Riemannian Algorithms on the Stiefel and the Fixed-Rank Manifold

Sutti, Marco

UNIVERSITÉ DE GENÈVE
Section de Mathématiques

FACULTÉ DES SCIENCES
Professeur Bart Vandereycken

Riemannian Algorithms on the Stiefel and the Fixed-Rank Manifold

THÈSE

Présentée à la Faculté des Sciences de l'Université de Genève
pour obtenir le grade de Docteur ès Sciences, mention Mathématiques

par

Marco SUTTI
de
Morbegno (Italie)

Thèse N°5514

GENÈVE
Atelier d'impression ReproMail
2020

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES

DOCTORAT ÈS SCIENCES, MENTION MATHEMATIQUES

Thèse de Monsieur Marco SUTTI

intitulée :

«Riemannian Algorithms on the Stiefel and the Fixed-Rank Manifold»

La Faculté des sciences, sur le préavis de Monsieur B. VANDEREYCKEN, professeur associé et directeur de thèse (Section de mathématiques), Monsieur M. GANDER, professeur ordinaire (Section de mathématiques), Monsieur N. BOUMAL, professeur (Tenure Track, Ecole Polythechnique Fédérale de Lausanne, Lausanne, Suisse), Monsieur A. USCHMAJEW, professeur (Leader of Max Planck Research Group, MPI Leipzig, Leipzig, Germany), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 24 novembre 2020

Thèse - 5514 -

Le Doyen

N.B. - La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

Sī superficiēs curva in quāmcumquē aliam superficiem explicātur, mēnsūra curvātūrae in singulīs pūnctīs invariāta manet.

Carl Friedrich Gauß,
Disquīsītiōnēs generālēs circā superficiēs curvās, 1827 Oct. 8

Abstract

This thesis is concerned with numerical algorithms on matrix manifolds. It is divided into four parts, and in all of them, we make extensive use of Riemannian geometry. The interest in considering optimization algorithms on matrix manifolds instead of classical algorithms lies in the fact that by exploiting the underlying geometric structure of the problems, they allow taking explicitly into account the constraints.

Shooting methods have been known for quite some time to find a numerical solution to a boundary value problem. Here, we describe and specialize these methods to the Stiefel manifold, discuss their limitations, and provide some numerical examples.

Another method for finding geodesics is the leapfrog algorithm of L. Noakes. This algorithm is related to the Gauss–Seidel method, a classical iterative method for solving a linear system of equations, which can be easily extended to nonlinear systems. We propose a convergence proof of leapfrog as a nonlinear Gauss–Seidel method. Our discussion is limited to the case of the Stiefel manifold, however it may be generalized to other embedded submanifolds. We discuss other aspects of leapfrog and present some numerical experiments.

We tackle the problem of numerical accuracy in line-search methods. It is known that when using standard Wolfe conditions, one can attain a numerical accuracy only on the order of the square root of the machine precision. The Hager–Zhang line search, which employs the approximate Wolfe conditions, offers a workaround to this problem. We give an overview of this technique, and we generalize it to the Riemannian framework. Numerical examples show that this new generalization permits to achieve an accuracy on the order of machine precision when using line search for optimization problems on manifolds.

Multilevel optimization is the extension of multigrid to unconstrained optimization. We introduce a new generalization of multilevel optimization to the case of Riemannian manifolds, and we demonstrate its effectiveness through numerical experiments for the manifold of fixed-rank matrices. Our method combines the classical components of multigrid and those of Riemannian optimization. To cope with the curvature of the manifold, we need to introduce the additional tools from Riemannian optimization that allow a generalization of the existing Euclidean algorithm to manifolds. Our generalization of the Hager–Zhang line search is also used.

Résumé

Cette thèse porte sur les algorithmes numériques sur les variétés matricielles. Elle est divisée en quatre parties, dans lesquelles on fait un usage intensif de la géométrie riemannienne. L'intérêt d'envisager des algorithmes d'optimisation sur des variétés matricielles plutôt que des algorithmes classiques réside dans le fait qu'en exploitant la structure géométrique sous-jacente des problèmes, ils permettent de prendre en compte explicitement les contraintes.

Les méthodes de tir permettent de trouver une solution numérique à un problème aux limites. Nous décrivons et spécialisons ces méthodes à la variété de Stiefel, discutons de leurs faiblesses, et fournissons quelques exemples numériques.

Une autre méthode pour trouver des géodésiques est l'algorithme leapfrog de L. Noakes. Cet algorithme est lié à la méthode de Gauss–Seidel, une méthode itérative classique pour résoudre un système linéaire d'équations, qui peut être aisément étendue à des systèmes non linéaires. Nous proposons une preuve de convergence de la méthode leapfrog comme méthode de Gauss–Seidel non linéaire. Notre discussion se limite au cas de la variété de Stiefel, mais elle peut être généralisée à d'autres sous-variétés plongées. Nous discutons d'autres aspects de la méthode leapfrog et présentons quelques expériences numériques.

Nous abordons le problème de la précision numérique dans les méthodes de recherche linéaire. On sait qu'en utilisant les critères de Wolfe classiques, on ne peut obtenir une précision numérique que de l'ordre de la racine carrée de la précision de la machine. La recherche linéaire de Hager–Zhang, qui utilise les critères de Wolfe approchés, offre une solution à ce problème. Nous donnons un aperçu de cette technique, et nous la généralisons au cadre riemannien. Des exemples numériques montrent que cette nouvelle généralisation permet d'obtenir une précision de l'ordre de la précision machine lors de l'utilisation de la recherche linéaire pour des problèmes d'optimisation sur des variétés.

L'optimisation multi-niveaux est l'extension de la méthode multigrille à l'optimisation sans contraintes. On introduit une nouvelle généralisation de l'optimisation multi-niveaux au cas des variétés riemanniennes, et nous illustrons son efficacité par des expériences numériques pour la variété des matrices de rang fixé. Notre méthode combine les composantes classiques du multigrille et celles de l'optimisation riemannienne. Pour faire face à la courbure de la variété, nous devons introduire les outils supplémentaires de l'optimisation riemannienne qui permettent une généralisation de l'algorithme euclidien existant aux variétés. Notre généralisation de la recherche linéaire de Hager–Zhang est également utilisée.

Remerciements

Qui me connaît un petit peu est au courant que mon expérience en Suisse a commencé il y a plus de sept ans. Il s'agit d'un laps de temps énorme : il y a eu plein de souvenirs indélébiles, de différents stades, et surtout le soutien et la compagnie de beaucoup de gens. Je crois qu'il est naturel que, pendant un espace de temps aussi long, on change beaucoup, on fait de nouvelles connaissances et on en perd de vieilles sur le chemin. Toutefois, dans les moments de réflexion, il m'arrive toujours de penser aussi à ces personnes qui ne figurent plus dans ma vie quotidienne. Ils ont aussi contribué, directement ou indirectement, à l'aboutissement de cette thèse, et je demande pardon si j'ai oublié de mentionner explicitement quelqu'un. Ces pages de remerciements sont dédiées à vous.

Le plus grand merci est sans doute destiné à Bart, mon directeur de thèse : en premier, pour m'avoir accepté comme étudiant de doctorat, et puis, pour les nombreuses discussions, pour ta patience, ta disponibilité, tes remarques tout au long de la rédaction. C'était un plaisir d'apprendre de toi dans un domaine que tu domines complètement alors que j'étais complètement désorienté quand je suis arrivé à l'Université de Genève. À cette époque-là, je n'étais pas très à l'aise avec le raisonnement mathématique, ni avec la manière d'écrire un article scientifique. Pendant ces quatre années, mon style d'écriture en a certainement bénéficié énormément. Tu as toujours été plein d'idées et je regrette que nous n'ayons finalement pas eu assez de temps pour tout concrétiser, et ce n'est que de ma faute. Merci aussi pour les dîners et les autres moments passés en dehors de la section de mathématiques.

Je remercie également Nicolas Boumal, Martin J. Gander et André Uschmajew pour avoir accepté d'être membres du jury. Au-delà des aspects purement liés à la rédaction de cette thèse, je vous remercie aussi pour les moments conviviaux que l'on a partagés ensemble. Nicolas, merci pour le cours très agréable et précis que tu as donné pendant l'école d'hiver à Villars-sur-Ollon en janvier 2020, ainsi que pour tes retours très précieux sur le manuscrit *Multilevel Riemannian optimization for low-rank problems*. Martin, merci pour ta bonne humeur au fils de toutes ces années, pour la balade et le barbecue au sommet du Salève et pour les grillades chez toi. André, merci pour nous avoir fait une visite guidée du département de maths de l'Université Technique de Berlin et pour la sortie au Biergarten lors de la conférence ICCOPT 2019.

Il m'est impossible de conclure mon expérience doctorale sans mentionner ce qui l'a précédée, à savoir, mon master à l'École Polytechnique Fédérale de Lausanne (EPFL). Je remercie donc les amis de l'époque de l'EPFL, et surtout Francesca et Irene, avec lesquelles mon séjour en Suisse a commencé dans le lointain août 2013.

Matteo, merci pour l'amitié et tous les bons moments que l'on a partagés pendant nos études à l'EPFL et après, pour les discussions, les sorties et les films.

Je ne peux pas oublier ma petite parenthèse estivale à Lugano comme stagiaire au Centre Suisse de Calcul Scientifique (CSCS). Je tiens à remercier vivement Claudio pour m'avoir donné l'opportunité de travailler avec lui, ainsi que Luca, Gabriella, Hussein et les autres collègues du CSCS.

En 2016, j'ai déménagé à Genève. J'avoue qu'avant de m'établir ici, je m'attendais de passer la plupart de mon temps à faire de la recherche, mais à Genève on doit aussi dédier pas mal de temps à l'enseignement. Apprendre une nouvelle langue à l'âge de vingt-sept ans, lors de mon arrivée à Lausanne, ce n'était pas évident. C'était encore moins évident de la maîtriser pour pouvoir l'utiliser dans l'enseignement mathématique. Et pourtant, des fois, quand la recherche n'engendrait que de la frustration, les meilleures satisfactions sont arrivées par l'enseignement. Je remercie donc les étudiants qui m'ont manifesté leurs appréciations pendant ces quatre années. Vos commentaires très positifs sur les séances d'exercices m'ont aidé en période de difficulté.

Merci à Adrien, Léo et Roberto pour m'avoir accueilli dans leur groupe de grimpeurs, et pour d'autres bons moments partagés pendant cette année imprévisible. Pedro, merci pour t'être intéressé à l'analyse numérique et pour avoir eu le courage d'aborder ma thèse.

Je suis également reconnaissant à toutes les personnes qui ont créé une ambiance agréable à la section de maths. Merci aux collègues qui m'ont accueilli lorsque j'y venais d'arriver, et qui sont partis avant moi, Gabriele, Pascaline, Vladimir, Ding (鲁玎), Bo (宋博), Maxime, Fathi, Minh, Parisa et Aitor pour tous les moments passés ensemble.

Pratik, merci pour les repas indiens et les pizzas chez toi, et pour ton soutien lors de mes moments de découragement, pour m'avoir compris sans avoir besoin de beaucoup de mots.

Eiichi, merci pour les nombreuses discussions à la section et, pendant ces derniers temps, virtuelles. La section de maths n'aurait pas été la même sans tes blagues.

Merci à mes compagnons de bureau qui m'ont supporté pendant environ trois ans, et qui m'ont aussi beaucoup aidé à progresser en français. Sandie, pour m'avoir appris ce que sont les contrepèteries et quelques mots de l'italien parlé en Suisse qui n'existent pas en Italie, comme *picobello*. Adrien, merci pour les suggestions de films et bandes dessinées, pour les cinémas et les randonnées, pour la relecture de ma thèse, mais pas pour *L'Avare* de Molière.

Tommaso, merci pour l'amitié, les dîners et les balades en vélo, et surtout pour les moments d'adrénaline au parc aventure à Onex, qui nous a fait sortir de notre zone de confort.

Merci aussi aux autres collègues qui sont arrivés après moi à la section (en ordre chronologique) : Ibrahim, Guillaume, Raphaël, Pablo, Conor, Michal, Giancarlo, Ausra. Je sais que souvent je n'ai pas eu le temps de discuter avec tout le monde comme j'aurais voulu, mais c'est sûr que mon expérience à la section n'aurait pas été la même sans vous.

Thibaut, merci pour notre balade en vélo au sommet du Salève, pour les entraînements ensemble pour la Course de l'Escalade 2018, et pour les après-midi jeux et crêpes à Confignon.

Je remercie également nos secrétaires, Annick et Nathalie, pour leur disponibilité et pour les moments de convivialité au Z-Bar. En particulier Joselle, merci pour ton aide pratique dans la jungle de l'administration genevoise et pour certains cahiers de charges très marrants.

Indubitablement, il y a eu de nombreux défis. Étant issu d'un milieu différent, celui de l'ingénierie, le plus grand défi a été celui de communiquer avec les collègues dont le chemin s'est toujours déroulé dans le sillon des mathématiques. Souvent, il a été très difficile d'expliquer mon parcours. Je suis donc très reconnaissant aux rares personnes qui ont su m'écouter sans préjugés ni moqueries.

Heureusement qu'à Genève il n'y a pas eu que le temps passé à la section de mathématiques. À ce propos, je voudrais remercier particulièrement ceux qui m'ont fait sentir moins étranger pendant mon séjour en Suisse, notamment la famille Hongler, Vân, Max, Isadora et Clément. Merci pour m'avoir accueilli comme si je faisais partie de votre famille.

J'ai eu aussi l'opportunité d'apprendre une nouvelle langue : le chinois mandarin. Je remercie donc Fan Leshan (范乐山), mon enseignant pendant la première année de cours, sans lequel mon début dans l'apprentissage du chinois n'aurait pas été aussi agréable.

Taisuke (泰介), merci pour les discussions intéressantes à propos de politique internationale et des droits humains : en s'éloignant des sujets matheux, elles m'ont aidé à garder ma santé mentale.

Jhih-Huang (志煌), merci pour m'avoir appris beaucoup de choses sur le taoïsme, et pour m'avoir fait découvrir le vélo et ton pays fascinant, Taïwan. Merci surtout pour m'avoir toujours soutenu, et pour avoir écouté (et supporté) mes doutes à chaque fois que je sentais le besoin d'en parler.

Finalmente i miei pensieri vanno all'Italia. In questi anni, le mie interazioni sociali, a causa di distanze e impegni, sono state notevolmente ridotte rispetto a quanto alcuni si sarebbero aspettati da me. Vorrei quindi scusarmi con coloro che avevano bisogno di stare con me durante la mia assenza, perché stavo perseguendo questo obiettivo. Come dicevo all'inizio, molte persone sono responsabili, direttamente o indirettamente, di ciò che ho fatto, e forse in alcune circostanze hanno sofferto la mia assenza quanto me, quindi meritano altrettanto credito per questo traguardo.

Prima di tutto vorrei ringraziare gli amici valtellinesi che hanno reso più piacevoli i miei numerosi ritorni a casa durante questi anni. Sonia, grazie per avermi accompagnato a Lugano quando cercavo il mio stage da fare durante il master all'EPFL. Sabrina e Alessandro, grazie per essere venuti a trovarmi a Ginevra; sebbene il caffè a Yvoire non fosse come quello italiano, ho apprezzato molto la vostra visita e la nostra piccola escursione in Alta Savoia.

Poi ci sono gli amici valtellinesi che in questi anni non erano in Valtellina, ma con i quali ho sempre mantenuto i legami. In particolare Lele, grazie per la tua amicizia che non è mai venuta meno durante tutti questi anni in cui spesso siamo stati molto lontani, e per le visite che mi hai reso, dapprima a Losanna nel 2014, e poi a Ginevra a gennaio 2020.

Ringrazio gli amici della zona del lago di Como, tra cui, in particolare, Stefano, per essere venuto a trovarmi dapprima a Lugano, durante la mia parentesi estiva al CSCS, e poi a Ginevra, all'inizio del mio dottorato. Ed Enrico: ci siamo visti raramente in questi anni, ma è sempre stato un piacere discutere con te, così come è stato un piacere leggere il tuo romanzo durante l'estate del 2020.

Ultimi in questa lista degli amici, ma non meno importanti, gli amici di Bergamo. Dan e Paolo, grazie per essere venuti a trovarmi a Losanna nel 2015, sfidando la lunghezza del viaggio e, soprattutto, i prezzi svizzeri. Andrea, grazie per avermi fatto conoscere la città di Anversa in bicicletta durante il mio soggiorno a Leuven, all'inizio 2017.

Le parole non saranno mai abbastanza adatte né sufficienti per esprimere il mio ringraziamento verso i miei genitori e la mia famiglia in generale, che mi è sempre stata di sostegno durante tutti questi anni, e che mi ha fornito un supporto che non è stato solo morale e psicologico, ma anche pratico.

Infine, ci sono alcune persone che se ne sono andate durante questi anni. Sono partito dall'Italia nel 2013, e quando tornavo di tanto in tanto, non c'erano più. Zio Ezio, nonna "Pace", zia Ernesta, zia Giovanna. Affrontare la perdita di quelle persone care che erano state al mio fianco fin da quando sono nato è stata un'ulteriore sfida per me.

Je voudrais ajouter une dernière réflexion à ces remerciements, une métaphore, en effet. Faire une thèse c'est un peu comme aller en vélo : il faut savoir trouver son propre rythme, comprendre quand il est nécessaire de changer la vitesse, ou quand il faut prendre des pauses. Comme ce n'est pas un travail en solitaire, il faut avoir de bons compagnons. Pour trouver son propre rythme, il faut éviter de se laisser influencer par les gens qui nous entourent. Car, de même qu'il n'y a pas un seul modèle de vélo, ainsi il n'y a pas un seul type de doctorat ; de même qu'il n'y a pas une seule façon de pédaler, ainsi il n'y a pas une seule façon de rédiger une thèse. Je trouve donc très approprié de conclure ces réflexions avec une citation de Wu Ming-Yi (吳明益), un auteur taïwanais que j'ai beaucoup apprécié :

The only necessity is to keep pedaling—quietly, composedly, no matter how thirsty you are or how difficult it may be.

Contents

Introduction

Motivation

Several applications in optimization, image, and signal processing deal with data belonging to matrix manifolds. These are manifolds in the sense of classical Riemannian geometry, where variables are matrices.

In this thesis, we present and study some numerical algorithms on matrix manifolds. This work is divided into four main parts, and in all of them, we make extensive use of Riemannian geometry.

Curves and surfaces were the original object of study of classical differential geometry, and Riemannian manifolds can be regarded as abstract generalizations of those objects. Hence, when thinking about manifolds, it will sometimes be useful to resort to these more familiar objects for illustration. There exist many nice introductory books on Riemannian geometry, for instance [KN69, Boo86, Sak96, dC92], but for the purposes of the later chapters of this thesis, the review of first-order Riemannian geometry given in the first chapter should be sufficient.

The interest in considering optimization algorithms on matrix manifolds instead of classical algorithms is in the fact that by exploiting the underlying geometric structure of the problems, they allow taking explicitly into account the constraints.

For matrix manifolds, we will often make reference to the pioneering work of [EAS98]. More recent reviews and details on matrix manifolds and related numerical algorithms can be found in [AMS04, HLW06, AMS08, AM12, AMT13, Bou20].

Matrix manifolds considered in this thesis

In this thesis, we will work with the two manifolds presented in this section. Some applications require evaluating the distance between two arbitrary points on the manifold. For some matrix manifolds, like the Grassmann manifold, explicit formulas are available, while for others, one has to resort to numerical algorithms. One example in the latter class is the Stiefel manifold, which is defined as the set $\mathrm{St}(n,p)$ with $p \leqslant n$ of all $n \times p$ orthonormal matrices

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n\times p} \colon\ X^{\mathsf{T}}X = I_p\}.$$

In other applications, the underlying geometric structure is exploited to obtain more effective algorithms. This is, in particular, the case when manifolds of low-rank matrices are used, since one can avoid forming the full matrices and work directly on the low-rank format.

The manifold of matrices of rank k is [Van13, AAM14]

$$\mathcal{M}_k = \{X \in \mathbb{R}^{m\times n} \colon \operatorname{rank}(X) = k\}.$$

Main ideas

The first part of this thesis is more focused on the geometry itself. It deals with the problem of finding the distance between two points on the Stiefel manifold $\mathrm{St}(n, p)$. In this part, we will make extensive use of the notion of *geodesic*. Later we will explain in more detail how geodesics are defined, but for this short introduction, it suffices to say that a geodesic is a curve with zero acceleration, which generalizes the notion of straight lines in Euclidean space to a Riemannian manifold [AMS08].

Geodesics are, *locally*, curves of shortest length, but *globally* they may not be. Indeed, geodesics are in general critical points for the length functional, and may or may not be minima. However, for a connected Riemannian manifold, the Hopf–Rinow theorem [Sak96, p. 84] ensures that any two points can be connected by a length-minimizing geodesic. The geodesic connecting two points on a manifold may not be unique. Figure 1 illustrates this concept for the case of the sphere: geodesics on a sphere are great circles, and the length-minimizing geodesic between any two points is the shorter of the two arcs of a great circle joining them. Shooting methods have been known for quite some time to find a numerical solution to a boundary value problem (BVP). In Chapter 2, we describe and specialize these methods to the Stiefel manifold, discuss their limitations, and provide some numerical examples.

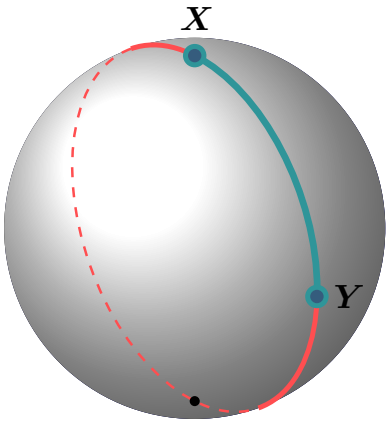


Figure 1 – Geodesics on the sphere.

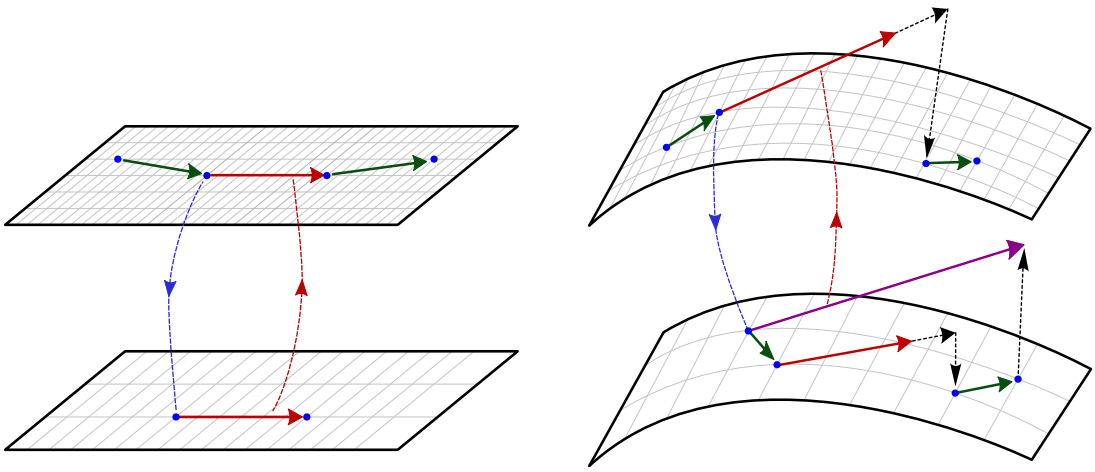
In the second part of the thesis, we study another method for finding geodesics: the leapfrog algorithm introduced by L. Noakes [Noa98]. Noakes realized that his algorithm was in some way imitating the Gauss–Seidel method, a classical iterative method for solving a linear system of equations, which can be easily extended to nonlinear systems. This connection between leapfrog and nonlinear Gauss–Seidel was not further investigated by the author of leapfrog, as it appears from the related papers [KN97, KN98a, KN98b, KN08]. Therefore, in Chapter 3 we propose a convergence proof of leapfrog as a nonlinear Gauss–Seidel method. Our discussion will be limited to the case of the Stiefel manifold, however it may be generalized to other embedded submanifolds. In Chapter 4, we continue the discussion on other aspects of leapfrog and present some numerical experiments.

In the third part of the thesis, we tackle the problem of numerical accuracy in line-search methods. It is known that when using standard Wolfe conditions, one can attain a numerical accuracy only on the order of the square root of the machine precision. Employing the approximate Wolfe conditions and using the line-search technique proposed by [HZ05, HZ06] provides a solution to this problem. In Chapter 5, we give an overview of the Hager–Zhang line search, and we generalize this technique to Riemannian manifolds. Numerical examples show that this new generalization permits to achieve an accuracy on the order of machine precision when using line search for optimization problems on manifolds.

In the final part of the thesis, we use the manifold of fixed-rank matrices $\mathcal{M}_k$ in the context of certain large-scale variational problems arising from the discretization of elliptic PDEs, where the optimization variable is rank-constrained.

Multilevel optimization is the extension of multigrid to unconstrained optimization, and the original idea goes back to the MG/Opt [Nas00, LN05]. Chapter 6 describes basic notions about multigrid and multilevel methods.

In Chapter 7, we introduce a new generalization of the multilevel optimization algorithm to the case of Riemannian manifolds, and we demonstrate its effectiveness through numerical experiments. All the classical components of multigrid are there, plus the additional components from Riemannian optimization. Figure 2 provides an illustration of this generalization, emphasizing the fact that due to the curvature of the manifold, we need to introduce all the additional tools from Riemannian optimization that allow a generalization of the existing Euclidean algorithm to manifolds.



(a) Euclidean (b) Riemannian

Figure 2 – With respect to the Euclidean case (panel (a)), in the Riemannian setting (panel (b)) we need to introduce new tools that allow us to cope with the curvature of the manifold.

Organization of this thesis

This thesis consists of four main parts: the first part comprises chapters 1–2, the second one chapters 3–4, the third part corresponds to chapter 5, while the fourth part encompasses chapters 6–7.

In Chapter 1, we introduce the notions of first-order Riemannian geometry required for the reading of this thesis. In Chapter 2, we apply shooting methods for calculating the dis-

tance between two points on the Stiefel manifold. In Chapter 3, we deal with the leapfrog algorithm and the proof of its convergence as a nonlinear Gauss–Seidel method. Other methods for computing the Riemannian distance between two points on a manifold are discussed in Chapter 4.

In Chapter 5, we review the Hager–Zhang line-search method, and introduce its Riemannian counterpart, which will be used in Chapter 7. In Chapter 6, we describe basic notions about multigrid methods and their derivations. Finally, in Chapter 7, we present our Riemannian multigrid line-search algorithm for low-rank problems.

More precisely, this thesis is divided as follows:

- Chapter 1: introduction to Riemannian geometry.
- Chapter 2: shooting methods to compute the distance on the Stiefel manifold.
- Chapter 3: proof of convergence of leapfrog as a nonlinear Gauss–Seidel method.
- Chapter 4: other methods and extensions on leapfrog.
- Chapter 5: a Riemannian Hager–Zhang line search.
- Chapter 6: introduction to multigrid methods and their derivations.
- Chapter 7: multilevel Riemannian optimization for low-rank problems.

Chapter 1

Riemannian geometry

In this chapter, we will introduce some notions of Riemannian geometry that will provide a useful and necessary background for the rest of this thesis. The first part of this chapter deals with fundamental definitions and first-order geometry, while the second part is more focused on line-search algorithms on manifolds.

Most of this chapter is based on [dC92, Sak96, Lee97, Lee18] for the classical theory of Riemannian geometry, and on [EAS98, AMS08, Bou20] for the algorithms on manifolds. When discussing the classical numerical optimization algorithms in Euclidean space, some material was also taken from [NW06, Ber95].

1.1 First-order geometry

A d-dimensional manifold is a set $\mathcal{M}$ covered with a suitable collection of *charts*, that identify certain subsets of $\mathcal{M}$ with open subsets of $\mathbb{R}^d$. The collection of charts, called *atlas*, provides the basic structure to do differential calculus on $\mathcal{M}$. The numerical algorithms on matrix manifolds exploit the matrix structure associated with the problems of interest.

1.1.1 Charts and atlases

Let $\mathcal{M}$ be a set, and $\mathcal{U}$ be an open subset of $\mathcal{M}$. Charts are useful because they allow us to study in $\mathbb{R}^d$ the objects associated with $\mathcal{U}$.

For example, let $f : \mathcal{U} \to \mathbb{R}$ be a real-valued function on $\mathcal{U}$, then $f \circ \varphi^{-1}$ is a function from $\mathbb{R}^d$ to $\mathbb{R}$, with domain $\varphi(\mathcal{U})$, i.e., $f \circ \varphi^{-1} : \varphi(\mathcal{U}) \to \mathbb{R}$, to which methods of real analysis apply. Each point of the set $\mathcal{M}$ has to be at least in one chart domain. If $x \in \mathcal{U} \cap \mathcal{V}$ then we need some kind of compatibility between the two mappings φ and ψ, i.e., in the overlaps we want some kind of smoothness.

The definition of atlas specifies such compatibility conditions at the overlaps of the charts.

Definition 1.1 (Atlas)**.** A (C^∞) *atlas* of $\mathcal{M}$ into $\mathbb{R}^d$ is a collection of charts $(\mathcal{U}_\alpha, \varphi_\alpha)$ of the set $\mathcal{M}$ such that

- The union of all the charts is the set $\mathcal{M}$, i.e., $\cup_\alpha \mathcal{U}_\alpha = \mathcal{M}$.
- For all α, β with $\mathcal{U}_\alpha \cap \mathcal{U}_\beta \neq \varnothing$, the sets $\varphi_\alpha(\mathcal{U}_\alpha \cap \mathcal{U}_\beta)$ and $\varphi_\beta(\mathcal{U}_\alpha \cap \mathcal{U}_\beta)$ are open sets of $\mathbb{R}^d$ and the *change of coordinates* $\varphi_\beta \circ \varphi_\alpha^{-1} : \mathbb{R}^d \to \mathbb{R}^d$ is *smooth* (i.e., of class C^∞) on its domain $\varphi_\alpha(\mathcal{U}_\alpha \cap \mathcal{U}_\beta)$.

We say that the elements of an atlas overlap smoothly.

Figure 1.1 illustrates the compatibility conditions between charts.

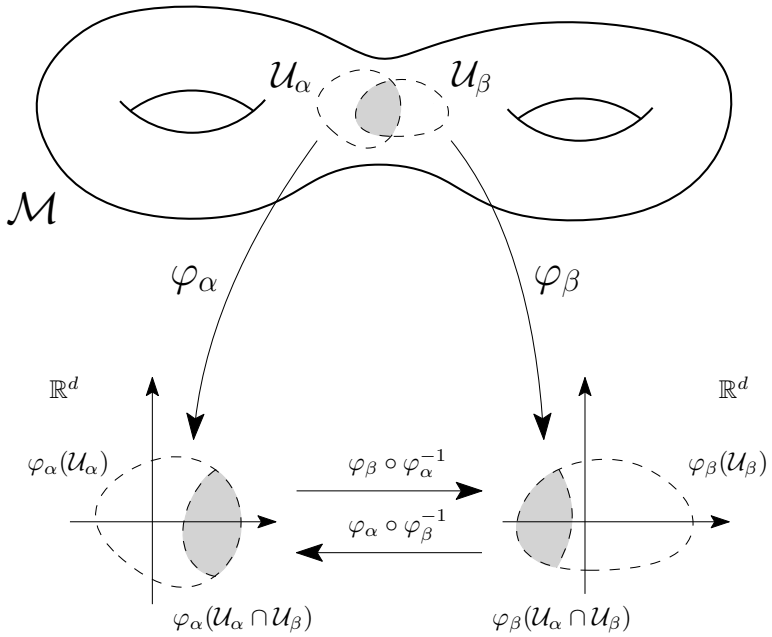


Figure 1.1 – Compatibility between charts.

Two atlases $\mathcal{A}_1$ and $\mathcal{A}_2$ are *equivalent* if $\mathcal{A}_1 \cup \mathcal{A}_2$ is an atlas.

Definition 1.2 (Maximal atlas)**.** Given $\mathcal{A}$, let $\mathcal{A}^+$ be the set of all charts $(\mathcal{U}, \varphi)$ such that $\mathcal{A} \cup \{(\mathcal{U}, \varphi)\}$ is also an atlas. $\mathcal{A}^+$ is called the *maximal atlas* generated by $\mathcal{A}$. The maximal atlas contains all the charts that one can have for the set $\mathcal{M}$. A maximal atlas of $\mathcal{M}$ is also called a *differentiable structure* on $\mathcal{M}$.

Now that we have defined the concept of maximal atlas, we can state a more rigorous definition of manifold.

Definition 1.3 (Manifold)**.** A (d-dimensional) *manifold* is a couple $(\mathcal{M}, \mathcal{A}^+)$, where $\mathcal{M}$ is a set and $\mathcal{A}^+$ is a maximal atlas of $\mathcal{M}$ into $\mathbb{R}^d$, such that the topology induced by $\mathcal{A}^+$ is Hausdorff and second-countable.

We will not go into the details of a Hausdorff topology here. For this thesis, it suffices to say that, roughly speaking, a topology is Hausdorff if disjoint points have disjoint neighborhoods.

A maximal atlas of a set $\mathcal{M}$ that induces a second-countable Hausdorff topology is called a *manifold structure*. In general, we call atlas of the manifold $(\mathcal{M}, \mathcal{A}^+)$ any atlas of $\mathcal{M}$ whose maximal atlas is $\mathcal{A}^+$. Similarly, a chart of the manifold $(\mathcal{M}, \mathcal{A}^+)$ is any chart of $\mathcal{M}$ that belongs to $\mathcal{A}^+$.

Now we turn our attention to some simple yet familiar examples of manifolds: vector spaces.

1.1.2 Vector spaces as manifolds

Let $\mathcal{E}$ be a d-dimensional vector space. Then, given a basis $(e_i)_{i=1,\dots,d}$ of $\mathcal{E}$, the function $\psi\colon \mathcal{E} \to \mathbb{R}^d$ defined by

$$x \mapsto \begin{bmatrix} x^1 \\ \vdots \\ x^d \end{bmatrix}, \quad \text{such that} \quad x = \sum_{i=1}^{d} x^i e_i,$$

is a chart of the set $\mathcal{E}$. In other words, every d-dimensional vector space is isomorphic to its space of coordinates $\mathbb{R}^d$. All charts built in this way are compatible, i.e., they satisfy point 2 in Definition 1.1. As a consequence, they form an atlas of $\mathcal{E}$, i.e., $\mathcal{E}$ has a manifold structure. Hence *every vector space is a linear manifold.* The linearity is implied by the linearity of the charts. As we will see later, the challenging case arises in the case of nonlinear manifolds, i.e., manifolds that are not endowed with a vector space structure.

We now look at some more concrete examples.

The manifold $\mathbb{R}^{n\times p}$. The set $\mathbb{R}^{n\times p}$ of $n \times p$ real matrices is a vector space. As such, it has a linear manifold structure. A chart on this manifold is $\varphi\colon \mathbb{R}^{n\times p} \to \mathbb{R}^{np}$ defined by $X \mapsto \operatorname{vec}(X)$, where $\operatorname{vec}(X)$ denotes the vector obtained by stacking the columns of X below one another. The manifold $\mathbb{R}^{n\times p}$ can be further turned into a Euclidean space with the inner product

$$\langle Z_1, Z_2 \rangle = \operatorname{vec}(Z_1)^{\mathsf{T}} \operatorname{vec}(Z_2) = \operatorname{trace}(Z_1^{\mathsf{T}} Z_2).$$

This inner product induces the Frobenius norm

$$\|Z\|_{\mathrm{F}} = \sqrt{\operatorname{trace}(Z^{\mathsf{T}} Z)},$$

which can be regarded as the Euclidean norm for matrices.

The manifold $\mathbb{R}_*^{n\times p}$. Let $\mathbb{R}_*^{n\times p}$ with $p \leqslant n$ be the set of all $n \times p$ matrices whose columns are linearly independent, i.e., matrices having full rank p. Observe that $\mathbb{R}_*^{n\times p}$ is an open subset of $\mathbb{R}^{n\times p}$ since its complement

$$\{X \in \mathbb{R}^{n\times p}\colon\ \det(X^{\mathsf{T}} X) = 0\}$$

is closed. The manifold $\mathbb{R}_*^{n\times p}$ is also known as the *noncompact Stiefel manifold* of full-rank $n \times p$ matrices. If $p = 1$ it corresponds to the Euclidean space $\mathbb{R}^n$ with the origin removed. If $p = n$ it becomes GL_n, the general linear group of all invertible $n \times n$ matrices.

We now go back to some more general theory.

1.1.3 Product manifolds

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two manifolds of dimensions d_1 and d_2, respectively. We define the *product manifold* $\mathcal{M}_1 \times \mathcal{M}_2$ whose elements are (x_1, x_2), with $x_1 \in \mathcal{M}_1$ and $x_2 \in \mathcal{M}_2$. Moreover, let $(\mathcal{U}_1, \varphi_1)$ be some chart of $\mathcal{M}_1$, and $(\mathcal{U}_2, \varphi_2)$ be some chart of $\mathcal{M}_2$. Then the mapping

$$\varphi_1 \times \varphi_2\colon \mathcal{U}_1 \times \mathcal{U}_2 \to \mathbb{R}^{d_1} \times \mathbb{R}^{d_2},$$

defined by

$$(x_1, x_2) \mapsto (\varphi_1(x_1), \varphi_2(x_2))$$

is a chart for the product manifold $\mathcal{M}_1 \times \mathcal{M}_2$. All the charts obtained in this way form an atlas for $\mathcal{M}_1 \times \mathcal{M}_2$. Thus $\mathcal{M}_1 \times \mathcal{M}_2$ is a product manifold with a topology equivalent to the product topology.

It is now time to introduce the first notions that allow us to perform calculus on manifolds.

1.1.4 Differentiable functions

In this section, we introduce the concept of differentiability for functions between manifolds. In the context of optimization algorithms on manifolds, mappings between manifolds occur in several situations:

- an optimization problem involves a *cost function*, which can be viewed as a mapping from manifold $\mathcal{M}$ to manifold $\mathbb{R}$;
- as *inclusions* in the theory of embedded submanifolds[1];
- as *retractions*, for instance in line-search methods on manifolds.

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two manifolds of dimensions d_1 and d_2, respectively, and let $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ be a mapping between these two manifolds. Let $x \in \mathcal{M}_1$, and let φ_1 and φ_2 be the charts that map $\mathcal{M}_1$ and $\mathcal{M}_2$ to $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, respectively. Observe that φ_1 is a chart around x, and φ_2 is a chart around $F(x)$. Thus, to go from $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_2}$ we have to "read through the charts", i.e., we have to use the map

$$\widehat{F} = \varphi_2 \circ F \circ \varphi_1^{-1},$$

which is a *coordinate representation of F around x*. The following diagram illustrates this concept.

$$\begin{array}{ccc} \mathcal{M}_1 & \xrightarrow{F} & \mathcal{M}_2 \\ {\scriptstyle\varphi_1}\downarrow & & \downarrow{\scriptstyle\varphi_2} \\ \mathbb{R}^{d_1} & \xrightarrow{\widehat{F}} & \mathbb{R}^{d_2} \end{array}$$

Definition 1.4 (Local smoothness of F at x)**.** We say that F is *differentiable* or *smooth at x* if $\widehat{F}$ is of class C^∞ at $\varphi_1(x)$.

We emphasize that this definition does not depend on the choice of the charts φ_1 and φ_2. Global smoothness of F is straightforward.

Definition 1.5 (Global smoothness of F)**.** We say that a function F is *smooth* if it is smooth for every x.

We are now ready to introduce the important notion of diffeomorphism, which can be regarded as a generalization of the concept of isomorphism to the case of smooth manifolds.

Definition 1.6 (Diffeomorphism)**.** A (smooth) *diffeomorphism* $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ is a bijection such that F and its inverse F^{-1} are both smooth. We say that two manifolds are *diffeomorphic* if there exists a diffeomorphism between them.

Let us introduce some more definitions about functions on manifolds which turn out to be useful in proving that certain sets are indeed smooth manifolds.

[1] Embedded submanifolds are defined later in Section 1.1.6.

1.1.4.1 Immersions and submersions

Let $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ be a differentiable function between two manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$, and let $x \in \mathcal{M}_1$ be a point of $\mathcal{M}_1$. Let $\mathrm{D}\widehat{F}(\varphi_1(x))\colon \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ be the *differential*[2] of $\widehat{F}$ at $\varphi_1(x) \in \mathbb{R}^{d_1}$, where $\widehat{F}$ denotes a coordinate representation of F, as defined above.

Definition 1.7 (Rank of a function)**.** The *rank* of F at x is the dimension of the image of the differential $\mathrm{D}\widehat{F}(\varphi_1(x))$.

As before, this definition does not depend on the charts either. We say that F is an *immersion* if its rank is equal to d_1 at each point of its domain (hence $d_1 \leqslant d_2$), and that F is a *submersion* if its rank is equal to d_2 at each point of its domain (hence $d_1 \geqslant d_2$). Equivalent characterizations are also possible as follows. We say that F is an immersion if and only if, around each point of its domain, it admits a coordinate representation $\widehat{F}\colon \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ that is the *canonical immersion*

$$(u^1, \dots, u^{d_1}) \mapsto (u^1, \dots, u^{d_1}, 0, \dots, 0), \qquad d_1 \leqslant d_2,$$

i.e., in the codomain $\mathbb{R}^{d_2}$ the last $d_2 - d_1$ coordinates are set to zero.

We say that F is a submersion if and only if, around each point of its domain, it admits a coordinate representation $\widehat{F}\colon \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ that is the *canonical submersion*

$$(u^1, \dots, u^{d_1}) \mapsto (u^1, \dots, u^{d_2}), \qquad d_1 \geqslant d_2,$$

i.e., in the codomain $\mathbb{R}^{d_2}$ the last $d_1 - d_2$ coordinates are neglected.

1.1.5 Matrix manifolds

A matrix manifold is any manifold that is constructed from $\mathbb{R}^{n\times p}$ by taking either embedded submanifolds or quotient manifolds. The two matrix manifolds that are object of study of this thesis are actually embedded submanifolds. Hence, in the next section, we will explore into reasonable detail embedded submanifolds.

The major matrix manifolds are

- noncompact Stiefel manifold;
- orthogonal Stiefel manifold;
- oblique manifold: $\{X \in \mathbb{R}^{n\times p}\colon \operatorname{diag}(X^{\mathsf{T}}X) = I_p\}$;
- generalized Stiefel manifold;
- manifold of symplectic matrices: $\{X \in \mathbb{R}^{2n\times 2n}\colon X^{\mathsf{T}}JX = J\}$, where

$$J = \begin{bmatrix} O_n & I_n \\ -I_n & O_n \end{bmatrix}.$$

[2] The differential of a mapping is discussed later in Section 1.1.8.

1.1.6 Embedded submanifolds

In general, a set $\mathcal{X}$ admits *more than one* manifold structure, but if it is a subset of a manifold $(\mathcal{M}, \mathcal{A}^+)$, then it admits *at most one* submanifold structure. In this case, the manifold $\mathcal{M}$ is referred to as the *embedding space*. The following proposition formalizes this result.

Proposition 1.8. *Let $\mathcal{N}$ be a subset of a manifold $\mathcal{M}$. Then $\mathcal{N}$ admits at most one differentiable structure that makes it an embedded submanifold of $\mathcal{M}$.*

In other words, it exists a *unique* differentiable structure that makes $\mathcal{N}$ an embedded submanifold. We emphasize that this proposition removes all the freedom of choice of a differentiable structure on $\mathcal{N}$.

When the embedding space is $\mathbb{R}^{n\times p}$, we say that $\mathcal{N}$ is a *matrix submanifold*. In this thesis, we deal with two particular cases of matrix submanifolds: the Stiefel manifold and the manifold of fixed-rank matrices.

How can we check if a subset $\mathcal{N} \subset \mathcal{M}$ is an embedded submanifold? Let us first introduce the notion of *coordinate slice*.

Definition 1.9 (Coordinate slice of dimension m)**.** Let $(\mathcal{U}, \varphi)$ be a chart of a manifold $\mathcal{M}$. A *φ-coordinate slice of $\mathcal{U}$ of dimension m* is a set of the form $\varphi^{-1}(\mathbb{R}^m \times \{0\})$, which corresponds to all the points of $\mathcal{U}$ whose last $n - m$ coordinates in the chart φ are equal to zero.

In other words, a coordinate slice is the image under φ^{-1} of the part of an m-dimensional plane in $\mathbb{R}^n$ which lies in the coordinate range.

The next proposition states that every embedded submanifold is locally a coordinate slice.

Proposition 1.10 (Submanifold property)**.** *A subset $\mathcal{N}$ of a manifold $\mathcal{M}$ is a d-dimensional embedded submanifold of $\mathcal{M}$ if and only if, around each point $x \in \mathcal{N}$, there exists a chart $(\mathcal{U}, \varphi)$ of $\mathcal{M}$ such that $\mathcal{N} \cap \mathcal{U}$ is a φ-coordinate slice of $\mathcal{U}$, i.e.,*

$$\mathcal{N} \cap \mathcal{U} = \{x \in \mathcal{U} \colon \varphi(x) \in \mathbb{R}^d \times \{0\}\}.$$

In this case, the chart $(\mathcal{N} \cap \mathcal{U}, \varphi)$ is a chart of the embedded submanifold $\mathcal{N}$.

What are the *sufficient conditions* for subsets of manifolds to be embedded submanifolds? We have the following two propositions.

Proposition 1.11 (Submersion theorem)**.** *Let $F \colon \mathcal{M}_1 \to \mathcal{M}_2$ (with $d_1 > d_2$). Let y be a point of $\mathcal{M}_2$. If the rank of F is equal to d_2 for every point of $F^{-1}(y)$, then $F^{-1}(y)$ is a closed embedded submanifold of $\mathcal{M}_1$, and $\dim(F^{-1}(y)) = d_1 - d_2$.*

Proposition 1.12 (Subimmersion theorem)**.** *Let $F \colon \mathcal{M}_1 \to \mathcal{M}_2$. Let y be a point of $F(\mathcal{M}_1)$. If F has constant rank $k < d_1$ in a neighborhood of $F^{-1}(y)$, then $F^{-1}(y)$ is a closed embedded submanifold of $\mathcal{M}_1$ of dimension $d_1 - k$.*

We now focus our attention on a concrete example of matrix submanifold.

1.1.6.1 The Stiefel manifold

The (orthogonal) Stiefel manifold is an embedded submanifold of $\mathbb{R}^{n\times p}$ that frequently arises in applications, and as such, it is the object of study of the first two parts of this thesis. It is defined as the set of all $n \times p$ orthonormal matrices

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n\times p} \colon X^\mathsf{T}X = I_p\},$$

where I_p denotes the $p \times p$ identity matrix. This set, endowed with its submanifold structure as discussed below, is called an orthogonal or compact Stiefel manifold. Let

$$X = \begin{bmatrix} | & | & & | \\ \boldsymbol{r_1} & \boldsymbol{r_2} & \cdots & \boldsymbol{r_p} \\ | & | & & | \end{bmatrix} \quad \text{and} \quad X^\mathsf{T} = \begin{bmatrix} — \boldsymbol{r_1^\mathsf{T}} — \\ — \boldsymbol{r_2^\mathsf{T}} — \\ \vdots \\ — \boldsymbol{r_p^\mathsf{T}} — \end{bmatrix},$$

where $\boldsymbol{r_i} \in \mathbb{R}^n$ are orthonormal vectors for all $i = 1, \ldots, p$. Then

$$X^\mathsf{T}X = \begin{bmatrix} \boldsymbol{r_1^\mathsf{T}r_1} & \boldsymbol{r_1^\mathsf{T}r_2} & \cdots & \boldsymbol{r_1^\mathsf{T}r_p} \\ \boldsymbol{r_2^\mathsf{T}r_1} & \boldsymbol{r_2^\mathsf{T}r_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \boldsymbol{r_p^\mathsf{T}r_1} & \cdots & \cdots & \boldsymbol{r_p^\mathsf{T}r_p} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}.$$

Clearly, $\mathrm{St}(n,p)$ is a subset of $\mathbb{R}^{n\times p}$, and we have seen above that $\mathbb{R}^{n\times p}$ admits a linear manifold structure. We are going to show that $\mathrm{St}(n,p)$ has indeed the structure of an embedded submanifold of $\mathbb{R}^{n\times p}$.

Proposition 1.13. *$\mathrm{St}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n\times p}$.*

Proof. To show that $\mathrm{St}(n,p)$ is an embedded submanifold of the manifold $\mathbb{R}^{n\times p}$, we can use the submersion theorem (Proposition 1.11). This means that we need to introduce a function F between two manifolds and show that it is a submersion. Here, we consider the two manifolds $\mathcal{M}_1 = \mathbb{R}^{n\times p}$ and $\mathcal{M}_2 = \mathcal{S}_{\mathrm{sym}}(p)$, where $\mathcal{S}_{\mathrm{sym}}(p)$ denotes the set of all $p \times p$ symmetric matrices, which is also a vector space, hence a linear manifold. As function F between these two manifolds, let us consider

$$F \colon \mathbb{R}^{n\times p} \to \mathcal{S}_{\mathrm{sym}}(p),$$

defined by

$$X \mapsto X^\mathsf{T}X - I_p.$$

Observe that $X^\mathsf{T}X - I_p$ is indeed a symmetric matrix. We point out that $\mathrm{St}(n,p)$ is the set of the inverse images of the null matrix under F, namely,

$$\mathrm{St}(n,p) = F^{-1}(O_p).$$

Here, O_p is the null matrix of size p-by-p, and it plays the role of the y in Proposition 1.11. We need to show that F is a submersion at each point X of $\mathrm{St}(n,p)$, i.e., that the differential of F maps *onto* $\mathcal{S}_{\mathrm{sym}}(p)$. The meaning of this is that the dimension of the image of $\mathrm{D}F$ is equal to $\dim(\mathcal{S}_{\mathrm{sym}}(p))$. This can formally be written as

$$\forall \widehat{Z} \in \mathcal{S}_{\mathrm{sym}}(p),\ \exists Z \in \mathbb{R}^{n\times p} \colon\ \mathrm{D}F(X)[Z] = \widehat{Z}.$$

To compute the differential of F, we can use the definition

$$F(X+Z) = F(X) + \mathrm{D}F(X)[Z] + o(\|Z\|).$$

Specializing it for the function $F(X) = X^\mathsf{T}X - I_p$, we get

$$(X^\mathsf{T} + Z^\mathsf{T})(X+Z) - I_p = X^\mathsf{T}X - I_p + X^\mathsf{T}Z + Z^\mathsf{T}X = F(X) + \mathrm{D}F(X)[Z],$$

from which we can identify $\mathrm{D}F(X)[Z] = X^\mathsf{T}Z + Z^\mathsf{T}X$. Now the question is the following: for any $\widehat{Z} \in \mathcal{S}_{\mathrm{sym}}(p)$, does there exist a $Z \in \mathbb{R}^{n\times p}$ such that $\mathrm{D}F(X)[Z] = \widehat{Z}$? The answer is yes, as can be checked by choosing $Z = \frac{1}{2}X\widehat{Z}$:

$$\mathrm{D}F(X)[Z] = X^\mathsf{T}Z + Z^\mathsf{T}X = \tfrac{1}{2}X^\mathsf{T}X\widehat{Z} + \tfrac{1}{2}\widehat{Z}^\mathsf{T}X^\mathsf{T}X = \widehat{Z}.$$

This shows that the rank of F, i.e., the dimension of the image of $\mathrm{D}F(X)[Z]$, is equal to $d_2 = \dim(\mathcal{S}_{\mathrm{sym}}(p))$ for every point of $F^{-1}(O_p)$. Then from Proposition 1.11 it follows that $\mathrm{St}(n,p) = F^{-1}(O_p)$ is an embedded submanifold of $\mathbb{R}^{n\times p}$. □

As a byproduct of the above proof, one can also obtain the dimension of $\mathrm{St}(n,p)$. Observe that the vector space $\mathcal{S}_{\mathrm{sym}}(p)$ has dimension $\frac{1}{2}p(p+1)$, since a symmetric matrix is completely determined by its upper triangular part. By Proposition 1.11, $\dim(\mathrm{St}(n,p)) = d_1 - d_2 = np - \frac{1}{2}p(p+1)$.

To conclude this section, let us state some basic properties of the Stiefel manifold $\mathrm{St}(n,p)$.

- It is *closed*, because it is the inverse image of the closed set $\{O_p\}$ under the continuous function $F\colon \mathbb{R}^{n\times p} \to \mathcal{S}_{\mathrm{sym}}(p)$.
- It is *bounded*; each column of $X \in \mathrm{St}(n,p)$ has norm 1, so the Frobenius norm of X is equal to $\sqrt{p}$.
- It is *compact*, since it is closed and bounded. This follows from the *Heine–Borel theorem* [AMS08, p. 193].

The Stiefel manifold $\mathrm{St}(n,p)$ may degenerate to some special cases. For $p = 1$, it reduces to the *unit sphere* S^{n-1} in $\mathbb{R}^n$. For $p = n$, the Stiefel manifold becomes the *orthogonal group* O_n, whose dimension is $\frac{1}{2}n(n-1)$.

1.1.7 Tangent vectors

Let us go back to some more general theory that is not restricted to matrix manifolds, and introduce some basic concepts of differential geometry that are used to generalize the notion of directional derivative to a real-valued function on a manifold.

Definition 1.14 (Curve in $\mathcal{M}$)**.** Let $\mathcal{M}$ be a manifold. A *curve in* $\mathcal{M}$ is a smooth mapping $\gamma\colon \mathbb{R} \to \mathcal{M}$, defined by $t \mapsto \gamma(t)$.

The derivative of the curve may be defined as

$$\gamma'(t) = \lim_{\tau\to 0} \frac{\gamma(t+\tau) - \gamma(t)}{\tau}. \tag{1.1}$$

However, we emphasize that the difference $\gamma(t+\tau) - \gamma(t)$ requires a vector space structure in order to make sense, thus this definition fails for an abstract nonlinear manifold. Nonetheless,

given a smooth function on a manifold $f\colon \mathcal{M} \to \mathbb{R}$, the function $f \circ \gamma\colon t \mapsto f(\gamma(t))$ is a smooth function from $\mathbb{R}$ to $\mathbb{R}$, with a well-defined classical derivative.

Let $\mathcal{F}_x(\mathcal{M})$ denote the set of smooth, real-valued functions defined on a neighborhood of a point $x \in \mathcal{M}$. The *tangent vector* to the curve γ at $t = 0$ is defined as the mapping $\dot{\gamma}(0)\colon \mathcal{F}_x(\mathcal{M}) \to \mathbb{R}$, that maps a function into a scalar $f \mapsto \dot{\gamma}(0)f$, where

$$\dot{\gamma}(0)f = \left.\frac{\mathrm{d}(f(\gamma(t)))}{\mathrm{d}t}\right|_{t=0}.$$

We emphasize that the tangent vector is a mapping, so it is not a vector as in the sense of classical geometry. Nonetheless, to preserve our intuition and for illustration purposes, we will often depict it as an arrow in a two- or three-dimensional space.

When $\mathcal{M}$ is (a submanifold of) a vector space $\mathcal{E}$, the mapping $\dot{\gamma}(0)$ and the derivative of a curve (1.1) are closely related by

$$\dot{\gamma}(0)f = \mathrm{D}f(\gamma(0))[\gamma'(0)].$$

A formal definition with a slightly different notation is the following.

Definition 1.15 (Tangent vector)**.** A *tangent vector* ξ_x to a manifold $\mathcal{M}$ at a point x is a mapping from $\mathcal{F}_x(\mathcal{M})$ to $\mathbb{R}$, such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = x$, satisfying

$$\xi_x f = \dot{\gamma}(0)f = \left.\frac{\mathrm{d}(f(\gamma(t)))}{\mathrm{d}t}\right|_{t=0},$$

for all $f \in \mathcal{F}_x(\mathcal{M})$. Such a curve γ is said to realize the tangent vector ξ_x.

The notion of tangent vector allows us to introduce another very important concept.

Definition 1.16 (Tangent space)**.** The *tangent space* to $\mathcal{M}$ at x, denoted $T_x\mathcal{M}$, is the set of all tangent vectors to $\mathcal{M}$ at x.

The crucial observation here is that the tangent space admits a vector space structure. Indeed, for two tangent vectors to $\dot{\gamma}_1(0)$ and $\dot{\gamma}_2(0)$ to $\mathcal{M}$ at x, the linearity property holds, since

$$(a\dot{\gamma}_1(0) + b\dot{\gamma}_2(0))f = a(\dot{\gamma}_1(0)f) + b(\dot{\gamma}_2(0)f),$$

and $(a\dot{\gamma}_1(0) + b\dot{\gamma}_2(0))$ is still a tangent vector. The fact that $T_x\mathcal{M}$ is a vector space is very important, since it provides a local vector space approximation to the manifold.

Later, in Section 1.2.1, we will define *retractions*, i.e., mappings from $T_x\mathcal{M}$ to $\mathcal{M}$, which can be used to locally transform an optimization problem on the manifold $\mathcal{M}$ into an optimization problem on the more friendly vector space $T_x\mathcal{M}$.

Remark 1.17. Observe that the tangent space $T_x\mathcal{M}$ has the same dimension d as the manifold $\mathcal{M}$, i.e., $\dim(T_x\mathcal{M}) = \dim(\mathcal{M})$.

This can be shown by using a coordinate chart. Let $(\mathcal{U}, \varphi)$ be a coordinate chart at x. A basis of $T_x\mathcal{M}$ is given by $(\dot{\gamma}_1(0), \ldots, \dot{\gamma}_d(0))$, with $\gamma_i(t) = \varphi^{-1}(\varphi(x) + te_i)$, where e_i denotes the ith canonical vector of $\mathbb{R}^d$. The tangent vectors $\dot{\gamma}_i(0)$ are defined as

$$\dot{\gamma}_i(0)f = \partial_i(f \circ \varphi^{-1})(\varphi(x)),$$

where ∂_i denotes the standard partial derivative with respect to the ith component. Finally, for any tangent vector $\dot\gamma(0)$ we have the decomposition

$$\dot\gamma(0) = \sum_{i=1}^{d} (\dot\gamma(0)\varphi_i)\, \dot\gamma_i(0),$$

where $\dot\gamma(0)\varphi_i$ are the coordinates of the tangent vector in $\mathbb{R}^d$. Figure 1.2 illustrates for a two-dimensional manifold the construction that we have just outlined.

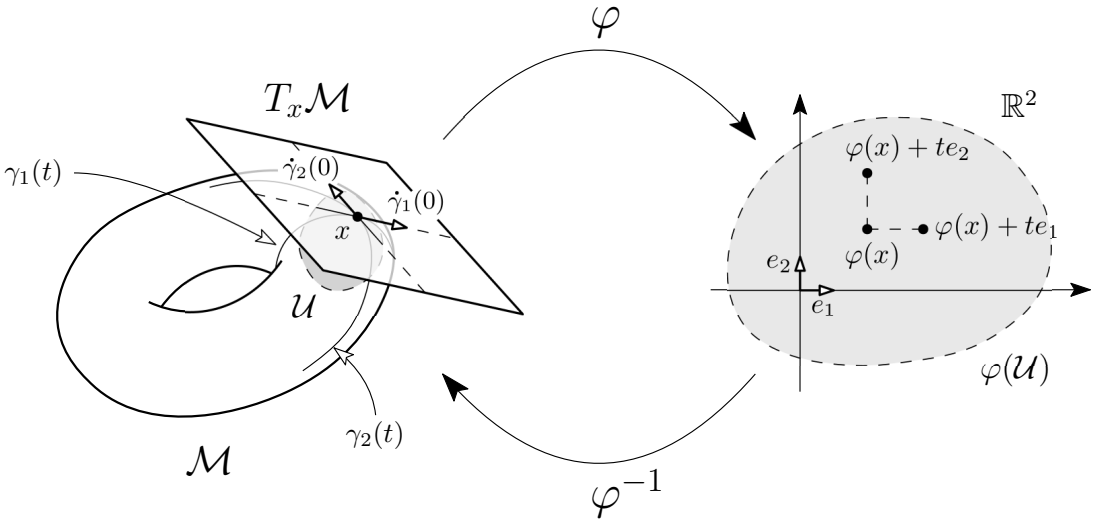


Figure 1.2 – The tangent space $T_x\mathcal{M}$ has the same dimension as the manifold $\mathcal{M}$.

1.1.7.1 Tangent vectors to a vector space

Let $\mathcal{E}$ be a vector space. We have seen above that a tangent vector ξ_x to $\mathcal{E}$ at x is a mapping $\xi_x \colon \mathcal{F}_x(\mathcal{E}) \to \mathbb{R}$, defined by

$$f \mapsto \xi_x f = \left.\frac{\mathrm{d}(f(\gamma(t)))}{\mathrm{d}t}\right|_{t=0},$$

where γ is a curve in $\mathcal{E}$ with $\gamma(0) = x$. The directional derivative of f at x along $\gamma'(0)$ coincides with the classical derivative of $f(\gamma(t))$ evaluated at $t = 0$, i.e.,

$$\xi_x f = \mathrm{D}f(x)[\gamma'(0)].$$

Moreover, $T_x\mathcal{E}$ is identified with $\mathcal{E}$ itself, i.e., $T_x\mathcal{E} \simeq \mathcal{E}$.

1.1.7.2 Tangent bundle

The *tangent bundle* is the set of all tangent vectors to $\mathcal{M}$, i.e., the union of all the tangent spaces to $\mathcal{M}$:

$$T\mathcal{M} = \bigcup_{x\in\mathcal{M}} T_x\mathcal{M}.$$

Since every $\xi \in T\mathcal{M}$ is in one and only one tangent space $T_x\mathcal{M}$, it follows that $\mathcal{M}$ is a quotient of $T\mathcal{M}$, with natural projection

$$\pi \colon T\mathcal{M} \to \mathcal{M},$$

defined by $\xi \in T_x\mathcal{M} \mapsto x$. This gives us the following different perspective. We can regard each x as the representative element of all $\xi \in T_x\mathcal{M}$, so that $\mathcal{M}$ can be viewed the set of all representative elements, i.e., a *quotient set*. It can be shown that the tangent bundle $T\mathcal{M}$ has a natural manifold structure.

1.1.7.3 Vector fields

A vector field ξ is a smooth function from the manifold to the tangent bundle $\mathcal{M} \to T\mathcal{M}$, defined by $x \mapsto \xi_x$. Hence, a vector field ξ assigns to each point $x \in \mathcal{M}$ a tangent vector $\xi_x \in T_x\mathcal{M}$.

Given ξ a vector field on $\mathcal{M}$, and $f \in \mathcal{F}(\mathcal{M})$, we define

$$(\xi f)_{(x)} = \xi_x(f).$$

Here, ξf denotes the real-valued function that maps x into $\xi_x(f)$, the tangent vector to $\mathcal{M}$ at x applied to f. Compare with the multiplication of a vector field by a function, which is defined as

$$(f\xi)_x = f(x)\xi_x, \qquad \forall x \in \mathcal{M},$$

and the addition of two vector fields, which is

$$(\xi + \zeta)_x = \xi_x + \zeta_x, \qquad \forall x \in \mathcal{M}.$$

Definition 1.18 (The coordinate vector field)**.** The vector field E_i on $\mathcal{U}$, defined by

$$(E_i f)_{(x)} = \partial_i(f \circ \varphi^{-1})(\varphi(x)) = \mathrm{D}(f \circ \varphi^{-1})(\varphi(x))[e_i],$$

is called the ith *coordinate vector field* of $(\mathcal{U}, \varphi)$.

Any vector field ξ admits a decomposition $\xi = \sum_i (\xi\varphi_i)E_i$ on $\mathcal{U}$, where $\xi\varphi_i$ is the function that gives the tangent vector applied to φ at x.

If the manifold is an n-dimensional vector space $\mathcal{E}$, then the coordinate vector field becomes

$$(E_i f)_{(x)} = \partial_i f(x) = \lim_{t \to 0} \frac{f(x + te_i) - f(x)}{t} = \mathrm{D}f(x)[e_i],$$

i.e., we do not need to "read through the charts".

We are now ready to introduce another fundamental concept that makes it possible to relate the tangent spaces to two different manifolds.

1.1.8 Differential of a mapping

The differential of a mapping is a function that maps a tangent vector to a manifold $\mathcal{M}$ into a tangent vector to another manifold $\mathcal{N}$. Let $F\colon \mathcal{M} \to \mathcal{N}$ be a smooth mapping between two manifolds. Recall that ξ_x is a smooth mapping from $\mathcal{F}_x(\mathcal{M})$ to $\mathbb{R}$. The mapping

$$\mathrm{D}F(x)[\xi_x]\colon \mathcal{F}_{F(x)}(\mathcal{N}) \to \mathbb{R},$$

defined by

$$(\mathrm{D}F(x)[\xi])f = \xi(f \circ F),$$

is a tangent vector to $\mathcal{N}$ at $F(x)$. Here, $\mathcal{F}_{F(x)}(\mathcal{N})$ denotes the set of smooth real-valued functions defined on a neighborhood of $F(x)$, and $\xi(f \circ F)$ is the tangent vector applied to the composite function $f \circ F$.

The mapping

$$\mathrm{D}F(x)\colon T_x\mathcal{M} \to T_{F(x)}\mathcal{N},$$

defined by

$$\xi \mapsto \mathrm{D}F(x)[\xi],$$

is a *linear mapping* called the *differential of F at x*. Figure 1.3 illustrates the notion of the differential.

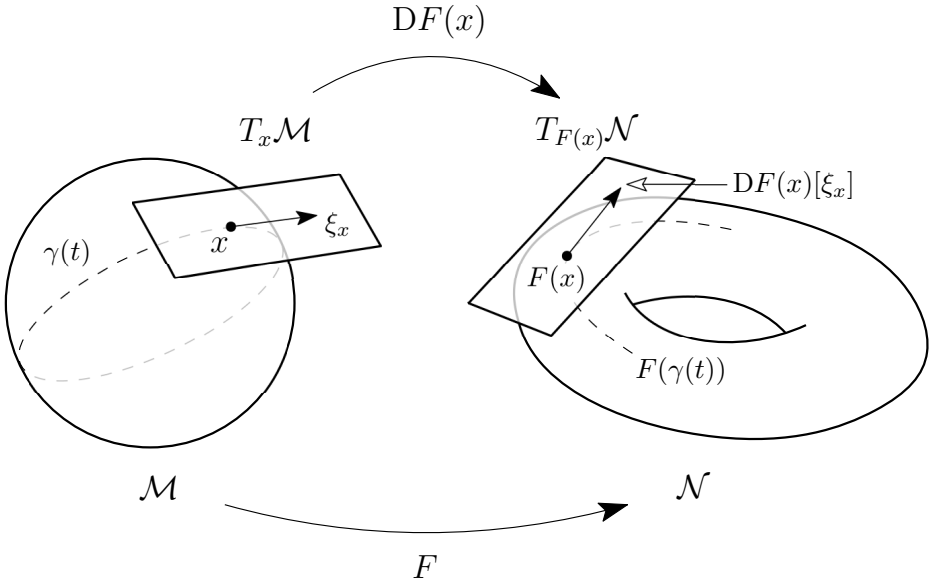


Figure 1.3 – The differential map of F at x.

Remark 1.19. F is an immersion if and only if $\mathrm{D}F(x)$ is an injection, for all $x \in \mathcal{M}$. F is a submersion if and only if $\mathrm{D}F(x)$ is a surjection, i.e., if $\mathrm{rank}(\mathrm{D}F(x)) = d_2$, for all $x \in \mathcal{M}$.

As we will show in the following section, the differential is useful for characterizing tangent spaces to embedded submanifolds.

1.1.9 Tangent spaces to embedded submanifolds

Let $\mathcal{E}$ be a vector space and let $\mathcal{M}$ be an embedded submanifold of $\mathcal{E}$. Let γ be a curve in $\mathcal{M}$. Since γ is a curve in $\mathcal{M}$, it also induces a tangent vector $\dot{\gamma}(0) \in T_x\mathcal{M}$. The relationship between $\gamma'(0)$ and $\dot{\gamma}(0)$ is given by $\dot{\gamma}(0)f = \mathrm{D}f(x)[\gamma'(0)]$. One can identify $T_x\mathcal{M}$ with the set $\{\gamma'(0)\colon \gamma \text{ curve in } \mathcal{M}, \gamma(0) = x\}$, which is a linear subspace of the vector space $T_x\mathcal{E} \simeq \mathcal{E}$.

If $\mathcal{M}$ is a *matrix submanifold*, i.e., $\mathcal{E} = \mathbb{R}^{n\times p}$, we have $T_x\mathcal{E} = \mathbb{R}^{n\times p}$, hence the tangent vectors to $\mathcal{M}$ are represented by $n \times p$ matrices.

The following remark is very important for a practical characterization of tangent spaces to embedded submanifolds.

Remark 1.20 (Characterization of tangent spaces to embedded submanifolds)**.** Let $\mathcal{M}$ be an embedded submanifold of $\mathcal{E}$. Let $F\colon \mathcal{E} \to \mathcal{M}$. The tangent vectors to $\mathcal{M}$ at x correspond to those vectors ξ that satisfy $\mathrm{D}F(x)[\xi] = 0$. Thus $T_x\mathcal{M}$ is the kernel of the linear operator $\mathrm{D}F(x)$

$$T_x\mathcal{M} = \ker(\mathrm{D}F(x)).$$

Example 1.21 (Tangent space on a sphere)**.** Let $t \mapsto x(t)$ be a curve in the unit sphere S^{n-1} through x_0 at $t = 0$. Since $x(t) \in S^{n-1}$ for all t, we have

$$x(t)^\mathsf{T} x(t) = 1.$$

Differentiating with respect to t, we get

$$\dot{x}^\mathsf{T} x + x^\mathsf{T} \dot{x} = 0.$$

For $t = 0$, this becomes

$$\dot{x}_0^\mathsf{T} x_0 + x_0^\mathsf{T} \dot{x}_0 = 0,$$

$\dot{x}_0$ being the tangent vector to S^{n-1} at x_0. The last equation represents the kernel of the differential operator of $t \mapsto x(t)$. This shows that $\dot{x}_0$ is an element of the set

$$\{z \in \mathbb{R}^n \colon x_0^\mathsf{T} z = 0\}.$$

The tangent space to S^{n-1} at x is the set of all vectors orthogonal to x in $\mathbb{R}^n$, i.e.,

$$T_x S^{n-1} = \{z \in \mathbb{R}^n \colon x^\mathsf{T} z = 0\}.$$

Example 1.22 (Tangent space on the orthogonal Stiefel manifold)**.** The orthogonal Stiefel manifold

$$\mathrm{St}(n, p) = \{X \in \mathbb{R}^{n \times p} \colon X^\mathsf{T} X = I_p\}$$

is an embedded submanifold of Euclidean space $\mathbb{R}^{n \times p}$ (see Section 1.1.6.1). Let $t \mapsto X(t)$ be a curve in $\mathrm{St}(n, p)$ through X_0 at $t = 0$, i.e., $X(t) \in \mathbb{R}^{n \times p}$, $X(0) = X_0$, and $X(t)^\mathsf{T} X(t) = I_p$ for all t. Differentiating with respect to t, we get

$$\dot{X}(t)^\mathsf{T} X(t) + X(t)^\mathsf{T} \dot{X}(t) = 0.$$

For $t = 0$, this becomes

$$\dot{X}_0^\mathsf{T} X_0 + X_0 \dot{X}_0 = 0,$$

$\dot{X}_0$ being the tangent vector to $\mathrm{St}(n, p)$ at X_0. We deduce that $\dot{X}_0$ belongs to the set

$$\{Z \in \mathbb{R}^{n \times p} \colon X_0^\mathsf{T} Z + Z^\mathsf{T} X_0 = 0\}. \tag{1.2}$$

We can recognize in $\mathrm{D}F(X_0)[Z]$ the expression $X_0^\mathsf{T} Z + Z^\mathsf{T} X_0 = 0$, thus (1.2) is the kernel of $\mathrm{D}F(X_0)$, with $F \colon X \mapsto X^\mathsf{T} X$. Hence the tangent space is

$$T_X \mathrm{St}(n, p) = \{Z \in \mathbb{R}^{n \times p} \colon X^\mathsf{T} Z + Z^\mathsf{T} X = 0\}.$$

An alternative way to characterize the tangent space $T_X \mathrm{St}(n, p)$ is the following. Let $X_\perp$ be an orthonormal matrix whose columns span the orthogonal complement of $\mathrm{span}(X)$. Since X is orthonormal, together with $X_\perp$ one can form an orthonormal basis of the space $\mathbb{R}^{n \times p}$, and we can decompose any tangent vector $\dot{X}$ on this basis as

$$\dot{X} = X\Omega + X_\perp K,$$

Ω being a p-by-p skew-symmetric matrix, $\Omega \in \mathcal{S}_{\mathrm{skew}}(p)$, and $K \in \mathbb{R}^{(n-p) \times p}$, with no restriction on K. So the tangent space to the Stiefel manifold can also be characterized by

$$T_X \mathrm{St}(n, p) = \{X\Omega + X_\perp K \colon \Omega = -\Omega^\mathsf{T},\ K \in \mathbb{R}^{(n-p) \times p}\}.$$

With this characterization in mind, and with the fact that $\dim(\mathrm{St}(n,p)) = \dim(T_X\mathrm{St}(n,p))$, it is straightforward to work out the dimension of the Stiefel manifold as

$$\dim(\mathrm{St}(n,p)) = \dim(\mathcal{S}_{\mathrm{skew}}) + \dim(\mathbb{R}^{(n-p)\times p}) = \tfrac{1}{2}p(p-1) + (n-p)p = np - \tfrac{1}{2}p(p+1),$$

which verifies the result obtained in Section 1.1.6.1.

As we mentioned before, if $p = n$ then the Stiefel manifold reduces to the special case of the orthogonal group

$$O_n = \{X \in \mathbb{R}^{n\times n} \colon X^{\mathsf{T}}X = I_n\},$$

and the tangent space at X is given by

$$T_X O_n = \{X\Omega \colon \Omega^{\mathsf{T}} = -\Omega\} = X\mathcal{S}_{\mathrm{skew}}(n). \tag{1.3}$$

In particular, if $X = I_n$, we have $T_{I_n}O_n = \mathcal{S}_{\mathrm{skew}}(n)$. This means that the tangent space to O_n at the identity matrix I_n is the set of skew-symmetric n-by-n matrices $\mathcal{S}_{\mathrm{skew}}(n)$. In the language of Lie groups, we say that $\mathcal{S}_{\mathrm{skew}}(n)$ is the *Lie algebra* of the Lie group O_n.

We now go back to some more general theory.

1.1.10 Riemannian metric, distance and gradients

We have seen in the previous sections that tangent vectors generalize to manifolds the notion of directional derivative. In order to generalize the steepest descent method to nonlinear manifolds, we still need a notion of *length* that applies to tangent vectors, in order to understand which direction from x gives the *steepest increase*.

To this aim, we endow $T_x\mathcal{M}$ with an *inner product* $\langle\cdot,\cdot\rangle_x$, i.e., a bilinear, symmetric positive definite form. The subscript x in $\langle\cdot,\cdot\rangle_x$ indicates that in general the inner product depends on the point $x \in \mathcal{M}$. The inner product $\langle\cdot,\cdot\rangle_x$ induces a norm $\|\xi_x\|_x = \sqrt{\langle\xi_x,\xi_x\rangle_x}$ on $T_x\mathcal{M}$. The *normalized direction of steepest ascent* is then given by

$$\operatorname*{arg\,max}_{\xi_x \in T_x\mathcal{M}\colon \|\xi_x\|=1} \mathrm{D}f(x)[\xi_x].$$

Most importantly, the introduction of the inner product structure permits to define the notion of *Riemannian manifold*.

Definition 1.23 (Riemannian manifold)**.** A manifold $\mathcal{M}$ endowed with a *smoothly-varying inner product* (called *Riemannian metric*[3] g) is called Riemannian manifold.

Strictly speaking, a Riemannian manifold is a couple $(\mathcal{M}, g)$, i.e., a manifold with a Riemannian metric on it.

Remark 1.24. A *vector space* endowed with an inner product structure is a particular case of Riemannian manifold called *Euclidean space*.

Definition 1.25 (Length of a curve)**.** The *length of a curve* $\gamma\colon [a,b] \to \mathcal{M}$ on a Riemannian manifold $(\mathcal{M}, g)$ is

$$L(\gamma) = \int_a^b \sqrt{g(\dot\gamma(t),\dot\gamma(t))}\,\mathrm{d}t.$$

[3] Riemannian inner product would be a more appropriate term in order to avoid confusion with a metric in the standard sense, but the original terminology stuck.

Definition 1.26 (Riemannian distance)**.** The *Riemannian distance* is defined as the shortest path between two points x and y

$$\operatorname{dist}\colon \mathcal{M} \times \mathcal{M} \to \mathbb{R}\colon \operatorname{dist}(x, y) = \inf_{\Gamma} L(\gamma),$$

where Γ denotes the set of all curves γ in $\mathcal{M}$ joining points x and y.

Assuming that $\mathcal{M}$ is Hausdorff (see Definition 1.3), the Riemannian distance defines a *metric* in the standard sense, i.e.,

- it is nonnegative: $\operatorname{dist}(x, y) \geqslant 0$;
- it is symmetric: $\operatorname{dist}(x, y) = \operatorname{dist}(y, x)$;
- it satisfies the triangular inequality: $\operatorname{dist}(x, z) + \operatorname{dist}(z, y) \geqslant \operatorname{dist}(x, y)$.

Definition 1.27 (Riemannian gradient)**.** Let f be a smooth scalar field on a Riemannian manifold $\mathcal{M}$. The *Riemannian gradient* of f at x, denoted $\operatorname{grad} f(x)$, is the *unique* element of $T_x\mathcal{M}$ such that

$$\langle \operatorname{grad} f(x), \xi \rangle_x = \mathrm{D}f(x)[\xi], \qquad \forall \xi \in T_x\mathcal{M}.$$

We point out that $\mathrm{D}f(x)[\cdot]$ is an element of the dual space of $T_x\mathcal{M}$, i.e., the space of all linear functionals from $T_x\mathcal{M}$ to $\mathbb{R}$. The gradient $\operatorname{grad} f(x)$ always exists because of Riesz representation theorem, which states that *every* element of the dual space can be written *uniquely* in the above form.

The Riemannian gradient has some remarkable properties that turn out to be very useful in the context of optimization.

- The direction of $\operatorname{grad} f(x)$ is the steepest-ascent direction of f at x, namely

$$\frac{\operatorname{grad} f(x)}{\| \operatorname{grad} f(x) \|} = \operatorname*{arg\,max}_{\xi \in T_x\mathcal{M}\colon \|\xi\| = 1} \mathrm{D}f(x)[\xi].$$

- The norm of $\operatorname{grad} f(x)$ gives the steepest slope of f at x, i.e.,

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

1.1.11 Riemannian submanifolds

Let $\mathcal{M}$ be an embedded submanifold of a Riemannian manifold $\overline{\mathcal{M}}$. Since $\mathcal{M}$ is a submanifold, it can inherit the Riemannian metric from its embedding space $\overline{\mathcal{M}}$

$$g_x(\xi, \zeta) = \bar{g}_x(\xi, \zeta), \qquad \xi, \zeta \in T_x\mathcal{M}.$$

Definition 1.28 (Normal space)**.** The orthogonal complement of $T_x\mathcal{M}$ in $T_x\overline{\mathcal{M}}$ is called *normal space* to $\mathcal{M}$ at x and it is defined by

$$(T_x\mathcal{M})^{\perp} = \left\{ \xi \in T_x\overline{\mathcal{M}}\colon \bar{g}_x(\xi, \zeta) = 0, \quad \forall \zeta \in T_x\mathcal{M} \right\}.$$

Any tangent vector $\xi \in T_x\overline{\mathcal{M}}$ can be uniquely decomposed into

$$\xi = \mathrm{P}_x\, \xi + \mathrm{P}_x^{\perp}\, \xi,$$

where P_x and $\mathrm{P}_x^{\perp}$ denote the orthogonal projections onto $T_x\mathcal{M}$ and $(T_x\mathcal{M})^{\perp}$, respectively.

Example 1.29 (Sphere)**.** The unit sphere S^{n-1} is a Riemannian submanifold of $\mathbb{R}^n$. The inner product on the sphere is inherited from the embedding space $\mathbb{R}^n$

$$\langle \xi, \eta \rangle_x = \xi^\mathsf{T} \eta.$$

The normal space at $x \in S^{n-1}$ is

$$(T_x S^{n-1})^\perp = \{x\alpha \colon \alpha \in \mathbb{R}\}.$$

The projections are given by

$$\mathrm{P}_x \, \xi = (I - xx^\mathsf{T}) \, \xi, \qquad \mathrm{P}_x^\perp \, \xi = xx^\mathsf{T} \xi.$$

Example 1.30 (Orthogonal Stiefel manifold)**.** We recall that the tangent space to $\mathrm{St}(n,p)$ at X is given by

$$T_X \mathrm{St}(n,p) = \{X\Omega + X_\perp K \colon \Omega = -\Omega^\mathsf{T},\ K \in \mathbb{R}^{(n-p)\times p}\}.$$

The Riemannian metric inherited by $T_X \mathrm{St}(n,p)$ from the embedding space $\mathbb{R}^{n\times p}$ is

$$\langle \xi, \eta \rangle_X = \mathrm{trace}(\xi^\mathsf{T} \eta).$$

The normal space is given by those matrices A such that

$$\langle \xi, A \rangle_X = 0, \quad \forall \xi \in T_X \mathrm{St}(n,p).$$

Take A in the form $A = XS$, with $X \in \mathrm{St}(n,p)$ and S a p-by-p symmetric matrix, $S \in \mathcal{S}_{\mathrm{sym}}(p)$. Then one can easily verify that

$$\langle \xi, A \rangle_X = \mathrm{trace}(\xi^\mathsf{T} A) = \mathrm{trace}((\Omega_\xi^\mathsf{T} X^\mathsf{T} + K_\xi^\mathsf{T} X_\perp^\mathsf{T}) \, XS) = \mathrm{trace}(\Omega_\xi^\mathsf{T} S) = 0.$$

Thus the normal space is given by

$$(T_X \mathrm{St}(n,p))^\perp = \{XS \colon S \in \mathcal{S}_{\mathrm{sym}}(p)\}.$$

The projection onto the tangent space $T_X \mathrm{St}(n,p)$ is

$$\mathrm{P}_X \, \xi = X \mathrm{skew}(X^\mathsf{T} \xi) + (I - XX^\mathsf{T}) \, \xi,$$

and the projection onto the normal space $(T_X \mathrm{St}(n,p))^\perp$ is

$$\mathrm{P}_X^\perp \, \xi = X \mathrm{sym}(X^\mathsf{T} \xi).$$

To prove these expressions for the projectors, we start by writing a generic tangent vector as the sum of all its projections

$$\xi = \mathrm{P}_X \, \xi + \mathrm{P}_X^\perp \, \xi = X\Omega + X_\perp K + XS.$$

Left-multiplying by X^T we get

$$X^\mathsf{T} \xi = \Omega + S,$$

and taking the transpose

$$\xi^\mathsf{T} X = -\Omega + S.$$

The symmetric and the skew-symmetric parts of $X^\mathsf{T}\xi$ are

$$\operatorname{sym}(X^\mathsf{T}\xi) = \frac{X^\mathsf{T}\xi + \xi^\mathsf{T}X}{2} = S, \qquad \operatorname{skew}(X^\mathsf{T}\xi) = \frac{X^\mathsf{T}\xi - \xi^\mathsf{T}X}{2} = \Omega.$$

Now we only need to find $X_\perp K$:

$$X_\perp K = \xi - X\Omega - XS = \xi - X\,\frac{X^\mathsf{T}\xi - \xi^\mathsf{T}X}{2} - X\,\frac{X^\mathsf{T}\xi + \xi^\mathsf{T}X}{2} = (I - XX^\mathsf{T})\,\xi.$$

Finally,

$$\xi = X\operatorname{skew}(X^\mathsf{T}\xi) + (I - XX^\mathsf{T})\,\xi + X\operatorname{sym}(X^\mathsf{T}\xi).$$

From the last expression one can identify the expressions for the projectors.

The main concepts of first-order Riemannian geometry have been introduced. We are now ready to discuss the first numerical algorithms on manifolds.

1.2 Line-search algorithms on manifolds

Line-search algorithms in $\mathbb{R}^n$ are based on the update formula

$$x_{k+1} = x_k + t_k\eta_k,$$

where $t_k \in \mathbb{R}$ is the step size and $\eta_k \in \mathbb{R}^n$ is the search direction. We want to develop an analogous formula and theory for optimization problems posed on nonlinear manifolds. We can identify the following main aspects that we need to consider in order to generalize line-search algorithms to manifolds:

- η_k will be a tangent vector to $\mathcal{M}$ at x_k, i.e., $\eta_k \in T_{x_k}\mathcal{M}$;
- the search is performed *along a curve in* $\mathcal{M}$ whose tangent vector at $t = 0$ is η_k.

The choice of such a curve leads us to the concept of retraction.

1.2.1 Retractions

In a line-search algorithm, given a point x, we compute $\eta = -\operatorname{grad} f(x)$ and then we move in the direction of η until a reasonable decrease is found, which is often defined as the sufficient decrease condition. In $\mathbb{R}^n$ the implementation of this idea is straightforward. On a manifold, we need to move in the direction of a tangent vector while remaining constrained to the manifold. In order to do so, we introduce the concept of a *retraction mapping*. Roughly speaking, a retraction R at x, denoted R_x, is a mapping from $T_x\mathcal{M}$ to $\mathcal{M}$ with a local rigidity condition that preserves gradients at x. Figure 1.4 illustrates the concept of retraction.

The Riemannian exponential mapping is also a retraction, but it is not computationally efficient. Indeed, retractions are a first-order approximation of the Riemannian exponential, and that is what makes them cheaper to compute in practical applications.

Hereafter we give a more formal definition.

Definition 1.31 (Retraction)**.** A retraction on $\mathcal{M}$ is a *smooth mapping* from the tangent bundle to the manifold, $R\colon T\mathcal{M} \to \mathcal{M}$, with the following properties:

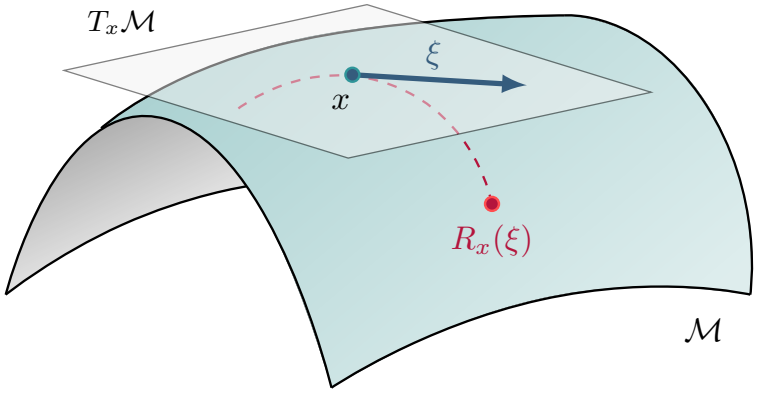


Figure 1.4 – Retraction mapping.

(i) $R_x(0_x) = x$, where R_x denotes the restriction of R to $T_x\mathcal{M}$ and 0_x is the zero element of $T_x\mathcal{M}$.

(ii) With the identification $T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}$, R_x satisfies the *local rigidity condition*

$$\mathrm{D}R_x(0_x) = \mathrm{id}_{T_x\mathcal{M}}.$$

We point out that R_x maps from $T_x\mathcal{M}$ to $\mathcal{M}$, in particular it maps 0_x to x. Moreover, we recall that the differential of a function between two manifolds is a mapping between their corresponding tangent spaces. Hence the differential of R_x at 0_x is the mapping

$$\mathrm{D}R_x(0_x)\colon T_{0_x}T_x\mathcal{M} \to T_x\mathcal{M},$$

but since $T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}$, we actually recognize in $\mathrm{D}R_x(0_x)$ the identity map of $T_x\mathcal{M}$, denoted $\mathrm{id}_{T_x\mathcal{M}}$. The identification $T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}$ holds because $T_x\mathcal{M}$ is a vector space.

Remark 1.32. For any $\xi \in T_x\mathcal{M}$, the curve $\gamma_\xi\colon t \mapsto R_x(t\xi)$ satisfies $\dot{\gamma}_\xi(0) = \xi$.

In the context of optimization algorithms, retractions have two main purposes:

- they turn points of $T_x\mathcal{M}$ into points of $\mathcal{M}$;
- they transform cost functions defined in a neighborhood of $x \in \mathcal{M}$ into cost functions defined on the vector space $T_x\mathcal{M}$.

Given a real-valued function $f\colon \mathcal{M} \to \mathbb{R}$, and a retraction $R\colon T\mathcal{M} \to \mathcal{M}$, one can define the "pullback" of f through R as $\hat{f} = f \circ R$. For $x \in \mathcal{M}$, the restriction of $\hat{f}$ to $T_x\mathcal{M}$ is $\hat{f}_x = f \circ R_x$, $\hat{f}_x\colon T_x\mathcal{M} \to \mathbb{R}$. We have the equality between the differentials

$$\mathrm{D}\hat{f}_x(0_x) = \mathrm{D}f(x).$$

Indeed, using the chain rule for the differential of a composite function

$$\mathrm{D}(f \circ g)(x) = \mathrm{D}f(g(x)) \circ \mathrm{D}g(x),$$

and Definition 1.31, we have

$$\mathrm{D}\hat{f}_x(0_x) = \mathrm{D}(f \circ R_x)(0_x) = \mathrm{D}f(R_x(0_x)) \circ \mathrm{D}R_x(0_x) = \mathrm{D}f(x).$$

In addition, if $\mathcal{M}$ is endowed with a Riemannian metric, then we also have the equality of the gradients

$$\operatorname{grad} \hat{f}_x(0_x) = \operatorname{grad} f(x).$$

To show this, recall that, by definition,

$$\forall \xi \in T_x\mathcal{M}, \qquad \mathrm{D}f(x)[\xi] = \langle \operatorname{grad} f(x), \xi \rangle,$$

and

$$\forall \xi \in T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}, \qquad \mathrm{D}\hat{f}_x(0_x)[\xi] = \langle \operatorname{grad} \hat{f}_x(0_x), \xi \rangle.$$

Since $\mathrm{D}\hat{f}_x(0_x) = \mathrm{D}f(x)$, this implies $\langle \operatorname{grad} f(x), \xi \rangle = \langle \operatorname{grad} \hat{f}_x(0_x), \xi \rangle$ for all $\xi \in T_x\mathcal{M}$. Since this holds for all ξ, we can drop the ξ and the inner product to obtain the equality $\operatorname{grad} f(x) = \operatorname{grad} \hat{f}_x(0_x)$.

In the next sections and examples we show how to define retractions on embedded submanifolds using the QR factorization and the polar decomposition.

1.2.1.1 Retractions on embedded submanifolds

Let $\mathcal{M}$ be an embedded submanifold of a vector space $\mathcal{E}$. Thus $T_x\mathcal{M}$ is a linear subspace of $T_x\mathcal{E} \simeq \mathcal{E}$. Since $x \in \mathcal{M} \subseteq \mathcal{E}$ and $\xi \in T_x\mathcal{M} \subseteq T_x\mathcal{E} \simeq \mathcal{E}$, with a little abuse of notation, we can write $x + \xi \in \mathcal{E}$ in the embedding space. So as a general recipe for embedded submanifolds, we can define a retraction $R_x(\xi)$ by

- moving along the direction ξ to get to the point $x + \xi$ in $\mathcal{E}$;
- mapping the point $x + \xi$ back to $\mathcal{M}$. When dealing with matrix manifolds, this step can be based on matrix decompositions, such as, e.g., the QR factorization or the polar decomposition.

This idea is formalized in the following proposition.

Proposition 1.33 (Retractions on embedded submanifolds [AMS08, Prop. 4.1.2])**.** *Let $\mathcal{M}$ be an embedded submanifold of $\mathcal{E}$ and let $\mathcal{N}$ be an abstract manifold such that* $\dim(\mathcal{M}) + \dim(\mathcal{N}) = \dim(\mathcal{E})$. *Assuming that:*

(i) there exists a diffeomorphism ϕ from $\mathcal{M} \times \mathcal{N}$ to an open submanifold $\mathcal{E}_*$ of $\mathcal{E}$, namely $\phi\colon \mathcal{M} \times \mathcal{N} \to \mathcal{E}_*$, defined by $(F, G) \mapsto \phi(F, G)$;

(ii) there exists a point $I \in \mathcal{N}$ satisfying $\forall F \in \mathcal{M}$, $\phi(F, I) = F$,

then the mapping

$$R_X(\xi)\colon \pi_1(\phi^{-1}(X + \xi)), \qquad X \in \mathcal{M},\ \xi \in T_X\mathcal{M},$$

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

Remark 1.34. Observe that since ϕ is a diffeomorphism, there exists the inverse mapping $\phi^{-1}\colon \mathcal{E}_* \to \mathcal{M} \times \mathcal{N}$. For instance, in the QR factorization, this will be $\phi^{-1}\colon \mathbb{R}^{n\times n} \to \mathrm{St}(n, p) \times \mathcal{S}_{\mathrm{upp}^+}(p)$, defined by $A \mapsto (Q, R)$. Here, $\mathcal{S}_{\mathrm{upp}^+}(p)$ denotes the set of all p-by-p upper triangular matrices with strictly positive diagonal elements.

Remark 1.35. Concretely, $\mathcal{N}$ can be a set of factors deriving from a matrix decomposition. Here, $\pi_1 \colon \mathcal{M} \times \mathcal{N} \to \mathcal{M} \colon (F, G) \mapsto F$ denotes the *projection onto the first component* of such a decomposition. For example, if one performs a QR factorization, applying π_1 would keep only the Q factor. This can be regarded as a kind of projection.

Proof of Proposition 1.33. Consider $(X + \xi) \in \mathcal{E}_*$ for any ξ in a neighborhood of 0_X. Since ϕ^{-1} is defined on the whole $\mathcal{E}_*$, it follows that $R_X(\xi)$ is defined for any ξ in a neighborhood of 0_X. We verify the properties of a retraction stated in Definition 1.31. Smoothness of R_X is direct. For the property $R_X(0_X) = X$, observe that

$$R_X(0_X) = \pi_1(\phi^{-1}(X + 0_X)) = \pi_1(\phi^{-1}(X)) = \pi_1((X, I)) = X.$$

For the local rigidity property, first note that the Taylor development gives

$$\phi(X + \xi, I) = \phi(X, I) + \mathrm{D}_1\phi(X, I)[\xi],$$

where D_1 denotes the derivative with respect to the first component. Because of assumption (ii) of Proposition 1.33, we have $\phi(X + \xi, I) = X + \xi$ and $\phi(X, I) = X$, which yield

$$X + \xi = X + \mathrm{D}_1\phi(X, I)[\xi],$$

and so the result

$$\forall \xi \in T_X\mathcal{M}, \quad \mathrm{D}_1\phi(X, I)[\xi] = \mathrm{D}\phi(X, I)[(\xi, 0)] = \xi.$$

Then since $(\pi_1 \circ \phi^{-1})(\phi(X, I)) = X$, it follows

$$\xi = \mathrm{D}(\pi_1 \circ \phi^{-1})(\phi(X, I))\,[\mathrm{D}_1\phi(X, I)[\xi]] = \mathrm{D}(\pi_1 \circ \phi^{-1})(X)[\xi] = \mathrm{D}R_X(0_X)[\xi],$$

which proves the claim that R_X is a retraction. □

Example 1.36 (Retraction on the unit sphere S^{n-1})**.** Let $\mathcal{M} = S^{n-1}, \mathcal{N} = \{\lambda \in \mathbb{R} \colon \lambda > 0\}$, and consider $\phi \colon \mathcal{M} \times \mathcal{N} \to \mathbb{R}^n_*$ defined by $(x, \lambda) \mapsto \lambda x$. Proposition 1.33 yields the retraction

$$R_x(\xi) = \frac{x + \xi}{\|x + \xi\|},$$

defined for all $\xi \in T_x S^{n-1}$. Observe that $R_x(\xi)$ is the point on the sphere S^{n-1} that minimizes the distance to $x + \xi$.

1.2.1.2 Retraction on the orthogonal group

Let $\mathcal{M} = O_n$, the orthogonal group, i.e., the set of all $Q \in \mathbb{R}^{n \times n}$ such that $Q^\mathsf{T}Q = I$. Let $A \in \mathbb{R}^{n \times n}_*$ be a full-rank matrix (see Section 1.1.2). In this section, we present two possibilities to define a retraction on the orthogonal group.

QR factorization. Let $A = QR$ with $Q \in O_n$ and $R \in \mathcal{S}_{\mathrm{upp}^+}(n)$, the set of all upper triangular matrices with strictly positive diagonal elements. The inverse of the QR factorization is

$$\phi \colon O_n \times \mathcal{S}_{\mathrm{upp}^+}(p) \to \mathbb{R}^{n \times n}_*,$$

defined by

$$(Q, R) \mapsto A = QR.$$

Now let $\mathrm{qf} = \pi_1 \circ \phi^{-1}$ denote the mapping that sends a matrix A to the Q factor of its QR factorization. The mapping qf can be computed by using the Gram–Schmidt orthonormalization procedure. To show that qf is a retraction, one needs to check that ϕ satisfies all the hypotheses of Proposition 1.33.

(i)
- ϕ is bijective because of the existence and uniqueness properties of the QR factorization;
- ϕ is smooth because the matrix product is smooth;
- ϕ^{-1} is C^∞, since Q is obtained by the Gram–Schmidt procedure, which is C^∞ over the set of full-rank matrices $\mathbb{R}_*^{n\times n}$, and R is obtained as $Q^{-1}A$.

(ii) The identity matrix I_n is the neutral element: $\phi(Q, I) = Q$, for all $Q \in O_n$.

Hence all the assumptions of Proposition 1.33 hold for ϕ. Recalling that tangent vectors to the orthogonal group have the form (1.3), we have that, for a matrix $X \in O_n$ and a tangent vector $X\Omega \in T_X O_n$,

$$R_X(X\Omega) = \mathrm{qf}(X + X\Omega) = \mathrm{qf}(X(I + \Omega)) = X\mathrm{qf}(I + \Omega)$$

is a retraction on the orthogonal group O_n.

Polar decomposition. The polar decomposition is the factorization $A = QP$ with $Q \in O_n$ and $P \in \mathcal{S}_{\mathrm{sym}^+}(n)$, i.e., the set of all symmetric positive definite matrices of order n. The inverse of the polar decomposition is

$$\phi\colon O_n \times \mathcal{S}_{\mathrm{sym}^+}(n) \to \mathbb{R}_*^{n\times n},$$

defined by

$$(Q, P) \mapsto A = QP.$$

The polar decomposition of A is given by [AMS08, p. 58]

$$\phi^{-1}(A) = \left(A(A^\mathsf{T}A)^{-1/2},\ \ (A^\mathsf{T}A)^{1/2}\right). \tag{1.4}$$

In fact, one can readily check that $A(A^\mathsf{T}A)^{-1/2} \in O_n$ and $(A^\mathsf{T}A)^{1/2} \in \mathcal{S}_{\mathrm{sym}^+}(n)$. Hence, for a matrix $X \in O_n$ and a tangent vector $X\Omega \in T_X O_n$, we have that

$$\begin{aligned} R_X(X\Omega) &= \pi_1(\phi^{-1}(X + X\Omega)) \\ &= X(I + \Omega)\left((X(I + \Omega))^\mathsf{T}X(I + \Omega)\right)^{-1/2} \\ &= X(I + \Omega)\left((I - \Omega)X^\mathsf{T}X(I + \Omega)\right)^{-1/2} \\ &= X(I + \Omega)(I - \Omega^2)^{-1/2} \end{aligned} \tag{1.5}$$

is a retraction on O_n. Computing this retraction requires an eigenvalue decomposition of $(I - \Omega^2)$ in order to calculate its matrix square root.

1.2.1.3 Retraction on the Stiefel manifold

As with the orthogonal group above, here we also present two possibilities for defining the retraction on the Stiefel manifold.

QR factorization. For a matrix $X \in \mathrm{St}(n, p)$ and a tangent vector $\xi \in T_X\mathrm{St}(n, p)$, the retraction based on a QR factorization is given by

$$R_X(\xi) = \mathrm{qf}(X + \xi),$$

where $\mathrm{qf}(A)$ denotes the Q factor of the decomposition of $A \in \mathbb{R}_*^{n\times p}$ as $A = QR$, with $Q \in \mathrm{St}(n, p)$ and $R \in \mathcal{S}_{\mathrm{upp}^+}(n)$. The retraction $R_X(\xi)$ can be computed in a finite number of arithmetic operations and square roots, using, e.g., the modified Gram–Schmidt algorithm.

Polar decomposition. The retraction on the Stiefel manifold based on the polar decomposition can be obtained by specializing (1.5) for the vector $X + \xi$

$$\begin{aligned} R_X(\xi) &= (X+\xi)\left((X+\xi)^\mathsf{T}(X+\xi)\right)^{-1/2} \\ &= (X+\xi)\,(X^\mathsf{T}X + X^\mathsf{T}\xi + \xi^\mathsf{T}X + \xi^\mathsf{T}\xi)^{-1/2} \\ &= (X+\xi)\,(I + \xi^\mathsf{T}\xi)^{-1/2}, \end{aligned}$$

where we used the fact that $X^\mathsf{T}\xi + \xi^\mathsf{T}X = 0$ since $\xi \in T_X\mathrm{St}(n,p)$. In general, to compute the matrix square root we need to perform an eigenvalue decomposition. When p is small, which is usually the case for the Stiefel manifold, the numerical cost of evaluating the polar retraction is reasonable since it involves the eigenvalue decomposition of the small matrix $(I_p + \xi^\mathsf{T}\xi)^{-1/2}$. The retraction based on the polar decomposition is actually a second-order approximation of the Riemannian exponential, and it represents an orthogonal projection on $\mathrm{St}(n,p)$. It gives the best approximation of any given matrix by an orthonormal matrix.

1.2.2 Line-search methods on manifolds

Line-search methods on manifolds are based on the update formula

$$x_{k+1} = R_{x_k}(t_k\eta_k),$$

where the search direction η_k is a tangent vector of $T_{x_k}\mathcal{M}$ and the step length t_k is a real scalar.

The recipe for constructing a line-search method can be summarized as follows:

- choose a retraction R;
- select a search direction η_k;
- select a step length t_k.

Figure 1.5 illustrates the components of a line-search method on a manifold.

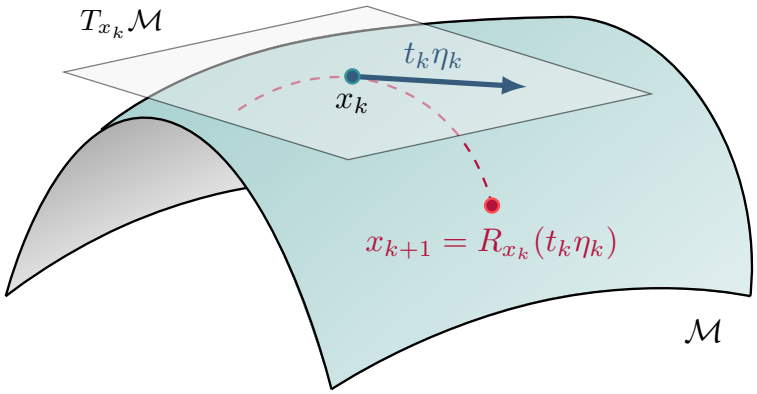


Figure 1.5 – Line search on a manifold.

In order to obtain global convergence results, we need to impose some restrictions on η_k and t_k. In particular, as x_k approaches a non-critical point, we would like to prevent the directions η_k from becoming orthogonal to the gradient direction, because this would cause the method to get stuck near that point.

Definition 1.37 (Gradient-related sequence)**.** Given a cost function f on a Riemannian manifold $\mathcal{M}$, we say that a sequence of tangent vectors $\{\eta_k\}$, $\eta_k \in T_{x_k}\mathcal{M}$, is *gradient related* if, for any subsequence of points $\{x_k\}_{k\in\mathcal{K}}$ that converges to a non-critical point of f, the corresponding subsequence of tangent vectors $\{\eta_k\}_{k\in\mathcal{K}}$ is bounded and satisfies

$$\lim_{k\to\infty} \sup_{k\in\mathcal{K}} \langle \operatorname{grad} f(x_k), \eta_k \rangle < 0.$$

This is a nonorthogonality type of condition. If $\{\eta_k\}$ is gradient related, it follows that if a subsequence $\{\operatorname{grad} f(x_k)\}_{k\in\mathcal{K}}$ tends to a nonzero vector, the corresponding subsequence of directions η_k is bounded and does not tend to be orthogonal to $\operatorname{grad} f(x_k)$. Roughly speaking, this means that the angle between the search direction η_k and $\operatorname{grad} f(x_k)$ does not get too close to 90 degrees [Ber95, p. 35]. This condition is very similar to the uniform angle condition of [BAC18, Lemma 2.10]. The latter allows to obtain algebraic convergence rates for the Riemannian gradient descent with a backtracking line-search procedure; see [BAC18, Theorem 2.11].

Figure 1.6 illustrates the concept of gradient-related sequence for vectors lying on a two-dimensional tangent space. The cyan half-plane highlights the part of the tangent plane where the relation $\langle \operatorname{grad} f(x_k), \eta_k \rangle < 0$ holds.

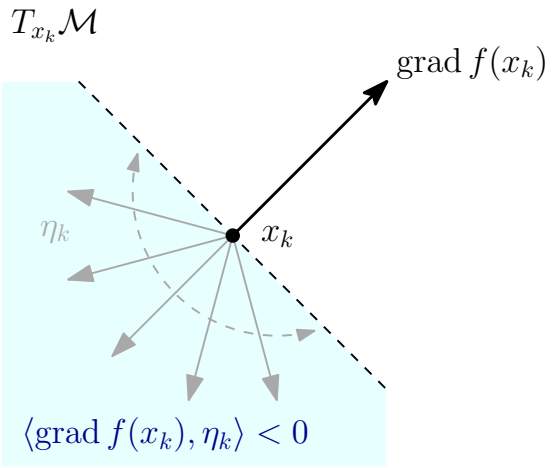


Figure 1.6 – Gradient-related vectors.

Definition 1.38 (Armijo point [AMS08, p. 62])**.** Given a cost function f on a Riemannian manifold $\mathcal{M}$ with retraction R, a point $x \in \mathcal{M}$, a tangent vector $\eta \in T_x\mathcal{M}$ and scalars $\bar{\alpha} > 0$, $\beta,\ \sigma \in (0,1)$, the *Armijo point* is

$$\eta^A = t^A\eta = \beta^m\bar{\alpha}\eta,$$

where $t^A = \beta^m\bar{\alpha}$ is the Armijo step size, and m is the smallest nonnegative integer such that

$$f(x) - f(R_x\eta^A) \geqslant -\sigma\langle \operatorname{grad} f(x), \eta^A\rangle_x.$$

Remark 1.39. The last expression is a condition of *sufficient decrease* for the cost function. Indeed, the left-hand side represents the decrease in f when moving along the direction of η^A while constrained to $\mathcal{M}$.

We are now ready to describe the line-search method on manifolds.

1.2.2.1 The accelerated Riemannian line-search algorithm

Given a Riemannian manifold $\mathcal{M}$, a smooth function f on $\mathcal{M}$, a retraction R from $T\mathcal{M}$ to $\mathcal{M}$, scalars $\bar{\alpha} > 0$ and $0 < c, \beta, \sigma < 1$, and an initial iterate $x_0 \in \mathcal{M}$, the line-search algorithm generates a sequence of iterates $\{x_k\}$ as follows. At each iteration $k = 0, 1, 2, \ldots$, it chooses a search direction η_k in the tangent space $T_{x_k}\mathcal{M}$ such that the sequence $\{\eta_i\}$ is gradient related (Definition 1.37). Then the new point x_{k+1} is chosen such that

$$f(x_k) - f(x_{k+1}) \geqslant c\left(f(x_k) - f(R_{x_k}(t_k^A \eta_k))\right), \tag{1.6}$$

where t_k^A is the Armijo step size (Definition 1.38) for the given $\bar{\alpha}$, β, σ, η_k.

Condition (1.6) leaves a lot of freedom in taking advantage of problem-related information that may produce a more efficient algorithm. Some possibilities to choose x_{k+1} in (1.6) are the following:

- $x_{k+1} = R_{x_k}(t_k^A \eta_k)$, where t_k^A is the Armijo point as described above;
- $x_{k+1} = R_{x_k}(t_k^* \eta_k)$, with t_k^* given by an exact line search $t_k^* = \arg\min_t f(R_{x_k}(t\eta_k))$, if this exact line search can be carried out efficiently;
- $x_{k+1} = R_{x_k}(\xi_k)$, with ξ_k defined by

  $$\xi_k = \arg\min_{\xi \in \mathcal{S}_k} f(R_{x_k}(\xi)),$$

  where $\mathcal{S}_k = \operatorname{span}\{\eta_k, R_{x_k}^{-1}(x_{k-1})\}$. This is a minimization over a two-dimensional subspace $\mathcal{S}_k$ of $T_{x_k}\mathcal{M}$. The subspace $\mathcal{S}_k$ contains the Armijo point associated with η_k, since η_k is in $\mathcal{S}_k$. This is a viable choice if the minimization over the subspace $\mathcal{S}_k$ can be carried out efficiently.

1.2.3 Convergence analysis

In this section, we discuss convergence concepts and limit points on manifolds, and then we give a convergence result for the line-search algorithm that we have just outlined above.

1.2.3.1 Convergence on manifolds

Definition 1.40 (Convergent sequence and limit point)**.** An infinite sequence $\{x_k\}_{k=0,1,\ldots}$ of points of a manifold $\mathcal{M}$ is *convergent* if there exists a chart $(\mathcal{U}, \psi)$ of $\mathcal{M}$, a point x_* of $\mathcal{U}$, and a $K > 0$ such that x_k is in $\mathcal{U}$ for all $k \geqslant K$ and such that the sequence $\{\psi(x_k)\}_{k=K,K+1,\ldots}$ converges to $\psi(x_*)$. The point $\psi^{-1}(\lim_{k\to\infty} \psi(x_k))$ is called the *limit point* of the convergent sequence $\{x_k\}_{k=0,1,\ldots}$.

Remark 1.41. The points of the sequence $\{x_k\}_{k=0,1,\ldots}$ can be outside $\mathcal{U}$, but after a certain $k = K$ they all fall inside $\mathcal{U}$.

Remark 1.42. Every convergent sequence of a Hausdorff manifold (see Definition 1.3) has one and only one limit point. For non-Hausdorff topologies, multiple distinct limit points are possible.

Equivalently, a sequence on a manifold is convergent if there exists a point x_* such that every neighborhood of x_* contains *all but* finitely many points of the sequence. Figure 1.7 illustrates the concept of convergent sequence and limit point on a manifold.

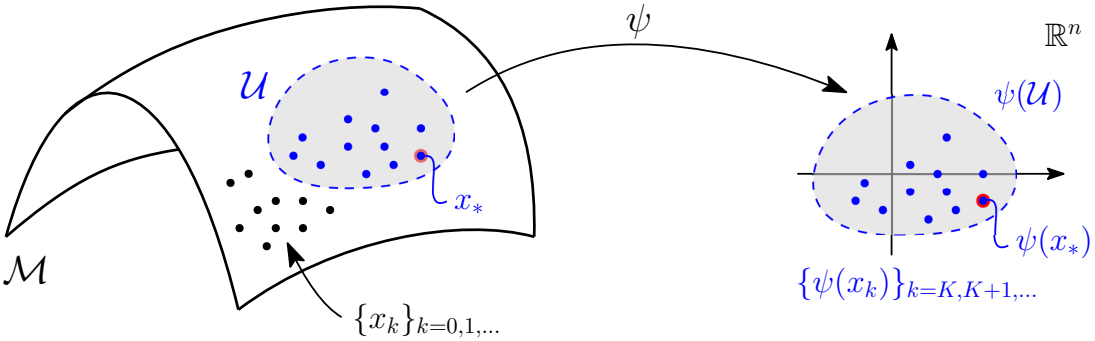


Figure 1.7 – Convergent sequence and limit point on a manifold.

1.2.3.2 Convergence of line-search methods

Theorem 1.43. *Let $\{x_k\}$ be an infinite sequence of iterates generated by the line-search algorithm of Section 1.2.2.1. Let f be a continuously differentiable scalar field, bounded below. Then every accumulation point of $\{x_k\}$ is a critical point of the cost function f.*

Remark 1.44. We are implicitly saying that a sequence can have more than one accumulation point, for example, from a sequence $\{x_k\}$ we may extract two subsequences such that they have two distinct accumulation points.

The proof of Theorem 1.43 can be done by contradiction, but it still remains quite technical, so we refer the interested reader to [AMS08, p. 65]. It should be pointed out that Theorem 1.43 only guarantees the convergence to critical points, but it does not tell us anything about their nature, i.e., it does not specify whether the critical points are local minimizers, local maximizers or saddle points. However, it is observed in practice that unless the initial point x_0 is designed in a "pathological way", line-search algorithms constructed according to the pattern discussed in Section 1.2.2.1 do produce sequences that converge to local minima of the cost function. These practical observations are supported by the stability analysis of critical points, which we do not discuss here.

1.2.4 Speed of convergence

How fast does the sequence $\{x_k\}$ converge to x_*? When $\mathcal{M}$ is a Riemannian manifold, it is possible to define a notion of linear convergence by using the Riemannian distance.

Definition 1.45 (Linear convergence)**.** Let $\mathcal{M}$ be a Riemannian manifold and let dist denote the Riemannian distance on $\mathcal{M}$ (see Definition 1.26). A sequence $\{x_k\}_{k=0,1,\ldots}$ *converges linearly* to a point $x_* \in \mathcal{M}$ if there exists a constant $c \in (0,1)$ and an integer $K \geqslant 0$ such that, for all $k \geqslant K$, it holds that

$$\operatorname{dist}(x_{k+1}, x_*) \leqslant c \operatorname{dist}(x_k, x_*). \tag{1.7}$$

The limit

$$\lim_{k\to\infty} \sup \frac{\operatorname{dist}(x_{k+1}, x_*)}{\operatorname{dist}(x_k, x_*)}$$

is called the *linear convergence factor* of the sequence. An iterative algorithm is said to converge locally linearly to a point x_* if there exists a neighborhood $\mathcal{U}$ of x_* and a constant $c \in (0,1)$ such that, for every initial point $x_0 \in \mathcal{U}$, the sequence $\{x_k\}$ generated by the algorithm satisfies (1.7).

Like other definitions that appeared in this chapter, this definition is also independent of the chart used.

We can also say that a sequence $\{x_k\}_{k=0,1,\ldots}$ on a Riemannian manifold converges linearly to x_* with constant c if and only if

$$\|R_{x_*}^{-1}(x_{k+1}) - R_{x_*}^{-1}(x_*)\| \leqslant c\,\|R_{x_*}^{-1}(x_k) - R_{x_*}^{-1}(x_*)\|,$$

for all k sufficiently large, where R is any retraction on $\mathcal{M}$ and $\|\cdot\|$ is the norm on $T_{x_*}\mathcal{M}$ induced by the Riemannian metric.

Let ε_g denote the accuracy for the gradient to satisfy the necessary optimality condition. Under the assumptions that f is bounded below on $\mathcal{M}$ and that $f \circ R_x$ has Lipschitz continuous gradient with constant L_g, [BAC18] showed that the Riemannian gradient descent with constant step size $1/L_g$ or with backtracking Armijo line search produces points with Riemannian gradient smaller than ε_g in $O(1/\varepsilon_g^2)$ iterations.

Chapter 2

Shooting methods on the Stiefel manifold

The object of study in this chapter is the compact Stiefel manifold, i.e.,

$$\mathrm{St}(n,p) = \left\{ X \in \mathbb{R}^{n\times p} : \; X^{\mathsf{T}}X = I_p \right\}.$$

As we have shown in Section 1.1.6.1, $\mathrm{St}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n\times p}$. In this chapter, we are concerned with computing the Riemannian distance (Definition 1.26) between two points on the Stiefel manifold. As we shall see, the distance between two points on a manifold is related to the concept of minimizing geodesic. Therefore, we start off this chapter by introducing the notion of *geodesics*.

2.1 Geodesics, exponential mapping and logarithm mapping

Geodesics are defined as curves with zero "acceleration", i.e., they solve the second-order ordinary differential equation (ODE)

$$\frac{\mathrm{D}^2}{\mathrm{d}t^2}\,\gamma(t) = 0,$$

where $\frac{\mathrm{D}^2}{\mathrm{d}t^2}$ denotes the *acceleration vector field*. Geodesics allow us to introduce the *Riemannian exponential* $\mathrm{Exp}_x \colon T_x\mathcal{M} \to \mathcal{M}$ that maps a tangent vector $\xi = \dot\gamma(0) \in T_x\mathcal{M}$ to the geodesic endpoint $\gamma(1) = y$: $\mathrm{Exp}_x(\xi) = y$. The Riemannian exponential is a local diffeomorphism (see Definition 1.6), i.e., it is locally invertible and its inverse is called the *Riemannian logarithm* of y at x: $\mathrm{Log}_x(y) = \xi$.

Thanks to a result of Riemannian geometry known as *Gauss's lemma*, the exponential map can be locally understood as a *radial isometry* [dC92, p. 69]. This means that one can measure the distance between two sufficiently close points on the manifold by computing the norm of the corresponding vector in the tangent space.

The diffeomorphicity of the exponential mapping is closely linked to the behavior of geodesics. While in Euclidean geometry straight lines are also distance-minimizing curves, in Riemannian geometry a geodesic $\gamma \colon [0,t] \to \mathcal{M}$ emanating from a point x is distance-minimizing only for small $t > 0$. In general, there exists a point $\gamma(t_c)$, called *cut point*, where the distance-minimizing property first breaks down [Sak96, p. 83]. The union of the cut points of all geodesics emanating from x is called *cut locus* of x; it is the boundary of the (star-shaped)

domain in which $\mathrm{Exp}_x : T_x\mathcal{M} \to \mathcal{M}$ is a diffeomorphism. The cut locus is closely linked not only to local properties such as the curvature of $\mathcal{M}$, but also to global topological properties [Sak96, ATV13].

The *injectivity radius* at a point x of a Riemannian manifold $\mathcal{M}$ is the largest radius for which the exponential map Exp_x is a diffeomorphism from the tangent space to the manifold; it is the least distance from x to the cut locus of x. The global injectivity radius of a manifold is the infimum of all the injectivity radii at all points of the manifold. Given two points x and x on a manifold $\mathcal{M}$, if $d(x, y) < \mathrm{inj}(\mathcal{M})$, then there exists a unique length-minimizing geodesic from x to y. For the Stiefel manifold, the injectivity radius is lower bounded by $0.89\,\pi$ [Ren13, Eq. (5.13)].

Given two points on the Stiefel manifold, our goal is to compute the length of the minimizing geodesic connecting them. For some manifolds, there are explicit formulas available for computing the distance, as in the case of the Grassmann manifold $\mathrm{Grass}(n, p)$. For instance, let $\mathcal{X}$ and $\mathcal{Y}$ belong to $\mathrm{Grass}(n, p)$, then the distance between $\mathcal{X}$ and $\mathcal{Y}$ is

$$\mathrm{dist}(\mathcal{X}, \mathcal{Y}) = \sqrt{\theta_1^2 + \cdots + \theta_p^2},$$

where θ_i, $i = 1, \ldots, p$, are the principal angles between $\mathcal{X}$ and $\mathcal{Y}$ (see [Won67, Thm. 8] and [AMS04, p. 211]). For the Stiefel manifold there is no such closed-form solution. In general, the problem of finding the distance given two points on a Riemannian manifold is related to the Riemannian logarithm function that we defined above. The problem of computing the Riemannian logarithm on the Stiefel manifold has already been tackled by several authors, who proposed some numerical algorithms. Rentmeesters [Ren13] and Zimmermann [Zim17, ZD19] proposed a similar algorithm which is only locally convergent and depends upon the definition of the matrix logarithm function.

Another method for finding geodesics is the leapfrog algorithm introduced by L. Noakes [Noa98]. This method has global convergence properties, but it slows down when the solution is approached [KN08, p. 2796]. This motivates the use of *shooting methods*, which have local quadratic convergence, when close to the solution. Indeed, shooting methods for finding the distance on the Stiefel manifold are the topic of this chapter. Moreover, Noakes realized that his leapfrog algorithm was in some way imitating the Gauss–Seidel method [Noa98, p. 39]. We will explore this connection later in Chapter 3.

2.1.1 Geodesics on the Stiefel manifold

A Riemannian metric has to be specified in order to turn $\mathrm{St}(n, p)$ into a Riemannian manifold, and in general different choices are possible. In this thesis, we consider the non-Euclidean *canonical metric* inherited by $\mathrm{St}(n, p)$ from its definition as a quotient space of the orthogonal group [EAS98, Eq. (2.39)]. Given $Y \in \mathrm{St}(n, p)$ and $\xi \in T_Y\mathrm{St}(n, p)$, the canonical metric reads

$$g_c(\xi, \xi) = \mathrm{trace}\big(\xi^\mathsf{T}(I - \tfrac{1}{2}YY^\mathsf{T})\,\xi\big). \tag{2.1}$$

Remark 2.1. Another popular choice, the *embedded metric* $g_e(\xi, \xi) = \mathrm{trace}(\xi^\mathsf{T}\xi)$, leads to very similar derivations, but we do not use it in this thesis.

By endowing the Stiefel manifold with the canonical metric, one can get the following second-order ordinary differential equation for the geodesic [EAS98, Eq. (2.41)]

$$\ddot{Y} + \dot{Y}\dot{Y}^\mathsf{T}Y + Y\big((Y^\mathsf{T}\dot{Y})^2 + \dot{Y}^\mathsf{T}\dot{Y}\big) = 0,$$

where $Y \equiv Y(t)$. An explicit formula for a geodesic that realizes a tangent vector ξ with base point Y_0 has been provided by Ross Lippert [EAS98, Eq. (2.42)]

$$Y(t) = Q \exp\left(\begin{bmatrix} \Omega & -K^\intercal \\ K & O_{n-p} \end{bmatrix} t \right) \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}, \tag{2.2}$$

with $Q = [Y_0\ Y_{0\perp}]$, $Y_{0\perp}$ being any matrix whose columns span $\mathcal{Y}_0^\perp = (\mathrm{span}(Y_0))^\perp$. We recall from Section 1.1.9 that tangent vectors to the Stiefel manifold may be expressed in the form

$$\xi = Y_0 \Omega + Y_{0\perp} K,$$

where Ω and K are the components of the tangent vector ξ in the subspaces spanned by the columns of Y_0 and $Y_{0\perp}$, respectively. In particular, $\Omega \in \mathcal{S}_{\mathrm{skew}}(p)$ and $K \in \mathbb{R}^{(n-p)\times p}$, $\mathcal{S}_{\mathrm{skew}}(p)$ being the vector space of p-by-p skew-symmetric matrices.

Remark 2.2. The matrix $Y_{0\perp}$ does not need to be orthonormal. Indeed, its only requirement is that it has to span $\mathcal{Y}_0^\perp = (\mathrm{span}(Y_0))^\perp$, the orthogonal subspace to $\mathcal{Y}_0 = \mathrm{span}(Y_0)$. See Appendix A.1.

2.2 Problem statement

In this section, we state the problem more formally. Given two points Y_0, Y_1 on $\mathrm{St}(n,p)$ that are sufficiently close to each other, finding the distance between them is equivalent to finding the tangent vector $\xi^* \in T_{Y_0}\mathrm{St}(n,p)$ with the shortest possible length such that [Lee18, Bou20]

$$\mathrm{Exp}_{Y_0}(\xi^*) = Y_1,$$

where Exp_{Y_0} denotes the Riemannian exponential mapping at Y_0. The solution to this problem is equivalent to the Riemannian logarithm of Y_1 with base point Y_0

$$\xi^* = \mathrm{Log}_{Y_0}(Y_1).$$

Figure 2.1 provides an artistic illustration of the problem statement.

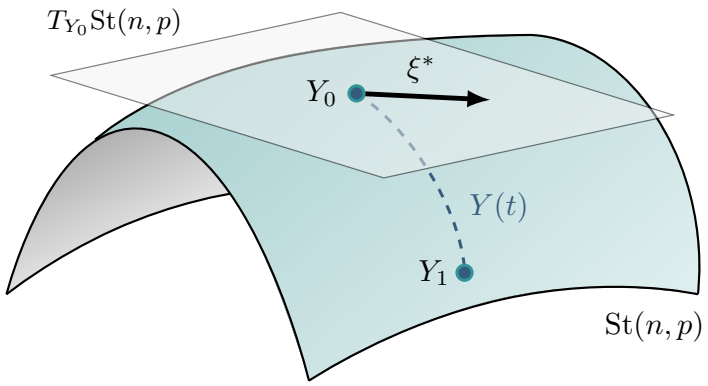


Figure 2.1 – Illustration of the problem statement.

In terms of the differential equation governing the geodesic, the problem statement may be written as follows:

Find $\xi^* \equiv \dot{Y}(0) \in T_{Y_0}\mathrm{St}(n,p)$ such that the second-order ODE

$$\ddot{Y} = -\dot{Y}\dot{Y}^\intercal Y - Y\big((Y^\intercal\dot{Y})^2 + \dot{Y}^\intercal\dot{Y}\big), \quad \text{with boundary conditions } \begin{cases} Y(0) = Y_0, \\ Y(1) = Y_1, \end{cases} \tag{2.3}$$

is satisfied. This kind of problem is known as a *boundary value problem* (BVP).

2.3 Single shooting method

The second-order ODE in problem (2.3) can be recast into a system of first-order ODEs. Let $Z_1(t) = Y(t)$, $Z_2(t) = \dot{Y}(t)$ be the geodesic and its derivative, respectively, and let

$$Z(t) = \begin{pmatrix} Z_1(t) \\ Z_2(t) \end{pmatrix}.$$

We get the initial value problem (we omit the dependence on t)

$$\begin{gathered} \dot{Z}(t) = \begin{pmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{pmatrix} = \begin{pmatrix} Z_2 \\ -Z_2 Z_2^\intercal Z_1 - Z_1\big((Z_1^\intercal Z_2)^2 + Z_2^\intercal Z_2\big) \end{pmatrix}, \\ \text{with initial conditions } Z(0) = \begin{pmatrix} Z_1(0) \\ Z_2(0) \end{pmatrix} = \begin{pmatrix} Y_0 \\ \xi \end{pmatrix}. \end{gathered} \tag{2.4}$$

Here, ξ is the unknown such that $Z_1(1) = Y_1$. In practice, since we already have the explicit formula (2.2) for the geodesic $Z_1(t)$, we do not need to solve the initial value problem (2.4). The explicit formula for Z_2 is just the derivative of Z_1 with respect to t, namely,

$$Z_2(t) = Q\,\exp\left(\begin{bmatrix} \Omega & -K^\intercal \\ K & O_{n-p} \end{bmatrix} t\right) \begin{bmatrix} \Omega \\ K \end{bmatrix}.$$

Now let us define the function

$$F(\xi) = \mathrm{vec}(Z_1(1,\xi) - Y_1), \tag{2.5}$$

where the dependence on ξ is explicit. Roughly speaking, this represents the mismatch between $Z_1(1)$, i.e., the geodesic at $t = 1$, and the boundary condition Y_1. We want to find ξ^* such that

$$F(\xi^*) = 0,$$

which can be solved by *Newton's method*. To apply Newton's method we need the Jacobian matrix of $F(\xi)$ with respect to ξ, denoted J_F^ξ. This actually reduces to $J_{Z_1}^\xi$, the Jacobian matrix of Z_1 with respect to ξ, since Y_1 appearing in $F(\xi)$ is not a function of ξ.

A pseudocode for the single shooting method is given in Algorithm 1. As stopping criterion, the norm of F is often used; in Section 2.3.2, we consider the 2-norm of the update $\delta\xi^{(k)}$. In the following sections, we will explain in more detail the algorithmic components of the single shooting method applied to the Stiefel manifold.

Algorithm 1: Single Shooting on the Stiefel manifold

Given Y_0, Y_1;
Result: ξ^* such that $\mathrm{Exp}_{Y_0}(\xi^*) = Y_1$
Set $\xi^{(k)} = \xi^{(0)}$;
while *a stopping criterion is met* **do**
  Compute Jacobian matrix $J_{Z_1}^{(k)}$;
  Compute $F^{(k)} = \mathrm{vec}(Z_1^{(k)} - Y_1)$;
  Solve $F^{(k)} + J_{Z_1}^{\xi^{(k)}} \delta\xi^{(k)} = 0$ for $\delta\xi^{(k)}$;
  Update $\xi^{(k)} \leftarrow \xi^{(k)} + \delta\xi^{(k)}$;
end

2.3.1 Parametrization of the tangent space

The tangent vector ξ belongs to $\mathbb{R}^{n\times p}$, but by inspecting its structure,

$$\xi = Y_0\Omega + Y_{0\perp}K,$$

one can observe that it only depends on $np - \frac{1}{2}(p+1)$ parameters (the dimension of the Stiefel manifold). Therefore we can express ξ as a function of these $np - \frac{1}{2}(p+1)$ parameters. By standard linear algebra arguments, it is possible to find a matrix $B \in \mathrm{R}^{p^2\times\frac{1}{2}p(p-1)}$ whose columns form a basis of $\mathcal{S}_{\mathrm{skew}}$. This allows us to write the vectorization of Ω as

$$\mathrm{vec}(\Omega) = Bs,$$

for some $s \in \mathbb{R}^{\frac{1}{2}p(p-1)}$ being a column vector representing Ω in the basis B of $\mathcal{S}_{\mathrm{skew}}$. The vectorization of the matrix K is simply $k = \mathrm{vec}(K) \in \mathbb{R}^{(n-p)p}$. Hence we can collect the coefficients of ξ in a single vector

$$x = \begin{pmatrix} s \\ k \end{pmatrix} \in \mathbb{R}^{np-\frac{1}{2}p(p+1)}.$$

Let us call

$$A(x) = \begin{bmatrix} \Omega & -K^{\mathsf{T}} \\ K & O_{n-p} \end{bmatrix}$$

the matrix in the argument of the exponential appearing in the geodesic (2.2). Then (2.2) can be rewritten as

$$Z_1(1,x) = Q\exp(A(x))\begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}.$$

Equation (2.5) becomes

$$F(x) = \mathrm{vec}\big(Z_1(1,x) - Y_1\big), \tag{2.6}$$

where we made clear the dependence on x. Newton's method consists in solving successive linearizations of this equation, i.e.,

$$F(x+\delta x) = Z_1(x+\delta x) - Y_1 = 0, \tag{2.7}$$

where we have omitted the vec operator for readability.

Here, the term $Z_1(x+\delta x)$ is the expression for the geodesic when we perturb the tangent vector. From $Z_1(x+\delta x)$ we can work out the Jacobian of Z_1 with respect to x, denoted $J^x_{Z_1}$. Applying matrix perturbation theory we obtain

$$Z_1(x+\delta x) = Z_1(x) + Q\ \mathrm{D}\exp(A(x))\big[\,\mathrm{D}A(x)[\delta x]\big] \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix} + o(\|\delta x\|), \tag{2.8}$$

where the notation $\mathrm{D}\exp(A(x))\big[\,\mathrm{D}A(x)[\delta x]\big]$ denotes the Fréchet derivative of the matrix exponential at $A(x)$ in the direction of $\mathrm{D}A(x)[\delta x]$. Clearly, a chain rule is involved in this term, so we first need to find $\mathrm{D}A(x)[\delta x]$. The perturbation of $A(x)$ yields

$$A(x+\delta x) = A(x) + \mathrm{D}A(x)[\delta x] + o(\|\delta x\|).$$

Let blkvec be the operator that performs a block-wise vectorization of $A(x)$, namely,

$$\begin{aligned}\mathrm{blkvec}(A(x)) = \mathrm{blkvec}\left(\begin{bmatrix} \Omega & -K^\mathsf{T} \\ K & O_{n-p} \end{bmatrix}\right) &= \begin{bmatrix} \mathrm{vec}(\Omega) \\ \mathrm{vec}(K) \\ \mathrm{vec}(-K^\mathsf{T}) \\ \mathrm{vec}(O_{n-p}) \end{bmatrix} = \begin{bmatrix} Bs \\ k \\ -\Pi_{n-p,p}\,k \\ O_{(n-p)^2\times 1} \end{bmatrix} \\ &= \begin{bmatrix} B & O_{p^2\times p(n-p)} \\ O_{p(n-p)\times \frac{1}{2}p(p-1)} & I_{p(n-p)} \\ O_{p(n-p)\times \frac{1}{2}p(p-1)} & -\Pi_{n-p,p} \\ O_{(n-p)^2\times \frac{1}{2}p(p-1)} & O_{(n-p)^2\times p(n-p)} \end{bmatrix} \begin{pmatrix} s \\ k \end{pmatrix},\end{aligned}$$

where $\Pi_{n-p,p}$ is the *perfect shuffle matrix* defined by

$$\mathrm{vec}(X^\mathsf{T}) = \Pi_{n-p,p}\ \mathrm{vec}(X).$$

From the last equation we can identify the Jacobian matrix of $A(x)$ with respect to x as

$$J^x_{A(x)} = \begin{bmatrix} B & O_{p^2\times p(n-p)} \\ O_{p(n-p)\times \frac{1}{2}p(p-1)} & I_{p(n-p)} \\ O_{p(n-p)\times \frac{1}{2}p(p-1)} & -\Pi_{n-p,p} \\ O_{(n-p)^2\times \frac{1}{2}p(p-1)} & O_{(n-p)^2\times p(n-p)} \end{bmatrix}.$$

Hence $\mathrm{vec}(\mathrm{D}A(x)[\delta x]) = J^x_{A(x)}\,\delta x$. We still need a map that links the block-wise vectorization $\mathrm{blkdiag}$ to the ordinary column-stacking vectorization vec. Since this mapping is linear, it can be represented by a matrix $T \in \mathbb{R}^{n^2\times n^2}$

$$\mathrm{vec}(\mathrm{D}A(x)[\delta x]) = T \cdot \mathrm{blkvec}(\mathrm{D}A(x)[\delta x]).$$

The perturbation of the matrix exponential yields

$$\exp(A+\delta A) = \exp(A) + \mathrm{D}\exp(A)[\delta A] + o(\|\delta A\|),$$

where $\mathrm{D}\exp(A)[\delta A]$ is the Fréchet derivative of the matrix exponential at A in the direction of δA. Vectorizing $\mathrm{D}\exp(A)[\delta A]$ we get

$$\mathrm{vec}(\mathrm{D}\exp(A)[\delta A]) = J^A_{\exp(A)}\,\mathrm{vec}(\delta A),$$

with $J^A_{\exp(A)}$ being the Jacobian of the matrix exponential. A closed-form expression for $J^A_{\exp(A)}$ is given in [Hig08, NH95],

$$J^A_{\exp(A)} = \Big(\exp(A^\mathsf{T}/2) \otimes \exp(A/2)\Big)\ \mathrm{sinch}\Big(\tfrac{1}{2}[A^\mathsf{T} \oplus (-A)]\Big)\,,$$

where $\oplus$ denotes the Kronecker sum: $A^\mathsf{T} \oplus (-A) = A^\mathsf{T} \otimes I_n - I_n \otimes A$, and sinch is the hyperbolic sinc,

$$\mathrm{sinch}(y) = \sinh(y)/y.$$

Vectorizing the second term on the right-hand side of (2.8) and wrapping things up, we get

$$\begin{aligned}
\mathrm{vec}\left(Q\,\mathrm{D}\exp(A(x))\big[\,\mathrm{D}A(x)[\delta x]\big]\begin{bmatrix} I_p \\ O \end{bmatrix}\right) &= \big([I_p\ \ O] \otimes Q\big)\,\mathrm{vec}\Big(\mathrm{D}\exp(A(x))\big[\,\mathrm{D}A(x)[\delta x]\big]\Big) \\
&= \big([I_p\ \ O] \otimes Q\big)\,J^A_{\exp(A)}\ \mathrm{vec}(\mathrm{D}A(x)[\delta x]) \\
&= \big([I_p\ \ O] \otimes Q\big)\,J^A_{\exp(A)}\,T\ \mathrm{blkvec}(\mathrm{D}A(x)[\delta x]) \\
&= \big([I_p\ \ O] \otimes Q\big)\,J^A_{\exp(A)}\,T\,J^x_{A(x)}\,\delta x.
\end{aligned}$$

From the last equation we can identify the Jacobian matrix of Z_1 with respect to x as

$$J^x_{Z_1} = \Big([I_p\ \ O_{p\times(n-p)}] \otimes Q\Big)\,J^A_{\exp(A)}\,T\,J^x_{A(x)}. \tag{2.9}$$

This means that the linearization of (2.7) yields

$$Z_1(x) + J^x_{Z_1}\,\delta x - Y_1 = 0,$$

i.e., the Newton update

$$J^x_{Z_1}\,\delta x = -F(x).$$

This is an *overdetermined* system to be solved for δx. Indeed, $J^x_{Z_1} : \mathbb{R}^{np-\frac{1}{2}p(p+1)} \to \mathbb{R}^{np}$, and since for $p \geqslant 1$ one has $np > np-\frac{1}{2}p(p+1)$, there are always more equations than unknowns. The system is overconstrained, but Newton's equation has a solution since $F(x) = 0$ is assumed to have a solution.

2.3.2 The initial guess

It is very well known that Newton's method has only local convergence properties, and in general a sufficiently good initial guess is needed in order for the method to converge. In this section, we describe the way we decided to initialize Newton's method. We need to choose an initial guess $\xi^{(0)}$ sufficiently close to ξ^*. To this aim, we use a first-order approximation of the matrix exponential $\exp(A) \approx I + A$ in (2.6) and solve for ξ. This yields the first-order approximation to the solution ξ^* as

$$\bar{\xi} = Y_1 - Y_0.$$

Then we project it onto $T_{Y_0}\mathrm{St}(n,p)$. We recall from Section 1.1.11 that the projection of a vector ξ onto the tangent space to the Stiefel manifold at Y is given by

$$\mathrm{P}_Y\,\xi = Y\mathrm{skew}(Y^\mathsf{T}\xi) + (I - YY^\mathsf{T})\,\xi.$$

The projection of $\bar{\xi}$ onto the tangent space at Y_0 yields

$$\mathrm{P}_{\bar{\xi}} = \mathrm{P}_{Y_0}(\bar{\xi}) = Y_0\,\mathrm{skew}\big(Y_0^\mathsf{T}(Y_1 - Y_0)\big) + (I_n - Y_0Y_0^\mathsf{T})(Y_1 - Y_0) = Y_1 - Y_0\,\mathrm{sym}(Y_0^\mathsf{T}Y_1).$$

To get $\xi^{(0)}$, we rescale this vector so that its norm is equal to the norm of $\bar{\xi}$,

$$\xi^{(0)} = \frac{\|\bar{\xi}\|}{\|\,\mathrm{P}_{\bar{\xi}}\,\|}\,\mathrm{P}_{\bar{\xi}}\,.$$

This procedure is illustrated in Figure 2.2.

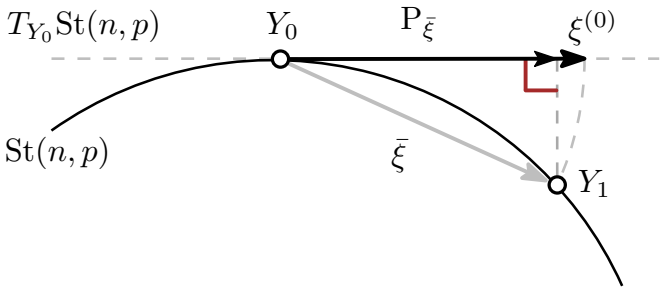


Figure 2.2 – Initial guess for the single shooting method.

2.3.3 A smaller formulation

It can be shown that the geodesic problem on $\mathrm{St}(n,p)$ is actually equivalent to a geodesic problem on $\mathrm{St}(2p,p)$ (see [EAS98, Ren13]). In the formulation above, the complexity of computing the matrix exponential is $O(n^3)$, but if $p \ll n$ then this smaller formulation can be used and its computational cost is only $O(p^3)$. In practice, it makes sense to consider the formulation on $\mathrm{St}(2p,p)$ only if $p < \frac{n}{2}$. In this section, we show how this "baby" formulation can be obtained.

Consider the same problem setting as in the previous sections, and let the QR factorization of K be

$$K = \begin{bmatrix} Q & Q_\perp \end{bmatrix} \begin{bmatrix} R \\ O_{(n-2p)\times p} \end{bmatrix} = QR,$$

where $\begin{bmatrix} Q & Q_\perp \end{bmatrix} \in \mathbb{R}^{(n-p)\times(n-p)}$ is the orthogonal factor of K, with $Q \in \mathbb{R}^{(n-p)\times p}$ and $Q_\perp \in \mathbb{R}^{(n-p)\times(n-2p)}$ orthonormal matrices, and $R \in \mathbb{R}^{p\times p}$ is upper triangular.

In Appendix A.2 we show that

$$Y_1 = \begin{bmatrix} Y_0 & Y_{0\perp}Q \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -R^\mathsf{T} \\ R & O_p \end{bmatrix}\right) \begin{bmatrix} I_p \\ O_p \end{bmatrix}. \tag{2.10}$$

Here, our aim is to find $\Omega \in \mathbb{R}^{p\times p}$ and $R \in \mathbb{R}^{p\times p}$ such that (2.10) holds true. Then, we can reconstruct the tangent vector as $\xi = Y_0\Omega + Y_{0\perp}QR$.

Let Y_1 be decomposed in the basis $\begin{bmatrix} Y_0 & Y_{0\perp}Q \end{bmatrix}$, and let M and N be the components of Y_1 in this basis

$$Y_1 = Y_0 M + Y_{0\perp} Q N. \tag{2.11}$$

This implies that

$$\begin{bmatrix} M \\ N \end{bmatrix} = \exp\left(\begin{bmatrix} \Omega & -R^\mathsf{T} \\ R & O_p \end{bmatrix}\right) \begin{bmatrix} I_p \\ O_p \end{bmatrix}. \tag{2.12}$$

Left-multiplication of (2.11) by Y_0^T and $Y_{0\perp}^\mathsf{T}$ yields, respectively $Y_0^\mathsf{T} Y_1 = M$ and $Y_{0\perp}^\mathsf{T} Y_1 = QN$. So one possible way to get N out of $Y_{0\perp}^\mathsf{T} Y_1$ is to compute its QR factorization

$$[Q, N] = \mathrm{qr}(Y_{0\perp}^\mathsf{T} Y_1). \tag{2.13}$$

The remarkable fact is that (2.12) is a geodesic problem on $\mathrm{St}(2p,p)$ with base point

$$\widehat{Y}_0 = \begin{bmatrix} I_p \\ O_p \end{bmatrix},$$

with $\widehat{\xi} = \widehat{Y}_0\Omega + \widehat{Y}_{0\perp}R$ the tangent vector to $\mathrm{St}(2p,p)$ at $\widehat{Y}_0$, and arrival point

$$\widehat{Y}_1 = \begin{bmatrix} M \\ N \end{bmatrix}.$$

Indeed, this problem setting yields the geodesic problem

$$\widehat{Y}_1 = \underbrace{[\widehat{Y}_0 \;\; \widehat{Y}_{0\perp}]}_{I_{2p}} \exp\left(\begin{bmatrix} \Omega & -R^\mathsf{T} \\ R & O_p \end{bmatrix}\right) \begin{bmatrix} I_p \\ O_p \end{bmatrix},$$

which is exactly (2.12).

This problem can be solved via the single shooting method described above to find $\widehat{\xi}^{(k)} = \widehat{Y}_0\Omega^{(k)} + \widehat{Y}_{0\perp}R^{(k)}$ at a certain iteration k (a stopping criterion is needed here). The components are given by $\Omega^{(k)} = \widehat{Y}_0^\mathsf{T}\widehat{\xi}^{(k)}$, $R^{(k)} = \widehat{Y}_{0\perp}^\mathsf{T}\widehat{\xi}^{(k)}$. Then the tangent vector $\xi^{(k)}$ of the original problem on $\mathrm{St}(n,p)$ can be recovered by

$$\xi^{(k)} = Y_0\Omega^{(k)} + Y_{0\perp}QR^{(k)},$$

where $Q \in \mathbb{R}^{(n-p)\times p}$ is the orthonormal factor of $Y_{0\perp}^\mathsf{T}Y_1$ as in (2.13).

2.3.4 Numerical example

As a concrete example to illustrate the single shooting algorithm, let us consider the Stiefel manifold $\mathrm{St}(15,4)$. We fix one point $X = [I_4 \;\; O_{11\times 4}]^\mathsf{T}$, while the other point Y is placed at a distance $L^* = 0.75\,\pi$ from X. In this way, the points X and Y are not too far from each other; indeed, they fall in the injectivity radius of $\mathrm{St}(n,p)$, which is lower bounded by $0.89\,\pi$ [Ren13, Eq. (5.13)]. By definition of the injectivity radius, this guarantees the existence and uniqueness of the minimizing geodesic between X and Y. By using single shooting, we want to recover this distance.

To monitor the convergence behavior of single shooting, we consider the norm of the update $\|\delta\xi^{(k)}\|_2$, as it appears in Algorithm 1, and stop the algorithm when 10^{-15} is reached. Figure 2.3 reports on the convergence behavior. The quadratic convergence of single shooting is clearly visible, and the threshold value of 10^{-15} is reached at the 5th iteration.

2.3.5 Some drawbacks

The local convergence behavior of Newton's method behind single shooting is such that the method will in general diverge unless the initial iterate is sufficiently close to a solution. Some simple analysis for the Jacobian of the matrix exponential involved in the single shooting method is provided in Appendix B.2.1. Moreover, unstable systems[1] remain difficult to treat, and Newton's method might possibly show bad convergence due to strong nonlinearity of the problem. These limitations make the single shooting method not very useful in practice.

[1]Unstable system or ill-conditioned BVPs: small perturbations of the data (i.e., the boundary conditions) cause big perturbations of the solution.

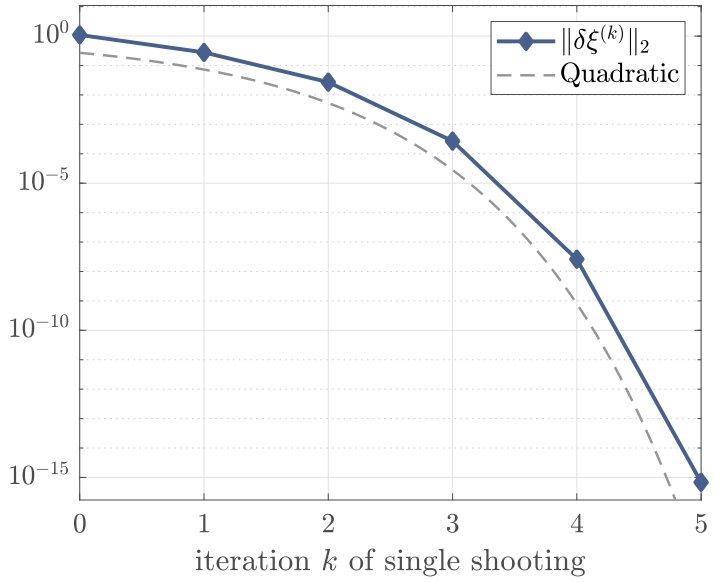


Figure 2.3 – Convergence of the update norm $\|\delta\xi^{(k)}\|_2$ for single shooting on $\mathrm{St}(15, 4)$.

2.4 Multiple shooting method

A way to improve over single shooting is to consider a partition of the original interval into many smaller subintervals, which leads us to the *multiple shooting method*. This slicing permits to reduce the nonlinearity of the problem and improve numerical stability. As in single shooting, there is also Newton's method behind multiple shooting. The difference is that many initial value problems are solved separately on all multiple shooting intervals. The resulting system to be solved is larger, but the banded structure of the Jacobian can be exploited. A thorough description of the multiple shooting method can be found in [SB91, p. 516]. Here, we will specialize the method in the context of the geodesic problem on the Stiefel manifold $\mathrm{St}(n, p)$.

Let X, Y be two points on a Stiefel manifold $\mathrm{St}(n, p)$. Consider a *piecewise* (or *broken*) *geodesic*[2] joining X to Y, having $m-1$ geodesic segments. Let $\Sigma_1^{(k)}$ denote the point on the Stiefel manifold on the kth subinterval, and $\Sigma_2^{(k)}$ the tangent vector to $\mathrm{St}(n, p)$ at $\Sigma_1^{(k)}$. Let Σ be the variable that collects the points and the tangent vectors for all k. The compatibility conditions of the geodesic and its first derivative, plus the two boundary conditions denoted by r_1 and r_2, can be encoded into a system of nonlinear equations to be solved for Σ

$$F(\Sigma) = \begin{bmatrix} Z_1^{(1)} - \Sigma_1^{(2)} \\ Z_2^{(1)} - \Sigma_2^{(2)} \\ Z_1^{(2)} - \Sigma_1^{(3)} \\ Z_2^{(2)} - \Sigma_2^{(3)} \\ \vdots \\ Z_1^{(m-1)} - \Sigma_1^{(m)} \\ Z_2^{(m-1)} - \Sigma_2^{(m)} \\ r_1 = \Sigma_1^{(1)} - X \\ r_2 = \Sigma_1^{(m)} - Y \end{bmatrix} = 0. \tag{2.14}$$

[2]For more details about the concept of broken geodesic, see Section 3.1.1 and Section 4.1.

Here, as in (2.4), $Z_1^{(k)}$ denotes the geodesic, whereas $Z_2^{(k)}$ is the derivative of the geodesic with respect to t. All the quantities $Z_i^{(k)}$ and $\Sigma_i^{(k)}$, $k = 1, \ldots, m-1$ and $i = 1, 2$, are to be understood as *vectorized* quantities.

Figure 2.4 illustrates the variables (points and tangent vectors) involved in the multiple shooting on the Stiefel manifold.

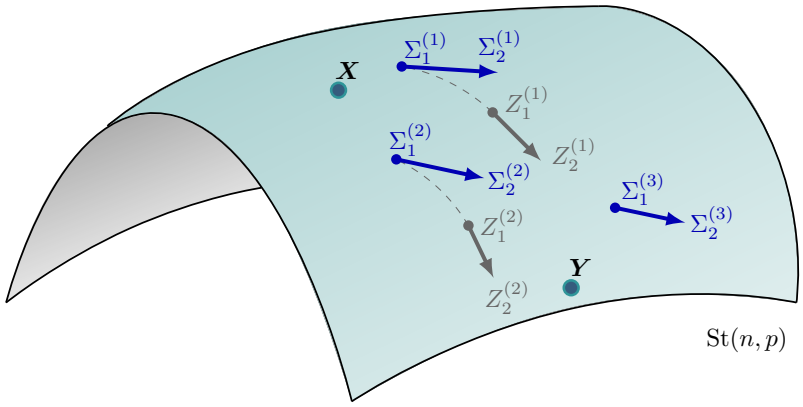


Figure 2.4 – Multiple shooting on the Stiefel manifold.

Now consider the perturbed system

$$F(\Sigma + \delta\Sigma) = 0, \qquad \text{with} \qquad \delta\Sigma = \begin{bmatrix} \delta\Sigma^{(1)} & \delta\Sigma^{(2)} & \cdots & \delta\Sigma^{(m)} \end{bmatrix}^{\mathsf{T}}.$$

A linearization of the previous equation gives

$$F(\Sigma) + J_F^{\Sigma} \cdot \delta\Sigma = 0, \tag{2.15}$$

where $J_F^{\Sigma} \in \mathbb{R}^{2mnp \times 2mnp}$ is a block Jacobian matrix. Each block $J_{Fk\ell}^{\Sigma} \in \mathbb{R}^{np \times np}$ is given by

$$J_{Fkk}^{\Sigma} = G^{(k)}, \qquad J_{Fk,k+1}^{\Sigma} = -I_{2np}, \qquad k = 1, \ldots, m-1,$$

$$J_{Fm,1}^{\Sigma} = C, \qquad J_{Fm,m}^{\Sigma} = D, \qquad J_{Fk\ell}^{\Sigma} = O_{2np} \quad \text{otherwise.}$$

Every $G^{(k)}$ is itself a Jacobian matrix for each subinterval defined as

$$G^{(k)} = \begin{bmatrix} \dfrac{\partial Z_1^{(k)}}{\partial \Sigma_1^{(k)}} & \dfrac{\partial Z_1^{(k)}}{\partial \Sigma_2^{(k)}} \\ \dfrac{\partial Z_2^{(k)}}{\partial \Sigma_1^{(k)}} & \dfrac{\partial Z_2^{(k)}}{\partial \Sigma_2^{(k)}} \end{bmatrix} = \begin{bmatrix} J_{Z_1}^{\Sigma_1} & J_{Z_1}^{\Sigma_2} \\ J_{Z_2}^{\Sigma_1} & J_{Z_2}^{\Sigma_2} \end{bmatrix}, \tag{2.16}$$

where we omitted the superscript $^{(k)}$ in the last matrix for ease of notation. We refer the reader to Appendix C for the explicit expressions of the Jacobian matrices appearing in (2.16). The Jacobian matrices associated to the boundary conditions are given by

$$C = \begin{bmatrix} \dfrac{\partial r_1}{\partial \Sigma_1^{(1)}} & \dfrac{\partial r_1}{\partial \Sigma_2^{(1)}} \\ \dfrac{\partial r_2}{\partial \Sigma_1^{(1)}} & \dfrac{\partial r_2}{\partial \Sigma_2^{(1)}} \end{bmatrix} = \begin{bmatrix} I_{np} & O_{np} \\ O_{np} & O_{np} \end{bmatrix}, \qquad D = \begin{bmatrix} \dfrac{\partial r_1}{\partial \Sigma_1^{(m)}} & \dfrac{\partial r_1}{\partial \Sigma_2^{(m)}} \\ \dfrac{\partial r_2}{\partial \Sigma_1^{(m)}} & \dfrac{\partial r_2}{\partial \Sigma_2^{(m)}} \end{bmatrix} = \begin{bmatrix} O_{np} & O_{np} \\ I_{np} & O_{np} \end{bmatrix}.$$

2.4.1 Condensing

The linear system (2.15) can be solved efficiently thanks to the structure of J_F^{Σ}, which allows any $\delta\Sigma^{(k)}$, $k = 2, \dots, m$, to be expressed as a function of $\delta\Sigma^{(1)}$ [SB91, p. 519]. Eventually, only one linear system of size $2np \times 2np$ has to be solved to find $\delta\Sigma^{(1)}$

$$M \cdot \delta\Sigma^{(1)} = -w,$$

where

$$M = C + D \cdot \prod_{k=m-1}^{1} G^{(k)}, \qquad w = F^{(m)} + D \cdot \sum_{k=1}^{m-1} \left(\prod_{\ell=k+1}^{m-1} G^{(\ell)} \right) \cdot F_k.$$

The other $\delta\Sigma^{(k)}$ are obtained as

$$\delta\Sigma^{(k)} = F^{(k-1)} + G^{(k-1)} \cdot \delta\Sigma^{(k-1)}, \quad k = 2, \dots, m.$$

The complexity of multiple shooting with this condensing strategy is $O(mn^3p^3)$.

2.4.2 Numerical example

Consider again the Stiefel manifold $\mathrm{St}(15, 4)$. Let $X = [I_4 \;\; O_{11\times 4}]^{\mathsf{T}}$, and let us place the other point Y at a distance $L^* = 0.89\,\pi$ from X. By using multiple shooting, we want to recover this distance. As number of points we choose $m = 7$, i.e., the path between X and Y is cut into 6 equidistant subintervals.

To monitor the convergence behavior, two quantities have been considered:

- $|L_k - L^*|$, where L_k is the length of the piecewise geodesic at iteration k.
- $\|F(\Sigma_k)\|_2$, where $F(\Sigma_k)$ is the nonlinear function defined by Equation 2.14.

Figure 2.5 reports on the convergence behavior. The quadratic convergence of multiple shooting is clearly visible, and the tolerance of 10^{-15} is reached at the 7th iteration.

2.4.3 Open questions

As we have already mentioned in Section 2.3.5, it is very difficult to say something on the global convergence of Newton's method. For local convergence, we have the result of the *Newton–Kantorovich theorem*. In practical applications, a sufficient number of iterations in the *leapfrog algorithm* (to be discussed in the next chapter) produces an iterate $\Sigma^{(k)}$ which satisfies the conditions of the Newton–Kantorovich theorem. One can observe that $F(\Sigma^{(k)})$ tends to zero as leapfrog progresses. For this reason, leapfrog can be used to initialize multiple shooting. We will see some concrete examples of this in Section 4.5.

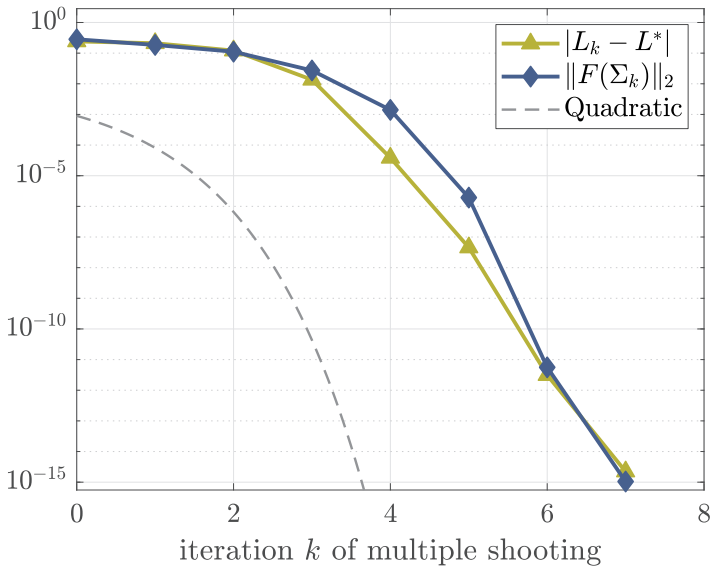


Figure 2.5 – Convergence of multiple shooting on $\mathrm{St}(15, 4)$.

Chapter 3

The leapfrog algorithm as nonlinear Gauss–Seidel

In the previous chapter, we have introduced some numerical algorithms to solve the geodesic problem on the Stiefel manifold. In particular, we focused on shooting methods, and we explored how they specialize to the Stiefel manifold, with corresponding advantages and disadvantages.

Another method for finding geodesics is the leapfrog algorithm introduced by L. Noakes [Noa98]. This method has global convergence properties, yet convergence of leapfrog slows down when the solution is approached [KN08, p. 2796]. Noakes also realized that his algorithm was in some way imitating the Gauss–Seidel method [Noa98, p. 39]. The Gauss–Seidel method is a well-known iterative method for solving a linear system of equations, and it can be readily extended to nonlinear systems of equations [OR00, p. 219]. The link between leapfrog and nonlinear Gauss–Seidel was not further investigated, since there is no trace of this idea being developed in the other related papers [KN97, KN98a, KN98b, KN08].

In this chapter, we will prove convergence of leapfrog as a nonlinear block Gauss–Seidel method. Even though our focus will be on $\mathrm{St}(n, p)$, most of our discussion may be generalized to other embedded submanifolds.

3.1 Leapfrog algorithm

The main idea behind the leapfrog algorithm of Noakes [Noa98] is to exploit the success of single shooting that we presented in Chapter 2 to construct a connecting geodesic when X and Y are two close points on $\mathcal{M}$. However, when X and Y are far apart, it is well known that single shooting will have difficulty finding the connecting geodesic. The leapfrog algorithm cuts this global problem into several local problems, where intermediate points $X_i \in \mathcal{M}$ are introduced between X and Y, for which the endpoint geodesic problem (2.4) can be solved by single shooting. This is similar to multiple shooting except that there is no explicit continuity equation and the geodesics are computed between X_{i-1} and X_{i+1}, hence they "skip" the middle point X_i. In multiple shooting, the continuity is enforced explicitly, whereas in leapfrog it follows automatically from the minimization of the length functional via a piecewise geodesic[1]. The algorithm then iteratively updates the piecewise geodesic to obtain a globally smooth geodesic between X and Y. This idea is not new and goes back as early as 1963 by Milnor [Mil63, III.§16].

[1] This and other aspects are further discussed in Chapter 4.

3.1.1 Formal description of the algorithm

In this section, we describe the leapfrog algorithm by following the presentation in [Noa98]. Let $\mathcal{M}$ be a C^∞ path-connected Riemannian manifold. Consider a *piecewise* (or *broken*) *geodesic* ω_X joining X_0 to X_{m-1}, having $m-1$ geodesic segments. Assuming X_i and X_{i+1} are sufficiently close to each other, ω_X is uniquely identified by the m-tuple $X = (X_0, X_1, \ldots, X_{m-1}) \in \mathcal{M}^m$, where X_i are the junctions of the geodesic segments. For $i = 1, \ldots, m-2$, each X_i is mapped onto the minimizing geodesic joining X_{i-1} and X_{i+1}. This achieves the largest possible decrease in length while keeping other variables fixed. Though there are several choices to do this, leapfrog maps X_i onto the midpoint of the geodesic joining X_{i-1} and X_{i+1}. By iterating this procedure, the algorithm generates a sequence $\Omega = \{\omega_{X^{(k)}} : [0,1] \to \mathcal{M} : k = 0, 1, \ldots\}$ of broken geodesics whose lengths are decreasing. Figure 3.1 illustrates one iteration of the leapfrog algorithm.

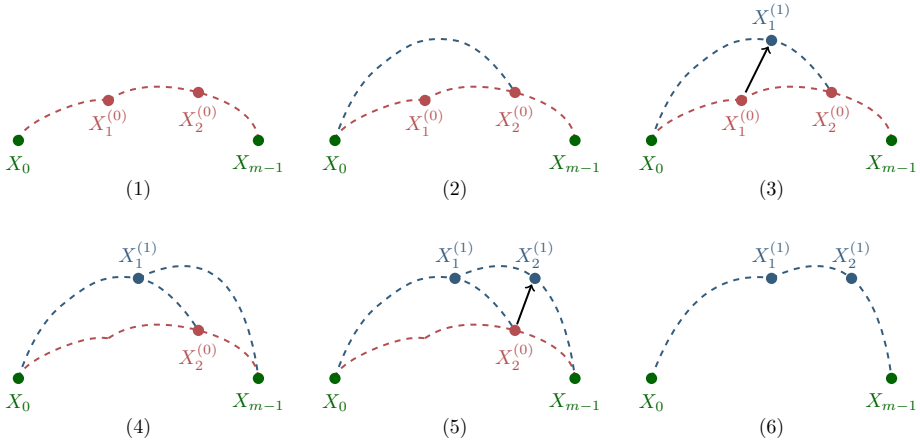


Figure 3.1 – Illustration of one full iteration of the leapfrog scheme for some non-Euclidean metric (the lengths for the Euclidean metric clearly increase during iteration).

The leapfrog algorithm. Let $\mathfrak{M} : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ denote the *midpoint map* defined by

$$\mathfrak{M}(X, Y) = \operatorname{Exp}_X\left(\tfrac{1}{2} \operatorname{Log}_X(Y)\right),$$

where we have silently assumed that $d(X, Y) < \operatorname{inj}(\mathcal{M})$ so that the Riemannian logarithm is well defined (see Section 2.1). One complete outer iteration (indexed by $k = 1, 2, \ldots$) of leapfrog comprises $m-2$ inner iterations indexed by $i = 1, \ldots, m-2$ that compute

$$X_i^{(k)} = \mathfrak{M}(X_{i-1}^{(k)}, X_{i+1}^{(k-1)}). \tag{3.1}$$

In other words, $X_i^{(k-1)}$ is replaced by the midpoint $X_i^{(k)}$ of the minimizing geodesic joining $X_{i-1}^{(k)}$ and $X_{i+1}^{(k-1)}$. This process is repeated until all points $X_1^{(k-1)}, \ldots, X_{m-2}^{(k-1)}$ have been updated in order. See Figure 3.1 for one such iteration of the leapfrog algorithm. The iteration is started with $X^{(0)} = (X_0^{(0)}, X_1^{(0)}, X_2^{(0)}, \ldots, X_{m-1}^{(0)})$, and repeated until a stopping criterion is satisfied. Since the endpoints do not change, we denote $X_0 = X_0^{(k)}$ and $X_{m-1} = X_{m-1}^{(k)}$ for all k.

It is clear that leapfrog implicitly generates a sequence

$$\Omega = \{\omega_{X^{(k)}} : [0,1] \to \mathcal{M} : k = 0, 1, \ldots\}$$

of broken geodesics $\omega_{X^{(k)}}$ that are defined from $X^{(k)}$. In addition, the length of $\omega_{X^{(k)}}$ is non-increasing in k since at each step two neighboring geodesics get replaced by one global geodesic connecting their endpoints.

3.1.2 Known results

Let $\mathcal{Y}$ be the set of all tuples $X = (X_0, X_1, \ldots, X_{m-1}) \in \mathcal{M}^m$ satisfying $d(X_{i-1}, X_i) \leqslant \delta$ for all $i = 1, 2, \ldots, m-2$. In [Noa98, §2], δ is related to the notion of Lebesgue number of an open cover. Here, we can assume that δ is equal to $\frac{1}{2}\operatorname{inj}(\mathcal{M})$, where inj is the injectivity radius of the manifold (see Section 2.1). Let $\mathcal{F}\colon \mathcal{Y} \to \mathcal{Y}$ represent one full leapfrog iteration and let X^* be the limit of any convergent subsequence of $S = \{\mathcal{F}^k(X^{(0)})\colon k \geqslant 1\}$ with $X^{(0)} \in \mathcal{Y}$. By compactness, [Noa98] shows that at least one convergent subsequence of S exists and that the limit of this subsequence are points that lie on a global geodesic connecting the endpoints X_0 and X_{m-1}. The following result is stated in [KN08, Theorem 5.2].

Theorem 3.1. *S has a unique accumulation point.*

The theorem guarantees convergence of the iterates $X^{(k)} = \mathcal{F}(X^{(k-1)})$ with $X^{(0)} \in \mathcal{Y}$. From [Noa98, Lemma 3.2] we also know that leapfrog will converge to a uniformly distributed m-tuple $X^* = (X_0, X_1^* \ldots, X_{m-2}^*, X_{m-1})$, i.e., $d(X_i^*, X_{i+1}^*)$ are all equal, for $i = 0, \ldots, m-2$. In other words, at convergence, the geodesic segments connecting the junction points will all have the same length. This aspect is further investigated in Section 4.3.

An apparent drawback in the current theory is that it lacks a classical convergence proof as a fixed-point iteration method, although leapfrog can be easily recognized as such. In the next section, we will provide the details of how to analyze leapfrog as a nonlinear block Gauss–Seidel method.

3.2 Convergence of leapfrog as nonlinear Gauss–Seidel

Let $\mathcal{M} = \operatorname{St}(n, p)$ with the Riemannian distance function d. The starting point is to realize that leapfrog solves the optimization problem

$$\min_{X_1,\ldots,X_{m-2}\in\operatorname{St}(n,p)} F(X_1, \ldots, X_{m-2}) \quad \text{with} \quad F(X_1, \ldots, X_{m-2}) = \sum_{i=1}^{m-1} d^2(X_{i-1}, X_i),$$

by cyclically minimizing over each variable X_i for $i = 1, 2, \ldots, m-2$. Specifically, at the kth iteration, leapfrog updates $X_i^{(k-1)}$ by the minimizer of the problem

$$\begin{aligned} &\min_{X_i\in\operatorname{St}(n,p)} F(X_1^{(k)}, \ldots, X_{i-1}^{(k)}, X_i, X_{i+1}^{(k-1)}, \ldots, X_{m-2}^{(k-1)}) \\ &= \min_{X_i\in\operatorname{St}(n,p)} d^2(X_{i-1}^{(k)}, X_i) + d^2(X_i, X_{i+1}^{(k-1)}) + \text{constant}. \end{aligned} \tag{3.2}$$

Since d is the Riemannian distance function, this problem coincides with the definition of the Riemannian center of mass[2,3] between the two points $X_{i-1}^{(k)}$ and $X_{i+1}^{(k-1)}$; see [Kar77,

[2]The Riemannian center of mass was constructed in [GK73]. As H. Karcher points out in [Kar14], "Probably in 1990 someone renamed it without justification into *karcher mean* and references to the older papers were omitted by those using the new name. (...) I think it is fair to say that a substantial amount of damage was caused by the renaming". For this reason, in this thesis, we decided to stick to the original name.

[3]A numerical experiment involving the Riemannian center of mass on the Stiefel manifold is discussed in Section 4.5.

Eq. (1.1)]. For the compact Stiefel manifold, a Riemannian center of mass always exists, but it does not need to be unique [Ren13, p. 37]. However, a sufficient condition for uniqueness is $d(X_{i-1}^{(k)}, X_{i+1}^{(k-1)}) < \operatorname{inj}(\operatorname{St}(n,p))$, where inj is the injectivity radius (see Section 2.1). This is true if all X_i are close enough (we will make this more precise later). In that case, the unique solution that solves (3.2) is the midpoint of the minimizing geodesic between $X_{i-1}^{(k)}$ and $X_{i+1}^{(k-1)}$. Leapfrog now proceeds to update the X_i in a Gauss–Seidel fashion where the most recent $X_{i-1}^{(k)}$ is used to update $X_i^{(k-1)}$. This kind of optimization scheme is known as *block coordinate descent method* of Gauss–Seidel type [OR00].

3.2.1 Nonlinear block Gauss–Seidel method

Let us first consider the case of Gauss–Seidel in $\mathbb{R}^n$. Let the variable $x \in \mathbb{R}^n$ be partitioned as $x = (x_1, x_2, \ldots, x_m)$, where $x_i \in \mathbb{R}^{q_i}$ and $\sum_i q_i = n$, and group correspondingly the components of $\widetilde{F}\colon D \subset \mathbb{R}^n \to \mathbb{R}^n$ into mappings $\widetilde{F}_i\colon \mathbb{R}^n \to \mathbb{R}^{q_i}$, $i = 1, \ldots, m$. The minimizers of the function $\widetilde{F}(x)$ satisfy the first-order optimality condition $\nabla \widetilde{F}(x) = 0$. Let us define $\mathcal{G}_i = \nabla \widetilde{F}_i$, $i = 1, \ldots, m$. If we interpret the linear Gauss–Seidel iteration in terms of obtaining $x_i^{(k)}$ as the solution of the ith equation of the system with the other $m-1$ block variables held fixed, then we may immediately consider the same prescription for nonlinear equations [OR00, p. 219]. Then solving

$$\mathcal{G}_i(x_1^{(k)}, \ \ldots, \ x_{i-1}^{(k)}, \ y, \ x_{i+1}^{(k-1)}, \ \ldots, \ x_m^{(k-1)}) = 0 \tag{3.3}$$

for y and defining $x_i^{(k)} = y$ describes a nonlinear block Gauss–Seidel process in which a complete iteration requires the solution of m nonlinear systems of dimensions q_i, $i = 1, \ldots, m$; see [OR00, p. 225]. The convergence theory in [OR00] applies only to functions whose domain of definition is Euclidean space $\mathbb{R}^n$. It cannot be applied to functions which are defined on manifolds, such as the Riemannian distance d that is only defined on a subset of $\mathbb{R}^n$, namely, the embedded submanifold. For this reason, in the next section we will introduce a smooth extension of the Riemannian distance function that can also be evaluated for points that do not belong to the manifold.

3.2.2 Extended objective function

As we have seen above, leapfrog solves in an alternating way the problem

$$\min_{X_1,\ldots,X_{m-2} \in \operatorname{St}(n,p)} F(X_1, \ldots, X_{m-2}) = \sum_{i=1}^{m-1} d^2(X_{i-1}, X_i),$$

where X_0 and X_{m-1} are the fixed endpoints. This objective function F is only defined on the manifold $\operatorname{St}(n,p)$. In this section, we will identify an *extended objective function* $\widetilde{F}$ that is defined on $\mathbb{R}^{n\times p}$ for which the standard nonlinear block Gauss–Seidel method produces the same iterates as the leapfrog algorithm. The key result of this section is stated in Proposition 3.7. This will allow us to analyze the convergence of leapfrog using standard results for nonlinear Gauss–Seidel.

We claim the extended cost function can be chosen as

$$\min_{X_1,\ldots,X_{m-2} \in \mathbb{R}^{n\times p}} \widetilde{F}(X_1, \ldots, X_{m-2}) = \sum_{i=1}^{m-1} \widetilde{d}^2(X_{i-1}, X_i),$$

with *extended distance function*

$$\widetilde{d}^2(\widetilde{X},\widetilde{Y}) = \begin{cases} d^2(\mathrm{P}_{\mathrm{St}}\widetilde{X},\mathrm{P}_{\mathrm{St}}\widetilde{Y}) + \|\widetilde{X}-\mathrm{P}_{\mathrm{St}}\widetilde{X}\|_{\mathrm{F}}^2 + \|\widetilde{Y}-\mathrm{P}_{\mathrm{St}}\widetilde{Y}\|_{\mathrm{F}}^2 \\ \qquad\qquad\qquad\qquad \text{if } \sigma_p(\widetilde{X})>0 \text{ and } \sigma_p(\widetilde{Y})>0, \\ +\infty \qquad\qquad\qquad\qquad \text{otherwise,} \end{cases} \tag{3.4}$$

where P_{St} denotes the orthogonal projector onto the Stiefel manifold.

The condition $\sigma_p(\widetilde{X}) > 0$ is equivalent to the existence of a unique best approximation of $\widetilde{X}$ in $\mathrm{St}(n,p)$. In other words, $\mathrm{P}_{\mathrm{St}}\widetilde{X}$ is well defined. Concretely, we can define the projector $\mathrm{P}_{\mathrm{St}}\colon \mathbb{R}^{n\times p} \to \mathrm{St}(n,p)$ by $\mathrm{P}_{\mathrm{St}}(Z) = Z(Z^{\mathsf{T}}Z)^{-1/2}$, that is, the orthogonal factor of the polar decomposition of Z (see Section 1.2.1.2, Equation (1.4)). Figure 3.2 illustrates the extended distance function $\widetilde{d}^2(\widetilde{X},\widetilde{Y})$.

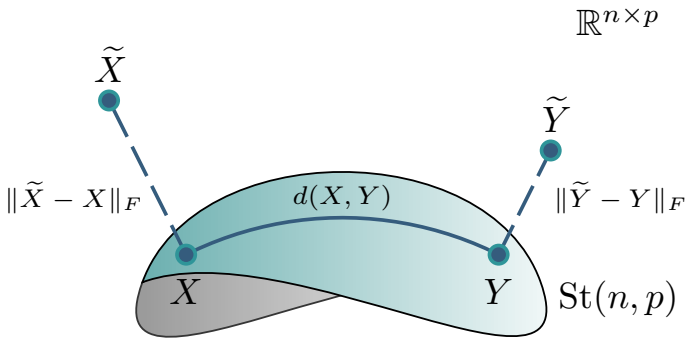


Figure 3.2 – The extended distance function.

3.2.3 Leapfrog as nonlinear Gauss–Seidel

In order to show that nonlinear Gauss–Seidel applied to $\widetilde{F}$ is equivalent to leapfrog for F, we need a few lemmas. The first one addresses the problem of how close the points on $\mathrm{St}(n,p)$ need to be so that their connecting geodesic is unique.

Lemma 3.2. *Let $X, Y \in \mathrm{St}(n,p)$ such that $d(X,Y) \leqslant \delta_g$, with $\delta_g = 0.89\,\pi$. Then there exists a unique minimizing geodesic between X and Y. As a consequence, also the Riemannian center of mass between X and Y exists and is uniquely defined.*

Proof. By definition of injectivity radius, if $d(X,Y) < \mathrm{inj}(\mathrm{St}(n,p))$, then there is only one minimizing geodesic between X and Y. From [Ren13, Eq. (5.13)], we know that the injectivity radius is lower bounded by $0.89\,\pi$. □

Remark 3.3. We can compare the Riemannian and Euclidean distances between X and $Y \in \mathrm{St}(n,p)$ asymptotically in the following way[4]. From the expansion of the canonical distance in (D.4), it is clear that

$$d(X,Y) \leqslant \|X-Y\|_{\mathrm{F}} + O(\|X-Y\|_{\mathrm{F}}^2) \quad \text{for} \quad \|X-Y\|_{\mathrm{F}} \to 0.$$

[4]For the Riemannian distance d_{e} based on the embedded metric, it is easy to see that $\|X-Y\|_{\mathrm{F}} \leqslant d_{\mathrm{e}}(X,Y)$ since the Euclidean length of a geodesic on $\mathrm{St}(n,p)$ is always larger than that of a straight line.

By neglecting $O(\|X-Y\|_{\mathrm{F}}^2)$, we thus have $d(X,Y) \lesssim \|X-Y\|_{\mathrm{F}}$. In particular, $\|X-Y\|_{\mathrm{F}} \leqslant \delta_g$ implies $d(X,Y) \lesssim \delta_g$.

Let $X_{i-1}, X_{i+1} \in \mathrm{St}(n,p)$. Denote

$$F_i(Y) = d^2(X_{i-1},Y) + d^2(Y,X_{i+1}), \qquad \widetilde{F}_i(\widetilde{Y}) = \widetilde{d}^2(X_{i-1},\widetilde{Y}) + \widetilde{d}^2(\widetilde{Y},X_{i+1}),$$

where X_{i-1}, X_{i+1} are constant and hidden in the notation.

Lemma 3.4. *With the notation from above assume that $d(X_{i-1},X_{i+1}) \leqslant \delta_g$, then the ith substep of leapfrog produces the same solution Y^* as the minimization of $\widetilde{F}_i$*

$$\operatorname*{arg\,min}_{Y\in\mathrm{St}(n,p)} F_i(Y) = \operatorname*{arg\,min}_{\widetilde{Y}\in\mathbb{R}^{n\times p}} \widetilde{F}_i(\widetilde{Y}) = Y^*,$$

with Y^* the Riemannian center of mass on $\mathrm{St}(n,p)$ of X_{i-1} and X_{i+1}.

Proof. Since $d(X_{i-1},X_{i+1}) \leqslant \delta_g$, Lemma 3.2 gives that the minimizer of F_i on $\mathrm{St}(n,p)$ is unique and equals the Riemannian center of mass Y^*. To show that it also equals the minimizer of $\widetilde{F}_i$ on $\mathbb{R}^{n\times p}$, take any $\widetilde{Y} \in \mathbb{R}^{n\times p}$. If $\sigma_k(\widetilde{Y}) > 0$, then we can write

$$\widetilde{Y} = Y + \Delta, \qquad Y = \mathrm{P}_{\mathrm{St}}\widetilde{Y} \in \mathrm{St}(n,p).$$

Using that Y^* is the minimizer of F_i on $\mathrm{St}(n,p)$, we thus get

$$\widetilde{F}_i(\widetilde{Y}) = d^2(X_{i-1},Y) + d^2(Y,X_{i+1}) + 2\|\Delta\|_{\mathrm{F}}^2 \geqslant F_i(Y) \geqslant F_i(Y^*).$$

The same inequality holds trivially if $\sigma_k(\widetilde{Y}) = 0$ since then $\widetilde{F}_i(\widetilde{Y}) = +\infty$. Finally, since $\widetilde{F}_i(Y^*) = F_i(Y^*)$, we obtain that $\widetilde{F}_i$ is also uniquely minimized by Y^*. □

Lemma 3.5. *Suppose that for all iterations $k = 0, 1, \ldots$, the iterates of leapfrog satisfy*

$$d(X_{i-1}^{(k)}, X_{i+1}^{(k-1)}) \leqslant \delta_g,$$

for all $i = 1, 2, \ldots, m-2$. Then, the leapfrog algorithm started in $X^{(0)}$ generates the same iterates as the nonlinear Gauss–Seidel algorithm started in $X^{(0)}$ and applied to

$$\min_{X_1,\ldots,X_{m-2}\in\mathbb{R}^{n\times p}} \widetilde{F}(X_1,\ldots,X_{m-2}).$$

Proof. By induction. Suppose true until substep $i-1$ of iteration k. Then, leapfrog computes the new iterate as

$$X_i^{(k)} = \operatorname*{arg\,min}_{Y\in\mathrm{St}(n,p)} d^2(X_{i-1}^{(k)}, Y) + d^2(Y, X_{i+1}^{(k-1)}).$$

The uniqueness of the minimizer follows from Lemma 3.2 and $d(X_{i-1}^{(k)}, X_{i+1}^{(k-1)}) \leqslant \delta_g$. Likewise, nonlinear Gauss–Seidel computes

$$\widetilde{X}_i^{(k)} = \operatorname*{arg\,min}_{\widetilde{Y}\in\mathbb{R}^{n\times p}} \widetilde{F}(X_1^{(k)},\ldots,X_{i-1}^{(k)},\widetilde{Y},X_{i+1}^{(k-1)},\ldots,X_{m-2}^{(k-1)}),$$

and the uniqueness of the minimizer follows from our reasoning below. Both minimization problems are the same as minimizing F_i and $\widetilde{F}_i$ from Lemma 3.4 but with $X_{i-1}^{(k)}$ and $X_{i+1}^{(k-1)}$ taking the roles of X_{i-1} and X_{i+1}, respectively. By Lemma 3.4, the minimizers of both problems are the same and hence $X_i^{(k)} = \widetilde{X}_i^{(k)}$. The above reasoning can also be applied to the base case $k = i = 1$ since $X_0^{(1)} = X_0^{(0)}$. Hence, we have proven the result. □

If the initial points are close enough, the iterates in leapfrog stay close.

Lemma 3.6. *Let $X^{(0)} \in \mathrm{St}(n,p)^m$ be such that $d(X_{i-1}^{(0)}, X_i^{(0)}) \leqslant \frac{1}{2}\delta_g$ for all $1 \leqslant i \leqslant m-1$. Then, leapfrog started at $X^{(0)}$ is well defined and all its iterates $X^{(k)}$ satisfy for all $1 \leqslant i \leqslant m-2$ and $k \geqslant 1$*

$$d(X_{i-1}^{(k)}, X_i^{(k)}) = d(X_i^{(k)}, X_{i+1}^{(k-1)}) \leqslant \tfrac{1}{2}\delta_g. \tag{3.5}$$

Proof. By induction. Suppose true for all substeps i until iteration $k-1$ and until substep $i-1$ of iteration k. This implies in particular

$$d(X_{i-1}^{(k)}, X_i^{(k-1)}) \leqslant \tfrac{1}{2}\delta_g, \quad d(X_i^{(k-1)}, X_{i+1}^{(k-1)}) \leqslant \tfrac{1}{2}\delta_g.$$

By triangle inequality for the Riemannian distance,

$$d(X_{i-1}^{(k)}, X_{i+1}^{(k-1)}) \leqslant d(X_{i-1}^{(k)}, X_i^{(k-1)}) + d(X_i^{(k-1)}, X_{i+1}^{(k-1)}) \leqslant \delta_g,$$

Lemma 3.2 gives that the leapfrog iteration is well defined and produces the unique minimizer

$$X_i^{(k)} = \operatorname*{arg\,min}_{Y \in \mathrm{St}(n,p)} d^2(X_{i-1}^{(k)}, Y) + d^2(Y, X_{i+1}^{(k-1)}).$$

We thus have

$$d^2(X_{i-1}^{(k)}, X_i^{(k)}) + d^2(X_i^{(k)}, X_{i+1}^{(k-1)}) \leqslant d^2(X_{i-1}^{(k)}, X_i^{(k-1)}) + d^2(X_i^{(k-1)}, X_{i+1}^{(k-1)}) \leqslant \tfrac{1}{2}\delta_g^2.$$

Since $X_i^{(k)}$ is the midpoint of the geodesic connecting $X_{i-1}^{(k)}$ to $X_{i+1}^{(k-1)}$, we also have

$$d(X_{i-1}^{(k)}, X_i^{(k)}) = d(X_i^{(k)}, X_{i+1}^{(k-1)}).$$

Combining these two results proves (3.5) until substep i at iteration k. Since $X_0^{(k+1)} = X_0^{(k)} = X_0^{(0)}$, the case for substep $i=1$ and iteration $k+1$ satisifes the same reasoning as above. The same is true for the base case $i = k = 1$, which ends the proof. □

Hence, combining Lemmas 3.5 and 3.6, we get our desired result:

Proposition 3.7. *Let $X^{(0)} \in \mathrm{St}(n,p)^m$ be such that $d(X_{i-1}^{(0)}, X_i^{(0)}) \leqslant \frac{1}{2}\delta_g$ for all $1 \leqslant i \leqslant m$. Then the leapfrog algorithm applied to F is equivalent to the nonlinear Gauss–Seidel method applied to $\widetilde{F}$.*

We can now proceed and analyze the convergence of this nonlinear Gauss–Seidel method using standard theory.

3.2.4 First-order optimality

From Proposition 3.7, we know that at iteration $k \geqslant 1$ and for subinterval $i \in \{1, \ldots, m-2\}$, leapfrog solves the following unconstrained optimization problem

$$\min_{X_i \in \mathbb{R}^{n \times p}} \widetilde{F}_i^k(X_i),$$

where the objective function is defined as

$$\widetilde{F}_i^k(Y) = \widetilde{d}^2(X_{i-1}^{(k)}, Y) + \widetilde{d}^2(Y, X_{i+1}^{(k-1)}).$$

Recall that $X_{i-1}^{(k)}, X_{i+1}^{(k-1)} \in \mathrm{St}(n,p)$ are the neighboring points of X_i and that $X_{i-1}^{(k)}$ was previously updated and that $X_{i+1}^{(k-1)}$ will be updated next.

Let us define

$$\mathcal{G}_i(Y) = \nabla_Y \widetilde{F}_i^k(Y) = \nabla_Y \widetilde{d}^2(X_{i-1}^{(k)}, Y) + \nabla_Y \widetilde{d}^2(X_{i+1}^{(k-1)}, Y).$$

At the minimizer X_i, the gradient of $\widetilde{F}_i^k$ vanishes, i.e., $\mathcal{G}_i(X_i) = 0$. Likewise, if we take all the minimizers $X = (X_1, \dots, X_{m-2})$ together, they will satisfy

$$\begin{cases} \mathcal{G}_1(X) = \nabla_{X_1} \widetilde{d}^2(X_0, X_1) + \nabla_{X_1} \widetilde{d}^2(X_1, X_2) = 0, \\ \mathcal{G}_2(X) = \nabla_{X_2} \widetilde{d}^2(X_1, X_2) + \nabla_{X_2} \widetilde{d}^2(X_2, X_3) = 0, \\ \qquad \vdots \\ \mathcal{G}_{m-2}(X) = \nabla_{X_{m-2}} \widetilde{d}^2(X_{m-3}, X_{m-2}) + \nabla_{X_{m-2}} \widetilde{d}^2(X_{m-2}, X_{m-1}) = 0. \end{cases}$$

This can be written compactly as $\mathcal{G}(X) = 0$, where $\mathcal{G}$ is defined componentwise $\mathcal{G}_i \colon \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}$, for $i = 1, \dots, m-2$.

3.2.5 Known results on local convergence

Assuming convergence to the limit point $X_1^*, X_2^*, \dots, X_{m-2}^*$, the asymptotic convergence rate is determined by the spectral radius of a certain blockwise partitioning of the Hessian of $\widetilde{F}$ at this limit point.

Let $\{X^{(k)}\} \subset \mathbb{R}^n$ be any sequence that converges to X^*. Then

$$R_1\{X^{(k)}\} = \limsup_{k\to\infty} \sqrt[k]{\|X^{(k)} - X^*\|}$$

is the *root-convergence factor* of the sequence [OR00, Definition 9.2.1]. If $\mathcal{I}$ is an iterative process with limit point X^*, and $C(\mathcal{I}, X^*)$ is the set of all sequences generated by $\mathcal{I}$ which converge to X^*, then

$$R_1(\mathcal{I}, X^*) = \sup\left\{ R_1\{X^{(k)}\} \colon \{X^{(k)}\} \in C(\mathcal{I}, X^*) \right\}$$

is the root-convergence factor of $\mathcal{I}$ at X^*.

Theorem 3.8 (Nonlinear block Gauss–Seidel theorem)**.** *Let $\mathcal{G} \colon \mathcal{D} \subset \mathbb{R}^{(m-2)np} \to \mathbb{R}^{(m-2)np}$ be continuously differentiable in an open neighborhood $\mathcal{B}_0 \subset \mathcal{D}$ of a point $X^* \in \mathcal{D}$ for which $\mathcal{G}(X^*) = 0$. Consider the decomposition of $\mathcal{G}'(X) = D - L - U$ into its block diagonal, strictly lower-, and strictly upper-triangular parts, and suppose that $D(X^*)$ is nonsingular and $\rho(M^{\mathrm{BGS}}(X^*)) < 1$, where $M^{\mathrm{BGS}} = (D-L)^{-1}U$. Then there exists an open ball $\mathcal{B} = \mathcal{B}(X^*, \delta)$ in $\mathcal{B}_0$ such that, for any $X^0 \in \mathcal{B}$, there is a unique sequence $\{X^{(k)}\} \subset \mathcal{B}$ which satisfies the nonlinear Gauss–Seidel prescription. Moreover, $\lim_{k\to\infty} X^{(k)} = X^*$ and $R_1(\mathcal{I}, X^*) = \rho(M^{\mathrm{BGS}}(X^*))$.*

Proof. As a direct extension of [OR00, Theorem 10.3.5]. □

This theorem shows the need for the Hessian of $\widetilde{F}$ (i.e., $\mathcal{G}'$) and its block $D - L - U$ decomposition. As we shall see, our matrix $\mathcal{G}'$ is given by the sum of two matrices $\mathcal{G}' = A + E$, where A is symmetric block tridiagonal and positive definite, and E can be regarded as a perturbation matrix. Since it is very difficult to compute the spectral radius of M^{BGS} with this

perturbation E, we will not use Theorem 3.8 directly. Instead, we will use the *Householder–John theorem* [Hac16, Corollary 3.42], which states that if $\mathcal{G}'$ is positive definite, then the M^{BGS} from Theorem 3.8 satisfies $\rho(M^{\mathrm{BGS}}) < 1$. In other words, (linear) block Gauss–Seidel for a symmetric and positive definite $\mathcal{G}'$ always converges monotonically in the energy norm [Hac16, Theorem 3.53]. Therefore, we only need to restrict the perturbation E such that the whole matrix $\mathcal{G}'$ is symmetric and positive definite. In order to do that, we will also use a block version of the Gershgorin circle theorem [FV62, Theorem 2].

3.2.6 Local convergence

As required in Theorem 3.8, we compute the Hessian as the Jacobian matrix $\mathcal{G}'(X)$, a square matrix of size $(m-2)np$. By symmetry of the Hessian, we can write this compactly as

$$\mathcal{G}' = \begin{bmatrix} D_{10}+D_{12} & L_{12}^{\mathsf{T}} & & & & & \\ L_{12} & D_{21}+D_{23} & L_{23}^{\mathsf{T}} & & & & \\ & L_{23} & D_{32}+D_{34} & L_{34}^{\mathsf{T}} & & & \\ & & \ddots & \ddots & & \ddots & \\ & & & L_{m-3,m-2} & D_{m-2,m-3}+D_{m-2,m-1} \end{bmatrix},$$

where

$$L_{ij} = \nabla_{X_i}\nabla_{X_j}\widetilde{d}^2(X_i, X_j) \qquad \text{and} \qquad D_{ij} = \nabla^2_{X_i}\widetilde{d}^2(X_i, X_j)$$

denote the mixed and double derivatives[5].

We now turn to the computation of these derivatives L_{ij} and D_{ij}. To that end, the following lemma is convenient since it writes $\widetilde{d}^2(X_i, X_j)$ as an expansion that does not explicitly use the Riemannian distance.

Lemma 3.9. *Let $\widetilde{X}, \widetilde{Y} \in \mathbb{R}^{n\times p}$ such that $\sigma_p(\widetilde{X}) > 0$ and $\sigma_p(\widetilde{Y}) > 0$, then*

$$\begin{aligned}\widetilde{d}^2(\widetilde{X}, \widetilde{Y}) = \|\mathrm{P}_{\mathrm{St}}\widetilde{X} - \mathrm{P}_{\mathrm{St}}\widetilde{Y}\|_{\mathrm{F}}^2 - \tfrac{1}{2}\|I_p - (\mathrm{P}_{\mathrm{St}}\widetilde{X})^{\mathsf{T}}\mathrm{P}_{\mathrm{St}}\widetilde{Y}\|_{\mathrm{F}}^2 \\ + \|\widetilde{X} - \mathrm{P}_{\mathrm{St}}\widetilde{X}\|_{\mathrm{F}}^2 + \|\widetilde{Y} - \mathrm{P}_{\mathrm{St}}\widetilde{Y}\|_{\mathrm{F}}^2 + O(\|\mathrm{P}_{\mathrm{St}}\widetilde{X} - \mathrm{P}_{\mathrm{St}}\widetilde{Y}\|_{\mathrm{F}}^4).\end{aligned} \tag{3.6}$$

Proof. See Appendix D.2. □

In the following, denote $\delta_{ij} = \|X_i - X_j\|_2$ for any $X_i, X_j \in \mathrm{St}(n,p)$.

Lemma 3.10. *Let $X_i \in \mathrm{St}(n,p)$. Then*

$$D_{ij} = 2I_{np} + \tfrac{1}{2}\left(X_i^T \otimes X_i\right)\Pi_{p,n} - \tfrac{1}{2}\left(I_p \otimes X_i X_i^T\right) + \Delta_{ij}, \tag{3.7}$$

$$L_{ij} = -2I_{np} + \tfrac{1}{2}(X_i^T \otimes X_i)\,\Pi_{p,n} + \tfrac{3}{2}(I_p \otimes X_i X_i^T) + \Lambda_{ij}, \tag{3.8}$$

with $\|\Delta_{ij}\|_2 \leqslant 14\delta_{ij} + 10\delta_{ij}^2$ and $\|\Lambda_{ij}\|_2 \leqslant \frac{11}{2}\delta_{ij} + 10\delta_{ij}^2 + 4\delta_{ij}^3$. Here, $\Pi_{p,n}$ is the vec-permutation matrix defined as the permutation matrix that satisfies $\mathrm{vec}(X) = \Pi_{n,p}\,\mathrm{vec}(X^T)$; see, e.g., [HS81, Eq. (5)].

Proof. See Appendix D.3. □

[5] Observe that $L_{ij} = L_{ji}^{\mathsf{T}}$ by equality of mixed derivatives but in general $D_{ij} \neq D_{ji}^{\mathsf{T}}$ since only the variable corresponding to the first index is derived.

Our aim is to diagonalize $\mathcal{G}'$. We will do this in a few steps. First, observe that $\mathcal{G}'$ remains block-tridiagonal if it is transformed using a compatible block diagonal matrix $\mathcal{Q} = \operatorname{diag}\{Q_1, Q_2, \ldots, Q_{m-2}\}$:

$$\mathcal{Q}^{\mathsf{T}}\mathcal{G}'\mathcal{Q} = \begin{bmatrix} Q_1^{\mathsf{T}}(D_{10}+D_{12})Q_1 & Q_1^{\mathsf{T}}L_{12}^{\mathsf{T}}Q_2 & & & \\ Q_2^{\mathsf{T}}L_{12}Q_1 & Q_2^{\mathsf{T}}(D_{21}+D_{23})Q_2 & Q_2^{\mathsf{T}}L_{23}^{\mathsf{T}}Q_3 & & \\ & Q_3^{\mathsf{T}}L_{23}Q_2 & Q_3^{\mathsf{T}}(D_{32}+D_{34})Q_3 & Q_3^{\mathsf{T}}L_{34}^{\mathsf{T}}Q_4 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \tag{3.9}$$

Here, the $Q_1, \ldots, Q_{m-2} \in \mathbb{R}^{np\times np}$ can be any orthogonal matrices. The lemma below shows us how to choose these matrices so that we obtain diagonal blocks in $\mathcal{Q}^{\mathsf{T}}\mathcal{G}'\mathcal{Q}$, up to first order in δ_{ij}.

Lemma 3.11. *Let $X_i^{\perp} \in \mathbb{R}^{n\times(n-p)}$ be such that $X_i^T X_i^{\perp} = O_{p\times(n-p)}$ and $(X_i^{\perp})^T X_i^{\perp} = I_{(n-p)}$. Define the orthogonal matrices*

$$\bar{Q}_i = \begin{bmatrix} I_p \otimes X_i & I_p \otimes X_i^{\perp} \end{bmatrix},$$

and similarly for $\bar{Q}_j$. Then, there exists an orthogonal matrix $\widehat{Q}$, only depending on n and p, such that $Q_i = \bar{Q}_i\widehat{Q}$ and $Q_j = \bar{Q}_j\widehat{Q}$ satisfy

$$\|Q_i^T D_{ij} Q_i - D\|_2 \leqslant C_D^{(ij)}, \qquad D = \operatorname{diag}\left\{I_{p(p-1)/2},\ 2\,I_{np-p(p-1)/2}\right\}, \tag{3.10}$$

$$\|Q_j^T L_{ij} Q_i - L\|_2 \leqslant C_L^{(ij)}, \qquad L = \operatorname{diag}\left\{-I_{p(p-1)/2},\ -2I_{(n-p)p},\ O_{p(p+1)/2}\right\}, \tag{3.11}$$

where $C_D^{(ij)} = 14\delta_{ij} + 10\delta_{ij}^2$ and $C_L^{(ij)} = \frac{15}{2}\delta_{ij} + \frac{31}{2}\delta_{ij}^2 + 14\delta_{ij}^3 + 4\delta_{ij}^4$.

Proof. See Appendix D.4. □

The matrix $\widehat{Q}$ above is related to the diagonalizaton of the vec-permutation matrix $\mathit{\Pi}_{p,p}$; see (D.10) in Appendix D.4 for its definition. It is therefore also independent of X_i. This is a crucial property to obtain the following result.

Lemma 3.12. *Define $\delta = \max_{0\leqslant i\leqslant m-2}\delta_{i,i+1}$ and assume $\delta \leqslant 1$. Then the minimal eigenvalue of $\mathcal{G}'$ is bounded by*

$$\lambda_{\min}(\mathcal{G}') \geqslant 2 - 2\cos\tfrac{\pi}{m-1} - 43\delta - 90\delta^2.$$

As a consequence, $\mathcal{G}'$ is symmetric and positive definite when

$$\delta < \frac{1}{180}\left(\sqrt{2\,569 - 720\cos\tfrac{\pi}{m-1}} - 43\right).$$

Proof. From Lemma 3.11, recall the diagonal matrices D and L, and the orthogonal matrices $Q_1, \ldots, Q_{m-2}$. Define $\mathcal{Q} = \operatorname{diag}\{Q_1, Q_2, \ldots, Q_{m-2}\}$. Substituting the nonzero blocks in (3.9) by

$$Q_i^{\mathsf{T}}(D_{i,i-1} + D_{i,i+1})Q_i = 2D + E_{ii}, \qquad Q_{i+1}^{\mathsf{T}}L_{i,i+1}Q_i = L + E_{i,i+1},$$

we can write $\mathcal{Q}^{\mathsf{T}}\mathcal{G}'\mathcal{Q}$ as

$$\mathcal{Q}^{\mathsf{T}}\mathcal{G}'\mathcal{Q} = \begin{bmatrix} 2D & L & & \\ L & 2D & L & \\ & \ddots & \ddots & \ddots \end{bmatrix} + \begin{bmatrix} E_{11} & E_{12}^{\mathsf{T}} & & \\ E_{12} & E_{22} & E_{23}^{\mathsf{T}} & \\ & \ddots & \ddots & \ddots \end{bmatrix} =: A + E. \tag{3.12}$$

Equation (3.12) is an approximate tridiagonalization of the matrix $\mathcal{G}'$. Observe that the symmetric matrices A and E have compatible block partitioning. Furthermore, from Lemma 3.11, we get immediately that

$$\|E_{ii}\|_2 \leqslant 28\delta + 20\delta^2 =: C_D, \qquad \|E_{i,i+1}\|_2 \leqslant \tfrac{15}{2}\delta + \tfrac{31}{2}\delta^2 + 14\delta^3 + 4\delta^4 =: C_L.$$

We will regard $\mathcal{Q}^\mathsf{T}\mathcal{G}'\mathcal{Q}$ as an $O(\delta)$ perturbation of A. Using properties of Kronecker products, we can write

$$A = 2I_{m-2} \otimes D + M \otimes L, \qquad M = \begin{bmatrix} 0 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(m-2)\times(m-2)}. \tag{3.13}$$

Thanks to the Kronecker structure in (3.13) and the diagonal matrices D and L, the eigenvalues of A are easily determined as

$$\lambda_{jk} = 2d_j + \mu_k \ell_j, \quad j = 1, \dots, np, \quad k = 1, \dots, m-2,$$

where d_j and ℓ_j are the diagonal entries of D and L, respectively, and μ_k are the eigenvalues of the Toeplitz matrix M. Using [Gov94, Eq. (2.7)], we find

$$\mu_k = -2\cos\tfrac{k\pi}{m-1}, \qquad k = 1, \dots, m-2.$$

Together with (3.10) and (3.11), this allows us to determine that the minimal value among all λ_{jk} corresponds to $j = 1$ and $k = m-2$. We thus obtain

$$\lambda_{\min}(A) = 2 - 2\cos\tfrac{\pi}{m-1} > 0 \quad \text{for all } m \geqslant 2.$$

By Weyl's inequality [SS90, Corollary 4.9], $\lambda_{\min}(\mathcal{G}') = \lambda_{\min}(A+E) > 0$ is guaranteed if $\|E\|_2 < \lambda_{\min}(A)$. To bound $\|E\|_2$, we use a block version of the Gershgorin circle theorem (see [FV62, Theorem 2] and also [Tre08, Remark 1.13.2]). Applied to the symmetric block tridiagonal matrix E, it guarantees that its eigenvalues are included in the union of intervals

$$\bigcup_{i=1}^{m-2} \bigcup_{k=1}^{np} [\varepsilon_k^{(i)} - R_i, \varepsilon_k^{(i)} + R_i], \qquad R_i = \|E_{i-1,i}\|_2 + \|E_{i,i+1}^\mathsf{T}\|_2 \leqslant 2C_L,$$

where $\varepsilon_k^{(i)}$ is the kth eigenvalue of E_{ii}. These eigenvalues $\varepsilon_k^{(i)}$ are all bounded in magnitude by C_D. Hence $\|E\|_2 \leqslant C_D + 2C_L = 43\delta + 51\delta^2 + 28\delta^3 + 8\delta^4$. Since $\delta < 1$, it is easily verified that $\|E\|_2 \leqslant 43\delta + 90\delta^2$ and thus the matrix $\mathcal{G}'$ remains positive definite if $43\delta + 90\delta^2 < \lambda_{\min}(A)$, i.e.,

$$\delta < \frac{1}{180}\left(\sqrt{2\,569 - 720\cos\tfrac{\pi}{m-1}} - 43\right).$$

□

All put together, we have the final result of local convergence.

Theorem 3.13. *If the leapfrog algorithm is started with δ satisfying the condition of Lemma 3.12, then it converges to the unique length-minimizing geodesic connecting X_0 and X_{m-1}, provided that the initial intermediate points are sufficiently close to that geodesic.*

Proof. We use [Hac16, Corollary 3.42] which states that if $\mathcal{G}'$ is positive definite and can be split into the sum of an arbitrary positive definite matrix and an arbitrary symmetric matrix, then the scalar Gauss–Seidel converges, i.e., $\rho(M^{\mathrm{BGS}}) < 1$, and the convergence is monotone with respect to the energy norm $\|\cdot\|_{\mathcal{G}'}$. By [Hac16, Theorem 3.53], we know that this theorem remains valid for any block version.

Now, the splitting (3.12) has exactly the form prescribed by [Hac16, Corollary 3.42], because A is positive definite and E is symmetric. By Lemma 3.12, we know that $\mathcal{G}'$ remains positive definite if $\delta < \frac{1}{180}\left(\sqrt{2\,569 - 720\cos\frac{\pi}{m-1}} - 43\right)$. Under these conditions, the leapfrog algorithm converges as a block Gauss–Seidel method to the length-minimizing geodesic connecting X_0 and X_{m-1}. □

3.3 Some observations and open problems

For m large, Lemma 3.12 gives that $\mathcal{G}'$ is positive definite when $\delta \lesssim \pi^2/43m^2$. Let $d_0 = \|X_0 - X_{m-1}\|_2$ be the distance between the two endpoints. Then by equidistant partitioning of the intermediate points, one has $\delta \simeq d_0/m$. To guarantee a positive definite $\mathcal{G}'$, we would then need $d_0/m \lesssim \pi^2/43m^2$ which implies $m \lesssim 0.23/d_0$.

This result is unsatisfactory, since it would have been desirable to guarantee positive definiteness of $\mathcal{Q}^{\mathsf{T}}\mathcal{G}'\mathcal{Q} = A + E$ with orthogonal $\mathcal{Q}$ by increasing the number of points m given a fixed d_0. Unfortunately, we cannot guarantee this with our proof. The problem is that $\|E\|_2 = O(\delta)$ whereas $\lambda_{\min}(A) = O(1/m^2)$, which lead to our condition that m needed to be smaller than some fixed fraction of the original distance d_0. If $\|E\|_2 = O(\delta^2)$, then there would be no condition on m since $\delta^2 \simeq d_0^2/m^2 \lesssim 1/m^2$ is sufficient to guarantee $\lambda_{\min}(A+E) > 0$. However, it would still not be satisfactory since the perturbation does not lead to an improvement with increasing m, for which one probably needs $\|E\|_2 = O(\delta^3)$. As we show below, there is strong numerical indication that with our choice of extended distance function this is not the case.

Numerical experiments reported in Figure 3.3 suggest that the minimal eigenvalues of $\mathcal{G}'$ and A differ by $O(\delta^2)$, whereas our perturbation analysis only showed $\|E\|_2 = O(\delta)$. It is however not trivial to prove this result. Indeed, up to first order, we can study the eigenvalues of the symmetric matrix $A + E$ by using the derivative formula [SS90, Theorem 2.3]

$$\lambda_{\min}(A + E) = \lambda_{\min}(A) + v_{\min}^{\mathsf{T}} E v_{\min} + O(\|E\|^2), \tag{3.14}$$

where $\lambda_{\min}(A)$ is assumed to be isolated (as it is the case) and $v_{\min}$ is its associated eigenvector. One possibility to improve on our bounds, at least asymptotically, would be to prove that $|v_{\min}^{\mathsf{T}} E v_{\min}| = O(\delta^3)$. However, in the same figure, $|v_{\min}^{\mathsf{T}} E v_{\min}|$ seems to be again $O(\delta^2)$. In addition, all these conclusions remain true in the limiting geodesic.

Another problem with the matrix A and $\mathcal{G}'$ is that it has a bad spectral gap γ (i.e., the difference of smallest and second smallest eigenvalue) when m grows. Numerical observations suggest that the spectral gap might be $O(1/m)$ which complicates non-asymptotic bounds.

As a last remark, one could resort to a more general theory for the convergence of nonlinear block Gauss–Seidel for a quasi-convex objective function [GS00], which requires quasi--convexity for each X_i alone. Looking at the Hessian $\mathcal{G}'$ where all X_j except X_i are constant, the only block that is left in the matrix $\mathcal{G}'$ is the diagonal one, namely $D_{i,i-1} + D_{i,i+1}$. Using Lemma 3.11, we immediately get the eigenvalues of this block. Now, for $C_D^{(ij)} < 1$ in (3.10) we get strong convexity in X_i alone. One problem with this approach is that the feasible set has to be a Cartesian product of convex subsets of $\mathbb{R}^{n\times p}$. Moreover, the result in [GS00] only

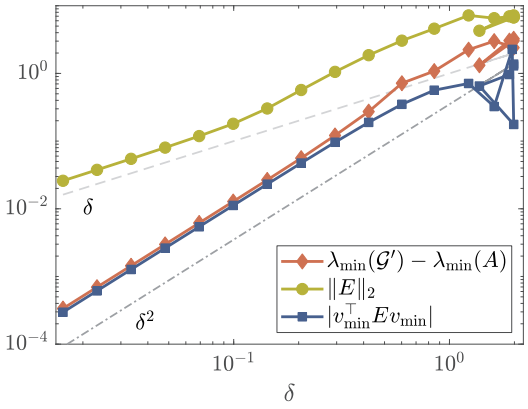


Figure 3.3 – Eigenvalue perturbations – not at the limiting geodesic.

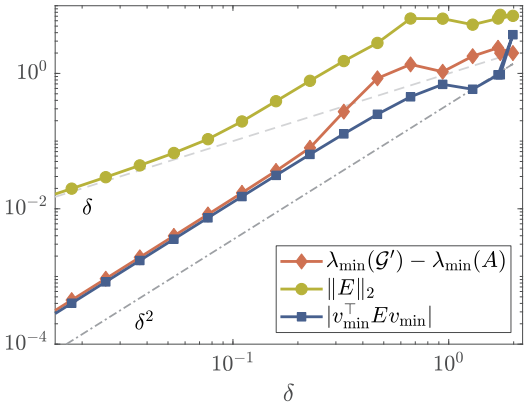


Figure 3.4 – Eigenvalue perturbations – at the limiting geodesic.

guarantees subsequence convergence, and there is no rate of convergence or contraction rate for the whole sequence. Hence the convergence behavior could also be slower than linear.

3.4 Numerical experiments

As a concrete example to demonstrate the leapfrog algorithm, let us consider the Stiefel manifold $\mathrm{St}(12, 3)$. We fix one point $X = [I_3 \;\; O_{9\times 3}]^\mathsf{T}$, while the other point Y is placed at the distance $L^* = 0.95\,\pi$ from X. This is done by creating a tangent vector to $\mathrm{St}(12, 3)$ at X of length L^*, and then mapping it to $\mathrm{St}(12, 3)$ via the Riemannian exponential (2.2). For this choice of L^*, single shooting will not work (recall that the injectivity radius on $\mathrm{St}(n, p)$ is at least $0.89\,\pi$). We want to recover this distance using the leapfrog algorithm and study its convergence.

For each value of $m \in \{10, 20, 50, 100\}$, we construct an initial guess $X^{(0)}$ by placing $m - 2$ intermediate points randomly along the linear segment connecting X and Y in the embedding space, and projecting them to the Stiefel manifold. We then apply leapfrog for 300 iterations and monitor the convergence behavior of

$$\text{err-}k = \|X^{(k)} - X^*\|_\mathrm{F},$$

where X^* is the solution of leapfrog (i.e., a uniformly distributed tuple corresponding to the global geodesic that was constructed above), and $X^{(k)}$ is the approximate solution at iteration k of leapfrog. This is illustrated in Figure 3.5, from which it is clear that for large m leapfrog always converges albeit very slowly.

Next, we apply leapfrog for 50 iterations and for each $m \in \{4, 6, 8, 10, \ldots, 100\}$ we repeat this experiment for 100 random initializations of the initial guess $X^{(0)}$. For each experiment i, we define the error reduction rate[6] as

$$\mu_k^{(i)} = \frac{\text{err-}(k+1)}{\text{err-}k}, \quad \text{for} \quad k = 0, 1, \ldots, 49, \quad i = 1, \ldots, 100,$$

and we compute the worst and the median reduction rates across all the experiments, namely, $\max_{i,k}\{\mu_k^{(i)}\}$ and $\mathrm{med}_i \max_k\{\mu_k^{(i)}\}$. Since during the first iterations leapfrog is faster, we also compute the convergence factor given by $\max_i\{\mu_0^{(i)}\}$.

[6] In the limit $k \to \infty$, this gives the asymptotic Q-rate of convergence of the sequence.

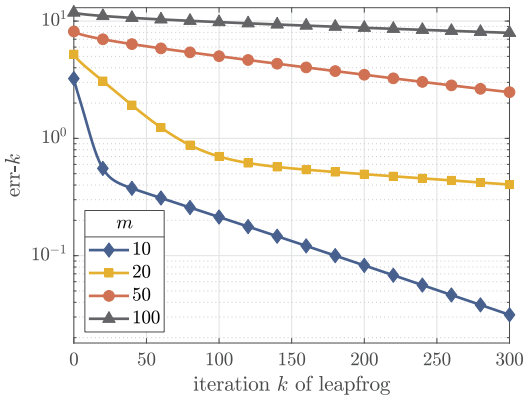


Figure 3.5 – Convergence behavior of err-k for increasing values of m.

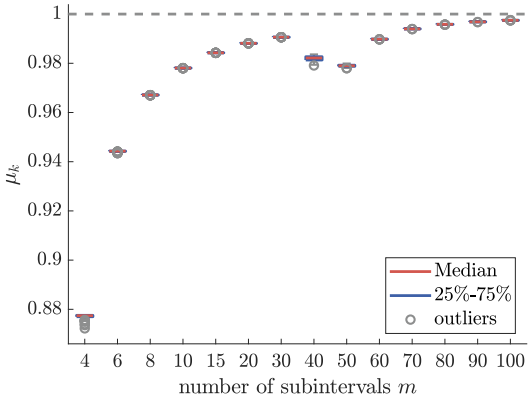


Figure 3.6 – Boxplot of $\max_k\{\mu_k^{(i)}\}$ for increasing values of m.

From Table 3.1, we see that the convergence of leapfrog deteriorates as m increases but it remains strictly smaller than 1. For small values of m, $\max_i\{\mu_0^{(i)}\}$ and $\max_{i,k}\{\mu_k^{(i)}\}$ are significantly different, whereas for large values of m, they are quite similar. The same conclusion can be reached from Figure 3.6 where boxplots show the dispersion and skewness in the $\mu_k^{(i)}$. Clearly, the convergence factors become very concentrated for large m.

Table 3.1 – Values of $\max_i\{\mu_0^{(i)}\}$, $\max_{i,k}\{\mu_k^{(i)}\}$ and $\mathrm{med}_i \max_k\{\mu_k^{(i)}\}$ versus number of points m, for the experiment described in Section 3.4.

m	4	6	8	10	15	20	30
$\max_i\{\mu_0^{(i)}\}$	0.5577	0.7058	0.7829	0.8296	0.8604	0.8824	0.8980
$\max_{i,k}\{\mu_k^{(i)}\}$	0.8776	0.9443	0.9671	0.9781	0.9843	0.9881	0.9906
$\mathrm{med}_i \max_k\{\mu_k^{(i)}\}$	0.8774	0.9443	0.9671	0.9781	0.9843	0.9881	0.9906
m	40	50	60	70	80	90	100
$\max_i\{\mu_0^{(i)}\}$	0.9390	0.9573	0.9728	0.9799	0.9843	0.9870	0.9888
$\max_{i,k}\{\mu_k^{(i)}\}$	0.9836	0.9799	0.9898	0.9940	0.9959	0.9969	0.9976
$\mathrm{med}_i \max_k\{\mu_k^{(i)}\}$	0.9822	0.9790	0.9898	0.9940	0.9958	0.9968	0.9975

Chapter 4

Extensions on leapfrog

In this chapter, we recall and work on some concepts that relate to the original definition of distance on a Riemannian manifold $\mathcal{M}$. As we have seen in Section 1.1.10, the Riemannian distance between two points is defined as the minimum value of the length functional over the set of all curves in $\mathcal{M}$ joining those two points. We will have a closer look at what the length and energy functionals are and how they can be useful in the context of numerical algorithms that calculate the Riemannian distance. Most of these concepts go back to Milnor [Mil63], and the leapfrog algorithm of Noakes also builds upon these notions.

4.1 Broken geodesics, length and energy functional

Let $\mathcal{M}$ be a Riemannian manifold, and X_0, X_{m-1} two points of $\mathcal{M}$. Consider a *broken geodesic* connecting X_0 and X_{m-1}, identified by the m-tuple

$$X \equiv (X_0, \ldots, X_{m-1}) \in \mathcal{M}^m.$$

From [Noa98, Definition 3.2], the curve $\omega_X(t)\colon [0,1] \to \mathcal{M}$ is defined as

$$\omega_X(t) = \gamma_i \left(\frac{t\, L(X) - \sum_{j=1}^{i-1} d(X_{j-1}, X_j)}{d(X_{i-1}, X_i)} \right),$$

with $\gamma_i\colon [0,1] \to \mathcal{M}$ being the minimizing geodesic from X_{i-1} to X_i. Figure 4.1 provides an illustration of a broken geodesic through four points.

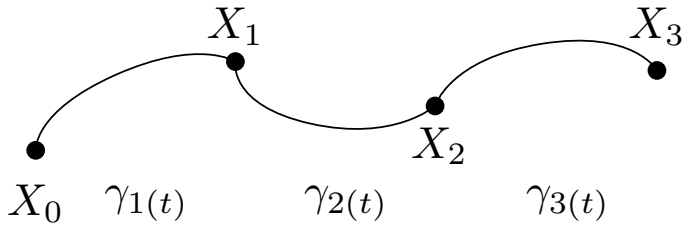


Figure 4.1 – A broken geodesic ω_X.

The *length functional* is given by the sum of the lengths of all geodesic segments, namely,

$$L(X) = \sum_{i=1}^{m-1} d(X_{i-1}, X_i). \tag{4.1}$$

The *energy functional* is defined as [Mil63, III.§16]

$$E(X) = \sum_{i=1}^{m-1} \frac{d^2(X_{i-1}, X_i)}{t_i - t_{i-1}},$$

where $t \in [t_{i-1}, t_i]$, t_i, and the junction times

$$t_i = \frac{\sum_{j=1}^{i} d(X_{j-1}, X_j)}{L(X)}, \qquad i = 1, \ldots, m-1, \qquad t_0 = 0. \tag{4.2}$$

If the parameter t is proportional to arc-length along ω_X, then one has the equality $L^2(X) = E(X)$ [Mil63, III.§12]. Indeed, using the above definitions, one has

$$E(X) = \sum_{i=1}^{m-1} \frac{d^2(X_{i-1}, X_i)}{\frac{\sum_{j=1}^{i} d(X_{j-1}, X_j)}{L(X)} - \frac{\sum_{j=1}^{i-1} d(X_{j-1}, X_j)}{L(X)}} = L(X) \sum_{i=1}^{m-1} \frac{d^2(X_{i-1}, X_i)}{d(X_{i-1}, X_i)} = L^2(X).$$

The important Corollary 12.3 in [Mil63, p. 72] states that *a path ω_X is a critical point for the energy functional E if and only if ω_X is a geodesic.*

4.2 Comparison between steepest descent and leapfrog

In this section, we formulate the problem of minimizing the energy functional from a steepest descent point of view. Given the endpoints X_0 and X_{m-1}, we want to find the $(m-2)$-tuple of junction points $\mathbb{X} \equiv (X_1, \ldots, X_{m-2})$ such that

$$\min_{\mathbb{X} \in \mathcal{M}^{m-2}} E(X), \quad \text{with} \quad X \equiv (X_0, \mathbb{X}, X_{m-1}).$$

Since $\mathcal{M}^{m-2}$ is a Cartesian product of $\mathcal{M}$, the Riemannian gradient of $E(X)$ is an $(m-2)$-tuple of tangent vectors at the junction points

$$\operatorname{grad}_{\mathbb{X}} E(X) = \left(\operatorname{grad}_{X_1} E(X), \ldots, \operatorname{grad}_{X_{m-2}} E(X) \right). \tag{4.3}$$

From [Kar77, Eq. (1.2.1)], we know that the Riemannian gradient of the Riemannian distance squared is given by

$$\operatorname{grad}_X d^2(X, Y) = -2 \operatorname{Log}_X(Y), \tag{4.4}$$

where $\operatorname{Log}_X(Y)$ is the Riemannian logarithm of Y at X. By using this result, each gradient in the tuple (4.3) is given by

$$\operatorname{grad}_{X_i} E(X) = -2 \left(\frac{\operatorname{Log}_{X_i}(X_{i-1})}{t_i - t_{i-1}} + \frac{\operatorname{Log}_{X_i}(X_{i+1})}{t_{i+1} - t_i} \right), \qquad i = 1, \ldots, m-2,$$

or, equivalently,

$$\operatorname{grad}_{X_i} E(X) = -2L(X) \left(\frac{\operatorname{Log}_{X_i}(X_{i-1})}{d(X_{i-1}, X_i)} + \frac{\operatorname{Log}_{X_i}(X_{i+1})}{d(X_i, X_{i+1})} \right), \qquad i = 1, \ldots, m-2.$$

The steepest descent method reads: for $k \geqslant 0$, until a stopping criterion is met, compute

$$X_i^{(k+1)} = \operatorname{Exp}_{X_i^{(k)}}(\alpha^{(k)} \eta_i^{(k)}), \qquad i = 1, \ldots, m-2, \tag{4.5}$$

where $\eta_i^{(k)} = -\operatorname{grad}_{X_i^{(k)}} E(X^{(k)}) \in T_{X_i^{(k)}}\mathcal{M}$ and $\alpha^{(k)}$ is the step size, which is computed via a line-search technique.

Using the same notation, we can formally express the leapfrog update as in (4.5). Recall from Section 3.1.1 that the midpoint map $\mathfrak{M} : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ is defined by

$$\mathfrak{M}(X, Y) = \operatorname{Exp}_X\left(\tfrac{1}{2}\operatorname{Log}_X(Y)\right).$$

One complete iteration (indexed by k) of leapfrog comprises $m-2$ inner iterations, indexed by j. Starting from $X^{(0)} = \left(X_0^{(0)}, X_1^{(0)}, X_2^{(0)}, \ldots, X_{m-1}^{(0)}\right)$, for $k \geqslant 0$, until a stopping criterion is met, leapfrog computes

$$X_j^{(k+1)} = \mathfrak{M}\left(X_{j-1}^{(k+1)}, X_{j+1}^{(k)}\right), \qquad \eta_j^{(k+1)} = \operatorname{Log}_{X_j^{(k)}} X_j^{(k+1)},$$

for $j = 1, \ldots, m-2$. Numerical experiments show that the sequence of update directions $\{\eta^{(k)}\}$, with $\eta^{(k)} = \left(\eta_1^{(k)}, \eta_2^{(k)}, \ldots, \eta_{m-2}^{(k)}\right)$, generated by the leapfrog algorithm is *gradient related*[1], namely for any subsequence $\{\mathbb{X}^{(k)}\}_{k\in\mathcal{K}}$ of $\{\mathbb{X}^{(k)}\}$ that converges to a non-critical point of E, the corresponding subsequence $\{\eta^{(k)}\}_{k\in\mathcal{K}}$ is bounded and satisfies

$$\lim_{k\to\infty} \sup_{k\in\mathcal{K}} \left\langle \operatorname{grad}_{\mathbb{X}^{(k)}} E(X^{(k)}),\ \eta^{(k)} \right\rangle < 0. \tag{4.6}$$

However, the uniform angle condition from [BAC18, Lemma 2.10] is not satisfied.

We emphasize that steepest descent, in contrast to leapfrog, does not converge to a uniformly distributed tuple (see Chapter 3). Indeed, there is nothing in the theory of steepest descent that would imply it to converge to a uniformly distributed tuple, whereas leapfrog instead makes the specific choice of the midpoint map. In the next section, we will illustrate this difference with a concrete example.

4.2.1 Steepest descent on the unit sphere

For an easier comparison, we detail the derivation of the steepest descent direction and the leapfrog direction for the unit sphere S^{n-1} endowed with the standard inner product on $\mathbb{R}^n$. Geodesics are given by

$$x(t) = x\cos(\|\dot{x}_0\|\, t) + \frac{\dot{x}_0}{\|\dot{x}_0\|}\sin(\|\dot{x}_0\|\, t),$$

where $x \equiv x(0)$ and $\dot{x}_0 \equiv \dot{x}(0)$. Therefore the Riemannian exponential at x is

$$y = \operatorname{Exp}_x(\dot{x}_0) = x(1) = x\cos(\|\dot{x}_0\|) + \frac{\dot{x}_0}{\|\dot{x}_0\|}\sin(\|\dot{x}_0\|),$$

and the corresponding Riemannian logarithm is

$$\operatorname{Log}_x(y) = \dot{x}_0 = \arccos(x^{\mathsf{T}}y)\frac{\mathrm{P}_x\, y}{\|\mathrm{P}_x\, y\|},$$

where $y \equiv x(1)$ and P_x is the projector onto $\big(\operatorname{span}(x)\big)^{\perp}$, i.e., $\mathrm{P}_x = I - xx^{\mathsf{T}}$.

Figure 4.2 provides an illustration of these objects for the unit sphere S^2.

[1] See Definition 1.37. For an illustration of this concept, we refer the reader to Figure 1.6.

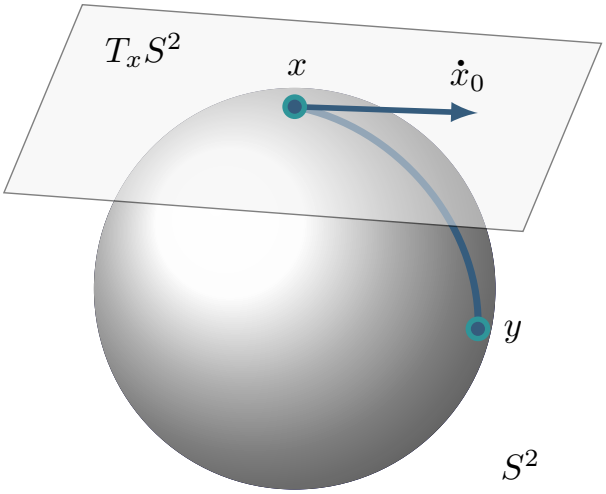


Figure 4.2 – Illustration of a sphere S^2 with its tangent plane at x, the geodesic connecting x and y, and the corresponding tangent vector.

The distance between two points $x,\ y \in S^{n-1}$ is the norm of $\operatorname{Log}_x(y)$, i.e.,

$$d(x, y) = \|\operatorname{Log}_x(y)\| = \arccos(x^\mathsf{T} y).$$

Therefore the Euclidean gradient of $d^2(x, y)$ with respect to x is

$$\nabla_x d^2(x, y) = -2 \arccos(x^\mathsf{T} y) \frac{y}{\sqrt{1 - (x^\mathsf{T} y)^2}}.$$

The Riemannian gradient is given by a projection of the Euclidean gradient onto $T_x S^{n-1}$

$$\mathrm{P}_x \nabla_x d^2(x, y) = -2 \arccos(x^\mathsf{T} y) \frac{\mathrm{P}_x\, y}{\sqrt{1 - (x^\mathsf{T} y)^2}} = -2\ \operatorname{Log}_x(y).$$

Incidentally, this result verifies (4.4) for the particular case of S^{n-1}.

Example. Consider a tuple $X = (x_0,\ x_1,\ x_2)$ on the unit sphere S^{n-1}. The associated energy functional is

$$E(X) = L^2(X) = \Big(d(x_0, x_1) + d(x_1, x_2)\Big)^2,$$

whose Riemannian gradient with respect to the junction point x_1 is

$$\begin{aligned}
\operatorname{grad}_{x_1} E(X) &= -2\big(d(x_0, x_1) + d(x_1, x_2)\big) \left(\frac{\operatorname{Log}_{x_1}(x_0)}{d(x_0, x_1)} + \frac{\operatorname{Log}_{x_1}(x_2)}{d(x_1, x_2)} \right) \\
&= -2 \Big(\arccos(x_0^\mathsf{T} x_1) + \arccos(x_1^\mathsf{T} x_2) \Big) \left(\frac{\mathrm{P}_{x_1} x_0}{\|\mathrm{P}_{x_1} x_0\|} + \frac{\mathrm{P}_{x_1} x_2}{\|\mathrm{P}_{x_1} x_2\|} \right) \\
&= -2 \Big(\arccos(x_0^\mathsf{T} x_1) + \arccos(x_1^\mathsf{T} x_2) \Big)\, \mathrm{P}_{x_1} \left(\frac{x_0}{\|\mathrm{P}_{x_1} x_0\|} + \frac{x_2}{\|\mathrm{P}_{x_1} x_2\|} \right).
\end{aligned}$$

Then steepest descent uses the direction $\eta = -\operatorname{grad}_{x_1} E(X)$.

For a direct comparison, let us compute the leapfrog update vector. With the tangent vector

$$\dot{x}_0 = \mathrm{Log}_{x_0}(x_2) = \arccos(x_0^\mathsf{T} x_2)\,\frac{\mathrm{P}_{x_0} x_2}{\|\mathrm{P}_{x_0} x_2\|},$$

the midpoint map gives the new point

$$\begin{aligned}
\widetilde{x}_1 &= \mathrm{Exp}_{x_0}(\tfrac{1}{2}\dot{x}_0) \\
&= x_0 \cos(\tfrac{1}{2}\|\dot{x}_0\|) + \frac{\dot{x}_0}{\|\dot{x}_0\|}\sin(\tfrac{1}{2}\|\dot{x}_0\|) \\
&= x_0 \cos(\tfrac{1}{2}\arccos(x_0^\mathsf{T} x_2)) + \frac{\mathrm{P}_{x_0} x_2}{\|\mathrm{P}_{x_0} x_2\|}\sin(\tfrac{1}{2}\arccos(x_0^\mathsf{T} x_2)) \\
&= x_0\sqrt{\frac{x_0^\mathsf{T} x_2 + 1}{2}} + \frac{\mathrm{P}_{x_0} x_2}{\|\mathrm{P}_{x_0} x_2\|}\sqrt{\frac{1 - x_0^\mathsf{T} x_2}{2}},
\end{aligned}$$

where we used trigonometric formulas to develop the second-last line. Finally, the leapfrog update vector is given by

$$\eta_{\mathrm{lf}} = \mathrm{Log}_{x_1}(\widetilde{x}_1) = \arccos(x_1^\mathsf{T}\widetilde{x}_1)\frac{\mathrm{P}_{x_1}\widetilde{x}_1}{\|\mathrm{P}_{x_1}\widetilde{x}_1\|}.$$

This example shows that the direction of the leapfrog update is different from the direction of steepest descent.

To illustrate this difference even further, let us focus on the special case of the unit sphere S^2 embedded in $\mathbb{R}^3$. The points on S^2 are parametrized according to spherical coordinates, namely,

$$x_i = \begin{bmatrix} \sin\phi_i \cos\theta_i \\ \sin\phi_i \sin\theta_i \\ \cos\phi_i \end{bmatrix}, \qquad \theta_i \in [0, 2\pi), \qquad \phi_i \in [0, \pi).$$

Here, θ is the azimuthal angle, while ϕ is the polar angle. Let us fix some $\theta_0,\ \phi_0$ for x_0 and $\theta_2,\ \phi_2$ for x_2, and let $\theta_1 \in [0, 2\pi)$ and $\phi_1 \in (\phi_0, \phi_2)$. Figure 4.3 shows the Riemannian gradient vector field (in blue) and the leapfrog update vector field (in red) for some points on the unit sphere S^2. The thick green line is the minimizing geodesic connecting x_0 and x_2.

Let us define the quantity

$$\widehat{\alpha} = \left\langle \frac{\mathrm{grad}_{x_1} E(x_1)}{\|\mathrm{grad}_{x_1} E(x_1)\|},\ \frac{\eta_{\mathrm{lf}}}{\|\eta_{\mathrm{lf}}\|} \right\rangle,$$

which provides a measure of the angle between the Riemannian gradient $\mathrm{grad}_{x_1} E(x_1)$ and the leapfrog update vector η_{lf}. A value of $\widehat{\alpha}$ equal to zero means that these two directions are perpendicular to each other. In Figure 4.4, the quantities $\|\mathrm{grad}_{X_1} E(X_1)\|$ and $\widehat{\alpha}$ are plotted as a function of $\theta_1,\ \phi_1$.

From Figures 4.3 and 4.4 it is evident that if x_1 approaches the geodesic, then $\widehat{\alpha} \to 0$, in other words, the vectors $\mathrm{grad}_{x_1} E(x_1)$ and η_{lf} are nearly perpendicular. As we mentioned above, this situation is due to the fact that the two methods try to achieve two different aims: steepest descent is just looking for the minimizing geodesic, no matter the location of x_1, while leapfrog also tries to place x_1 at the midpoint of the geodesic.

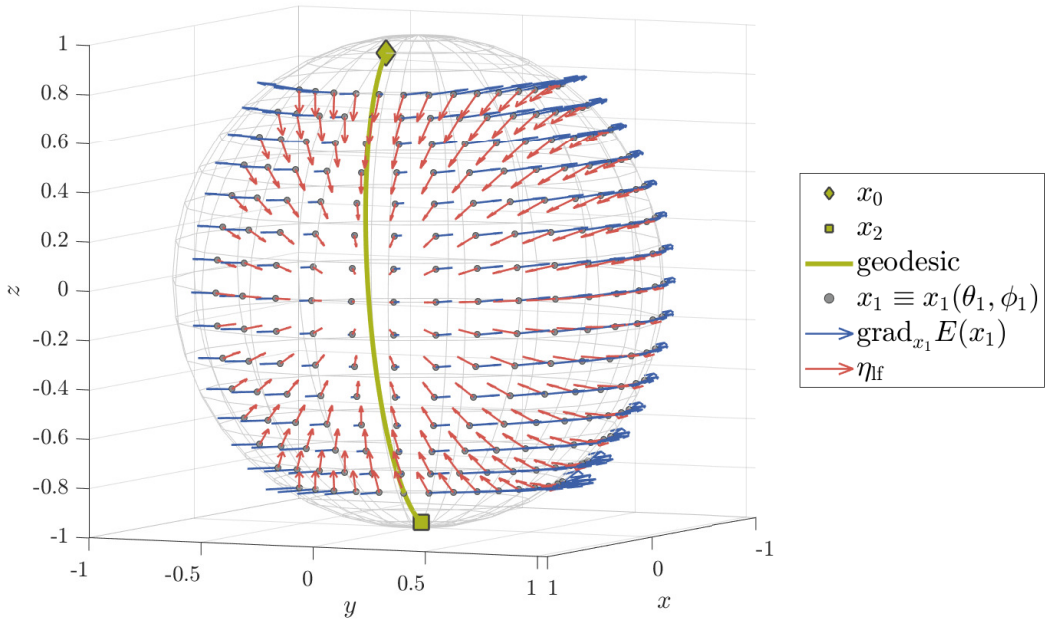


Figure 4.3 – Gradient and leapfrog vector fields on the unit sphere S^2.

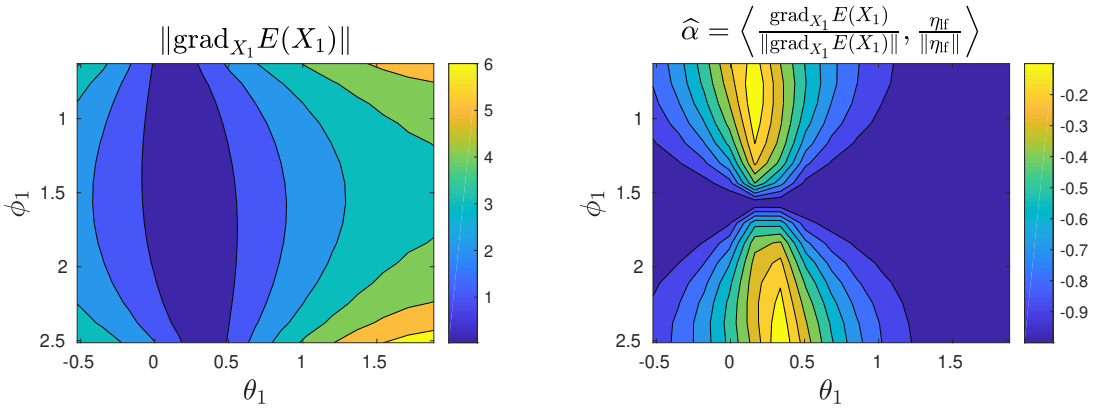


Figure 4.4 – Contours on S^2 for the gradient norm and $\widehat{\alpha}$ as a function of $\theta_1,\ \phi_1$.

4.2.2 Gradient-related sequence in Euclidean space

In this section, we will explicitly verify that, in a Euclidean space with the standard inner product $\langle x, y\rangle = x^{\mathsf{T}}y$, the sequence of update directions generated by the leapfrog algorithm is gradient related. In other words, we will verify that the term $\langle \mathrm{grad}_{X^{(k)}} E(X^{(k)}),\ \eta^{(k)}\rangle$ appearing in (4.6) is strictly negative for all $k \geqslant 1$.

The motivation to do this comes from the fact that, for embedded submanifolds with a Riemannian metric inherited from the embedding space, the Euclidean distance is equal to the Riemannian distance up to third-order terms (see Appendix D.1).

Given $x, y \in \mathbb{R}^n$, the logarithm $\mathrm{Log}_x(y)$ is the vector pointing to y with base point x, namely,

$$\mathrm{Log}_x(y) = y - x.$$

As usual, the distance between x and y is the norm of the logarithm

$$d(x, y) = \|\mathrm{Log}_x(y)\| = \|y - x\|,$$

whose Euclidean gradient with respect to x is

$$\nabla_x d(x, y) = -\frac{\mathrm{Log}_x(y)}{d(x, y)} = -\frac{y - x}{\|y - x\|},$$

Now let us consider a piecewise path (a, x, b) in $\mathbb{R}^2$, as illustrated in Figure 4.5.

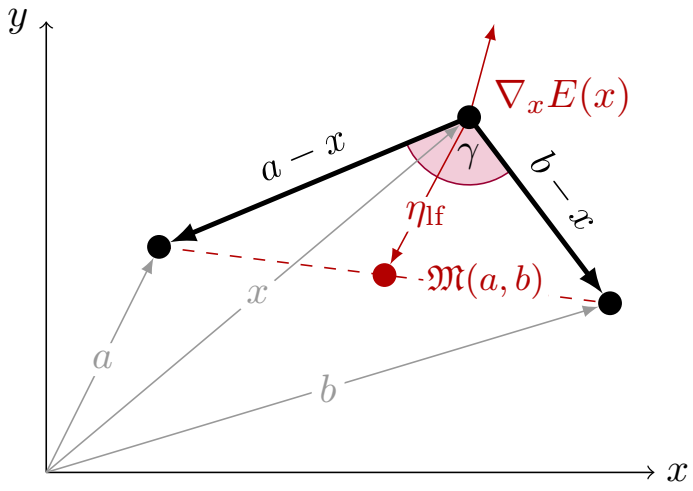


Figure 4.5 – Leapfrog update vector in $\mathbb{R}^2$.

The energy of this path is given by

$$E(x) = L^2(x) = \left(\|a - x\| + \|b - x\|\right)^2.$$

The gradient of $E(x)$ with respect to x is given by

$$\nabla_x E(x) = -2(\|a - x\| + \|b - x\|)\left(\frac{a - x}{\|a - x\|} + \frac{b - x}{\|b - x\|}\right).$$

Observe that the direction of $\nabla_x E(x)$ coincides with the direction of the angle bisector of γ, where γ denotes the angle opposite side $b - a$ (see Figure 4.5).

The midpoint map is simply given by

$$\mathfrak{M}(a,b) = \frac{a+b}{2},$$

and hence the leapfrog update vector is

$$\eta_{\text{lf}} = \operatorname{Log}_x \mathfrak{M}(a,b) = \frac{a+b}{2} - x.$$

We want to check whether the quantity $\widehat{\alpha} = \langle \nabla_x E(x),\ \eta_{\text{lf}} \rangle$ is strictly negative or not.

$$\begin{aligned}\widehat{\alpha} &= -2 \underbrace{\left(\|a-x\| + \|b-x\|\right)}_{=:C>0} \left\langle \frac{a-x}{\|a-x\|} + \frac{b-x}{\|b-x\|}, \frac{a+b}{2} - x \right\rangle \\ &= -2C \left(\frac{\langle a-x, a+b\rangle}{2\|a-x\|} - \frac{\langle a-x, x\rangle}{\|a-x\|} + \frac{\langle b-x, a+b\rangle}{2\|b-x\|} - \frac{\langle b-x, x\rangle}{\|b-x\|} \right).\end{aligned}$$

After some manipulations, we get

$$\widehat{\alpha} = -2C \left(\frac{\|a\|^2 + 2\|x\|^2 - 3\langle a,x\rangle + \langle a,b\rangle - \langle b,x\rangle}{2\|a-x\|} + \frac{\|b\|^2 + 2\|x\|^2 - 3\langle b,x\rangle + \langle a,b\rangle - \langle a,x\rangle}{2\|b-x\|} \right).$$

Using the law of cosines $\|a-x\|^2 = \|a\|^2 + \|x\|^2 - 2\langle a,x\rangle$, $\|b-x\|^2 = \|b\|^2 + \|x\|^2 - 2\langle b,x\rangle$, and the properties of the inner product $\langle a,b\rangle - \langle a,x\rangle - \langle b,x\rangle + \|x\|^2 = \langle a, b-x\rangle - \langle x, b-x\rangle = \langle a-x, b-x\rangle = \|a-x\|\|b-x\|\cos\gamma$, one can obtain

$$\begin{aligned}\widehat{\alpha} &= -2C \left(\frac{\|a-x\|^2 + \|a-x\|\|b-x\|\cos\gamma}{2\|a-x\|} + \frac{\|b-x\|^2 + \|a-x\|\|b-x\|\cos\gamma}{2\|b-x\|} \right) \\ &= -2C \left(\frac{\|a-x\| + \|b-x\|\cos\gamma}{2} + \frac{\|b-x\| + \|a-x\|\cos\gamma}{2} \right) \\ &= -C^2(1+\cos\gamma).\end{aligned}$$

Finally, one gets the condition

$$-C^2(1+\cos\gamma) < 0 \quad \Longleftrightarrow \quad (1+\cos\gamma) > 0 \quad \Longleftrightarrow \quad \gamma \neq \pi.$$

Thus, when considering Euclidean space, the sequence of update directions generated by the leapfrog algorithm is always gradient related, unless x_1 lies already on a geodesic, which is the case when $\gamma = \pi$. But this is the case of a critical point, to which the definition of gradient-related sequence does not apply. Indeed, in such a case, the problem is already solved and there is nothing to do.

What we did in the last section is indeed a big simplification. We only considered $\mathbb{R}^2$ and only one junction point as a variable (i.e., only two subintervals), hence ignoring what happens when we have more than one junction point. However, this big simplification did allow us to verify two main things, namely: that the leapfrog update vector is different from the negative gradient of the energy functional, and that the leapfrog update vector remains gradient related.

4.3 Convergence to uniformly distributed tuple

One of the main properties of leapfrog is the convergence to a uniformly distributed tuple. In other words, at convergence, the broken geodesic not only is a globally C^1 geodesic, but its junction points are also equally spaced from each other. Here, we reformulate the result of [Noa98, Lemma 3.2] for convenience.

Lemma 4.1 (Convergence to uniformly distributed tuple)**.** *Let ω_X be a geodesic γ. Then the sequence of iterates generated by leapfrog converges to the uniformly distributed m-tuple*

$$\Big(X_0, \gamma(\tfrac{1}{m-1}), \gamma(\tfrac{2}{m-1}), \dots, \gamma(\tfrac{i}{m-1}), \dots, X_{m-1}\Big).$$

To further investigate this property, let us consider the tangent vectors associated to this tuple, i.e., $(\xi_0, \xi_1, \dots, \xi_{m-1})$. Then the result of Lemma 4.1 is equivalent to saying that, at convergence, the tangent vectors all have the same length, i.e.,

$$\|\xi_i\| = \|\xi_{i+1}\|, \quad i = 0, \dots, m-3.$$

The main assumption of Lemma 4.1 is that ω_X is already a geodesic γ. This implies that its length $L_\gamma = L(X)$ does not vary under further leapfrog iterations, but only the distribution of the points X_i along the geodesic γ may change. So another way to define the junction times at iteration k is

$$t_i^{(k)} = \frac{\sum_{j=0}^{i-1} \|\xi_j^{(k)}\|}{L_\gamma}, \qquad i = 1, \dots, m-1, \quad \text{with} \quad t_0^{(k)} = 0,\ t_{m-1}^{(k)} = 1,\ \forall k. \tag{4.7}$$

Here, we are going to verify the recursion

$$t_i^{(k)} = \tfrac{1}{2}\Big(t_{i-1}^{(k)} + t_{i+1}^{(k-1)}\Big), \tag{4.8}$$

which appears in the proof of [Noa98, Lemma 3.1]. Observe that the midpoint map acts on the lengths of the geodesic segments as

$$\|\xi_i^{(k)}\| = \tfrac{1}{2}\Big(\|\xi_{i-1}^{(k)}\| + \|\xi_{i+1}^{(k-1)}\|\Big), \quad i = 0, \dots, m-2, \quad \text{with} \quad \xi_{-1}^{(k)} = \xi_0^{(k-1)}, \quad k > 0. \tag{4.9}$$

Starting from

$$\|\xi_{-1}^{(k)}\| = \|\xi_0^{(k-1)}\|,$$

and using the identities

$$\sum_{j=-1}^{i-2} \|\xi_j^{(k)}\| - \sum_{j=0}^{i-2} \|\xi_j^{(k)}\| = \sum_{j=0}^{i} \|\xi_j^{(k-1)}\| - \sum_{j=1}^{i} \|\xi_j^{(k-1)}\|,$$

$$\sum_{j=-1}^{i-2} \|\xi_j^{(k)}\| + \sum_{j=1}^{i} \|\xi_j^{(k-1)}\| = \sum_{j=0}^{i-2} \|\xi_j^{(k)}\| + \sum_{j=0}^{i} \|\xi_j^{(k-1)}\|,$$

rearranging indices we get

$$\sum_{j=0}^{i-1} \|\xi_{j-1}^{(k)}\| + \sum_{j=0}^{i-1} \|\xi_{j+1}^{(k-1)}\| = \sum_{j=0}^{i-2} \|\xi_j^{(k)}\| + \sum_{j=0}^{i} \|\xi_j^{(k-1)}\|.$$

Using (4.9) on the left-hand side

$$2\sum_{j=0}^{i-1} \|\xi_j^{(k)}\| = \sum_{j=0}^{i-2} \|\xi_j^{(k)}\| + \sum_{j=0}^{i} \|\xi_j^{(k-1)}\|,$$

then dividing by L_γ and using the definition (4.7), we get the recursion (4.8).

4.3.1 The stochastic matrix

Let us make some further considerations on the leapfrog algorithm. The midpoint map guarantees the following recursive inequality concerning the lengths of the geodesic segments:

$$\|\xi_i^{(k)}\| \leqslant \tfrac{1}{2}\Big(\|\xi_{i-1}^{(k)}\| + \|\xi_{i+1}^{(k-1)}\|\Big), \quad i = 0, \dots, m-2, \quad \text{with} \quad \xi_{-1}^{(k)} = \xi_0^{(k-1)}, \quad k > 0.$$

This is almost the same as (4.9), the only difference being the equality replaced by the inequality since, in general, ω_X is not a geodesic. The inequality is more general than (4.9), and always true due to the way the leapfrog iterates are constructed.

Making the recursion explicit, one can obtain the following inequalities (the $\leqslant$ sign has to be interpreted elementwise)

$$\begin{bmatrix} \|\xi_0^{(k+1)}\| \\ \|\xi_1^{(k+1)}\| \\ \vdots \\ \|\xi_{m-1}^{(k+1)}\| \\ \|\xi_{m-2}^{(k+1)}\| \end{bmatrix} \leqslant \left[\begin{array}{c|cccccc} \frac{1}{2} & \frac{1}{2} & & & & & \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & & & & \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & & & \\ \vdots & \vdots & \ddots & \ddots & \ddots & & \\ \frac{1}{2^{m-2}} & \frac{1}{2^{m-2}} & \frac{1}{2^{m-3}} & \frac{1}{2^{m-4}} & \cdots & \frac{1}{2} \\ \hline \frac{1}{2^{m-2}} & \frac{1}{2^{m-2}} & \frac{1}{2^{m-3}} & \frac{1}{2^{m-4}} & \cdots & \frac{1}{2} \end{array}\right] \begin{bmatrix} \|\xi_0^{(k)}\| \\ \|\xi_1^{(k)}\| \\ \vdots \\ \|\xi_{m-1}^{(k)}\| \\ \|\xi_{m-2}^{(k)}\| \end{bmatrix}.$$

Denoting the vector by $\boldsymbol{\delta}$ and the matrix by T, we compactly rewrite

$$\boldsymbol{\delta}^{(k+1)} \leqslant T\boldsymbol{\delta}^{(k)}. \tag{4.10}$$

The matrix $T \in \mathbb{R}^{(m-1)\times(m-1)}$ has a Hessenberg structure and several other interesting properties:

- It is a *doubly stochastic matrix* [BP94, p. 48], i.e.,

$$\forall i, j \quad T_{ij} \in \mathbb{R}_+, \qquad \sum_{j=1}^{m-1} T_{ij} = 1 \quad \forall i, \qquad \sum_{i=1}^{m-1} T_{ij} = 1 \quad \forall j.$$

- It is *irreducible* [BP94, p. 29], i.e.,

$$\forall i, j, \quad \exists N \in \mathbb{N} \quad \text{such that} \quad (T^N)_{ij} > 0.$$

  For our matrix, $N = m - 2$ to obtain a nonzero coefficient $T_{1,m-1}$ on the upper right corner. To show this, first we observe that since all the matrix coefficients are positive, by taking matrix powers they stay positive. Moreover, at each multiplication, a diagonal gets filled with strictly positive coefficients. So it takes a power of $N = m - 2$ to make the coefficient $T_{1,m-1}$ strictly positive.

- The *Perron root* (or Perron–Frobenius eigenvalue) of T is $r = 1$.

- It exists a vector $\boldsymbol{v}$ such that $T\boldsymbol{v} = r\boldsymbol{v}$ and whose components are all strictly positive. For our matrix, $\boldsymbol{v} = \frac{1}{\sqrt{m-1}}(1, \dots, 1)^\mathsf{T}$, i.e., a normalized right eigenvector associated with $r = 1$. It also exists a vector $\boldsymbol{w}$ such that $\boldsymbol{w}^\mathsf{T}T = r\boldsymbol{w}^\mathsf{T}$ and whose components are all strictly positive. For our matrix, $\boldsymbol{w} \equiv \boldsymbol{v}$.

Let us now consider the "shadow sequence" with the equality sign associated with (4.10), namely

$$\widetilde{\boldsymbol{\delta}}^{(k+1)} = T\widetilde{\boldsymbol{\delta}}^{(k)}.$$

By recursively applying this equality one obtains

$$\widetilde{\boldsymbol{\delta}}^{(k+1)} = T^{k+1}\widetilde{\boldsymbol{\delta}}^{(0)}.$$

Using the fact that $\widetilde{\boldsymbol{\delta}}^{(k+1)}$ is an upper bound on $\boldsymbol{\delta}^{(k+1)}$, one can write

$$\boldsymbol{\delta}^{(k+1)} \leqslant T^{k+1}\widetilde{\boldsymbol{\delta}}^{(0)}.$$

By the *Perron-Frobenius theorem* [BP94, p. 45], it holds

$$\lim_{k\to\infty} \frac{T^k}{r^k} = \boldsymbol{v}\boldsymbol{w}^\mathsf{T},$$

where $\boldsymbol{v}$ and $\boldsymbol{w}$ are normalized such that $\boldsymbol{w}^\mathsf{T}\boldsymbol{v} = 1$. The matrix $\boldsymbol{v}\boldsymbol{w}^\mathsf{T}$ is the projection onto the eigenspace associated with the Perron root r. In our case, the above limit becomes

$$\lim_{k\to\infty} T^k = \boldsymbol{v}\boldsymbol{v}^\mathsf{T} = \frac{1}{m-1}\,\mathbb{1}_{m-1},$$

where $\mathbb{1}_{m-1}$ is an all-ones square matrix of size $m-1$. This implies that, in the limit for $k \to \infty$,

$$\boldsymbol{\delta}^{(\infty)} \leqslant \boldsymbol{v}\boldsymbol{v}^\mathsf{T}\widetilde{\boldsymbol{\delta}}^{(0)} = \frac{1}{m-1}\,\mathbb{1}_{m-1}\,\widetilde{\boldsymbol{\delta}}^{(0)} = \left(\frac{1}{m-1}\sum_{j=1}^{m-1}\tilde{\delta}_j^{(0)}\right)(1,\dots,1)^\mathsf{T}.$$

This result tells us that, at convergence, each component of $\boldsymbol{\delta}^{(\infty)}$ is bounded by an arithmetic mean of all the components of $\widetilde{\boldsymbol{\delta}}^{(0)}$. Nonetheless, it remains an open problem to find a lower bound in order to prove convergence in this way.

4.4 Broken geodesic shooting method

In this section, we propose an alternative shooting algorithm which exploits the idea of broken geodesics introduced by [Noa98, KN08] and discussed in Section 4.1. The main difference between this method and multiple shooting is that it produces a continuous curve at each iteration.

As before, let us consider m points, corresponding to $m-1$ subintervals. The base point of each subinterval is found by using the geodesic equation with base point from the previous subinterval. Hence, only the $m-1$ tangent vectors $\xi_i,\ i = 0,\dots,m-2$, are unknowns of the problem. We adopt the notation

$$X_0 \equiv X, \quad X_i \equiv \gamma_i(X_{i-1}, \xi_{i-1}, t=1), \quad \dot{X}_i \equiv \dot{\gamma}_i(X_{i-1}, \xi_{i-1}, t=1), \quad i = 1,\dots,m-1,$$

with γ_i being the geodesic that realizes ξ_{i-1}, with base point X_{i-1}. Figure 4.6 provides an illustration of the problem statement and the notation adopted. Here, X_3 depends on ξ_2 and X_2, which depends on ξ_1 and X_1, and so on.

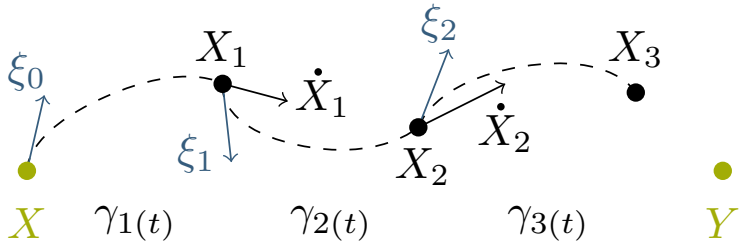


Figure 4.6 – Broken geodesic shooting method.

Our system of nonlinear equations collects the mismatching for the tangent vectors and for the arrival point

$$F(\xi) = \begin{bmatrix} \dot{X}_1 - \xi_1 \\ \dot{X}_2 - \xi_2 \\ \vdots \\ r = X_{m-1} - Y \end{bmatrix} = 0. \tag{4.11}$$

The idea is very similar to multiple shooting, except that here only the continuity of the derivative of the geodesic has to be enforced.

Now, consider a perturbation of the nonlinear system

$$F(\xi + \delta\xi) = 0, \quad \text{with} \quad \delta\xi = [\, \delta\xi_0 \quad \delta\xi_1 \quad \cdots \quad \delta\xi_{m-2} \,]^{\mathsf{T}}.$$

A linearization of this system yields

$$F(\xi) + J_F^{\xi} \cdot \delta\xi = 0, \tag{4.12}$$

with the Jacobian $J_F^{\xi} \in \mathbb{R}^{(m-1)np \times (m-1)np}$ having a lower block-Hessenberg structure:

$$J_F^{\xi} = \begin{bmatrix} J^{\xi_0}_{\dot{X}_1} & -I_{np} & & & \\ J^{X_1}_{\dot{X}_2} J^{\xi_0}_{X_1} & J^{\xi_1}_{\dot{X}_2} & -I_{np} & & \\ \vdots & \ddots & \ddots & \ddots & \\ J^{X_{m-3}}_{\dot{X}_{m-2}} J^{X_{m-4}}_{X_{m-3}} \cdots J^{\xi_0}_{X_1} & \ddots & \ddots & J^{\xi_{m-3}}_{\dot{X}_{m-2}} & -I_{np} \\ J^{X_{m-2}}_{X_{m-1}} J^{X_{m-3}}_{X_{m-2}} \cdots J^{\xi_0}_{X_1} & J^{X_{m-2}}_{X_{m-1}} J^{X_{m-3}}_{X_{m-2}} \cdots J^{\xi_1}_{X_2} & \cdots & J^{X_{m-2}}_{X_{m-1}} J^{\xi_{m-3}}_{X_{m-2}} & J^{\xi_{m-2}}_{X_{m-1}} \end{bmatrix}.$$

The blocks on the main diagonal are calculated as

$$(J_F^{\xi})_{ii} = J^{\xi_{i-1}}_{\dot{X}_i}, \qquad i = 1, \ldots, m-2, \quad \text{and} \quad (J_F^{\xi})_{m-1,m-1} = J^{\xi_{m-2}}_{X_{m-1}},$$

where $J^{\xi_{i-1}}_{\dot{X}_i}$ denotes the Jacobian matrix of ξ_{i-1} with respect to $\dot{X}_i$. The off-diagonal blocks are given by

$$(J_F^{\xi})_{ij} = J^{X_{i-1}}_{\dot{X}_i} \left(\prod_{k=i-1}^{j+1} J^{X_{k-1}}_{X_k} \right) J^{\xi_{j-1}}_{X_j}, \qquad i = 1, \ldots, m-2, \quad 0 < j < i,$$

$$(J_F^{\xi})_{m-1,j} = \left(\prod_{k=m-1}^{j+1} J^{X_{k-1}}_{X_k} \right) J^{\xi_{j-1}}_{X_j}, \qquad j = 1, \ldots, m-2,$$

where the matrix products are ordered products, and

$$(J_F^\xi)_{ij} = -I_{np}, \quad j = i+1, \qquad (J_F^\xi)_{ij} = O_{np}, \quad j > i+1, \;\; i = 1, \dots, m-2.$$

Observe that the last line of J_F^ξ is built differently because it enforces the boundary condition at Y. Indeed, at convergence, X_{m-1} has to be equal to Y. Moreover, we emphasize that we have all the analytic expressions for computing the smaller Jacobians appearing in J_F^ξ, because these are the same as the Jacobians for single and multiple shooting (see Chapter 2 and Appendix C).

To gain insight into the above derivation, let us expand $\dot{X}_i$ as

$$\begin{aligned}
\dot{X}_i &\equiv \dot{\gamma}_i(X_{i-1}, \xi_{i-1}, t=1) \\
&= \dot{\gamma}_i\big(\gamma_{i-1}(X_{i-2}, \xi_{i-2}, t=1), \xi_{i-1}, t=1\big) \\
&= \dot{\gamma}_i\Big(\gamma_{i-1}\big(\gamma_{i-2}(X_{i-3}, \xi_{i-3}, t=1), \xi_{i-2}, t=1\big), \xi_{i-1}, t=1\Big) \\
&= \dot{\gamma}_i\Big(\gamma_{i-1}\Big(\gamma_{i-2}\big(\gamma_{i-3}(\dots(\gamma_1(X_0, \xi_0, t=1), \xi_1, t=1), \dots, \xi_{i-2}, t=1\Big), \xi_{i-1}, t=1\Big).
\end{aligned}$$

For instance, the Jacobian $\frac{\partial \dot{X}_i}{\partial \xi_0}$ is given by the chain rule

$$\frac{\partial \dot{X}_i}{\partial \xi_0} = \frac{\partial \dot{X}_i}{\partial X_{i-1}} \frac{\partial X_{i-1}}{\partial X_{i-2}} \cdots \frac{\partial X_2}{\partial X_1} \frac{\partial X_1}{\partial \xi_0},$$

that we can write in matrix notation as

$$J_{\dot{X}_i}^{\xi_0} = J_{\dot{X}_i}^{X_{i-1}} J_{X_{i-1}}^{X_{i-2}} \cdots J_{X_2}^{X_1} J_{X_1}^{\xi_0}.$$

At every iteration the method solves (4.12) for $\delta\xi$. Since J_F^ξ can be quite big, it makes sense to consider the same condensing strategy as the one adopted for multiple shooting (see Section 2.4.1).

4.4.1 Condensing

The linear system (4.12) can be solved efficiently thanks to the structure of J_F^ξ, which allows any $\delta\xi_i$, $i = 1, \dots, m-2$, to be expressed as a function of $\delta\xi_0$. This condensing strategy can be summarized as follows:

$$\widetilde{w}_i = F_i + \sum_{j=2}^{i} (J_F^\xi)_{ij}\, \widetilde{w}_{j-1}, \qquad \widetilde{M}_i = (J_F^\xi)_{i1} + \sum_{j=2}^{i} (J_F^\xi)_{ij}\, \widetilde{M}_{j-1}, \qquad i = 1, \dots, m-1,$$

$$w = \widetilde{w}_{m-1}, \quad M = \widetilde{M}_{m-1}.$$

Eventually, only one linear system of size $np \times np$ has to be solved to find $\delta\xi_0$:

$$M \cdot \delta\xi_0 = -w,$$

and the remaining $\delta\xi_i$ are obtained as

$$\delta\xi_i = F_i + \sum_{j=1}^{i} (J_F^\xi)_{ij}\, \delta\xi_{j-1}, \qquad i = 1, \dots, m-2.$$

4.4.2 Complexity of the algorithm

Solving system (4.12) via the condensing strategy has an asymptotic complexity $O(m^2n^3p^3)$. Without condensing, using MATLAB `lu`, the complexity is $O(m^3n^3p^3)$. If the original problem on $\mathrm{St}(n,p)$ is reduced to a problem on $\mathrm{St}(2p,p)$ (see Section 2.3.3), then we get with the condensing strategy $O(m^2p^6)$, and without condensing $O(m^3p^6)$. As a consequence, for large values of m, the broken geodesic algorithm with condensing strategy will be more efficient than the MATLAB `lu`. Figure 4.7 provides an illustration of this fact for $\mathrm{St}(40,p)$ with $m=6$.

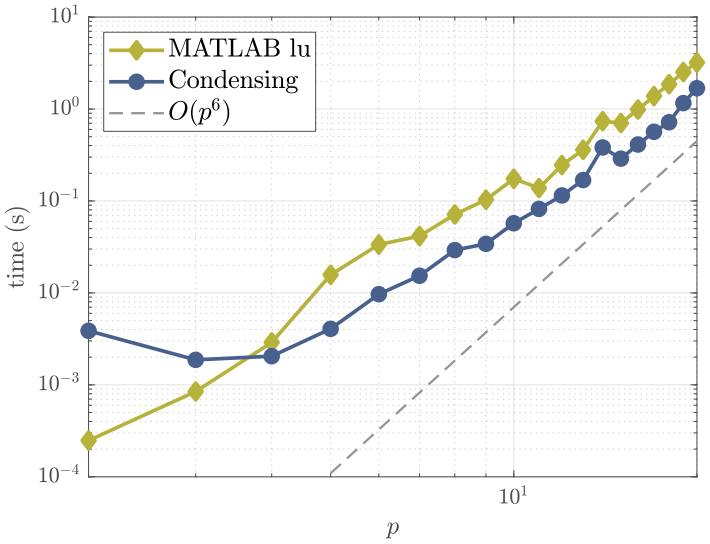


Figure 4.7 – Computational time of the broken geodesic algorithm for $\mathrm{St}(40,p)$ with $m=6$.

By contrast, multiple shooting with condensing has a complexity $O(mp^6)$, i.e., the complexity is linear in m. Hence this complexity analysis shows that multiple shooting is in general more efficient than the broken geodesic shooting method.

4.4.3 Leapfrog revisited

With the notation introduced above, the leapfrog algorithm can be compactly rewritten as follows. Starting from a broken geodesic whose junction points are

$$\left(X_0^{(0)}, X_1^{(0)}, X_2^{(0)}, \ldots, X_{m-1}^{(0)}\right), \quad \text{with} \quad X_0^{(0)} \equiv X, \quad X_{m-1}^{(0)} \equiv Y,$$

the leapfrog algorithm computes, for $k \geqslant 1$,

$$X_0^{(k)} = X_0^{(k-1)}, \qquad \xi_i^{(k)} = \tfrac{1}{2}\operatorname{Log}_{X_i^{(k)}}\left(X_{i+2}^{(k-1)}\right),$$

$$X_{i+1}^{(k)} = \operatorname{Exp}_{X_i^{(k)}}\left(\xi_i^{(k)}\right), \qquad i = 0, \ldots, m-3,$$

until a stopping criterion is satisfied. We recall that leapfrog assumes the subintervals to be small enough so that the Riemannian logarithm can be computed via single shooting.

Leapfrog has the remarkable property of converging to a curve having continuous first derivatives. In other words, at convergence the curve will be globally C^1:

$$\lim_{k\to\infty} \dot{X}_i^{(k)} - \xi_i^{(k)} = 0, \qquad i = 1, \ldots, m-3.$$

Observe that these equations are the same as those of the broken geodesic algorithm, as stated in the nonlinear system (4.11).

4.5 Numerical experiments and applications

In this section, we present some simple numerical experiments about the leapfrog and the multiple shooting algorithms. We report on their convergence behavior and discuss a couple of applications.

From an algorithmic point of view, we propose the following scheme, summarized by Figure 4.8:

- Since we do not know a priori whether Y_0 and Y_1 are very close or very distant, the first attempt to solve the endpoint geodesic problem is always done with single shooting.
- If single shooting works[2], then the problem is solved and we are done. In practice, we perform single shooting and check whether it converges or not. If single shooting does not converge, we start with leapfrog with two subintervals, i.e., with $m = 3$ points, which is the smallest partition possible.
- If leapfrog with two subintervals does not work, i.e., if the single shooting behind leapfrog does not work, we keep increasing the number of subintervals until it works. The reason for this is that the single shooting behind leapfrog has to converge on each subinterval.
- When leapfrog works, we perform a few iterations and then we use the iterate found by leapfrog as an initial guess for multiple shooting.
- The problem is solved with multiple shooting, which converges quadratically to the solution.

4.5.1 Leapfrog and multiple shooting

As a concrete example to demonstrate the leapfrog and the multiple shooting algorithms, let us consider the Stiefel manifold $\mathrm{St}(12, 3)$. We fix one point $X = [I_3 \ O_{9\times 3}]^\mathsf{T}$, while the other point Y is placed at a distance $L^* = 0.95\,\pi$ from X. This choice is made in order to have two points that are far enough from each other; i.e., this problem is such that it cannot be solved by using single shooting alone. Recall also that a lower bound on the injectivity radius of $\mathrm{St}(n, p)$ is given by $0.89\,\pi$ [Ren13, Eq. (5.13)], so it makes sense to consider a distance $L^* > 0.89\,\pi$ in order to test these algorithms. By using our numerical algorithms, we want to recover this distance. As number of points we choose $m = 4$, i.e., the path between X and Y is cut into 3 subintervals.

To monitor the convergence behavior, two quantities have been considered:

- $|L_k - L^*|$, where L_k is the length of the piecewise geodesic at iteration k.
- $\|F(\Sigma_k)\|_2$, where $F(\Sigma_k)$ is the nonlinear function of multiple shooting, as defined in Section 2.4.

[2] In the experiments, we choose 10 as maximum number of single shooting iterations. We consider that single shooting fails when this number is exceeded.

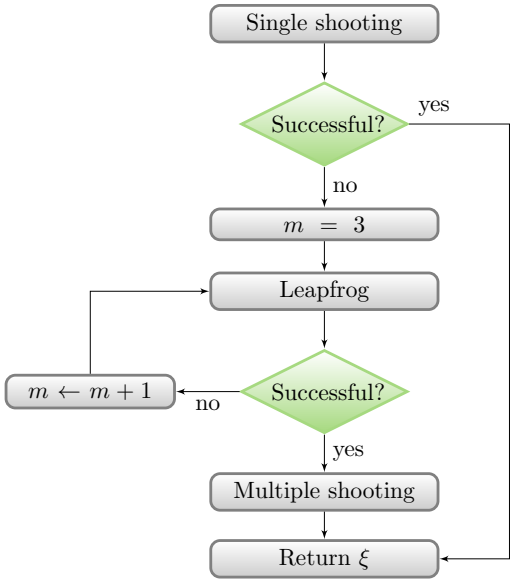


Figure 4.8 – Flowchart of the Stiefel Log algorithm.

Figure 4.9 reports on the convergence behavior of leapfrog. Leapfrog is stopped when $\|F(\Sigma_k)\|_2$ reaches the threshold value of 10^{-3} (this happens at the 28th iteration). We estimate that at this threshold the iterates will fall in the so-called basin of attraction of Newton's method, so that multiple shooting will succeed when started with the iterate generated by leapfrog. The linear convergence behavior of leapfrog is clearly visible.

Figure 4.10 reports on the convergence behavior of multiple shooting. Multiple shooting is started from where leapfrog left the job; one can check this by comparing the last iteration in leapfrog with the 0th iteration of multiple shooting. It is apparent the quadratic convergence behavior and the onset of the plateau at around machine precision $\varepsilon_{\text{mach}} \approx 10^{-16}$.

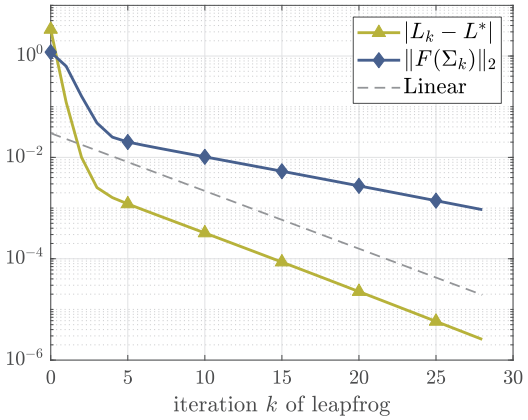


Figure 4.9 – Convergence of leapfrog for $\mathrm{St}(12, 3)$, with $m = 4$.

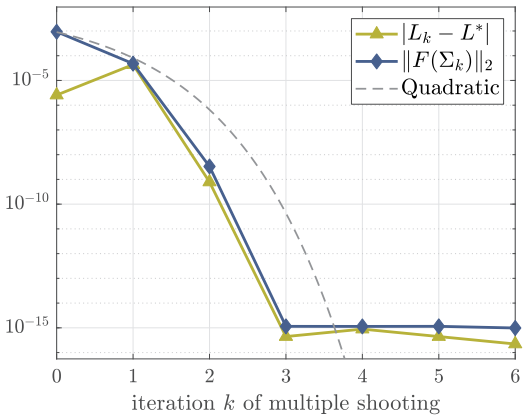


Figure 4.10 – Convergence of multiple shooting for $\mathrm{St}(12, 3)$, with $m = 4$.

4.5.2 Riemannian center of mass on the space of univariate probability density functions

We present an application that uses means on a Riemannian manifold $\mathcal{M}$. Given N points $q_i \in \mathcal{M}$, their *Riemannian center of mass* is defined by the optimization problem

$$\mu = \operatorname*{arg\,min}_{p \in \mathcal{M}} \frac{1}{2N} \sum_{i=1}^{N} d(p, q_i)^2,$$

where $d(p, q_i)$ is the Riemannian distance between two points on $\mathcal{M}$.

On manifolds of positive curvature there are in general many Riemannian centers of mass. The Stiefel manifold has also positive curvature and an upper bound on its sectional curvature is given by $5/4$ [Ren13, p. 95].

The Riemannian center of mass, and hence the Riemannian distance, is used to calculate an average probability density function (PDF). This is a simple problem since we consider the unit n-sphere S^n, which is a special case of Stiefel manifold, but it remains interesting because it allows for a nice visualization of the outcome. Before presenting a concrete example, let us introduce some important notions.

Let $\mathcal{P}$ be the space of univariate PDFs on the unit interval $[0, 1]$

$$\mathcal{P} = \Big\{ g \colon [0,1] \to \mathbb{R}_{\geqslant 0} \colon \int_0^1 g(x)\,\mathrm{d}x = 1 \Big\}.$$

By introducing the *half-density representation* of the elements of $\mathcal{P}$

$$q(t) = \sqrt{g(t)},$$

the set $\mathcal{P}$ can be identified with the space

$$\mathcal{Q} = \big\{ q \colon [0,1] \to \mathbb{R}_{\geqslant 0} \colon \|q\| = 1 \big\}.$$

This identification allows us to attach a spherical structure to $\mathcal{P}$, and the unit n-sphere $S^n = \{x \in \mathbb{R}^{n+1} : \|x\| = 1\}$ can be used to approximate the space of univariate PDFs on the unit interval $[0, 1]$. We refer the reader to [SK16, §7.5.3] for further details.

Given a certain number of PDFs, one might be interested into computing summary statistics of all them, and this can be given by their Riemannian center of mass. As a concrete example, we consider three PDFs, sampled at 100 points. This discretization makes them belong to $\mathrm{St}(100, 1)$, i.e., the unit sphere S^{99}. Figure 4.11 shows the three PDFs on the left panel, and their Riemannian center of mass on the right panel. The resulting Riemannian center of mass is a PDF that summarizes the features (e.g., peak locations, spread around the peaks) of the three original PDFs.

4.5.3 Interpolation on the Stiefel manifold for model order reduction

In this section, we consider an example in the same order of ideas as in [AF11]. Specifically, we look at the interpolation of *linear parametric reduced-order models*. It is beyond the scope of this thesis to discuss reduced-order models (ROMs); for a comprehensive review of model order reduction techniques, we refer the reader to [BGW15].

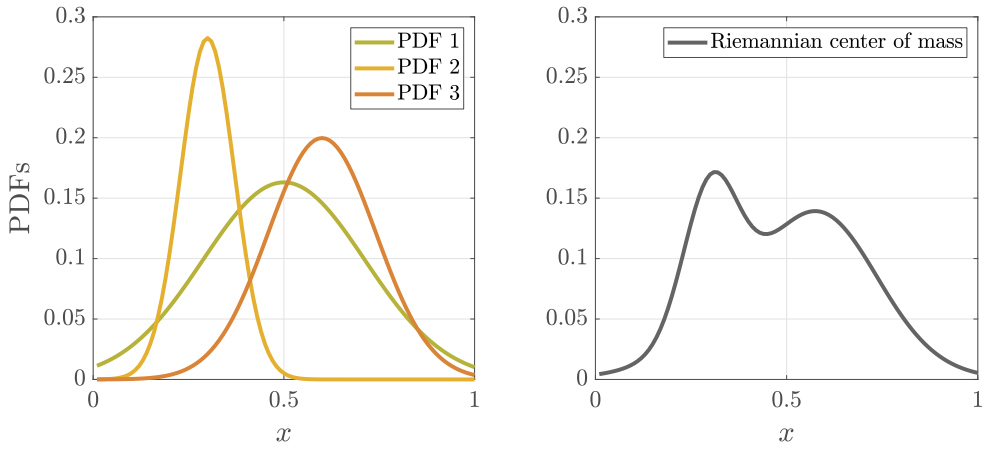


Figure 4.11 – Riemannian center of mass of three PDFs.

Let us consider the dynamical model parametrized with respect to $\mathbf{p} = [p_1, \dots, p_d]^\mathsf{T}$

$$\begin{cases} \dot{\mathbf{x}}(t;\mathbf{p}) = A(\mathbf{p})\,\mathbf{x}(t;\mathbf{p}) + B(\mathbf{p})\,\mathbf{u}(t) \\ \mathbf{y}(t;\mathbf{p}) = C(\mathbf{p})\,\mathbf{x}(t;\mathbf{p}), \end{cases}$$

with $\mathbf{x}(t;\mathbf{p}) \in \mathbb{R}^n$ the vector of state variables, $\mathbf{u}(t) \in \mathbb{R}^m$ the vector of inputs, and $\mathbf{y}(t) \in \mathbb{R}^q$ the vector of outputs. The system matrices are $A(\mathbf{p}) \in \mathbb{R}^{n\times n}$, $B(\mathbf{p}) \in \mathbb{R}^{n\times m}$, and $C(\mathbf{p}) \in \mathbb{R}^{q\times n}$.

The *reduced dynamical system* is

$$\begin{cases} \dot{\mathbf{x}}_r(t;\mathbf{p}) = A_r(\mathbf{p})\,\mathbf{x}_r(t;\mathbf{p}) + B_r(\mathbf{p})\,\mathbf{u}(t) \\ \mathbf{y}_r(t;\mathbf{p}) = C_r(\mathbf{p})\,\mathbf{x}_r(t;\mathbf{p}), \end{cases}$$

with $\mathbf{x}_r = V^\mathsf{T}\mathbf{x}$ the reduced-size vector, and system matrices $A_r = V^\mathsf{T}AV$, $B_r = V^\mathsf{T}B$, $C_r = CV$, where $V \equiv V(\mathbf{p}) \in \mathrm{St}(n,r)$. To obtain the matrix V, one needs to apply a ROM technique. Here, we adopt a proper orthogonal decomposition (POD) with N snapshots [BGW15, §3.3.1]. Let X be the snapshot matrix that collects N snapshots of the solution at different times $t_1, \dots, t_N$:

$$X = [\mathbf{x}(t_1;\mathbf{p}), \dots, \mathbf{x}(t_N;\mathbf{p})].$$

Then the POD basis V is chosen as the r left singular vectors of X that correspond to the r largest singular values. In MATLAB notation:

$$[U, \sim, \sim\,] = \mathrm{svd}(X), \quad \text{then} \quad V = U(:, 1\!:\!r).$$

The process of interpolation on manifolds is explained in [AF11, p. 2180] and [BGW15, §4.2.1]. It can be summarized as follows, with Figure 4.12 as a reference illustration. For each parameter in a set of parameter values $\{\mathbf{p}_1, \dots, \mathbf{p}_K\}$, one uses a model order reduction technique to derive a reduced-order basis $V_i \in \mathrm{St}(n,r)$. This yields a set of local basis matrices $\{V_1, \dots, V_K\}$. One of these matrices (V_3 in the figure) is chosen as reference point to expand a tangent space to $\mathrm{St}(n,r)$. Then, given a new parameter value $\hat{\mathbf{p}}$, a basis $\widehat{V}$ can be obtained by interpolating the local basis matrices on the tangent space. This process remains the same also for general manifolds.

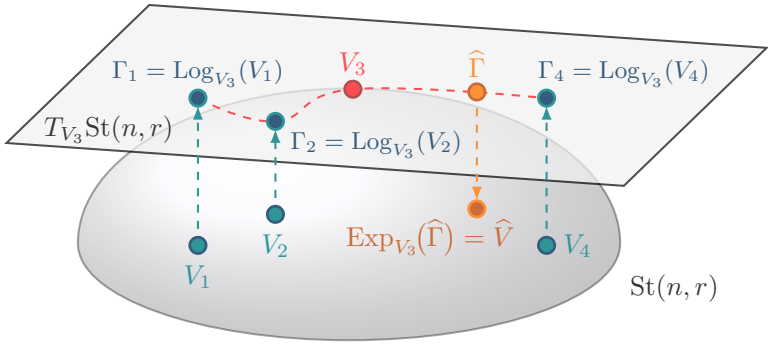


Figure 4.12 – Interpolation on $\mathrm{St}(n,r)$.

As a concrete application, we consider the transient heat equation on a square domain with 4 disjoint discs, which model four cookies lying on a square tray in an oven [Tob12, p. 86]. The problem is discretized with a finite element mesh with piecewise linear basis functions, resulting in a parametrized dynamical system of size $n = 1169$ of the form

$$\dot{\mathbf{x}}(t;\mathbf{p}) = -A(\mathbf{p})\,\mathbf{x}(t;\mathbf{p}) + \mathbf{b},$$

where

$$\mathbf{p} = (p_1,\, p_2,\, p_3,\, p_4) \in [0,1]^4, \qquad A(\mathbf{p}) = \Big(A_0 + \sum_{i=1}^{4} p_i A_i\Big),$$

and the matrices $A_1, \ldots, A_4$ contain the contributions from the corresponding disc. The right-hand side $\mathbf{b}$ is obtained from the discretization of the source term $f \equiv 1$. Figure 4.13 illustrates the discretized problem.

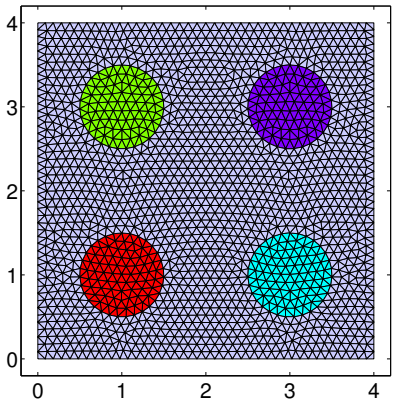


Figure 4.13 – Mesh for 2×2 discs (from [Tob12, Fig. 4.13]).

In our example, $\mathbf{p} = (p_1,\, 0.10,\, 0.15,\, 0.70)$, with $p_1 \in [0.12, 1]$, i.e., the first parameter varies while the others are fixed. As ROM technique, we adopt a POD with 500 snapshots in time, with a reduced-model size $r = 4$.

We monitored the following error quantities:

- The error between $\widehat{V}_{\mathrm{POD}}$, the basis obtained by directly applying a POD, and $\widehat{V}_{\mathrm{interp}}$, the basis obtained by interpolating on $\mathrm{St}(n,r)$ as described above:

$$\text{err-interp} = \|\widehat{V}_{\mathrm{POD}} - \widehat{V}_{\mathrm{interp}}\|_2.$$

- For the new operating point $\hat{\mathbf{p}} = (\hat{p}_1,\ 0.10,\ 0.15,\ 0.70)$, with $\hat{p}_1 = 0.40$, the relative error on the output of the reduced model with respect to the output of the full model (see [BGW15, §2.4]):

$$\text{err-}\mathbf{y} = \frac{\|\mathbf{y}_r(t, \hat{\mathbf{p}}) - \mathbf{y}(t, \hat{\mathbf{p}})\|_{L_2}}{\|\mathbf{y}(t, \hat{\mathbf{p}})\|_{L_2}}.$$

To perform the interpolation on the tangent space, the Matlab function `interp1` for 1D interpolation was used with three different methods: piecewise linear interpolation (linear), piecewise cubic spline interpolation (spline) and shape-preserving piecewise cubic interpolation (pchip).

Figure 4.14 reports on the convergence behavior of err-interp with respect to the number K of local basis matrices. It is clear that err-interp improves as we increase the number of local basis matrices. Moreover, the spline method appears to be the most accurate among the ones considered.

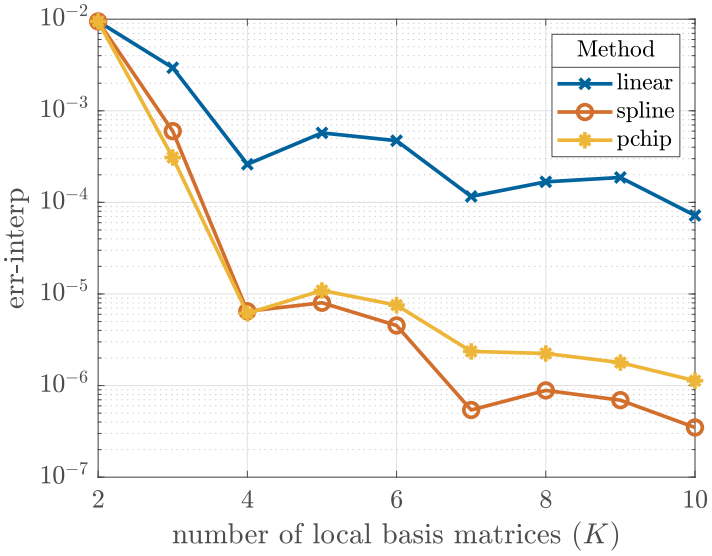


Figure 4.14 – Convergence of err-interp.

In the next example, we monitor the convergence behavior of err-$\mathbf{y}$ with respect to the size r of the reduced model, $r = 1, 2, \ldots, 20$. We choose $\mathbf{p} = (0.12,\ 0.10,\ 0.15,\ 0.70)$ and considered five different PODs, with an increasing number of snapshots, 10, 100, 500, 1 000, 2 500 respectively. We estimate that for applications in various fields of engineering, an err-$\mathbf{y}$ of about 1% is already good enough. From Figure 4.15 one can observe that for reduced models obtained from 500, 1 000, 2 500 snapshot PODs, the 1% error is achieved for a size $r = 4$. When using less snapshots (like 10, 100), one needs $r = 9,\ 10$ to achieve err-$\mathbf{y} = 1\%$.

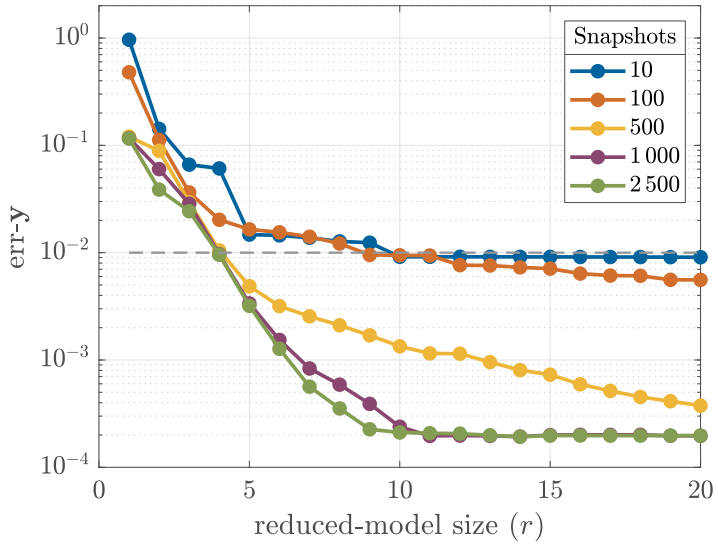


Figure 4.15 – Convergence of err-$\mathbf{y}$.

Chapter 5

Riemannian Hager–Zhang line search

In optimization methods that only use first-order information, the convergence to stationary points is typically linear at best. In contrast to second-order algorithms like Newton's method, this makes it difficult to achieve high accuracy in finite precision arithmetic when using standard line searches, like the weak Wolfe conditions. As we shall see, a more accurate line search was proposed by Hager and Zhang [HZ05, HZ06] in the context of a new nonlinear CG method. In this chapter, we explain in some detail how the Hager–Zhang line search works. Most importantly, we generalize this line-search method to the Riemannian setting. The algorithm obtained is applied to two optimization problems from [AMS08] in order to illustrate the improved accuracy. More problems on the manifold of fixed-rank matrices are presented in Chapter 7.

5.1 Inaccuracy in standard line search

The usual stopping criterion for line search is the weak Wolfe conditions, which we recall here. Let f be a differentiable objective function. Let x_k be the current iterate, $g_k = \nabla f(x_k)$ the gradient, and d_k the search direction. The weak Wolfe conditions for the step size $\alpha_k > 0$ are defined by

$$f(x_k + \alpha_k d_k) - f(x_k) \leqslant \delta \alpha_k \, d_k^\mathsf{T} g_k, \qquad d_k^\mathsf{T} \nabla f(x_k + \alpha_k d_k) \geqslant \sigma \, d_k^\mathsf{T} g_k,$$

with $0 < \delta \leqslant \sigma < 1$. The first inequality is known as *sufficient decrease*, or *Armijo, condition*, while the second represents a *curvature condition*. One can reformulate the weak Wolfe conditions in terms of $\phi(\alpha_k) = f(x_k + \alpha_k d_k)$ as follows:

$$\delta \, \phi'(0) \geqslant \frac{\phi(\alpha_k) - \phi(0)}{\alpha_k}, \qquad \phi'(\alpha_k) \geqslant \sigma \, \phi'(0), \tag{5.1}$$

with $0 < \delta \leqslant \sigma < 1$. Since we are moving along a descent direction for f, we observe that the slope $\phi'(0)$ and the difference $\phi(\alpha_k) - \phi(0)$ are negative. The first inequality is thus asking for the decrease in ϕ at α_k to be larger than $\delta \, \phi'(0)$. The second inequality is asking for the slope of ϕ at α_k to be larger than $\sigma \, \phi'(0)$. Figure 5.1 illustrates the weak Wolfe conditions in terms of ϕ.

In finite precision arithmetic, the weak Wolfe conditions can be difficult to satisfy very accurately due to roundoff error when x is very close to the local minimum of f. This is easy

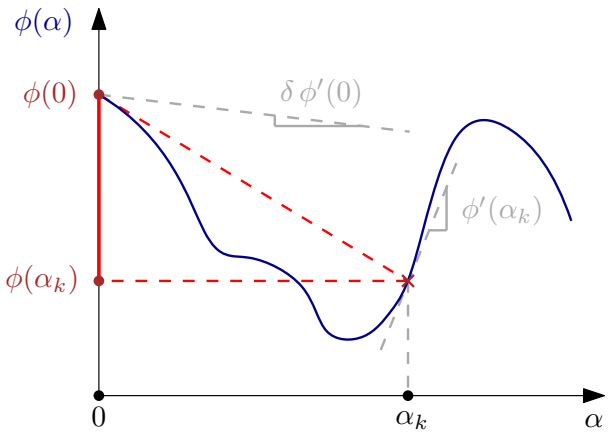


Figure 5.1 – Weak Wolfe conditions in terms of ϕ.

to see for a smooth objective function with a strict local minimum, like the one depicted in Figure 5.2. The function f is locally quadratic and its minimum can only be determined by the line-search method within $\sqrt{\varepsilon_{\text{mach}}}$, with $\varepsilon_{\text{mach}}$ the machine epsilon.

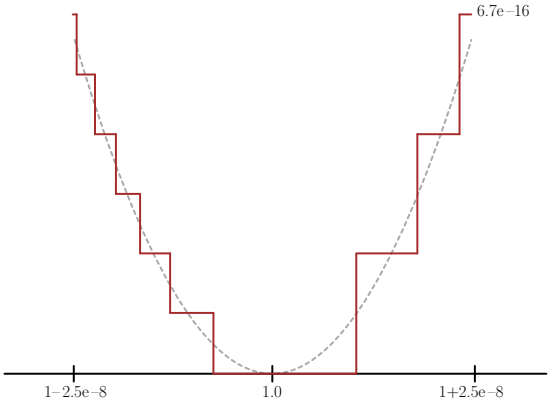


Figure 5.2 – Exact and numerical graphs of $f(x) = 1 - 2x + x^2$ near $x = 1$ (adapted from [HZ05, §4]). The dotted line is the exact f, while the solid line is its representation in double precision with $\varepsilon_{\text{mach}} \approx 10^{-16}$.

Remark 5.1. Newton's method with unitary step size does not have this problem of numerical accuracy, since no line search is involved. Even if a line search is used far from the optimum, eventually no line search will be used close to it, where we know that Newton's method converges quadratically, and hence this problem will not arise. In a first-order method with a fixed step this problem will not show up either. By this observation, we want to stress that the problem of numerical accuracy is due to the line-search method adopted, and in general it is not intrinsic to first-order methods.

5.2 Approximate Wolfe conditions

Hager and Zhang proposed in [HZ05, §4] to relax the weak Wolfe conditions (5.1) and formulate the *approximate* Wolfe conditions based on the derivative of the objective function. Using these conditions as stopping criterion for line search permits to reach an accuracy

within the machine precision $\varepsilon_{\text{mach}}$. Roughly speaking, the idea is that finding the zero of the derivative of a quadratic (which is just a straight line) is better conditioned numerically than finding the minimizer of the quadratic itself.

The main observation of Hager and Zhang is that, in a neighborhood of a local minimum, the first condition in (5.1) is difficult to satisfy since $\phi(\alpha) \approx \phi(0)$. This makes the subtraction $\phi(\alpha) - \phi(0)$ relatively inaccurate [HZ06, §3]. To prevent this loss of accuracy, they introduce the approximate Wolfe conditions [HZ05, Eq. (4.1)]

$$(2\delta - 1)\,\phi'(0) \geqslant \phi'(\alpha_k) \geqslant \sigma\,\phi'(0), \qquad 0 < \delta < 0.5, \quad \delta \leqslant \sigma < 1. \tag{5.2}$$

Here, the first inequality is an approximation of the first condition in (5.1), but the second inequality coincides with the second condition in (5.1).[1] The approximation comes from replacing ϕ by its quadratic interpolant q that satisfies the conditions $q(0) = \phi(0)$, $q'(0) = \phi'(0)$, and $q'(\alpha_k) = \phi'(\alpha_k)$. In other words, the interpolant q has the form

$$q(\alpha) = a\,\alpha^2 + b\,\alpha + c, \qquad q'(\alpha) = 2a\,\alpha + b,$$

with the conditions

$$q(0) = c = \phi(0), \qquad q'(0) = b = \phi'(0),$$

$$q'(\alpha_k) = 2a\,\alpha_k + \phi'(0) = \phi'(\alpha_k) \quad \Longrightarrow \quad a = \frac{\phi'(\alpha_k) - \phi'(0)}{2\alpha_k}.$$

So the quadratic model is

$$q(\alpha) = \left(\frac{\phi'(\alpha_k) - \phi'(0)}{2\alpha_k}\right)\alpha^2 + \phi'(0)\,\alpha + \phi(0).$$

For $\alpha = \alpha_k$ we have

$$\begin{aligned} q(\alpha_k) &= \left(\frac{\phi'(\alpha_k) - \phi'(0)}{2}\right)\alpha_k + \phi'(0)\,\alpha_k + q(0) \\ &= \left(\frac{\phi'(\alpha_k) + \phi'(0)}{2}\right)\alpha_k + q(0), \end{aligned}$$

hence

$$\frac{q(\alpha_k) - q(0)}{\alpha_k} = \frac{\phi'(\alpha_k) + \phi'(0)}{2}. \tag{5.3}$$

The finite difference quotient on the right-hand side of the first Wolfe condition in (5.1) can be approximated by (5.3)

$$\frac{\phi(\alpha_k) - \phi(0)}{\alpha_k} \approx \frac{q(\alpha_k) - q(0)}{\alpha_k} = \frac{\phi'(\alpha_k) + \phi'(0)}{2}. \tag{5.4}$$

We emphasize that this expression is only valid for this specific interpolant q. With this approximation, the subtraction $q(\alpha_k) - q(0)$ can be computed more accurately as $\phi'(\alpha_k) + \phi'(0)$, thereby circumventing the possible cancellation due to roundoff errors in the original difference $\phi(\alpha_k) - \phi(0)$. Substituting (5.4) into (5.1) yields the first approximate Wolfe condition in (5.2), namely,

$$(2\delta - 1)\,\phi'(0) \geqslant \phi'(\alpha_k).$$

[1] For this reason, it would therefore be more appropriate to talk about the *approximate Armijo* condition rather than the *approximate Wolfe* conditions, but we stick with the latter name as in [HZ05, HZ06].

5.3 The Hager–Zhang bracketing

The algorithm for generating and updating the bracketing interval is based on the secant and bisection methods, as described in [HZ06, §3]. This is similar to standard line-search methods, except that the method tries to enforce the approximate Wolfe conditions (5.2). In this section, we will only give an outline of the algorithm; we refer the reader to [HZ06] for a more detailed description. We first describe the termination criteria, and then the line-search procedure itself.

The Hager–Zhang line search is terminated whenever a step size α_k is generated such that one of the following termination criteria is satisfied:

- T1: The original Wolfe conditions (5.1) are satisfied (with a standard Armijo procedure);
- T2: The approximate Wolfe conditions (5.2) are satisfied (approximation with an interpolant as explained in the previous section) and the additional condition [HZ06, Eq. (27)]

$$\phi(\alpha_k) \leqslant \phi(0) + \varepsilon_k, \tag{5.5}$$

  where $\varepsilon_k \geqslant 0$ is an estimate for the error in the value of f at iteration k.

  As in [HZ05, p. 182], for the numerical experiments we took

$$\varepsilon_k = \varepsilon\,|f(x_k)|, \tag{5.6}$$

  where ε is a small fixed parameter, $\varepsilon = 10^{-6}$. Condition (5.5) allows for a small growth in the value of the objective f. Roughly speaking, this criterion permits to terminate the line search when the value of f at the accepted step (i.e., $\phi(\alpha_k) = f(x_k + \alpha_k d_k)$) is not much larger than the value of f at the previous iterate (i.e., $\phi(0) = f(x_k)$).

The method from [HZ06, §3] generates a nested sequence of bracketing intervals that are guaranteed to contain an acceptable step length α. A typical interval $[a, b]$ in this sequence satisfies condition (5.5) and the *opposite slope condition*, i.e.,

$$\phi(a) \leqslant \phi(0) + \varepsilon_k, \qquad \phi'(a) < 0, \qquad \phi'(b) \geqslant 0. \tag{5.7}$$

This is nothing else than the opposite sign condition of the bisection method translated to the derivative, meaning that the derivative changes sign in the bracketing interval (and, thus, must have a root). Figure 5.3 illustrates the opposite slope condition (5.7) for the function $\phi(\alpha)$, for a bracketing interval $[a, b]$.

Given a bracketing interval $[a, b]$ satisfying (5.7) and a point c generated by either a secant step or a bisection step, the *update* of the bracketing interval is performed according to the procedure described in [HZ06, p. 123]. After completing this procedure, we have a new interval

$$[\bar{a}, \bar{b}] \subset [a, b],$$

whose endpoints satisfy (5.7).

The input c for the update routine is generated by a *secant step*. Basically, this is a step taken towards a local minimum of $\phi(\alpha)$, given by the recurrence relation of the secant method applied to $\phi'(\alpha)$, i.e.,

$$c = \frac{a\,\phi'(b) - b\,\phi'(a)}{\phi'(b) - \phi'(a)}.$$

This special secant step is used to achieve rapid convergence. However, if the secant step is converging too slowly, then a bisection step is used instead. This is checked via the condition

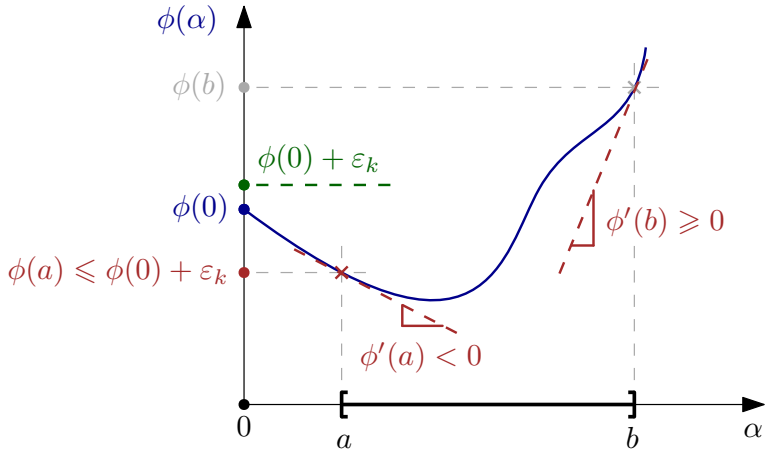


Figure 5.3 – The opposite slope condition together with the condition $\phi(a) \leqslant \phi(0) + \varepsilon_k$.

$(\bar{b} - \bar{a}) > \gamma(b_j - a_j)$. The choice $\gamma = 0.66$ that we made in the numerical experiments ensures that the length of the interval $[a, b]$ decreases by a factor of $2/3$ in each iteration of the line-search algorithm.

A pseudocode for the Hager–Zhang line search algorithm is outlined in Algorithm 2.

Algorithm 2: Hager–Zhang line search

Generate a starting guess c;
Generate an initial interval $[a, b]$ satisfying (5.7); set $j = 0$;
while *T1 or T2 is satisfied* **do**
  Use a secant method to update the bracketing interval;
  If the secant method is converging too slowly, use a bisection step $c = (\bar{a} + \bar{b})/2$ and update the bracketing interval;
  Increment j: $j = j + 1$;
end

5.3.1 Numerical examples

To illustrate the convergence behavior of the Hager–Zhang line search, we consider two numerical examples in which we compare the results of steepest descent using weak Wolfe conditions and the Hager–Zhang line search.

As in [HZ05, p. 186], we did not consider objective functions whose optimal cost is zero. The reason is that if the optimal cost is zero, then the estimate (5.6) for the error in the function value gets very poor as the iterates approach the minimizer (i.e., as $f(x_k)$ tends to zero). Since we wish to obtain a decrease in the objective value, a small $\varepsilon = 10^{-6}$ is chosen for the error tolerance in (5.6). In all the numerical experiments with the Hager–Zhang line search of this chapter we used the following parameter values:

$$\delta = 0.1, \qquad \sigma = 0.9, \qquad \varepsilon = 10^{-6}, \qquad \gamma = 0.66.$$

5.3.1.1 Quadratic cost function

In this first example, we consider the quadratic cost function $f : \mathbb{R}^{n\times n} \to \mathbb{R}$, defined by

$$f(X) = \tfrac{1}{2}\operatorname{trace}(X^{\mathsf{T}}AX) - \operatorname{trace}(X^{\mathsf{T}}B).$$

Here, we choose $n = 100$, the condition number of the symmetric positive definite matrix A as $\kappa(A) = 10$, and $B = AX^*$, where X^* is the exact solution to the problem $AX = B$. The starting point of the optimization is a random initial guess $X^{(0)}$.

From Figure 5.4, we see that the gradient norm stagnates at about 10^{-8} for the weak Wolfe conditions (WW). In contrast, the approximate Wolfe conditions used by the Hager–Zhang line search (HZ) allow to reach an accuracy on the order of $\varepsilon_{\text{mach}}$ ($\approx 10^{-16}$ in double precision) in both the objective value and the gradient norm. The error $\|X_k - X_*\|/\|X_*\|$ also shows that a small gradient is needed, and a termination criterion based only on the objective value is not sufficient.

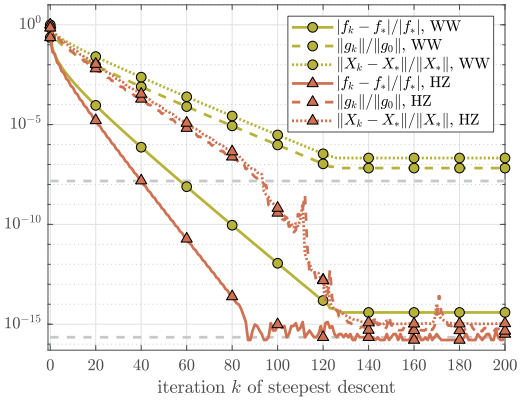


Figure 5.4 – Convergence behavior of steepest descent with WW or HZ line search, for a quadratic cost function.

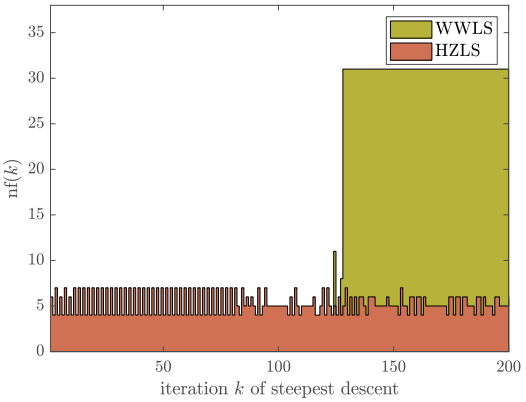


Figure 5.5 – Number of function evaluations per steepest descent iteration, for a quadratic cost function.

Figure 5.5 reports on the number of function evaluations per steepest descent iteration. For this example, when using the approximate Wolfe conditions the number of function evaluations is about 55% less than the one attained by using the weak Wolfe conditions. This is most likely because the standard line search wastes a lot of effort in bracketing the function $\phi(\alpha)$ that becomes noisy due to roundoff error when α is close to a stationary point.

5.3.1.2 Rosenbrock function

The second example deals with the minimization of the Rosenbrock function, a standard test function in optimization. A two-dimensional Rosenbrock function is given by the following expression

$$z(x, y) = 100\,(y - x^2)^2 + (1 - x)^2 + 1,$$

that has a unique minimum value of 1 which is attained at the point $[1, 1]$. Figure 5.6 illustrates this Rosenbrock function from two different angles[2], with the global minimum represented by a red disk. As one can see, the global minimum lies inside a long, narrow, deeply curved flat valley. To find the valley is trivial; however, it is very difficult to converge to the global minimum.

For the numerical experiments, we minimize the above two-dimensional Rosenbrock function starting from an initial guess very close to the exact solution, using 200 steepest descent iterations. From Figure 5.7, we see that the gradient norm stagnates very early for the weak Wolfe conditions, at about 10^{-2}. In contrast, the approximate Wolfe conditions

[2] A log scaling $f(x, y) = \log(1 + z)$ has been applied in order to obtain a decent visualization.

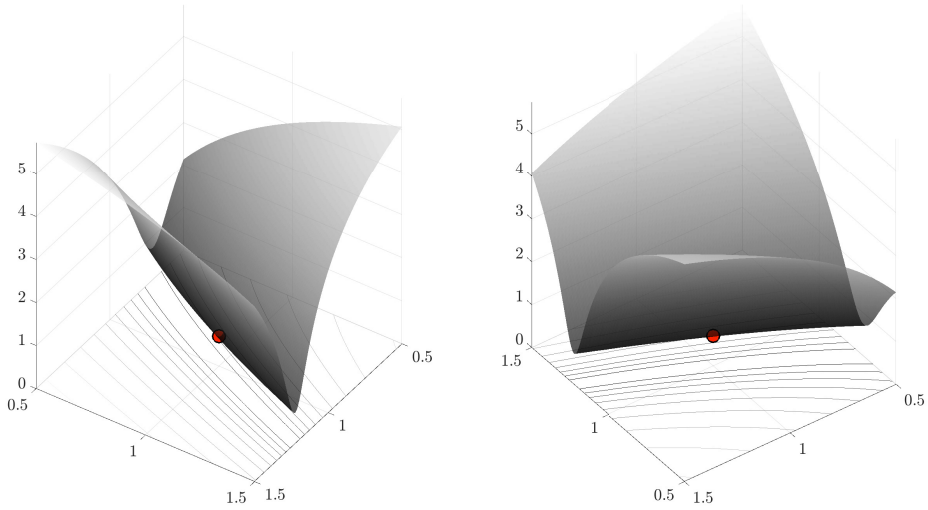


Figure 5.6 – Log-scaled 2D Rosenbrock function.

allow us to achieve a better accuracy in the gradient norm, even if this stagnates at around 10^{-9}, which remains far from the $\varepsilon_{\text{mach}}$ ($\approx 10^{-16}$ in double precision). This might be due to the nature of the valley where the global minimum is lying, and that we are not using any second-order information about the valley. However, the error in the solution $\|X_k - X_*\|/\|X_*\|$ stagnates at around 10^{-6} with the weak Wolfe conditions, while it reaches 10^{-13} with the Hager–Zhang line search.

Figure 5.8 shows that the number of function evaluations per steepest descent iteration is in general higher when using the weak Wolfe conditions. This makes the Hager–Zhang line search not only more accurate, but also cheaper than the standard line search.

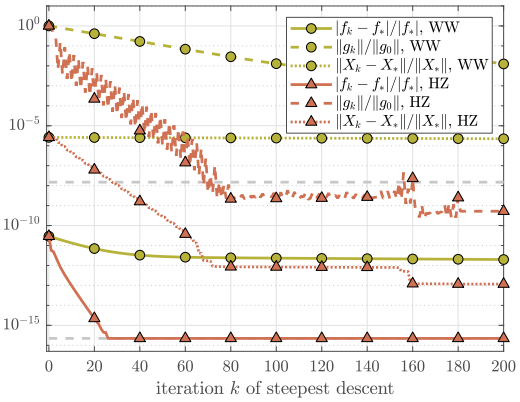


Figure 5.7 – Convergence behavior of steepest descent with WW or HZ line search, for the 2D Rosenbrock function.

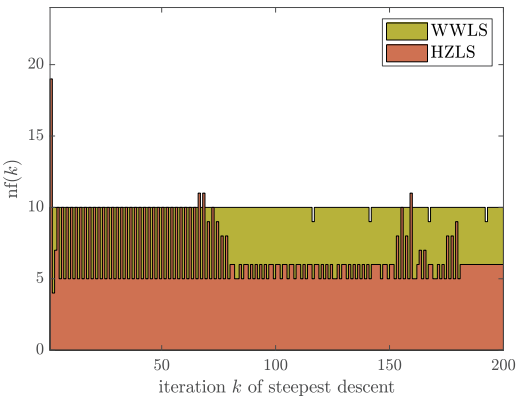


Figure 5.8 – Number of function evaluations per steepest descent iteration, for the 2D Rosenbrock function.

5.4 Riemannian Hager–Zhang line search

The Hager–Zhang line search explained in the previous sections can be readily extended to Riemannian manifolds by applying it to the retracted objective function $\phi(t) = f(R_x(t \cdot \eta))$

along the search direction $\eta \in T_x\mathcal{M}$ with $t \geqslant 0$ the step length (see Section 1.2.2). We call this generalization *Riemannian Hager–Zhang line search*. The function ϕ considered in the line-search procedure is given by a composition of the objective f and the retraction chosen. Since ϕ' is needed in order to apply the approximate Wolfe conditions, we have to compute the derivative of the retraction $\mathrm{d}R_x(t \cdot \eta)/\mathrm{d}t$, which is cumbersome for general retractions R_x. Fortunately, in Riemannian optimization we can choose a retraction that better suits our needs. Moreover, some programming languages (e.g., C++, Python) allow for easy derivation with automatic differentiation.

Here we state the problem more formally. Let $\mathcal{M}$ be a Riemannian manifold, $x \in \mathcal{M}$ and R_x the retraction at x. Let $f\colon \mathcal{M} \to \mathbb{R}$ be an objective function. Observe that, for a fixed tangent vector η, one has $R_x\colon \mathbb{R} \to \mathcal{M}$, defined by $t \mapsto R_x(t\eta)$. Hence let $\phi\colon \mathbb{R} \to \mathbb{R}$, defined by $t \mapsto \phi(t) = f(R_x(t\eta))$. By chain rule, one can get the derivative of $\phi(t)$ as

$$\phi'(t) = \big\langle \nabla f(R_x(t\eta)), \tfrac{\mathrm{d}}{\mathrm{d}t} R_x(t\eta) \big\rangle = \operatorname{trace}\Big(\nabla f(R_x(t\eta))^{\mathsf{T}} \tfrac{\mathrm{d}}{\mathrm{d}t} R_x(t\eta) \Big),$$

where ∇f is the Euclidean gradient of f.

The retraction R_x and its derivative $\frac{\mathrm{d}}{\mathrm{d}t} R_x(t\eta)$ depend on the choice of the manifold. In Section 7.3.8, we will see how to apply the Riemannian Hager–Zhang line search on the manifold of fixed-rank matrices in the context of a new multilevel Riemannian optimization algorithm. In the rest of this chapter, we already present two examples to illustrate that the Riemannian Hager–Zhang line search does provide more accurate results than the ones obtained when using standard line-search techniques on manifolds. The two examples discussed below are the Rayleigh quotient on the sphere [AMS08, p. 73], and the Brockett cost function on the Stiefel manifold [AMS08, p. 80].

5.4.1 Numerical examples

For both the examples presented in this section, we first detail how to compute the derivatives of the retractions by explicit formulas, and then we switch to the associated numerical experiments.

5.4.1.1 Derivative of the retraction on the unit sphere

Consider the retraction on the unit sphere S^{n-1} (see Section 1.2.1.1, and [AMS08, p. 57])

$$R_x(t\eta) = \frac{x + t\eta}{\|x + t\eta\|_2},$$

defined for all $\eta \in T_x S^{n-1}$ and $t \in \mathbb{R}$. Perturbing t with a small $\varepsilon > 0$ gives

$$\begin{aligned} R_x\big((t+\varepsilon)\eta\big) &= \frac{x + (t+\varepsilon)\eta}{\|x + (t+\varepsilon)\eta\|_2} \\ &= \frac{1}{\|x + t\eta\|_2} \cdot \left(x + t\eta + \varepsilon\eta - (x + t\eta) \frac{(x + t\eta)^{\mathsf{T}}\eta}{\|x + t\eta\|_2^2} \varepsilon \right). \end{aligned}$$

From the first-order terms in ε, we can identify the derivative

$$\tfrac{\mathrm{d}}{\mathrm{d}t} R_x(t\eta) = \frac{1}{\|x + t\eta\|_2} \cdot \left(I_n - \frac{(x + t\eta)(x + t\eta)^{\mathsf{T}}}{\|x + t\eta\|_2^2} \right) \eta.$$

The asymptotic complexity of computing R_x is $O(n)$. The derivative $\frac{\mathrm{d}}{\mathrm{d}t}R_x$ can be computed at an additional cost of $O(n)$ via the formula:

$$\frac{\mathrm{d}}{\mathrm{d}t}R_x(t\eta) = \frac{\eta}{\|x + t\eta\|_2} - R_x(t\eta) \cdot \frac{(x + t\eta)^\mathsf{T}\eta}{\|x + t\eta\|_2^2}.$$

The $O(n)$ complexity is due to the calculation of the scalar product $(x + t\eta)^\mathsf{T}\eta$, whose cost is $\approx 2n$.

5.4.1.2 Rayleigh quotient on the sphere

We consider the problem of computing a dominant eigenvector of a symmetric matrix $A^{n\times n}$. Let λ_1 be the largest eigenvalue of A, and v_1 the associated normalized eigenvector. The largest eigenvalue λ_1 is a maximum value of the function $f\colon S^{n-1} \to \mathbb{R}$, defined by

$$x \mapsto x^\mathsf{T}Ax,$$

which is known as the *Rayleigh quotient on the sphere*. The global maximizers of the Rayleigh quotient are $\pm v_1$. We refer the reader to [AMS08, p. 74] for a complete characterization of the critical points of the Rayleigh quotient.

In our numerical experiment, we consider $n = 1000$, and 600 steepest descent iterations. We compare the results for steepest descent using the standard Manopt line search with Armijo condition[3] versus the Hager–Zhang line search.

Figure 5.9 reports on the convergence behavior of the gradient norm. As in the Euclidean examples that we discussed in the previous sections, it is apparent that the Riemannian steepest descent with Hager–Zhang line search leads to a more accurate result, allowing to reach double precision with $\varepsilon_{\text{mach}} \approx 10^{-16}$.

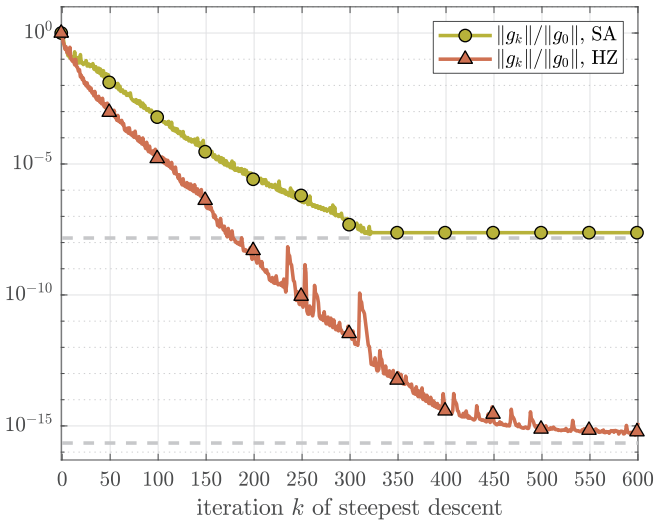


Figure 5.9 – Convergence behavior of steepest descent with standard Armijo (SA) or Hager–Zhang (HZ) line search when applied to the Rayleigh quotient on the sphere. The horizontal dashed lines indicate $\sqrt{\varepsilon_{\text{mach}}}$ and $\varepsilon_{\text{mach}}$.

[3] See [BMAS14].

5.4.1.3 Derivative of the QR retraction on the Stiefel manifold

The next numerical example deals with a cost function defined on the Stiefel manifold, so we need to compute the derivative of a retraction on $\mathrm{St}(n,p)$. We choose the retraction based on the QR factorization (see Section 1.2.1.3, and [AMS08, Eq. (4.8)])

$$R_X(t\xi) = \mathrm{qf}(X + t\xi), \tag{5.8}$$

where $\mathrm{qf}(A)$ denotes the Q factor of the decomposition of $A \in \mathbb{R}_*^{n\times p}$ as $A = QR$, where Q belongs to $\mathrm{St}(n,p)$ and R is an upper triangular p-by-p matrix with strictly positive diagonal elements. This choice makes this decomposition unique.

An explicit formula for the directional derivative $\mathrm{D}\,\mathrm{qf}(X)[\xi]$ of (5.8) is given in the calculations of [Cha12, Eq. (17)]. Here, we rewrite it in the form

$$\tfrac{\mathrm{d}}{\mathrm{d}t}R_X(t\xi) = \xi R^{-1} - Q\,\mathrm{up}\Big[R^{-\mathsf{T}}\xi^{\mathsf{T}}Q + Q^{\mathsf{T}}\xi R^{-1}\Big],$$

where up is defined for any matrix $M \in \mathbb{R}^{n\times n}$ as [Cha12, Eq. (1)]

$$\mathrm{up}(M) = \mathrm{triu}(M) - \mathrm{diag}(\tfrac{1}{2}\,\mathrm{diag}(M)) = \begin{bmatrix} \frac{1}{2}m_{11} & m_{12} & \cdots & m_{1n} \\ & \frac{1}{2}m_{22} & \ddots & m_{2n} \\ & & \ddots & \vdots \\ & & & \frac{1}{2}m_{nn} \end{bmatrix}.$$

Here, triu denotes the operator that extracts the upper triangular part of a matrix, while diag extracts the main diagonal from a matrix if its argument is a matrix, or builds a diagonal matrix from a vector if its argument is a vector.

The computational complexity of the thin QR factorization involved in the retraction R_X is $4p^2(n-p/3)$ [Hig08, p. 337]. Once the retraction has been computed, the derivative $\frac{\mathrm{d}}{\mathrm{d}t}R_X$ can be calculated at an additional cost of $O(np^2)$.

The derivative of the QR retraction is used in the following section to apply the Riemannian Hager–Zhang line search on the Stiefel manifold.

5.4.1.4 Brockett cost function on the Stiefel manifold

We consider a cost function defined as a weighted sum $\sum_i \mu_i x_{(i)}^{\mathsf{T}} A x_{(i)}$ of Rayleigh quotients on the sphere under an orthogonality constraint, $x_{(i)}^{\mathsf{T}} x_{(j)} = \delta_{ij}$. This function can be written in matrix form as

$$f\colon \mathrm{St}(n,p) \to \mathbb{R}\colon X \mapsto \mathrm{trace}(X^{\mathsf{T}}AXN),$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric and $N = \mathrm{diag}(\mu_1, \ldots, \mu_p)$, with $0 < \mu_1 < \ldots < \mu_p$. This function is known in the literature as the *Brockett cost function* [Bro93]. Its Euclidean gradient is given by $\nabla f = 2AXN$. We refer the reader to [AMS08, p. 81] for a characterization of the critical points of the Brockett cost function.

In our numerical experiment, we consider $n = 10$, $p = 3$, and 700 steepest descent iterations. As in the previous example, we compare the results for steepest descent using the standard Manopt line search with Armijo condition versus the Riemannian Hager–Zhang line search. Figure 5.10 reports on the convergence behavior of the gradient norm. As in the case of the Rayleigh quotient on the sphere, the steepest descent with Hager–Zhang line search is clearly more accurate, allowing to reach double precision $\varepsilon_{\mathrm{mach}} \approx 10^{-16}$.

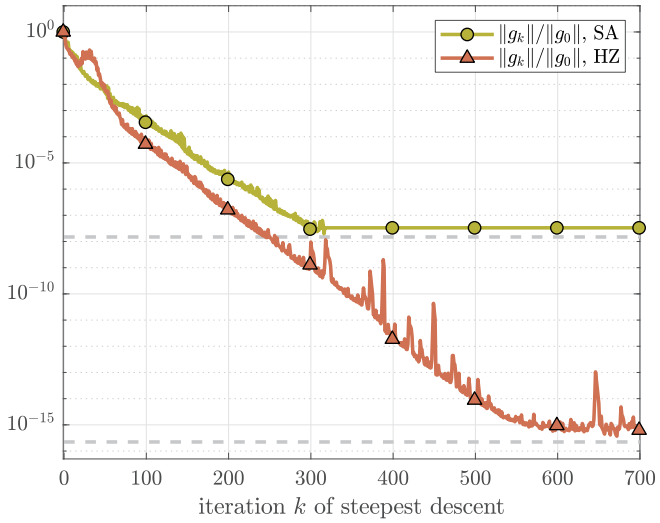


Figure 5.10 – Convergence behavior of steepest descent with standard Armijo (SA) or Hager–Zhang (HZ) line search when applied to the Brockett cost function on the Stiefel manifold.

5.5 Observations and open problems

An open problem for the Hager–Zhang line search is to prove convergence of the method that employs it. In [HZ05, p. 190], the authors proved global convergence of their conjugate gradient method only under the standard Wolfe conditions.

One observation is that the quadratic interpolant introduced in Section 5.2 is actually an instance of a more general interpolation problem known as *Hermite–Birkhoff interpolation* [Fin08]. This is a kind of interpolation problem in which one prescribes the function values and/or derivative values at the given interpolation points. In our setting, we are concerned with the Hermite–Birkhoff interpolant where the given information contains function values and/or only first derivative values at given interpolation points. Let us recall that the conditions for the quadratic interpolant q introduced in Section 5.2 are $q(0) = \phi(0)$, $q'(0) = \phi'(0)$, and $q'(\alpha_k) = \phi'(\alpha_k)$, and let us define the error $e(\alpha)$ as

$$e(\alpha) = \phi(\alpha) - q(\alpha).$$

A bound for the error term is given by [Fin08, Eq. (23)]. By applying the theory in [Fin08] to our case, we obtain the error bound

$$|e(\alpha)| \leqslant \alpha^2 \left(|\alpha - \alpha_k| + \frac{\alpha_k}{2} \right) \frac{\left|\phi^{(3)}(\bar{c})\right|}{6},$$

where $\bar{c}$ is in the interval spanned by all the interpolation points, i.e., $\bar{c} \in [0, \alpha_k]$.

This means that when replacing the quotient

$$\frac{\phi(\alpha_k) - \phi(0)}{\alpha_k} \quad \text{with} \quad \frac{q(\alpha_k) - q(0)}{\alpha_k}$$

in Equation (5.1), we are actually committing an error $e(\alpha_k)/\alpha_k$, that can be bounded by

$$|e(\alpha_k)| \leqslant \tfrac{1}{12}\, \alpha_k^3 \left|\phi^{(3)}(\bar{c})\right|.$$

The issue here is that we do not know a priori the expression ϕ, nor the expression of $\bar{c}$ (which depends on the interpolation knots).

As we observed in Section 5.3, the condition (5.5) allows for a small increase in the objective function. This suggests that the Hager–Zhang bracketing might be viewed as a nonmonotone line-search procedure [GLL86]. In the Riemannian framework, [GSAS20] recently proposed a new Riemannian gradient descent with a nonmonotone line search.

Chapter 6

Multigrid methods

Multigrid methods are a class of methods for discretizing and solving PDEs and, in particular, they are among the most efficient numerical schemes for the solution of elliptic PDEs. The underlying ideas of multigrid methods lend themselves to several generalizations in which grids are not necessarily used, like multilevel, multiscale and multiresolution methods. The algebraic multigrid method is another generalization which extends the fundamental multigrid ideas to matrix problems in a purely algebraic manner.

In this chapter, we will introduce standard multigrid and describe the classical multigrid components. We will also present some concrete numerical examples along the way. The aim is to provide an easy introduction to multigrid methods before Chapter 7, where we extend the multigrid ideas to optimization on Riemannian manifolds. Most of this chapter is based on [TOS00, BHM00, Hac03].

6.1 Some notation

The *continuous* boundary value problem (BVP) is

$$\begin{cases} L^{\Omega}u(\boldsymbol{x}) = f^{\Omega}(\boldsymbol{x}) & \boldsymbol{x} \in \Omega, \\ L^{\Gamma}u(\boldsymbol{x}) = f^{\Gamma}(\boldsymbol{x}) & \boldsymbol{x} \in \Gamma = \partial\Omega, \end{cases} \tag{6.1}$$

where L^{Ω} is an elliptic operator and L^{Γ} is a boundary operator. In this chapter, u always denotes the exact solution of the continuous problem, and u_h the exact solution of the discrete problem. Moreover, we always consider two-dimensional BVPs, which is sufficient for the applications in Chapter 7. For the discretized quantities, we use the term of *discrete differential operator*, and of *grid functions* and *grid operators* instead of vectors and matrices, respectively. The *discrete* boundary value problem is

$$\begin{cases} L_h^{\Omega}u_h(x,y) = f_h^{\Omega}(x,y) & (x,y) \in \Omega_h, \\ L_h^{\Gamma}u_h(x,y) = f_h^{\Gamma}(x,y) & (x,y) \in \Gamma_h = \partial\Omega_h, \end{cases} \tag{6.2}$$

where h is the *discretization parameter*, and

$$u_h(x,y) = u_h(x_i, y_j) = u_h(ih_x, jh_y).$$

For the sake of notation, often we simply write $u_{i,j}$. These coefficients are collected in a matrix U.

Remark 6.1. The discrete elliptic operator L_h^Ω and the discrete boundary operator L_h^Γ are *grid* operators, i.e., they are mappings between *spaces of grid functions*.

We usually write the discrete boundary value problem in the shortened form

$$L_h u_h = f_h \quad \text{in } \Omega_h,$$

where u_h and f_h are grid functions on Ω_h, and L_h is a discrete linear operator

$$L_h \colon \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_h),$$

$\mathcal{G}(\Omega_h)$ being a finite dimensional vector space of grid functions on Ω_h.

6.1.1 Inner products and norms

We define the following inner product for grid functions [TOS00, §1.3.3]:

$$\langle u_h, v_h \rangle_2 = \frac{1}{\#\Omega_h} \sum_{\boldsymbol{x} \in \Omega_h} u_h(\boldsymbol{x}) \, \overline{v_h(\boldsymbol{x})},$$

where $\#\Omega_h$ denotes the number of grid points in Ω_h. The scaling factor $(\#\Omega_h)^{-1}$ allows us to compare grid functions living on different grids, and also the corresponding *continuous* functions on Ω. The induced norm is

$$\|u_h\|_2 = \sqrt{\frac{1}{\#\Omega_h} \sum_{\boldsymbol{x} \in \Omega_h} u_h^2(\boldsymbol{x})} = \sqrt{\frac{1}{\#\Omega_h}} \, \|u_h\|_{\mathrm{F}}, \tag{6.3}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the usual Frobenius norm of a matrix. We point out that (6.3) is just the discrete analogue of the L^2-norm of a two-dimensional continuous function, i.e.,

$$\|u\|_2 = \sqrt{\iint_\Omega u^2(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}}.$$

For the discrete operators L_h on $\mathcal{G}(\Omega_h)$, the *operator norm* is defined as the spectral norm

$$\|B_h\|_S = \sqrt{\rho(B_h B_h^{\mathsf{T}})},$$

where B_h is any linear operator $L_h \colon \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_h)$, and ρ is the spectral radius.

6.1.2 Stencil notation

The so-called stencil notation is used to represent discrete operators L_h. We first define it for an infinite grid. A general stencil $[s_{\kappa_1\kappa_2}]_h$ defines an operator on the set of grid functions w_h by

$$[s_{\kappa_1\kappa_2}]_h w_h(x, y) = \sum_{\kappa_1, \kappa_2} s_{\kappa_1\kappa_2} w_h(x + \kappa_1 h_x, y + \kappa_2 h_y).$$

Since we are usually interested in discrete operators defined only on finite grids Ω_h, in order to identify L_h^Ω with its stencil representation we need to restrict the stencil $[s_{\kappa_1\kappa_2}]_h$ to Ω_h. This implies that only a *finite* number of coefficients $[s_{\kappa_1\kappa_2}]_h$ are nonzero. In practical applications, the five-point stencil or the compact nine-point stencil are commonly used; they are, respectively,

$$\begin{bmatrix} & s_{0,1} & \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ & s_{0,-1} & \end{bmatrix}_h, \qquad \begin{bmatrix} s_{-1,1} & s_{0,1} & s_{1,1} \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ s_{-1,-1} & s_{0,-1} & s_{1,-1} \end{bmatrix}_h.$$

Close to the boundary points, the stencils may need to be modified to include an appropriate treatment of boundary conditions.

In this multigrid presentation, in order to fix our ideas, we will often refer to the following model problem.

6.1.3 Poisson's equation

Poisson's equation is a classical model for a discrete elliptic boundary value problem. We consider the two-dimensional discrete Poisson equation with Dirichlet boundary conditions

$$\begin{cases} -\Delta_h u_h(x,y) = f_h^{\Omega}(x,y) & (x,y) \in \Omega_h, \\ u_h(x,y) = f_h^{\Gamma}(x,y) & (x,y) \in \Gamma_h. \end{cases} \tag{6.4}$$

Here, $\Omega = [0,1]^2 \subset \mathbb{R}^2$, with $h = 1/n$, $n \in \mathbb{N}$, and $L_h = -\Delta_h$ is an approximation of the partial differential operator L, defined by $Lu = -\Delta u = -u_{xx} - u_{yy}$, on the square grid G_h.

6.2 Principles and properties

6.2.1 Fundamental principles

In this section we introduce the two fundamental principles of a multigrid method using problem (6.4). Let us consider a grid function $u_h^{m+1}(x_i, y_j)$ which is updated from its neighboring points using a lexicographical Gauss–Seidel method (m is the iteration index):

$$\begin{aligned} u_h^{m+1}(x_i,y_j) = \frac{1}{4}\Big[& h^2 f_h(x_i,y_j) + u_h^{m+1}(x_i - h, y_j) + u_h^{m}(x_i + h, y_j) \\ & + u_h^{m+1}(x_i, y_j - h) + u_h^{m}(x_i, y_j + h)\Big]. \end{aligned} \tag{6.5}$$

Figure 6.1 illustrates the stencil for the lexicographical Gauss–Seidel method. The approximation at (x_i, y_j) (black dot in the middle) is updated from its four neighboring points. Among these, the two points in the lower-left corner (represented by the red dots) have already been updated; the other two points (represented by the white dots) still have to be updated.

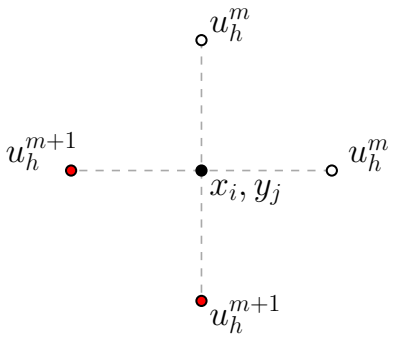
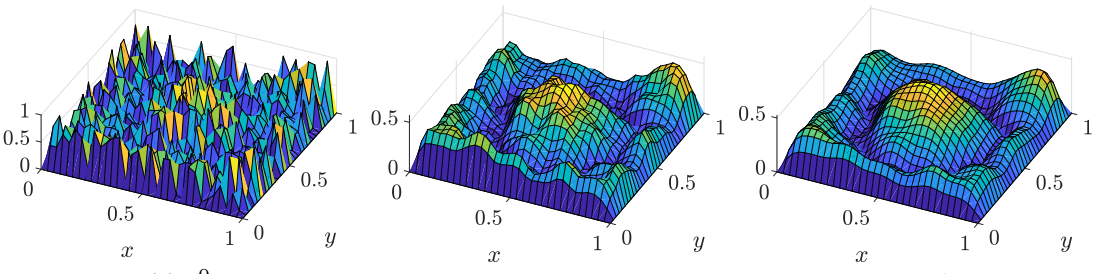


Figure 6.1 – Stencil of the lexicographical Gauss–Seidel method.

Let e_h^m be the error of the approximation at iteration m with respect to the exact discrete solution, defined by

$$e_h^m(x_i, y_j) = u_h(x_i, y_j) - u_h^m(x_i, y_j).$$

If one applies the Gauss–Seidel method (6.5) to Poisson's equation (6.4), one can observe that, after a few steps, the error of the approximation becomes *smooth*. It does not necessarily become small, but it does become smooth. This is illustrated in Figure 6.2, where we have reported the error for a random initial guess, the error after 5 iterations, and the error after 10 iterations. The smoothness of the error after 10 iterations, even when starting with a random initial guess, is striking.



(a) e_h^0 (b) e_h^5 (c) e_h^{10}

Figure 6.2 – Error of lexicographical Gauss–Seidel applied to problem (6.4): (a) random initial guess; (b) after 5 iterations; (c) after 10 iterations.

Hence, the iteration formula (6.5) can be interpreted as an error averaging process. We now introduce the two basic ideas, or principles, underlying every multigrid (or even multilevel) scheme.

Two basic principles of multigrid:

1. **Smoothing principle**. Many classical iterative methods (e.g., Gauss–Seidel) when applied to discrete elliptic problems show a strong smoothing effect on the error of any approximation.

2. **Coarse-grid correction principle**. A smooth error term can be well represented on a coarse grid. Indeed, a grid function that is sufficiently smooth on a given grid can be transferred on a coarser grid without any significant loss of information.

To analyze this behavior, we look at the Fourier expansion of the error $e_h = e_h^m$. At a given iteration m, this can be written as (we omit the superscript m for readability)

$$e_h(x,y) = \sum_{k,\ell=1}^{n-1} \alpha_{k,\ell}\, \varphi_h^{k,\ell}(x,y), \tag{6.6}$$

where $\varphi_h^{k,\ell}(x,y) = \sin(k\pi x)\sin(\ell\pi y)$ are the *eigenfunctions of the discrete Laplacian* Δ_h. The fact that e_h becomes smooth after some iterations means that the *high-frequency components* in (6.6), i.e., the terms $\alpha_{k,\ell}\,\varphi_h^{k,\ell}(x,y)$ with k or ℓ large, become small after a few iterations, whereas the low-frequency components hardly change.

Let us consider a grid Ω_h with discretization parameter $h = 1/n$ and a coarser grid Ω_H with mesh size $H > h$. For example, we can choose $H = 2h$, which corresponds to what is called *standard coarsening*. One can observe that, for $(x,y) \in \Omega_{2h}$, it holds

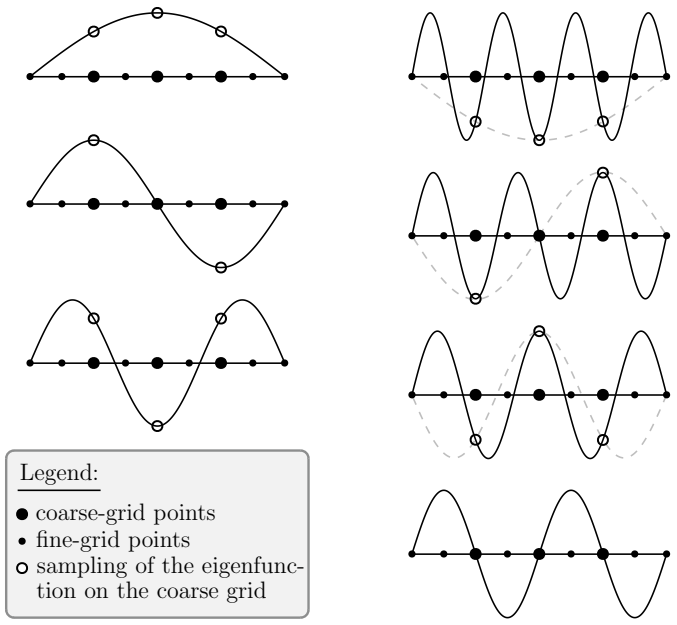
$$\varphi^{k,\ell}(x,y) = -\varphi^{n-k,\ell}(x,y) = -\varphi^{k,n-\ell}(x,y) = \varphi^{n-k,n-\ell}(x,y). \tag{6.7}$$

In other words, on the coarse grid Ω_{2h} these four eigenfunctions coincide. To verify this, take for instance $-\varphi^{n-k,\ell}(x,y)$, and observe that

$$\begin{aligned}-\varphi^{n-k,\ell}(x,y) &= -\sin((n-k)\pi x)\sin(\ell\pi y)\\ &= -\sin\left(2i\pi - k\pi\frac{2i}{n}\right)\sin\left(\ell\pi\frac{2j}{n}\right)\\ &= \sin\left(k\pi\frac{2i}{n}\right)\sin\left(\ell\pi\frac{2j}{n}\right) = \varphi^{k,\ell}(x,y),\end{aligned}$$

where in the second line we used the fact that for $(x,y) \in \Omega_{2h}$ one has $x = iH = 2i/n$ and $y = jH = 2j/n$, while in the third line we used the trigonometric formula $\sin(\alpha - \beta) = \sin\alpha\cos\beta - \cos\alpha\sin\beta$ to simplify the first factor. Similar calculations can be carried out to show the other equalities in (6.7). Moreover, note that for k or ℓ equal to $n/2$ the above eigenfunctions vanish on Ω_{2h}, because $\sin(i\pi)$ and $\sin(j\pi)$ vanish for all $i,\ j$. So in the context of model problem (6.4), it is reasonable to separate low and high frequencies as follows.

Definition 6.2. For $k,\ \ell \in \{1,\dots,n-1\}$, we say that $\varphi^{k,\ell}$ is an eigenfunction (or a *component*) of *low frequency* if $\max(k,\ell) < n/2$, or of *high frequency* if $n/2 \leqslant \max(k,\ell) < n$.



(a) low-frequency components (b) high-frequency components

Figure 6.3 – Low- and high-frequency components for a 1D example ($N = 4, n = 8$). Adapted from [TOS00, p. 18].

Remark 6.3. On the coarse grid Ω_{2h}, only the low frequencies are visible, because all high frequencies coincide with a low frequency. This can be seen from (6.7) where, if $\max(k,\ell)$ is low (i.e., smaller than $n/2$), then $\varphi^{k,\ell}$ is a low frequency and $\varphi^{n-k,n-\ell}$ is a high frequency. The phenomenon of high frequencies coinciding with low ones is called *aliasing of frequencies*. Figure 6.3 illustrates some low- and high-frequency components and the aliasing phenomenon. It is clear from panel (b) that the undersampling on the coarser grid makes the high-frequency components (solid black lines) indistinguishable from the low-frequency components (dashed gray lines).

Remark 6.4. The terms "high" and "low" frequency are related to both the fine grid Ω_h and the coarse grid Ω_H considered. They are *not* absolute notions.

6.2.2 Multigrid features and properties

The combination of the multigrid iteration together with an appropriate smoother leads to a highly efficient Poisson solver. The most desirable property of a multigrid method is the *h-independent convergence*. It can be observed numerically that the convergence of a multigrid Poisson solver is essentially independent of the size of the finest grid in the multigrid cycle. We show this property through a numerical experiment in Section 6.3.5.1.

As confirmed by mathematical theory, multigrid methods work well for elliptic PDEs with a sufficient degree of regularity and formulated on nice domains. However, in practical applications, for example for PDE systems with non-elliptic features and nonlinear terms, such a theory is usually not so widely available.

The *typical components* of a multigrid method are the smoothing procedure, the coarsening strategy, the coarse-grid operators, the intergrid transfer operators and the cycle type.

Concerning the grids, a hierarchy of coarse grids is needed for multigrid. Assume n is a power of 2, $n = 2^p$, $p \in \mathbb{N}^*$, then the discretization parameter is $h = 2^{-p}$. Then we can form the grid sequence

$$\Omega_h,\ \Omega_{2h},\ \Omega_{4h},\ \ldots,\ \Omega_{h_0},$$

where Ω_{h_0} is the coarsest grid. From the multigrid point of view, unstructured grids are a complication. Usually finite differences and finite volumes are used with Cartesian grids, while finite element methods are used with unstructured grids.

Multigrid can be used as an iterative linear solver for a discrete elliptic BVP. It can also be used as a solver for the differential problem itself, i.e., the error is computed with respect to the differential problem. This version is called full multigrid (FMG). The FMG can be optimal: the number of operations is $O(N)$, where N is the number of unknowns in the problem considered. From a practical point of view, efficiency means that the proportionality constants in this asymptotic term $O(N)$ are moderately small.

Multigrid methods have a wide range of applications, as they are not restricted to a certain discretization approach, but can be used in connection with any type of grid-based discretization, and also finite element meshes. Adaptive versions of multigrid are possible, in which finer and finer grids are only constructed in those parts of the domain where the current discretization error is significantly large. For one-dimensional problems, multigrid usually degenerates to well-known optimal solvers, so it does not really make much sense to discuss it in the 1D case, unless for analysis purposes.

6.3 Going into more detail of multigrid

As we mentioned above, multigrid is based on two main principles: *error smoothing* and *coarse-grid correction*. In this section, we discuss these two aspects in more detail.

6.3.1 Error smoothing

Classical iterative solvers like Jacobi or Gauss–Seidel exhibit smoothing properties that depend on the choice of a relaxation parameter value. For the Gauss–Seidel method, they also depend on the ordering of the grid points. Iterative methods of Jacobi or Gauss–Seidel type are also called *relaxation methods* (or smoothing methods, or smoothers) when they are used for the purpose of error smoothing.

There exist many classical iterative solvers for the solution of a linear system $Au = f$. The general iteration of an iterative solver for this equation can be recast as

$$u^{m+1} = Mu^m + s, \qquad m = 0, 1, \ldots,$$

where M is called *iteration matrix*. So the original equation $Au = f$ is equivalent to the *fixed point equation* $u = Mu + s$. There are several ways of specifying M, e.g., as an approximate solution of the defect equation, via splitting, or preconditioning.

The *asymptotic convergence speed* is characterized by the spectral radius of M, i.e.,

$$\rho(M) = \max_i \{|\lambda_i| \colon \lambda_i \text{ eigenvalue of } M\}.$$

In other words, the spectral radius is the asymptotic convergence factor of the iteration, i.e., for $m \to \infty$,

$$\lim_{m\to\infty} \frac{\|u - u^{m+1}\|}{\|u - u^m\|} \leqslant \rho(M).$$

6.3.1.1 Jacobi type iteration

If we apply the ω-Jacobi method to our model problem (6.4), we get the iteration

$$\begin{aligned} u_h^{m+1} &= u_h^m - \frac{\omega h^2}{4}\left(L_h u_h^m - f_h\right) = u_h^m - \frac{\omega h^2}{4} L_h u_h^m + \frac{h^2}{4} f_h \\ &= \left(I_h - \frac{\omega h^2}{4} L_h\right) u_h^m + \frac{h^2}{4} f_h = S_h(\omega)\, u_h^m + \frac{\omega h^2}{4} f_h, \end{aligned}$$

where we defined the iteration matrix $S_h(\omega) = I_h - \frac{\omega h^2}{4} L_h$. To study the convergence properties of the method, we need the eigenfunctions of S_h, which are given by

$$\varphi_h^{k,\ell}(x) = \sin(k\pi x)\sin(\ell\pi y),$$

with $(x, y) \in \Omega_h$, and $k,\ \ell = 1, \ldots, n-1$. It is well known that the asymptotic convergence speed of ω-Jacobi is $\rho(S_h) = 1 - O(\omega h^2)$, which is unsatisfactory [Saa03]. Nonetheless, ω-Jacobi becomes more valuable if we look at it as a smoother, and not as an iterative solver. Indeed, if ω is appropriately chosen, then the highly oscillatory eigenfunctions are reduced much more quickly.

6.3.1.2 Smoothing properties of Jacobi relaxation

In this section, we recall some results for the Jacobi relaxation. Let w_h and $\overline{w}_h$ denote the approximation before and after one relaxation step, respectively. The errors e_h and $\bar{e}_h$ before and after one relaxation step are defined by

$$e_h = u_h - w_h \qquad \text{and} \qquad \bar{e}_h = u_h - \overline{w}_h.$$

Since the eigenfunctions of the operator form a basis for the space of grid functions, we can expand these errors into discrete eigenfunction series

$$e_h = \sum_{k,\ell=1}^{n-1} \alpha_{k,\ell}\, \varphi_h^{k,\ell}, \qquad \bar{e}_h = \sum_{k,\ell=1}^{n-1} \chi_h^{k,\ell}\, \alpha_{k,\ell}\, \varphi_h^{k,\ell}.$$

For the analysis, we need to look at the factors $\chi_h^{k,\ell}$ appearing in the expansion of $\bar{e}_h$. Moreover, we need to distinguish between low and high frequencies. We emphasize that this distinction depends on the coarser grid Ω_{2h} used. The smoothing factor associated to the iteration matrix $S_h(\omega)$ is given by the worst factor among all $\chi_h^{k,\ell}$, i.e., the greatest in absolute value among those associated to high frequencies. This is formalized by the following definition.

Definition 6.5 (Smoothing factor of $S_h(\omega)$)**.** The *smoothing factor* is the *worst* factor by which high-frequency error components are reduced per relaxation step, defined as

$$\mu(h;\omega) = \max_{k,\ell}\{|\chi_h^{k,\ell}(\omega)|\colon \underbrace{n/2 \leqslant \max(k,\ell) \leqslant n-1}_{\text{high frequencies}}\},$$

and its supremum μ^* over h

$$\mu^*(\omega) = \sup_{h\in\mathcal{H}} \mu(h;\omega),$$

where $\mathcal{H}$ denotes a set of reasonable mesh sizes.

The smaller the smoothing factor, the better are the properties of a given relaxation procedure. It turns out that ω-Jacobi has no smoothing properties (i.e., $\mu(h;\omega) \geqslant 1$) for $\omega \leqslant 0$ or $\omega > 1$. Conversely, for $0 < \omega < 1$ the smoothing factor is strictly smaller than 1, independently of h. The optimal choice of ω is given by the value $4/5$, which attains the minimum

$$\inf_{\omega\in[0,1]} \mu^*(\omega) = \mu^*(4/5).$$

This means that one step of ω-Jacobi with $\omega = 4/5$ reduces all high-frequency error components by at least a factor of $3/5$ (independently of the mesh size h).

Remark 6.6. Gauss–Seidel with over-relaxation parameter ω^* has asymptotic convergence speed $\rho(\omega^*\text{-GS}) = 1 - O(h)$, instead of $\rho(\text{GS}) = 1 - O(h^2)$. For model problem (6.4), we have the factor $\mu(\text{GS-LEX}) = 0.50$ for lexicographical Gauss–Seidel, corresponding to the choice of relaxation parameter $\omega = 1$.

6.3.2 Transfer operators

We describe the intergrid transfer operators on the interval $[0,1]$ referring to Figure 6.4 for illustration. Let the finer level be $\ell_\mathrm{f} = 3$, so that we have the discretization parameter $h = 2^{-3}$, then the number of interior grid points on the finer level is $m = 2^3 - 1 = 7$. For the coarser level, $\ell_\mathrm{c} = 2$, one has the discretization parameter $H = 2^{-2} = 2h$ and $M = 2^2 - 1 = 3$ interior grid points. On the finer level, we consider the grid function $u = (u_1, u_2, \ldots, u_7)^\mathsf{T} \in \mathbb{R}^7$, which we transfer on the coarser level to obtain the grid function $v = (v_1, v_2, v_3)^\mathsf{T} \in \mathbb{R}^3$. For the restriction operator, one option is to adopt the *full-weighting* (FW) restriction (panel (a) of Figure 6.4). In our example, this is an operator $R\colon \mathbb{R}^7 \to \mathbb{R}^3$, defined by

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & \\ & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & \\ & & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_7 \end{bmatrix}.$$

Another choice for the restriction is the *injection* operator (panel (b) of Figure 6.4)

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & & & & \\ & & 0 & 1 & 0 & & \\ & & & & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_7 \end{bmatrix}.$$

For the prolongation, the most common choice is to use *linear interpolation*. In our example, the prolongation operator $P\colon \mathbb{R}^3 \to \mathbb{R}^7$ is defined by

$$P = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} & & & & \\ & & \frac{1}{2} & 1 & \frac{1}{2} & & \\ & & & & \frac{1}{2} & 1 & \frac{1}{2} \end{bmatrix}^{\mathsf{T}}.$$

Observe that we have the relationship $P = 2R^{\mathsf{T}}$. Prolongation by linear interpolation is illustrated in panel (c) of Figure 6.4.

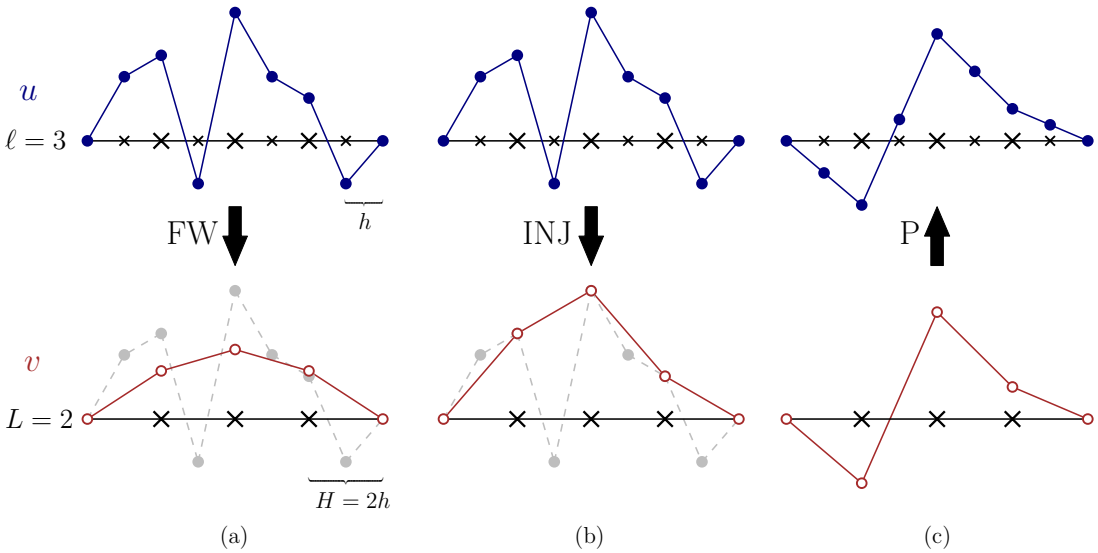


Figure 6.4 – Some intergrid transfer operators for the example described in Section 6.3.2: (a) full-weighting restriction; (b) injection; (c) prolongation by linear interpolation.

We now have all the necessary multigrid components to describe the basic two-grid cycle.

6.3.3 Two-grid cycle

The two-grid cycle is the natural basis for any multigrid algorithm, and lends itself to be easily generalized. We describe it for the solution of a discrete linear elliptic BVP

$$L_h u_h = f_h,$$

on a grid Ω_h. Let the *defect*, or *residual*, be defined as

$$d_h^m = f_h - L_h u_h^m,$$

where u_h is the exact discrete solution, and u_h^m is an approximation of the solution u_h at iteration m. In a (linear) multigrid method, one usually solves the *defect equation*

$$L_h e_h^m = d_h^m$$

for the error e_h^m. The defect d_h^m is transferred to the coarse grid Ω_H by using the *restriction* operator I_h^H

$$d_H^m = I_h^H d_h^m.$$

Then one has to solve the *coarse-grid correction equation*

$$L_H \hat{e}_H^m = d_H^m$$

for the correction $\hat{e}_H^m$, with $L_H \colon \mathcal{G}(\Omega_H) \to \mathcal{G}(\Omega_H)$. The correction is then transferred to the fine grid Ω_h by using the interpolation (or *prolongation*) operator I_H^h:

$$\hat{e}_h^m = I_H^h \hat{e}_H^m,$$

and used to correct the approximation u_h.

The iteration operator associated to the two-grid cycle is given by

$$M_h \colon \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_h), \qquad M_h \coloneqq I_h - C_h L_h,$$

where $C_h = I_H^h L_H^{-1} I_h^H$. So the approximate solution at iteration m is given by:

$$u_h^m = (I_h - M_h^m) L_h^{-1} f_h.$$

The pseudocode for the two-grid cycle is presented in the following box.

for $i = 1, 2, \ldots,$ **do**

(1) **Pre-smoothing**: $\bar{u}_h = \mathrm{SMOOTH}^{\nu_1}(u_h^{(i)}, L_h, f_h)$

(2) i. **Compute** residual: $d_h = f_h - L_h \bar{u}_h$

ii. **Restrict** the residual: $d_H = I_h^H d_h$

iii. **Solve** $L_H e_H = d_H$ for e_H

iv. **Prolong** the coarse-grid correction: $e_h = I_H^h e_H$

v. **Apply** the correction: $\hat{u}_h = \bar{u}_h + e_h$

(3) **Post-smoothing**: $u_h^{(i+1)} = \mathrm{SMOOTH}^{\nu_2}(\hat{u}_h, L_h, f_h)$

end for

6.3.4 Multigrid cycle

As mentioned earlier, the two-grid cycle can be easily generalized to more complex cycles. By recursively applying the same idea to coarser and coarser grids, one can obtain the multigrid cycle. Let us consider the sequence of coarser and coarser grids Ω_{h_ℓ}, i.e.,

$$\Omega_{h_{\ell_\mathrm{f}}},\ \Omega_{h_{\ell_\mathrm{f}-1}},\ \ldots,\ \Omega_{h_{\ell_\mathrm{c}}},$$

where ℓ_{f} denotes the finest level, ℓ_{c} the coarsest level, and $\ell_{\mathrm{c}} \leqslant \ell \leqslant \ell_{\mathrm{f}}$. Moreover, let the restriction and prolongation operators be

$$I_{\ell}^{\ell-1} \colon \mathcal{G}(\Omega_{\ell}) \to \mathcal{G}(\Omega_{\ell-1}), \qquad I_{\ell-1}^{\ell} \colon \mathcal{G}(\Omega_{\ell-1}) \to \mathcal{G}(\Omega_{\ell}).$$

The multigrid algorithm, or $(\ell_{\mathrm{f}}+1)$-grid cycle, is described in the box below [TOS00, p. 47].

$u_{\ell}^{m+1} = \mathrm{MGCYC}(\ell, \gamma, u_{\ell}^{m}, L_{\ell}, f_{\ell}, \nu_1, \nu_2)$

(1) **Pre-smoothing**: $\bar{u}_{\ell}^{m} = \mathrm{SMOOTH}^{\nu_1}(u_{\ell}^{m}, L_{\ell}, f_{\ell})$

(2) i. **Compute defect**: $\bar{d}_{\ell}^{m} = f_{\ell} - L_{\ell}\bar{u}_{\ell}^{m}$

ii. **Restrict the defect**: $\bar{d}_{\ell-1}^{m} = I_{\ell}^{\ell-1}\bar{d}_{\ell}^{m}$

iii. **Solve** $L_{\ell-1}\hat{e}_{\ell-1}^{m} = \bar{d}_{\ell-1}^{m}$ for $\hat{e}_{\ell-1}^{m}$ by:

- If $\ell = \ell_{\mathrm{c}}$, use a direct or fast iterative solver;
- If $\ell > \ell_{\mathrm{c}}$, by performing γ ($\geqslant 1$) ℓ-grid cycles, using the zero grid function as a first approximation:

$$\hat{e}_{\ell-1}^{m} = \mathrm{MGCYC}^{\gamma}(\ell-1, \gamma, 0, L_{\ell-1}, \bar{d}_{\ell-1}^{m}, \nu_1, \nu_2)$$

iv. **Prolongation** of the coarse-grid correction: $\hat{e}_{\ell}^{m} = I_{\ell-1}^{\ell}\hat{e}_{\ell-1}^{m}$

v. **Apply** the correction: $u_{\ell}^{m,\text{after CGC}} = \bar{u}_{\ell}^{m} + \hat{e}_{\ell}^{m}$

(3) **Post-smoothing**: $u_{\ell}^{m+1} = \mathrm{SMOOTH}^{\nu_2}(u_{\ell}^{m,\text{after CGC}}, L_{\ell}, f_{\ell})$

The parameter γ is called the *cycle index*. In practice, only $\gamma = 1$ (V-cycle) and $\gamma = 2$ (W-cycle) are used [BHM00, p. 42]. Figure 6.5 illustrates a V- and a W-cycle with $\gamma = 2$, when using four grid levels. The black bullet • denotes a smoothing step, while ○ represents a direct solution step on the coarsest level ℓ_{c}.

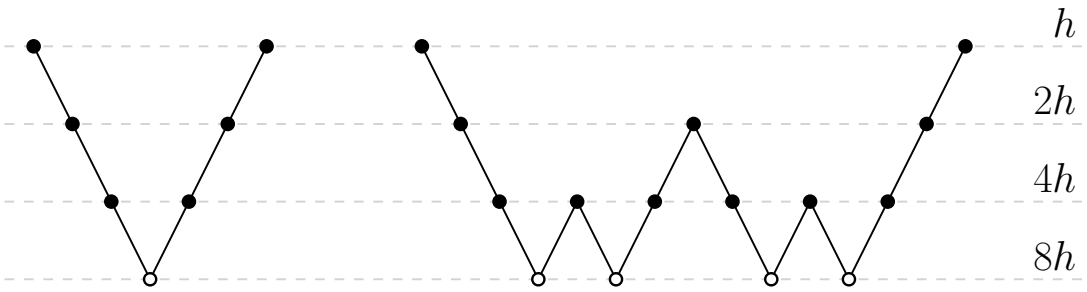


Figure 6.5 – Illustration of a V- and a W-cycle with cycle index $\gamma = 2$ and four grid levels.

6.3.5 Laplace equation on the unit square

Let us have a look at a concrete example by applying the multigrid cycle to the two-dimensional Laplace equation on the unit square $\Omega = [0,1]^2$

$$\begin{cases} -\Delta u = f(x,y) & (x,y) \in \Omega, \\ u(x,y) = 0 & (x,y) \in \partial\Omega, \end{cases} \qquad \Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}. \tag{6.8}$$

We discretize this problem with a uniform grid of $m = 2^{\ell_f} - 1$ interior grid points in each direction with $h = \frac{1}{m+1} = 2^{-\ell_f}$. The forcing term is

$$f_{i,j} = f(x_i, y_j)$$

for $i, j = 1, \ldots, m$, with $x_i = ih$, $y_j = jh$ and $u_{i,j} = u(x_i, y_j)$.

Using centered finite differences to discretize the second derivatives give

$$\frac{1}{h^2}(-u_{i-1,j} + 2u_{i,j} - u_{i+1,j}) + \frac{1}{h^2}(-u_{i,j-1} + 2u_{i,j} - u_{i,j+1}) = f_{i,j}$$

for $i, j = 1, \ldots, m$. Example for $i = j = 1$:

$$\frac{1}{h^2}(-u_{01} + 2u_{11} - u_{21}) + \frac{1}{h^2}(-u_{10} + 2u_{11} - u_{12}) = f_{11}.$$

Let U be the matrix that collects all the unknowns at the grid points. The boundary conditions do not enter in the matrix U, and they are moved to the right-hand side. For example, the previous expression is rewritten as

$$\frac{1}{h^2}(2u_{11} - u_{21}) + \frac{1}{h^2}(2u_{11} - u_{12}) = f_{11} + \frac{1}{h^2}u_{01} + \frac{1}{h^2}u_{10}.$$

The coefficients u_{11} and u_{21} appear on the first column of U, while u_{11} and u_{12} appear on the first row of U, and so on. This suggests the compact matrix form, known as *Lyapunov equation*,

$$A_{1D}U + UA_{1D} = C, \tag{6.9}$$

where

$$A_{1D} = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix},$$

and C contains the samplings of the function $f(x, y)$ on our grid plus the boundary conditions. By applying vectorization, (6.9) becomes

$$(I \otimes A_{1D} + A_{1D} \otimes I)\,\mathrm{vec}(U) = \mathrm{vec}(C).$$

For a two-dimensional problem, the discretized second derivative is

$$A_{2D} = \begin{bmatrix} 4 & -1 & & & -1 & & \\ -1 & 4 & -1 & & & \ddots & \\ & -1 & 4 & -1 & & & -1 \\ & & \ddots & \ddots & \ddots & & \\ -1 & & & \ddots & \ddots & \ddots & \\ & \ddots & & & -1 & 4 & -1 \\ & & -1 & & & -1 & 4 \end{bmatrix}.$$

It is easy to see that there is a relationship between A_{1D} and A_{2D} since

$$A_{2D} = I \otimes A_{1D} + A_{1D} \otimes I.$$

6.3.5.1 Numerical example

Here, we consider the model problem (6.8) with $f(x, y) = 0$ and a non-homogeneous Dirichlet boundary condition at the upper side of the square domain, given by

$$u(x, 1) = 1,$$

and a homogeneous Dirichlet condition elsewhere on the boundary. We solve this problem using full-weighting restriction, linear interpolation and a multigrid V-cycle with a smoother based on ω-Jacobi. The initial guess is a random grid function that satisfies the Dirichlet boundary conditions. A total of 20 multigrid V-cycles are carried out.

Figure 6.6 shows the solution and error surfaces at the 10th iteration with discretization level $\ell_{\mathrm{f}} = 5$, corresponding to a total of 1089 grid points. One can observe that the error is smooth, with a peak on the order of 10^{-5}.

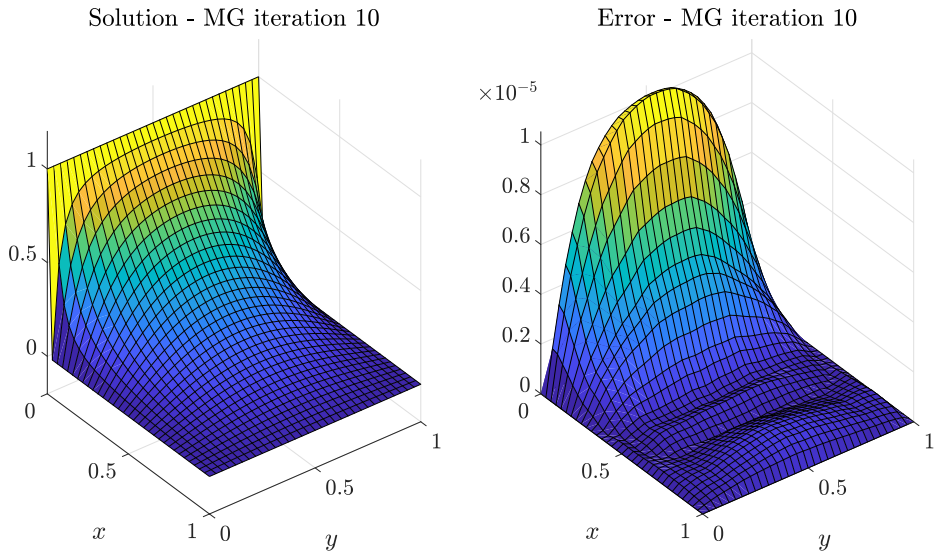


Figure 6.6 – Solution and error surfaces at the 10th iteration for the problem described in Section 6.3.5.

Figure 6.7 shows the convergence behavior of the error norm[1] for different finest levels, $\ell_{\mathrm{f}} = 7, 8, 9, 10$. It appears that twenty multigrid iterations permit to achieve an error norm in the order of 10^{-11}. The convergence behavior is essentially independent of the size of the finest grid in the multigrid cycle. As we mentioned above, this is one of the most desirable properties of multigrid.

6.4 The Full Approximation Scheme (FAS)

Although multigrid methods were originally introduced to solve large-scale linear systems deriving from the discretization of PDEs, they can also be used to solve nonlinear problems [TOS00, p. 147]. Generally speaking, there are two approaches to do this:

- *Global linearization method.* For example, in Newton's method applied to a nonlinear problem, at each iteration step one has to solve a linear system. Multigrid can be used to solve each of these linear problems.

[1]The norm considered here is the induced norm of a grid function as defined in Section 6.1.1, Equation (6.3).

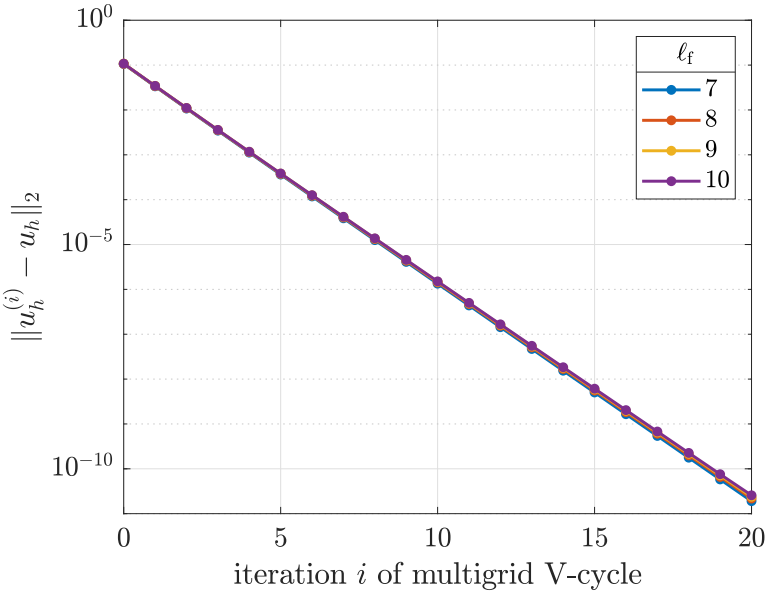


Figure 6.7 – The mesh-independent convergence of (linear) multigrid for the problem described in in Section 6.3.5. All lines are almost on top of each other.

- *Apply multigrid directly to the nonlinear problem.* The two multigrid principles (error smoothing and coarse-grid correction) are not restricted to linear problems but can be immediately extended to nonlinear problems. This leads to the *Full Approximation Scheme* (FAS). For *linear problems*, the FAS reduces to the linear multigrid we discussed above. FAS also constitutes the basis of a number of advanced numerical techniques, and can be generalized to optimization problems, as we will see in Chapter 7.

In this section, we are going to illustrate the FAS by means of the model problem

$$\begin{cases} N(u) = f^{\Omega} & \text{in } \Omega, \\ B(u) = f^{\Gamma} & \text{on } \Gamma, \end{cases} \tag{6.10}$$

where N indicates a *nonlinear* elliptic differential operator, while B is a boundary operator. By discretizing (6.10) on a finite-dimensional grid Ω_h with discretization parameter h, one gets a nonlinear system of discrete equations

$$N_h(u_h) = f_h, \tag{6.11}$$

where N_h is the discrete nonlinear operator. On the coarse grid, we have the usual coarse discretization of this operator, denoted by N_H.

Remark 6.7. In (6.11), boundary conditions have been eliminated so that they are now implicitly contained in the discrete right-hand side f_h.

6.4.1 FAS two-grid cycle

We describe one iteration cycle of the nonlinear (h, H) two-grid method for solving (6.11). The main difference with respect to linear multigrid is that here we do not work with the errors, but with the *full approximations* to the discrete solution themselves. The residual equation on Ω_h reads

$$N_h(w_h) = r_h + N_h(\bar{u}_h), \tag{6.12}$$

where $w_h = \bar{u}_h + e_h$ is the full approximation, $\bar{u}_h$ is the smoothed approximation and e_h is the error. We point out that since the discrete operator N_h is nonlinear, we have generally $N_h(\bar{u}_h + e_h) \neq N_h(\bar{u}_h) + N_h(e_h)$. This is the reason why it is not sufficient to work with the errors e_h, but we need to use the full approximations $w_h = \bar{u}_h + e_h$ instead. On the coarse grid Ω_H, Equation (6.12) is approximated by

$$N_H(w_H) = r_H + N_H(\bar{u}_H), \tag{6.13}$$

$w_H = \bar{u}_H + e_H$ being the full approximation on the coarse grid, and $\bar{u}_H = \hat{I}_h^H \bar{u}_h$. This equation has to be solved for the coarse-grid correction e_H. After solution on the coarse grid, the coarse-grid correction e_H is computed as the difference of $\bar{u}_H$ and w_h, and then it is transferred to Ω_h by using the prolongation operator I_H^h.

Observe that in FAS the restriction operator $\hat{I}_h^H$ for the relaxed approximation $\bar{u}_h$ is usually different from I_h^H, which is used to transfer the residual r_h to the coarse grid. The most common choice for $\hat{I}_h^H$ is injection, for vertex-centered grids, while for I_h^H a full-weighting restriction operator is used.

The FAS two-grid cycle is described in the box below. Here, SMOOTH stands for a nonlinear relaxation procedure having appropriate error smoothing properties, such as, for instance, nonlinear Gauss–Seidel or Jacobi method, and their weighted versions.

for $i = 1, 2, \ldots,$ **do**

(1) **Pre-smoothing**: $\bar{u}_h = \mathrm{SMOOTH}^{\nu_1}(u_h^{(i)}, N_h, f_h)$

(2) i. **Compute** residual: $r_h = f_h - N_h(\bar{u}_h)$

ii. **Restrict** the residual: $r_H = I_h^H r_h$

iii. **Restrict** the smoothed approximation: $\bar{u}_H = I_h^H \bar{u}_h$

iv. **Solve** $N_H(\bar{u}_H + e_H) = r_H + N_H(\bar{u}_H)$ for e_H

v. **Prolong** the coarse-grid correction: $e_h = I_H^h e_H$

vi. **Apply** the correction: $\hat{u}_h = \bar{u}_h + e_h$

(3) **Post-smoothing**: $u_h^{(i+1)} = \mathrm{SMOOTH}^{\nu_2}(\hat{u}_h, N_h, f_h)$

end for

6.4.2 Formulating FAS for a 2D BVP

We are now ready to show how to apply FAS for the solution of a concrete nonlinear boundary value problem. Consider the two-dimensional boundary value problem [BHM00, p. 102]

$$\begin{cases} -\Delta u(x,y) + \gamma\, u(x,y)\, e^{u(x,y)} = f(x,y) & \text{in } \Omega, \\ u(x,y) = 0 & \text{on } \partial\Omega, \end{cases} \tag{6.14}$$

with $\Omega = [0,1]^2$. We consider a uniform grid with $n = 2^{\ell_{\mathrm{f}}} + 1$ grid points in each dimension, $(x_i, y_j) = (ih, jh)$ for $0 \leqslant i, j \leqslant n-1$. As before, we discretize the second derivatives using second-order accurate centered finite differences. This leads to the discretized problem

$$\frac{-u_{i-1,j} + 2u_{i,j} - u_{i+1,j}}{h^2} + \frac{-u_{i,j-1} + 2u_{i,j} - u_{i,j+1}}{h^2} + \gamma\, u_{i,j}\, e^{u_{i,j}} = f_{i,j}$$

for $0 < i, j < n-1$. Moreover, we consider the homogeneous Dirichlet boundary conditions $u_{0,j} = u_{n-1,j} = u_{i,0} = u_{i,n-1} = 0$ for all $i,\ j$, whenever these terms appear in the equations.

Because the nonlinear component equations of the system are nonlinear, the nonlinear Gauss–Seidel method uses scalar Newton's method to solve the (i, j)th equation for $u_{i,j}$:

$$\frac{4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} + \gamma\, u_{i,j}\, e^{u_{i,j}} = f_{i,j}.$$

We get the following nonlinear equation

$$F(u_{i,j}) = 4u_{i,j} + \gamma h^2 u_{i,j}\, e^{u_{i,j}} - h^2 f_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1} - u_{i,j+1} = 0,$$

that has to be solved for $u_{i,j}$. The fixed-point iteration function of Newton's method is

$$\Phi_{\text{Newton}}(x) = x - \frac{F(x)}{F'(x)}.$$

Since the derivative of $F(u_{i,j})$ with respect to $u_{i,j}$ is

$$F'(u_{i,j}) = 4 + \gamma h^2 (1 + u_{i,j}) e^{u_{i,j}},$$

we obtain the update

$$u_{i,j} \leftarrow u_{i,j} - \frac{4u_{i,j} + \gamma h^2 u_{i,j}\, e^{u_{i,j}} - h^2 f_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1} - u_{i,j+1}}{4 + \gamma h^2 (1 + u_{i,j}) e^{u_{i,j}}}.$$

On the coarsest level $\ell_{\text{c}} = 1$, we have a 3×3 grid with only one interior point $u_{1,1}$. The equation that has to be solved for $u_{1,1}$ is

$$16u_{1,1} + \gamma\, u_{1,1}\, e^{u_{1,1}} = f_{1,1}.$$

This equation is nonlinear in $u_{1,1}$, so its solution on the coarse grid can be obtained by using Newton's method. The corresponding Newton's iteration function is

$$\Phi_{\text{Newton}}(x) = x - \frac{16x + \gamma\, x\, e^x - f_{1,1}}{16 + \gamma\, (1 + x)\, e^x}.$$

6.4.2.1 Numerical example

For problem (6.14), let us consider the forcing term

$$f(x, y) = 2((x - x^2) + (y - y^2)) + \gamma (x - x^2)(y - y^2)\, e^{(x-x^2)(y-y^2)},$$

which corresponds to the exact solution

$$u(x, y) = (x - x^2)(y - y^2).$$

We choose $\ell_{\text{f}} = 5$, corresponding to a total of 1089 grid points. We solve this problem using full-weighting restriction, linear interpolation and a FAS V-cycle with a smoother based on nonlinear Gauss–Seidel. The initial guess is a random grid function that satisfies the homogeneous Dirichlet boundary conditions. A total of twenty FAS V-cycles are carried out. Figure 6.8 illustrates the shape of the solution and the error at the last iteration. It is clear the smoothness of the solution and of the error, which is in the order of 10^{-12}.

Figure 6.9 reports on the convergence behavior of the error norm for several finest levels, $\ell_{\text{f}} = 7, 8, 9, 10$. As it appears from the numerical experiments, FAS also exhibits a nice mesh-independent convergence behavior, and 20 iterations allow to reach a plateau in the error norm at around 10^{-13}.

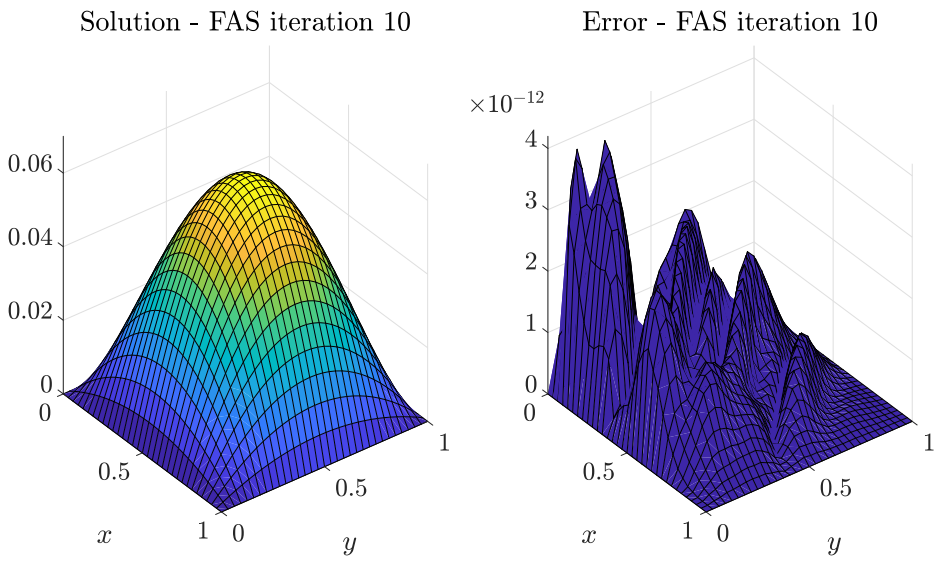


Figure 6.8 – FAS solution and error surfaces at the 10th iteration for the problem described in Section 6.4.2.

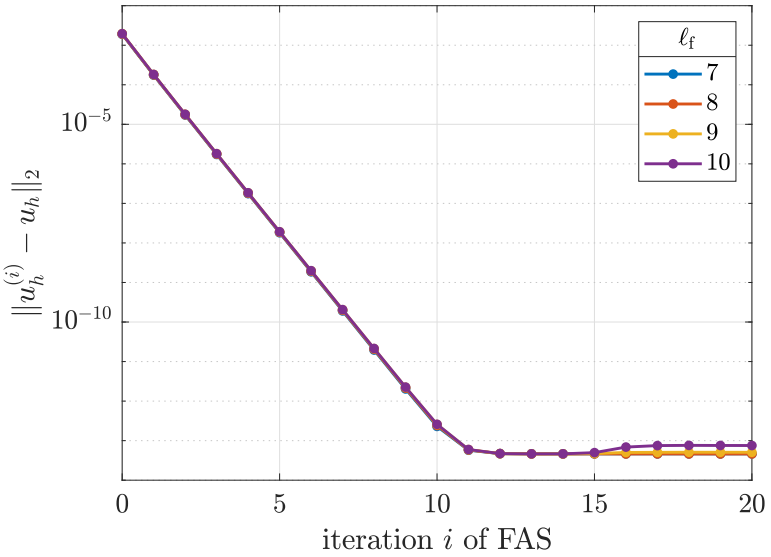


Figure 6.9 – The mesh-independent convergence of FAS for the problem described in Section 6.4.2. All lines are almost on top of each other.

Chapter 7

Multilevel Riemannian optimization for low-rank problems

7.1 Introduction

The topic of this chapter is the efficient solution of certain large-scale variational problems arising from the discretization of elliptic PDEs. We combine in particular Riemannian optimization on the manifold of fixed-rank matrices with ideas from nonlinear multigrid and multilevel optimization. The low-rank manifold will allow us to approximate the solution with significantly less degrees of freedom. In addition, the idea of recursive coarse-grid corrections from multigrid will lead to almost mesh-independent convergence of our algorithm similar to classical multigrid algorithms.

Approximating very large matrices by low rank is a popular technique to speed up numerical calculations. In the context of high-dimensional problems, this is done in so-called low-rank matrix and tensor methods, where tensors are the higher order analog of two-dimensional matrices [Hac12]. One of the early examples are low-rank solvers for the Lyapunov equation, $AX + XA^\mathsf{T} = C$, and other matrix equations; see [Sim16] for a recent overview. In order to approximate the unknown solution X by low rank, an iterative method has to be used that directly constructs the low-rank approximation. Of particular importance for this thesis are methods that accomplish this via Riemannian optimization [AMS08]: the minimization problem (obtained after a possible reformulation of the original problem) is restricted to the manifold of fixed-rank matrices thereby guaranteeing a low-rank representation of critical points. Examples of such methods are [MMBS13, Van13, Ste16] for matrix and tensor completion, [SWC12] for metric learning, [VV10, MV14, KSV16] for matrix and tensor equations, and [RO18, RNO19] for eigenvalue problems. In the context of discretized PDEs these optimization problems are very ill-conditioned, making simple first-order methods like gradient descent unmanageably slow. In [VV10, KSV16, RO18], for example, the gradient is therefore preconditioned with the inverse of the local Hessian. Solving these Hessian equations is done by a preconditioned iterative scheme, thereby mimicking the class of quasi or truncated Newton methods. We also refer to [UV19] for a recent overview of geometric methods for obtaining low-rank approximations.

Multilevel optimization is the extension of multigrid, and in particular, the full approximation scheme (FAS) to unconstrained optimization. In the MG/Opt method from [Nas00, LN05], the idea was introduced how to modify the objective functions on each scale so that they correspond to FAS coarse-grid corrections. Several extensions and theoretical convergence proofs were proposed, including optimization with trust-regions [TTWM09] and line

searches [WG09]. Related to this chapter is the low-rank multigrid method from [GH07] for matrix equations arising from the discretization of elliptic PDEs. It applies a low-rank approximation after every step of the classical multigrid algorithm from [Pen97] for the linear Sylvester matrix equation. A similar multigrid approach with truncation of the matrix iterates to low rank is used by [ES18] for the solution of large linear systems of equations arising from the finite element discretization of stochastic PDEs. Our proposed method is different in the sense that it is closer to MG/Opt and other multilevel optimization algorithms and that it works directly with the manifold of fixed-rank matrices.

This chapter is structured as follows. We first recall important ideas from multilevel optimization and the geometry of fixed-rank matrices that will be needed later on. The main contribution is in Section 7.3 where we present our new algorithm entitled Riemannian multigrid line search (RMGLS). The presentation will be sufficiently general to be applicable to any multilevel hierarchy of manifolds but the implementation will be explained only for low-rank matrices. Numerical experiments for both a linear and a nonlinear variational problem are presented in Section 7.4. Finally, in Section 7.5, we compare our method to other low-rank and multilevel methods.

7.2 Preliminaries on multilevel optimization and geometry of fixed-rank matrices

As mentioned above, our algorithm is a generalization of known (Euclidean) multilevel algorithms to Riemannian manifolds. It will then be able to calculate low-rank approximations for the variational problems discussed in Section 7.4 by minimizing a cost function over the manifold of fixed-rank matrices. Before we present this algorithm in Section 7.3, we briefly recall two important concepts for its derivation: MG/Opt [Nas00], a variant of multigrid for optimization problems, and retraction-based Riemannian optimization [AMS08], a local optimization method well suited to minimize over the set of fixed-rank matrices.

7.2.1 Multilevel optimization in Euclidean space

The full approximation scheme (FAS) presented in Section 6.4 can be generalized to a multilevel algorithm for minimizing a differentiable objective function f. The original idea goes back to the MG/Opt [Nas00, LN05]. We briefly explain the main idea for two grids since the algorithm on more grids is recursively defined from it and we will explain the algorithm for Riemannian manifolds in more detail in Section 7.3. Let the subscripts $\cdot_h$, $\cdot_H$ denote quantities on the fine $\Omega_h \simeq \mathbb{R}^n$ and the coarse grid $\Omega_H \simeq \mathbb{R}^N$, respectively. Let $f_h \colon \Omega_h \to \mathbb{R}$ be our original objective function f that we optimize with an initial guess $\bar{x}_h \in \Omega_h$ that is sufficiently smoothed. As in FAS, we introduce a modification to the coarse-grid objective function f_H. Let $g^{\mathrm{E}}(z_1, z_2) \coloneqq z_1^{\mathsf{T}} z_2$ denote the Euclidean inner product and $I_h^H \colon \Omega_h \to \Omega_H$ the restriction operator. At iteration i of MG/Opt, let $x_H^{(i)} = I_h^H \bar{x}_h \in \Omega_H$ be the iterate on the coarse grid. Then by minimizing the model $\psi_H \colon \Omega_H \to \mathbb{R}$, defined by

$$x_H \mapsto \psi_H(x_H) = f_H(x_H) - g^{\mathrm{E}}(x_H, \kappa_H), \tag{7.1}$$

with

$$\kappa_H \coloneqq \nabla f_H(x_H^{(i)}) - I_h^H \nabla f_h(\bar{x}_h), \tag{7.2}$$

one obtains a two-grid cycle for optimizing f_h. On the coarser level, the minimization of (7.1) starts at the smoothed approximation $x_H^{(i)}$, hence we can rewrite (7.1) in the following way:

find an update e_H such that

$$\psi_H(x_H^{(i)} + e_H) \coloneqq f_H(x_H^{(i)} + e_H) - g^{\mathrm{E}}(x_H^{(i)} + e_H, \kappa_H) \tag{7.3}$$

is sufficiently minimized at $x_H^{(i+1)} = x_H^{(i)} + e_H$. This coarse-grid update e_H is then transported back to the fine grid using the interpolation operator $I_H^h \colon \Omega_H \to \Omega_h$, and used to correct $\bar{x}_h$.

The linear modification (7.1) to f_H is one of the central tenets of multilevel optimization, as proposed in the MG/Opt method of [Nas00, LN05] and similar multilevel algorithms in [GST08, WG09]. The model ψ_H is actually a generalization of the coarse-grid correction equation of the FAS scheme in the context of optimization. Indeed, applying FAS for solving the nonlinear critical point equation $\nabla f_h(x) = 0$ at the approximation $x_h^{(i)}$ gives the coarse-grid correction [TOS00, Chap. (5.3.4)]

$$\nabla f_H(x_H^{(i)} + e_H) - \nabla f_H(x_H^{(i)}) = -I_h^H \nabla f_h(\bar{x}_h)$$

that has to be solved for e_H. A solution of this equation can be trivially written as

$$\nabla f_H(x_H^{(i)} + e_H) - (\ \nabla f_H(x_H^{(i)}) - I_h^H \nabla f_h(\bar{x}_h)\ ) = 0,$$

which is exactly a critical point of (7.3) with definition (7.2) for κ_H.

As in classical multigrid methods, the error has to be smooth in order to be representable on the coarse grid. For classical multigrid or FAS, iterative methods such as weighted Jacobi and Gauss–Seidel, and their nonlinear versions, can be used to smooth the error. Analogously, in the optimization framework, one can use cheap first-order optimization methods. Practice has shown that weighted versions of steepest descent, coordinate search and limited memory BFGS are effective smoothers for a wide range of large-scale multilevel optimization problems; see, e.g., [GMS+10].

Except for the introduction of the model (7.1), the principle behind the multigrid two-grid cycle remains the same in the optimization context. Figure 7.1 illustrates the two-grid cycle of a multilevel optimization scheme. The initial guess at iterate i is denoted by $x_h^{(i)}$ and the pre-smoothing update by p_h, likewise $\widehat{p}_h$ is the post-smoothing update, resulting in the next iterate $x_h^{(i+1)}$. In the next section, we will generalize this two-grid optimization cycle (and figure) to Riemannian manifolds.

7.2.2 The manifold of fixed-rank matrices

Computing a rank-k approximation to a matrix $X \in \mathbb{R}^{m\times n}$ can be seen as an optimization problem on the manifold of fixed-rank matrices

$$\mathcal{M}_k = \{X \in \mathbb{R}^{m\times n} \colon \operatorname{rank}(X) = k\}.$$

Using the SVD, one has the equivalent characterization

$$\begin{aligned}\mathcal{M}_k = \{U\Sigma V^{\mathsf{T}} \colon\ & U \in \operatorname{St}(m,k),\ V \in \operatorname{St}(n,k),\\ & \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k) \in \mathbb{R}^{k\times k},\ \sigma_1 \geqslant \cdots \geqslant \sigma_k > 0\},\end{aligned}$$

where $\operatorname{St}(m,k)$ is the Stiefel manifold of $m \times k$ real matrices with orthonormal columns (see Section 1.1.6.1), and $\operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$ is a square matrix with $\sigma_1, \sigma_2, \ldots, \sigma_k$ on its diagonal. The following proposition shows that $\mathcal{M}_k$ is indeed a smooth manifold and has a compact representation for its tangent space.

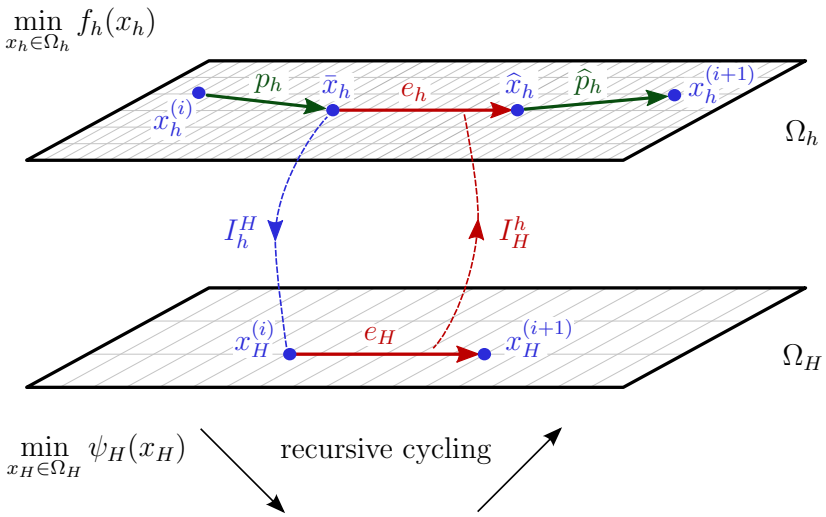


Figure 7.1 – A two-grid cycle for minimizing an objective function.

Proposition 7.1 ([Van13, Prop. 2.1]). *The set $\mathcal{M}_k$ is a smooth submanifold of dimension $(m+n-k)k$ embedded in $\mathbb{R}^{m\times n}$. Its tangent space $T_X\mathcal{M}_k$ at $X = U\Sigma V^T \in \mathcal{M}_k$ is given by*

$$T_X\mathcal{M}_k = \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \mathbb{R}^{k\times k} & \mathbb{R}^{k\times(n-k)} \\ \mathbb{R}^{(m-k)\times k} & 0_{(m-k)\times(n-k)} \end{bmatrix} \begin{bmatrix} V & V_\perp \end{bmatrix}^T. \tag{7.4}$$

In addition, every tangent vector $\xi \in T_X\mathcal{M}_k$ can be written as

$$\xi = UMV^T + U_\mathrm{p}V^T + UV_\mathrm{p}^T, \tag{7.5}$$

with $M \in \mathbb{R}^{k\times k}$, $U_\mathrm{p} \in \mathbb{R}^{m\times k}$, $V_\mathrm{p} \in \mathbb{R}^{n\times k}$ such that $U_\mathrm{p}^T U = V_\mathrm{p}^T V = 0$.

Observe that since $\mathcal{M}_k \subset \mathbb{R}^{m\times n}$, we represent tangent vectors in (7.4) and (7.5) as matrices of the same dimensions. The Riemannian metric is the restriction of the Euclidean metric on $\mathbb{R}^{m\times n}$ to the submanifold $\mathcal{M}_k$,

$$g_X(\xi,\eta) = \langle \xi, \eta \rangle = \operatorname{trace}(\xi^\mathsf{T}\eta), \quad \text{with } X \in \mathcal{M}_k \text{ and } \xi, \eta \in T_X\mathcal{M}_k.$$

The Riemannian gradient of a smooth function $f\colon \mathcal{M}_k \to \mathbb{R}$ at $X \in \mathcal{M}_k$ is defined as the unique tangent vector $\operatorname{grad} f(X)$ in $T_X\mathcal{M}_k$ such that

$$\langle \operatorname{grad} f(X), \xi \rangle = \mathrm{D}\, f(X)[\xi] \quad \text{for all } \xi \in T_X\mathcal{M}_k,$$

where $\mathrm{D}\, f$ denotes the directional derivatives of f. More concretely, for embedded submanifolds, the Riemannian gradient is given by the orthogonal projection onto the tangent space of the Euclidean gradient of f seen as a function on the embedding space $\mathbb{R}^{m\times n}$; see, e.g., [AMS08, Eq. (3.37)]. Defining $\mathrm{P}_U = UU^\mathsf{T}$ and $\mathrm{P}_U^\perp = I - \mathrm{P}_U$ for any $U \in \mathrm{St}(m,k)$, the orthogonal projection onto the tangent space at X is [Van13, Eq. (2.5)]

$$\mathrm{P}_{T_X\mathcal{M}_k}\colon \mathbb{R}^{m\times n} \to T_X\mathcal{M}_k, \quad Z \mapsto \mathrm{P}_U\, Z\, \mathrm{P}_V + \mathrm{P}_U^\perp\, Z\, \mathrm{P}_V + \mathrm{P}_U\, Z\, \mathrm{P}_V^\perp.$$

Then, denoting $\nabla f(X)$ the Euclidean gradient of f at X, the Riemannian gradient is given by

$$\operatorname{grad} f(X) = \mathrm{P}_{T_X\mathcal{M}_k}(\nabla f(X)). \tag{7.6}$$

7.2.3 The orthographic retraction

A retraction is a smooth map from the tangent space to the manifold, $R_X : T_X\mathcal{M}_k \to \mathcal{M}_k$, used to map tangent vectors to points on the manifold (see Section 1.2.1). It is, essentially, any smooth first-order approximation of the exponential map of the manifold; see, e.g., [AM12, Definition 1]. In order to establish convergence of the Riemannian algorithms, it is sufficient for the retraction to be defined only locally.

In our setting, we have chosen the orthographic retraction on $\mathcal{M}_k$. The reason for this choice is that for the orthographic retraction we have explicit expressions for the retraction and its inverse. Given a point $X = U\Sigma V^\mathsf{T} \in \mathcal{M}_k$ and a tangent vector ξ in the format (7.5), the retraction of ξ at X is given by [AO15, §3.2]

$$R_X(\xi) = [U(\Sigma + M) + U_\mathrm{p}]\,(\Sigma + M)^{-1}\,[(\Sigma + M)V^\mathsf{T} + V_\mathrm{p}^\mathsf{T}]. \tag{7.7}$$

Figure 7.2 illustrates the orthographic retraction. As a special case, observe that if X is full rank, then $UU^\mathsf{T} = U^\mathsf{T}U = I$ and $VV^\mathsf{T} = V^\mathsf{T}V = I$, therefore $U_\mathrm{p} = 0$ and $V_\mathrm{p} = 0$, so $R_X(\xi) = U(\Sigma + M)V^\mathsf{T} = X + \xi$.

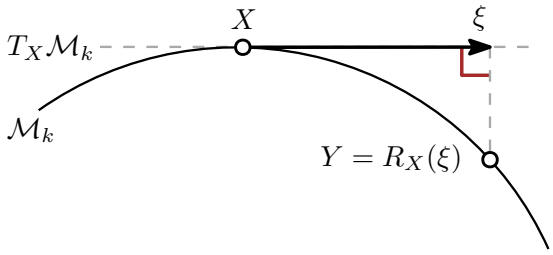


Figure 7.2 – The orthographic retraction.

The inverse of the orthographic retraction is simply given by the orthogonal projection of $Y - X$ on $T_X\mathcal{M}_k$:

$$\xi \coloneqq R_X^{-1}(Y) = \mathrm{P}_{T_X\mathcal{M}_k}(Y - X) = \mathrm{P}_{T_X\mathcal{M}_k}(Y) - X. \tag{7.8}$$

Equivalently, this can be written in tangent vector format (7.5) with the factors

$$M_\xi = U^\mathsf{T}YV - \Sigma, \qquad U_{p,\xi} = (I - UU^\mathsf{T})YV, \qquad V_{p,\xi} = (I - VV^\mathsf{T})Y^\mathsf{T}U.$$

When implementing R_X and R_X^{-1}, it is important to exploit the factored forms of the rank-k matrices X and Y, and the parametrization (7.5) of the tangent vector ξ. In that case, the flop counts of R_X and R_X^{-1} are both $O(nk^2 + k^3)$. See also [AO15, §3.2].

7.3 Riemannian multigrid line search for low-rank matrices

In this section, we describe the central contribution of this chapter: a Riemannian multilevel linesearch algorithm, called RMGLS, for the approximate low-rank solution of optimization problems. We detail how the two-grid optimization cycle of MG/Opt can be generalized to the retraction-based framework for the geometry of fixed-rank matrices, both of which were described in the previous section.

Our algorithm involves the classical components of multigrid (smoothers, prolongation and restriction operators, and a coarse-grid correction) and Riemannian optimization (line search, retractions, gradients). Since this generalization is possible for other types of manifolds, we have presented it with general manifolds in mind. However, remarks on the implementation apply only to the manifold of fixed-rank matrices.

7.3.1 Description of the scheme

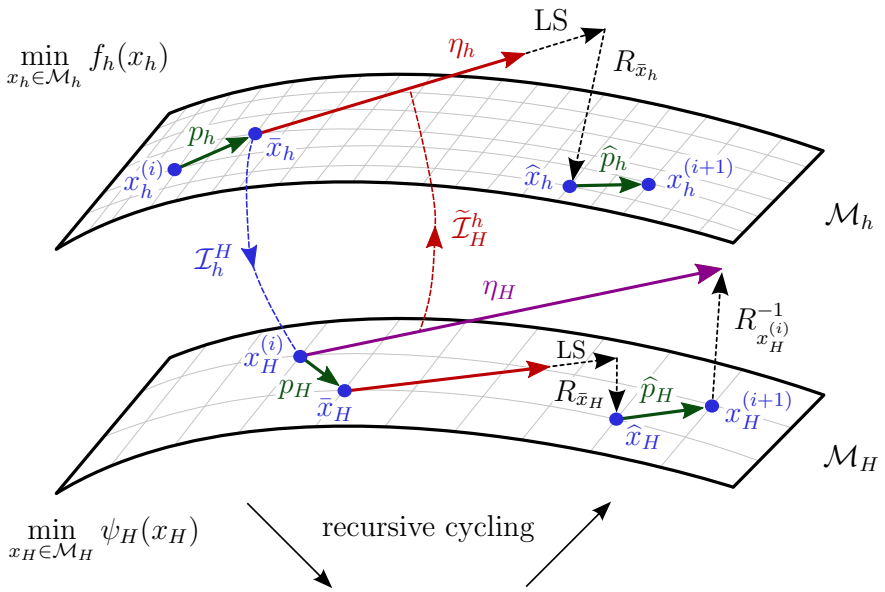


Figure 7.3 – The Riemannian multigrid line search (RMGLS) scheme. The coarse-grid correction is computed either directly or by a recursive application of RMGLS. It is instructive to compare this figure to the Euclidean version in Figure 7.1.

We first describe the algorithm for a two-grid cycle, making reference to Figure 7.3. Recall that quantities related to the fine grid and to the coarse grid are denoted by the subscripts $\cdot_h$ and $\cdot_H$, respectively. For example, $\mathcal{M}_h$ and $\mathcal{M}_H$ are the fine and coarse-scale manifolds, respectively.

Starting from an approximation $x_h^{(i)}$ on $\mathcal{M}_h$, we first perform some pre-smoothing steps, then the smoothed approximation $\bar{x}_h$ is restricted to $\mathcal{M}_H$. This gives us $x_H^{(i)}$, for which we compute a correction η_H on $\mathcal{M}_H$. If $\mathcal{M}_H$ is a sufficiently small manifold, η_H is computed directly with a trust-region method to minimize ψ_H. Otherwise, it is the inverse retraction of the result $x_H^{(i+1)}$ obtained from the recursive application of the two-grid scheme with $\mathcal{M}_H$ as fine-scale manifold. In the figure, the latter option is depicted for illustration, including the steps performed on $\mathcal{M}_H$. In both cases, the interpolation η_h of the coarse-scale correction η_H to the fine scale is applied to $\bar{x}_h$ via line search. The updated approximation $\widehat{x}_h$ is then post-smoothed and we finally obtain $x_h^{(i+1)}$ as result of one iteration of RMGLS.

An important difference compared to multilevel optimization on Euclidean space is the explicit difference between the approximations $x_h^{(i)}, \bar{x}_h, \widehat{x}_h, x_h^{(i+1)}, x_H^{(i)}, x_H^{(i+1)}$ that are points on the manifolds $\mathcal{M}_h$ and $\mathcal{M}_H$, and the updates and corrections $p_h, \widehat{p}_h, \eta_h, \eta_H$ that are tangent vectors on the tangent spaces of $\mathcal{M}_h$ and $\mathcal{M}_H$. This is also clearly visible in Figure 7.3 where the approximations are depicted as full circles and tangent vectors as arrows.

In the next subsections, we will explain every component of the algorithm, except for the line search, which has been explained in Chapter 5. The final algorithm in pseudocode is listed in Section 7.3.7.

7.3.2 Tensor-product multigrid

Observe that a matrix in $\mathbb{R}^{n\times n}$ can be regarded as an element of the tensor-product space $\mathbb{R}^n \otimes \mathbb{R}^n \simeq \mathbb{R}^{n\times n}$. Starting from this observation, it is possible to construct a multigrid algorithm by taking tensor products of standard multigrid components. This approach is known as *tensor-product multigrid* [RW95, Pen97].

For example, let $I_h^H : \mathbb{R}^n \to \mathbb{R}^N$ and $I_H^h : \mathbb{R}^N \to \mathbb{R}^n$ denote the standard restriction and prolongation operators for a linear multigrid algorithm with $\mathbb{R}^n$ the fine and $\mathbb{R}^N$ the coarse grid. Let ℓ_f denote the fine-scale level, $h = 2^{-\ell_\mathrm{f}}$, $H = 2h$, $n = 2^{\ell_\mathrm{f}} - 1$, and $N = 2^{\ell_\mathrm{f}-1} - 1$. Then in 1D the restriction I_h^H could be the $N \times n$ injection matrix defined as

$$(I_h^H)_{ij} = \begin{cases} 1, & \text{if } j = 2i; \\ 0, & \text{otherwise.} \end{cases}$$

Some concrete instances of transfer operators in 1D are given in Section 6.3.2.

Higher-order extensions for I_h^H and I_H^h, like full weighting and linear interpolation, are defined analogously; see [TOS00]. Following the tensor-product idea, we can then easily construct a *restriction operator* on the space of matrices by applying I_h^H to the rows and columns of X,

$$\mathcal{I}_h^H : \mathbb{R}^{n\times n} \to \mathbb{R}^{N\times N}, \quad X \mapsto I_h^H X (I_h^H)^\mathsf{T}. \tag{7.9}$$

Likewise, an *interpolation operator* for matrices is constructed as

$$\mathcal{I}_H^h : \mathbb{R}^{N\times N} \to \mathbb{R}^{n\times n}, \quad X \mapsto I_H^h X (I_H^h)^\mathsf{T}.$$

Hence, we have obtained transfer operators between the fine and coarse grids $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{N\times N}$, respectively.

7.3.3 Riemannian transfer operators

In our setting, the transfer operators from above are to be applied to rank-k matrices. Let us denote these manifolds by $\mathcal{M}_h^k \subset \mathbb{R}^{n\times n}$ and $\mathcal{M}_H^k \subset \mathbb{R}^{N\times N}$.

First, we can directly compute the restriction from $\mathcal{M}_h^k$ to $\mathcal{M}_H^k$ by (7.9) since both manifolds are embedded in matrix space. It is clear from (7.9) that $\operatorname{rank}(\mathcal{I}_h^H(X_h)) \leqslant k$ if X_h is a rank-k matrix. In numerical calculations, the rank of $\mathcal{I}_h^H(X_h)$ is always equal to k, but if it were strictly less we could simply reduce the defining rank of the coarse manifold.[1] The computation of $\mathcal{I}_h^H(X_h)$ is carried out directly on its factorized SVD form, and followed by a reorthogonalization to preserve the SVD format of the result. The entire procedure is summarized in the following box.

Restriction of $X_h = U_h \Sigma_h V_h^\mathsf{T} \in \mathcal{M}_h^k$:

(1) Compute **compact QRs**: $Q_U R_U = I_h^H U_h$ and $Q_V R_V = I_h^H V_h$

(2) Compute **compact SVD**: $\widehat{U}\widehat{\Sigma}\widehat{V}^\mathsf{T} = R_U \Sigma_h R_V^\mathsf{T}$

(3) Compute **factors**: $U_H = Q_U \widehat{U},\ \Sigma_H = \widehat{\Sigma},\ V_H = Q_V \widehat{V}$

Result is $X_H = U_H \Sigma_H V_H^\mathsf{T} \in \mathcal{M}_H^{\bar{k}}$ in SVD form, with $\bar{k} = \operatorname{rank}(X_H)$.

[1] In the next step of our algorithm RMGLS, the rank of the coarse iterate will typically grow after smoothing and we can then again continue with $\mathcal{M}_H^k$ as our coarse manifold.

Figure 7.4 illustrates the restriction procedure on the low-rank format.

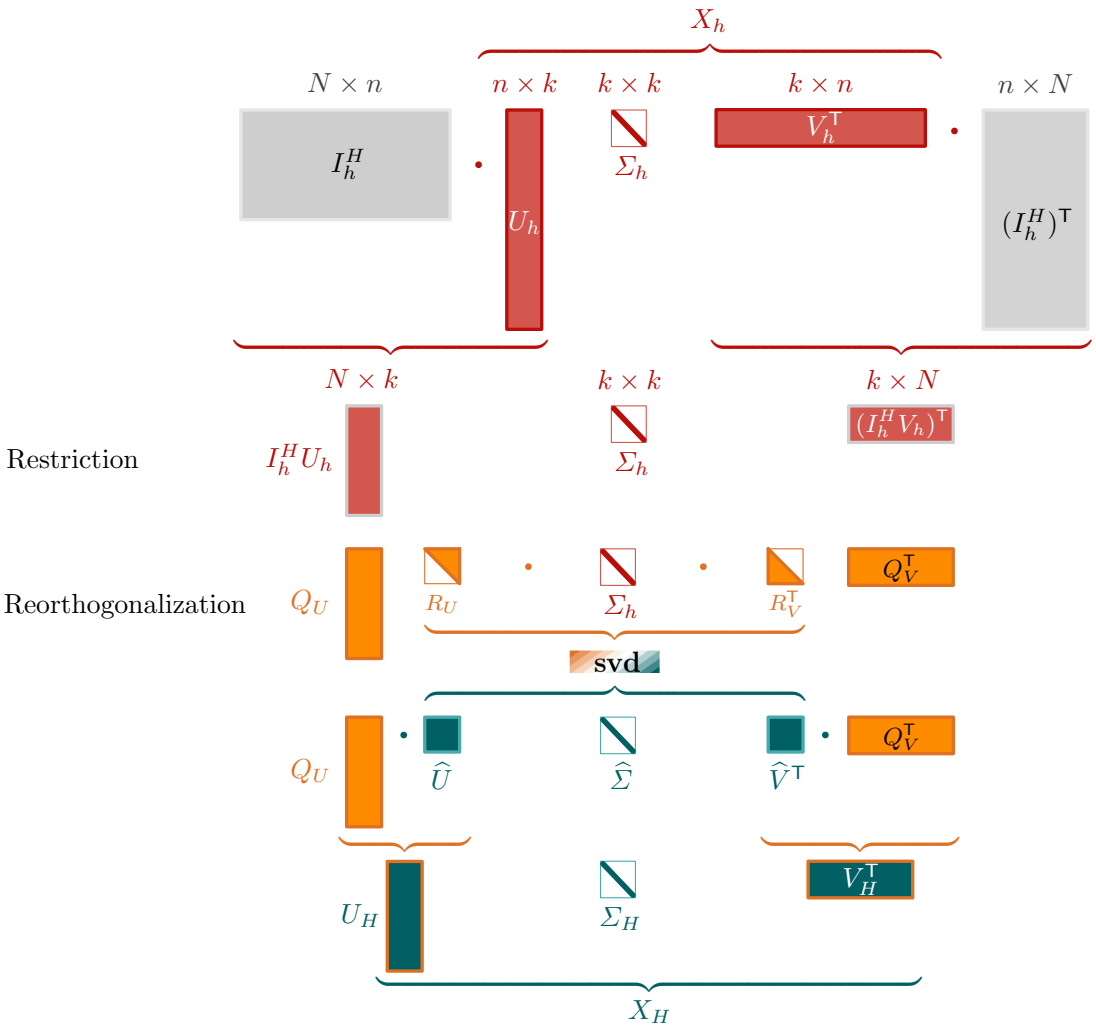


Figure 7.4 – Restriction operator on the low-rank format.

Next, when transferring tangent vectors between manifolds of different scales, the result of the transfer operators is not necessarily in the tangent space at the transferred points. We therefore follow the transfer operators by an orthogonal projection onto the new tangent space,

$$\widetilde{\mathcal{I}}_h^H = \mathrm{P}_{T_{X_H}\mathcal{M}_H} \circ \mathcal{I}_h^H|_{T_{X_h}\mathcal{M}_h} \quad \text{and} \quad \widetilde{\mathcal{I}}_H^h = \mathrm{P}_{T_{X_h}\mathcal{M}_H} \circ \mathcal{I}_H^h|_{T_{X_H}\mathcal{M}_h}. \tag{7.10}$$

This projection step is related to the so-called vector transport in retraction-based Riemannian optimization and can be seen as a first-order approximation of parallel transport in Riemannian geometry; see [AMS08]. As explained in the box below, the computation of the interpolation $\widetilde{\mathcal{I}}_h^H$ exploits the factored form of tangent vectors. The implementation of the restriction $\widetilde{\mathcal{I}}_H^h$ is similar and omitted.

Interpolation of $\xi_H = U_H M_H V_H + U_{p,H} V_H^\mathsf{T} + U_H V_{p,H}^\mathsf{T} \in T_{X_H}\mathcal{M}_H^k$
Required: $X_h = U_h \Sigma_h V_h^\mathsf{T} \in \mathcal{M}_h^k$ and $X_H = U_H \Sigma_H V_H^\mathsf{T} \in \mathcal{M}_H^k$

(1) Compute **factors**: $\widehat{U}_{p,h} = I_H^h U_{p,H},\ \widehat{M}_h = M_H,\ \widehat{V}_{p,h} = I_H^h V_{p,H}$

(2) **Normalize**: $U_{p,h} = (I - U_h U_h^\mathsf{T})\,\widehat{U}_{p,h},\ V_{p,h} = (I - V_h V_h^\mathsf{T})\,\widehat{V}_{p,h}$
$M_h = U_h^\mathsf{T}\widehat{U}_{p,h} + \widehat{V}_{p,h}^\mathsf{T} V_h + \widehat{M}_h$

Result is $\xi_h = U_h M_h V_h + U_{p,h} V_h^\mathsf{T} + U_h V_{p,h}^\mathsf{T} \in T_{X_h}\mathcal{M}_h^k$ in the form (7.5).

Like in [WG09], we will use injection and linear interpolation in the numerical experiments. In that case, the flop counts for computing $\mathcal{I}_h^H$, $\widetilde{\mathcal{I}}_H^h$, and $\widetilde{\mathcal{I}}_h^H$ in factored form as explained above are both $O(nk^2 + k^3)$ for $\mathcal{M}_h^k \subset \mathbb{R}^{n\times n}$.

7.3.4 Smoothers

In the context of optimization on manifolds, a smoother can be any cheap first-order optimization method for minimizing f_h: given $x_h^{(i)}$, it returns a tangent vector ξ_h such that, after retraction, the error of the new iterate $\bar{x}_h = R_{x_h^{(i)}}(\xi_h)$ is smooth. In the Euclidean multilevel algorithm of [WG09], for example, a few steps of L-BFGS are used.

In our experiments, we simply use a fixed number of steps of Riemannian steepest descent; see [AMS08]. In addition, we halve the step length found by the line-search method so that the resulting step better approximates one step of the Richardson iteration in linear multigrid.

7.3.5 The Riemannian coarse-grid correction

Similar to Euclidean multilevel optimization, explained in Section 7.2.1, we also modify the objective function in the Riemannian setting. To illustrate the generalization to the manifold case, let us first rewrite the Euclidean model (7.3) as

$$\psi_H^{\text{Euclidean}} \colon \mathbb{R}^n \to \mathbb{R}, \quad x_H \mapsto \psi_H^{\text{Euclidean}}(x_H) = f_H(x_H) - g^{\mathrm{E}}(x_H, \kappa_H), \tag{7.11}$$

where $x_H \coloneqq x_H^{(i)} + e_H$ is the full approximation. In the following, we describe how we turn this model into a function on manifolds.

Let us assume that the algorithm at the coarse level starts at $x_H^{(i)} \in \mathcal{M}_H^k$. We consider as optimization variable a point on the manifold $x_H \in \mathcal{M}_H^k$. In the Riemannian setting, such a point x_H cannot be evaluated as in (7.11) since the inner product $g^{\mathrm{E}}(x_H, \kappa_H)$ is only defined for tangent vectors. We will therefore lift x_H to the tangent space at $x_H^{(i)}$ by means of the inverse retraction when evaluating the inner product.[2] A coarse objective function suitable for Riemannian optimization is therefore given by

$$\psi_H \colon \mathcal{M}_H^k \to \mathbb{R}, \quad x_H \mapsto \psi_H(x_H) = f_H(x_H) - g_{x_H^{(i)}}(R_{x_H^{(i)}}^{-1}(x_H), \kappa_H), \tag{7.12}$$

where $R_{x_H^{(i)}}^{-1}$ is the inverse retraction at $x_H^{(i)}$, $g_{x_H^{(i)}}$ denotes the Riemannian metric at $x_H^{(i)}$, and $\kappa_H \in T_{x_H^{(i)}}\mathcal{M}_H^k$ is defined as

$$\kappa_H = \operatorname{grad} f_H(x_H^{(i)}) - \widetilde{\mathcal{I}}_h^H(\operatorname{grad} f_h(\bar{x}_h)). \tag{7.13}$$

[2] Recall from Section 7.2.3 that this inverse is easy to compute for the orthographic retraction.

Here, grad denotes the Riemannian gradient and $\widetilde{\mathcal{I}}_h^H(\operatorname{grad} f_h(\bar{x}_h))$ is the restricted Riemannian gradient coming from the fine-scale manifold. The restriction operator $\widetilde{\mathcal{I}}_h^H$ is defined as in (7.10), and the subtraction of the two tangent vectors is carried out in the factored format (7.5). Let us denote by $x_H^{(i+1)}$ the approximate minimizer of ψ_H, and define the tangent vector $\eta_H \coloneqq R^{-1}_{x_H^{(i)}}(x_H^{(i+1)})$.

7.3.6 Gradient of the coarse-grid model

During the optimization process, we need the Riemannian gradient of the coarse-grid correction function ψ_H. Recall from (7.6) that this is simply the orthogonal projection of the Euclidean gradient onto the tangent space.

To this end, let us simplify the notation by omitting $\cdot_H$ in (7.12) to denote ψ_H as

$$\psi(x) = f(x) - g_{x^{(i)}}(R^{-1}_{x^{(i)}}(x), \kappa), \tag{7.14}$$

where $x, x^{(i)} \in \mathcal{M}_k$ and where the tangent vector $\kappa \in T_{x^{(i)}}\mathcal{M}_k$ does not depend on x; see (7.13). The only difficulty is thus the Euclidean gradient of the second term in (7.14). Thanks to our choice of Riemannian metric on $\mathcal{M}_k$, we have

$$g_{x^{(i)}}(R^{-1}_{x^{(i)}}(x), \kappa) = \langle\, R^{-1}_{x^{(i)}}(x),\, \kappa \,\rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product of two matrices. By the chain rule, the Euclidean gradient of $\langle\, R^{-1}_{x^{(i)}}(x),\, \kappa \,\rangle$ can therefore be written as the directional derivative

$$\nabla \langle\, R^{-1}_{x^{(i)}}(x),\, \kappa \,\rangle = \langle\, \nabla R^{-1}_{x^{(i)}}(x),\, \kappa \,\rangle = \mathrm{D}\, R^{-1}_{x^{(i)}}(x)[\kappa].$$

For the orthographic retraction R_x, we know from (7.8) that its inverse satisfies

$$R^{-1}_{x^{(i)}}(x) = \mathrm{P}_{T_{x^{(i)}}\mathcal{M}_k}(x) - x^{(i)}.$$

Since this is an affine linear function in x, its Fréchet derivative is simply the orthogonal projection. We therefore obtain

$$\mathrm{D}\, R^{-1}_{x^{(i)}}(x)[\kappa] = \mathrm{P}_{T_{x^{(i)}}\mathcal{M}_k}(\kappa) = \kappa,$$

since $\kappa \in T_{x^{(i)}}\mathcal{M}_k$ by construction. Combining, we finally obtain the Riemannian gradient of ψ as

$$\operatorname{grad} \psi(x) = \mathrm{P}_{T_{x^{(i)}}\mathcal{M}_k}(\nabla f(x)) - \kappa.$$

Remark 7.2. In the Euclidean multilevel optimization method from [WG09, Eq. (2.6)], an important property called *first-order coherence* is introduced. In our Riemannian setting, it amounts to

$$g_{x_H}(\operatorname{grad} \psi_H(x_H), \xi_H) = g_{x_h}(\operatorname{grad} f_h(x_h), \xi_h),$$

for any search direction $\xi_H \in T_{x_H}\mathcal{M}_H$ with $x_H = \mathcal{I}_h^H(x_h)$ and $\xi_h = \widetilde{\mathcal{I}}_H^h(\xi_H)$. This is a desirable property since it ensures the same slope of the objective functions on the fine and coarse grids. Practically, this equation imposes a relation between the intergrid transfer operators in the multilevel algorithm. In our setting as explained in Section 7.3.3, one can show that it requires $I_H^h = (I_h^H)^\mathsf{T}$. This is indeed a typical choice in multigrid algorithms. It is, for example, satisfied for the injection I_H^h and linear interpolation I_h^H.

7.3.7 Final algorithm: Riemannian multigrid line search

In the following box, we have listed the final Riemannian multigrid line search algorithm to optimize an objective function on a Riemannian manifold. The smoother is denoted by the function SMOOTH and corresponds to ν_1 or ν_2 steps of steepest descent for f_h.

One RMGLS iteration starting at $x_h^{(i)}$ to minimize f_h.

(1) **Pre-smoothing**: $\bar{x}_h = \mathrm{SMOOTH}^{\nu_1}(x_h^{(i)}, f_h)$

(2) **Coarse-grid correction**:

(a) **Restrict** to the coarse manifold: $x_H^{(i)} = \mathcal{I}_h^H(\bar{x}_h)$

(b) Compute the **linear correction term**:

$$\kappa_H = \operatorname{grad} f_H(x_H^{(i)}) - \widetilde{\mathcal{I}}_h^H(\operatorname{grad} f_h(\bar{x}_h))$$

(c) Define the **coarse-grid objective function**

$$\psi_H(x_H) = f_H(x_H) - g_{x_H^{(i)}}(R^{-1}_{x_H^{(i)}}(x_H), \kappa_H)$$

(d) Compute an **approximate minimizer** $x_H^{(i+1)}$ starting at $x_H^{(i)}$ to minimize ψ_H using either

- a Riemannian trust-region method (if $\mathcal{M}_H$ is small)
- one recursive RMGLS iteration (otherwise)

(e) Compute the **coarse-grid correction**: $\eta_H = R^{-1}_{x_H^{(i)}}(x_H^{(i+1)})$

(f) **Interpolate** to the fine manifold: $\eta_h = \widetilde{\mathcal{I}}_H^h(\eta_H)$

(g) Compute the **corrected approximation** on the fine manifold:

$$\widehat{x}_h = R_{\bar{x}_h}(\alpha^* \eta_h) \quad \text{with } \alpha^* \text{ obtained from line search}$$

(3) **Post-smoothing**: $x_h^{(i+1)} = \mathrm{SMOOTH}^{\nu_2}(\widehat{x}_h, f_h)$

Remark 7.3. The RMGLS algorithm above is very similar to the way one FAS multigrid iteration is presented in [TOS00, p. 157].

Remark 7.4. For efficiency reasons, it is crucial to implement the algorithm without forming full matrices, i.e., always exploiting the low-rank format explicitly, also when evaluating the objective function f. More details will be given in Section 7.4.

7.3.8 Riemannian Hager–Zhang line search

The Riemannian Hager–Zhang line search explained in Chapter 5 can be immediately used on the manifold of fixed-rank matrices $\mathcal{M}_k$. As mentioned above, we have chosen the orthographic retraction because it has an explicit inverse; see Section 7.2.3. Let us show that its derivative can also be efficiently calculated.

Let $X \in \mathcal{M}_k$ and R_X the orthographic retraction. Recall that $f\colon \mathcal{M}_k \to \mathbb{R}$. By the chain rule, we get for $\phi(t) = f(R_X(t\eta))$ that

$$\phi'(t) = \langle \nabla f(R_X(t\eta)), \tfrac{\mathrm{d}}{\mathrm{d}t} R_X(t\eta) \rangle = \operatorname{trace}\Big(\nabla f(R_X(t\eta))^{\mathsf{T}} \tfrac{\mathrm{d}}{\mathrm{d}t} R_X(t\eta) \Big), \tag{7.15}$$

where ∇f is the Euclidean gradient of f. Using (7.7), we can work out the standard derivative

$$\begin{aligned} \tfrac{\mathrm{d}}{\mathrm{d}t} R_X(t\eta) = {} & (U(\Sigma + tM) + tU_{\mathrm{p}})(\Sigma + tM)^{-1}(MV^{\mathsf{T}} + V_{\mathrm{p}}^{\mathsf{T}}) \\ & - (U(\Sigma + tM) + tU_{\mathrm{p}})(\Sigma + tM)^{-1} M (\Sigma + tM)^{-1}((\Sigma + tM)V^{\mathsf{T}} + tV_{\mathrm{p}}^{\mathsf{T}}) \\ & + (U_{\mathrm{p}} + UM)(\Sigma + tM)^{-1}((\Sigma + tM)V^{\mathsf{T}} + tV_{\mathrm{p}}^{\mathsf{T}}), \end{aligned}$$

where $X = U\Sigma V^{\mathsf{T}}$ and $\eta = UMV^{\mathsf{T}} + U_{\mathrm{p}}V^{\mathsf{T}} + UV_{\mathrm{p}}^{\mathsf{T}}$ as in (7.5).

In (7.15), we need to evaluate $\operatorname{trace}(A^{\mathsf{T}}B)$. For computational efficiency, we want to avoid the naive multiplication $A^{\mathsf{T}}B$ since it costs $O(n^3)$ flops. A more efficient approach is to rewrite the derivative in the factorized format $\frac{\mathrm{d}}{\mathrm{d}t} R_X(t\eta) = GH^{\mathsf{T}}$ by defining

$$G = \begin{bmatrix} -(U + tU_{\mathrm{p}}(\Sigma + tM)^{-1})M(\Sigma + tM)^{-1} & U + tU_{\mathrm{p}}(\Sigma + tM)^{-1} & U_{\mathrm{p}} + UM \end{bmatrix}$$

and

$$H = \begin{bmatrix} V(\Sigma + tM)^{\mathsf{T}} + tV_{\mathrm{p}} & VM^{\mathsf{T}} + V_{\mathrm{p}} & V + tV_{\mathrm{p}}(\Sigma + tM)^{-\mathsf{T}} \end{bmatrix}.$$

Observe that $G, H \in \mathbb{R}^{n \times 3k}$. Assuming a similar factorization for $\nabla f(R_X(t\eta)) = \widetilde{G}\widetilde{H}^{\mathsf{T}}$ with $\widetilde{G}, \widetilde{H} \in \mathbb{R}^{n \times \widetilde{k}}$, the trace in (7.15) can then be computed as

$$\phi'(t) = \operatorname{trace}(\widetilde{H}\widetilde{G}^{\mathsf{T}}GH^{\mathsf{T}}) = \operatorname{trace}((\widetilde{G}^{\mathsf{T}}G)(H^{\mathsf{T}}\widetilde{H}))$$

at a cost of $O((n + k)k\widetilde{k})$. In typical applications targeting low-rank approximations, $\widetilde{k}$ is larger than k but significantly smaller than n. For example, in our numerical experiments below, $\widetilde{k} = O(k^2)$ showing a large reduction from $O(n^3)$ when k is small. Figure 7.5 provides an illustration of this computational trick.

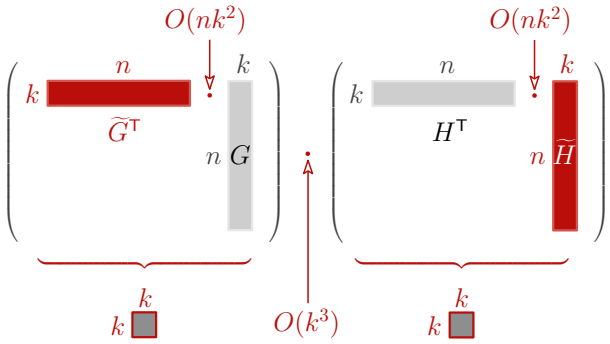


Figure 7.5 – The “trace trick”.

7.4 Numerical experiments for two variational problems

We report on numerical properties of the proposed algorithm, RMGLS, by applying it to the variational problems presented in this section. These are large-scale finite-dimensional optimization problems arising from the discretization of infinite-dimensional problems. Because

of their underlying PDEs, these variational problems present a natural multilevel structure. Variational problems of this type have been considered as benchmarks in other nonlinear multilevel algorithms [Hen03, GST08, WG09]. For the theoretical aspects of variational problems, some good references are [BS07, LDL16].

The experiments below were performed by recursively executing RMGLS in a V-cycle manner for both problems, as explained in Section 7.3.7. Unless otherwise noted, the Riemannian version of the Hager–Zhang line search was used. The algorithm was implemented in MATLAB and it is available on Yareta.

7.4.1 A linear problem (Lyapunov equation)

We consider the minimization problem

$$\begin{cases} \min_w \mathcal{F}(w(x,y)) = \int_\Omega \frac{1}{2}\|\nabla w(x,y)\|^2 - \gamma(x,y)\, w(x,y)\,\mathrm{d}x\,\mathrm{d}y \\ \quad \text{such that} \quad w = 0 \text{ on } \partial\Omega, \end{cases} \tag{7.16}$$

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$, $\Omega = [0,1]^2$ and γ is a function that satisfies homogeneous Dirichlet boundary conditions on $\partial\Omega$. The variational derivative (Euclidean gradient) of $\mathcal{F}$ is

$$\frac{\delta \mathcal{F}}{\delta w} = -\Delta w - \gamma. \tag{7.17}$$

A critical point of (7.16) is thus also a solution of the elliptic PDE $-\Delta w = \gamma$.

7.4.1.1 Discretization of the objective function

We use a standard finite difference discretization for (7.16). In particular, Ω is represented at level ℓ as a square grid

$$\Omega_\ell = \{(x_i, y_j) \mid x_i = ih_\ell,\ y_j = jh_\ell,\ i = 0,1,\ldots,n_\ell,\ j = 0,1,\ldots,n_\ell\}, \quad n_\ell = 2^\ell,$$

yielding a square mesh of uniform mesh width $h_\ell = 1/n_\ell$. The unknown w on Ω_ℓ is denoted by $w_{ij} \coloneqq w(x_i, y_j)$, and likewise for $\gamma_{ij} \coloneqq \gamma(x_i, y_j)$, where we have omitted the dependence on ℓ in the notation for readability. The partial derivatives are discretized as forward finite differences

$$\partial w_{x_{ij}} = \tfrac{1}{h}(w_{i+1,j} - w_{i,j}), \qquad \partial w_{y_{ij}} = \tfrac{1}{h}(w_{i,j+1} - w_{i,j}). \tag{7.18}$$

The discretized version of $\mathcal{F}$ therefore becomes

$$\mathcal{F}_h = h^2 \sum_{i,j=0}^{2^\ell - 1} (\tfrac{1}{2}(\partial w_{x_{ij}}^2 + \partial w_{y_{ij}}^2) - \gamma_{ij}\, w_{ij}). \tag{7.19}$$

In order to find a low-rank approximation of (7.16) with RMGLS, the unknown w_{ij} from above will be approximated as the ijth entry of a matrix $W_h \in \mathbb{R}^{n\times n}$ of rank k. For efficiency, this matrix is always represented in the factored form $W_h = U\Sigma V^\mathsf{T}$. Likewise, we gather all γ_{ij} in a factored matrix $\Gamma_h = U_\gamma \Sigma_\gamma V_\gamma^\mathsf{T}$ of rank k_γ. In the experiments below, $k_\gamma = 5$.

For reasons of computational efficiency, it is important to exploit these low-rank forms in the execution of RMGLS. For the objective value $\mathcal{F}_h$ this can be done as follows. First, observe that the first term satisfies

$$I \coloneqq \sum_{i,j=0}^{2^\ell - 1} (\partial w_{x_{ij}}^2 + \partial w_{y_{ij}}^2) = \|\partial W_x\|_\mathrm{F}^2 + \|\partial W_y\|_\mathrm{F}^2, \tag{7.20}$$

where $\partial W_x, \partial W_y \in \mathbb{R}^{n\times n}$ contain the derivatives $\partial w_{x_{ij}}, \partial w_{y_{ij}}$. Then it is easy to verify from (7.18) that

$$\partial W_x = LW_h \quad \text{and} \quad \partial W_y = W_h L^\mathsf{T} \quad \text{with} \quad L = \frac{1}{h}\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}.$$

Substituting this factorization and $W_h = U\Sigma V^\mathsf{T}$ in (7.20), we get

$$I = \|(LU)\Sigma\|_\mathrm{F}^2 + \|(LV)\Sigma\|_\mathrm{F}^2.$$

To recast the second term in (7.19) using matrices, observe that

$$II \coloneqq \sum_{i,j} \gamma_{ij}\, w_{ij} = \sum_{i,j} (\Gamma_h \odot W_h)_{ij} = \operatorname{trace}(\Gamma_h^\mathsf{T} W_h) = \operatorname{trace}\big(\Sigma_\gamma (U_\gamma^\mathsf{T} U)\Sigma (V^\mathsf{T} V_\gamma)\big),$$

where $\odot$ denotes the elementwise (or Hadamard) product of two matrices. Summing the terms I and II, we finally obtain

$$\mathcal{F}_h = \tfrac{h^2}{2}\Big(\|(LU)\Sigma\|_\mathrm{F}^2 + \|(LV)\Sigma\|_\mathrm{F}^2 - 2\operatorname{trace}\big(\Sigma_\gamma (U_\gamma^\mathsf{T} U)\Sigma (V^\mathsf{T} V_\gamma)\big)\Big),$$

which can be evaluated in $O((n + k_\gamma)k_\gamma k)$ flops.

7.4.1.2 Discretization of the gradient

The discretization of (7.17) gives

$$G_h = h^2 \left(AW_h + W_h A - \Gamma_h\right), \tag{7.21}$$

where A is the discretization of $-\Delta$ by a second-order central difference, i.e.,

$$A = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}. \tag{7.22}$$

Observe that $G_h = 0$ in (7.21) — and hence for W_h a critical point of (7.19) — coincides with a solution to the Lyapunov equation $AW_h + W_h A = \Gamma_h$.

Like for the discretized objective function above, we represent the discretized gradient G_h as a factored matrix. Using the same notation as above, this can be done as follows:

$$\begin{aligned} G_h &= h^2 \left(AU\Sigma V^\mathsf{T} + U\Sigma V^\mathsf{T} A - \Gamma_h\right) \\ &= h^2 \left((AU)\Sigma V^\mathsf{T} + U\Sigma (V^\mathsf{T} A) - U_\gamma \Sigma_\gamma V_\gamma^\mathsf{T}\right) \\ &= h^2 \begin{bmatrix} AU & U & U_\gamma \end{bmatrix} \operatorname{blkdiag}(\Sigma, \Sigma, -\Sigma_\gamma) \begin{bmatrix} V & AV & V_\gamma \end{bmatrix}^\mathsf{T}, \end{aligned}$$

where $\operatorname{blkdiag}(\Sigma, \Sigma, -\Sigma_\gamma)$ is the block diagonal matrix created by aligning the matrices Σ, Σ, and $-\Sigma_\gamma$ along the main diagonal. The gradient G_h can be represented in only $O(nk)$ flops for computing AU and AV.

We introduce the notation ξ_h for the Riemannian gradient and recall that it is given by the projection (7.6)

$$\xi_h = \mathrm{P}_{T_{W_h}\mathcal{M}_h^k}(G_h).$$

Its norm $\|\cdot\|_{\mathrm{F}}$ can be directly computed from the format (7.5) as

$$\|\xi_h\|_{\mathrm{F}} = \sqrt{\|M\|_{\mathrm{F}}^2 + \|U_{\mathrm{P}}\|_{\mathrm{F}}^2 + \|V_{\mathrm{P}}\|_{\mathrm{F}}^2}.$$

7.4.1.3 Discretized Hessian

On the coarsest level, where a Riemannian trust-region method is used, we need to provide the directional derivative of the gradient, defined as

$$\operatorname{Hess} f(x)[h] = \mathrm{D}(\operatorname{grad} f)(x)[h],$$

where $x, h \in \mathcal{E}$. Let η denote a tangent vector to $\mathcal{M}_h^k$ at W_h, i.e., $\eta \in T_{W_h}\mathcal{M}_h^k$. We consider η being factorized as $U_\eta \Sigma_\eta V_\eta^{\mathsf{T}}$. Then the directional derivative of the gradient is

$$\operatorname{Hess} \mathcal{F}_h(W_h)[\eta] = h^2(A\eta + \eta A),$$

and its factored form is

$$\operatorname{Hess} \mathcal{F}_h(W_h)[\eta] = h^2 \begin{bmatrix} AU_\eta & U_\eta \end{bmatrix} \operatorname{blkdiag}(\Sigma_\eta, \Sigma_\eta) \begin{bmatrix} V_\eta & AV_\eta \end{bmatrix}^{\mathsf{T}}.$$

7.4.1.4 Numerical results

As mentioned above, the unconstrained minimizer of $\mathcal{F}_h$ over $\mathbb{R}^{n\times n}$ is also a solution of a Lyapunov equation. Restricted to $\mathcal{M}_h^k$ and for small k_γ, this is a typical benchmark problem for low-rank methods; see, e.g., [Sim16, §4.4]. In particular, it guarantees the existence of an approximation of rank

$$k = O\big(\log(1/\varepsilon)\ \log(\kappa(A))\, k_\gamma\big),$$

with error at most ε and $\kappa(A)$ the condition number of A. In the experiments, we have $k_\gamma = 5$ and

$$\gamma(x, y) = e^{x-2y} \sum_{j=1}^{5} 2^{j-1} \sin(j\pi x) \sin(j\pi y). \tag{7.23}$$

We now report on the behavior of RMGLS. In all cases, we used 5 pre- and 5 post-smoothing steps, and coarsest scale $\ell_{\mathrm{c}} = 5$. To monitor the convergence behavior of RMGLS, we have considered three quantities. In all formulas, $\cdot^{(i)}$ indicates that a quantity was evaluated at the ith outer iteration of RMGLS.

(a) The relative error of the discretized objective function $\mathcal{F}_h$:

$$\text{err-}\mathcal{F}(i) \coloneqq |\mathcal{F}_h^{(i)} - \mathcal{F}_h^{(*)}|/|\mathcal{F}_h^{(*)}|.$$

Here, $\mathcal{F}_h^{(*)}$ is the minimal value over $\mathcal{M}_h^k$ of the original objective function in (7.16). It is approximated by minimizing $\mathcal{F}_h$ on $\mathcal{M}_h^k$ with the Riemannian trust-region (RTR) method [AMS08], terminated when the Riemannian gradient norm is smaller than 10^{-13}.

(b) The Frobenius norm of the normalized Riemannian gradient:

$$\text{R-grad}(i) \coloneqq \|\xi_h^{(i)}\|_{\mathrm{F}} / \|\xi_h^{(0)}\|_{\mathrm{F}}.$$

(c) The relative error in Frobenius norm of the low-rank approximation:

$$\text{err-}W(i) \coloneqq \|W_h^{(i)} - W_h^{(*)}\|_{\mathrm{F}} / \|W_h^{(*)}\|_{\mathrm{F}}.$$

Here, $W_h^{(*)}$ is the minimizer of $\mathcal{F}_h$ over $\mathbb{R}^{n\times n}$. It is computed with a Euclidean trust-region method, terminated when the Euclidean gradient norm is smaller than 10^{-14}. No rank truncation was used for $W_h^{(*)}$ and no problem with multiple local minima occurred.[3]

In Figure 7.6, the convergence of the objective function and gradient norm are depicted for RMGLS with finest scale $\ell_{\mathrm{f}} = 8$ and rank $k = 5$. We observe that the objective function has converged already after 25 iterations, whereas the gradient norm continues to decrease until iteration 35. This difference indicates that using a stopping criterion based on the objective function alone can be misleading if we want the most accurate stationary point, and it is better to use a criterion based on the gradient norm.

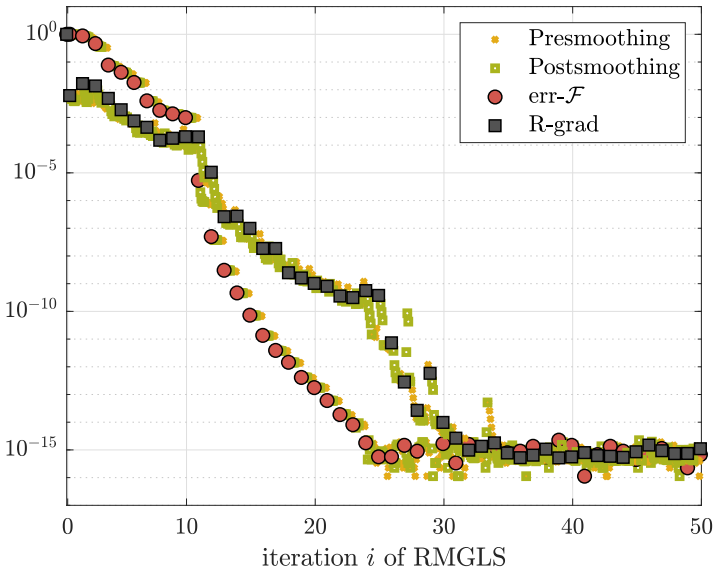
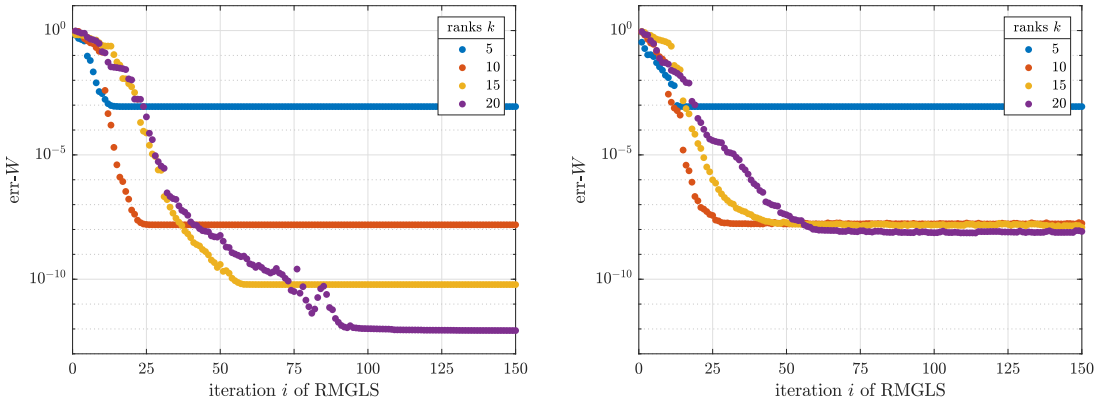


Figure 7.6 – Convergence of err-$\mathcal{F}$ and R-grad for level $\ell_{\mathrm{f}} = 8$ and rank $k = 5$, for the problem of Section 7.4.1.

Figure 7.7 shows the convergence of err-W for increasing ranks k. We compare a line search based on the new Hager–Zhang conditions to the weak Wolfe conditions. The plateaus in both panels are due to the fact that the approximate solution is computed in low-rank format and it is compared to the full-rank reference solution $W_h^{(*)}$. The latter has good low-rank approximations, which is confirmed by the later onset of the stagnation phase when increasing the rank in RMGLS. Panel (b) of Figure 7.7 shows that a line-search procedure with weak Wolfe conditions does not allow us to reduce err-W below $\sqrt{\varepsilon_{\mathrm{mach}}} \approx 10^{-8}$ in double precision arithmetic. This clearly makes the case that the Hager–Zhang line search is useful if we want to obtain more accurate low-rank approximations, as it is visible in panel (a).

[3] For the linear problem, we can of course also directly solve the Lyapunov equation. However, this is not feasible for the nonlinear problem below.



(a) *Hager–Zhang* (b) *weak Wolfe*

Figure 7.7 – Convergence of err-W for level $\ell_\mathrm{f} = 8$, for the problem of Section 7.4.1.

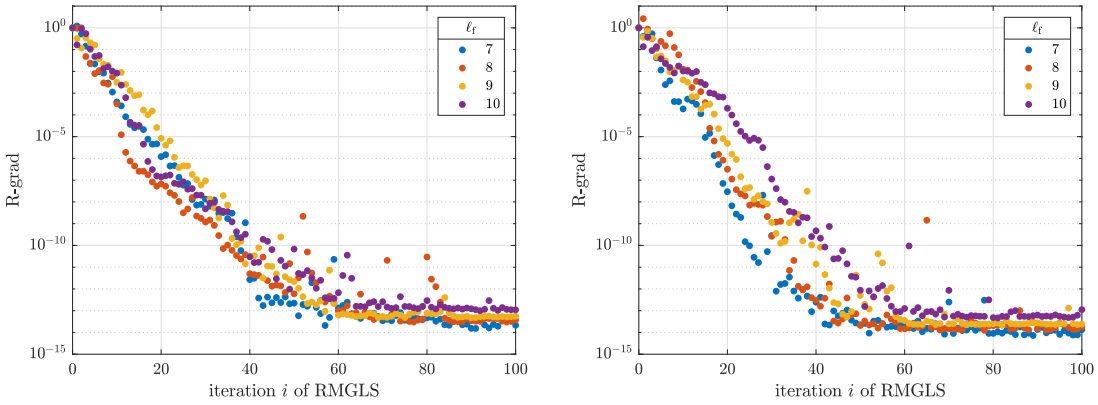
To assess the accuracy of the solutions obtained for the Lyapunov equation, we also use the standard residual

$$r(W_h) \coloneqq \|AW_h + W_hA - \Gamma_h\|_\mathrm{F}.$$

We also consider the following relative residual based on the backward error [Sim07, Eq. (3.6)]

$$r_\mathrm{BW}(W_h) \coloneqq \frac{\|AW_h + W_hA - \Gamma_h\|_\mathrm{F}}{2\|A\|_2\|W_h\|_\mathrm{F} + \|\Gamma_h\|_\mathrm{F}}.$$

Figure 7.8 compares the convergence behavior of R-grad for different fine-scale manifolds with $\ell_\mathrm{f} = 7, 8, 9, 10$. The corresponding sizes of the discretizations are 16 384 (●), 65 536 (●), 262 144 (●) and 1 048 576 (●). Panel (a) corresponds to rank $k = 5$, while panel (b) refers to $k = 10$. One can observe that the convergence behavior is not very dependent on the mesh size, thereby confirming that RMGLS has an almost mesh-independent convergence typical of multigrid methods. In Table 7.1, the final R-grad, err-W, backward error $r_\mathrm{BW}(W_h)$ and residual $r(W_h)$ of the Lyapunov equation are displayed.



(a) *rank* $k = 5$ (b) *rank* $k = 10$

Figure 7.8 – Convergence of R-grad for several finest levels ℓ_f, for the problem of Section 7.4.1.

The numerical experiments presented in this section show that RMGLS, our Riemannian multilevel optimization algorithm with Hager–Zhang line search, converges as we would expect from an effective multigrid method. Satisfying the approximate Wolfe conditions in the

Table 7.1 – Final gradient norm and residuals for the problems of Figure 7.8. The error of the best rank-5 approximation is $\approx 8.73 \times 10^{-4}$. The error of the best rank-10 approximation is $\approx 1.41 \times 10^{-8}$.

	ℓ_f	size	R-grad(100)	$\frac{r_{\mathrm{BW}}(W_h^{(100)})}{r_{\mathrm{BW}}(W_h^{(0)})}$	$r(W_h^{(100)})$	err-W(100)
rank 5	7 (●)	16 384	2.15×10^{-14}	4.91×10^{-5}	1.27×10^{-4}	8.73×10^{-4}
	8 (●)	65 536	3.76×10^{-14}	1.65×10^{-5}	6.34×10^{-5}	8.74×10^{-4}
	9 (●)	262 144	5.55×10^{-14}	5.57×10^{-6}	3.17×10^{-5}	8.75×10^{-4}
	10 (●)	1 048 576	1.10×10^{-13}	1.97×10^{-6}	1.59×10^{-5}	8.75×10^{-4}
rank 10	7 (●)	16 384	1.35×10^{-14}	4.47×10^{-9}	1.63×10^{-8}	1.52×10^{-8}
	8 (●)	65 536	1.83×10^{-14}	1.54×10^{-9}	8.46×10^{-9}	1.54×10^{-8}
	9 (●)	262 144	2.43×10^{-14}	5.25×10^{-10}	4.27×10^{-9}	1.55×10^{-8}
	10 (●)	1 048 576	1.12×10^{-13}	1.82×10^{-10}	2.14×10^{-9}	1.55×10^{-8}

Hager–Zhang line search seems to be sufficient for the method to converge to local minima that are accurate when measured in the relative error and residual norms.

7.4.1.5 Rank adaptivity

In the framework of Riemannian optimization, rank-adaptivity can be introduced by successive runs of increasing rank, using the previous solution as a warm start for the next rank. For recent discussions about this approach see [UV15, KSV16]. An example is given for the problem described in this section, with (7.23) as right-hand side, again with finest level $\ell_\mathrm{f} = 8$ and coarsest level $\ell_\mathrm{c} = 5$, using 5 smoothing steps. Starting from rank $k^{(0)} = 5$, we run RMGLS for 10 iterations, and use the approximate solution to warm start the algorithm with ranks $k^{(i)} = k^{(i-1)} + 5$, $i = 1, \ldots, 4$. Figure 7.9 compares the convergence behavior of this adaptive strategy with the non-adaptive RMGLS, for a target rank $k = 25$. It is apparent that the adaptive RMGLS is more efficient than its non-adaptive counterpart. For example, at the 30th iteration, $r(W_h^{(30)}) \approx 2.50 \times 10^{-4}$ for the non-adaptive RMGLS, whereas it is already $r(W_h^{(30)}) \approx 4.57 \times 10^{-10}$ in the adaptive version.

7.4.2 A nonlinear problem

Next, we consider the variational problem from [WG09, Example 5.1] involving an exponential as nonlinear term:

$$\begin{cases} \min\limits_{w} \mathcal{F}(w) = \displaystyle\int_\Omega \tfrac{1}{2}\|\nabla w\|^2 + \lambda(w-1)\,e^w - \gamma\, w \,\mathrm{d}x\,\mathrm{d}y \\ \quad \text{such that} \quad w = 0 \text{ on } \partial\Omega, \end{cases} \tag{7.24}$$

where $\lambda = 10$, $\Omega = [0,1]^2$, and

$$\gamma(x,y) = ((9\pi^2 + \lambda e^{(x^2-x^3)\sin(3\pi y)})(x^2 - x^3) + 6x - 2)\sin(3\pi y).$$

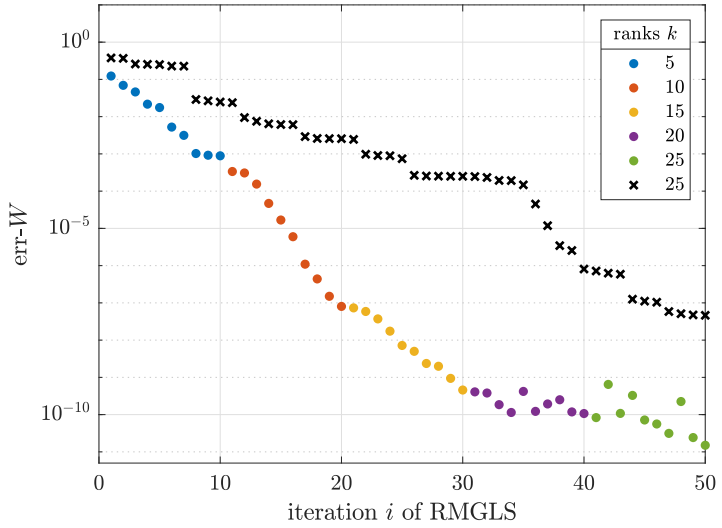


Figure 7.9 – Rank-adaptivity for RMGLS applied to the problem of Section 7.4.1, with (7.23) as right-hand side. Starting from rank 5, the rank is increased by 5 every 10 iterations, until $k = 25$. The black crosses illustrate the behavior of non-adaptive RMGLS with rank $k = 25$.

The variational problem (7.24) corresponds to the nonlinear PDE [Hen03, Eq. (5.4)]

$$\begin{cases} -\Delta w + \lambda w e^w - \gamma = 0 & \text{in } \Omega, \\ w = 0 & \text{on } \partial\Omega. \end{cases}$$

The exact solution $w_{\text{ex}} = (x^2 - x^3)\sin(3\pi y)$ has rank 1, making it less interesting as test case for our low-rank method. In addition, a discretization of the exponential term e^w does not admit a good low-rank approximation for w close to the exact solution.

The following modification,

$$\begin{cases} -\Delta w + \lambda w(w+1) - \gamma = 0 & \text{in } \Omega, \\ w = 0 & \text{on } \partial\Omega, \end{cases} \tag{7.25}$$

is better suited as test case: as we will show below, the nonlinearity $w(w+1)$ can be computed efficiently when w is low rank and the exact solution is full rank but has good low-rank approximations.

To obtain the variational problem corresponding to (7.25), $-\Delta w$ gives rise to the term $\frac{1}{2}\|\nabla w\|^2$ in the integrand of the objective functional, as seen in Section 7.4.1. The term in γ also remains the same. For the nonlinear term in the middle, we calculate the integral of $\lambda w(w+1)$ with respect to w, which gives $\lambda w^2(\frac{1}{3}w + \frac{1}{2})$. Finally, we can formulate the variational problem as

$$\begin{cases} \min\limits_{w} \mathcal{F}(w) = \displaystyle\int_\Omega \tfrac{1}{2}\|\nabla w\|^2 + \lambda w^2(\tfrac{1}{3}w + \tfrac{1}{2}) - \gamma\, w \,\mathrm{d}x\,\mathrm{d}y \\ \quad \text{such that} \quad w = 0 \text{ on } \partial\Omega. \end{cases} \tag{7.26}$$

For γ, we choose the same right-hand side adopted in (7.23).

7.4.2.1 Discretization of the objective function

Discretizing (7.26) similarly as in Section 7.4.1.1, we obtain

$$\mathcal{F}_h = h^2 \sum_{i,j=0}^{2^\ell-1} \left(\tfrac{1}{2}(\partial w_{x_{ij}}^2 + \partial w_{y_{ij}}^2) + \lambda w_{ij}^2 (\tfrac{1}{3} w_{ij} + \tfrac{1}{2}) - \gamma_{ij} w_{ij} \right).$$

The first term and the third term have the same matrix form as the one seen in Section 7.4.1.1. For the second term, we have

$$\sum_{i,j} \frac{\lambda}{2}\, w_{ij}^2 = \frac{\lambda}{2} \operatorname{trace}(W_h^\mathsf{T} W_h) = \frac{\lambda}{2} \|\Sigma\|_\mathrm{F}^2,$$

and

$$\sum_{i,j} \frac{\lambda}{3}\, w_{ij}^3 = \frac{\lambda}{3} \operatorname{trace}(W_h^\mathsf{T} (W_h \odot W_h)). \tag{7.27}$$

For the term $W_h \odot W_h$, we perform the element-wise multiplication in factorized form as explained in [KT14, §7] and store the result in the format $U_\odot \Sigma_\odot V_\odot^\mathsf{T}$:

$$W_h \odot W_h = (U *^\mathsf{T} U)(\Sigma \otimes \Sigma)(V *^\mathsf{T} V)^\mathsf{T} = U_\odot \Sigma_\odot V_\odot^\mathsf{T},$$

where $*^\mathsf{T}$ denotes a transposed variant of the Khatri-Rao product (see definition in [KT14, §7]). Observe that $\operatorname{rank}(W_h \odot W_h) \leqslant k^2$. As a consequence, the term (7.27) becomes

$$\frac{\lambda}{3} \operatorname{trace}(W_h^\mathsf{T} (W_h \odot W_h)) = \frac{\lambda}{3} \operatorname{trace}\big(V(U\Sigma)^\mathsf{T} U_\odot \Sigma_\odot V_\odot^\mathsf{T}\big) = \frac{\lambda}{3} \operatorname{trace}\big(\Sigma (U^\mathsf{T} U_\odot) \Sigma_\odot (V_\odot^\mathsf{T} V)\big).$$

Finally, the discretized functional in matrix form is

$$\begin{aligned} \mathcal{F}_h = \tfrac{h^2}{2} \Big(&\|(LU)\Sigma\|_\mathrm{F}^2 + \|(LV)\Sigma\|_\mathrm{F}^2 + \lambda \|\Sigma\|_\mathrm{F}^2 \\ &+ \tfrac{2}{3}\lambda \operatorname{trace}\big(\Sigma (U^\mathsf{T} U_\odot) \Sigma_\odot (V_\odot^\mathsf{T} V)\big) - 2 \operatorname{trace}\big(\Sigma_\gamma (U_\gamma^\mathsf{T} U) \Sigma (V^\mathsf{T} V_\gamma)\big) \Big), \end{aligned}$$

which can be evaluated in $O\big(nk(k_\gamma + k^2) + k(k_\gamma^2 + k^3)\big)$ flops.

7.4.2.2 Discretization of the gradient

The gradient of $\mathcal{F}$ is the functional derivative

$$\frac{\delta \mathcal{F}}{\delta w} = -\Delta w + \lambda w(w+1) - \gamma.$$

The discretized Euclidean gradient in matrix form is given by

$$G_h = h^2 \left(A W_h + W_h A + \lambda W_h \odot W_h + \lambda W_h - \Gamma_h \right),$$

with A as in (7.22). Substituting the formats $W_h = U\Sigma V^\mathsf{T}$, $W_h \odot W_h = U_\odot \Sigma_\odot V_\odot^\mathsf{T}$, and $\Gamma_h = U_\gamma \Sigma_\gamma V_\gamma^\mathsf{T}$, we get the factorized form

$$G_h = h^2 \begin{bmatrix} (A+\lambda I)U & U & U_\odot & U_\gamma \end{bmatrix} \operatorname{blkdiag}(\Sigma, \Sigma, \lambda\Sigma_\odot, -\Sigma_\gamma) \begin{bmatrix} V & AV & V_\odot & V_\gamma \end{bmatrix}^\mathsf{T}.$$

The gradient G_h can be represented in only $O(nk)$ flops for computing $(A+\lambda I)U$ and AV.

7.4.2.3 Discretized Hessian

The directional derivative of the gradient is

$$\operatorname{Hess} \mathcal{F}_h(W_h)[\eta] = h^2(A\eta + \eta A + 2\lambda W_h \odot \eta + \lambda\eta).$$

Let the Hadamard product $W_h \odot \eta$ in factorized form be $W_h \odot \eta = U_\odot \Sigma_\odot V_\odot^{\mathsf{T}}$. Then the factored form of the directional derivative of the gradient is

$$h^2 \begin{bmatrix} AU_\eta & U_\eta & U_\odot & U_\eta \end{bmatrix} \operatorname{blkdiag}\left(\Sigma_\eta, \Sigma_\eta, 2\lambda\Sigma_\odot, \lambda\Sigma_\eta\right) \begin{bmatrix} V_\eta & AV_\eta & V_\odot & V_\eta \end{bmatrix}^{\mathsf{T}}.$$

7.4.2.4 Numerical results

We repeat the same set of experiments as for the previous problem to verify the convergence of the error and residual functions defined in Section 7.4.1.4. The coarse level was again taken as $\ell_c = 5$. Comparing Figures 7.10, 7.11, 7.12, and 7.13 for this nonlinear problem to the ones of the linear problem, we observe that the earlier conclusions remain virtually the same.

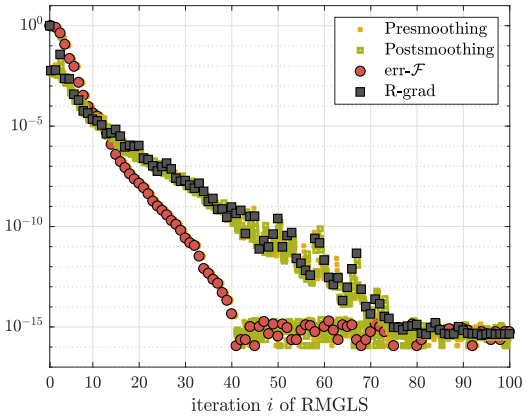


Figure 7.10 – Convergence of err-$\mathcal{F}$ and R-grad for level $\ell_f = 8$ and rank $k = 5$, for the problem of Section 7.4.2.

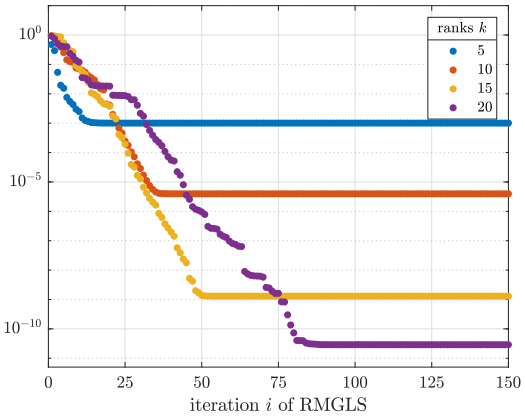


Figure 7.11 – Convergence of err-W, with Hager–Zhang line search, for $\ell_f = 8$ and the rank values $k = 5, 10, 15, 20$.

7.5 Comparison with other methods

We compare Euclidean trust regions (ETR), Euclidean multilevel optimization (EML), EML with low rank via truncated SVD, Riemannian trust regions (RTR) with fixed rank, and our RMGLS with fixed rank. ETR and EML do not use any low-rank approximation, whereas the other methods do.

All methods were implemented in MATLAB. ETR and RTR were executed using solvers from the Manopt package [BMAS14] with the Riemannian embedded submanifold geometry from [Van13] for RTR. EML was implemented by ourselves based on the same multigrid components as RMGLS, as already explained in Section 7.4. EML with truncated SVD applies truncation via the SVD with a fixed rank after every computational step in EML.

Table 7.2 summarizes the results for the linear problem described in Section 7.4.1. It is apparent that the Euclidean algorithms soon become very inefficient as the problem size grows. Hence we omit results for bigger problem sizes. For the smaller problems, the residual of the final approximation was always very small since there was no rank truncation.

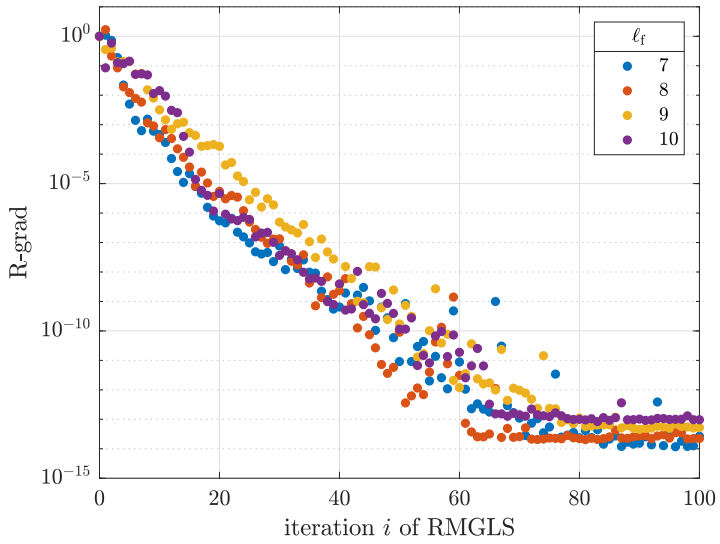


Figure 7.12 – Convergence of R-grad for several finest levels ℓ_f and rank $k = 5$, for the problem of Section 7.4.2.

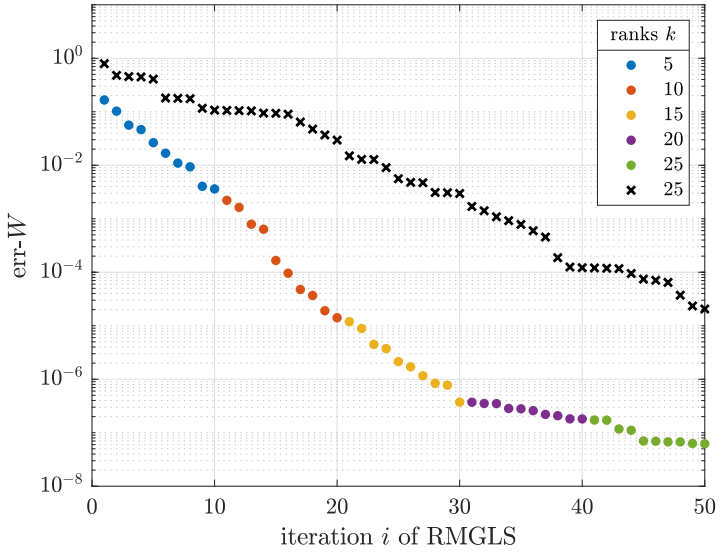


Figure 7.13 – Rank-adaptivity for RMGLS applied to the problem of Section 7.4.2, with (7.23) as right-hand side. Starting from rank 5, the rank is increased by 5 every 10 iterations, until $k = 25$. The black crosses illustrate the behavior of non-adaptive RMGLS with rank $k = 25$.

In the low-rank version of EML with truncated SVD, the algorithm was stopped *before* stagnation in the residual norm started to occur, as determined by manual inspection. This was done so that the algorithm certainly did not run longer than needed.[4] All the other low-rank algorithms were stopped when the norm of the Riemannian gradient was below the threshold value of 10^{-12}.

Observe that the accuracy achieved by EML with truncated SVD is not as good compared to RTR and RMGLS. This was *not* due to our stopping condition but probably because of the fixed-rank truncations throughout the multigrid cycle in EML. It is possible that a more careful choice of ranks can improve on the accuracy, but we did not investigate this issue since RTR and RMGLS are also using fixed-rank truncations.

The Riemannian algorithms on the manifold of fixed-rank matrices show a more efficient behavior. For problems having a relatively small size ($\ell_{\mathrm{f}} = 10,\ 11$), RTR is more efficient than RMGLS, while for bigger problems, RMGLS is much more efficient that RTR. The fastest computational time for a given level is highlighted in bold text. In particular, for $\ell_{\mathrm{f}} = 14$, our RMGLS is almost 6 times more efficient than the RTR. This demonstrates that for very big problem sizes the Riemannian multilevel strategy is the most advantageous.

An important observation is that the Riemannian algorithms can be terminated based on the Riemannian gradient, since it provably can be made very small, as it is clear from the tables and also from the figures in the previous section. This property allows us to stop the algorithm when the local gradient is smaller than a certain threshold. On the contrary, the EML low-rank algorithm does not have this property and, since the (Riemannian) gradient might never become very small, the stopping criterion has to be based on stagnation detection.

Another observation concerns the "multiplying factor" across the levels for different methods. We are mostly interested in comparing the scaling factors for RTR and RMGLS when enlarging the level ℓ_{f}, since the other methods are visibly more expensive than these two. From Table 7.2, we obtain on average scaling factors of 3.5 for RTR and 1.7 for RMGLS, respectively. Finally, Table 7.3 shows that if we increase the rank, it is possible to achieve better accuracy in the residuals $r(W_h^{(\mathrm{end})})$ for both EML with rank truncation and RMGLS. In addition, RMGLS is considerably faster than EML for the same rank and for the biggest problem. Table 7.4 summarizes the results for the nonlinear problem described in Section 7.4.2. Similar considerations as above can be done for this problem. We point out that the higher computational times in the low-rank algorithms are due to the calculations of the Hadamard products in factored form. From Table 7.4 we can obtain the following average multiplying factors across the levels: 4.3 for RTR, 2.0 for RMGLS. These are in good agreement with the ones computed for the linear problem.

7.6 Conclusions

In this chapter, we have shown how to combine multilevel optimization with optimization on low-rank manifolds. Compared to other approaches, no explicit preconditioning needs to be performed to solve an ill-conditioned Newton equation. Numerical experiments demonstrated that for two variational problems our method succeeds in computing good low-rank approximations with an almost mesh-independent convergence behavior. In addition, we discussed how to apply the accurate Riemannian Hager–Zhang line search presented in Chapter 5 to the manifold of fixed-rank matrices.

[4] Although in practice such a stopping condition can not be implemented.

Table 7.2 – Comparisons of different methods for the problem described in Section 7.4.1. The — means the Riemannian gradient $\xi_h^{(\mathrm{end})}$ does not apply.

ℓ_f	size	time (s)	$\|\xi_h^{(\mathrm{end})}\|_F$	$r(W_h^{(\mathrm{end})})$
ETR (no rank truncation, no multilevel)				
9	262 144	19	—	9.2451×10^{-15}
10	1 048 576	164	—	5.2284×10^{-15}
11	4 194 304	1 787	—	1.0223×10^{-14}
EML (8 smoothing steps, $\ell_\mathrm{c} = 7$)				
no rank truncation				
9	262 144	16	—	6.2645×10^{-13}
10	1 048 576	77	—	3.4368×10^{-13}
11	4 194 304	459	—	4.2710×10^{-12}
truncation to rank 5				
9	262 144	9	—	4.5166×10^{-5}
10	1 048 576	35	—	2.2084×10^{-5}
11	4 194 304	58	—	1.7780×10^{-4}
RTR – rank 5 (no multilevel)				
10	1 048 576	**6**	1.5002×10^{-14}	1.5873×10^{-5}
11	4 194 304	**20**	2.7687×10^{-14}	7.9369×10^{-6}
12	16 777 216	66	6.6810×10^{-14}	3.9685×10^{-6}
13	67 108 864	237	1.1654×10^{-13}	1.9842×10^{-6}
14	268 435 456	929	2.6852×10^{-13}	9.9212×10^{-7}
RMGLS – rank 5 (8 smoothing steps, $\ell_\mathrm{c} = 7$)				
10	1 048 576	18	6.1634×10^{-13}	1.5873×10^{-5}
11	4 194 304	26	2.5091×10^{-13}	7.9369×10^{-6}
12	16 777 216	**52**	6.5807×10^{-13}	3.9685×10^{-6}
13	67 108 864	**94**	9.2574×10^{-13}	1.9842×10^{-6}
14	268 435 456	**161**	6.1323×10^{-13}	9.9212×10^{-7}

Table 7.3 – Comparisons of EML with rank truncation and RMGLS for different ranks applied to the problem described in Section 7.4.1. In both cases, 8 smoothing steps and coarsest level $\ell_c = 7$ are used.

			EML		RMGLS		
	ℓ_f	size	time (s)	$r(W_h^{(\text{end})})$	time (s)	$\|\xi_h^{(\text{end})}\|_F$	$r(W_h^{(\text{end})})$
rank 10	9	262 144	**13**	9.1637×10^{-9}	18	7.6740×10^{-13}	4.2704×10^{-9}
	10	1 048 576	55	3.3303×10^{-9}	**31**	5.0568×10^{-13}	2.1431×10^{-9}
	11	4 194 304	451	4.9866×10^{-5}	**76**	3.1722×10^{-13}	1.0704×10^{-9}
rank 15	9	262 144	43	4.7685×10^{-11}	**42**	6.1597×10^{-13}	4.2953×10^{-11}
	10	1 048 576	107	2.4681×10^{-11}	**103**	6.1486×10^{-13}	2.1541×10^{-11}
	11	4 194 304	495	4.5356×10^{-11}	**174**	8.9919×10^{-13}	1.0940×10^{-11}

Table 7.4 – Comparisons of different methods for the problem described in Section 7.4.2. The — means the Riemannian gradient $\xi_h^{(\mathrm{end})}$ does not apply.

ℓ_f	size	time (s)	$\|\xi_h^{(\mathrm{end})}\|_F$	$r(W_h^{(\mathrm{end})})$
ETR: no rank truncation, no multilevel				
9	262 144	23	—	8.8890×10^{-15}
10	1 048 576	196	—	6.0243×10^{-15}
11	4 194 304	1 990	—	1.1352×10^{-14}
EML: $\ell_\mathrm{c} = 7$, 8 smoothing steps				
no rank truncation				
9	262 144	22	—	2.6912×10^{-13}
10	1 048 576	75	—	9.4184×10^{-13}
11	4 194 304	506	—	4.2292×10^{-13}
truncation to rank 5				
9	262 144	11	—	4.4994×10^{-5}
10	1 048 576	47	—	5.0028×10^{-5}
11	4 194 304	72	—	1.6216×10^{-4}
RTR: rank 5, no multilevel				
10	1 048 576	**11**	1.3449×10^{-14}	1.5614×10^{-5}
11	4 194 304	**31**	9.3240×10^{-14}	7.8072×10^{-6}
12	16 777 216	151	5.9424×10^{-14}	3.9036×10^{-6}
13	67 108 864	554	1.1696×10^{-13}	1.9518×10^{-6}
14	268 435 456	3 338	2.1950×10^{-13}	9.7591×10^{-7}
RMGLS: rank 5, $\ell_\mathrm{c} = 7$, 8 smoothing steps				
10	1 048 576	44	5.3255×10^{-13}	1.5614×10^{-5}
11	4 194 304	51	4.0362×10^{-13}	7.8072×10^{-6}
12	16 777 216	**120**	9.6698×10^{-13}	3.9036×10^{-6}
13	67 108 864	**209**	3.8296×10^{-13}	1.9518×10^{-6}
14	268 435 456	**549**	9.8448×10^{-13}	9.7591×10^{-7}

Table 7.5 – Comparisons of EML with rank truncation and RMGLS for different ranks applied to the problem described in Section 7.4.2. In both cases, 8 smoothing steps and coarsest level $\ell_c = 7$ are used.

			EML		RMGLS		
	ℓ_f	size	time (s)	$r(W_h^{(\text{end})})$	time (s)	$\|\xi_h^{(\text{end})}\|_F$	$r(W_h^{(\text{end})})$
rank 10	9	262 144	30	4.7324×10^{-7}	**21**	7.8437×10^{-13}	3.7321×10^{-7}
	10	1 048 576	123	3.4975×10^{-7}	**61**	4.0398×10^{-13}	1.8660×10^{-7}
	11	4 194 304	797	1.2826×10^{-5}	**153**	5.5800×10^{-13}	9.3301×10^{-8}
rank 15	9	262 144	107	7.4928×10^{-10}	**92**	2.0183×10^{-13}	4.2886×10^{-10}
	10	1 048 576	380	9.6225×10^{-10}	**207**	6.5306×10^{-13}	2.6044×10^{-10}
	11	4 194 304	3 113	4.3682×10^{-10}	**532**	1.3610×10^{-13}	8.3563×10^{-11}

Appendix A

Single shooting

A.1 Freedom in choosing the geodesic

As mentioned in Remark 2.2, the matrix $Y_{0\perp}$ does not need to be orthonormal; in fact, its only requirement is that it has to span $\mathcal{Y}_0^\perp = (\mathrm{span}(Y_0))^\perp$, the orthogonal subspace to $\mathcal{Y}_0 = \mathrm{span}(Y_0)$. In this appendix, we are going to show this, starting from the geodesic

$$Y(t) = \begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -K^\intercal \\ K & O_{n-p} \end{bmatrix} t \right) \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}.$$

Let M be any $(n-p)$-by-$(n-p)$ invertible matrix, and define

$$\widetilde{M} = \begin{bmatrix} I_p & \\ & M \end{bmatrix}, \qquad \widetilde{M}^{-1} = \begin{bmatrix} I_p & \\ & M^{-1} \end{bmatrix}, \qquad \widetilde{M}^{-1}\widetilde{M} = \widetilde{M}\widetilde{M}^{-1} = I_n.$$

Observe that

$$\begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix} \widetilde{M}^{-1} = \begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix} \begin{bmatrix} I_p & \\ & M^{-1} \end{bmatrix} = \begin{bmatrix} Y_0 & Y_{0\perp} M^{-1} \end{bmatrix},$$

and

$$\widetilde{M} \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix} = \begin{bmatrix} I_p & \\ & M \end{bmatrix} \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix} = \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}.$$

In the following steps, we use these facts together with the property $\widetilde{M}\exp(A)\widetilde{M}^{-1} = \exp(\widetilde{M}A\widetilde{M}^{-1})$, which holds for any invertible matrix $\widetilde{M}$.

$$\begin{aligned} Y(t) &= \begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix} \widetilde{M}^{-1}\widetilde{M} \exp\left(\begin{bmatrix} \Omega & -K^\intercal \\ K & O_{n-p} \end{bmatrix} t \right) \widetilde{M}^{-1}\widetilde{M} \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}, \\ &= \begin{bmatrix} Y_0 & Y_{0\perp} M^{-1} \end{bmatrix} \exp\left(\widetilde{M} \begin{bmatrix} \Omega & -K^\intercal \\ K & O_{n-p} \end{bmatrix} \widetilde{M}^{-1} t \right) \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}, \\ &= \begin{bmatrix} Y_0 & Y_{0\perp} M^{-1} \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -K^\intercal M^{-1} \\ MK & O_{n-p} \end{bmatrix} t \right) \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}. \end{aligned}$$

Since the matrix M is invertible, it can be regarded as a change of basis. Hence, it appears from the last expression that there is freedom in choosing $Y_{0\perp}$, since it can be any matrix whose columns form a basis for $\mathcal{Y}_0^\perp$.

A.2 Smaller formulation

In this section, we prove that, when $p \leqslant \frac{n}{2}$, the geodesic problem on $\mathrm{St}(n,p)$ can be reformulated into an equivalent problem on $\mathrm{St}(2p,p)$. We start from (2.2) with $t=1$, namely,

$$Y_1 = \begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -K^\mathsf{T} \\ K & O_{n-p} \end{bmatrix} \right) \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}.$$

Consider the QR decomposition of K

$$K = \begin{bmatrix} Q & Q_\perp \end{bmatrix} \begin{bmatrix} R \\ O_{(n-2p)\times p} \end{bmatrix} = QR,$$

where $\begin{bmatrix} Q & Q_\perp \end{bmatrix} \in \mathbb{R}^{(n-p)\times(n-p)}$ is the orthogonal factor of K, whose blocks $Q \in \mathbb{R}^{(n-p)\times p}$, $Q_\perp \in \mathbb{R}^{(n-p)\times(n-2p)}$ are orthonormal, and $R \in \mathbb{R}^{p\times p}$ is upper triangular. Inserting this decomposition in the matrix

$$\begin{bmatrix} \Omega & -K^\mathsf{T} \\ K & O_{n-p} \end{bmatrix},$$

we get

$$\begin{bmatrix} \Omega & \begin{bmatrix} -R^\mathsf{T} & O \end{bmatrix} \begin{bmatrix} Q & Q_\perp \end{bmatrix}^\mathsf{T} \\ \begin{bmatrix} Q & Q_\perp \end{bmatrix} \begin{bmatrix} R \\ O \end{bmatrix} & O_{n-p} \end{bmatrix} = \begin{bmatrix} I_p & \\ & \begin{bmatrix} Q & Q_\perp \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Omega & -R^\mathsf{T} & \\ R & O_p & \\ & & O_{n-2p} \end{bmatrix} \begin{bmatrix} I_p & \\ & \begin{bmatrix} Q & Q_\perp \end{bmatrix}^\mathsf{T} \end{bmatrix}.$$

Substituting this expression into the argument of the matrix exponential, and using the property $\exp(\widetilde{Q} M \widetilde{Q}^\mathsf{T}) = \widetilde{Q} \exp(M) \widetilde{Q}^\mathsf{T}$ for any orthogonal matrix $\widetilde{Q}$, we get

$$Y_1 = \begin{bmatrix} Y_0 & Y_{0\perp} \end{bmatrix} \begin{bmatrix} I_p & \\ & \begin{bmatrix} Q & Q_\perp \end{bmatrix} \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -R^\mathsf{T} & \\ R & O_p & \\ & & O_{n-2p} \end{bmatrix} \right) \begin{bmatrix} I_p & \\ & \begin{bmatrix} Q & Q_\perp \end{bmatrix}^\mathsf{T} \end{bmatrix} \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix}.$$

Using the fact that the argument of $\exp$ is a block diagonal matrix, we can write

$$Y_1 = \begin{bmatrix} Y_0 & Y_{0\perp}Q & Y_{0\perp}Q_\perp \end{bmatrix} \begin{bmatrix} \exp\left(\begin{bmatrix} \Omega & -R^\mathsf{T} \\ R & O_p \end{bmatrix} \right) & \\ & I_{(n-2p)} \end{bmatrix} \begin{bmatrix} I_p \\ O_p \\ O_{(n-2p)\times p} \end{bmatrix}.$$

We collect the matrices in order to make the products conformable

$$Y_1 = \begin{bmatrix} \underbrace{\begin{bmatrix} Y_0 & Y_{0\perp}Q \end{bmatrix}}_{\in\mathbb{R}^{n\times 2p}} & \underbrace{Y_{0\perp}Q_\perp}_{\in\mathbb{R}^{n\times(n-2p)}} \end{bmatrix} \begin{bmatrix} \exp\left(\begin{bmatrix} \Omega & -R^\mathsf{T} \\ R & O_p \end{bmatrix} \right) & \\ & I_{(n-2p)} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} I_p \\ O_p \end{bmatrix} \\ O_{(n-2p)\times p} \end{bmatrix}.$$

Finally we have obtained the smaller formulation (2.10)

$$Y_1 = \begin{bmatrix} Y_0 & Y_{0\perp}Q \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -R^\mathsf{T} \\ R & O_p \end{bmatrix} \right) \begin{bmatrix} I_p \\ O_p \end{bmatrix}.$$

Appendix **B**

Fréchet derivatives

B.1 First-order Fréchet derivative of a matrix function

The Fréchet derivative of a matrix function $f\colon \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at $X \in \mathbb{C}^{n\times n}$ is the unique linear function $\mathrm{D}f(X)[\cdot]$ of the matrix $E \in \mathbb{C}^{n\times n}$, that satisfies

$$f(X+E) - f(X) - \mathrm{D}f(X)[E] = o(\|E\|). \tag{B.1}$$

The mapping itself is denoted by either $\mathrm{D}f(X)[\cdot]$ or $\mathrm{D}f(X)$, while the value of the mapping for direction E (i.e. the directional derivative) is denoted by $\mathrm{D}f(X)[E]$.

Since $\mathrm{D}f(X)\colon \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is a linear operator, one can write

$$\mathrm{vec}\big(\mathrm{D}f(X)[E]\big) = J_f^X \,\mathrm{vec}(E) \tag{B.2}$$

for some $n^2 \times n^2$ complex matrix J_f^X independent of E. We refer to J_f^X as the Kronecker representation of the Fréchet derivative, or simply as the Jacobian matrix.

B.2 Singular values of J_f^X

In this section, we report some results that are used in the analysis of the single shooting method on the Stiefel manifold (see Section B.2.1). The operator norm of $\mathrm{D}f(X)$ for the Frobenius norm is defined by

$$\|\mathrm{D}f(X)\|_{\mathrm{F}} = \max_{Z\neq 0} \frac{\|\mathrm{D}f(X)[Z]\|_{\mathrm{F}}}{\|Z\|_{\mathrm{F}}} = \max_{\|Z\|_{\mathrm{F}}=1} \|\mathrm{D}f(X)[Z]\|_{\mathrm{F}}.$$

By vectorizing $\mathrm{D}f(X)[Z]$ as in (B.2), and using the fact that the Euclidean norm of $z = \mathrm{vec}(Z)$ equals the Frobenius norm of Z, we can also write

$$\|\mathrm{D}f(X)\|_{\mathrm{F}} = \max_{\|z\|_2=1} \|J_f^X z\|_2 = \|J_f^X\|_2 = \sigma_{\max}(J_f^X),$$

where $\sigma_{\max}(J_f^X)$ is the largest singular value of J_f^X.

We have the following important theorem.

Theorem B.1 ([Hig08, Cor. 3.16]). *If $X \in \mathbb{C}^{n\times n}$ is normal, then*

$$\sigma_{\max}(J_f^X) = \max_{\lambda,\mu\in\Lambda(X)} \big|f[\lambda,\mu]\big|, \tag{B.3}$$

where $\Lambda(X)$ denote the eigenvalues of X, and $f[\lambda, \mu]$ is the first-order divided difference defined by

$$f[\lambda, \mu] = \begin{cases} \frac{f(\lambda)-f(\mu)}{\lambda-\mu}, & \lambda \neq \mu, \\ f'(\lambda), & \lambda = \mu. \end{cases} \tag{B.4}$$

If $\mathrm{D}f(X)$ is invertible, we have a similar property for the minimal singular value:

Theorem B.2. *If $X \in \mathbb{C}^{n\times n}$ is normal, then*

$$\sigma_{\min}(J_f^X) = \min_{\lambda,\mu\in\Lambda(X)} |f[\lambda, \mu]|.$$

Proof. We adjust the proof of [Hig08, Cor. 3.16] accordingly. We start from the variational property [GVL13, Theorem 8.6.1]

$$\sigma_{\min}(J_f^X) = \min_{\|E\|_{\mathrm{F}}=1} \|\,\mathrm{D}f(X)[E]\|_{\mathrm{F}},$$

and we use $\mathrm{D}f(X)[E] = Z(\,\mathrm{D}f(D)[\widetilde{E}])Z^{-1}$, with $D = \mathrm{diag}(\lambda_i)$ and $\widetilde{E} = Z^{-1}EZ$, as in [Hig08, Cor. 3.12]. Then

$$\sigma_{\min}(J_f^X) = \min_{\|\widetilde{E}\|_{\mathrm{F}}=1} \|Z(\,\mathrm{D}f(D)[\widetilde{E}])Z^{-1}\|_{\mathrm{F}} = \min_{\|\widetilde{E}\|_{\mathrm{F}}=1} \|\,\mathrm{D}f(D)[\widetilde{E}]\|_{\mathrm{F}} = \min_{i,j} |f[\lambda_i, \lambda_j]|,$$

where for the last equality we used the same reasoning as in the proof of [Hig08, Cor. 3.13]. □

B.2.1 Analysis of $J^A_{\exp(A)}$

As we did in Section 2.3.1, let us denote

$$A = \begin{bmatrix} \Omega & -K^{\mathsf{T}} \\ K & O_{n-p} \end{bmatrix}$$

the matrix in the argument of the exponential appearing in the geodesic (2.2), with $\Omega \in \mathcal{S}_{\mathrm{skew}}(p)$ and $K \in \mathbb{R}^{(n-p)\times p}$, and let the Jacobian of $\exp(A)$ with respect to A be [Hig08]

$$J^A_{\exp(A)} = \left(\exp(A^{\mathsf{T}}/2) \otimes \exp(A/2)\right) \mathrm{sinch}\left(\tfrac{1}{2}[A^{\mathsf{T}} \oplus (-A)]\right).$$

Since A is normal, we can apply the theorems presented above to bound the singular values of the Jacobian $J^A_{\exp(A)}$ of the matrix exponential of A.

Theorem B.3. *Let A and $J^A_{\exp(A)}$ be as defined above, and let $\alpha = \|A\|_2$. We have*

$$\sigma_{\max}(J^A_{\exp(A)}) = 1 \qquad \textit{and} \qquad \sigma_{\min}(J^A_{\exp(A)}) = |\mathrm{sinc}\,\alpha|\,.$$

Proof. Since A is a real skew-symmetric matrix, the eigenvalues of A are purely imaginary. Hence we may denote them as $\mathrm{i}x$ and $\mathrm{i}y$, with $x, y \in \mathbb{R}$. Let us rewrite (B.3) as

$$\|J^A_{\exp(A)}\|_2 = \sigma_{\max}(J^A_{\exp(A)}) = \max_{|x|,|y|\leqslant\alpha} |\exp[\mathrm{i}x, \mathrm{i}y]|,$$

where $|x|, |y| \leqslant \alpha$ because the absolute value of an eigenvalue of a normal matrix cannot exceed any norm of that matrix. The maximum is attained for $y = x$, and using the definition in (B.4), we get

$$\sigma_{\max}(J^A_{\exp(A)}) = \max_{|x|\leqslant\alpha} |\exp[\mathrm{i}x, \mathrm{i}x]| = \max_{|x|\leqslant\alpha} |\exp'(\mathrm{i}x)| = \max_{|x|\leqslant\alpha} |\exp(\mathrm{i}x)| = 1.$$

This shows that the maximum singular value of $J^A_{\exp(A)}$ is always 1.

For the minimum singular value, let us specialize Theorem B.2 to our case:

$$\sigma_{\min}\left(J^A_{\exp(A)}\right) = \min_{|x|,|y|\leqslant\alpha} \left|\exp[\mathrm{i}x, \mathrm{i}y]\right| = \min_{|x|,|y|\leqslant\alpha} \underbrace{\left|\frac{e^{\mathrm{i}x} - e^{\mathrm{i}y}}{\mathrm{i}x - \mathrm{i}y}\right|}_{=:g(x,y)}.$$

The minima of $g(x, y)$ are attained on the anti-diagonal at the corners, namely, when $x = \alpha, y = -\alpha$ and $x = -\alpha, y = \alpha$. This gives:

$$\sigma_{\min}\left(J^A_{\exp(A)}\right) = \left|\frac{e^{\mathrm{i}\alpha} - e^{-\mathrm{i}\alpha}}{2\,\mathrm{i}\alpha}\right| = \left|\frac{\sin\alpha}{\alpha}\right| = \left|\operatorname{sinc}\alpha\right|.$$

□

Figure B.1 illustrates the function $|\operatorname{sinc}\alpha|$ for α in the interval $[0, 5]$.

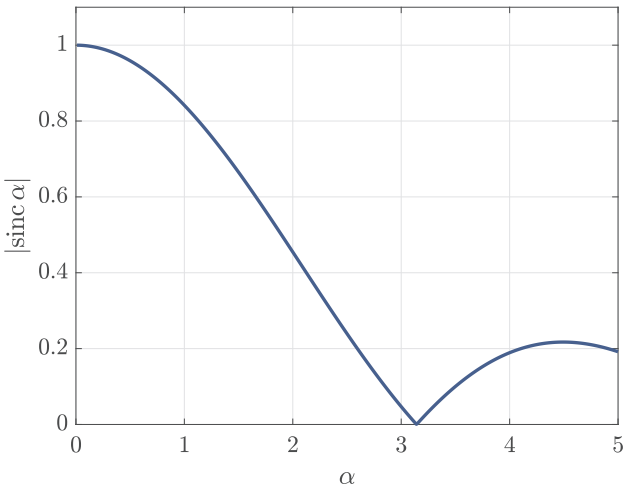


Figure B.1 – A plot of $|\operatorname{sinc}\alpha|$ for α in the interval $[0, 5]$.

Observe that for $\alpha = \pi$ the sinc function is equal to zero, hence $J^A_{\exp(A)}$ becomes singular. Since $J^A_{\exp(A)}$ appears in the expression for the Jacobian of the geodesic, Equation (2.9),

$$J^x_{Z_1} = \left(\left[I_p \;\; O_{p\times(n-p)}\right] \otimes Q\right) J^A_{\exp(A)}\, T\, J^x_{A(x)},$$

the above result is related to the cases in which single shooting fails.

Appendix C

Jacobians for multiple shooting

In this appendix, we report the explicit formulas for the Jacobian matrices that are used in the multiple shooting method on the Stiefel manifold $\mathrm{St}(n,p)$ (see Section 2.4).

Let Σ_1 denote a base point and Σ_2 its corresponding tangent vector as explained in Section 2.4 and illustrated in Figure 2.4.

To compute the Jacobian matrices appearing in (2.16), we formulate the geodesic equation (2.2) using the singular value decomposition of the base point Σ_1, namely, $\Sigma_1 = USV^\mathsf{T}$. Let us consider the partitioned matrices (MATLAB notation)

$$U_p = U(:,1\!:\!p), \quad U_\perp = U(:,p+1\!:\!\mathrm{end}), \quad V_p = V(:,1\!:\!p), \quad V_\perp = V(:,p+1\!:\!\mathrm{end}),$$

and let $\widetilde{Q} = \begin{bmatrix}\Sigma_1 & U_\perp\end{bmatrix}$. Then the SVD formulations of the geodesic and its derivative are

$$Z_1(t) = \widetilde{Q}\,\exp(tA)\begin{bmatrix} I_p \\ O_{(n-p)\times p}\end{bmatrix}, \qquad Z_2(t) = \widetilde{Q}\,\exp(tA)\,A\begin{bmatrix} I_p \\ O_{(n-p)\times p}\end{bmatrix},$$

where

$$A(\widetilde{Q},\Sigma_2) = \begin{bmatrix} [I_p \;\; O]\widetilde{Q}^\mathsf{T}\Sigma_2 & -\big[[O \;\; I_{n-p}]\widetilde{Q}^\mathsf{T}\Sigma_2\big]^\mathsf{T} \\ [O \;\; I_{n-p}]\widetilde{Q}^\mathsf{T}\Sigma_2 & O_{n-p}\end{bmatrix} = \begin{bmatrix}\Sigma_1^\mathsf{T}\Sigma_2 & -(U_\perp^\mathsf{T}\Sigma_2)^\mathsf{T} \\ U_\perp^\mathsf{T}\Sigma_2 & O_{n-p}\end{bmatrix}.$$

C.1 Jacobians with respect to the base point

Let us first compute the Jacobians of the geodesic and its derivative with respect to the base point Σ_1, i.e.,

$$J_{Z_1}^{\Sigma_1} = \frac{\partial Z_1}{\partial \Sigma_1} \quad \text{and} \quad J_{Z_2}^{\Sigma_1} = \frac{\partial Z_2}{\partial \Sigma_1}.$$

We adopt for the functions involved the notation:

- $s(\Sigma_1) = \texttt{svd}(\Sigma_1)$, performs the SVD of Σ_1 and returns U_p, $U_\perp$, V_p, $V_\perp$;
- $\tilde{q}(s(\Sigma_1)) = \begin{bmatrix}\Sigma_1 & U_\perp\end{bmatrix} = \widetilde{Q}$, builds the matrix $\widetilde{Q}$ from Σ_1 and $U_\perp$;
- $h(\tilde{q}(s(\Sigma_1))) = A$, builds the matrix argument of $\exp$;
- $g(h(\widetilde{Q})) = \exp(A)$, performs the matrix exponential of A.

To compute $\frac{\partial Z_1}{\partial \Sigma_1}$ we have to consider the derivative of a product and the chain rule for a composite function:

$$\begin{aligned}\mathrm{D}Z_1(\Sigma_1)[E] = \mathrm{D}\tilde{q}(s(\Sigma_1),\, \mathrm{D}s(\Sigma_1)[E])\, \exp(A) \begin{bmatrix} I_p \\ O \end{bmatrix} \\ + \widetilde{Q}\, \mathrm{D}g\Big(h(\tilde{q}(s(\Sigma_1))),\, \mathrm{D}h\big[\tilde{q}(s(\Sigma_1)),\, \mathrm{D}\tilde{q}\,(s(\Sigma_1),\, \mathrm{D}s(\Sigma_1)[E])\,\big]\Big) \begin{bmatrix} I_p \\ O \end{bmatrix}.\end{aligned}$$

As in Appendix B, $\mathrm{D}f(A)[E]$ denotes the Fréchet derivative of f at the matrix A in the direction of E. Vectorizing the last expression we get

$$\mathrm{vec}(\mathrm{D}Z_1(\Sigma_1)[E]) = \Big(\big[I_p \;\; O\big] \exp(A)^\mathsf{T} \otimes I_n\Big)\, \mathrm{vec}(\mathrm{D}\tilde{q}) + \Big(\big[I_p \;\; O\big] \otimes \widetilde{Q}\Big)\, \mathrm{vec}(\mathrm{D}g). \quad \text{(C.1)}$$

Here,

$$\mathrm{vec}(\mathrm{D}g(A)[E]) = J^A_{\exp(A)}\, \mathrm{vec}(\mathrm{D}h),$$

with $J^A_{\exp(A)}$ the Jacobian of $\exp$ with respect to its argument. As we did for single shooting (see Section 2.3.1), we introduce a linear map T that maps a block-wise vectorization into the ordinary column-stacking vectorization. This is achieved by:

$$\mathrm{vec}(\mathrm{D}h) = T \cdot \mathrm{blkvec}(\mathrm{D}h),$$

where

$$\mathrm{blkvec}(\mathrm{D}h) = \begin{bmatrix} \mathrm{vec}\big(\big[I_p \;\; O\big]\, \mathrm{D}\tilde{q}^\mathsf{T} \Sigma_2\big) \\ \mathrm{vec}\big(\big[O \;\; I_{n-p}\big]\, \mathrm{D}\tilde{q}^\mathsf{T} \Sigma_2\big) \\ -\mathrm{vec}\big(\big[O \;\; I_{n-p}\big]\, \mathrm{D}\tilde{q}^\mathsf{T} \Sigma_2\big)^\mathsf{T} \\ \mathrm{vec}(O_{n-p}) \end{bmatrix} = J_h^{\Sigma_1}\, \mathrm{vec}(\mathrm{D}\tilde{q}^\mathsf{T}) = J_h^{\Sigma_1}\, \Pi_{n,n}\, \mathrm{vec}(\mathrm{D}\tilde{q}),$$

with

$$J_h^{\Sigma_1} = \begin{bmatrix} \Sigma_2^\mathsf{T} \otimes \big[I_p \;\; O\big] \\ \Sigma_2^\mathsf{T} \otimes \big[O \;\; I_{n-p}\big] \\ -\Pi_{(n-p),p}\, \big(\Sigma_2^\mathsf{T} \otimes \big[O \;\; I_{n-p}\big]\big) \\ O_{(n-p)^2 \times n^2} \end{bmatrix}.$$

Observe that

$$\mathrm{vec}(\widetilde{Q}) = \mathrm{vec}\big(\big[\Sigma_1 \;\; U_\perp\big]\big) = \begin{bmatrix} \mathrm{vec}(\Sigma_1) \\ \mathrm{vec}(U_\perp) \end{bmatrix},$$

hence

$$\mathrm{vec}(\mathrm{D}\tilde{q}(\Sigma_1)[E]) = \begin{bmatrix} \mathrm{vec}(\mathrm{D}\Sigma_1) \\ \mathrm{vec}(\mathrm{D}U_\perp) \end{bmatrix} = \begin{bmatrix} I_{np} \\ J^{\Sigma_1}_{U_\perp} \end{bmatrix} \mathrm{vec}(E) = J^{\Sigma_1}_{\tilde{q}}\, \mathrm{vec}(E), \quad \text{(C.2)}$$

where the Jacobian of $U_\perp$ with respect to Σ_1 can be derived from [Vac94] as:

$$J^{\Sigma_1}_{U_\perp} = -\Big(U_\perp^\mathsf{T} \otimes \big(U_p S_p^{-1} V_p^\mathsf{T}\big)\Big)\, \Pi_{n,p}.$$

Eventually, the vectorization of $\mathrm{D}g(A)[E]$ is

$$\mathrm{vec}(\mathrm{D}g(A)[E]) = J^A_{\exp(A)} \underbrace{T J_h^{\Sigma_1}\, \Pi_{n,n}\, J_{\tilde{q}}^{\Sigma_1}}_{=: J_A^{\Sigma_1}}\, \mathrm{vec}(E), \quad \text{(C.3)}$$

from which we identify the Jacobian of the exponential with respect to Σ_1, namely,

$$J^{\Sigma_1}_{\exp(A)} = J^{A}_{\exp(A)} J^{\Sigma_1}_{A}.$$

Substituting (C.2) and (C.3) into (C.1) and dropping $\mathrm{vec}(E)$, we obtain the Jacobian of the geodesic with respect to Σ_1

$$J^{\Sigma_1}_{Z_1} = \Big([I_p \;\; O] \exp(A)^{\mathsf{T}} \otimes I_n \Big) J^{\Sigma_1}_{\tilde{q}} + \Big([I_p \;\; O] \otimes \widetilde{Q} \Big) J^{\Sigma_1}_{\exp(A)}.$$

By using the same procedure, one can get the Jacobian of the derivative of the geodesic with respect to Σ_1, i.e.,

$$\begin{aligned} J^{\Sigma_1}_{Z_2} = &\Big([I_p \;\; O] A^{\mathsf{T}} \exp(A)^{\mathsf{T}} \otimes I_n \Big) J^{\Sigma_1}_{\tilde{q}} + \Big([I_p \;\; O] A^{\mathsf{T}} \otimes \widetilde{Q} \Big) J^{\Sigma_1}_{\exp(A)} \\ &+ \Big([I_p \;\; O] \otimes \widetilde{Q} \exp(A) \Big) J^{\Sigma_1}_{A}. \end{aligned}$$

C.2 Jacobians with respect to the tangent vector

To obtain the Jacobians with respect to the tangent vector Σ_2, one can proceed in a very similar way as in the previous section. The Jacobian of the geodesic with respect to Σ_2 is given by

$$J^{\Sigma_2}_{Z_1} = \Big([I_p \;\; O] \otimes \widetilde{Q} \Big) J^{\Sigma_2}_{\exp(A)},$$

and the Jacobian of the derivative of the geodesic with respect to Σ_2 is

$$J^{\Sigma_2}_{Z_2} = \Big([I_p \;\; O] \otimes \widetilde{Q} \Big) \Big[(A^{\mathsf{T}} \otimes I_n) J^{\Sigma_2}_{\exp(A)} + (I_n \otimes \exp(A)) J^{\Sigma_2}_{A} \Big].$$

Here,

$$J^{\Sigma_2}_{\exp(A)} = J^{A}_{\exp(A)} J^{\Sigma_2}_{A} \quad \text{and} \quad J^{\Sigma_2}_{A} = T J^{\Sigma_2}_{h},$$

with

$$J^{\Sigma_2}_{h} = \begin{bmatrix} I_p \otimes \Big([I_p \;\; O] \widetilde{Q}^{\mathsf{T}} \Big) \\ I_p \otimes \Big([O \;\; I_{n-p}] \widetilde{Q}^{\mathsf{T}} \Big) \\ -\Pi_{(n-p),p} \Big(I_p \otimes [O \;\; I_{n-p}] \widetilde{Q}^{\mathsf{T}} \Big) \\ O_{(n-p)^2 \times np} \end{bmatrix} \in \mathbb{R}^{n^2 \times np}.$$

Appendix D

Proofs to Chapter 3

D.1 Proof of Remark 3.3

As we mentioned in Remark 3.3, from the expansion of the canonical distance in Equation (D.4) (see next section), it is clear that

$$d_{\mathrm{c}}(X,Y) \leqslant \|X-Y\|_{\mathrm{F}} + O(\|X-Y\|_{\mathrm{F}}^2) \quad \text{for} \quad \|X-Y\|_{\mathrm{F}} \to 0.$$

If $O(\|X-Y\|_{\mathrm{F}}^2)$ is small enough, then $d_{\mathrm{c}}(X,Y) \lesssim \|X-Y\|_{\mathrm{F}}$.

For the Riemannian distance $d_{\mathrm{e}}(X,Y)$ based on the embedded metric, it is much easier to see that $\|X-Y\|_{\mathrm{F}} \leqslant d_{\mathrm{e}}(X,Y)$, for any X, Y on a manifold $\mathcal{M}$. Indeed, the Euclidean distance is the minimum length of all possible paths in the embedding space, whereas the Riemannian distance is the minimum length of all possible paths on the manifold (see Definition 1.26). Since the embedding space contains the manifold, the paths that are considered in the second case are also paths in the first case, and that implies that the Euclidean distance $\|\cdot\|_{\mathrm{F}}$ must be smaller than or equal to the Riemannian distance d_{e}. Figure D.1 illustrates this fact for the Stiefel manifold $\mathrm{St}(10,4)$.

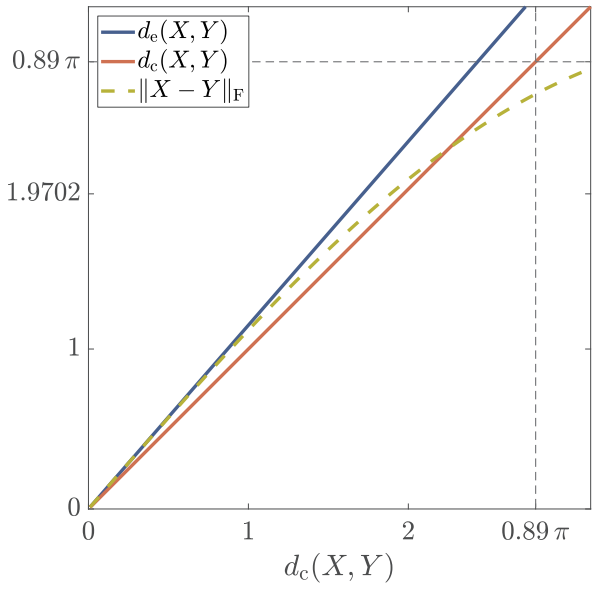


Figure D.1 – Comparison between embedded, canonical, and Euclidean distance for $\mathrm{St}(10,4)$.

D.2 Proof of Lemma 3.9

The expansion (3.6) is simple to obtain once the Riemannian distance is related to the Euclidean one.

Proof of Lemma 3.9. Take $X, Y \in \mathrm{St}(n,p)$ sufficiently close so that we can define the Riemannian logarithm $\xi = \mathrm{Log}_X(Y)$ (see Remark 3.3). By definition of the Riemannian distance d_{c} for the canonical metric g_{c}, we have

$$d_{\mathrm{c}}^2(X,Y) = \|\xi\|_{\mathrm{c}}^2 = g_{\mathrm{c}}(\xi,\xi).$$

Writing a tangent vector as $\xi = X\Omega + X_\perp K \in T_X\mathrm{St}(n,p)$ (see Section 1.1.9) and using (2.1), we can evaluate g_{c} as

$$g_{\mathrm{c}}(\xi,\xi) = \mathrm{trace}(\xi^{\mathsf{T}}(I_n - \tfrac{1}{2}\,XX^{\mathsf{T}})\,\xi) = \tfrac{1}{2}\,\|\Omega\|_{\mathrm{F}}^2 + \|K\|_{\mathrm{F}}^2 = \|\xi\|_{\mathrm{F}}^2 - \tfrac{1}{2}\,\|\Omega\|_{\mathrm{F}}^2.$$

Using $\Omega = X^{\mathsf{T}}\xi$, we also have

$$d_{\mathrm{c}}^2(X,Y) = \|\xi\|_{\mathrm{F}}^2 - \tfrac{1}{2}\,\|X^{\mathsf{T}}\xi\|_{\mathrm{F}}^2. \tag{D.1}$$

Since ξ is the initial velocity vector of the geodesic connecting X to Y, it follows that

$$\xi = Y - X + O(\|\xi\|_{\mathrm{F}}^2). \tag{D.2}$$

This can be seen by expanding the matrix exponential in the expression (2.2) of the geodesic:

$$\begin{aligned} Y &= \begin{bmatrix} X & X_\perp \end{bmatrix} \left(I_n + \begin{bmatrix} X^{\mathsf{T}}\xi & -\xi^{\mathsf{T}}X_\perp \\ X_\perp^{\mathsf{T}}\xi & O_{n-p} \end{bmatrix} + O\left(\|\xi\|_{\mathrm{F}}^2\right) \right) \begin{bmatrix} I_p \\ O_{(n-p)\times p} \end{bmatrix} \\ &= X + \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} X & X_\perp \end{bmatrix}^{\mathsf{T}} \xi + O(\|\xi\|_{\mathrm{F}}^2). \end{aligned}$$

We obtain (D.2) using the fact that $\begin{bmatrix} X & X_\perp \end{bmatrix}$ is an orthogonal matrix. In addition, [Bel03, Lemma 4.2.1, p. 59] shows that

$$\|\xi\|_{\mathrm{F}}^2 = \|X - Y\|_{\mathrm{F}}^2 + O(\|X - Y\|_{\mathrm{F}}^4). \tag{D.3}$$

Then inserting the equations (D.2) and (D.3) into (D.1) leads to

$$d_{\mathrm{c}}^2(X,Y) = \|X - Y\|_{\mathrm{F}}^2 - \tfrac{1}{2}\|X^{\mathsf{T}}(X - Y)\|_{\mathrm{F}}^2 + O(\|X - Y\|_{\mathrm{F}}^4). \tag{D.4}$$

Using this in (3.4), one obtains (3.6). □

D.3 Proof of Lemma 3.10

The aim is to compute $L_{ij} = \nabla_{X_i}\nabla_{X_j}\widetilde{d}^2(X_i, X_j)$ and $D_{ij} = \nabla^2_{X_i}\widetilde{d}^2(X_i, X_j)$, where $X_j \in \mathrm{St}(n,p)$. Let us simplify notation and define

$$\begin{aligned} \widetilde{d}^2(X,Y) = {} & \|\mathrm{P}_{\mathrm{St}}X - \mathrm{P}_{\mathrm{St}}Y\|_{\mathrm{F}}^2 - \tfrac{1}{2}\|I_p - (\mathrm{P}_{\mathrm{St}}X)^{\mathsf{T}}\mathrm{P}_{\mathrm{St}}Y\|_{\mathrm{F}}^2 \\ & + \|X - \mathrm{P}_{\mathrm{St}}X\|_{\mathrm{F}}^2 + \|Y - \mathrm{P}_{\mathrm{St}}Y\|_{\mathrm{F}}^2 + O(\|\mathrm{P}_{\mathrm{St}}X - \mathrm{P}_{\mathrm{St}}Y\|_{\mathrm{F}}^4). \end{aligned}$$

Clearly, $L_{ij} = \nabla_X\nabla_Y\widetilde{d}^2(X,Y)$ and $D_{ij} = \nabla^2_X\widetilde{d}^2(X,Y)$ with $X = X_i$ and $Y = X_j$. Recall from Section 3.2.2 that we can specify the projector on the Stiefel manifold as $\mathrm{P}_{\mathrm{St}}(Y) = Y(Y^{\mathsf{T}}Y)^{-1/2}$, that is, the orthogonal factor of the polar decomposition of Y.

Proof of Lemma 3.10. Directly developing the whole of $\widetilde{d}^2(X,Y)$ we get

$$\begin{aligned}\widetilde{d}^2(X,Y) = \operatorname{trace}\Big(&\tfrac{7}{2}I_p + X^\mathsf{T}X + Y^\mathsf{T}Y - (\mathrm{P}_{\mathrm{St}}X)^\mathsf{T}\mathrm{P}_{\mathrm{St}}Y - 2X^\mathsf{T}\mathrm{P}_{\mathrm{St}}X - 2Y^\mathsf{T}\mathrm{P}_{\mathrm{St}}Y \\ &- \tfrac{1}{2}(\mathrm{P}_{\mathrm{St}}Y)^\mathsf{T}\mathrm{P}_{\mathrm{St}}X(\mathrm{P}_{\mathrm{St}}X)^\mathsf{T}\mathrm{P}_{\mathrm{St}}Y\Big) + O(\|\mathrm{P}_{\mathrm{St}}X - \mathrm{P}_{\mathrm{St}}Y\|_{\mathrm{F}}^4).\end{aligned} \tag{D.5}$$

To compute the gradient and the Hessian of $\widetilde{d}^2(X,Y)$, consider the perturbation $\widetilde{X} = X+E$, with $X \in \mathrm{St}(n,p)$, $\|E\|_{\mathrm{F}}$ small, and expand the previous expression.

First of all, for a symmetric matrix A, one can easily show by diagonalizing that

$$(I+A)^{-1/2} = I - \tfrac{1}{2}A + \tfrac{3}{8}A^2 + O(\|A\|^3), \qquad \|A\| \to 0,$$

from which we can obtain the expansion for the perturbed projector

$$\begin{aligned}\mathrm{P}_{\mathrm{St}}\widetilde{X} &= \widetilde{X}(\widetilde{X}^\mathsf{T}\widetilde{X})^{-1/2} \\ &= X + E - \tfrac{1}{2}XX^\mathsf{T}E - \tfrac{1}{2}XE^\mathsf{T}X - \tfrac{1}{2}XE^\mathsf{T}E - \tfrac{1}{2}EX^\mathsf{T}E - \tfrac{1}{2}EE^\mathsf{T}X \\ &\quad + \tfrac{3}{8}X(X^\mathsf{T}E)^2 + \tfrac{3}{8}X(E^\mathsf{T}X)^2 + \tfrac{3}{8}XX^\mathsf{T}EE^\mathsf{T}X + \tfrac{3}{8}XE^\mathsf{T}XX^\mathsf{T}E + O(\|E\|_{\mathrm{F}}^3).\end{aligned} \tag{D.6}$$

After substituting the expansion (D.6) for $\mathrm{P}_{\mathrm{St}}(\widetilde{X})$ in (D.5) and isolating first- and second-order terms in E, we find the expressions for the gradient and the Hessian. Here, only the final results are reported.

The gradient with respect to X is

$$\nabla_X \widetilde{d}^2(X,\widetilde{Y}) = -(I_n - \tfrac{1}{2}XX^\mathsf{T})\mathrm{P}_{\mathrm{St}}\widetilde{Y} + \tfrac{1}{2}X(\mathrm{P}_{\mathrm{St}}\widetilde{Y})^\mathsf{T}X - (I_n - XX^\mathsf{T})\mathrm{P}_{\mathrm{St}}\widetilde{Y}(\mathrm{P}_{\mathrm{St}}\widetilde{Y})^\mathsf{T}X,$$

and the gradient with respect to Y is

$$\nabla_Y \widetilde{d}^2(\widetilde{X},Y) = -(I_n - \tfrac{1}{2}YY^\mathsf{T})\mathrm{P}_{\mathrm{St}}\widetilde{X} + \tfrac{1}{2}Y(\mathrm{P}_{\mathrm{St}}\widetilde{X})^\mathsf{T}Y - (I_n - YY^\mathsf{T})\mathrm{P}_{\mathrm{St}}\widetilde{X}(\mathrm{P}_{\mathrm{St}}\widetilde{X})^\mathsf{T}Y.$$

The Hessian matrix with respect to X is

$$\begin{aligned}\nabla_X^2 \widetilde{d}^2(X,Y) = \operatorname{sym}\Big[&Y^\mathsf{T}X \otimes I_n + (Y^\mathsf{T}\otimes X)\,\Pi_{p,n} + I_p \otimes YX^\mathsf{T}\Big] - \tfrac{3}{4}\operatorname{sym}\Big[(Y^\mathsf{T}XX^\mathsf{T}\otimes X)\,\Pi_{p,n} \\ &+ (X^\mathsf{T}\otimes XY^\mathsf{T}X)\,\Pi_{p,n} + I_p\otimes XX^\mathsf{T}YX^\mathsf{T} + Y^\mathsf{T}X\otimes XX^\mathsf{T}\Big] \\ &+ 2\operatorname{sym}\Big[(X^\mathsf{T}YY^\mathsf{T}\otimes X)\,\Pi_{p,n} + I_p\otimes XX^\mathsf{T}YY^\mathsf{T} - (X^\mathsf{T}YY^\mathsf{T}XX^\mathsf{T}\otimes X)\,\Pi_{p,n}\Big] \\ &+ (X^\mathsf{T}\otimes X)\,\Pi_{p,n} + I_p\otimes XX^\mathsf{T} - I_p\otimes YY^\mathsf{T} + X^\mathsf{T}YY^\mathsf{T}X\otimes I_n \\ &- I_p\otimes XX^\mathsf{T}YY^\mathsf{T}XX^\mathsf{T} - X^\mathsf{T}YY^\mathsf{T}X\otimes XX^\mathsf{T},\end{aligned}$$

where $\operatorname{sym}(A) = (A+A^\mathsf{T})/2$. In order to simplify $\nabla_X^2\widetilde{d}^2(X,Y)$, we will take $Y = X+\Delta$ with $\|\Delta\| \to 0$. After some algebraic manipulations, we obtain[1]

$$\begin{aligned}\nabla_X^2 \widetilde{d}^2(X,X+\Delta) = &\,2I_{np} + \tfrac{1}{2}(X^\mathsf{T}\otimes X)\,\Pi_{p,n} - \tfrac{1}{2}(I_p\otimes XX^\mathsf{T}) + \\ &+ 3\operatorname{sym}\big(X^\mathsf{T}\Delta\otimes I_n + (\Delta^\mathsf{T}\otimes X)\,\Pi_{p,n}\big) + \operatorname{sym}(I_p\otimes \Delta X^\mathsf{T}) \\ &- \tfrac{11}{4}\operatorname{sym}\big((\Delta^\mathsf{T}XX^\mathsf{T}\otimes X)\,\Pi_{p,n} + \Delta^\mathsf{T}X\otimes XX^\mathsf{T}\big) \\ &- \tfrac{3}{4}\operatorname{sym}\big((X^\mathsf{T}\otimes X\Delta^\mathsf{T}X)\,\Pi_{p,n} + I_p\otimes XX^\mathsf{T}\Delta X^\mathsf{T}\big) \\ &+ 2\operatorname{sym}\big((X^\mathsf{T}\Delta\Delta^\mathsf{T}\otimes X)\Pi_{p,n} + I_p\otimes XX^\mathsf{T}\Delta\Delta^\mathsf{T} - (X^\mathsf{T}\Delta\Delta^\mathsf{T}XX^\mathsf{T}\otimes X)\Pi_{p,n}\big) \\ &- I_p\otimes\Delta\Delta^\mathsf{T} + X^\mathsf{T}\Delta\Delta^\mathsf{T}X\otimes I_n - I_p\otimes XX^\mathsf{T}\Delta\Delta^\mathsf{T}XX^\mathsf{T} - X^\mathsf{T}\Delta\Delta^\mathsf{T}X\otimes XX^\mathsf{T}.\end{aligned}$$

[1]We stress that $\nabla_X^2\widetilde{d}^2$ denotes the derivative with respect to the first argument of $\widetilde{d}^2$.

Observe that every term on the right-hand side above can be bounded by at most a second power of $\|\Delta\|_2$ since $\|\operatorname{sym}(A)\|_2 \leqslant \|A\|_2$, $\|A \otimes B\|_2 = \|A\|_2\|B\|_2$ and $X \in \operatorname{St}(n,p)$. Hence, we obtain after some manipulation that

$$\|\nabla_X^2 \widetilde{d}^2(X, X+\Delta) - \nabla_X^2 \widetilde{d}^2(X,X)\|_2 \leqslant 14\|\Delta\|_2 + 10\|\Delta\|_2^2.$$

Writing the result with $X = X_i$ and $X + \Delta = X_j$, we recover the expression (3.7) for the Hessian $D_{ij} = \nabla_{X_i}^2 \widetilde{d}^2(X_i, X_j)$.

Next, for the term L_{ij}, to obtain the gradient of $\nabla_X \widetilde{d}^2(X, \widetilde{Y})$ with respect to X we can expand $\operatorname{P}_{\operatorname{St}}\widetilde{X}$ at first order in E

$$\operatorname{P}_{\operatorname{St}}\widetilde{X} = \widetilde{X}(\widetilde{X}^\mathsf{T}\widetilde{X})^{-1/2} = X + E - \tfrac{1}{2}XX^\mathsf{T}E - \tfrac{1}{2}XE^\mathsf{T}X + O(\|E\|_{\mathrm{F}}^2).$$

After some manipulations, we arrive at the mixed term

$$\begin{aligned}
\nabla_X \nabla_Y \widetilde{d}^2(X,Y) = &- I_{np} + \tfrac{1}{2}(I_p \otimes YY^\mathsf{T}) - \tfrac{1}{4}(I_p \otimes YY^\mathsf{T}XX^\mathsf{T}) - \tfrac{1}{4}(X^\mathsf{T} \otimes YY^\mathsf{T}X)\,\Pi_{p,n} \\
&+ \tfrac{1}{2}(I_p \otimes XX^\mathsf{T}) + \tfrac{1}{2}(X^\mathsf{T} \otimes X)\Pi_{p,n} + \tfrac{1}{2}(Y^\mathsf{T} \otimes Y)\Pi_{p,n} - \tfrac{1}{4}(Y^\mathsf{T}XX^\mathsf{T} \otimes Y)\Pi_{p,n} \\
&- \tfrac{1}{4}(Y^\mathsf{T}X \otimes YX^\mathsf{T}) + (Y^\mathsf{T} \otimes YY^\mathsf{T}X)\Pi_{p,n} - (Y^\mathsf{T}XX^\mathsf{T} \otimes YY^\mathsf{T}X)\Pi_{p,n} \\
&- Y^\mathsf{T}X \otimes YY^\mathsf{T}XX^\mathsf{T} + Y^\mathsf{T}X \otimes YY^\mathsf{T} - (Y^\mathsf{T} \otimes X)\,\Pi_{p,n} + (Y^\mathsf{T}XX^\mathsf{T} \otimes X)\,\Pi_{p,n} \\
&+ Y^\mathsf{T}X \otimes XX^\mathsf{T} - Y^\mathsf{T}X \otimes I_n.
\end{aligned}$$

Similarly, we can calculate the other mixed term, which is

$$\begin{aligned}
\nabla_Y \nabla_X \widetilde{d}^2(X,Y) = &- I_{np} + \tfrac{1}{2}(I_p \otimes XX^\mathsf{T}) - \tfrac{1}{4}(I_p \otimes YY^\mathsf{T}XX^\mathsf{T}) - \tfrac{1}{4}(Y^\mathsf{T}XX^\mathsf{T} \otimes Y)\,\Pi_{p,n} \\
&+ \tfrac{1}{2}(I_p \otimes YY^\mathsf{T}) + \tfrac{1}{2}(Y^\mathsf{T} \otimes Y)\Pi_{p,n} + \tfrac{1}{2}(X^\mathsf{T} \otimes X)\Pi_{p,n} - \tfrac{1}{4}(X^\mathsf{T} \otimes YY^\mathsf{T}X)\Pi_{p,n} \\
&- \tfrac{1}{4}(Y^\mathsf{T}X \otimes YX^\mathsf{T}) + (Y^\mathsf{T}XX^\mathsf{T} \otimes X)\,\Pi_{p,n} - (Y^\mathsf{T}XX^\mathsf{T} \otimes YY^\mathsf{T}X)\,\Pi_{p,n} \\
&- Y^\mathsf{T}X \otimes YY^\mathsf{T}XX^\mathsf{T} + Y^\mathsf{T}X \otimes XX^\mathsf{T} - (Y^\mathsf{T} \otimes X)\,\Pi_{p,n} + (Y^\mathsf{T} \otimes YY^\mathsf{T}X)\,\Pi_{p,n} \\
&+ Y^\mathsf{T}X \otimes YY^\mathsf{T} - Y^\mathsf{T}X \otimes I_n.
\end{aligned}$$

Observe that, by swapping the arguments and taking the transpose, we have the equality

$$\nabla_Y \nabla_X \widetilde{d}^2(X,Y) = \Big(\nabla_X \nabla_Y \widetilde{d}^2(Y,X)\Big)^\mathsf{T}.$$

As above, in order to bound the spectrum of $\nabla_X \nabla_Y \widetilde{d}^2(X,Y)$, we expand it with $Y = X + \Delta$ with $\|\Delta\| \to 0$. After some algebraic manipulations, we obtain

$$\begin{aligned}
\nabla_X \nabla_Y \widetilde{d}^2(X, X+\Delta) = &- 2I_{np} + \tfrac{1}{2}(X^\mathsf{T} \otimes X)\,\Pi_{p,n} + \tfrac{3}{2}(I_p \otimes XX^\mathsf{T}) \\
&- \tfrac{1}{4}(X^\mathsf{T} \otimes X\Delta^\mathsf{T}X)\,\Pi_{p,n} + \tfrac{1}{2}(\Delta^\mathsf{T} \otimes X)\,\Pi_{p,n} - \tfrac{1}{4}(\Delta^\mathsf{T}XX^\mathsf{T} \otimes X)\,\Pi_{p,n} \\
&+ \tfrac{3}{4}(\Delta^\mathsf{T}X \otimes XX^\mathsf{T}) - \tfrac{5}{4}(I_p \otimes X\Delta^\mathsf{T}XX^\mathsf{T}) + \tfrac{3}{2}(I_p \otimes X\Delta^\mathsf{T}) - \Delta^\mathsf{T}X \otimes I_n \\
&- \tfrac{5}{4}(\Delta^\mathsf{T}XX^\mathsf{T} \otimes \Delta)\,\Pi_{p,n} - \tfrac{5}{4}(I_p \otimes \Delta\Delta^\mathsf{T}XX^\mathsf{T}) + \tfrac{3}{2}(I_p \otimes \Delta\Delta^\mathsf{T}) \\
&+ \Delta^\mathsf{T}X \otimes X\Delta^\mathsf{T} + \tfrac{3}{2}(\Delta^\mathsf{T} \otimes \Delta)\,\Pi_{p,n} - (\Delta^\mathsf{T}XX^\mathsf{T} \otimes X\Delta^\mathsf{T}X)\,\Pi_{p,n} \\
&- \tfrac{1}{4}(X^\mathsf{T} \otimes \Delta\Delta^\mathsf{T}X)\,\Pi_{p,n} + (\Delta^\mathsf{T} \otimes X\Delta^\mathsf{T}X)\,\Pi_{p,n} - \tfrac{1}{4}(\Delta^\mathsf{T}X \otimes \Delta X^\mathsf{T}) \\
&- \Delta^\mathsf{T}X \otimes X\Delta^\mathsf{T}XX^\mathsf{T} + (\Delta^\mathsf{T} \otimes \Delta\Delta^\mathsf{T}X)\,\Pi_{p,n} + \Delta^\mathsf{T}X \otimes \Delta\Delta^\mathsf{T} \\
&- \Delta^\mathsf{T}X \otimes \Delta\Delta^\mathsf{T}XX^\mathsf{T} - (\Delta^\mathsf{T}XX^\mathsf{T} \otimes \Delta\Delta^\mathsf{T}X)\,\Pi_{p,n}.
\end{aligned}$$

Observe that every term on the right-hand side above can be bounded by at most a third power of $\|\Delta\|_2$. Hence, we obtain that

$$\|\nabla_X \nabla_Y \widetilde{d}^2(X, X+\Delta) - \nabla_X \nabla_Y \widetilde{d}^2(X,X)\|_2 \leqslant \tfrac{11}{2}\|\Delta\|_2 + 10\|\Delta\|_2^2 + 4\|\Delta\|_2^3.$$

Writing the result with $X = X_i$ and $Y = X_j$, we recover the expression (3.8) for the gradient $L_{ij} = \nabla_{X_i}\nabla_{X_j}\widetilde{d}^2(X_i, X_j)$. □

D.4 Proof of Lemma 3.11

We first start with $i = j$, which corresponds to $\Delta_{ij} = \Lambda_{ij} = 0$ in Lemma 3.10, and prove the following auxiliary lemma.

Lemma D.1. *Define the orthogonal matrix*

$$\bar{Q}_i = \begin{bmatrix} I_p \otimes X_i & I_p \otimes X_i^{\perp} \end{bmatrix} \in O(np).$$

Then there exists an orthogonal matrix $\widehat{Q}$, only depending on n and p, such that $Q_i = \bar{Q}_i \widehat{Q}$ satisfies

$$Q_i^T D_{ii} Q_i = D = \begin{bmatrix} I_{p(p-1)/2} & \\ & 2I_{np-p(p-1)/2} \end{bmatrix} \tag{D.7}$$

and

$$Q_i^T L_{ii} Q_i = L = \begin{bmatrix} -I_{p(p-1)/2} & & \\ & -2I_{(n-p)p} & \\ & & O_{p(p+1)/2} \end{bmatrix}. \tag{D.8}$$

Proof. By properties of the so-called vec-permutation matrices (see [HS81, Eq. (5), (6), (23)]), there exists a permutation matrix $\Pi_{p,n} \in \mathbb{R}^{np \times np}$ that satisfies

$$\Pi_{p,n}(X_i^{\mathsf{T}} \otimes X_i)\,\Pi_{p,n} = X_i \otimes X_i^{\mathsf{T}}, \quad \Pi_{p,n}^{-1} = \Pi_{p,n}^{\mathsf{T}}.$$

This shows that $(X_i^{\mathsf{T}} \otimes X_i)\,\Pi_{p,n} = \Pi_{p,n}^{\mathsf{T}}(X_i \otimes X_i^{\mathsf{T}})$ is symmetric. Furthermore,

$$((X_i^{\mathsf{T}} \otimes X_i)\,\Pi_{p,n})^2 = (X_i^{\mathsf{T}} \otimes X_i)\,\Pi_{p,n}\,\Pi_{p,n}^{\mathsf{T}}(X_i \otimes X_i^{\mathsf{T}}) = I_p \otimes X_i X_i^{\mathsf{T}}.$$

Denoting the symmetric matrix $S_i = (X_i^{\mathsf{T}} \otimes X_i)\,\Pi_{p,n}$, we can then use Lemma 3.10 to write

$$D_{ii} = 2I_{np} + \tfrac{1}{2}S_i - \tfrac{1}{2}S_i^2, \qquad L_{ii} = -2I_{np} + \tfrac{1}{2}S_i + \tfrac{3}{2}S_i^2. \tag{D.9}$$

It thus suffices to diagonalize S_i. Using the matrix $\bar{Q}_i$ defined in the statement of the lemma, direct calculation shows that

$$\bar{Q}_i^{\mathsf{T}} S_i \bar{Q}_i = \begin{bmatrix} (X_i^{\mathsf{T}} \otimes I_p)\,\Pi_{p,n}(I_p \otimes X_i) & \\ & O_{(n-p)p} \end{bmatrix} = \begin{bmatrix} \Pi_{p,p} & \\ & O_{(n-p)p} \end{bmatrix} =: \widehat{\Pi},$$

where we used that $\Pi_{p,n}(I_p \otimes X_i)\Pi_{p,p} = X_i \otimes I_p$, with $\Pi_{p,p} \in \mathbb{R}^{p^2 \times p^2}$ another vec-permutation matrix that is also symmetric (see [HS81, Eq. (6), (15)]). The matrix $\widehat{\Pi}$ above therefore has the spectral decomposition

$$\widehat{\Pi} = \widehat{Q}\widehat{\Lambda}\widehat{Q}^{\mathsf{T}}, \qquad \widehat{\Lambda} = \begin{bmatrix} -I_{p(p-1)/2} & & \\ & O_{(n-p)p} & \\ & & I_{p(p+1)/2} \end{bmatrix}, \tag{D.10}$$

for some orthogonal matrix $\widehat{Q}$ that indeed does not depend on X_i, as claimed. By defining the orthogonal matrix $Q_i = \bar{Q}_i \widehat{Q}$, we have thus shown that $Q_i^{\mathsf{T}} S_i Q_i = \widehat{\Lambda}$, and by (D.9) also that

$$Q_i^{\mathsf{T}} D_{ii} Q_i = 2I_{np} + \tfrac{1}{2}\widehat{\Lambda} - \tfrac{1}{2}\widehat{\Lambda}^2, \quad Q_i^{\mathsf{T}} L_{ii} Q_i = -2I_{np} + \tfrac{1}{2}\widehat{\Lambda} + \tfrac{3}{2}\widehat{\Lambda}^2.$$

It is straightforward to verify that these matrices can be written as the claimed matrices D and L. □

Lemma 3.11 is now proven as a perturbation of the case above.

Proof of Lemma 3.11. From Lemma 3.10, we know that $L_{ij} = L_{ii} + \Lambda_{ij}$. Lemma D.1 therefore gives

$$\begin{aligned} Q_j^\mathsf{T} L_{ij} Q_i &= (Q_j - Q_i)^\mathsf{T} L_{ij} Q_i + Q_i^\mathsf{T} L_{ij} Q_i \\ &= (Q_j - Q_i)^\mathsf{T} L_{ij} Q_i + L + Q_i^\mathsf{T} \Lambda_{ij} Q_i. \end{aligned}$$

Taking norms and recalling that $\delta_{ij} = \|Q_j - Q_i\|_2$, we obtain

$$\|Q_j^\mathsf{T} L_{ij} Q_i - L\|_2 \leqslant \delta_{ij}(\|L_{ii}\|_2 + \|\Lambda_{ij}\|_2) + \|\Lambda_{ij}\|_2.$$

Since $\|L_{ii}\|_2 = \|L\|_2 \leqslant 2$ by Lemma D.1, this shows (3.11). The bound (3.10) is similarly proven. □

Bibliography

[AAM14] Absil, P.-A., Amodei, L., and Meyer, G. Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries. *Computational Statistics*, 29:569–590, 2014.

[AF11] Amsallem, D. and Farhat, C. An Online Method for Interpolating Linear Parametric Reduced-Order Models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.

[AM12] Absil, P.-A. and Malick, J. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

[AMS04] Absil, P.-A., Mahony, R., and Sepulchre, R. Riemannian Geometry of Grassmann Manifolds with a View on Algorithmic Computation. *Acta Applicandae Mathematica*, 80(2):199–220, Jan 2004.

[AMS08] Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[AMT13] Absil, P. A., Mahony, R., and Trumpf, J. An Extrinsic Look at the Riemannian Hessian. In *Geometric Science of Information*, pages 361–368, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[AO15] Absil, P.-A. and Oseledets, I. V. Low-rank retractions: a survey and new results. *Computational Optimization and Applications*, 62(1):5–29, Sep 2015.

[ATV13] Afsari, B., Tron, R., and Vidal, R. On the Convergence of Gradient Descent for Finding the Riemannian Center of Mass. *SIAM Journal on Control and Optimization*, 51(3):2230–2260, 2013.

[BAC18] Boumal, N., Absil, P.-A., and Cartis, C. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33, 02 2018.

[Bel03] Belkin, M. *Problems of Learning on Manifolds*. PhD thesis, The University of Chicago, 2003.

[Ber95] Bertsekas, D. *Nonlinear Programming*. Athena Scientific, 1995.

[BGW15] Benner, P., Gugercin, S., and Willcox, K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review*, 57(4):483–531, 2015.

[BHM00] Briggs, W. L., Henson, V. E., and McCormick, S. F. *A Multigrid Tutorial, Second Edition.* Society for Industrial and Applied Mathematics, second edition, 2000.

[BMAS14] Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[Boo86] Boothby, W. M. *An introduction to differentiable manifolds and Riemannian geometry; 2nd ed.* Pure Appl. Math. Academic Press, Orlando, FL, 1986.

[Bou20] Boumal, N. An introduction to optimization on smooth manifolds. Available online, November 2020.

[BP94] Berman, A. and Plemmons, R. J. *Nonnegative Matrices in the Mathematical Sciences.* Society for Industrial and Applied Mathematics, 1994.

[Bro93] Brockett, R. W. Differential geometry and the design of gradient algorithms. *Proc. of Sympo. in Pure Math*, 54:69–92, 1993.

[BS07] Brenner, S. and Scott, R. *The Mathematical Theory of Finite Element Methods.* Texts in Applied Mathematics. Springer New York, 2007.

[Cha12] Chang, X.-W. On the perturbation of the Q-factor of the QR factorization. *Numerical Linear Algebra with Applications*, 19(3):607–619, 2012.

[dC92] do Carmo, M. *Riemannian Geometry.* Mathematics (Boston, Mass.). Birkhäuser, 1992.

[EAS98] Edelman, A., Arias, T. A., and Smith, S. T. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[ES18] Elman, H. C. and Su, T. A Low-Rank Multigrid Method for the Stochastic Steady-State Diffusion Problem. *SIAM Journal on Matrix Analysis and Applications*, 39(1):492–509, 2018.

[Fin08] Finden, W. An error term and uniqueness for Hermite–Birkhoff interpolation involving only function values and/or first derivative values. *Journal of Computational and Applied Mathematics*, 212(1):1 – 15, 2008.

[FV62] Feingold, D. G. and Varga, R. S. Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem. *Pacific J. Math.*, 12:1241—1250, 1962.

[GH07] Grasedyck, L. and Hackbusch, W. A Multigrid Method to Solve Large Scale Sylvester Equations. *SIAM Journal on Matrix Analysis and Applications*, 29(3):870–894, 2007.

[GK73] Grove, K. and Karcher, H. How to Conjugate C^1-Close Group Actions. *Mathematische Zeitschrift*, 132:11–20, 1973.

[GLL86] Grippo, L., Lampariello, F., and Lucidi, S. A Nonmonotone Line Search Technique for Newton's Method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.

[GMS+10] Gratton, S., Mouffe, M., Sartenaer, A., Toint, P. L., and Tomanos, D. Numerical experience with a recursive trust-region method for multilevel nonlinear bound-constrained optimization. *Optimization Methods and Software*, 25(3):359–386, 2010.

[Gov94] Gover, M. J. C. The Eigenproblem of a Tridiagonal 2-Toeplitz Matrix. *Linear Algebra and its Applications*, 197-198:63 – 78, 1994.

[GS00] Grippo, L. and Sciandrone, M. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Operations Research Letters*, 26(3):127 – 136, 2000.

[GSAS20] Gao, B., Son, N. T., Absil, P.-A., and Stykel, T. Riemannian optimization on the symplectic Stiefel manifold. 2020.

[GST08] Gratton, S., Sartenaer, A., and Toint, P. Recursive Trust-Region Methods for Multiscale Nonlinear Optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

[GVL13] Golub, G. H. and Van Loan, C. F. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences, 4rd edition, 2013.

[Hac03] Hackbusch, W. *Multi-Grid Methods and Applications*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2003.

[Hac12] Hackbusch, W. *Tensor Spaces and Numerical Tensor Calculus*. Springer, 2012.

[Hac16] Hackbusch, W. *Iterative Solution of Large Sparse Systems of Equations*. Springer, 2016.

[Hen03] Henson, V. E. Multigrid methods nonlinear problems: an overview. In Bouman, C. A. and Stevenson, R. L., editors, *Computational Imaging*, volume 5016, pages 36–48. International Society for Optics and Photonics, SPIE, 2003.

[Hig08] Higham, N. J. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[HLW06] Hairer, E., Lubich, C., and Wanner, G. *Geometric Numerical Integration*. Springer, Berlin, Heidelberg, 2006.

[HS81] Henderson, H. V. and Searle, S. R. The vec-permutation matrix, the vec operator and Kronecker products: a review. *Linear and Multilinear Algebra*, 9(4):271–288, 1981.

[HZ05] Hager, W. W. and Zhang, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005.

[HZ06] Hager, W. W. and Zhang, H. Algorithm 851: CG_DESCENT, a Conjugate Gradient Method with Guaranteed Descent. *ACM Trans. Math. Softw.*, 32(1):113–137, March 2006.

[Kar77] Karcher, H. Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.*, 30(5):509–541, September 1977.

[Kar14] Karcher, H. Riemannian center of mass and so called karcher mean, 2014.

[KN69] Kobayashi, S. and Nomizu, K. *Foundations of Differential Geometry*. Wiley, 1969.

[KN97] Kaya, C. Y. and Noakes, J. L. Geodesics and an optimal control algorithm. *Proceedings of the 36th IEEE Conference on Decision and Control (CDC)*, pages 4918–4919, 1997.

[KN98a] Kaya, C. Y. and Noakes, J. L. The leap-frog algorithm and optimal control: Background and demonstration. *Proceedings of International Conference on Optimization Techniques and Applications (ICOTA '98)*, pages 835–842, 1998.

[KN98b] Kaya, C. Y. and Noakes, J. L. The leap-frog algorithm and optimal control: Theoretical aspects. *Proceedings of International Conference on Optimization Techniques and Applications (ICOTA '98)*, pages 843–850, 1998.

[KN08] Kaya, C. Y. and Noakes, J. L. Leapfrog for Optimal Control. *SIAM Journal on Numerical Analysis*, 46(6):2795–2817, 2008.

[KSV16] Kressner, D., Steinlechner, M., and Vandereycken, B. Preconditioned Low-rank Riemannian Optimization for Linear Systems with Tensor Product Structure. *SIAM Journal on Scientific Computing*, 38(4):A2018–A2044, 2016.

[KT14] Kressner, D. and Tobler, C. Algorithm 941: Htucker—A Matlab Toolbox for Tensors in Hierarchical Tucker Format. *ACM Trans. Math. Softw.*, 40(3):22:1–22:22, April 2014.

[LDL16] Le Dret, H. and Lucquin, B. *Partial Differential Equations: Modeling, Analysis and Numerical Approximation*. Birkhäuser, Basel, 2016.

[Lee97] Lee, J. M. *Riemannian Manifolds: An Introduction to Curvature*. Graduate Texts in Mathematics. Springer New York, 1997.

[Lee18] Lee, J. M. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2018.

[LN05] Lewis, R. M. and Nash, S. G. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM J. Sci. Comput.*, 26(6):1811–1837, June 2005.

[Mil63] Milnor, J. W. *Morse Theory*, volume 51 of *Annals of Math. Studies*. Princeton University Press, 1963.

[MMBS13] Mishra, B., Meyer, G., Bach, F., and Sepulchre, R. Low-rank optimization with trace norm penalty. *SIAM Journal on Optimization*, 23(4):2124–2149, 2013.

[MV14] Mishra, B. and Vandereycken, B. A Riemannian approach to low-rank algebraic Riccati equations. In *21st International Symposium on Mathematical Theory of Networks and Systems*, 2014.

[Nas00] Nash, S. G. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14(1-2):99–116, 2000.

[NH95] Najfeld, I. and Havel, T. F. Derivatives of the Matrix Exponential and Their Computation. *Advances in Applied Mathematics*, 16(3):321–375, 1995.

[Noa98] Noakes, J. L. A global algorithm for geodesics. *Journal of the Australian Mathematical Society. Series A. Pure Mathematics and Statistics*, 65(1):37–50, 1998.

[NW06] Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[OR00] Ortega, J. and Rheinboldt, W. *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics, 2000.

[Pen97] Penzl, T. A Multi-Grid Method for Generalized Lyapunov Equations. 1997.

[Ren13] Rentmeesters, Q. *Algorithms for data fitting on some common homogeneous spaces*. PhD thesis, Université catholique de Louvain, Louvain, Belgium, 2013.

[RNO19] Rakhuba, M., Novikov, A., and Oseledets, I. Low-rank Riemannian eigensolver for high-dimensional Hamiltonians. *Journal of Computational Physics*, 396:718–737, 2019.

[RO18] Rakhuba, M. and Oseledets, I. Jacobi–Davidson Method on Low-Rank Matrix Manifolds. *SIAM Journal on Scientific Computing*, 40(2):A1149–A1170, 2018.

[RW95] Rosen, I. G. and Wang, C. A Multilevel Technique for the Approximate Solution of Operator Lyapunov and Algebraic Riccati Equations. *SIAM Journal on Numerical Analysis*, 32(2):514–541, 1995.

[Saa03] Saad, Y. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003.

[Sak96] Sakai, T. *Riemannian Geometry*. Fields Institute Communications. American Mathematical Soc., 1996.

[SB91] Stoer, J. and Bulirsch, R. *Introduction to numerical analysis*. Texts in applied mathematics. Springer, New York, 1991.

[Sim07] Simoncini, V. A New Iterative Method for Solving Large-Scale Lyapunov Matrix Equations. *SIAM Journal on Scientific Computing*, 29(3):1268–1288, 2007.

[Sim16] Simoncini, V. Computational methods for linear matrix equations. *SIAM Review*, 58(3):377–441, 2016.

[SK16] Srivastava, A. and Klassen, E. P. *Functional and Shape Data Analysis*. Springer series in Statistics. Springer, 2016.

[SS90] Stewart, G. W. and Sun, J.-g. *Matrix Perturbation Theory*. Academic Press, 1990.

[Ste16] Steinlechner, M. Riemannian Optimization for High-Dimensional Tensor Completion. *SIAM Journal on Scientific Computing*, 38(5):S461–S484, 2016.

[SWC12] Shalit, U., Weinshall, D., and Chechik, G. Online learning in the embedded manifold of low-rank matrices. *Journal of Machine Learning Research*, 13:429–458, 2012.

[Tob12] Tobler, C. *Low-rank Tensor Methods for Linear Systems and Eigenvalue Problems*. PhD thesis, ETH, Zürich, Switzerland, 2012.

[TOS00] Trottenberg, U., Oosterlee, C., and Schuller, A. *Multigrid*. Elsevier Science, 2000.

[Tre08] Tretter, C. *Spectral Theory of Block Operator Matrices and Applications*. Imperial College Press, 2008.

[TTWM09] Toint, P. L., Tomanos, D., and Weber-Mendonça, M. A multilevel algorithm for solving the trust-region subproblem. *Optimization Methods and Software*, 24(2):299–311, 2009.

[UV15] Uschmajew, A. and Vandereycken, B. Greedy rank updates combined with Riemannian descent methods for low-rank optimization. In *2015 International Conference on Sampling Theory and Applications (SampTA)*, pages 420–424, May 2015.

[UV19] Uschmajew, A. and Vandereycken, B. *Variational methods for nonlinear geometric data and applications*, chapter Geometric methods on low-rank matrix and tensor manifolds. Springer, 2019.

[Vac94] Vaccaro, R. J. A Second-Order Perturbation Expansion for the SVD. *SIAM Journal on Matrix Analysis and Applications*, 15(2):661–671, 1994.

[Van13] Vandereycken, B. Low-Rank Matrix Completion by Riemannian Optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[VV10] Vandereycken, B. and Vandewalle, S. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 31(5):2553–2579, 2010.

[WG09] Wen, Z. and Goldfarb, D. A line search multigrid method for large-scale nonlinear optimization. *SIAM Journal on Optimization*, 20(3):1478–1503, 2009.

[Won67] Wong, Y.-C. Differential Geometry of Grassmann Manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, 57(3):589–594, 1967.

[ZD19] Zimmermann, R. and Debrabant, K. Parametric Model Reduction via Interpolating Orthonormal Bases. In Radu, F. A., Kumar, K., Berre, I., Nordbotten, J. M., and Pop, I. S., editors, *Numerical Mathematics and Advanced Applications ENUMATH 2017*, pages 683–691, Cham, 2019. Springer International Publishing.

[Zim17] Zimmermann, R. A Matrix-Algebraic Algorithm for the Riemannian Logarithm on the Stiefel Manifold under the Canonical Metric. *SIAM Journal on Matrix Analysis and Applications*, 38(2):322–342, 2017.