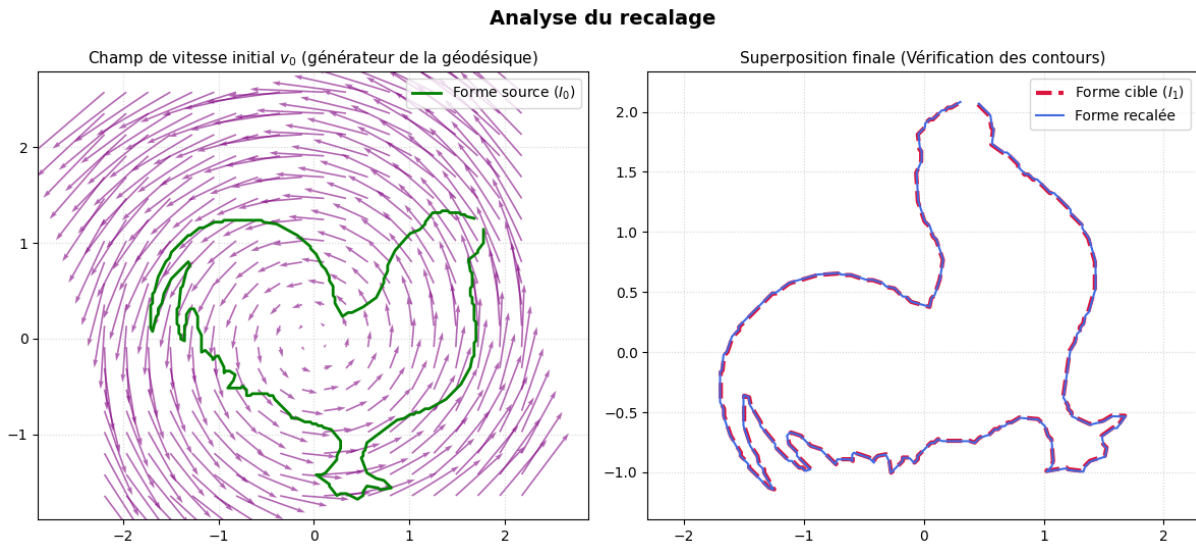


FIGURE B.2 – Complète l'exemple (1.2)

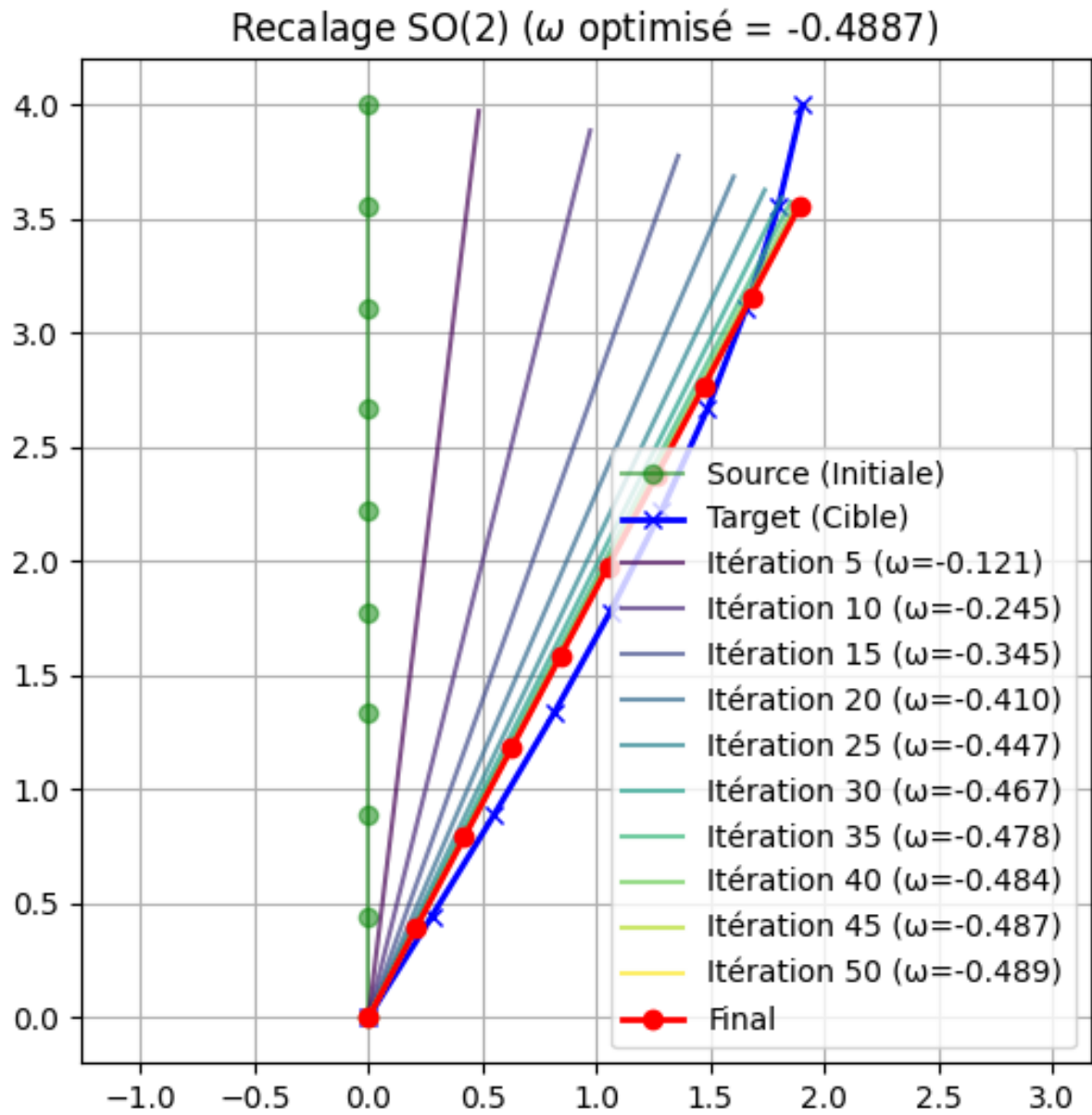


FIGURE B.3 – Complète l'exemple (6.1)

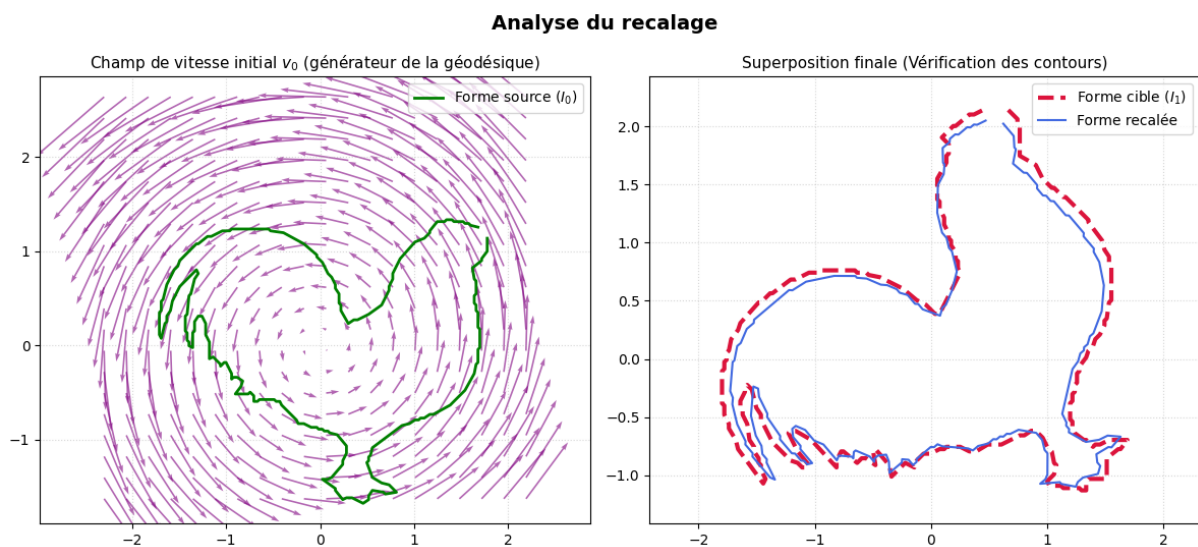


FIGURE B.4 – Complète l'exemple (6.2)

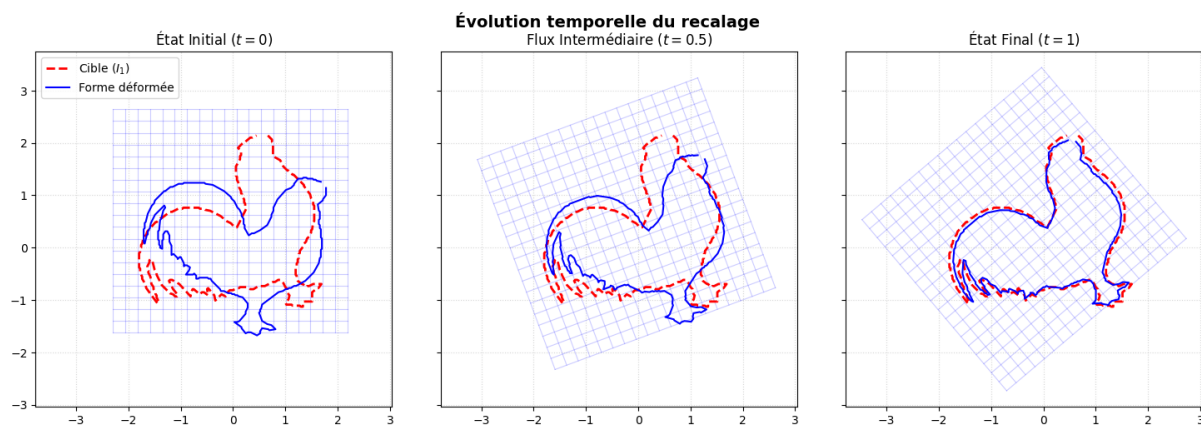


FIGURE B.5 – Complète l'exemple (6.2)

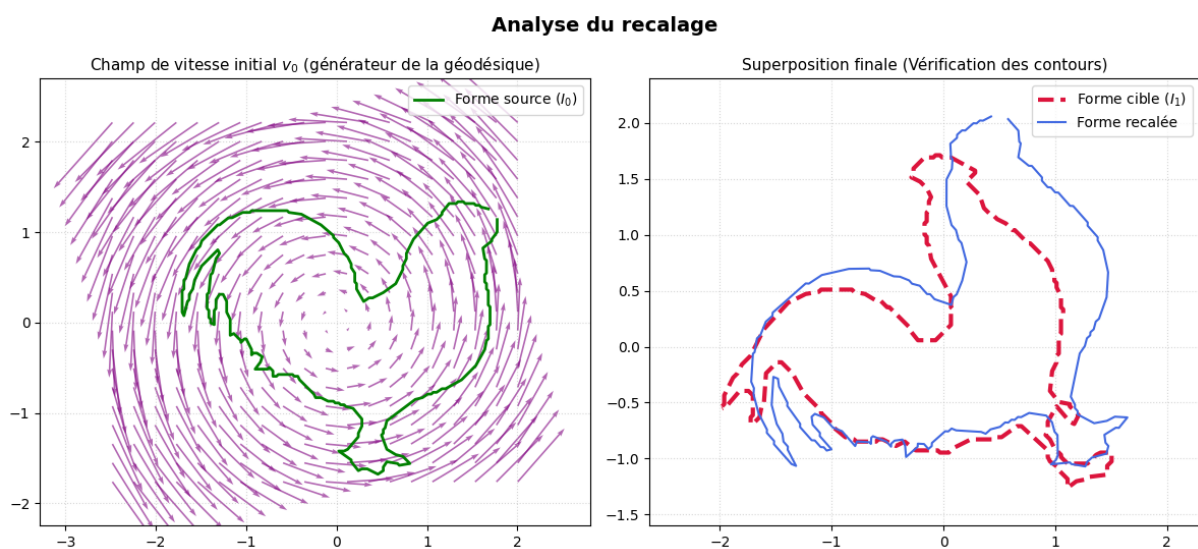


FIGURE B.6 – Complète l'exemple (6.2)

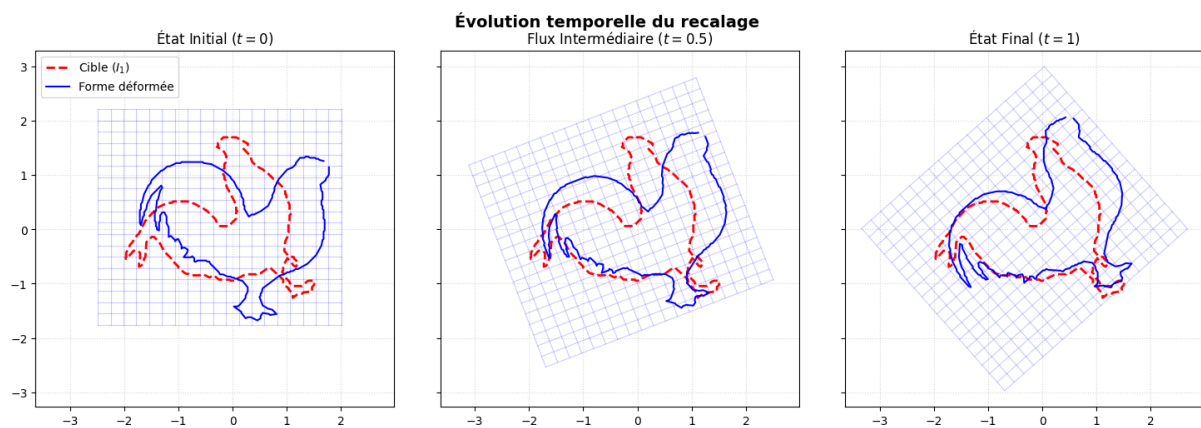


FIGURE B.7 – Complète l'exemple (6.2)

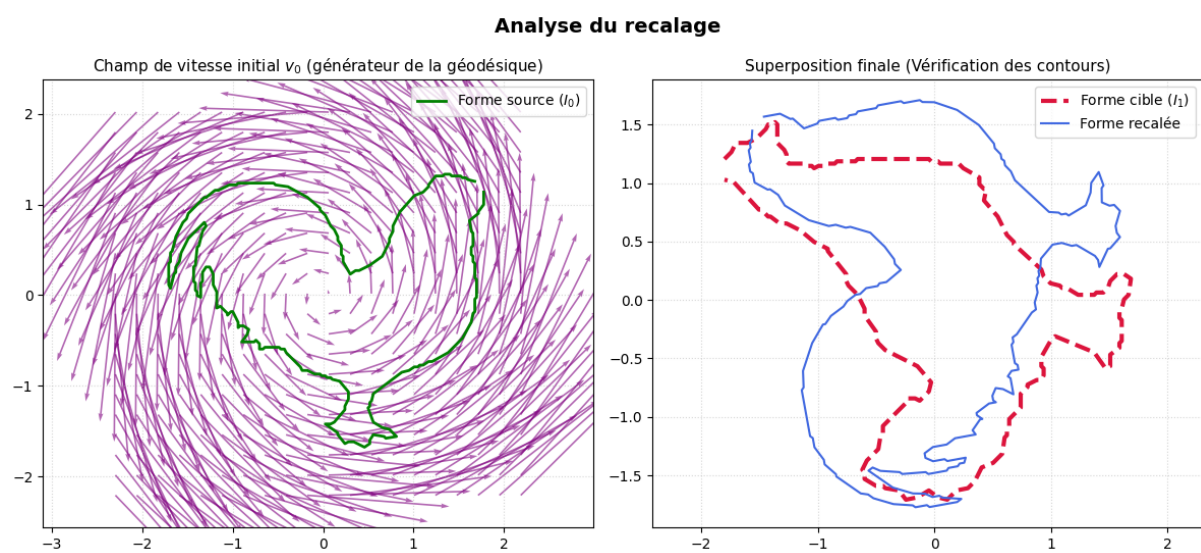


FIGURE B.8 – Complète l'exemple (6.2)

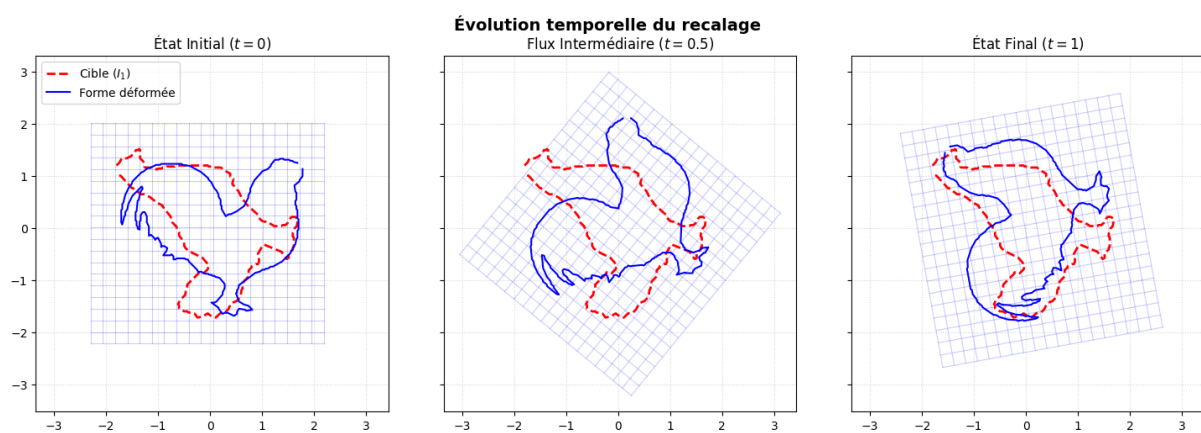


FIGURE B.9 – Complète l'exemple (6.2)