KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Computer Science

Riemannian and multilevel optimization for rank-constrained matrix problems

with applications to Lyapunov equations

Bart Vandereycken

Dissertation presented in
partial fulfillment of the
requirements for the degree
of Doctor in Engineering

December 2010

Riemannian and multilevel optimization for rank-constrained matrix problems

with applications to Lyapunov equations

Bart Vandereycken

Jury:
Prof. Dr. ir. Paul Van Houtte, president
Prof. Dr. ir. Stefan Vandewalle, promotor
Prof. Dr. ir. Dirk Roose
Prof. Dr. ir. Marc Van Barel
Prof. Dr. Moritz Diehl
Prof. Dr. Pierre-Antoine Absil
(U.C.L., Louvain-la-Neuve)
Prof. Dr. Rodolphe Sepulchre
(Université de Liège)
Prof. Dr. Daniel Kressner
(ETH Zürich)

Dissertation presented in
partial fulfillment of the
requirements for the degree
of Doctor in Engineering

U.D.C. 519.64

December 2010

D/2010/7515/136
978-94-6018-301-0

Rev. 1.1

Preface

This thesis is the result of my research over the past years at the Computer Science department at K.U.Leuven. I found writing it much like doing the research itself. Most of the time, it is really fun! Spending hours reading and studying new mathematics, discussing ideas with other people, shaping your thoughts and performing numerical experiments. Yet sometimes, there is an unpleasant hurdle to be overcome and only perseverance will help you fix nasty proofs or debug stubborn pieces of code. Although I can hardly imagine this thesis to be a real page turner, I hope reading it will give you a similar experience: not a chore, but an interesting read.

Doing research is not a solitary business, so I would like to thank some of the people that made this work possible. Foremost, a big thank you goes out to Stefan Vandewalle for being my advisor throughout these years. It was my Master's thesis, co-supervised by Daan Huybrechs, that led me to starting a PhD with Stefan. For giving me that opportunity, I am very grateful. Since the beginning of my PhD, I had the tendency to wander off from my original topic. Stefan, thank you for letting me find my own way, encouraging me to attend international conferences and introducing me to many people.

Thanks to all the members of the jury for the many useful comments that improved the text. Since he attended my first seminar, I was touched by Moritz Diehl his enthusiasm and his ability to relate my work to that of others. It helped me gain a broader picture; thank you, Moritz. My Riemannian journey would not have been possible without Pierre-Antoine Absil and Rodolpe Sepulchre. I am very proud

and thankful to have such specialists in my jury. I am grateful to Daniel Kressner, for making the trip from Zürich and for giving me the opportunity to work at ETH. Daniel, I am looking forward working together next year. Further, thanks go to Paul Van Houtte for chairing and to Dirk Roose and Marc Van Barel for completing my jury and asking some tough questions.

I gratefully acknowledge the financial support received from the Research Foundation – Flanders (FWO) and OPTEC – the K.U.Leuven Center of Excellence: Optimization in Engineering.

Also, I want to thank some of my colleagues at the Computer Science Department that made the past five years a genuine enjoyable period. I happily shared an office with Daan, Pieter and Jerzy, and, for some briefer moments, with Eveline and Joris. Being a teaching assistant for P&O, it was a pleasure to team up with Peter, Ward, Bert, Joris and Michael. It sure made playing with Legos even more fun! Conducting research was balanced with the occasional invigorating table tennis match. A big sorry to Pieter, Joris, Giovanni and Hanne for my near-tantrums as a bad looser. Finally, besides nurturing the scientific mind, there was ample opportunity to feed the body and soul at Alma together with Sam, Eveline, Liesbeth, Elias, Joris, Pieter, Bert and Hanne. Many of you became very good friends and I am grateful our academic paths crossed.

Thanks to all my friends outside work, for bringing some sanity away from academia. To Tim, Jef, Annick, Kristof, Esther, Liesbeth and Leen for past and current friendships. Special thanks goes to Philip, Ann, Kristoff for being amongst my closest friends all these years, and a super special thanks goes out to my girlfriend Hanne.

To my family, in particular, my parents and grandparents; they taught me the importance of education, perseverance and thoroughness, all of which are essential to research. While it must have been difficult trying to understand what my research was actually about, rest assured, it could not be accomplished without you. Thank you!

Bart Vandereycken

Abstract

This thesis proposes a new framework for the numerical solution of certain rank-constrained matrix problems. By exploiting that the constrained set is a smooth manifold, we can solve these rank-constrained matrix problems by using techniques from Riemannian optimization. Specifically, we use the retraction-based optimization framework to minimize a certain cost function on $\mathbf{S}_+^{n,p}$, the set of n-by-n symmetric positive semidefinite matrices of rank p.

The application of Riemannian optimization requires some typical objects from differential geometry, like the tangent space and the Levi–Civita connection. In the first part of the thesis, we therefore derive these objects for a submanifold geometry embedded in the Euclidean space. In addition, we also derive and study the geodesics of this space.

The efficiency of the proposed framework is assessed by solving large-scale Lyapunov matrix equations that originate from discretized partial differential equations (PDEs). We propose two novel numerical methods that can be used to find a low-rank approximation of the solution of such a Lyapunov equation.

The first algorithm is the application of the Riemannian Trust-Region (RTR) method to a specifically chosen objective function in combination with the earlier derived submanifold geometry of $\mathbf{S}_+^{n,p}$. In order to obtain an algorithm that is scalable for realistic PDEs, we derive a preconditioner that allows us to solve certain subproblems of the RTR method much faster. Numerical experiments indicate that this algorithm is competitive compared to the state-of-the-art for low-rank

Lyapunov solvers, while, at the same, it is more general and flexible.

The other algorithm is the generalization of multilevel optimization to Riemannian manifolds. Compared to the previous solver, this multilevel Riemannian algorithm exploits the multilevel character of PDEs directly. This avoids the need for deriving a specific preconditioner, which makes this solver more flexible. However, in order to become an efficient multigrid solver, the typical multigrid components, like the smoother and the coarse grid operator, have to be chosen complementary. We show by a Local Fourier Analysis that the usual multigrid practice can be used for choosing these components. The numerical experiments illustrate that when the components are chosen in a correct way, one can indeed achieve textbook multigrid efficiency.

The last part of the thesis is devoted to the derivation of a new geometry for $\mathbf{S}_+^{n,p}$. While the embedded geometry from above is well-suited for our numerical algorithms, it is less attractive theoretically; in particular, it is not a complete metric space. The novelty of this part is the construction of a specific geometry for $\mathbf{S}_+^{n,p}$ such that the geodesics are complete and can be derived in closed form. Finally, we compare this geometry with some other geometries in the literature.

Samenvatting

Het onderwerp van dit werk is een nieuw raamwerk voor de numerieke oplossing van bepaalde matrixproblemen met rangbeperkingen. Door de matrices met vaste rang te beschouwen als een gladde variëteit, kunnen deze matrixproblemen opgelost worden met methodes uit de Riemannse optimalisatie. Meer bepaald kunnen we de methodes die steunen op retracties gebruiken om een bepaalde kostfunctie te minimaliseren op $\mathbf{S}_+^{n,p}$, de verzameling van vierkante, symmetrisch positief definiete matrices van dimensie n en rang p.

De toepassing van Riemannse optimalisatie vereist echter enkele typische objecten uit de differentiaalmeetkunde, zoals de raakruimte en de Levi–Civita-verbinding. Daarom is het eerste deel van dit werk geweid aan het afleiden van deze objecten voor $\mathbf{S}_+^{n,p}$ als een deelvariëteit. Daarenboven leiden we ook de geodeten af en onderzoeken hen in meer detail.

In het tweede deel onderzoeken we de effectiviteit van het voorgestelde raamwerk voor het oplossen van grootschalige Lyapunov matrixvergelijkingen voor partiële differentiaalvergelijkingen. We stellen twee nieuwe numerieke methodes voor die kunnen gebruikt worden om een benadering van lage rang te zoeken voor de oplossingen van deze Lyapunov vergelijkingen.

De eerste methode is de toepassing van de zogenaamde RTR methode uit de Riemannse optimalisatie. We leiden een specifieke kostfunctie af die, in combinatie met de eerder afgeleide geometrie, kan geminimaliseerd worden met behulp van deze RTR methode. Om een schaalbaar algoritme te bekomen, leiden we

een preconditioner af die de meest rekenintensieve deelproblemen van de RTR methode veel sneller kan oplossen. De numerieke experimenten laten zien dat dit algoritme competitief is met de state-of-the-art wat lage-rang methodes voor Lyapunov vergelijkingen betreft. Bovendien is de voorgestelde methode algemener en veelzijdiger.

Het andere algoritme is de veralgemening van meerschalige optimalisatie voor Riemannse variëteiten. Vergeleken met de vorige methode, kan dit algoritme het meerschalige karakter van partiële differentiaalvergelijkingen meteen uitbuiten. Hierdoor hoeft er geen speciale preconditioner meer afgeleid te worden en is de solver dus veelzijdiger. Om een efficiënte solver te bekomen is het daarentegen wel belangrijk dat de typische componenten van een meerschalig algoritme juist worden gekozen. We tonen aan, door middel van een Lokale Fourier Analyse, dat de typische strategiën die voor klassieke meerschalige algoritmen gebruikt worden ook hier toepasbaar zijn. De numerieke experimenten illusteren dat bij de juiste keuze van deze componenten we dan inderdaad de typische efficiëntie verkrijgen die ook bij meer klassieke meerschalige solvers van toepassing zijn.

Het laatste deel van de thesis is geweid aan het afleiden van een nieuwe geometrie voor $\mathbf{S}_+^{n,p}$. Hoewel de geometrie die $\mathbf{S}_+^{n,p}$ beschrijft als een deelvariëteit uiterst geschikt is voor onze numerieke optimalisatie, is ze niet zo aantrekkelijk vanuit een theoretisch standpunt; ze is met name niet compleet. De nieuwigheid van dit deel is de constructie van een nieuwe geometrie die wel compleet is. Tenslotte vergelijken we de geodeten van deze geometrie met andere bestaande formuleringen in de literatuur.

Contents

List of Symbols

Matrix spaces

$\mathbf{R}$	set of real numbers
$\mathbf{R}^d$	set d dimensional real vectors
$\mathbf{R}^{n\times m}$	set of $n \times m$ real matrices
I_n	the identity matrix in $\mathbf{R}^{n\times n}$
$0_{n\times m}$	the matrix of all zeros in $\mathbf{R}^{n\times m}$
$\mathbf{R}_*^{n\times m}$	set of full rank matrices
	$:= \{X \in \mathbf{R}^{n\times m} \mid \operatorname{rank}(X) = \min(n, m)\}$
$\mathbf{S}^n$	set of symmetric matrices
	$:= \{X \in \mathbf{R}^{n\times n} \mid X = X^T\}$
$\operatorname{skew}(n)$	set of skew-symmetric matrices
	$:= \{X \in \mathbf{R}^{n\times n} \mid X = -X^T\}$
$\mathbf{S}_+^n$	set of symmetric and positive semidefinite (s.p.s.d.) matrices
	$:= \{X \in \mathbf{S}^n \mid X \succeq 0\}$
$\mathbf{S}_{++}^n$	set of symmetric and positive definite (s.p.d.) matrices
	$:= \{X \in \mathbf{S}^n \mid X \succ 0\}$
$\mathbf{S}_+^{n,p}$	set of s.p.s.d. matrices of rank p
	$:= \{X \in \mathbf{S}_+^n \mid \operatorname{rank}(X) = p\}$
$\mathrm{St}^{n,p}$	set of orthonormal matrices, the Stiefel manifold
	$:= \{X \in \mathbf{R}^{n\times p} \mid p \leq n, X^T X = I_p\}$

Differential geometry

$\mathcal{M}$	a C^∞ smooth manifold
x, y, z	elements of $\mathcal{M}$
$T_x\mathcal{M}$	the tangent space of $\mathcal{M}$ at x
$T\mathcal{M}$	the tangent bundle
ν, ξ, η	vector fields on $\mathcal{M}$
$\nu(x), \xi(x), \eta(x)$	vector fields evaluated at $x \in \mathcal{M}$
$\mathcal{X}(\mathcal{M})$	set of all smooth vector fields on $\mathcal{M}$
ν_x, ξ_x, η_x	tangent vectors in $T_x\mathcal{M}$
$\gamma(t)$	curve on a manifold
$\mathfrak{F}_x(\mathcal{M})$	set of all real-valued functions on $\mathcal{M}$ smooth at x
g_x	Riemannian metric evaluated at x
$(\mathcal{M}, g)$	manifold $\mathcal{M}$ equipped with metric g
F	smooth mapping between two manifolds
$\mathrm{D}F$	differential of F
$\mathcal{M}_1 \simeq \mathcal{M}_2$	diffeomorphic manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$
∇	affine connection, the Levi–Civita connection
$\nabla_\nu \eta$	the connection of η w.r.t. ν
$[\nu, \eta]$	Lie bracket between ν and η
Exp_x	the exponential mapping in x based on geodesics
exp	the exponential mapping based on one-parameter subgroups

Lie groups

$\mathcal{G}$	Lie group
δ	left group action
δ_x	$g \mapsto \delta(g, x)$
$\mathcal{M}/\mathcal{G}$	orbit space
$\mathcal{H}$	closed Lie subgroup
$\mathcal{G}/\mathcal{H}$	left coset space of $\mathcal{G}$ modulo $\mathcal{H}$
Stab_x	stabilizer group
$\mathbf{GL}^n$	the general linear group $:= \mathbf{R}^{n\times n}_*$
$\mathbf{O}^n$	the orthogonal group $:= \{X \in \mathbf{R}^{n\times n} \mid X^T X = I_n\}$

Embedded submanifolds

$\overline{\mathcal{M}}$	total space, i.e., the embedding space
$\overline{g}_x$	Riemannian metric of $\overline{\mathcal{M}}$
$N_x\mathcal{M}$	normal space of $\mathcal{M}$ at x
$\mathrm{P}^{\mathrm{t}}_x$	orthogonal projector onto $T_x\mathcal{M}$
$\mathrm{P}^{\mathrm{n}}_x$	orthogonal projector onto $N_x\mathcal{M}$
$g^{\mathrm{E}}(Z_1, Z_1)$	Euclidean metric $:= \mathrm{tr}(Z_1^T Z_2)$

Quotient manifolds

$\sim$	equivalence relation
$[\,x\,]$	equivalence class of x
$\mathcal{M}/\sim$	the quotient of $\mathcal{M}$ by $\sim$
π	quotient map, canonical projection
$\overline{\mathcal{M}}$	total space
$\overline{x}$	representative of x $:= \pi^{-1}(x)$
$\mathcal{V}_{\overline{x}}$	vertical space in $\overline{x}$
$\mathcal{H}_{\overline{x}}$	horizontal space in $\overline{x}$
$\overline{\xi}$	horizontal lift of ξ
$\mathrm{P}^{\mathrm{v}}_x$	orthogonal projector onto $\mathcal{V}_x$
$\mathrm{P}^{\mathrm{h}}_x$	orthogonal projector onto $\mathcal{H}_x$

Optimization on manifolds

$\mathrm{grad}\, f(x)$	Riemannian gradient of f at x
$\mathrm{Hess}\, f(x)[\xi]$	Riemannian Hessian of f at x evaluated for ξ
m_x	model function for f at x
R_x	retraction mapping
ρ	Trust-Region performance
Δ	Trust-Region radius

Manifold $\mathbf{S}^{n,p}_{+}$

x	element of $\mathbf{S}^{n,p}_{+}$ as an abstract element
Y	matrix in $\mathbf{R}^{n\times p}_{*}$ such that $x = YY^T \in \mathbf{S}^{n}_{++}$

List of Algorithms

1 Introduction

1.1 Rank-constrained optimization

The topic of this thesis is the numerical solution of certain rank-constrained matrix problems. Let $f : \mathbf{R}^{n\times n} \to \mathbf{R}$ be a smooth objective function defined on $\mathbf{R}^{n\times n}$, the set of n-by-n real matrices. Then these problems are of the form

$$\min f(x) \quad \text{subject to } x \in \mathbf{S}_+^{n,p}, \tag{1.1}$$

where the constraint set $\mathbf{S}_+^{n,p}$ is the set of all n-by-n symmetric and positive semidefinite matrices of rank p.

Rank-constrained matrix problems arise in many different fields, and, in general, they are considered difficult. The rank constraint can originate because of very different reasons. In this thesis, we use $\mathbf{S}_+^{n,p}$ as a tool to reduce the number of parameters that are needed to store certain very large matrices. This is accomplished by approximating these large matrices by matrices of much lower rank. Specifically, we will solve an equation, the Lyapunov matrix equation, by minimizing a particular objective function such that x is a fixed rank matrix in $\mathbf{S}_+^{n,p}$.

Suppose now that we have defined a suitable objective function f. We then need to decide on the numerical method to solve problem (1.1). It may be tempting to use standard techniques for constrained optimization to solve (1.1), but the presence of the low-rank constraint makes this very difficult. We address this problem by exploiting the fact that $\mathbf{S}_+^{n,p}$ is a *smooth manifold*.

As an introduction to a more rigorous treatment later in the thesis, let us first see how the geometry of $\mathbf{S}_+^{n,p}$ *looks* like in the case of 2-by-2 matrices. The set of 2-by-2 symmetric and positive semidefinite matrices, denoted by $\mathbf{S}_+^2$, can be parameterized with three variables $(x, y, z) \in \mathbf{R}^3$ as follows:

$$X = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \in \mathbf{S}_+^2 \iff x \geq 0,\ z \geq 0,\ xz \geq y^2.$$

These relations define an implicit function for the boundary of $\mathbf{S}_+^2$. A part of this boundary is drawn in Figure 1.1, where we indicated three special subsets:

1) $\mathbf{S}_+^{2,2}$; the interior of the cone,
2) $\mathbf{S}_+^{2,1}$; the part of the boundary consisting of rank one matrices, and
3) $\mathbf{S}_+^{2,0}$; the zero matrix.

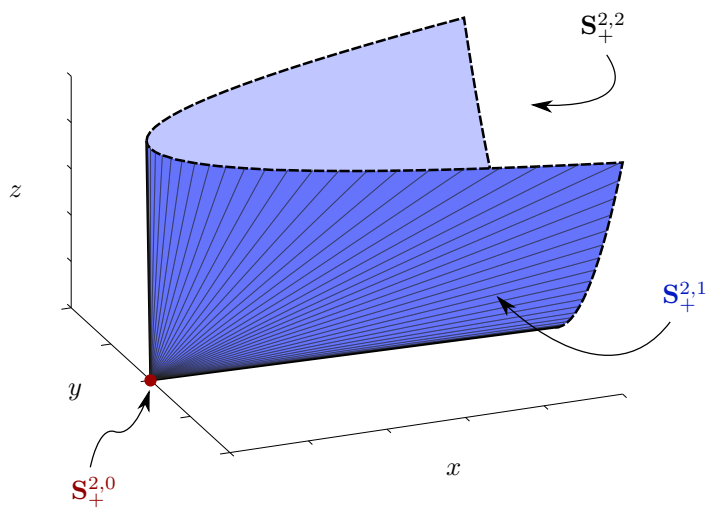


Figure 1.1: The smooth manifolds of $\mathbf{S}_+^2$.

Observe that the zero matrix is not included in the set $\mathbf{S}_+^{2,1}$ since $\mathbf{S}_+^{2,1}$ contains only matrices of rank one. Then it is clear from the figure that $\mathbf{S}_+^{2,1}$ is smooth as a surface in $\mathbf{R}^3$. Hence, we call $\mathbf{S}_+^{2,1}$ a *smooth manifold*. In fact, we will later show that all sets $\mathbf{S}_+^{n,p}$ are smooth manifolds, for all positive integers $p \leq n$.

Now that we have established that the constrained set $\mathbf{S}_+^{n,p}$ in (1.1) possesses a smooth structure, we will exploit this smoothness in the numerical methods for minimizing f. In particular, our optimization algorithms will be based on the framework of *Riemannian optimization*.

1.2 Riemannian optimization

Riemannian optimization is the generalization of standard Euclidean optimization methods to smooth manifolds. The Riemannian algorithms abandon the flat, Euclidean space and formulate problem (1.1) directly on the curved manifold $\mathbf{S}_+^{n,p}$ instead. As a result, one can eliminate the low-rank constraint and get an unconstrained optimization problem that, by construction, will only use feasible points. Although optimizing on a smooth manifold is more complicated than optimizing on a flat space, there are general techniques available; see Absil *et al.* (2008) for an overview in case of matrix manifolds.

In this thesis, we will use methods from the recent retraction-based framework of Riemannian optimization. The principle behind this retraction-based methods can be summarized by aid of Figure 1.2.

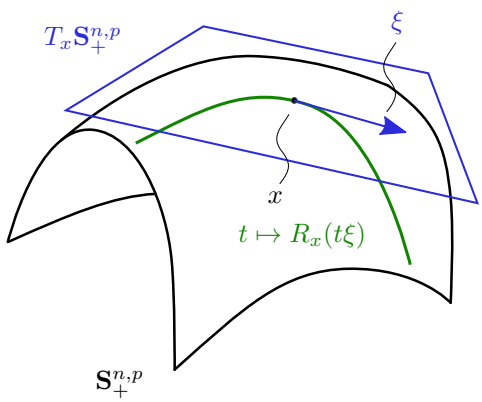


Figure 1.2: Retraction-based optimization.

Suppose the numerical method arrived at the iterate x on the manifold $\mathbf{S}_+^{n,p}$. In this iterate, we can construct a linear space, tangent to the manifold, which is called the tangent space and is denoted by $T_x\mathbf{S}_+^{n,p}$. The direction of a search for a better iterate will always be a vector ξ that lies in this tangent space. The actual search is performed along a curve that lies in $\mathbf{S}_+^{n,p}$. This line search $t \mapsto R_x(t\xi)$ on the manifold is encoded by the mapping

$$R_x : T_x\mathbf{S}_+^{n,p} \to \mathbf{S}_+^{n,p},$$

called the *retraction mapping*. After one step, the iteration repeats itself by taking a different step in the tangent space of the new iterate. Eventually, one obtains the following iteration.

1: **for** $i = 1, 2, \ldots$ **do**
2: Determine a search direction $\xi \in T_x\mathbf{S}_+^{n,p}$.
3: Perform a line-search along $t \mapsto R_x(t\xi)$.
4: Determine t_* such that $f(R_x(t_*\xi))$ makes sufficient progress.
5: Obtain the new iterate $x \leftarrow R_x(t_*\xi)$.
6: **end for**

The iteration above can be understood as an optimization algorithm from the line-search framework. In this thesis, we prefer optimization methods from the framework of Trust-Region. Applied to smooth manifolds, this is the Riemannian Trust-Region (RTR) method of Absil *et al.* (2007): a matrix-free and globally convergent, second-order method suitable for large-scale optimization on Riemannian manifolds.

Simply put, the RTR method is a generalization of the classic unconstrained Trust-Region (TR) method to Riemannian manifolds. Each iteration consists of two phases: first, approximating the solution of the so-called Trust-Region subproblem, followed by the computation of a new iterate. The algorithm follows the same principles as the previous algorithm.

1: **for** $i = 1, 2, \ldots$ **do**
2: Approximately minimize the Trust-Region subproblem

$$\min_{\xi \in T_x\mathbf{S}_+^{n,p}, \|\xi\| \leq \Delta} m_x(\xi) := f(x) + g_x(f(x), \xi) + \frac{1}{2} g_x(\operatorname{Hess} f(x)[\xi], \xi)$$

3: Construct the new trial iterate $\hat{x} = R_x(\xi)$.
4: Update the iterate by rejecting or accepting $\hat{x}$ depending on its quality.
5: Update the Trust-Region radius Δ.
6: **end for**

The algorithm computes a series of approximations $x \in \mathbf{S}_+^{n,p}$ by using a series of second-order models $m_x : T_x\mathbf{S}_+^{n,p} \to \mathbf{R}$ associated with every x. These models are each defined on the tangent space $T_x\mathbf{S}_+^{n,p}$ at x and are based on the *Riemannian gradient* and *Riemannian Hessian*. They are evaluated by means of the Riemannian metric, denoted by $g_x(\cdot, \cdot)$.

The Riemannian algorithms, like the ones from above, are formulated by means of concepts from differential geometry. These objects are the tangent space, the metric, the Riemannian gradient and the Riemannian Hessian, to name a few. Since we will perform optimization on $\mathbf{S}_+^{n,p}$, we need to derive these objects specifically for $\mathbf{S}_+^{n,p}$. This requires studying the Riemannian geometry of the sets $\mathbf{S}_+^{n,p}$ in more detail.

1.3 Riemannian geometry

Let $\mathbf{R}^{n\times n}$ denote the set of $n \times n$ real matrices, and let

$$\mathbf{S}^n := \{X \in \mathbf{R}^{n\times n} \mid X = X^T\}$$

denote the symmetric matrices. Then

$$\mathbf{S}^n_+ := \{X \in \mathbf{S}^n \mid X \succeq 0\},$$

is the set of *symmetric positive semidefinite (s.p.s.d.) matrices*.

We will mainly study $\mathbf{S}^n_+$ from the point of differential geometry. We make a clear distinction between its interior and its boundary. The interior of $\mathbf{S}^n_+$ is the set of *symmetric positive definite (s.p.d.) matrices*, denoted by

$$\mathbf{S}^n_{++} := \mathbf{S}^{n,n}_+ = \{X \in \mathbf{S}^n \mid X \succ 0\}.$$

The boundary $\mathbf{S}^n_+ \backslash \mathbf{S}^n_{++}$ consists of all the rank-deficient matrices in $\mathbf{S}^n_+$. In particular, we denote the set of *fixed-rank matrices* of rank p in this boundary by

$$\mathbf{S}^{n,p}_+ := \{X \in \mathbf{S}^n_+ \mid \operatorname{rank}(X) = p\}.$$

Clearly, one has $\mathbf{S}^n_+ \backslash \mathbf{S}^n_{++} = \cup_{p=0}^{n-1} \mathbf{S}^{n,p}_+$.

It has been well known for a long time that $\mathbf{S}^n_{++}$ is a smooth manifold; see, e.g., Koecher (1957) and Skovgaard (1984). Virtually all the applications that exploit the geometry of $\mathbf{S}^n_{++}$ use the property that this Riemannian structure is a reductive homogeneous space (Nomizu, 1954). The geometry that is derived from this structure is termed the *natural geometry*, since it is indeed natural to choose. It is quite exceptional, however, that a particular manifold has such a natural choice for its Riemannian geometry. If there is such a choice, there is usually no reason to look for other geometries.

In this thesis, we are not concerned with the geometry of $\mathbf{S}^n_{++}$ but with the geometry of subsets of the boundary of $\mathbf{S}^n_+$, or more precisely, the geometry of $\mathbf{S}^{n,p}_+$. Contrary to the established, natural geometry of $\mathbf{S}^n_{++}$, there is no longer such a natural choice for the geometry of $\mathbf{S}^{n,p}_+$. We will therefore propose two possible geometries: $\mathbf{S}^{n,p}_+$ as an embedded submanifold and $\mathbf{S}^{n,p}_+$ as a homogeneous space. We will show that each has its own merits but also its disadvantages. The embedded geometry is primarily used for optimization, while the homogenous space geometry has mathematically more appealing properties.

1.4 Other approaches for matrix problems with rank constraints

Apart from methods that compute low-rank solutions for a specific type of rank-constrained problem, there are not many existing methods available for general problems. We briefly outline four main approaches.

One of these approaches is to parametrize $x = YY^T$, with Y an $n \times p$ matrix, and minimize $f(YY^T)$ for Y. This has been used successfully in the context of low-rank SDP solvers (Burer & Monteiro, 2003). While this effectively lowers the dimension of the search space, it suffers from non-local minimizers which can cause problems for second-order methods like Newton's method. Indeed, every $Z = YQ$, with Q an orthogonal matrix of size p, is also a minimizer. Furthermore, this parameterization adds an unnecessary nonlinearity to the problem, which makes the Newton method difficult to precondition.

In Absil *et al.* (2009b) and Journée *et al.* (2010), the authors also apply Riemannian optimization to optimize over manifolds which are closely related to $\mathbf{S}_+^{n,p}$. However, instead of our proposed embedded geometry, they optimize over a quotient space. This approach is in fact closely related to optimizing on $\mathbf{S}_+^{n,p}$ since these two manifolds are diffeomorphic. We will come back to these manifolds in Chapter 6.6.

Another method, which has, due to its geometrical foundations, more in common to our approach, is the dynamical low-rank approximation of Koch & Lubich (2007). This method consists of evolving a low-rank matrix on the manifold of matrices of fixed rank, for which the authors derive a set of ODEs for the factors of this low-rank approximation. By numerically integrating this set of ODEs, one can dynamically update the low-rank matrix and, for example, approximately solve a Lyapunov matrix equation.

What is different to our method is that we can (approximately) solve the matrix equations directly by an optimization algorithm on the manifold. The application of Koch & Lubich (2007), on the other hand, requires the integration to steady-state. Since large-scale systems involving PDEs are usually very stiff, this necessitates implicit time stepping, but now on the manifold $\mathbf{S}_+^{n,p}$. Hence, one still needs to solve equations on a manifold.

The minimization of the rank of a matrix is a nonsmooth and nonconvex objective. In Fazel (2002), it has been shown that the trace norm (the sum of the eigenvalues) is the tightest convex relaxation of the rank function on $\mathbf{S}_+^n$. This can be generalized to arbitrary matrices by means of the nuclear norm, i.e., the sum of the singular values (Recht *et al.*, 2010). Since the minimization of this nuclear norm over a convex set can be formulated as an SDP, one can obtain the global optimum in polynomial complexity. One can even show that certain problems, like matrix

completion, can be solved in an almost optimal way by this relaxation (Candès & Tao, 2009).

However, the usual off-the-shelf SDP solvers can not be used for large-scale problems since they scale typically as $O(n^6)$. While in some cases, the interior-point algorithm can be formulated to exploit the low-rank structure, their complexity still behaves like $O(n^4)$ (Liu & Vandenberghe, 2009). There exist faster methods, suitable for large-scale problems, that minimize specific nuclear norm relaxed problems, see, e.g., Cai *et al.* (2010); Goldfarb & Ma (2010), but they are only of first-order and do not solve the original (relaxed) problem.

1.5 Main research goals and contributions

The major contribution of this thesis is the introduction of a new geometric framework for computing low-rank approximations to solutions of matrix equations. The method is based on optimizing an objective function on $\mathbf{S}_+^{n,p}$, the Riemannian manifold of symmetric positive semidefinite matrices of fixed rank.

A central argument for introducing a low-rank structure is that it can often be exploited in an algorithm to reduce the cubic (or worse) complexity to $O(np^c)$, with c small. This way, large-scale matrix problems involving positive semidefinite matrices become feasible. In this thesis, we will derive such scalable algorithms in the framework of Riemannian optimization. By formulating the approximation problems on the Riemannian manifold $\mathbf{S}_+^{n,p}$, we will show that this leads to efficient and efficient algorithms.

In order to apply Riemannian optimization techniques on $\mathbf{S}_+^{n,p}$, we study its geometry in detail. These results are kept general in the expectation that the geometric framework can be applied also to other matrix equations (e.g., the Riccati equation), and to similar matrix manifolds (e.g., the set of non-symmetric matrices of fixed rank). Since the aim is the development of scalable algorithms for large-scale applications, we will devote a significant amount of attention to finding efficient expressions for the objects from differential geometry. These objects include, e.g., the tangent space, the geodesics and the retractions.

For assessing the efficiency of our Riemannian approach, we focus on the stable generalized Lyapunov equation and show that the Riemannian algorithms can lead to an efficient and scalable solver which is competitive with the state-of-the-art low-rank solvers. Furthermore, we propose a multilevel Riemannian optimization algorithm that can exploit the multilevel nature of discretizations of partial differential equations. Applied to the Lyapunov equation, the numerical experiments indicate that this approach shows the two key properties of multigrid

methods: mesh-independent convergence and a cost per iteration that scales linearly with the problem size.

A major part of this thesis is devoted to a systematic study of the Riemannian geometry of $\mathbf{S}_+^{n,p}$. In particular, we will derive two different geometries for $\mathbf{S}_+^{n,p}$. Although low-rank approximations for Lyapunov equations is our primary application area, our geometric study is broader than is strictly needed for Riemannian optimization. Due to the relevance of the Riemannian geometry of $\mathbf{S}_{++}^n$, it is sensible that some of this geometrical analysis can be generalized to the fixed-rank case of $\mathbf{S}_+^{n,p}$. This necessitates a well-thought geometry for $\mathbf{S}_+^{n,p}$. To the best of our knowledge, this has only recently started to appear in the literature; see Bonnabel & Sepulchre (2009, 2010); Journée *et al.* (2010); Meyer *et al.* (2010).

Furthermore, we will derive and analyze the geodesics for these two geometries. The reason for deriving geodesics is that they are of central importance in many analyses regarding Riemannian manifolds. In light of the extension of the geometrical analysis of $\mathbf{S}_+^n$ to the fixed-rank case, it seems that geodesics will be indispensable, certainly for deriving theoretical results.

1.6 Outline of the thesis

We gave in this Chapter 1 an overview of our Riemannian framework for solving rank-constrained problems. After having introduced the smooth manifold $\mathbf{S}_+^{n,p}$, we sketched the basic idea of the Riemannian optimization algorithm.

We start in Chapter 2 with a general introduction to Riemannian geometry and optimization on manifolds. All of this chapter is well known in differential geometry, although the content and style of presentation were specifically chosen to prepare for the later chapters.

Chapter 3 introduces an embedded geometry for the submanifold $\mathbf{S}_+^{n,p}$. Most of this chapter was published in Vandereycken & Vandewalle (2010). The derivation of the geodesics appeared in Vandereycken *et al.* (2009). The new contributions consist of a constructive proof for the embeddedness of $\mathbf{S}_+^{n,p}$ (Section 3.2.4) and the comparison with the orthographic retraction (Sections 3.5.3 and 3.5.4).

Chapter 4 is the application of Riemannian optimization and the embedded geometry to solve low-rank solutions for Lyapunov equations. The results concerning the minimization of the energy norm are mostly based on Vandereycken & Vandewalle (2010). The new additions are all the derivations concerning the residual norm (for the most part, Sections 4.4.2 and 4.5.2).

In Chapter 5, we outline an extension and a convergence analysis of a tensor-product multigrid method for Lyapunov equations. The convergence results of this method

were published in Vandereycken & Vandewalle (2009). The new contributions consist of the Riemannian multilevel algorithm for low-rank approximations (Section 5.3 and 5.4).

Chapter 6 deals with a homogeneous space geometry for $\mathbf{S}_+^{n,p}$. In contrast to the embedded geometry, the homogeneous space was constructed such that the metric space is complete. This material is currently submitted and available as Vandereycken *et al.* (2010).

Finally, Chapter 7 formulates the conclusions of this thesis, summarizes the main contributions and indicates possible future research. Two appendices concerning Lie groups (Appendix A) and linear algebra (Appendix B) end the thesis.

2

Riemannian geometry

This chapter reviews some of the basic concepts in differential geometry and lays down the notational conventions for the remainder of the thesis. The concepts in this chapter consist of mostly basic facts about Riemannian manifolds and we present them without proof. In addition, we introduce the concept of optimization on manifolds by a method with superlinear convergence. For a more mathematically convincing exposition, we refer to the classic introductory works to Riemannian geometry like Boothby (1986) and Lee (2003); for the optimization perspective, we refer to Absil *et al.* (2008).

2.1 Introduction

Consider Figure 1.1 again that represents $\mathbf{S}_{+}^{2,1}$. We claimed that this subset was a smooth manifold since it *looks* like a smooth surface. How can this smoothness be proved rigorously? Before even doing this, we first need to know what a smooth manifold is, or more specifically, what a *Riemannian manifold* is. In this chapter, we review the necessary definitions, theorems and properties. The abstract version of Figure 1.1 is depicted in Figure 2.1. When appropriate, we illustrate properties based on this abstract manifold. Furthermore, all definitions and properties of this chapter will be needed somewhere later in the thesis.

In addition, we introduce the general principle of retraction-based Riemannian optimization for the optimization of an objective function on a manifold. We explain

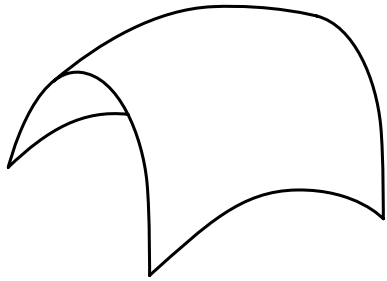


Figure 2.1: An abstract smooth manifold $\mathcal{M}$.

how second-order models can be derived using the earlier introduced geometric objects. These models can then be optimized by a robust version of Newton's method, namely the Riemannian Trust-Region method from Absil *et al.* (2007).

2.2 Riemannian manifolds

Manifolds are usually intuitively associated with surfaces. However, in terms of Riemannian geometry, they have well-defined definitions and properties.

2.2.1 Smooth manifolds

A smooth manifold of dimension d is intuitively defined as a set $\mathcal{M}$ which locally looks like a d-dimensional Euclidean space but which can be very different globally. Since optimization generally requires taking derivatives and gradients of functions, we will need to perform calculus on $\mathcal{M}$. Hence we require a smooth structure on $\mathcal{M}$ that allows us to do this. Throughout this thesis, smooth will always be synonymous for C^∞, i.e., differentiable up to all degrees.

How does this intuitive definition translate into a more rigorous mathematical language? The notion of "locally Euclidean" implies that every point of $\mathcal{M}$ has a neighborhood which is homeomorphic[1] to an open subset of $\mathbf{R}^d$. This identification is done by charts where a *chart* is a bijection from a subset $\mathcal{U} \subset \mathcal{M}$ onto a subset of $\mathbf{R}^d$. Such a chart is visible as mapping ϕ_1 (or ϕ_2) in Figure 2.2. Every chart is defined only locally and in order to get a global description of $\mathcal{M}$, one assembles all charts into an *atlas*.

Definition 2.1 (Lee (2003, Lemma 1.23))**.** A smooth atlas on a topological space $\mathcal{M}$ is a countable collection of pairs $\{(\mathcal{U}_i, \phi_i)\}$ that has the following properties.

[1] A homeomorphism is a bijection which is continuous and has a continuous inverse.

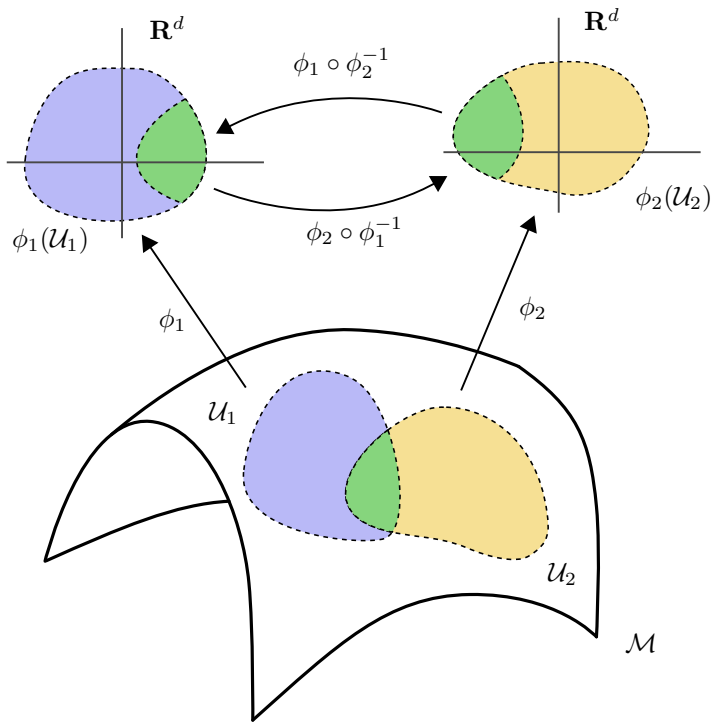


Figure 2.2: Charts on a manifold.

1. The open sets $\mathcal{U}_i$ cover $\mathcal{M}$, i.e.,

   $$\mathcal{M} \subseteq \bigcup_i \mathcal{U}_i.$$

2. For each i, the chart

   $$\phi_i : \mathcal{U}_i \to \mathbf{R}^d$$

   is a homeomorphism with $\phi_i(\mathcal{U}_i)$ an open subset of $\mathbf{R}^d$.

3. For any pair i, j with $\mathcal{U}_i \cap \mathcal{U}_j \neq 0$, the sets $\phi_i(\mathcal{U}_i \cap \mathcal{U}_j)$ and $\phi_j(\mathcal{U}_i \cap \mathcal{U}_j)$ are open in $\mathbf{R}^d$ and the mapping

   $$\phi_j \circ \phi_i^{-1} : \mathbf{R}^d \to \mathbf{R}^d$$

   is smooth on $\mathcal{U}_i \cap \mathcal{U}_j$.

This definition is illustrated in Figure 2.2. Observe that the charts ϕ_1 and ϕ_2 have to overlap smoothly on their common domains $\mathcal{U}_1 \cap \mathcal{U}_2$. The smoothness of $\mathcal{M}$ is expressed in terms of smoothness of the mappings $\phi_j \circ \phi_i^{-1}$ for all i, j.

One can think of a smooth manifold as the pair $(\mathcal{M}, \mathcal{A})$ with $\mathcal{A}$ such a smooth atlas for $\mathcal{M}$. However, we refrain from giving a formal definition of a smooth manifold as this would require some further topological issues[2]. Since these technicalities do not add much to the remainder of the thesis, this intuition is sufficient and we refer to the works mentioned in the beginning of this section for a precise definition.

A manifold is called *path-connected*, or, in short, *connected*, if any two points can be joined by a piecewise smooth curve. This means that the manifold cannot be expressed as the disjoint union of two nonempty open sets.

In some cases making an atlas is trivial since the chart is global.

Example 2.2. The Euclidean space of $n \times p$ matrices, $\mathbf{R}^{n \times p}$, is naturally identified with $\mathbf{R}^{np}$. The vectorization operator $\text{vec} : \mathbf{R}^{n \times p} \to \mathbf{R}^{np}$ of (B.5), which stacks the columns of a matrix on top of each other, from left to right, defines such an identification. Since the domain of this chart is the whole manifold $\mathbf{R}^{n \times p}$, the atlas contains only one chart.

The previous example is of course a trivial one. Although it is possible to rigorously construct an atlas for some nontrivial manifolds, like the n-sphere, defining a manifold by means of an atlas is usually too cumbersome. Most of the time, one constructs manifolds in an indirect way. We will explain two common constructions, that of embedded submanifolds in Section 2.3 and that of quotient manifolds in Section 2.4. In Chapters 3 and 6, we will show that $\mathbf{S}_+^{n,p}$ can be constructed in both ways.

2.2.2 Tangent space

The single most important property of a manifold is the *tangent space*: in each point $x \in \mathcal{M}$, there is a linear space, denoted $T_x\mathcal{M}$, of which the elements are called *tangent vectors*. If one thinks of $\mathcal{M}$ as a surface in a Euclidean space $\mathbf{R}^n$, then the tangent space is the usual tangent plane at x. Although this characterization is correct, it relies too much on the specific structure of $\mathbf{R}^n$ to be applicable for general manifolds. A more abstract definition can be based on the differentiation of curves.

Let $\gamma(t)$ be a *smooth curve* in $\mathcal{M}$; it is a smooth[3] mapping

$$\gamma : \mathbf{R} \to \mathcal{M}, \ t \mapsto \gamma(t).$$

Let $f : \mathcal{M} \to \mathbf{R}$ be a function, smooth around $x \in \mathcal{M}$. The set of all such real-valued functions is denoted by $\mathfrak{F}_x(\mathcal{M})$. Since $f \circ \gamma$ is a smooth function from

[2] In particular, we require the atlas to be maximal and the manifold to have a Hausdorff and second-countable topology.

[3] See Section 2.2.3 for the concept of smoothness of mappings.

$\mathbf{R}$ to $\mathbf{R}$, the classical derivatives are well defined. A tangent vector can now be defined in the following way; see, e.g., Absil *et al.* (2008, Def. 3.5.1).

Definition 2.3. Let γ be a smooth curve in $\mathcal{M}$ with $\gamma(0) = x$. Mapping

$$\nu_x : \mathfrak{F}_x \to \mathbf{R},\ f \mapsto \nu_x f := \left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0},$$

is called the tangent vector in x to the curve γ at $t = 0$.

The curve γ in the previous definition is said to *realize* the tangent vector ν_x. There are infinitely many curves that realize a given tangent vector. This does, however, not invalidate Def. 2.3, since an alternative way of defining tangent vectors is by the identification as the equivalence class of such realizing curves; see Lee (2003, p. 77)

Let the dimension of $\mathcal{M}$ be d. Then the tangent space $T_x\mathcal{M}$ has the structure of a d-dimensional linear space: given $a, b \in \mathbf{R}$ and $\nu_x, \eta_x \in T_x\mathcal{M}$, then $a\nu_x + b\eta_x$ defined as

$$(a\nu_x + b\eta_x)f := a(\nu_x f) + b(\eta_x f), \quad \text{for all } f \in \mathfrak{F}_x(\mathcal{M}),$$

is again a tangent vector in $T_x\mathcal{M}$ since it satisfies Def. 2.3.

The *tangent bundle* $T\mathcal{M}$ is defined as the union of the tangent spaces at all elements of $\mathcal{M}$:

$$T\mathcal{M} := \bigcup\nolimits_{x\in\mathcal{M}} T_x\mathcal{M}.$$

A smooth *vector field* is a smooth mapping $\nu : \mathcal{M} \to T\mathcal{M}$ that assigns to each point $x \in \mathcal{M}$ a tangent vector $\nu_x \in T_x\mathcal{M}$, i.e.,

$$\nu :\ \mathcal{M} \to T\mathcal{M},\ x \mapsto \nu_x \in T_x\mathcal{M}.$$

The set of all smooth vector fields on $\mathcal{M}$ is denoted by $\mathcal{X}(\mathcal{M})$.

The evaluation of a vector field results in a tangent vector, i.e., $\nu(x) = \nu_x \in T_x\mathcal{M}$. This distinction between tangent vectors and vector fields is emphasized by the subscript $_x$. Since every tangent vector can be smoothly extended to a smooth vector field, we sometimes drop this subscript when there is no confusion possible and simply denote a tangent vector by ν.

The application of a vector field $\nu \in \mathcal{X}(M)$ to a function f on $\mathcal{M}$ is denoted by νf. Evaluation at $x \in \mathcal{M}$ results in $(\nu f)(x) := \nu_x f$, as defined in Definition 2.3.

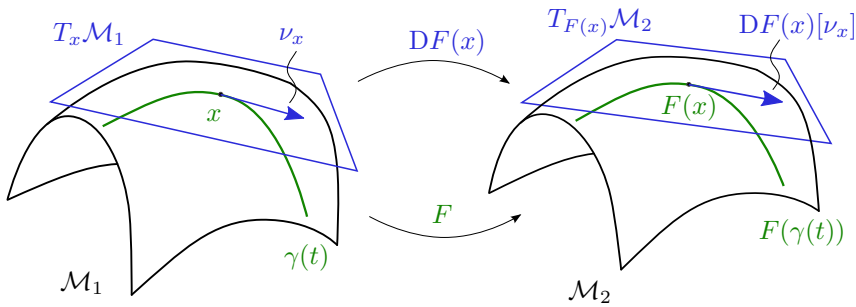


Figure 2.3: The differential $\mathrm{D}F(x)$ of F at x.

2.2.3 Mappings between manifolds

Let $F : \mathcal{M}_1 \to \mathcal{M}_2$ be a mapping between two manifolds of dimension d_1 and d_2 respectively. Since there is a specific terminology with respect to mappings between manifolds, we define some of them here.

Smoothness. The smoothness of F in $x \in \mathcal{M}_1$ can be interpreted as the usual C^∞ smoothness of the partial derivatives of the Euclidean function

$$\widehat{F} := \phi_2 \circ F \circ \phi_1^{-1} : \mathbf{R}^{d_1} \to \mathbf{R}^{d_2},$$

where ϕ_1 and ϕ_2 are any pair of charts around $x \in \mathcal{M}_1$ and $F(x) \in \mathcal{M}_2$ respectively. We call F smooth, if it is smooth on its whole domain. Unless stated otherwise, we will always assume that a mapping is smooth.

Differential. The differential[4] of a mapping $F : \mathcal{M}_1 \to \mathcal{M}_2$ at any element $x \in \mathcal{M}_1$ is the generalization of the Fréchet derivative for vector spaces (see App. B.2.1). It is a linear map between tangent spaces, denoted as

$$\mathrm{D}F(x) : \ T_x\mathcal{M}_1 \to T_{F(x)}\mathcal{M}_2, \ \nu_x \mapsto \mathrm{D}F(x)[\nu_x].$$

To define $\mathrm{D}F(x)[\nu_x]$, we use the interpretation in Def. 2.3 of a tangent vector as a mapping $\mathfrak{F}_{F(x)}(\mathcal{M}_2) \to \mathbf{R}$. This gives the definition

$$(\mathrm{D}F(x)[\nu_x])f := \nu_x(f \circ F), \quad \text{for all } f \in \mathfrak{F}_{F(x)}(\mathcal{M}_2).$$

From a more geometric viewpoint, we get that $F(\gamma(t))$ is a curve that realizes $\mathrm{D}F(x)[\nu_x]$ for any curve $\gamma(t)$ that realizes ν_x; see Fig. 2.3.

[4] also called the push-forward, the (total) derivative and the tangent map

Chain rule The differential satisfies a similar version of the chain rule (B.10) for the Fréchet differential. Let $F : \mathcal{M}_1 \to \mathcal{M}_2$ and $G : \mathcal{M}_2 \to \mathcal{M}_3$ be two mappings, then (Lee, 2003, Lemma 3.5)

$$\mathrm{D}(G \circ F)(x) : \; T_x\mathcal{M}_1 \to T_{G(F(x))}\mathcal{M}_3, \; \nu_x \mapsto \mathrm{D}G(\, F(x)\,)[\,\mathrm{D}F(x)[\nu_x]\,].$$

Immersion, submersion and diffeomorphism. The *rank* of a mapping F at $x \in \mathcal{M}_1$ is the dimension of the range of the differential $\mathrm{D}F(x)$. If F has the same rank at every point, we call F a mapping of *constant rank*.

The following are three important mappings of constant rank.

Definition 2.4. An immersion is a smooth map $F : \mathcal{M}_1 \to \mathcal{M}_2$ with $\dim(\mathcal{M}_1) \leq \dim(\mathcal{M}_2)$ which is of constant rank $\dim(\mathcal{M}_1)$.

Definition 2.5. A submersion is a smooth map $F : \mathcal{M}_1 \to \mathcal{M}_2$ with $\dim(\mathcal{M}_1) \geq \dim(\mathcal{M}_2)$ which is of constant rank $\dim(\mathcal{M}_2)$.

Definition 2.6. A diffeomorphism is a smooth map $F : \mathcal{M}_1 \to \mathcal{M}_2$ with $\dim(\mathcal{M}_1) = \dim(\mathcal{M}_2)$ which is of constant rank $\dim(\mathcal{M}_1)$.

By the definition of the differential $DF(x) : T_x\mathcal{M}_1 \to T_{F(x)}\mathcal{M}_2$, the mappings from above have an alternative definition.

Definition 2.7. An immersion is a smooth map $F : \mathcal{M}_1 \to \mathcal{M}_2$ for which $DF(x)$ is injective at each $x \in \mathcal{M}_1$.

Definition 2.8. A submersion is a smooth map $F : \mathcal{M}_1 \to \mathcal{M}_2$ for which $DF(x)$ is surjective at each $x \in \mathcal{M}_1$.

Definition 2.9. A diffeomorphism is a smooth map $F : \mathcal{M}_1 \to \mathcal{M}_2$ for which $DF(x)$ is bijective at each $x \in \mathcal{M}_1$.

The following theorem in Lee (2003, Thm. 7.14) is a powerful characterization of constant rank functions.

Theorem 2.10. *Let $F : \mathcal{M}_1 \to \mathcal{M}_2$ be a smooth map of constant rank.*

(a) If F is surjective, then it is a submersion.

(b) If F is injective, then it is an immersion.

(c) If F is bijective, then it is a diffeomorphism.

Diffeomorphisms express that manifolds are essentially the same from a differential geometric viewpoint. Two manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ are said to be *diffeomorphic*, denoted by $\mathcal{M}_1 \simeq \mathcal{M}_2$, if there exists a diffeomorphism between them.

2.3 Embedded submanifolds

Embedded submanifolds coincide with the intuitive interpretation of a manifold as a surface in Euclidean space; take, e.g., a sphere in three-dimensional space. However, not all smooth subsets are necessarily submanifolds that are compatible with the differential structure of their embedding space.

Suppose we have a subset $\mathcal{N}$ of a smooth manifold $\mathcal{M}$. If the inclusion map

$$i : \mathcal{N} \to \mathcal{M}, \quad x \mapsto x$$

is an immersion, we call $\mathcal{N}$ an *immersed submanifold* of $\mathcal{M}$.

Even when $\mathcal{N}$ is an immersed submanifold of $\mathcal{M}$, their topologies can be different. Usually the embedding space $\mathcal{M}$ is some well-known manifold; in fact, in this thesis it will always be (a subset of) a Euclidean space. Hence, we would like to base the differential properties of $\mathcal{N}$ on those of $\mathcal{M}$. This requires that the topology of $\mathcal{N}$ coincides with the subspace topology[5] of $\mathcal{M}$. Submanifold $\mathcal{N}$ is then called an *embedded submanifold*. If such a topology exists, then it is unique.

Theorem 2.11 (Absil *et al.* (2008, Prop. 3.3.1))**.** *Let $\mathcal{N}$ be a subset of a smooth manifold $\mathcal{M}$. Then $\mathcal{N}$ admits at most one differentiable structure that makes it an embedded submanifold of $\mathcal{M}$.*

Embedded submanifolds are the natural setting for calculus on submanifolds since the differentiable structure is compatible with the one of the embedding space. Unless stated otherwise, we will always assume that a submanifold is an embedded submanifold. We already mentioned, that in our case $\mathcal{M}$ will always be (a subset of) $\mathbf{R}^{n\times m}$, which makes $\mathcal{N}$ a *matrix submanifold*.

A submanifold embedded in $\mathcal{M}$ is called *closed*, if it is closed as a set in the subspace topology of $\mathcal{M}$.[6]

2.3.1 Submanifolds as level sets

Instead of proving that an inclusion map is a topological embedding, it is more common to construct an embedded submanifold as the level set of a submersion. A *level set* at a point $y \in \mathcal{M}_2$ of a mapping $F : \mathcal{M}_1 \to \mathcal{M}_2$ is defined as the inverse image of y under F; it is denoted as

$$F^{-1}(y) = \{x \in \mathcal{M}_1 \mid F(x) = y\}.$$

[5] The subspace topology on $\mathcal{N}$ is defined as the coarsest topology for which i is continuous. Hence $\mathcal{N}$ is homeomorphic with $i(\mathcal{N})$.

[6] In the literature the term closed manifold is sometimes used for a compact manifold without boundary. We do not use this terminology.

Theorem 2.12 (Lee (2003, Cor. 8.9)). *Let $F : \mathcal{M}_1 \to \mathcal{M}_2$ be a submersion. Then each level set of F is a closed embedded submanifold in $\mathcal{M}_1$ of dimension* $\dim(\mathcal{M}_1) - \dim(\mathcal{M}_2)$.

This theorem can be weakened in the sense that we do not need to have a constant rank mapping on the whole domain of F. It is sufficient that the rank equals $\dim(\mathcal{M}_2)$ at all points $x \in F^{-1}(y)$ for some $y \in \mathcal{M}_2$. We call such a y a *regular value*.

Theorem 2.13 (Absil *et al.* (2008, Prop. 3.3.3)). *Let $F : \mathcal{M}_1 \to \mathcal{M}_2$ be a smooth mapping and let $y \in \mathcal{M}_2$ be a regular value. Then $F^{-1}(y)$ is a closed embedded submanifold in $\mathcal{M}_1$ of dimension* $\dim(\mathcal{M}_1) - \dim(\mathcal{M}_2)$.

Example 2.14. Let $p \leq n$ be two positive numbers. The set of all orthonormal matrices,

$$\mathrm{St}^{n,p} = \{X \in \mathbf{R}_*^{n\times p} \mid X^T X = I_p\},$$

is called the Stiefel manifold. It is an embedded submanifold of $\mathbf{R}^{n\times p}$ since it coincides with the level set $F^{-1}(0)$ of the mapping $F : \mathbf{R}^{n\times p} \to \mathbf{S}^p$, $X^T X - I_p$. One can show that F is indeed a submersion; see, e.g., Absil *et al.* (2008, Sec. 3.3.2). When $n = p$, it reduces to the orthogonal group $\mathbf{O}^n \equiv \mathrm{St}^{n,n}$.

The previous theorem is useful in proving that certain sets are embedded submanifolds, like the example above. However, it is not always easy or even possible to find only one function for which the whole submanifold is the level set. Luckily, this theorem can be relaxed to a description using local submersions.

Theorem 2.15 (Lee (2003, Prop. 8.12)). *Let $\mathcal{N}$ be a subset of a smooth manifold $\mathcal{M}$ of dimension n. Then $\mathcal{N}$ is an embedded submanifold in $\mathcal{M}$ of dimension k if and only if every point $p \in \mathcal{N}$ has a neighborhood $\mathcal{U}_p$ in $\mathcal{M}$ such that $\mathcal{U}_p \cap \mathcal{N}$ is a level set of a submersion $F_p : \mathcal{U}_p \to \mathbf{R}^{n-k}$.*

2.3.2 Tangent space of a submanifold

For submanifolds that are embedded in a vector space $\mathbf{R}^n$, the tangent space as defined in Definition 2.3 has a particular straightforward interpretation. Let ν_x be a tangent vector in $x \in \mathcal{M}$ and let $\gamma(t) : \mathbf{R} \to \mathcal{M}$ be a curve that realizes ν_x. Since the curve γ is embedded in $\mathbf{R}^n$, the usual derivative with respect to t of this curve is well defined, i.e.,

$$\dot{\gamma}(0) := \left.\frac{\mathrm{d}\gamma(t)}{\mathrm{d}t}\right|_{t=0} = \lim_{t\to 0} \frac{\gamma(t) - \gamma(0)}{t} \in \mathbf{R}^n.$$

The tangent space $T_x\mathcal{M}$ can now be alternatively defined as

$$T_x\mathcal{M} = \{\, \dot\gamma(0) \mid \gamma(t) \text{ a curve in } \mathcal{M} \text{ with } \gamma(0) = x \,\}.$$

This makes indeed sense according to Def. 2.3. Let f be a smooth function on $\mathcal{M}$, then it can be locally extended to a smooth function on $\mathbf{R}^n$. Now we have with the above defined curve $\gamma(t)$ that

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

Hence ν_x can be naturally identified with $\dot\gamma(0)$, and vice versa; see Figure 2.4.

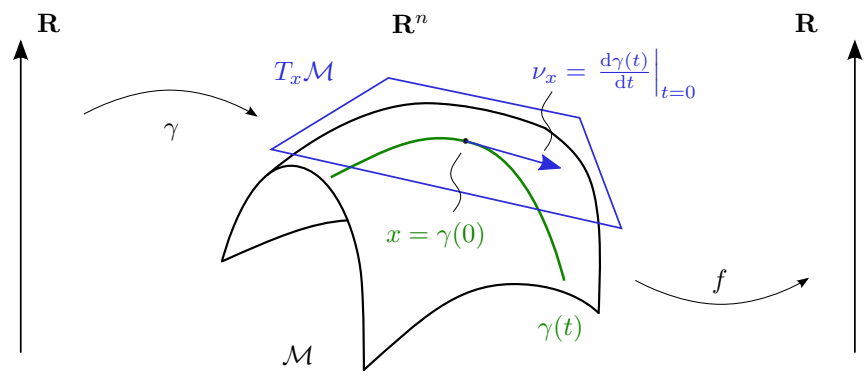


Figure 2.4: Tangent vector ν_x of a submanifold $\mathcal{M}$ embedded in a vector space $\mathbf{R}^n$.

2.4 Quotient manifolds

Quotient manifolds are abstract spaces that arise when we have a manifold where certain subsets are equivalent. This property is expressed in terms of an *equivalence relation*; see Def. A.19.

Suppose $\mathcal{M}$ is a manifold—in our case, usually a vector space $\mathbf{R}^{n\times p}$—that is equipped with an equivalence relation $\sim$. If we collect all elements on $\mathcal{M}$ that are equivalent with an $x \in \mathcal{M}$, we obtain the *equivalence class* of x, denoted by

$$[\,x\,] := \{\, y \in \mathcal{M} \mid y \sim x \,\}.$$

One can regard such an equivalence class as a subset in $\mathcal{M}$, called a *fiber*. These fibers are visualized in Fig. 2.5. Since all elements of $\mathcal{M}$ belonging to the same fiber are equivalent, it is natural to regard them as the same element in some new

space. This space, called the *quotient* of $\mathcal{M}$ by $\sim$, is the set of all the equivalence classes

$$\mathcal{M}/\sim := \{\,[x] \mid x \in \mathcal{M}\,\}.$$

It is a topological space, endowed with the quotient topology[7] for the mapping

$$\pi : \mathcal{M} \to \mathcal{M}/\sim,\ x \mapsto [x].$$

This map π is called the *quotient map* or *canonical projection*. This construction is shown in Fig. 2.5.

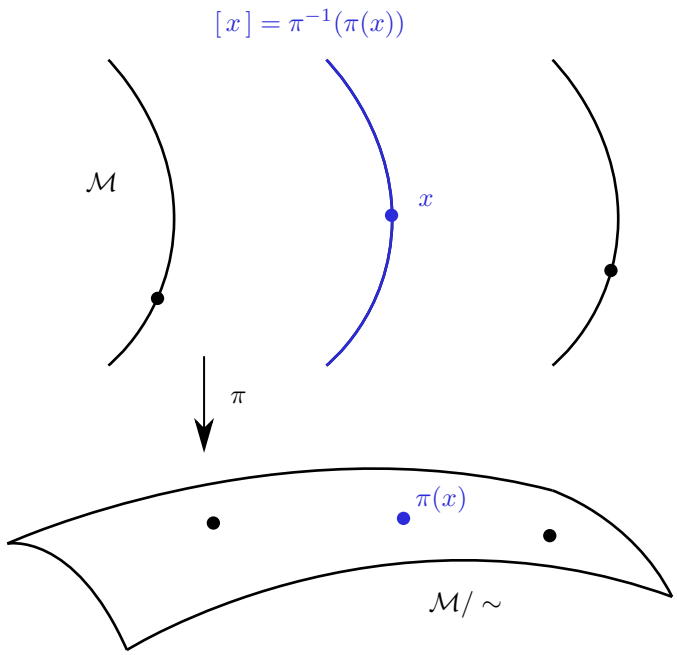


Figure 2.5: A quotient space $\mathcal{M}/\sim$ and its fibers $\pi^{-1}(\pi(x))$.

In the same way that subsets are not always smooth embedded submanifolds, quotient spaces are not always smooth quotient manifolds. In particular, the canonical projection π has to be a submersion. If this is the case, then there is only one differentiable structure that makes $\mathcal{M}/\sim$ a smooth quotient manifold.

Theorem 2.16 (Absil *et al.* (2008, Prop. 3.4.1))**.** *Let $\mathcal{M}$ be a smooth manifold and let $\mathcal{M}/\sim$ be a quotient of $\mathcal{M}$. Then $\mathcal{M}/\sim$ admits at most one differentiable structure that makes it a quotient manifold of $\mathcal{M}$.*

[7] The quotient topology on $\mathcal{M}/\sim$ is defined as the finest topology for which π is continuous.

Again similar as with submanifolds, one usually does not prove that a quotient space is a smooth quotient manifold based on the topology of π. Instead, one relies on properties of certain mappings. We will explain the most widely used: smooth actions of a Lie group.

2.4.1 Quotient manifolds by Lie group actions

One of the best-known examples of quotient manifolds arise in the context of actions by Lie groups. We refer to App. A.1 for a brief introduction into the necessary concepts of Lie groups. We will assume only matrix Lie groups where the product rule coincides with the usual matrix multiplication.

Definition 2.17. A (left) action of a Lie group $\mathcal{G}$ on a manifold $\mathcal{M}$ is a smooth map $\delta : \mathcal{G} \times \mathcal{M} \to \mathcal{M}$ satisfying

(a) $\delta(e, x) = x$, for all $x \in \mathcal{M}$ with e the identity element of $\mathcal{G}$.

(b) $\delta(g_1, \delta(g_2, x)) = \delta(g_1 g_2, x)$, for all $g_1, g_2 \in \mathcal{G}$ and $x \in \mathcal{M}$.

Actions can satisfy some additional properties. They are called

(a) *proper*: if the graph of δ is a proper map[8];

(b) *free*: if for all $x \in \mathcal{M}$, $\delta(g, x) = x$ implies $g = e$, with e the identity element of $\mathcal{G}$;

(c) *transitive*: if for any $x_1, x_2 \in \mathcal{M}$, there exists a $g \in \mathcal{G}$ such that $\delta(g, x_1) = x_2$.

Lie group actions express an equivalence relation on $\mathcal{M}$ in the following sense. Consider the action δ, and define for some $x \in \mathcal{M}$ the mapping

$$\delta_x : \mathcal{G} \to \mathcal{M},\ p \mapsto \delta_x(p) := \delta(p, x).$$

Then the equivalence relation

$$x_1 \sim x_2 \quad \Longleftrightarrow \quad x_2 = \delta_{x_1}(g) \text{ for some } g \in \mathcal{G}$$

induces the equivalent class

$$[\, x \,] = \delta_x(\mathcal{G}) := \operatorname{ran}(\delta_x).$$

[8] A map $f : A \to B$ is called proper if and only if the preimage of every compact set in B is compact in A.

In the context of Lie groups, the equivalent class $[\,x\,]$ is called the *orbit* through x. The quotient $\mathcal{M}/\sim$, i.e., the collection of all orbits, is termed the *orbit space*, and is denoted by

$$\mathcal{M}/\mathcal{G} := \mathcal{M}/\sim = \{\,\delta_x(\mathcal{G}) \mid x \in \mathcal{M}\,\}$$

In general the orbit space is not a smooth quotient manifold. However, the *Quotient Manifold Theorem* states that under certain conditions on the action, the orbit space is always a smooth quotient manifold.

Theorem 2.18 (Lee (2003, Thm. 9.16))**.** *Suppose a Lie group $\mathcal{G}$ acts smoothly, freely and properly on a smooth manifold $\mathcal{M}$. Then the orbit space $\mathcal{M}/\mathcal{G}$ is a quotient manifold of $\mathcal{M}$ with dimension* $\dim(\mathcal{M}) - \dim(\mathcal{G})$*. Furthermore, the canonical projection $\pi : \mathcal{M} \to \mathcal{M}/\mathcal{G}$ is a submersion.*

2.4.2 Homogeneous spaces as quotient manifolds

When the orbit space from above is constructed as the quotient of a Lie group with a closed subgroup, it is a so-called *homogeneous space*.

Definition 2.19. Let $\mathcal{M}$ be a smooth manifold endowed with an action $\delta : \mathcal{G} \times \mathcal{M} \to \mathcal{M}$ of the group $\mathcal{G}$. If the action δ is transitive, we call $\mathcal{M}$ a homogeneous space.

First, we describe a general construction to obtain such a homogenous space as an orbit space. Let $\mathcal{G}$ be a Lie group and $\mathcal{H} \subset \mathcal{G}$ a closed Lie subgroup. Define the equivalence relation on $\mathcal{G}$ as

$$g_1 \sim g_2 \quad \Longleftrightarrow \quad g_1 = g_2 h \text{ for some } h \in \mathcal{H},$$

for any $g_1, g_2 \in \mathcal{G}$. The equivalence class of $g \in \mathcal{G}$

$$g\mathcal{H} := [\,g\,] = \{gh \mid h \in \mathcal{H}\}$$

is now termed the *left coset* of g modulo $\mathcal{H}$. In addition, the quotient of $\mathcal{G}$ by $\sim$ is named the *left coset space* of $\mathcal{G}$ modulo $\mathcal{H}$, and is denoted by

$$\mathcal{G}/\mathcal{H} := \mathcal{G}/\sim .$$

The quotient space from above is always a smooth homogeneous manifold as stated in the *Homogeneous Space Construction Theorem*.

Theorem 2.20 (Lee (2003, Thm. 9.18))**.** *Let $\mathcal{G}$ be a Lie group and let $\mathcal{H}$ be a closed Lie subgroup of $\mathcal{G}$. Then the left coset space $\mathcal{G}/\mathcal{H}$ is a quotient manifold of $\mathcal{G}$ with*

dimension $\dim(\mathcal{G}) - \dim(\mathcal{H})$. *Furthermore, the canonical projection* $\pi : \mathcal{G} \to \mathcal{G}/\mathcal{H}$ *is a submersion and the left action*

$$\delta : \mathcal{G} \times \mathcal{G}/\mathcal{H} \to \mathcal{G}/\mathcal{H},\ (g_1, g_2\mathcal{H}) \mapsto g_1(g_2\mathcal{H}) = (g_1 g_2)\mathcal{H}$$

is transitive.

A closed subgroup $\mathcal{H}$ of $\mathcal{G}$ is any group which is closed as a set in the subspace topology of $\mathcal{G}$. In other words, $\mathcal{H}$ contains all its limit points.

The quotient manifold $\mathcal{G}/\mathcal{H}$ turns out be of central importance since *any* homogeneous space with transitive action δ is diffeomorphic to the quotient of $\mathcal{G}$ with some closed Lie subgroup $\mathcal{H} \subset \mathcal{G}$. This subgroup is given by the so-called *stabilizer* of the action. It is defined as the subgroup of $\mathcal{G}$ that leaves the action fixed:

$$\mathrm{Stab}_x := \{g \in \mathcal{G} \mid \delta_x(g) = \delta(g, x) = x\}. \tag{2.1}$$

Theorem 2.21 (Lee (2003, Thm. 9.20))**.** *Let* $\mathcal{M}$ *be a homogeneous manifold with transitive action* $\delta : \mathcal{G} \times \mathcal{M} \to \mathcal{M}$*, and let* Stab_x *be the stabilizer for any* $x \in \mathcal{M}$*. Then the map*

$$\Delta_x : \mathcal{G}/\mathrm{Stab}_x \to \mathcal{M},\ [\,g\,] = g\mathrm{Stab}_x \mapsto \delta_x(g) = \delta(x, g)$$

is a diffeomorphism.

Example 2.22 (Lee (2003, Ex. 9.25))**.** Let $\mathrm{Gr}^{n,p}$ be the Grassmann manifold of p-dimensional subspaces in $\mathbf{R}^n$. Let $U \in \mathbf{R}_*^{n\times p}$ be a basis for a subspace in $\mathrm{Gr}^{n,p}$, and similarly for V. Then $\mathbf{GL}^n$ acts transitively on the left as $U = AV$ with $A \in \mathbf{GL}^n$. The stabilizer in U is given by

$$\mathrm{Stab}_U = \begin{bmatrix} \mathbf{GL}^p & \mathbf{R}^{p\times(n-p)} \\ 0_{(n-p)\times p} & \mathbf{GL}^{n-p} \end{bmatrix},$$

which is indeed a closed Lie subgroup of $\mathbf{GL}^n$. Hence, $\mathrm{Gr}^{n,p} \simeq \mathbf{GL}^n/\mathrm{Stab}_U$.

2.4.3 Tangent space of a quotient manifold

Similar to a submanifold of a Euclidean space, the tangent vectors of a quotient manifold of a Euclidean space admit a concrete representation. This representation is, however, somewhat more involved.

Let $\mathcal{M} := \overline{\mathcal{M}}/\sim$ be a quotient manifold of a Euclidean space $\overline{\mathcal{M}}$. We call $\overline{\mathcal{M}}$ the *total space* of $\mathcal{M}$. Since $\pi : \overline{\mathcal{M}} \to \mathcal{M}$ is a submersion, every fiber $\pi^{-1}(x)$ of $x \in \mathcal{M}$

is an embedded submanifold of $\overline{\mathcal{M}}$ as a level set. Let $\overline{x} \in \pi^{-1}(x)$ be an element of this fiber. We call the tangent space of the fiber at $\overline{x}$ the *vertical space*, i.e.,

$$\mathcal{V}_{\overline{x}} := T_{\overline{x}}(\pi^{-1}(x)).$$

In addition, define $\mathcal{H}_{\overline{x}}$, termed the *horizontal space* at $\overline{x}$, as the complementary subspace of $\mathcal{V}_{\overline{x}}$ in $T_{\overline{x}}\overline{\mathcal{M}}$. In other words,

$$\mathcal{V}_{\overline{x}} \oplus \mathcal{H}_{\overline{x}} = T_{\overline{x}}\overline{\mathcal{M}},$$

like in Figure 2.6.

Since we assumed that $\overline{\mathcal{M}}$ is Euclidean, we have that $T_{\overline{x}}\overline{\mathcal{M}} \simeq \overline{\mathcal{M}}$ consists of concrete vectors (or matrices). Furthermore, since $\dim(\mathcal{H}_{\overline{x}}) = \dim(T_x\mathcal{M})$, it is possible to represent a tangent vector $\nu_x \in T_x\mathcal{M}$ by a concrete vector in $\overline{\nu}_{\overline{x}} \in \mathcal{H}_{\overline{x}}$ in the following way. Suppose that the vectors ν_x and $\overline{\nu}_{\overline{x}}$ are related by the canonical projection π, i.e.,

$$\mathrm{D}\pi(\overline{x})[\overline{\nu}_{\overline{x}}] = \nu_x, \tag{2.2}$$

then $\overline{\nu}_{\overline{x}}$ is called the unique *horizontal lift* of the tangent vector ν_x. For a concrete manifold, the relation (2.2) will have to be proved explicitly.

2.5 Riemannian metric

Suppose we equip every tangent space with an inner product $g_x(\cdot,\cdot)$, i.e., a bilinear, symmetric and positive-definite form

$$g_x(\nu_x, \eta_x), \quad \text{for all } \nu_x, \eta_x \in T_x\mathcal{M}.$$

If this form varies smoothly over the tangent bundle, then g defines a *Riemannian metric*. This turns $\mathcal{M}$ into a *Riemannian manifold*. We denote this pairing as $(\mathcal{M}, g)$.

The *length of a curve* $\gamma : [0,1] \to \mathcal{M}$ on a Riemannian manifold $(\mathcal{M}, g)$ is defined by

$$L(\gamma) := \int_0^1 \sqrt{g(\dot{\gamma}(t), \dot{\gamma}(t))}\mathrm{d}t.$$

The *Riemannian distance* on a connected manifold $(\mathcal{M}, g)$ is defined as

$$\mathrm{dist} : \mathcal{M} \times \mathcal{M} \to \mathbf{R},\ \mathrm{dist}(x_1, x_2) = \inf_{\Gamma(x_1,x_2)} L(\gamma), \tag{2.3}$$

where $\Gamma(x_1, x_2)$ is the set of all piecewise continuous curves γ in $\mathcal{M}$ that join $\gamma(0) = x_1$ and $\gamma(1) = x_2$. This infimum does not need to be attained by any

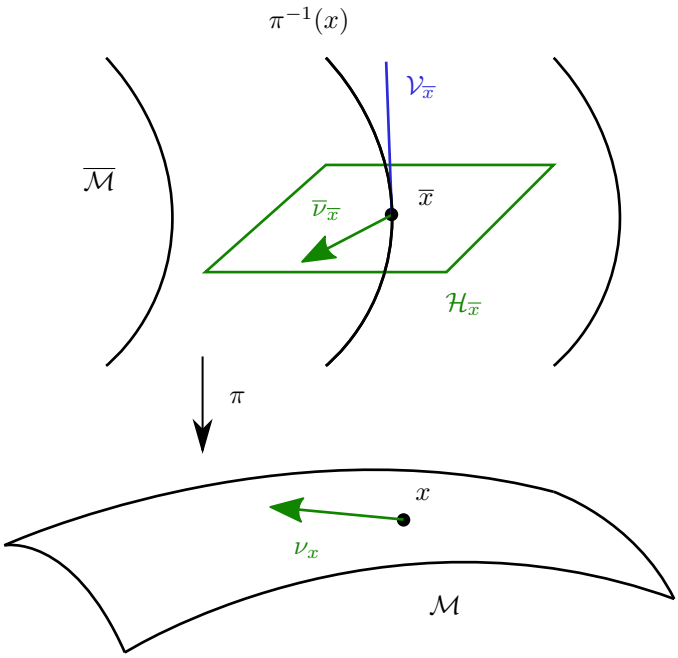


Figure 2.6: Tangent vector ν_x and its horizontal lift $\overline{\nu}_{\overline{x}}$.

specific curve and even when it does, i.e, the infimum is a minimum, this curve does not need to be unique.

This Riemannian distance defines a *metric* in the classical meaning: a bilinear form that is positive definite, symmetric and satisfies the triangle inequality (Lee, 1997, Lemma 6.2). Since every Riemannian metric g on a connected manifold $\mathcal{M}$ induces a metric (2.3), one can unambiguously speak of the *metric space* $(\mathcal{M}, g)$.

The metric space $(\mathcal{M}, g)$ is called *complete* if the limit of every Cauchy sequence in $\mathcal{M}$ convergences w.r.t. the metric (2.3) to an element of $\mathcal{M}$.

Let dist and $\widetilde{\text{dist}}$ be the metrics (2.3) on $(\mathcal{M}, g)$ and $(\widetilde{\mathcal{M}}, \widetilde{g})$, respectively. A diffeomorphism $F : (\mathcal{M}, g) \to (\widetilde{\mathcal{M}}, \widetilde{g})$ is called an *isometry* if it preserves distances:

$$\text{dist}(x_1, x_2) = \widetilde{\text{dist}}(F(x_1), F(x_2)), \quad \text{for all } x_1, x_2 \in \mathcal{M}.$$

2.5.1 Riemannian submanifolds

Let $\mathcal{M}$ be an embedded submanifold of a Riemannian manifold $(\overline{\mathcal{M}}, \overline{g})$. Since the tangent space $T_x\mathcal{M}$ is embedded in $T_x\overline{\mathcal{M}}$ as a subspace, tangent vectors of $T_x\mathcal{M}$ can be regarded as elements of $T_x\overline{\mathcal{M}}$ also. By restricting the metric of $\overline{g}$ to $T_x\mathcal{M}$, we get a metric g on $\mathcal{M}$:

$$g_x(\nu, \eta) := \overline{g}_x(\nu, \eta), \quad \text{for all } \nu, \eta \in T_x\mathcal{M}.$$

This turns $(\mathcal{M}, g)$ into a *Riemannian submanifold* of $(\overline{\mathcal{M}}, \overline{g})$.

The orthogonal complement of $T_x\mathcal{M}$ is called the *normal space* of $x \in \mathcal{M}$, denoted by

$$N_x\mathcal{M} := \{\, \nu \in T_x\overline{\mathcal{M}} \mid \overline{g}_x(\nu, \eta) = 0 \text{ for all } \eta \in T_x\mathcal{M} \,\}.$$

Obviously, $T_x\mathcal{M} \oplus N_x\mathcal{M} = T_x\overline{\mathcal{M}}$, which allows to define the orthogonal projectors (w.r.t. $\overline{g}$) on and along these spaces:

$$\mathrm{P}^{\mathrm{t}}_{\overline{x}} : T_x\overline{\mathcal{M}} \to T_x\mathcal{M} \tag{2.4}$$

$$\mathrm{P}^{\mathrm{n}}_{\overline{x}} : T_x\overline{\mathcal{M}} \to N_x\mathcal{M} \tag{2.5}$$

with $\overline{g}_x(\mathrm{P}^{\mathrm{t}}_{\overline{x}}(\nu), \mathrm{P}^{\mathrm{n}}_{\overline{x}}(\nu)) = 0$ for all $\nu \in T_x\overline{\mathcal{M}}$.

2.5.2 Riemannian quotient manifolds

Let $\mathcal{M}$ be a quotient manifold of a Riemannian manifold $(\overline{\mathcal{M}}, \overline{g})$ with canonical projection $\pi : \overline{\mathcal{M}} \to \mathcal{M}$. Let $\pi(\overline{x}) = x$. Recall from Section 2.4.3 that the tangent space $T_x\mathcal{M}$ can be represented by lifted vectors in the horizontal space $\mathcal{H}_{\overline{x}}\overline{\mathcal{M}}$ if these vectors are π-related, i.e., if they satisfy relation (2.2).

One can now define a metric g on $\mathcal{M}$ by restricting the metric $\overline{g}$ to the horizontal space and demanding that this expression does not depend on the specific choice of $\overline{x}$. In other words,

$$g_x(\nu, \eta) := \overline{g}_{\overline{x}}(\overline{\nu}, \overline{\eta}), \quad \text{for all } \nu, \eta \in T_x\mathcal{M},$$

is constant for any horizontal lift $\overline{\nu} \in \mathcal{H}_{\overline{x}}\overline{\mathcal{M}}$ of $\nu \in T_x\mathcal{M}$ in a fixed x (and likewise for $\overline{\eta}$). This turns $(\mathcal{M}, g)$ into a *Riemannian quotient manifold* of $(\overline{\mathcal{M}}, \overline{g})$.

Observe that the mapping $\mathrm{D}\pi|_{\mathcal{H}_{\overline{x}}} \to T_x\mathcal{M}$ is an isometry since it is a bijection that preserves the inner product of tangent vectors normal to the fibers. In the language of O'Neill (1966), this is called a *Riemannian submersion*.

Obviously, $\mathcal{H}_{\overline{x}}\mathcal{M} \oplus \mathcal{V}_{\overline{x}}\mathcal{M} = T_x\overline{\mathcal{M}}$, which allows to define the orthogonal projectors (w.r.t. $\overline{g}$) on and along these spaces:

$$\mathrm{P}_x^{\mathrm{h}} : T_{\overline{x}}\overline{\mathcal{M}} \to \mathcal{H}_{\overline{x}}\mathcal{M} \tag{2.6}$$

$$\mathrm{P}_x^{\mathrm{v}} : T_{\overline{x}}\overline{\mathcal{M}} \to \mathcal{V}_{\overline{x}}\mathcal{M} \tag{2.7}$$

with $\overline{g}_{\overline{x}}(\mathrm{P}^{\mathrm{h}}_{\overline{x}}(\nu), \mathrm{P}^{\mathrm{v}}_{\overline{x}}(\nu)) = 0$ for all $\nu \in T_{\overline{x}}\overline{\mathcal{M}}$.

2.6 Levi–Civita connection

Many algorithms in optimization require second-order information. In general, this second-order information is obtained by taking the derivative of one vector field with respect to another. The classical Newton iteration in Euclidean space, for example, can be formulated in terms of solving a specific directional derivative of the gradient.

In Euclidean space, taking the derivative of one vector field along another one, i.e.,

$$\mathrm{D}\eta(x)[\nu_x] = \lim_{t\to 0} \frac{\eta(x + t\nu_x) - \eta(x)}{t}, \tag{2.8}$$

always results in a vector field. On a general manifold $\mathcal{M}$, however, this is not the case: for vector fields η, ν on $\mathcal{M}$, equation (2.8) does not need to be a vector field on $\mathcal{M}$ (even if all the operations in the expression of the limit are well defined). Therefore, the principle of taking derivatives of vector fields on manifolds is generalized to the so-called affine connection.

Recall that $\mathfrak{F}_x(\mathcal{M})$ is the set of all smooth functions in $x \in \mathcal{M}$ and that $\mathcal{X}(\mathcal{M})$ is the set of all smooth vector fields on $\mathcal{M}$. Then the *affine connection* is a smooth mapping, denoted by

$$\nabla : \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \to \mathcal{X}(\mathcal{M}),\ (\nu, \eta) \mapsto \nabla_\nu \eta.$$

that satisfies the following properties:

(a) $\nabla_{f\eta}\nu = f\nabla_\eta \nu$

(b) $\nabla_\eta(a\nu + b\xi) = a\nabla_\eta\nu + b\nabla_\eta\xi$

(c) $\nabla_\eta(f\nu) = (\eta f)\nu + f\nabla_\eta\nu$

for all $f \in \mathfrak{F}_x(\mathcal{M})$, $a, b \in \mathbf{R}$ and $\eta, \nu, \xi \in \mathcal{X}(\mathcal{M})$. Remark that ηf is the application of the vector field η to the function f, see also Definition 2.3.

For any smooth manifold, there are an infinite number of affine connections, but there is a specific connection which is always unique. It is called the Levi–Civita connection, and unless stated otherwise, we will always assume the connection to be the Levi–Civita connection.

Before we can characterize this Levi–Civita connection, we need the notion of a Lie bracket between two vector fields $\nu, \eta \in \mathcal{X}(\mathcal{M})$. In general, $\nu\eta$ does not result in a vector field, nor does $\eta\nu$. However, $\eta\nu - \nu\eta$ is always a vector field on $\mathcal{M}$ (Lee, 2003, Lemma 4.12). This operation is called the *Lie bracket* and is denoted as

$$[\eta, \nu] := \eta\nu - \nu\eta.$$

Theorem 2.23 (Lee (1997, Thm. 5.4))**.** *Let $(\mathcal{M}, g)$ be a Riemannian manifold. There exists a unique affine connection ∇ that is*

(a) symmetric: $\nabla_\eta \nu - \nabla_\nu \eta = [\eta, \nu]$,

(b) compatible with the Riemannian metric: $\xi g(\eta, \nu) = g(\nabla_\xi \eta, \nu) + g(\eta, \nabla_\xi \nu)$,

for all $\eta, \nu, \xi \in \mathcal{X}(\mathcal{M})$. This connection is called the Levi–Civita connection.

The Levi–Civita connection can be characterized as the unique symmetric and affine connection that satisfies *Koszul's formula* (Lee, 1997, eq. (5.1)):

$$\begin{aligned} 2g(\nabla_\xi \eta, \nu) = \xi g(\eta, \nu) + \eta g(\nu, \xi) - \nu g(\xi, \eta) \\ - g(\xi, [\eta, \nu]) + g(\eta, [\nu, \xi]) + g(\nu, [\xi, \eta]). \end{aligned} \tag{2.9}$$

For general manifolds, deriving the Levi–Civita connection based on (2.9) is not always straightforward. However, in the case of Riemannian submanifolds (Section 2.5.1) and Riemannian quotient manifolds (Section 2.5.2), the metric is based on the metric of the total space $\overline{\mathcal{M}}$. Now, the connection can be derived using the connection $\overline{\nabla}$ on $\overline{\mathcal{M}}$.

Recall that, when the total space $(\overline{\mathcal{M}}, \overline{g})$ is Euclidean (hence, $\overline{g}_x$ is independent of x), the connection $\overline{\nabla}$ admits the simple interpretation (2.8) as the classical directional derivative. Together with the following two propositions, this makes the computation of the connection ∇ straightforward.

Theorem 2.24 (Absil *et al.* (2008, Prop. 5.3.2))**.** *Let $(\mathcal{M}, g)$ be a Riemannian submanifold embedded in a Riemannian manifold $(\overline{\mathcal{M}}, \overline{g})$, and let $\mathrm{P}_x^{\mathrm{t}}$ denote the orthogonal projection (2.4) onto $T_x\mathcal{M}$ for any $x \in \mathcal{M}$. Then the Levi–Civita connection ∇ on $\mathcal{M}$ satisfies*

$$\nabla_\nu \eta(x) = \mathrm{P}_x^{\mathrm{t}}(\overline{\nabla}_\nu \eta(x)),$$

for all vector fields $\eta, \nu \in \mathcal{X}(\mathcal{M})$ with $\overline{\nabla}$ the Levi–Civita connection on $\overline{\mathcal{M}}$.

Theorem 2.25 (Absil *et al.* (2008, Prop. 5.3.3))**.** *Let $(\mathcal{M}, g)$ be a Riemannian quotient manifold of a Riemannian manifold $(\overline{\mathcal{M}}, \overline{g})$, and let $\mathrm{P}^{\mathrm{h}}_{\overline{x}}$ denote the orthogonal projection (2.6) onto the horizontal space $\mathcal{H}_{\overline{x}}$ for any $\overline{x} \in \overline{\mathcal{M}}$. Then the horizontal lift of the Levi–Civita connection ∇ on $\mathcal{M}$ satisfies*

$$\overline{\nabla_\nu \eta(x)} = \mathrm{P}^{\mathrm{h}}_{\overline{x}}(\overline{\nabla}_{\overline{\nu}} \overline{\eta}(x)),$$

for all vector fields $\eta, \nu \in \mathcal{X}(\mathcal{M})$ with $\overline{\nabla}$ the Levi–Civita connection on $\overline{\mathcal{M}}$.

2.7 Curves on manifolds

We have already seen that curves on a manifold are useful for illustrating and defining geometric concepts. Here, we will explain two types of curves in some more detail. The first is theoretically by far the most important type on a Riemannian manifold: it is the generalization of straight lines in Euclidean space, called geodesics.

In Chapter 4, we will employ curves for a more practical reason, namely, as the generalization of a line-search in classical Euclidean optimization. Although geodesics are the natural candidate for this role, they are usually too costly to compute in practice. Hence, we will use approximations of geodesics, the so-called retractions.

2.7.1 Geodesics

We will assume that the connection ∇ is the Levi–Civita connection.

Definition 2.26. Let $(\mathcal{M}, g)$ be a Riemannian manifold with connection ∇. The parameterized curve $\gamma : (a, b) \to \mathcal{M}$ is called a geodesic if and only if it is a curve of constant velocity, i.e.,

$$\nabla_{\dot\gamma(t)} \dot\gamma(t) = 0, \quad \text{for all } a < t < b. \tag{2.10}$$

The geodesics are usually assumed to be *maximal*, i.e., the interval $a < t < b$ is taken to be the largest such that condition (2.10) is still satisfied. These (maximal) geodesics need not to be defined for all $t \in \mathbf{R}$. If they are, they are called *complete geodesics* and can be extended indefinitely. A geodesic that escapes to infinity in finite time can never be complete.

For any given $\nu_x \in T_x\mathcal{M}$, there always exists a geodesic $\gamma(t)$ such that $\gamma(0) = x$ and $\dot\gamma(0) = \nu_x$. Furthermore, this (maximal) geodesic is unique (Boothby, 1986, Cor. VII.5.6).

If the infimum in (2.3) is attained, then every minimizing curve γ is a geodesic when it is given a unit speed parameterization (Lee, 1997, Thm. 6.6). Such a geodesic is called a *length-minimizing geodesic*. The other way around is only valid locally: every geodesic is locally length-minimizing (Lee, 1997, Thm. 6.12)

Furthermore, there may exist points that can never be connected by a single geodesic. For some manifolds, one can guarantee that these curves do exist and this has far-reaching consequences, as given by the *Hopf–Rinow Theorem*.

Theorem 2.27 (Kobayashi & Nomizu (1963, Thm. 4.1–2))**.** *Let $(\mathcal{M}, g)$ be a connected manifold. Then the following properties are mutually equivalent.*

(a) Any geodesic can be extended indefinitely.

(b) With metric (2.3), $(\mathcal{M}, g)$ is a complete metric space.

(c) The closure of any bounded subset of $\mathcal{M}$ w.r.t. metric (2.3) is compact.

(d) Any two points on $\mathcal{M}$ can be connected by a length-minimizing geodesic.

The length-minimizing property of geodesics gives rise to the following alternative definition.

Definition 2.28 (Postnikov (2001, Prop. 11.1))**.** Let $(\mathcal{M}, g)$ be a Riemannian manifold with connection ∇. The unit speed curve $\gamma : (a, b) \to \mathcal{M}$ is called a geodesic if and only if it is a critical point for the functional

$$S(\gamma) := \int_a^b g(\dot{\gamma}(t), \dot{\gamma}(t))\mathrm{d}t. \tag{2.11}$$

Geodesics are nicely behaved w.r.t. isometries. This is called the *Naturality of the Levi–Civita connection*.

Theorem 2.29 (Lee (1997, Prop. 5.6))**.** *Let $\phi : (\mathcal{M}, g) \to (\widetilde{\mathcal{M}}, \widetilde{g})$ be an isometry. If γ is a geodesic on $\mathcal{M}$, then $\phi \circ \gamma$ is a geodesic on $\widetilde{\mathcal{M}}$.*

2.7.2 The exponential map

Geodesics can be used to define the Exponential mapping (not to be confused with another map that is also called the exponential map but is based on one-parameter subgroups; see, in the context of Lie groups, App. A.3).

Definition 2.30. Let ∇ be the connection on the manifold $\mathcal{M}$ and let $\mathcal{D}_x \subset T_x\mathcal{M}$ be a neighborhood around the zero tangent vector. Then the exponential map at x is defined as

$$\mathrm{Exp}_x : \mathcal{D}_x \to \mathcal{M},\ \xi \mapsto \gamma_x(1),$$

where γ_x is the unique geodesic for ∇ with foot $x = \gamma_x(0)$ and in the direction of $\xi = \dot{\gamma}_p(0)$.

Every $x \in \mathcal{M}$ of a Riemannian manifold $\mathcal{M}$ has a neighborhood $\mathcal{U}_x \subset \mathcal{M}$ which is the diffeomorphic image under Exp_x (Boothby, 1986, Thm. VI.6.6). Hence, $\mathcal{D}_x$ in the definition above is always non-empty, however, is does not need to be the whole tangent space. The domain of Exp_x will be $T_x\mathcal{M}$ for every $x \in \mathcal{M}$ when the geodesics are complete (Boothby, 1986, Remark VII.7.10).

2.7.3 Retractions

As explained before, we will use approximations of geodesics, called *retractions*, to perform line-search on a manifold. In order that the forthcoming algorithms are well-defined and convergent, these approximations will need to satisfy some properties. In specific, their approximation of a geodesic needs to be accurate enough.

Definition 2.31 (Absil *et al.* (2008, Def. 4.1.1))**.** A *first-order retraction* on $\mathcal{M}$ is a mapping $R : T\mathcal{M} \to \mathcal{M}$, smooth around zero, with the following properties. Let R_x be the restriction of R to $T_x\mathcal{M}$, then

(a) $R_x(0) = x$,

(b) Local rigidity: For every tangent vector $\xi \in T_x\mathcal{M}$, the curve $\gamma_\xi : t \mapsto R_x(t\xi)$ realizes ξ in x, in other words, $\dot{\gamma}_\xi(0) = \xi$.

As the presence of "first-order" suggests, these retractions approximate the exponential mapping up to first order. The next step is a *second-order* retraction where the second-order derivative must be interpreted in the sense of the Levi–Civita connection ∇.

Definition 2.32 (Absil *et al.* (2008, Prop. 5.5.5))**.** A *second-order retraction* R_x on $\mathcal{M}$ is a first-order retraction which satisfies in addition the zero initial acceleration condition,

$$\nabla_{\dot{\gamma}(0)}\dot{\gamma}(0) = 0,$$

where $\gamma(t) := R_x(t\xi)$ and ∇ is the connection on $(\mathcal{M}, g)$.

2.8 Retraction-based optimization on manifolds

In the introduction we outlined the general principle of retraction-based optimization on manifolds. In this section, we review the necessary tools and concepts to

make this schematic algorithm well defined. Much of this requires the geometric concepts that we introduced earlier.

2.8.1 The Riemannian Trust-Region method

The Riemannian optimization algorithm that we loosely introduced in Section 1.2 is made specific in Algorithm 1. This algorithm, called Riemannian Trust-Region (RTR) and developed in Absil *et al.* (2007), is the adaptation of the classical Trust-Region (TR) algorithm adapted to Riemannian manifolds. Except for the model definition, which we will explain in the next section, only the step calculation by aid of the retraction R_x is different from a classical Trust-Region method.

Due to the possibly large-scale nature of the Trust-Region subproblems, we minimize them with a truncated conjugate gradient (tCG) method (Toint, 1981; Steihaug, 1983). This tCG method can be preconditioned by a symmetric and positive definite operator if needed.

2.8.2 Second-order models

Gradient-based optimization requires the notion of a gradient as the direction of steepest ascent of an objective function. Newton's method (and its robustified variations by, e.g., Trust-Region) require in addition second-order information, encoded in the Hessian. In case of manifolds, these concepts need to be adapted to the Riemannian setting. In the next two sections, we will review the concept of Riemannian gradient and Riemannian Hessian of a function $f : \mathcal{M} \to \mathbf{R}$ on a Riemannian manifold $(\mathcal{M}, g)$.

In the end, the models can be built using the same principles as the models for Euclidean optimization, namely by means of these Riemannian gradients and Hessians. The result is a second-order model for an objective function f in x:

$$T_x\mathcal{M} \to \mathbf{R},\ \xi \mapsto f(x) + g_x(\operatorname{grad} f(x), \xi) + \frac{1}{2} g_x(\operatorname{Hess} f(x)[\xi], \xi).$$

2.8.3 The Riemannian gradient

Definition 2.33. Let f be a function defined on a manifold $(\mathcal{M}, g)$. The *Riemannian gradient* of f at x, denoted as $\operatorname{grad} f(x)$, is the unique tangent vector in $T_x\mathcal{M}$ satisfying

$$g_x(\operatorname{grad} f(x), \nu) = \mathrm{D}f(x)[\nu], \quad \text{for all } \nu \in T_x\mathcal{M}. \tag{2.13}$$

Algorithm 1 Riemannian Trust-Region (RTR) of Absil *et al.* (2007) with TR strategy from Nocedal & Wright (1999) on a Riemannian manifold $(\mathcal{M}, g)$.

Require: maximal TR radius $\bar{\Delta} > 0$, initial TR radius $\Delta_1 \in (0, \bar{\Delta})$, retraction R_x, objective function f

1: **for** $i = 1, 2, \ldots$ **do**

2: **Model definition**: define the second-order model

$$\widehat{m}_i : T_{x_i}\mathcal{M} \to \mathbf{R},\ \xi \mapsto f(x_i) + g_x(\operatorname{grad} f(x_i), \xi) + \frac{1}{2} g_x(\operatorname{Hess} f(x_i)[\xi], \xi).$$

3: **Step calculation**: solve (approximately) with truncated CG

$$\eta_i = \arg\min \widehat{m}_i(\xi) \quad \text{s.t. } \sqrt{g_x(\xi, \xi)} \leq \Delta_i. \tag{2.12}$$

4: **Acceptance of trial point**: compute

$$\rho_i = \frac{f(R_{x_i}(0)) - f(R_{x_i}(\eta_i))}{\widehat{m}_i(0) - \widehat{m}_i(\eta_i)}.$$

5: **if** $\rho_i \geq 0.05$ **then**

6: Accept step and set $x_{i+1} = R_{x_i}(\eta_i)$.

7: **else**

8: Reject step and set $x_{i+1} = x_i$.

9: **end if**

10: **Trust-Region radius update**: set

$$\Delta_{i+1} = \begin{cases} \min(2\Delta_i, \bar{\Delta}) & \text{if } \rho_i \geq 0.75 \text{ and } \sqrt{g_x(\eta_i, \eta_i)} = \Delta_i, \\ 0.25\sqrt{g_x(\eta_i, \eta_i)} & \text{if } \rho_i \leq 0.25, \\ \Delta_i & \text{otherwise.} \end{cases}$$

11: **end for**

The Riemannian gradient has the well-known interpretation of the direction of steepest ascent, but restricted to variations on $\mathcal{M}$,

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

where $\|\nu\| := \sqrt{g_x(\nu, \nu)}$.

Riemannian submanifolds. Suppose $\mathcal{M}$ is a Riemannian submanifold embedded in a Euclidean space $\overline{\mathcal{M}}$. Then $f : \mathcal{M} \to \mathbf{R}$ can be smoothly extended to a

function $\overline{f} : \overline{\mathcal{M}} \to \mathbf{R}$. Now the Riemannian gradient can be computed based on the Riemannian gradient of $\overline{f}$. We denote this gradient by $\operatorname{grad} \overline{f}(x)$.

Theorem 2.34 (Absil *et al.* (2008, Ch. 3.6))**.** *Let $\mathcal{M}$ be a Riemannian submanifold of a Riemannian manifold $\overline{\mathcal{M}}$. Suppose the function $\overline{f} : \overline{\mathcal{M}} \to \mathbf{R}$ has $\operatorname{grad} \overline{f}(x)$ as Riemannian gradient in $x \in \overline{\mathcal{M}}$. Then the Riemannian gradient of $f : \mathcal{M} \to \mathbf{R}$, $x \mapsto \overline{f}(x)$ satisfies*

$$\operatorname{grad} f(x) = \mathrm{P}_x^{\mathrm{t}}(\operatorname{grad} \overline{f}(x)),$$

with $\mathrm{P}_x^{\mathrm{t}}$ the orthogonal projection (2.4) onto $T_x\mathcal{M}$.

Riemannian quotient manifolds. For a Riemannian quotient manifold $\mathcal{M} := \overline{\mathcal{M}}/\sim$ of a Riemannian manifold $\overline{\mathcal{M}}$, one can compute the gradient based on horizontal lift. Since the fibers are equivalent, this is sensible only for functions $f : \overline{\mathcal{M}} \to \mathbf{R}$ that are constant along the fibers, i.e.,

$$f(x) = f(y), \quad \text{for all } x \sim y.$$

Theorem 2.35 (Absil *et al.* (2008, Ch. 3.6))**.** *Let $\mathcal{M}$ be a Riemannian quotient manifold of a Riemannian manifold $\overline{\mathcal{M}}$ with $\pi : \overline{\mathcal{M}} \to \mathcal{M}$ the canonical projection.. Let $\overline{f} : \overline{\mathcal{M}} \to \mathbf{R}$ be constant along the fibers and let $\operatorname{grad} \overline{f}(\overline{x})$ denote the Riemannian gradient of $\overline{f}$ in $\overline{x} \in \overline{\mathcal{M}}$. Then the horizontal lift of the Riemannian gradient of $f : \mathcal{M} \to \mathbf{R}$, $x \mapsto \overline{f}(\pi^{-1}(x))$ satisfies*

$$\overline{\operatorname{grad} f(x)} = \operatorname{grad} \overline{f}(\overline{x}),$$

for any $\overline{x} = \pi^{-1}(x)$.

2.8.4 The Riemannian Hessian

Definition 2.36. Let f be a function defined on a manifold $(\mathcal{M}, g)$. The *Riemannian Hessian* of f at x in the direction of $\nu \in T_x\mathcal{M}$, denoted as $\operatorname{Hess} f(x)[\nu]$, is the unique symmetric and linear mapping

$$\operatorname{Hess} f(x) : T_x\mathcal{M} \to T_x\mathcal{M}$$

that satisfies

$$\operatorname{Hess} f(x)[\nu] = \nabla_\nu \operatorname{grad} f(x), \tag{2.14}$$

with ∇ the Levi–Civita connection.

The Riemannian Hessian captures again the second-order information and this by also incorporating the curvature of $\mathcal{M}$,

$$g_x(\operatorname{Hess} f(x)[\nu], \nu) = \nabla_{\dot\gamma(0)}\dot\gamma(0), \tag{2.15}$$

where $\gamma(t)$ is a geodesic with $\gamma(0) = x$ and $\dot\gamma(0) = \nu$.

Recall from Theorems 2.24 and 2.25, that for Riemannian submanifolds and quotient manifolds, the connection can be determined from the structure space. This allows us, together with property (2.14), to compute the Hessian for a function f. However, in practice this can become quite complicated due to the presence of the many projections.

Luckily, there is a more constructive approach: for a second-order retraction, the Riemannian Hessian of a function f coincides with the Euclidean Hessian of the lifted function $\widehat{f}_x := f \circ R_x$.

Theorem 2.37 (Absil *et al.* (2008, Prop. 5.5.5))**.** *Let R_x be a second-order retraction on $\mathcal{M}$, then*

$$\operatorname{Hess} f(x) = \operatorname{Hess}(f \circ R_x)(0) \qquad \textit{for all } x \in \mathcal{M}.$$

3

The embedded geometry of symmetric matrices of fixed rank

In this chapter, we describe a specific Riemannian geometry for the set of symmetric positive semidefinite matrices of fixed rank. As is typical in differential geometry, there is seldom one fixed choice for the geometry of a particular manifold. There are, however, certain geometries that are preferable over others.

Here, we focus on the simplest one: an embedding in the space of real matrices, equipped with the Euclidean metric. The rationale for this geometry is its ease of implementation for Riemannian algorithms. Due to its fairly simple embedding, it will be particularly suited for the optimization algorithms in the later chapters. However, from a theoretical point of view, this geometry is not so attractive, since its geodesics are not complete. Later, in Chapter 6, we will introduce another geometry which resolves this issue.

Most of this chapter was published in Vandereycken & Vandewalle (2010). The derivation of the geodesics appeared in Vandereycken *et al.* (2009). The unpublished contributions consist of a constructive proof for the embeddedness of $\mathbf{S}_+^{n,p}$ (Section 3.2.4) and the comparison with the orthographic retraction (Sections 3.5.3 and 3.5.4).

3.1 Introduction

Let $p \leq n$ be two positive integers. The focus of this chapter is the embedded geometry of $\mathbf{S}_+^{n,p}$, the set of all real $n \times n$ symmetric positive semidefinite matrices of rank p.

As explained in the introduction, we will perform retraction-based Riemannian optimization on $\mathbf{S}_+^{n,p}$. To be able to apply Riemannian optimization, we first need to derive some of the geometric objects of the previous chapter for this submanifold. These include the tangent space, the metric, the orthogonal projections and the Levi–Civita connection. Furthermore, we will derive the geodesics and a few retractions.

3.1.1 Notational conventions

Throughout the thesis, we will adhere to a specific notation regarding objects for $\mathbf{S}_+^{n,p}$. While our notation is more or less standard in the literature for matrix manifolds, it may seem ambiguous at first. The need for such a specific notation originates from the fact that geometric objects regarding $\mathbf{S}_+^{n,p}$ can be abstract (theoretical) as well as concrete (implementable). We have chosen a notation that clearly reflects this ambivalent nature.

Like in the introduction to manifolds of Chapter 2, the theorems and properties of differential geometry are typically formulated in an abstract way. When our derivations regarding $\mathbf{S}_+^{n,p}$ borrows from this abstract setting, we stick to the notation of lower case letters x, y, z and Greek symbols ν, η, ξ.

On the other hand, since we are dealing with a matrix manifold, most of these objects are also representable as concrete matrices. It is customary to denote matrices by large capitals X, Y, Z. When our viewpoint involves implementation or concrete representations, we use the notation with large capitals.

The previous distinction is not always clear-cut since sometimes the viewpoint can be both abstract and concrete. Still, we feel that theorems and properties from differential geometry are more transparent using the abstract notation, while issues that involve linear algebra benefit from the matrix notation. For example, the simple notation

$$x = YY^T \in \mathbf{S}_+^{n,p}$$

of the next section reflects that the s.p.s.d. matrix x, as an abstract element of $\mathbf{S}_+^{n,p}$, can be represented (implemented) using the matrix $Y \in \mathbf{R}_*^{n \times p}$.

3.2 Embedded submanifold

In this section we show that $\mathbf{S}_+^{n,p}$ is a smooth manifold for every $p \leq n$. More specifically, we prove that it is an embedded submanifold of $\mathbf{R}^{n\times n}$.

3.2.1 Some characterizations

To begin, we introduce some characterizations for $\mathbf{S}_+^{n,p}$ that will be useful for the rest of the thesis. Recall that $\mathbf{R}_*^{n\times p}$ denotes the set of all full-rank real $n \times p$ matrices and that $\mathrm{St}^{n,p}$ is the Stiefel manifold of all orthonormal matrices in $\mathbf{R}_*^{n\times p}$. Furthermore, we will make use of the general linear group $\mathbf{GL}^n := \mathbf{R}_*^{n\times n}$ and its connected component $\mathbf{GL}_+^n = \{X \in \mathbf{GL}^n \mid \det(X) > 0\}$.

We have the following characterization of $\mathbf{S}_+^{n,p}$.

Proposition 3.1.

$$\mathbf{S}_+^{n,p} = \{x \in \mathbf{S}_+^n \mid \operatorname{rank}(x) = p\} = \{YY^T \mid Y \in \mathbf{R}_*^{n\times p}\}.$$

Proof. The first identity is simply by definition. We prove the second identity by inclusion in both directions. The inclusion $\{YY^T : Y \in \mathbf{R}_*^{n\times p}\} \subseteq \mathbf{S}_+^{n,p}$ is obvious. The inclusion $\mathbf{S}_+^{n,p} \subseteq \{YY^T : Y \in \mathbf{R}_*^{n\times p}\}$ can be shown using a compact eigenvalue decomposition: for each $x \in \mathbf{S}_+^{n,p}$, take $x = VDV^T$ with $V \in \mathrm{St}^{n,p}$ and $D = \operatorname{diag}(d)$ with strictly positive diagonal $d \in \mathbf{R}^p$. Define $D^{1/2} = \operatorname{diag}(\sqrt{d})$ as the matrix square root of D, then we can take $Y = VD^{1/2}$. □

The following is a trivial consequence of the eigenvalue decomposition of the proof above, but we state it here explicitly for visibility.

Corollary 3.2.

$$\mathbf{S}_+^{n,p} = \{VDV^T \mid V \in \mathrm{St}^{n,p},\ D = \operatorname{diag}(d),\ d \in \mathbf{R}^p \ \textit{with}\ d_i > 0, i = 1, \ldots, p\}.$$

Throughout the text, we will sometimes give a remark regarding the implementation of differential algebraic objects for $\mathbf{S}_+^{n,p}$. Apart from the practical relevance, such remarks emphasize that these objects can be computed efficiently, i.e., in $O(np^c)$ complexity with c small.

Remark 3.3. The parameterizations of Prop. 3.1 and Cor. 3.2 can be computed from each other. Given $x = VDV^T$, compute $D^{1/2}$ as the matrix square root of D. Then we get

$$Y := VD^{1/2} \quad \Longrightarrow \quad x = VDV^T = (VD^{1/2})(VD^{1/2})^T = YY^T.$$

Given $x = YY^T$, compute $Y = Q_1 \Sigma Q_2^T$ as the compact SVD. Then we have

$$V := Q_1,\ D := \Sigma^2 \quad \Longrightarrow \quad x = YY^T = (Q_1 \Sigma Q_2^T)(Q_2 \Sigma Q_1^T) = VDV^T.$$

3.2.2 Congruence as a Lie group action

Consider the mapping

$$\theta :\ \mathbf{GL}^n \times \mathbf{S}^n \to \mathbf{S}^n,\ (A, x) \mapsto AxA^T, \tag{3.1}$$

that expresses congruence between symmetric matrices. By Sylvester's law of inertia (Thm. B.1), we know that congruent matrices have the same number of positive, negative and zero eigenvalues (the inertia). Hence, every matrix that is congruent to the matrix

$$e := \begin{bmatrix} I_p & 0_{p\times(n-p)} \\ 0_{(n-p)\times p} & 0_{(n-p)\times(n-p)} \end{bmatrix} \in \mathbf{S}_+^{n,p}, \tag{3.2}$$

will be in $\mathbf{S}_+^{n,p}$. The other way around, i.e., that every matrix in $\mathbf{S}_+^{n,p}$ is congruent to e, is shown in the following proposition.

Proposition 3.4. *Every matrix in $\mathbf{S}_+^{n,p}$ is congruent to e, as defined in (3.2).*

Proof. We need to show that there exists an $A \in \mathbf{GL}^n$ for every $x \in \mathbf{S}_+^{n,p}$ such that $\theta(A, e) = x$. By Prop. 3.1, we have that $x = YY^T$ for some $Y \in \mathbf{R}_*^{n\times p}$. Then such an A is given by $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$ with $Y_\perp \in \mathbf{R}_*^{n\times(n-p)}$ a basis for the orthogonal complement of Y in $\mathbf{GL}^n$. □

In other words, the mapping

$$\theta_e : \mathbf{GL}^n \to \mathbf{S}_+^{n,p},\ A \mapsto AeA^T. \tag{3.3}$$

is surjective and its range is exactly the set $\mathbf{S}_+^{n,p}$.

Throughout the thesis, we will use e as the canonical element of $\mathbf{S}_+^{n,p}$. In the following chapters, the simple mapping θ_e will turn out to be the starting point for our study of the geometry of $\mathbf{S}_+^{n,p}$. Based on θ_e we can already see that $\mathbf{S}_+^{n,p}$ consists of only one component.

Proposition 3.5. *The space $\mathbf{S}_+^{n,p}$ is path-connected.*

Proof. If suffices to connect e to any $x \in \mathbf{S}_+^{n,p}$ by a curve that stays in $\mathbf{S}_+^{n,p}$. Let $x = YY^T$ and define $X := \begin{bmatrix} Y & Y_\perp \end{bmatrix}$ with $Y_\perp$ a basis for the orthonormal part of Y in $\mathbf{GL}^n$. By possibly multiplying one of the columns of Y by -1, one can always accomplish that $\det(X) > 0$ while $x = YY^T$ is still satisfied. Since $\mathbf{GL}_+^n$ is

path-connected, we can construct a curve $A(t) \in \mathbf{GL}_+^n$ which goes from $A(0) = I_n$ to $A(1) = X$. Now, the curve $t \mapsto A(t)eA(t)^T$ will be in $\mathbf{S}_+^{n,p}$ for $0 \leq t \leq 1$ and it obviously connects e to x. □

Remark 3.6. It is straightforward to verify that θ satisfies Definition A.7 for an *action of the Lie group* $\mathbf{GL}^n$. Using the terminology of Definition A.8, we recognize θ_e as the *orbit* through e of this Lie group action. Proposition 3.4 actually shows that θ is *transitive*, which we will exploit extensively in Chapter 6.

3.2.3 Matrices with fixed inertia or fixed rank

We briefly show how our forthcoming derivations regarding the geometry of $\mathbf{S}_+^{n,p}$ can be generalized to other fixed-rank manifolds.

Consider again mapping (3.1). Instead of fixing e for x in θ, we can just as easily choose another symmetric matrix. Let us denote this matrix by z. The inertia of a symmetric matrix (see App. B.1.1 for its definition) is denoted by inertia$(\cdot)$. If inertia$(z) \neq$ inertia(e), then the orbit through z will be disjunct from $\mathbf{S}_+^{n,p}$. In other words, the map

$$\theta_z : \mathbf{GL}^n \to \mathcal{M}_z,\ A \mapsto AzA^T$$

will have a range $\mathcal{M}_z$ disjunct with $\mathbf{S}_+^{n,p}$. Since our derivations of the geometry for $\mathbf{S}_+^{n,p}$ in this chapter and in Chapter 6 is based on mapping θ_e, it allows us to describe each orbit

$$\mathcal{M}_z = \{x \in \mathbf{S}^n \mid \text{inertia}(x) = \text{inertia}(z)\}$$

using the same techniques, but now based on mapping θ_z. For instance, one can show that each orbit is the *smooth manifold of symmetric matrices with fixed inertia*. Except from some remarks later in the thesis, we did not pursue this rigorously.

Remark 3.7. For any symmetric matrix z, the manifold $\mathcal{M}_z$ belongs to the class of so-called *spectral manifolds*, consisting of all matrices whose ordered vector of eigenvalues belongs to some special submanifold of $\mathbf{R}^n$; see Example 4.23 in Daniilidis *et al.* (2009). To derive the geometric structure of these spectral manifolds, the authors employ mathematical tools that are fundamentally different from ours. Their approach is more general but only applicable to local submanifolds and it does not include derivations of geodesics and Hessians.

Remark 3.8. In Helmke & Moore (1994, Prop. 1.14), the authors show that the $m \times n$ real matrices of rank $p \leq \min(m, n)$ are a smooth embedded submanifold of $\mathbf{R}^{m \times n}$. The proof is based on the Lie group action

$$\delta : (\mathbf{GL}^m \times \mathbf{GL}^n) \times \mathbf{R}^{m \times n} \to \mathbf{R}^{m \times n},\ ((A, B), x) \mapsto AxB^{-1}.$$

and exploits the semialgebraic nature of δ, like we do with θ in Section 3.2.5. Apart from the fact that the action consists of two Lie groups and uses inversion instead of transpose, this looks very similar to our action θ. It is reasonable to assume that the techniques in this thesis can also be applied for orbits of this manifold, i.e., *the smooth manifold of (rectangular) matrices of fixed rank*.

3.2.4 An embedded submanifold as local level sets

Now, we are ready for the main result of this section, namely, proving that $\mathbf{S}_+^{n,p}$ is an embedded submanifold. Contrary to the existing proofs in Helmke & Moore (1994); Helmke & Shayman (1995), our approach is inherently constructive.

The proof is based on Theorem 2.15, which has the following principle. By the construction of a smooth atlas, being a submanifold is a local property (Lee, 2003, Lemma 8.1). Hence, it suffices to show that $\mathbf{S}_+^{n,p}$ is the union of several embedded submanifolds of a certain, fixed dimension d. In order to show this, we will construct for each $x \in \mathbf{S}_+^{n,p}$, an open neighborhood $\mathcal{U}_x \subset \mathbf{R}^{n\times n}$ around x, such that $\mathbf{S}_+^{n,p} \cap \mathcal{U}_x$ is the level set of some submersion $F_x : \mathcal{U}_x \to \mathbf{R}^d$; see Figure 3.1. By virtue of Theorem 2.12, the level sets of these submersions are embedded submanifolds of dimension d.

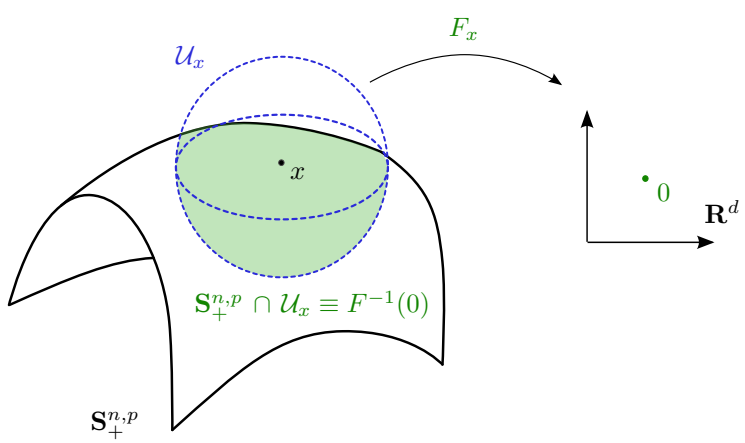


Figure 3.1: Every neighborhood on $\mathbf{S}_+^{n,p}$ is the level set of a submersion of rank d.

Let us first introduce the following block-partitioning of a general matrix $A \in \mathbf{R}^{n\times n}$:

$$A = \begin{bmatrix} A_{11} \in \mathbf{R}^{p\times p} & A_{12} \in \mathbf{R}^{p\times(n-p)} \\ A_{21} \in \mathbf{R}^{(n-p)\times p} & A_{22} \in \mathbf{R}^{(n-p)\times(n-p)} \end{bmatrix}. \tag{3.4}$$

The neighborhoods from above will be formulated in terms of this partitioning. In addition, we require the following property of the Schur complement of a matrix, which can be attributed to Guttman (1946).

Lemma 3.9. *Let*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \in \mathbf{R}^{n\times n} \tag{3.5}$$

be partitioned as (3.4)*. If* A_{11} *is non-singular, then we have the equivalence*

$$\operatorname{rank}(A) = p \quad \Longleftrightarrow \quad A_{22} - A_{21}A_{11}^{-1}A_{12} = 0_{n-p}.$$

Proof. Observe that we have the following equality with the Schur complement $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix}.$$

Since the two outer matrices of the r.h.s. are both of full rank, we get that $\operatorname{rank}(A) = \operatorname{rank}(A_{11}) + \operatorname{rank}(S) = p + \operatorname{rank}(S)$. □

Assume that $A \in \mathbf{S}_+^{n,p}$ is partitioned as (3.4) with $A_{11} \succ 0$. Since the eigenvalues of a matrix depend continuously on the elements of this matrix (Stewart, 2001, Thm. 3.1), the set

$$\mathcal{U}_A = \left\{ \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \in \mathbf{R}^{n\times n} : \Re\,\lambda_{\min}(X_{11}) > 0 \right\},$$

with the same partitioning as (3.4), is a neighborhood of A. This results in the set $\mathbf{S}_+^{n,p} \cap \mathcal{U}_A$ that consists of all matrices $X \in \mathbf{S}_+^{n,p}$ with a full-rank block X_{11}, or, by Lemma 3.9, a zero Schur complement $X_{22} - X_{21}X_{11}^{-1}X_{12}$.

Next, we construct the mapping

$$F_A : \mathcal{U}_A \to \operatorname{skew}(p) \times \mathbf{R}^{p\times(n-p)} \times \mathbf{R}^{(n-p)\times(n-p)},$$

$$\begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \mapsto \left(X_{11} - X_{11}^T,\, X_{12} - X_{21}^T,\, X_{22} - X_{21}X_{11}^{-1}X_{12}\right), \tag{3.6}$$

where skew(p) is the set of all $p \times p$ skew-symmetric matrices. Since skew(p) is a vector space, we can regard F_A as a mapping $\mathcal{U}_A \to \mathbf{R}^d$ with

$$d = p(p-1)/2 + (n-p)p + (n-p)^2 = n^2 - np + p(p-1)/2. \tag{3.7}$$

Observe that $x \in \mathbf{S}_+^{n,p} \cap \mathcal{U}_A$ if and only if $F_A(x) = (0,0,0)$. Hence, the level set $F^{-1}(0,0,0)$ coincides with $\mathbf{S}_+^{n,p} \cap \mathcal{U}_A$. Next, we show that F_A is a submersion.

Lemma 3.10. *Mapping F_A in (3.6) is a submersion. Hence, $F_A^{-1}(0,0,0)$ is an embedded submanifold of $\mathcal{U}_A$ with dimension $np - p(p-1)/2$.*

Proof. First, we proof that F_A is a submersion. By Definition 2.8, F_A should be a smooth map with a surjective differential. Smoothness is evident since F_A, being restricted to $\mathcal{U}_A$, is everywhere defined and consists of only smooth matrix operations (see App. A.2). Its differential is defined as the map

$$\mathrm{D}F_A(X) : T_X\mathcal{U}_A \simeq \mathbf{R}^{n\times n} \to \mathrm{skew}(p) \times \mathbf{R}^{p\times(n-p)} \times \mathbf{R}^{(n-p)\times(n-p)}$$

that satisfies (see App. B.2 for calculus of differentials)

$$\begin{aligned}\mathrm{D}F_A(X)[\Delta] = \big(&\Delta_{11} - \Delta_{11}^T,\ \Delta_{12} - \Delta_{21}^T,\\ &\Delta_{22} - \Delta_{21}X_{11}^{-1}X_{12} - X_{21}X_{11}^{-1}\Delta_{12} + X_{21}X_{11}^{-1}\Delta_{11}X_{11}^{-1}X_{12}\big),\end{aligned}$$

where the tangent vector

$$\Delta = \begin{bmatrix}\Delta_{11} & \Delta_{12}\\ \Delta_{21} & \Delta_{22}\end{bmatrix} \in \mathbf{R}^{n\times n} \tag{3.8}$$

has again partitioning (3.5). Next, we show that $\mathrm{D}F_A(X)$ is surjective for all $X \in \mathcal{U}_A$: for any $(\Omega, M, N) \in \mathrm{skew}(p) \times \mathbf{R}^{p\times(n-p)} \times \mathbf{R}^{(n-p)\times(n-p)}$, we can take Δ with $\Delta_{11} = \Omega/2, \Delta_{12} = M, \Delta_{21} = 0, \Delta_{22} = N + X_{21}X_{11}^{-1}M + X_{21}X_{11}^{-1}\Omega X_{11}^{-1}X_{12}$ such that $\mathrm{D}F_A(X)[\Delta] = (\Omega, M, N)$.
Since F_A is a submersion, the level set $F_A^{-1}(0,0,0)$ is an embedded submanifold by virtue of Theorem 2.12. Its dimension equals $\dim(\mathbf{R}^{n\times n}) - \dim(\mathbf{R}^d) = n^2 - d$. Hence, with (3.7) we obtain $\dim(F_A^{-1}(0,0,0)) = np - p(p-1)/2$. □

So far, we have only defined a neighborhood and corresponding submersion for matrices A that have a full-rank block A_{11}. This does obviously not include every matrix in $\mathbf{S}_+^{n,p}$. However, by means of a permutation of the rows and columns, every matrix in $\mathbf{S}_+^{n,p}$ can be put in this form and this allows us to create a new neighborhood around each $x \in \mathbf{S}_+^{n,p}$ with corresponding submersion.

Proposition 3.11. *The set $\mathbf{S}_+^{n,p}$ is an embedded submanifold in $\mathbf{R}^{n\times n}$ of dimension $np - p(p-1)/2$.*

Proof. The proof consists of two steps.
Step 1. Assume that $A \in \mathbf{S}_+^{n,p}$ is partitioned as in (3.4) with $A_{11} \succ 0$. Then we already established in Lemma 3.10 that this A has a neighborhood $\mathcal{U}_A$ such that $\mathbf{S}_+^{n,p} \cap \mathcal{U}_A$ is an embedded submanifold of $\mathcal{U}_A$.
Step 2. For general $B \in \mathbf{S}_+^{n,p}$ that does not have a positive-definite block B_{11}, we can always permute the rows and columns of B such that it does. Let us denote this rearrangement by the bijection R, hence, $R(B) = A$ with A defined above. Now, consider the neighborhood $\mathcal{U}_B := R^{-1}(\mathcal{U}_A)$ and the mapping $F_B := F_A \circ R$. This $F_B : \mathcal{U}_A \to \mathbf{R}^d$ is obviously again a submersion and so every $B \in \mathbf{S}_+^{n,p}$ has a neighborhood $\mathcal{U}_B \subset \mathbf{R}^{n \times n}$ such that $\mathbf{S}_+^{n,p} \cap \mathcal{U}_B$ is an embedded submanifold of $\mathcal{U}_B$. By Theorem 2.15 this means that $\mathbf{S}_+^{n,p}$ is an embedded submanifold of $\mathbf{R}^{n \times n}$. □

Remark 3.12. The proof of Prop. 3.11 was inspired by a proof in Lee (2003, Ex. 8.14) for the manifold of fixed-rank non-symmetric matrices. In case of Prop. 3.11 however, the proof is more involved since it requires dealing with the symmetry and the positive semidefiniteness of $\mathbf{S}_+^{n,p}$.

3.2.5 An embedded submanifold from semialgebraic geometry

In Helmke & Moore (1994, Ch. 5) and Helmke & Shayman (1995, Prop. 2.1), one can find another technique for proving that $\mathbf{S}_+^{n,p}$ is an embedded submanifold. It relies on properties of so-called *smooth semialgebraic actions* that arise in semialgebraic geometry. We repeat this proof here to show how it is different from our proof above.

Semialgebraic geometry is a branch of mathematics that studies solutions of polynomial equations and inequalities as geometric objects. We do not have the ambition to give an introduction to this broad field, since it is beyond the scope of the thesis. Luckily, only a few relatively simple definitions and theorems suffice for our application. The crucial theorem is the following; see App. A.4 for an explanation of the necessary concepts.

Theorem 3.13 (Thm. A.13)**.** *Let $\delta : \mathcal{G} \times \mathbf{R}^n \to \mathbf{R}^n$ be a smooth action of a Lie group $\mathcal{G}$. Suppose that δ is a semialgebraic mapping, then for each $x \in \mathbf{R}^n$ the orbit of δ through x is a smooth embedded submanifold of $\mathbf{R}^n$.*

We show that this theorem can be directly applied to $\mathbf{S}_+^{n,p}$. In Section 3.2.1, we already referred to mapping

$$\theta : \mathbf{GL}^n \times \mathbf{S}^n \to \mathbf{S}^n, \ (A, x) \mapsto AxA^T,$$

as a smooth Lie group action. If we identify the vector space $\mathbf{S}^n$ as $\mathbf{R}^{n(n+1)/2}$, then the set $\mathbf{GL}^n \times \mathbf{S}^n$ is semialgebraic (Def. A.14). Map θ is algebraic since it only consists of matrix multiplication and transpose. Hence by Thm. A.16, mapping θ is semialgebraic (Def. A.15).

Now, one can proof that $\mathbf{S}_+^{n,p}$ is a submanifold embedded in $\mathbf{R}^{n\times n}$ in a different way. This was done originally in Helmke & Moore (1994, Ch. 5) and Helmke & Shayman (1995, Prop. 2.1).

Proof of Prop. 3.11. By virtue of Theorem 3.13, we get that the orbit of θ through e is an embedded submanifold of $\mathbf{S}^n$. This orbit through e is exactly the set $\mathbf{S}_+^{n,p}$. Since $\mathbf{S}^n$ can be trivially embedded into $\mathbf{R}^{n\times n}$, this concludes the proof of embeddedness. The dimension of the manifold is the same as the dimension of the tangent space. We postpone the derivation of the tangent space to the next section[1]. □

Remark 3.14. This proof is quite elegant and concise. It is also general since it can be applied to any smooth semialgebraic action of a Lie group. On the other hand, it does not reveal the underlying structure of the manifold, like the charts. Furthermore, although the proof of Theorem 3.13 is relatively simple, it is not widespread and only available in Gibson (1979). Compared to our more constructive proof in Section 3.2.4, it is a matter of opinion which one is "better". We leave the choice to the reader.

3.3 Geometric objects

In this section, we derive most of the typical geometric objects from Chapter 2 for the embedded submanifold $\mathbf{S}_+^{n,p}$. Specifically, we formulate the tangent and normal spaces, the metric, the orthogonal projections and the Levi–Civita connection. All of these objects form the building blocks of the Riemannian optimization methods of Chapters 4 and 5. The other missing ingredients are geodesics and retractions, but we deal with them in Sections 3.4 and 3.5.

3.3.1 Tangent space

The tangent space of $\mathbf{S}_+^{n,p}$ can be determined by the differential of the map (3.3). Recall that this is the surjective map

$$\theta_e : \mathbf{GL}^n \to \mathbf{S}_+^{n,p},\ A \mapsto AeA^T.$$

The differential of θ_e at $A \in \mathbf{GL}^n$ is given by

$$\mathrm{D}\theta_e(A) : T_A\mathbf{GL}^n \to T_{AeA^T}\mathbf{S}_+^{n,p},\ \Delta \mapsto \Delta eA^T + Ae\Delta^T. \tag{3.9}$$

[1] Since the derivation of the tangent space does not depend on the embeddedness of $\mathbf{S}_+^{n,p}$, we avoid a circular argument.

Observe that the differential at arbitrary $A \in \mathbf{GL}^n$ is related to the differential at I_n by a a full-rank linear transformation:

$$\begin{aligned} \mathrm{D}\theta_e(A)[\Delta] &= \Delta e A^T + A e \Delta^T \\ &= A(A^{-1}\Delta e + e\Delta^T A^{-T})A^T \\ &= A\left(\mathrm{D}\theta_e(I_n)[A^{-1}\Delta]\right)A^T. \end{aligned}$$

So the rank of θ_e is constant, which makes θ_e a submersion based on Theorem 2.10. By definition of a submersion, the range of $\mathrm{D}\theta_e(X)$ is the whole tangent space of $\mathbf{S}_+^{n,p}$ at $x := \theta_e(A) = AeA^T$. This gives from (3.9)

$$\begin{aligned} T_x\mathbf{S}_+^{n,p} &= \{\Delta e A^T + A e \Delta^T \mid \Delta \in \mathbf{R}^{n\times n}\} \\ &= \{\Delta A^{-1} x + x A^{-T}\Delta^T \mid \Delta \in \mathbf{R}^{n\times n}\} \\ &= \{\Delta x + x\Delta^T \mid \Delta \in \mathbf{R}^{n\times n}\}. \end{aligned} \tag{3.10}$$

The dimension of the tangent space is the same as the dimension of $\mathbf{S}_+^{n,p}$, which is $pn - p(p-1)/2$.

Clearly, expression (3.10) is an over-parameterization. A minimal parameterization is given by the following proposition.

Proposition 3.15. *The tangent space of $\mathbf{S}_+^{n,p}$ at $x = YY^T$ is given by*

$$T_x\mathbf{S}_+^{n,p} = \left\{ \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} H & K^T \\ K & 0 \end{bmatrix} \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} \;\middle|\; H \in \mathbf{S}^p,\, K \in \mathbf{R}^{(n-p)\times p} \right\}, \tag{3.11}$$

with $Y_\perp \in \mathbf{R}_*^{n\times(n-p)}$ a basis for the orthogonal complement of Y in $\mathbf{GL}^n$.

Proof. The right-hand side of (3.11) has the correct number of degrees of freedom, it is a linear space and it is included in expression (3.10) for $T_x\mathbf{S}_+^{n,p}$; take $\Delta = (YH/2 + Y_\perp K)(Y^TY)^{-1}Y^T$. □

Another representation, which will sometimes be useful because of the explicit presence of the orthogonal matrices, is the parameterization corresponding to that of Cor. 3.2.

Corollary 3.16. *The tangent space of $\mathbf{S}_+^{n,p}$ at $x = VDV^T$ is given by*

$$T_x\mathbf{S}_+^{n,p} = \left\{ \begin{bmatrix} V & V_\perp \end{bmatrix} \begin{bmatrix} H & K^T \\ K & 0 \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix} \;\middle|\; H \in \mathbf{S}^p,\, K \in \mathbf{R}^{(n-p)\times p} \right\}, \tag{3.12}$$

with $V_\perp \in \mathrm{St}^{n\times(n-p)}$ an orthonormal basis for the orthogonal complement of V in $\mathbf{GL}^n$.

Remark 3.17. Observe that the formulation of the tangent space in Prop. 3.15 uses $Y_\perp$ and that of Cor. 3.2 uses $V_\perp$. Both are prohibitively expensive to construct for large n. Luckily, an algorithm usually only needs to work with tangent vectors and not the whole tangent space. By working out the product with $Y_\perp$, we can store a tangent vector in $x = YY^T$ cheaply as

$$\nu_x = YHY^T + ZY^T + YZ^T \quad \text{with } H \in \mathbf{S}^p,\ Z \in \mathbf{R}^{n\times p},\ Z^TY = 0. \tag{3.13}$$

or, in $x = VDV^T$ as

$$\nu_x = V\widetilde{H}V^T + \widetilde{Z}V^T + V\widetilde{Z}^T \quad \text{with } H \in \mathbf{S}^p,\ \widetilde{Z} \in \mathbf{R}^{n\times p} \text{ and } \widetilde{Z}^TV = 0. \tag{3.14}$$

These two parameterizations are related as

$$\widetilde{H} = \Sigma Q^T H Q \Sigma, \quad \widetilde{Z} = ZQ\Sigma$$

with $Y = V\Sigma Q^T$ a compact SVD.

3.3.2 Riemannian metric

The Euclidean metric

$$g^{\mathrm{E}}(Z_1, Z_2) := \operatorname{tr}(Z_1^T Z_2), \quad \text{for all } Z_1, Z_2 \in \mathbf{R}^{n\times n}. \tag{3.15}$$

is arguably the most simple metric for $\mathbf{R}^{n\times n}$. Hence, we will use is also for $\mathbf{S}_+^{n,p}$ by simply restricting it to $\mathbf{S}_+^{n,p}$.

Proposition 3.18. *The relation*

$$g_x^E(\nu_x, \eta_x) := \operatorname{tr}(\nu_x \eta_x), \quad \textit{for all } \nu_x, \eta_x \in T_x\mathbf{S}_+^{n,p},$$

defines a Riemannian metric on $\mathbf{S}_+^{n,p}$*. This turns* $(\mathbf{S}_+^{n,p}, g^E)$ *into a Riemannian submanifold of* $\mathbf{R}^{n\times n}$*.*

Remark 3.19. Remark 3.17 shows how to parameterize tangent vectors in $T_x\mathbf{S}_+^{n,p}$. Suppose we have

$$\nu_x = YH_\nu Y^T + Z_\nu Y^T + YZ_\nu^T = V\widetilde{H}_\nu V^T + \widetilde{Z}_\nu V^T + V\widetilde{Z}_\nu^T$$

$$\eta_x = YH_\eta Y^T + Z_\eta Y^T + YZ_\eta^T = V\widetilde{H}_\eta V^T + \widetilde{Z}_\eta V^T + V\widetilde{Z}_\eta^T,$$

then the metric can be computed as

$$\begin{aligned} g_x^{\mathrm{E}}(\nu_x, \eta_x) &= \operatorname{tr}[Y^TY(H_\nu Y^TYH_\eta + 2Z_\nu^T Z_\eta)] \\ &= \operatorname{tr}(\widetilde{H}_\nu\widetilde{H}_\eta + 2\widetilde{Z}_\nu^T\widetilde{Z}_\eta) \end{aligned}$$

3.3.3 Normal space

The normal space at $x \in \mathbf{S}_+^{n,p}$ consists of all matrices perpendicular (w.r.t. g^{E}) to the tangent space at x, i.e.,

$$N_x\mathbf{S}_+^{n,p} = \{Z \in \mathbf{R}^{n\times n} \mid \operatorname{tr}(Z^T\nu) = 0 \text{ for all } \nu \in T_x\mathbf{S}_+^{n,p}\}. \tag{3.16}$$

Using the form (3.10) for the tangent vectors T, we can write the orthogonality constraint as

$$\operatorname{tr}(Z^T\nu) = \operatorname{tr}(Z^T\Delta x + Z^T x\Delta^T) = \operatorname{tr}(\Delta x(Z^T + Z)).$$

This expression has to vanish for all $\Delta \in \mathbf{R}^{n\times n}$, so we see that the normal space must have the form

$$N_x\mathbf{S}_+^{n,p} = \{Z \in \mathbf{R}^{n\times n} \mid x(Z^T + Z) = 0\}. \tag{3.17}$$

Again, we can simplify this for a factored matrix.

Proposition 3.20. *The normal space at $x = YY^T$ is given by*

$$N_x\mathbf{S}_+^{n,p} = \left\{ \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} \Omega & -L^T \\ L & M \end{bmatrix} \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} \;\middle|\; \Omega \in \operatorname{skew}(p), \right.$$
$$\left. M \in \mathbf{R}^{(n-p)\times(n-p)},\, L \in \mathbf{R}^{(n-p)\times p} \right\}. \tag{3.18}$$

The dimension of the normal space is $n^2 - pn + p(p-1)/2$.

Proof. The right-hand side of (3.18) has the correct number of degrees of freedom $n^2 - \dim T_x\mathbf{S}_+^{n,p}$ and it is a linear space. Furthermore, it consists of matrices that are perpendicular to $T_x\mathbf{S}_+^{n,p}$: take $\nu_x \in T_x\mathbf{S}_+^{n,p}$ as in eq. (3.11) and Z as in (3.18), then

$$\begin{aligned}
\operatorname{tr}(Z^T\nu_x) &= \operatorname{tr}\left(\begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} -\Omega & L^T \\ -L & M^T \end{bmatrix} \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} H & K^T \\ K & 0 \end{bmatrix} \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} \right) \\
&= \operatorname{tr}\left(\begin{bmatrix} -\Omega & L^T \\ -L & M^T \end{bmatrix} \begin{bmatrix} Y^TY & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} H & K^T \\ K & 0 \end{bmatrix} \begin{bmatrix} Y^TY & 0 \\ 0 & I \end{bmatrix} \right) \\
&= \operatorname{tr}\left(\begin{bmatrix} -\Omega Y^TYHY^TY + L^TKY^TY & \times \\ \times & -LY^TYK^T \end{bmatrix} \right) \\
&= \operatorname{tr}(\Omega Y^TYHY^TY)
\end{aligned}$$

Observe that, since $\Omega \in \operatorname{skew}(p)$ and $Y^TYHY^TY \in \mathbf{S}^p$, the last expression is indeed zero by virtue of property (B.3). □

3.3.4 Orthogonal projections

Having established the tangent and normal space as two complementary spaces, we can define the projection on the tangent space along the normal space, and vice versa.

Proposition 3.21. *The orthogonal projections (w.r.t. g^E) onto $T_x\mathbf{S}_+^{n,p}$ and $N_x\mathbf{S}_+^{n,p}$ in $x = YY^T \in \mathbf{S}_+^{n,p}$ are given by, respectively,*

$$\mathrm{P}_x^{\mathrm{t}}(Z) = \frac{1}{2}(\mathrm{P}_Y(Z+Z^T)\mathrm{P}_Y + \mathrm{P}_Y^{\perp}(Z+Z^T)\mathrm{P}_Y + \mathrm{P}_Y(Z+Z^T)\mathrm{P}_Y^{\perp}),$$

$$\mathrm{P}_x^{\mathrm{n}}(Z) = Z - \mathrm{P}_x^{\mathrm{t}}(Z) = \frac{1}{2}(\mathrm{P}_Y^{\perp}(Z+Z^T)\mathrm{P}_Y^{\perp} + Z - Z^T),$$

with $\mathrm{P}_Y := Y(Y^TY)^{-1}Y^T$ and $\mathrm{P}_Y^{\perp} := I - \mathrm{P}_Y$.

Proof. Express that $Z - (YH + Y_\perp K)Y^T - Y(YH + Y_\perp K)^T$, with H and K constrained as in Prop. 3.15, belongs to the normal space (3.17). □

For reasons that will be clear later on, it is convenient to split the projector $\mathrm{P}_x^{\mathrm{t}}$ into two other orthogonal projectors that are mutually orthogonal also. The first is the projection onto the subspace $\{YHY \mid H \in \mathbf{S}^p\}$ and is denoted by $\mathrm{P}_x^{\mathrm{t,s}}$ due to the presence of the symmetric matrices H. It satisfies

$$\mathrm{P}_x^{\mathrm{t,s}}(Z) = \frac{1}{2}\mathrm{P}_Y(Z+Z^T)\mathrm{P}_Y. \tag{3.19}$$

The other is the projection onto $\{Y_\perp KY^T + YK^TY_\perp^T \mid K \in \mathbf{R}^{(n-p)\times p}\}$ and is denoted by $\mathrm{P}_x^{\mathrm{t,p}}$ since it involves the matrix $Y_\perp$ that is perpendicular to Y. It is given by

$$\mathrm{P}_x^{\mathrm{t,p}}(Z) = \frac{1}{2}(\mathrm{P}_Y^{\perp}(Z+Z^T)\mathrm{P}_Y + \mathrm{P}_Y(Z+Z^T)\mathrm{P}_Y^{\perp}). \tag{3.20}$$

Observe that we have indeed $\mathrm{P}_x^{\mathrm{t}} = \mathrm{P}_x^{\mathrm{t,s}} + \mathrm{P}_x^{\mathrm{t,p}}$. Furthermore, since $\mathrm{tr}(P_Y P_Y^{\perp}) = \mathrm{tr}(P_Y^{\perp} P_Y) = 0$, we have that $g^{\mathrm{E}}(\mathrm{P}_x^{\mathrm{t,s}}(Z), \mathrm{P}_x^{\mathrm{t,p}}(Z)) = 0$.

Since these projectors are linear operators, they can be represented as matrices. The typical technique is to use vectorization which stacks the columns of matrix into a vector, expressed by the vectorization operator $\mathrm{vec} : \mathbf{R}^{n\times n} \to \mathbf{R}^{n^2}$; see App. B.1.4. Applying the $\mathrm{vec}\,(\cdot)$ operator to the expressions (3.19)-(3.20) and using

properties (B.6) and (B.8), we obtain the $n^2 \times n^2$ matrices

$$P_x^{\mathrm{t,s}} = \frac{1}{2}(P_Y \otimes P_Y)(I + \Pi), \tag{3.21}$$

$$P_x^{\mathrm{t,p}} = \frac{1}{2}(P_Y \otimes P_Y^{\perp} + P_Y^{\perp} \otimes P_Y)(I + \Pi), \tag{3.22}$$

$$P_x^{\mathrm{t}} = P_x^{\mathrm{t,s}} + P_x^{\mathrm{t,p}}. \tag{3.23}$$

Since these operators are matrices, we have used the usual cursive notation. This is in contrast to the more general operators $\mathrm{P}_x^{\mathrm{t}}, \mathrm{P}_x^{\mathrm{t,s}}$ and $\mathrm{P}_x^{\mathrm{t,p}}$ from above that are denoted in roman.

Matrix Π in (3.21)–(3.22) is the symmetric permutation matrix, known in Van Loan (2000) as the *perfect shuffle* $S_{n,n}$, that satisfies $\mathrm{vec}(A^T) = \Pi\,\mathrm{vec}(A)$. We refer to Van Loan (2000) for the concrete expression of $S_{n,n}$. To verify the symmetry of the projection matrices, we can use the property that Π allows one to switch Kronecker products as follows: $\Pi(A \otimes B)\Pi = B \otimes A$, for square A, B of equal size.

3.3.5 Levi–Civita connection

The canonical choice for the connection is the Levi–Civita connection. Since $(\mathbf{S}_+^{n,p}, g^{\mathrm{E}})$ is a Riemannian submanifold of $(\mathbf{R}^{n\times n}, g^{\mathrm{E}})$ with the Euclidean metric g^{E} of Prop. 3.18, this connection equals the classical directional derivative followed by an orthogonal projection onto the tangent space.

Proposition 3.22. *Let η be a vector field on $(\mathbf{S}_+^{n,p}, g^E)$ with g^E the metric of Prop. 3.18. Then the Levi–Civita connection ∇ on $(\mathbf{S}_+^{n,p}, g^E)$ in $x \in \mathbf{S}_+^{n,p}$ is given by*

$$\nabla_\nu \eta(x) = \mathrm{P}_x^{\mathrm{t}}(\mathrm{D}\eta(x)[\nu_x])$$

for all $\nu_x \in T_x\mathbf{S}_+^{n,p}$.

Proof. Apply Theorem 2.24 together with (2.8). □

3.4 Geodesics

Since geodesics are fundamental objects in differential geometry, we derive them here for $(\mathbf{S}_+^{n,p}, g^{\mathrm{E}})$. By Definition 2.26, a geodesic is a curve $t \mapsto \gamma(t)$ that *stays* on $\mathbf{S}_+^{n,p}$ with *zero acceleration* $\nabla_{\dot\gamma(t)}\dot\gamma(t) = 0$ for all t where the geodesic is defined.

Let the $\gamma(t) \in \mathbf{S}_+^{n,p}$ for some t. Then these two defining conditions for a geodesic can be expressed in terms of the tangent and normal spaces, as displayed in Fig. 3.2:

$$\dot{\gamma}(t) \in T_{\gamma(t)}\mathbf{S}_+^{n,p} \tag{3.24}$$

$$\ddot{\gamma}(t) \in N_{\gamma(t)}\mathbf{S}_+^{n,p} \tag{3.25}$$

Indeed, the integrated flow of any curve that starts on $\mathbf{S}_+^{n,p}$ and obeys condition (3.24) will lead to a curve that stays on $\mathbf{S}_+^{n,p}$ (Hairer *et al.*, 2006, Thm. 5.2). Furthermore, condition (3.25) is equivalent to $\nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) = 0$ if we apply Prop. 3.22 for the connection.

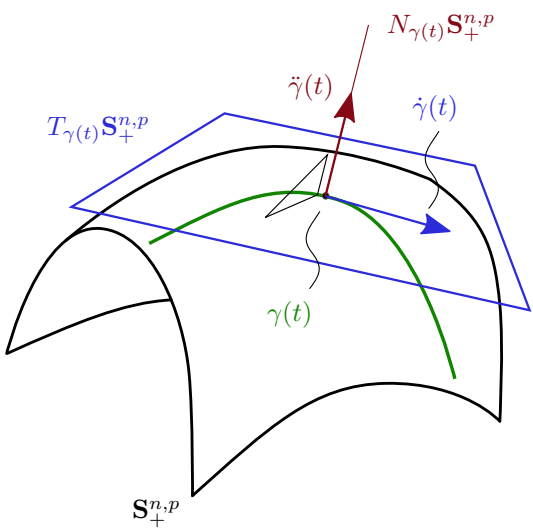


Figure 3.2: A geodesic $\gamma(t)$ on the Riemannian submanifold $\mathbf{S}_+^{n,p}$.

In this section, we will derive an ordinary differential equation (ODE) for the curve $\gamma(t)$ such that these two conditions are satisfied. This allows us to describe all the geodesics on $(\mathbf{S}_+^{n,p}, g^{\mathrm{E}})$. In specific, we will derive an initial value problem (IVP) that, given the *foot* of the geodesic $\gamma(0) = \gamma_0$ and the *initial direction* $\dot{\gamma}(0) = \dot{\gamma}_0$, can be numerically integrated to the whole geodesic curve.

3.4.1 Derivation of the ODE

First, make the substitution $\gamma(t) = Y(t)Y(t)^T$ with $Y(t) \in \mathbf{R}_*^{n\times p}$. Now $\gamma(t)$ belongs to $\mathbf{S}_+^{n,p}$ by construction, hence $\dot{\gamma} = \dot{Y}Y^T + Y\dot{Y}^T \in T_x\mathbf{S}_+^{n,p}$ and condition (3.24) is

trivially satisfied and the problem is reduced to finding an ODE for $\dot{Y}(t)$.

Observe that any vector $\dot{Y}(t)$ can be decomposed as

$$\dot{Y}(t) = \mathrm{P}_Y \dot{Y}(t) + \mathrm{P}_Y^{\perp} \dot{Y}(t) = Y(t)(H(t) + \Omega(t)) + \mathrm{P}_Y^{\perp} Z(t)$$

with unknown matrices $H(t) \in \mathbf{S}^p$, $\Omega(t) \in \mathrm{skew}(p)$ and $Z(t) \in \mathbf{R}^{n\times p}$. By working out the expression $\dot{\gamma} = \dot{Y}Y^T + Y\dot{Y}^T$, one can see that Ω has no influence on $\dot{\gamma}$, hence it is sufficient that the ODE for $Y(t)$ is of the form

$$\dot{Y}(t) = Y(t)H(t) + \mathrm{P}_{Y(t)}^{\perp} Z(t), \tag{3.26}$$

where we additionally need to derive an ODE for the matrices $H(t) \in \mathbf{S}^p$ and $Z(t) \in \mathbf{R}^{n\times p}$.

Next, we impose condition (3.25) on $\ddot{\gamma} = \ddot{Y}Y^T + Y\ddot{Y}^T + 2\dot{Y}\dot{Y}^T$ and see what this means for $Y(t)$, $H(t)$ and $Z(t)$. First, we work out

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}(\mathrm{P}_Y^{\perp} Z) &= \frac{\mathrm{d}}{\mathrm{d}t}(Z - Y(Y^TY)^{-1}Y^TZ) \\
&= \mathrm{P}_Y^{\perp}\dot{Z} - \dot{Y}(Y^TY)^{-1}Y^TZ - Y(Y^TY)^{-1}\dot{Y}^TZ \\
&\quad + Y(Y^TY)^{-1}[\dot{Y}^TY + Y^T\dot{Y}](Y^TY)^{-1}Y^TZ \\
&= \mathrm{P}_Y^{\perp}\dot{Z} - \mathrm{P}_Y^{\perp}Z(Y^TY)^{-1}Y^TZ - Y(Y^TY)^{-1}Z^T\mathrm{P}_Y^{\perp}Z,
\end{aligned}$$

where the derivative of the matrix inverse can be computed from (B.11). We continue with

$$\begin{aligned}
\ddot{Y} &= \dot{Y}H + Y\dot{H} + \frac{\mathrm{d}}{\mathrm{d}t}(\mathrm{P}_Y^{\perp}Z) \\
&= YH^2 + \mathrm{P}_Y^{\perp}ZH + Y\dot{H} + \mathrm{P}_Y^{\perp}\dot{Z} - \mathrm{P}_Y^{\perp}Z(Y^TY)^{-1}Y^TZ \\
&\quad - Y(Y^TY)^{-1}Z^T\mathrm{P}_Y^{\perp}Z.
\end{aligned} \tag{3.27}$$

Now, project $\ddot{\gamma}$ onto the tangent and the normal space. Define the following four subspaces that are mutually orthogonal:

$$\{YX_{YY}Y^T \mid X_{YY} \in \mathbf{R}^{p\times p}\}, \tag{3.28}$$

$$\{Y_\perp X_{\perp Y}Y^T \mid X_{\perp Y} \in \mathbf{R}^{(n-p)\times p}\}, \tag{3.29}$$

$$\{YX_{Y\perp}Y_\perp^T \mid X_{Y\perp} \in \mathbf{R}^{p\times(n-p)}\}, \tag{3.30}$$

$$\{Y_\perp X_{\perp\perp}Y_\perp^T \mid X_{\perp\perp} \in \mathbf{R}^{(n-p)\times(n-p)}\}. \tag{3.31}$$

Observe that together they span $\mathbf{R}^{n\times n}$. Hence, we have

$$\ddot{\gamma} = YX_{YY}Y^T + \mathrm{P}_Y^{\perp}X_{\perp Y}Y^T + YX_{Y\perp}\mathrm{P}_Y^{\perp} + \mathrm{P}_Y^{\perp}X_{\perp\perp}\mathrm{P}_Y^{\perp} \tag{3.32}$$

for some matrices $X_{YY}, X_{\perp Y}, X_{Y\perp}, X_{\perp\perp}$ as in (3.28)–(3.31). Plugging in (3.26)–(3.27) for $\dot{Y}$ and $\ddot{Y}$, we obtain

$$X_{YY} = 4H^2 + 2\dot{H} - (Y^TY)^{-1}Z^T\mathrm{P}_Y^{\perp}Z - Z^T\mathrm{P}_Y^{\perp}Z(Y^TY)^{-1} \tag{3.33}$$

$$X_{\perp Y} = 3ZH + \dot{Z} - Z(Y^TY)^{-1}Y^TZ \tag{3.34}$$

$$X_{Y\perp} = X_{\perp Y}^T \tag{3.35}$$

$$X_{\perp\perp} = 2ZZ^T.$$

Condition (3.25) can be expressed in terms of these matrices. By definition of the projectors $\mathrm{P}_x^{\mathrm{t}}, \mathrm{P}_x^{\mathrm{n}}$ in Prop. 3.21, we have the equivalence

$$\ddot{\gamma}(t) \in N_{\gamma(t)}\mathbf{S}_+^{n,p} \quad \Longleftrightarrow \quad \mathrm{P}_{\gamma(t)}^{\mathrm{t}}(\ddot{\gamma}(t)) = 0.$$

After plugging in (3.32), this becomes

$$\begin{aligned}\ddot{\gamma}(t) \in N_{\gamma(t)}\mathbf{S}_+^{n,p} \quad \Longleftrightarrow \quad & Y(X_{YY} + X_{YY}^T)Y^T + Y(X_{Y\perp} + X_{\perp Y}^T)\mathrm{P}_Y^{\perp} \\ & + Y_{\perp}(X_{Y\perp}^T + X_{\perp Y})\mathrm{P}_Y = 0.\end{aligned}$$

The r.h.s. expression has to vanish for all Y. Together with (3.35) and exploiting the symmetry of X_{YY}, we finally obtain

$$\ddot{\gamma}(t) \in N_{\gamma(t)}\mathbf{S}_+^{n,p} \quad \Longleftrightarrow \quad X_{YY} = 0,\ X_{Y\perp} = 0. \tag{3.36}$$

Imposing these last conditions with (3.33)–(3.34), we arrive at the ODE for a geodesic $\gamma(t) = Y(t)Y(t)^T$.

Proposition 3.23. *The geodesic $\gamma(t) = Y(t)Y(t)^T$ on $(\mathbf{S}_+^{n,p}, g^E)$ with foot $\gamma(0) = Y_0Y_0^T$ and direction $\dot{\gamma}(0) = 2Y_0H_0Y_0^T + Z_0Y_0^T + Y_0Z_0^T$ is the solution of the IVP*

$$\dot{Y} = YH + \mathrm{P}_Y^{\perp}Z, \tag{3.37}$$

$$\dot{H} = -2H^2 + \frac{1}{2}(Y^TY)^{-1}Z^T\mathrm{P}_Y^{\perp}Z + \frac{1}{2}Z^T\mathrm{P}_Y^{\perp}Z(Y^TY)^{-1}, \tag{3.38}$$

$$\dot{Z} = -3ZH + Z(Y^TY)^{-1}Y^TZ, \tag{3.39}$$

with initial conditions $Y(0) = Y_0$, $H(0) = H_0$ and $Z(0) = \widetilde{Z}_0$ such that $\mathrm{P}_{Y_0}^{\perp}\widetilde{Z}_0 = Z_0$.

3.4.2 Analytical solution of a straight line

We did not find an analytical solution for this IVP for all arbitrary initial conditions. However, in case $\mathrm{P}^{\perp}_{Y(0)} Z(0) = 0$ the geodesic reduces to a straight line $\gamma(t) = \gamma_0 + t\dot{\gamma}_0$ with $\gamma_0 = Y_0 Y_0^T$ and $\dot{\gamma}_0 = 2Y_0 H_0 Y_0^T$ for all t where $\gamma(t)$ remains in $\mathbf{S}^{n,p}_+$. This can be seen from (3.37)–(3.39) as follows: by assumption

$$\gamma(t) = \gamma_0 + t\dot{\gamma}_0 = Y_0(I + 2tH_0)Y_0^T = Y(t)Y(t)^T,$$

and so $Y(t) := Y_0(I + 2tH_0)^{1/2}$ for all t where $I + 2tH_0 \succ 0$. Since H_0 commutes with all powers of $I + 2tH_0$, differentiating $Y(t)$ reduces to scalar differentiation. This gives,

$$\dot{Y}(t) = Y_0(I + 2tH_0)^{-1/2} H_0 = Y(t)(I + 2tH_0)^{-1} H_0$$

and so $H(t) := (I + 2tH_0)^{-1} H_0$. Finally,

$$\dot{H}(t) = -(I + 2tH_0)^{-2}(2H_0^2) = -2H^2(t)$$

and $Z(t) = 0$.

The example above shows the interval where the geodesics are well defined can be finite. Hence, the geodesics are not complete. For the same reason, Euclidean geodesics in $\mathbf{S}^{n,n}_+$ are not complete, see Moakher (2005).

Proposition 3.24. *The metric space* $(\mathbf{S}^{n,p}_+, g^E)$ *is not complete.*

3.4.3 Well-conditioned ODE

As we will see in Section 3.4.4, the numerical integration of the ODE in Prop. 3.23 tends to fail due to a blowup of $Z(t)$. We will therefore modify the equation of motion for $Z(t)$ such that the resulting ODE is better conditioned.

It is obvious from the ODE that only $\mathrm{P}^{\perp}_Y Z$ has an influence on Y, and thus on γ, so there is some freedom for Z as long as $\mathrm{P}^{\perp}_Y Z$ remains the same. Indeed, replacing the equation of motion (3.39) for $Z(t)$ by

$$\dot{Z} = -3ZH + Z(Y^T Y)^{-1} Y^T Z + YM, \tag{3.40}$$

with $M \in \mathbf{R}^{p \times p}$ arbitrary, will still result in the same expressions for $\dot{Y}$ and $\dot{H}$ in Prop. 3.23. Hence, this transformation does not change the geodesic $\gamma = YY^T$. We can use this freedom to our benefit to keep the factor Z well behaved. Specifically, we will demand that the new $Z(t)$ is orthogonal to $Y(t)$ at all times t.

Suppose $Z(t)$ is orthogonal to $Y(t)$ for all t where the geodesic exists, then $\mathrm{tr}(Y(t)^T Z(t)) = 0$. Differentiating gives

$$\mathrm{tr}(\dot{Y}(t)^T Z(t)) + \mathrm{tr}(Y(t)\dot{Z}(t)) = 0.$$

Substituting (3.26) for $\dot{Y}$ and (3.40) for the new $\dot{Z}$, we get, after exploiting $Y^TZ = 0$, that

$$\operatorname{tr}(Z(t)^T \mathrm{P}^{\perp}_{Y(t)} Z(t)) + \operatorname{tr}(Y(t)^T Y(t) M(t)) = 0.$$

Since this equality holds for arbitrary Y, the traces can be dropped and we have

$$M(t) = -(Y(t)^T Y(t))^{-1}(Z(t)^T Z(t)).$$

Finally, we obtain the following ODE.

Proposition 3.25. *The geodesic $\gamma(t) = Y(t)Y(t)^T$ on $(\mathbf{S}_+^{n,p}, g^E)$ with foot $\gamma(0) = Y_0Y_0^T$ and direction $\dot{\gamma}(0) = 2Y_0H_0Y_0^T + Z_0Y_0^T + Y_0Z_0^T$ satisfies the IVP*

$$\dot{Y} = YH + Z,$$

$$\dot{H} = -2H^2 + \frac{1}{2}(Y^TY)^{-1}(Z^TZ) + \frac{1}{2}(Z^TZ)(Y^TY)^{-1},$$

$$\dot{Z} = -3ZH - Y(Y^TY)^{-1}(Z^TZ),$$

with initial conditions $Y(0) = Y_0$, $H(0) = H_0$ and $Z(0) = Z_0$ where $Z_0^TY_0 = 0$.

By design, this ODE has the invariant that $Z(t)$ stays perpendicular to $Y(t)$ for all t where the geodesic exists. From the numerical experiments below, it will be obvious that this ODE is indeed much more suited for numerical integration.

It is possible to compute H using a smaller ODE then the one from Prop. 3.25. Introducing $A(t) = Y(t)^TY(t)$ and $B(t) = Z(t)^TZ(t)$, we see that $H(t)$ satisfies the ODE

$$\dot{A} = AH + HA,$$

$$\dot{B} = -3(BH + HB),$$

$$\dot{H} = -2H^2 + \frac{1}{2}A^{-1}B + \frac{1}{2}BA^{-1}.$$

Once the matrices $A(t)$, $B(t)$ and $H(t)$ are obtained, the geodesic satisfies the linear homogeneous ODE

$$\begin{bmatrix}\dot{Y} & \dot{Z}\end{bmatrix} = \begin{bmatrix}Y & Z\end{bmatrix}\begin{bmatrix}H & -A^{-1}B \\ I_p & -3H\end{bmatrix}.$$

3.4.4 Numerical example

Let us illustrate the behavior of the geodesics for the small example $\mathbf{S}_{+}^{2,1}$. Take the following initial conditions for a geodesic $\gamma(t) = Y(t)Y(t)^T$:

$$Y(0) = \begin{bmatrix} Y_1(0) \\ Y_1(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H(0) = -1, \quad Z(0) = \begin{bmatrix} Z_1(0) \\ Z_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.41}$$

Since we have two expressions for the geodesics, we integrate both the ODEs of Prop. 3.23 and of Prop. 3.25 with `ode45` from MATLAB. The absolute and relative error tolerances is 10^{-10} and 10^{-7} respectively.

In Figure 3.3(a), we show the three unique elements of

$$\gamma(t) = \begin{bmatrix} \gamma_{1,1}(t) & \gamma_{1,2}(t) \\ \gamma_{2,1}(t) & \gamma_{2,2}(t) \end{bmatrix}.$$

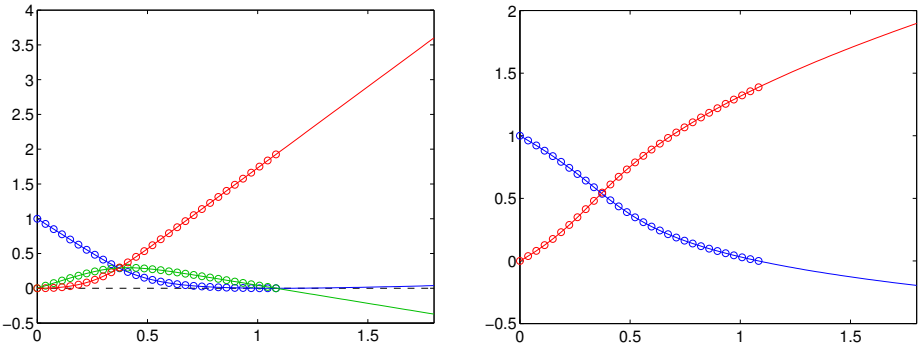
In addition, the two entries of $Y(t)$ are depicted in Fig. 3.3(b). Although the obtained curves $\gamma(t)$ and $Y(t)$ are the same for both ODEs, only the one from Prop. 3.25 can be integrated for all times. At around $t = 1.1$, the numerical integration of the ODE from Prop. 3.23 fails.

Figure 3.4 shows some insights in the matrices that need to be integrated. The first thing we notice in Fig. 3.4(a) is that $Z_2(t) \to \infty$ when $t \simeq 1.1$ for the ODE from Prop. 3.23. On the other hand, the components of $Z(t)$ for the ODE from Prop. 3.25 remain bounded. This well-conditioned ODE was constructed such that $Z(t)$ remains perpendicular to $Y(t)$. In Fig. 3.4(b), the norm of the projected part of $Z(t)$ in $Y(t)$ is shown to be about 10^{-10}, the absolute error tolerance of the numerical integration. Clearly, the well-conditioned ODE succeeds in keeping $Y(t)$ orthogonal to $Z(t)$, which in its turn allows us to integrate the whole geodesic.

By definition, an embedded geodesic should have an acceleration that is perpendicular to the tangent space. In other words, $\mathrm{P}^{\mathrm{t}}_{\gamma(t)}\ddot{\gamma}(t)$ should be zero. This is indeed the case, as can be seen in Fig. 3.4(c). Due to the perpendicular acceleration vector, the velocity $\dot{\gamma}(t)$ should remain constant. This is verified in Fig. 3.4(d). In addition, we see that in the beginning, the acceleration $\ddot{\gamma}$ is nonzero, while for $t \to \infty$, the acceleration $\ddot{\gamma}$ tends to zero.

3.5 Retractions

In this section we derive two types of retraction curves for the embedded geometry. Both have a practical use in the Riemannian optimization algorithms of the next chapters. The first one, based on projection, will be implemented to actually perform the retraction in search algorithms. The second one, a truncated Taylor series, will be used to analytically derive Hessians of objective functions.



(a) $\gamma_{1,1}(t)$, $\gamma_{1,2}(t) = \gamma_{2,1}(t)$, $\gamma_{2,2}(t)$ (b) $Y_1(t)$, $Y_2(t)$

Figure 3.3: The components of the geodesic $\gamma(t) = Y(t)Y(t)^T$ on $\mathbf{S}_+^{2,1}$ in function of t. The initial data is (3.41) and the geodesic is integrated based on Prop. 3.23 (○) and Prop. 3.25 (**−**).

In addition to these two types of retractions, we briefly mention the orthographic retraction, which was recently proposed in Absil & Malick (2010). In a small numerical experiment, we compare the retractions and show that the orthogonal projection is most likely to be preferred for the numerical algorithms.

3.5.1 Orthogonal projection

One of the simplest choices for a retraction is the orthogonal projection onto $\mathbf{S}_+^{n,p}$,

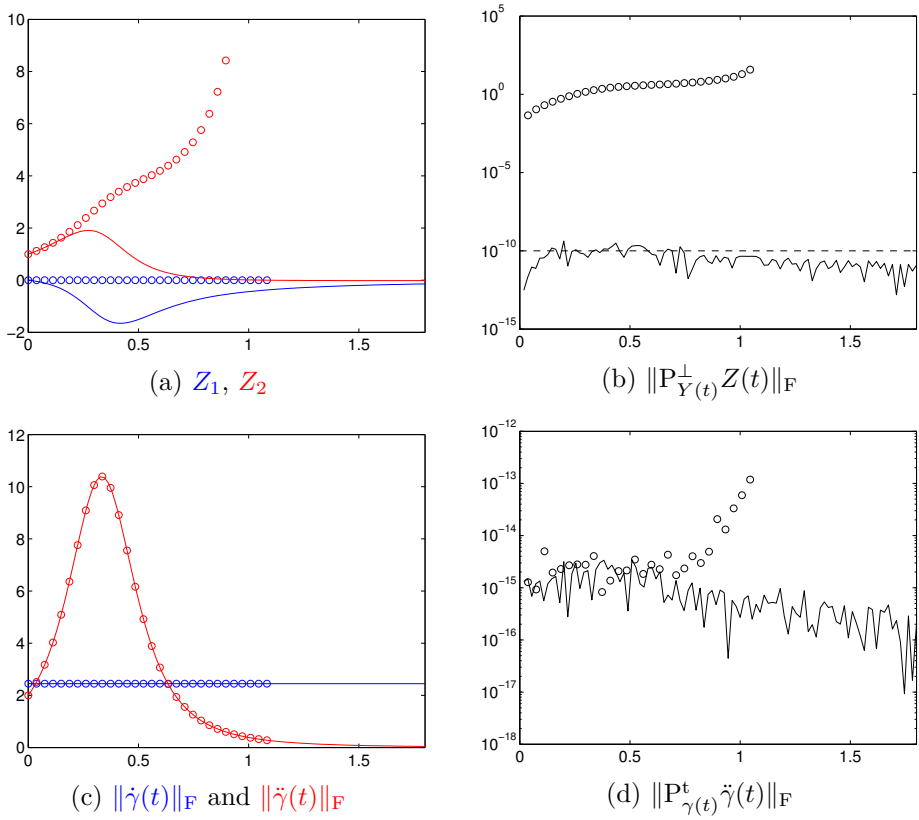
$$R_x^{\text{proj}}(\xi) := \text{P}_{\mathbf{S}_+^{n,p}}(x + \xi) \tag{3.42}$$

where $\text{P}_{\mathbf{S}_+^{n,p}}$ selects the nearest element to $\mathbf{S}_+^{n,p}$ in the Frobenius norm, i.e.,

$$\text{P}_{\mathbf{S}_+^{n,p}} : \mathbf{R}^{n\times n} \to \mathbf{S}_+^{n,p}, \quad X \mapsto \arg\min_{z\in\mathbf{S}_+^{n,p}} \|X - z\|_{\text{F}}.$$

In the more general context of approximation theory, this is also called the *metric projection*.

Orthogonal projections are widespread, both in the general context of retraction-based Riemannian optimization (Manton, 2002; Absil *et al.*, 2008); and in the specific contexts of low-rank solvers for Lyapunov equations (Gugercin *et al.*, 2003; Grasedyck & Hackbusch, 2007) and for low-rank matrix completion (Meka *et al.*, 2010; Goldfarb & Ma, 2010). It will come as no surprise that this choice owes greatly to the fact that we stay as close to the manifold as possible.



(a) Z_1, Z_2 (b) $\|\mathrm{P}^{\perp}_{Y(t)} Z(t)\|_{\mathrm{F}}$

(c) $\|\dot{\gamma}(t)\|_{\mathrm{F}}$ and $\|\ddot{\gamma}(t)\|_{\mathrm{F}}$ (d) $\|\mathrm{P}^{\mathrm{t}}_{\gamma(t)} \ddot{\gamma}(t)\|_{\mathrm{F}}$

Figure 3.4: Numerical properties of the ODEs that define the geodesics on $\mathbf{S}^{2,1}_{+}$ of Figure 3.3.

Since (3.42) is a projection onto an open non-convex set, it may not be everywhere defined nor single-valued on the whole tangent space. Put in the language of differential geometry, this means that mapping $t \mapsto R^{\mathrm{proj}}_x(t\xi)$ may fail to exist for all $t \in \mathbf{R}$, i.e., be complete. Luckily, retractions for Riemannian optimization do not need to be complete; they only have to be defined in some neighborhood. The following theorem gives a characterization of the projection (3.42) and this neighborhood.

Theorem 3.26 (Helmke & Moore (1994, Cor. 2.3)). *Let $A \in \mathbf{S}^n$ have n_+ positive and n_- negative eigenvalues. Let its eigenvalue decomposition be $A = V \operatorname{diag}(\lambda_1, \ldots, \lambda_n) V^T$ with $V \in \mathbf{O}^n$ and $\lambda_1 \geq \ldots \geq \lambda_{n_+} > 0 > \lambda_{n-n_-+1} \geq \ldots \geq \lambda_n$. The best s.p.s.d. approximation of rank p in the Frobenius norm exists if and*

only if $n_+ \geq p$. *One such minimizer is given by*

$$\mathrm{P}_{\mathbf{S}_+^{n,p}}(A) = V \operatorname{diag}(\lambda_1, \ldots, \lambda_p, 0, \ldots, 0) V^T.$$

We shall use this theorem to compute the retraction provided $n_+ \geq p$. Since the domain of R_x^{proj} is restricted to the tangent space, we can further analyze when the retraction is well defined.

Proposition 3.27. *Suppose* $n \geq 2p$. *Let* $\mathrm{P}_x^{\mathrm{t,s}}$ *and* $\mathrm{P}_x^{\mathrm{t,p}}$ *be the projections* (3.19)-(3.20). *Then retraction* R_x^{proj} *defined by* (3.42) *exists for* $\xi \in T_x\mathbf{S}_+^{n,p}$ *in* $x \in \mathbf{S}_+^{n,p}$ *if*

(a) the rank of $\mathrm{P}_x^{\mathrm{t,p}}(\xi)$ *is* $2p$, *or*

(b) $\mathrm{P}_x^{\mathrm{t,s}}(x+\xi)$ *has* p *strictly positive eigenvalues.*

Condition (b) can always be satisfied for ξ *small enough.*

Proof. Elaboration of $x + \xi$ with $x = YY^T$ and $\xi \in T_x\mathbf{S}_+^{n,p}$ as in Prop. 3.15 , gives

$$x+\xi = \begin{bmatrix} Y & Y_\perp \end{bmatrix} T \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} \quad \text{with } T = \begin{bmatrix} I_p + H & N^T \\ N & 0 \end{bmatrix},\ H \in \mathbf{S}^p,\ N \in \mathbf{R}^{(n-p)\times p}.$$

Theorem 3.26 guarantees that the retraction exists if the $n \times n$ matrix T has at least p positive eigenvalues. By Nocedal & Wright (1999, Thm. 16.6), we have $\operatorname{inertia}(T) = \operatorname{inertia}(Z^T(I_p + H)Z) + (l, l, 0)$ with $l = \operatorname{rank}(N)$ and Z a basis for the null space of N. Since $n \geq 2p$, l equals the column rank of N and Z will have $p - l \leq p$ columns.
Suppose condition (a) is true, then N will be full rank, so $p = l$, and T will have at least p strictly positive eigenvalues.
If condition (b) is satisfied, then all p eigenvalues of $I_p + H$ are strictly positive. Now all $p - l$ eigenvalues of $Z^T(I_p + H)Z$ are strictly positive as well and T will have $p - l + l = p$ strictly positive eigenvalues.
The fact that condition (b) can always be satisfied for ξ small enough follows from the observation that $I_p + H \succ 0$ for H small enough. □

Recall from Definition 2.32 that a second-order retraction is a second-order approximation of a geodesic: the curve $t \mapsto R_x(t\xi)$ needs to realize the tangent vector ξ and it has zero acceleration w.r.t. the Levi–Civita connection.

Proposition 3.28. *Retraction* R_x^{proj} *defined by* (3.42) *is a second-order retraction on* $(\mathbf{S}_+^{n,p}, g^E)$.

Proof. We verify the three conditions from Def. 2.31 and Def. 2.32. The property $R_x^{\mathrm{proj}}(0) = x$ is trivially satisfied.
First-order: Smoothness around zero and local rigidity follow from Lemma 2.1 in Lewis & Malick (2008) for projections on general submanifolds, which states that $\operatorname{grad} \mathrm{P}_{\mathbf{S}_+^{n,p}}(x) = \mathrm{P}_x^{\mathrm{t}}$. The existence in a neighborhood of each x follows from Prop. 3.27 for ξ small enough.
Second-order: Slight variation of Miller & Malick (2005, Th. 4.9). We claim that $R_x^{\mathrm{proj}}(t\xi) - \gamma_\xi(t) = O(t^3)$ with $\gamma_\xi(t)$ the geodesic with foot x and direction ξ. Observing that $\gamma_\xi(t) \in \mathbf{S}_+^{n,p}$ for all t and expanding $\gamma_\xi(t)$ in series, we have the following error term

$$\begin{aligned} R_x^{\mathrm{proj}}(t\xi) - \gamma_x(t) &= R_x^{\mathrm{proj}}(t\xi) - \mathrm{P}_{\mathbf{S}_+^{n,p}}\gamma_x(t) \\ &= R_x^{\mathrm{proj}}(t\xi) - \mathrm{P}_{\mathbf{S}_+^{n,p}}[x + t\dot{\gamma}_x(0) + \frac{1}{2}t^2\ddot{\gamma}_x(0) + O(t^3)]. \end{aligned}$$

Since $\dot{\gamma}_\xi(0) = \xi$ and $\ddot{\gamma}_\xi(t)$ belongs to the normal space at $\gamma_\xi(t)$ (see condition (3.25) for the geodesics on $(\mathbf{S}_+^{n,p}, g^{\mathrm{E}})$), the first order terms cancel and the second order term is projected out by $\mathrm{P}_{\mathbf{S}_+^{n,p}}$. What remains in the error term is of third order, hence the retraction R_x^{proj} is of second order. □

Remark 3.29. The projection-based retraction $R_x^{\mathrm{proj}}(\xi)$ can be computed with an eigenvalue decomposition of $x + \xi$; see Theorem 3.26. In general, computing this decomposition with a direct method implies an $O(n^3)$ cost. Luckily, we can exploit the fact that we only need to retract from the tangent space. Suppose we need to retract $\xi = VHV^T + ZV^T + VZ^T$ in the point $x = VDV^T$, then $x + \xi$ can be written as

$$x + \xi = \begin{bmatrix} V & V_p \end{bmatrix} \begin{bmatrix} D + H & R^T \\ R & 0 \end{bmatrix} \begin{bmatrix} V^T \\ V_p^T \end{bmatrix}$$

with $Z = V_pR \in \mathbf{R}^{n\times p}$ a compact QR factorization. Observe that because $V^TZ = 0$, we have that $V^TV_p = 0$ and thus $\begin{bmatrix} V & V_p \end{bmatrix} \in \mathrm{St}^{n\times 2p}$ is orthonormal. Since the Frobenius norm is unitary invariant, it suffices to compute the eigenvalue decomposition of a small $2p \times 2p$ matrix to project $x + \xi$:

$$R_x^{\mathrm{proj}}(\xi) = \mathrm{P}_{\mathbf{S}_+^{n,p}}(x + \xi) = \begin{bmatrix} V & V_p \end{bmatrix} \mathrm{P}_{\mathbf{S}_+^{2p,p}} \left(\begin{bmatrix} D + H & R^T \\ R & 0 \end{bmatrix} \right) \begin{bmatrix} V^T \\ V_p^T \end{bmatrix}.$$

Here, $\mathbf{S}_+^{2p,p}$ is the manifold of s.p.s.d. matrices of size $2p$ and rank p. This brings the dominating costs of this retraction to $O(np^2)$ for the compact QR and $O(p^3)$ for the eigenvalue decomposition.

3.5.2 Truncated Taylor series

The Riemannian optimization algorithms need the Hessians of the objective functions at hand to define the second-order models. Since R_x^{proj} is a second-

order retraction, we can derive the Riemannian Hessian of the objective function f by computing the Euclidean Hessian of $f \circ R_x^{\text{proj}}$, see Theorem 2.37. However, in order to obtain an analytical expression of the Hessian, it is convenient to have a second-order expansion of the retraction as well.

Retraction R_x^{proj} is not readily available in series, but we can construct new expansion that is, by design, given as a Taylor series. If we then make this retraction of second order, its Taylor expansion will match that of R_x^{proj} up to the second-order terms. We will do this by carefully inspecting $x + \xi$ after we split ξ into $\mathrm{P}_x^{\mathrm{t,s}}(\xi) + \mathrm{P}_x^{\mathrm{t,p}}(\xi)$.

Observe that the pseudo-inverse (Golub & Van Loan, 1996, Sect. 5.5.4) of matrix $x = YY^T \in \mathbf{S}_+^{n,p}$, denoted by $x^\dagger$, can be written as

$$x^\dagger = Y(Y^TY)^{-2}Y^T \in \mathbf{S}_+^{n,p}.$$

The second-order Taylor series is then given in the following proposition.

Proposition 3.30. *Let $x \in \mathbf{S}_+^{n,p}$, with $x^\dagger \in \mathbf{S}_+^{n,p}$ its pseudo-inverse, be given. Let $\mathrm{P}_x^{\mathrm{t,s}}$ and $\mathrm{P}_x^{\mathrm{t,p}}$ be the projections (3.19)-(3.20). Then the mapping $R_x^{(2)} : T_x\mathbf{S}_+^{n,p} \to \mathbf{S}_+^{n,p}$ that satisfies*

$$R_x^{(2)} : \xi \mapsto wx^\dagger w^T, \quad \text{with } w = x + \frac{1}{2}\xi^s + \xi^p - \frac{1}{8}\xi^s x^\dagger \xi^s - \frac{1}{2}\xi^p x^\dagger \xi^s, \tag{3.43}$$

where $\xi^s := \mathrm{P}_x^{\mathrm{t,s}}(\xi)$ and $\xi^p := \mathrm{P}_x^{\mathrm{t,p}}(\xi)$, is a second-order retraction on $(\mathbf{S}_+^{n,p}, g^E)$. It has the following Taylor series expansion:

$$R_x^{(2)}(\xi) = x + \xi^s + \xi^p + \xi^p x^\dagger \xi^p + O(\|\xi\|^3). \tag{3.44}$$

Proof. First, we show that $R_x^{(2)}$ is a *retraction*, i.e. a mapping $B_x \to \mathbf{S}_+^{n,p}$ in a neighborhood $B_x \subset T_x\mathbf{S}_+^{n,p}$. Choose $x = YY^T$, then $x^\dagger = Y(Y^TY)^{-2}Y^T$. This allows us to write the components of the tangent vector $\xi = \xi^s + \xi^p$ as $\xi^s = YSY^T$ and $\xi^p = Y_\perp NY^T + YN^TY_\perp^T$. Elaborating the term w in (3.43) and using the relations $\xi^p x^\dagger = Y_\perp NY^T x^\dagger$, $\xi^s x^\dagger \xi^s = YS^2Y^T$ and $\xi^p x^\dagger \xi^s = Y_\perp NSY^T$ we arrive at

$$\begin{aligned} w &= (Y + \frac{1}{2}YS + Y_\perp N - \frac{1}{8}YS^2 - \frac{1}{2}Y_\perp NS)Y^T + YN^TY_\perp^T \\ &= ZY^T + YN^TY_\perp^T \end{aligned}$$

where we introduced the matrix $Z \in \mathbf{R}^{n\times p}$. Since $Y^Tx^\dagger Y = I_p$, we see that $R_x^{(2)}$ can be written as $R_x^{(2)} : \xi \mapsto wx^\dagger w^T = ZZ^T$. Thus, the image of $R_x^{(2)}$ consists of s.p.s.d. matrices of rank not larger than p. Furthermore, there will always be a neighborhood of $T_x\mathbf{S}_+^{n,p}$ that results in matrices Z of full rank k (take S small

enough).
Next, we prove that $R_x^{(2)}$ is of *second-order*. Expanding $wx^\dagger w$ fully up to second-order terms in ξ^s and ξ^p and using the relations $xx^\dagger\xi^s = \xi^s xx^\dagger = \xi^s$ and $xx^\dagger\xi^p + \xi^p xx^\dagger = \xi^p$ we see that many of the second-order terms cancel. Finally, we obtain the series (3.44) as

$$R_x^{(2)}(\xi) = x + \xi^s + \xi^p + \xi^p x^\dagger \xi^p + O(\|\xi\|^3).$$

From this $R_x^{(2)}(0) = x$ and local rigidity (first-order) are obvious. Since $\xi^p x^\dagger \xi^p = Y_\perp N^2 Y_\perp^T \in N_x \mathbf{S}_+^{n,p}$, zero acceleration (second-order) is proved. □

By taking only the first-order terms in (3.43), one obtains a first-order retraction.

Corollary 3.31. *Let $x \in \mathbf{S}_+^{n,p}$, with $x^\dagger \in \mathbf{S}_+^{n,p}$ its pseudo-inverse, be given. Let $\mathrm{P}_x^{\mathrm{t,s}}$ and $\mathrm{P}_x^{\mathrm{t,p}}$ be the projections (3.19)-(3.20). Then the mapping $R_x^{(1)} : T_x\mathbf{S}_+^{n,p} \to \mathbf{S}_+^{n,p}$ that satisfies*

$$R_x^{(1)} : \xi \mapsto wx^\dagger w^T, \quad \text{with } w = x + \frac{1}{2}\xi^s + \xi^p, \tag{3.45}$$

where $\xi^s := \mathrm{P}_x^{\mathrm{t,s}}(\xi)$ and $\xi^p := \mathrm{P}_x^{\mathrm{t,p}}(\xi)$, is a first-order retraction on $(\mathbf{S}_+^{n,p}, g^E)$.

3.5.3 Orthographic projection

In addition to the previous retractions, we mention one more retraction based on a recently developed framework by Absil & Malick (2010). It allows to construct projection-like retractions on general matrix manifolds. One of these constructions is the orthographic retraction, so called because of the relation with the orthographic projection on the sphere. If we specialize Proposition 24 of Absil & Malick (2010) to the symmetric case, we obtain a second-order retraction on $\mathbf{S}_+^{n,p}$.

Writing the matrix x and the tangent vector ξ in the short-hand notations of Remark 3.3 and 3.17, i.e.,

$$x = VDV^T, \tag{3.46}$$

$$\xi = VHV^T + ZV^T + VZ^T, \tag{3.47}$$

we obtain the following Theorem.

Theorem 3.32. *Let $x \in \mathbf{S}_+^{n,p}$ and $\xi \in T_x\mathbf{S}_+^{n,p}$ satisfy (3.46) and (3.47), respectively. Then the mapping $R_x^{\mathrm{ograph}} : T_x\mathbf{S}_+^{n,p} \to \mathbf{S}_+^{n,p}$ given by*

$$R_x^{\mathrm{ograph}} : \xi \mapsto XX^T \quad \text{with } X = V(D+H)^{1/2} + Z(D+H)^{-1/2}, \tag{3.48}$$

is a second-order retraction on $(\mathbf{S}_+^{n,p}, g^E)$.

Due to the presence of the matrix square root $(D + H)^{1/2}$, this retraction is well-defined provided $D + H \succ 0$. In can also be computed efficiently since the dominating costs are a square root of $p \times p$ matrix and the matrix multiplications.

By virtue of Absil & Malick (2010, Prop. 24), we already know that R_x^{ograph} is a second-order approximation of the geodesic. However, it is still instructive to verify this explicitly. Working out the product XX^T in (3.48), we get

$$\begin{aligned} R_x^{\text{ograph}}(\xi) &= V(D+H)V^T + ZV^T + VZ^T + Z(D+H)^{-1}Z^T \\ &= VDV^T + VHV^T + ZV^T + VZ^T + ZD^{-1}Z^T \\ &\quad - ZD^{-1}HD^{-1}Z^T + ZO(\|H\|^2)Z^T, \end{aligned}$$

where we used the expansion $(D + H)^{-1} = D^{-1} - D^{-1}HD^{-1} + O(\|H\|^2)$. One can immediately recognize x and ξ in the first four terms. Recall that we defined $\xi^p := \mathrm{P}^{\mathrm{t,p}}(\xi) = ZV^T + VZ^T$ and $x^\dagger := VD^{-1}V^T$. Together with the identity $ZD^{-1}Z^T = \xi^p \xi^\dagger \xi^p$ and dropping the third order terms in ξ, we obtain

$$R_x^{\text{ograph}}(\xi) = x + \xi + \xi^p \xi^\dagger \xi^p + O(\|\xi\|^3).$$

Comparing this with the expansion of $R_x^{(2)}$ in (3.44), we can conclude that R_x^{ograph} is indeed of second order.

3.5.4 Numerical comparison

Let ξ be an arbitrary fixed tangent vector. Then, we have constructed five different retractions on $(\mathbf{S}_+^{n,p}, g^{\mathrm{E}})$:

(1) $\gamma_x(t)$, the geodesic of Prop. 3.25;

(2) $R_x^{\text{proj}}(t\xi)$, the orthogonal projection (3.42);

(3) $R_x^{(1)}(t\xi)$, the first-order Taylor series (3.45);

(4) $R_x^{(2)}(t\xi)$, the second-order Taylor series (3.43);

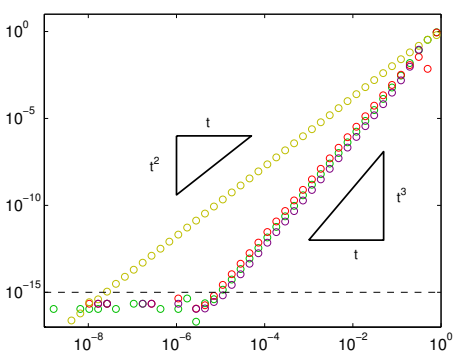
(5) $R_x^{\text{ograph}}(t\xi)$, the orthographic projection (3.48).

In this section, we investigate their behavior numerically for small and large values of t.

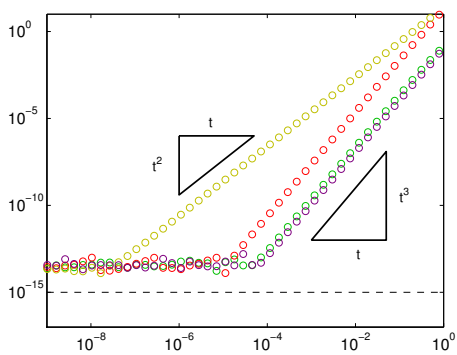
Behavior for small t. The last four retractions from above can be seen as an approximation of the geodesic $\gamma_x(t)$. Except for $R_x^{(1)}$, they are all accurate up to second-order. We verify this numerically.

Let $\gamma_x(t)$ be a geodesic with foot $\gamma_x(0) = x$ and initial direction $\dot{\gamma}_x(0) = \xi$ and let $R_x(t\xi)$ be any of the retractions (2)–(5) from above with the same initial conditions. In Figure 3.5, we have plotted the difference $\|\gamma_x(t) - R_x(t\xi)\|_{\mathrm{F}}$ for these four retractions and for two different choices of (x, ξ).

The first choice, Figure 3.5(a), corresponds to the small case of Section 3.4.4, while the second choice, Figure 3.5(b), is a random problem of size $n = 20$ and rank $k = 5$. In both cases we integrated the ODE of Prop. 3.25 numerically with `ode45` from MATLAB. The absolute and relative error tolerances were 10^{-15} and 10^{-13} respectively. This absolute tolerance is also indicated in the figures. It is evident from both figures that retraction $R_x^{(1)}$ is of first order, while the other three are all of second order.



(a) $\mathbf{S}_+^{2,1}$ with initial condition from Section 3.4.4.

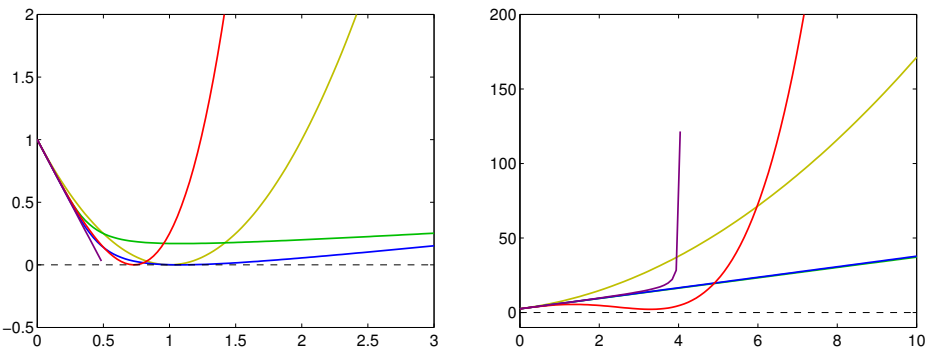


(b) $\mathbf{S}_+^{20,5}$ with normalized random initial condition.

Figure 3.5: The norm of the difference between the geodesic $\gamma_x(t)$ and four different retractions $R_x(t\xi)$ in function of t. The retractions are R_x^{proj} as ○, $R_x^{(1)}$ as ○, $R_x^{(2)}$ as ○ and R_x^{ograph} as ○.

Behavior for large t. Looking at Figure 3.5(b), it seems that R_x^{ograph} is preferable to all the other retractions: it is of second-order, it is the most accurate when $t \to 0$ and it is cheap to compute. However, while the behavior for $t \to 0$ is important, for the numerical performance of the Riemannian algorithms, the long-term behavior is equally important. If a retraction is well-defined in a large neighborhood, it is, for practical applications, almost as good as a complete curve. For numerical reasons, it is in addition important that the curve is well-conditioned, by which we mean that the entries do not become too small and/or too big.

We have investigated this numerically for the same curves as above. Figure 3.6 shows matrix element $\gamma_{1,1}$ of these curves $\gamma(t)$. For both figures, the orthographic projection R_x^{ograph} leaves $\mathbf{S}_+^{n,p}$ quite soon; around $t = 0.5$ and $t = 4$ for Figure 3.6(a) and (b), respectively. In both cases, $D + tH$ in (3.48) was no longer positive definite. On the other hand, the retractions $R_x^{(1)}$, $R_x^{(2)}$, R_x^{proj} and the geodesic are defined for the whole time interval. The Taylor series, however, show the typical blow-up of polynomials for $t \to \infty$, whereas the geodesic and R^{proj} have a nicer long-term behavior. Visually, R_x^{proj} seems to capture the behavior of the geodesic much better than any of the three other retractions. In fact, both curves are indistinguishable on Figure 3.6(b).



(a) $\mathbf{S}_+^{2,1}$ with initial condition from Section 3.4.4.

(b) $\mathbf{S}_+^{20,5}$ with normalized random initial condition.

Figure 3.6: Element $\gamma_{1,1}$ of the geodesic $\gamma_x(t)$ and four different retractions $R_x(t\xi)$ in function of t. The geodesic is shown as — and the retractions R_x^{proj} as —, $R_x^{(1)}$ as —, $R_x^{(2)}$ as — and R_x^{ograph} as —.

Suitability for optimization. Which retraction will be most suited for the retraction-based optimization algorithms in the next chapters? This depends on the application at hand. Regarding the derivation of the Riemannian Hessians, it is preferable to use a second-order retraction that is available as a series. For this we will use $R_x^{(2)}$. On the other hand, for the actual implementation of the retraction, it seems that R_x^{proj} is to be preferred. This retraction behaves well, both for large as for small t. The curve is much better conditioned than the others, which should lead to a more robust numerical procedure.

3.6 Conclusions

This chapter was devoted to the geometry of $\mathbf{S}_+^{n,p}$ as an embedded submanifold of the real matrices. We choose a simple Euclidean metric for this geometry which led to lean expressions for gradients, Hessians and retractions. Since this geometry will be applied to the large-scale optimization problems of the next chapters, we spent considerable attention to the efficient expressions of the geometric objects.

In addition, we derived the geodesics and their first- and second-order approximations. All these curves were shown not to be complete, although some retractions are numerically better behaved than others. Furthermore, their suitability for the forthcoming practical problems was assessed in some numerical experiments.

4

Low-rank solutions of Lyapunov equations

In the previous chapter, we have laid the foundations of our Riemannian optimization approach for rank-constrained matrix problems by describing an embedded geometry for $\mathbf{S}_+^{n,p}$. In this chapter, we will apply it to a practical application: the approximation of the solution of a Lyapunov matrix equation by a low-rank matrix. The black-box approach of the existing Riemannian algorithms, while effective in practice, does not lead to a solver that is competitive with the state-of-the-art. We will solve this by designing a preconditioner.

The results of this chapter concerning the minimization of the energy norm are mostly based on Vandereycken & Vandewalle (2010). The unpublished additions are the derivations concerning the residual norm (for the most part, Sections 4.4.2 and 4.5.2).

4.1 Introduction

The subject of this chapter is the problem of approximating solutions of large-scale matrix equations by iterative methods. We will focus on the *generalized Lyapunov equation*

$$AXM^T + MXA^T = C \tag{4.1}$$

with given nonsingular matrices $A, M \in \mathbf{R}^{n\times n}$. Furthermore, we assume that $A + A^T \succ 0$, $M + M^T \succ 0$ and $C \in \mathbf{S}^n_+$. Under these assumptions, equation (4.1) has a unique solution, and this solution X is symmetric and (at least) positive semidefinite.

A special case of (4.1) is the *standard Lyapunov equation*, or simply, the Lyapunov equation,

$$AX + XA^T = C, \tag{4.2}$$

which coincides with (4.1) by setting $M = I$.

4.1.1 Solvability

In this section, we collect some well-known properties about the solutions of generalized Lyapunov equations.

Equation (4.1) is linear. This can be seen directly by verifying the definitions of a linear operator. Since it is linear, we can represent equation (4.1) as a system of linear equations represented by a single matrix. The isomorphism $\operatorname{vec} : \mathbf{R}^{n\times n} \to \mathbf{R}^{n^2}$ of (B.5) gives such a possible representation.

Let $x = \operatorname{vec}(X)$ and $c = \operatorname{vec}(C)$ be the vectorized unknown and right-hand side. Then, by applying property (B.6), we obtain

$$\begin{aligned} & AXM^T + MXA^T = C \\ \iff \quad & \operatorname{vec}(AXM^T + MXA^T) = \operatorname{vec}(C) \\ \iff \quad & \mathcal{L}\, x = c, \end{aligned}$$

where we introduced the system matrix

$$\mathcal{L} := A \otimes M + M \otimes A. \tag{4.3}$$

Thanks to the special Kronecker structure of $\mathcal{L}$, its singularity—and hence, the solvability of (4.1)—can be characterized in terms of the generalized eigenvalues of the matrix pencil $A - \lambda M$. We follow the standard conventions regarding generalized eigenvalues and eigenvectors as detailed in Section B.1.1.

Theorem 4.1. *Chu (1987, Thm. 1) Let $\lambda_i \in \lambda(A, M)$ be a generalized eigenvalue of the pencil $A - \lambda M$. Then the Lyapunov equation (4.1) has a unique solution if and only if*

(a) the pencil $A - \lambda M$ is regular and all λ_i are finite,

(b) $\lambda_i + \lambda_j \neq 0$ *for any pair* i, j.

Singularity of A and/or M implies singularity of $\mathcal{L}$. *Since we are only interested in Lyapunov equations with unique solutions, we will assume that both* A *and* M *are nonsingular*, as indicated in the introduction.

There exist well-known transformations to turn nonsingular generalized Lyapunov equations into standard Lyapunov equations. The first one is the following:

$$\begin{aligned} & AXM^T + MXA^T = C \\ \iff \quad & M^{-1}(AXM^T + MXA^T)M^{-T} = M^{-1}CM^{-T} \\ \iff \quad & \bar{A}X + X\bar{A}^T = \bar{C} \end{aligned} \tag{4.4}$$

with $\bar{A} := M^{-1}A$ and $\bar{C} := M^{-1}CM^{-T}$.

When $A, M \succ 0$, the previous transformation turns a symmetric system into a nonsymmetric one. A symmetry-preserving transformation that uses the matrix square root $M^{1/2}$ of M is the following:

$$\begin{aligned} & AXM + MXA = C \\ \iff \quad & M^{-1/2}(AXM + MXA)M^{-1/2} = M^{-1/2}CM^{-1/2} \\ \iff \quad & \bar{A}\bar{X} + \bar{X}\bar{A}^T = \bar{C} \end{aligned} \tag{4.5}$$

with $\bar{A} := M^{-1/2}AM^{-1/2} \succ 0$, $\bar{X} := M^{1/2}XM^{1/2}$ and $\bar{C} := M^{-1/2}CM^{-1/2}$.

We stress that the transformation of a generalized Lyapunov equation to a standard one is mainly of theoretical interest. For most large-scale applications, computing and possibly storing M^{-1} or $M^{-1/2}$ will be prohibitively expensive.

4.1.2 Positivity of the solution

Since the main focus of this thesis is approximation on the manifold $\mathbf{S}_+^{n,p}$, it is important to know when equation (4.1) has a positive (semi)definite solution. This is answered in the following well-known theorem; see, e.g., Snyders & Zakai (1970, Thm. 2.2) and Penzl (1998, Thm. 3).

Theorem 4.2. *Let* A, M *be nonsingular and let* $\mathrm{Re}(\lambda_i) > 0$ *for all* $\lambda_i \in \lambda(A, M)$. *Let* X *be the unique solution of* (4.1)*, then we have*

(a) $C \in \mathbf{S}_{++}^n \implies X \in \mathbf{S}_{++}^n$,

(b) $C \in \mathbf{S}_+^n \Longrightarrow X \in \mathbf{S}_+^n$.

When A, M are nonsingular and $\mathrm{Re}(\lambda_i) > 0$, the Lyapunov equation (4.1) is termed an *anti-stable generalized Lyapunov equation*[1]. By our assumptions on A and M in the introduction, this is always the case. Hence, every X of (4.1) will be at least an s.p.s.d. matrix.

The relation (b) of Thm. 4.2 can be made stronger if we take into account the controllability of $M^{-1}A$ w.r.t. C. We refer to App. B.1.5 for the definition of controllability.

Proposition 4.3. *Let A, M be nonsingular and let $\mathrm{Re}(\lambda_i) > 0$ for all $\lambda_i \in \lambda(A, M)$. Let X be the unique solution of (4.1), then we have*

$$C \in \mathbf{S}_+^n \Longrightarrow X \in \mathbf{S}_{++}^n$$

if and only if $(M^{-1}A, M^{-1}C)$ is controllable.

Proof. Apply Snyders & Zakai (1970, Cor. 4.3) to the transformed eq. (4.4). □

It is not difficult to construct right-hand sides that do not satisfy the conditions of Prop. 4.3. Let $v \in \mathbf{R}^n$ be a real generalized eigenvector of the pencil $A - \lambda M$ with corresponding real eigenvector λ. If $b = Mv$, then $(M^{-1}A, M^{-1}bb^T)$ is not controllable for $n > 1$ since the controllability matrix is always of rank 1. The solution of the Lyapunov equation (4.1) is given explicitly as

$$X = \frac{1}{2\lambda} vv^T,$$

which is indeed only positive semidefinite. Random matrices A and C will be controllable with probability one and, unless there is some special structure in A or C, this is also the case for most realistic applications. (However, since controllability is a crucial property in control theory, it is overly simplistic to assume that most realistic applications are controllable.)

When $\mathcal{L}$ is singular, the conditions in Snyders & Zakai (1970) for positiveness of the solution become more complicated. For a systematic treatment of the solvability and positivity of generalized Lyapunov equations with singular M, we refer to Stykel (2002a,b). Since we assume M nonsingular, the results above suffice.

[1] The stable Lyapunov equation is defined as $AXM^T + MXA^T = C$ with $\mathrm{Re}(\lambda_i) < 0$ for all $\lambda_i \in \lambda(A, M)$.

4.2 Low-rank approximations for large-scale problems

Lyapunov equations are of significant importance in control theory (Benner, 2006), model reduction (Moore, 1981) and stochastic analysis of dynamical systems (Scheerlinck *et al.*, 2001); see Antoulas (2005) for a general overview. A recurring pattern is that the solution X of the Lyapunov equation (4.1) can be associated with a Gramian of the linear time-invariant (LTI) system

$$M\frac{\mathrm{d}x(t)}{\mathrm{d}t} = Ax(t) + Bu(t), \quad x(0) = x_0,$$
$$y(t) = Cx(t) \tag{4.6}$$

with state x, input u and output y. These Gramians capture useful information about the energy of a system such as the controllability, observability and covariance matrices. In general, this matrix X is dense even if the system is modeled by sparse matrices.

In this section, we briefly introduce an important application of Lyapunov equations: model reduction of large-scale LTI systems. Since this technique requires the solution of possibly very large Lyapunov equations, we review the traditional solution methods for Lyapunov equations and their shortcomings for large-scale systems. Finally, we discuss a family of recently proposed Lyapunov solvers that are especially designed for large-scale systems. Our forthcoming Riemannian method shares the same rationale of these methods: computing low-rank approximations for the solutions of Lyapunov equations.

4.2.1 Model reduction by balanced truncation

Suppose we wish to approximate the LTI system (4.6) by another LTI system,

$$\widehat{M}\frac{\mathrm{d}\widehat{x}(t)}{\mathrm{d}t} = \widehat{A}\,\widehat{x}(t) + \widehat{B}\,\widehat{u}(t), \quad \widehat{x}(0) = \widehat{x}_0,$$
$$\widehat{y}(t) = \widehat{C}\,\widehat{x}(t) \tag{4.7}$$

where $\widehat{M}, \widehat{A} \in \mathbf{R}^{\widehat{n}\times\widehat{n}}$ with $\widehat{n} < n$. Since the matrices in system (4.7) are smaller, this system will only be an approximation. Ideally, we want it to be the best approximation of (4.6) in some norm amongst all LTI systems of size $\widehat{n}$.

Model reduction by balanced truncation (Moore, 1981) will compute an almost optimal approximation in the so-called H_∞-norm. It is based on the controllability $\mathcal{P}$ and observability $\mathcal{Q}$ Gramians, given as the solutions of the following Lyapunov equations

$$A\mathcal{P}M^T + M\mathcal{P}A^T = -BB^T \quad \text{and} \quad A^T\mathcal{Q}M + M^T\mathcal{Q}A = -C^TC.$$

Using these Gramians, the balanced truncation method consists in transforming the system (4.6) into a balanced form for which the new controllability and observability Gramians become diagonal and equal. The reduced-order system (4.7) is then formed by truncating those states that are both difficult to reach and to observe (Moore, 1981).

In the context of balanced truncation, large-scale Lyapunov equations arise naturally when the system is modeled by a system of partial differential equations (PDE). Semi-discretization in space by, e.g., the finite element method (FEM) results in a Lyapunov equation with a sparse system matrix A and a sparse mass matrix M. The dimension n of these matrices is usually very large. A major challenge when solving matrix equations for such a large-scale problem is that the dense matrix X has n^2 entries. Storing this matrix when $n \gg 1000$ will be problematic, let alone solving the Lyapunov equation itself.

4.2.2 Limited applicability of standard methods

Matrix $\mathcal{L}$ as defined in (4.3) is of dimension $n^2 \times n^2$. Solving the corresponding system of linear equations by standard Gaussian elimination requires $O(n^6)$ work and is therefore not practical unless n is very small, say, $n < 50$. Even when A and M are sparse, and sparse elimination techniques are used, the vectorization approach quickly shows its limits for realistic applications.

This limited applicability of the vectorization approach is typical for other matrix equations also. The origin of this problem lies in the fact that matrix equations are defined on the tensor product space $\mathbf{R}^n \otimes \mathbf{R}^n$, whereas the physical system, modeled by A and M, is defined on $\mathbf{R}^n$. This is termed the *curse of dimensionality* (Bellman, 1957) and it requires special consideration regarding discretizations and solvers.

Luckily, for the Lyapunov equations there exist many solvers that can exploit the specific structure of equation (4.1). The archetypical method for solving the standard Lyapunov equation

$$AX + XA^T = C \tag{4.8}$$

is that of Bartels & Stewart (1972). Let $A = QTQ^T$ be a real Schur decomposition (Golub & Van Loan, 1996, Thm. 7.4.1) with $Q \in \mathbf{O}^n$ and $T \in \mathbf{R}^{n\times n}$ upper quasi-triangular. Then solving system (4.8) is equivalent to solving

$$T\widetilde{X} + \widetilde{X}T^T = Q^T C Q \tag{4.9}$$

with $\widetilde{X} = Q^T X Q$. Since T is quasi upper-triangular with 1×1 or 2×2 blocks on the diagonal, all entries of $\widetilde{X}$ can be explicitly solved in a particular order when the 2×2 blocks are treated separately. Together with the matrix multiplications

by Q and the cost of the Schur decomposition, this amounts to a total of $O(n^3)$ work for computing the $n \times n$ matrix X. Clearly, this makes this approach suitable only for small-scale problems up to $n \simeq 1000$.

Essentially all direct methods for the Lyapunov equation have this $O(n^3)$ complexity. On the other hand, optimal iterative methods, like multigrid, can compute the solution with a computational effort that is proportional to the number of unknowns. Since there are n^2 unknowns, this results in an $O(n^2)$ complexity. This allows us to solve systems that are slightly larger, say $n \simeq 10^4$, but this is still only medium-scale. Discretizations of realistic PDEs can easily have more than $n = 10^6$ unknowns, so even the best (classical) iterative methods that take $O(n^2)$ work will still be out of reach.

4.2.3 The low-rank property

In general, the curse of dimensionality is unavoidable and cannot be solved. However, many problems in higher dimensions are intrinsically somehow of lower dimension, which means that, in principle, the unknown can be approximated with a significantly reduced number of parameters. The problem is finding a particular set of basis functions that reveal this lower dimensional structure. This can be done a priori, like in sparse grids, or in a black-box approach during the iteration, like in Quasi Monte-Carlo methods.

In the context of matrix problems, another black-box technique is approximating the solution by a low-rank matrix of rank $k \ll n$. In this way the number of unknowns is reduced to $O(nk)$. If one can compute this low-rank approximation in $O(nk^c)$ flops with c small, then solving such a large-scale matrix equation becomes feasible. There already exist a significant number of low-rank Lyapunov solvers using this principle (see below) that are very performant in practice. Our method is based on a new principle, namely that of Riemannian optimization.

Clearly, low-rank approximations are not always suitable. Although it is reasonable to expect that the quality of the approximation will improve with growing rank, this rank can be very large, maybe too large to be of any practical use. Indeed, consider the equation $AX + XA = 2A$ with solution $X = I$, the identity matrix. Any low-rank approximation will unavoidably be very poor. However, there are a significant number of applications where the solution does exhibit the so called *low-rank property*: the eigenvalues of the matrix X have an exponential decay and the accuracy of the best low-rank approximation increases rapidly with growing rank. This has been studied in Penzl (2000); Sorensen & Zhou (2002); Antoulas *et al.* (2002); Grasedyck (2004) for the case of $M = I$ and a low-rank matrix C in (4.1). There, bounds have been proposed that depend on the spectrum of A and, to some extent, explain the low-rank phenomenon. If we consider, e.g., a matrix A with condition number κ, these bounds suggest that we can approximate X with a

relative accuracy of ϵ using a rank $k = O(\log(\kappa)\log(1/\epsilon))$, see Grasedyck (2004, Remark 1).

Most of these bounds are however of limited applicability since they are of an asymptotic nature that overestimate the rank by far and/or require constants which are difficult to compute; see, e.g., Grasedyck (2004, Table I). Furthermore, in case of non-symmetric matrices A, M they become considerably less tight. Luckily, for most applications, one does not need to know how the rank behaves in function of the accuracy. In practice, one can try to compute a low-rank approximation, and, depending on some measure of the error, decide that the current low-rank approximation is accurate enough. The theoretical bounds for the eigenvalue decay of X then give a reasoning why this is indeed a sensible strategy. In Table 4.1, we give a typical example of the rank needed to approximate the solution of a Lyapunov equation within a certain relative error. The equation used a rank one, random right-hand side matrix C.

n	10^{-4}	10^{-6}	10^{-8}	10^{-10}
256	6	8	10	12
576	7	9	11	14
1024	7	10	12	15

Table 4.1: Minimal rank needed to approximate the solution of Lyapunov equation (4.1) with a rank one matrix C. Matrix A is the discretization of a one-dimensional diffusion operator with variable coefficients and $M = I$.

4.2.4 Existing low-rank Lyapunov solvers

If we assume that there exists a good low-rank approximation for the solution X, one can devise methods to compute it. Most of the existing low-rank solvers in the literature fit the principle of reformulating a well-known iterative method to the low-rank case. The inspiration of these iterative methods can be

(a) the ADI and Smith method, see Penzl (1999); Li & White (2004);

(b) Krylov subspace techniques, see Saad (1990); Jaimoukha & Kasenally (1994); Jbilou & Riquet (2006); Simoncini (2007); or,

(c) the power method, see Hodel *et al.* (1996); Vasilyev & White (2005).

(d) a hybrid method that combines some of the methods from above; see Nong & Sorensen (2009); Jbilou (2010); Benner & Saak (2010).

The algorithms from (a)–(b) above are based on an exact reformulation of the listed methods. In each step i, they perform the iteration on a factor Y_i instead of on the

whole iterate $X_i = Y_i Y_i^T$. During each iteration, a number of columns are added to Y_i and so, the rank of the approximation X_i will grow. If convergence is fast, the approximation will have low rank. In this case, these solvers can be very efficient, since they are relatively cheap per iteration. If convergence is slow however, there is not much that can be done, except to keep on iterating and possibly compressing iterates to lower rank along the way; see Gugercin *et al.* (2003). The methods from (c) are based on the power method for computing a dominant subspace of X directly. Since, X is not available, they rely on an approximation of the power method to obtain a practical method. In principle, they can keep the rank fixed during the iteration, but a small rank can lead to erratic convergence. Another technique to accelerate convergence and possibly keeping the ranks bounded is by combining some of the methods (a)–(c) from above.

A major problem regarding these solvers is that they are very specific to the Lyapunov equation and the ability to reformulate the iteration in terms of low rank factors. In general, one would like to use other, more established iterative techniques for solving the Lyapunov equation in low-rank; in particular, preconditioning. However, there seems to be little room to directly incorporate preconditioning compatible with the structure of a Lyapunov equation, although the method in Simoncini (2007) can be viewed as preconditioning with A^{-1}. Another problem is that these algorithms need to form the product of A with a square root of C. For general matrices, this square root can be costly to compute. Therefore, the algorithms are usually only applied to the case where $C = BB^T$ as this gives a trivial square root B.

A quite different solver is the one in Grasedyck & Hackbusch (2007). It is based on a so-called low-rank arithmetic: if the addition of two matrices of rank k is followed by a projection onto the set of rank k matrices, one can perform a standard solver for linear systems, like multigrid, efficiently with low-rank matrices. In each iteration, the rank of the iterate will temporarily grow, only to be truncated back to low rank. When the ranks of these iterates are not too large, the reduction to lower rank can be done relatively efficient by, e.g., a compact SVD. This way, the solver of Grasedyck & Hackbusch (2007) circumvents the problem of slow convergence by doing a geometric multigrid iteration with low-rank matrices. It has the benefit that it can combine an optimal and fast solver with a low-rank solution, provided that this low-rank arithmetic does not destroy convergence.

The downside of this algorithm is that the convergence of the method is rather sensitive to the ranks chosen at each level of the multigrid algorithm. Furthermore, geometric multigrid is usually used with better smoothers and acting as a preconditioner, or exchanged in favor of algebraic multigrid. It is far from clear however, how to formulate these classical iterations in terms of low-rank factors. The potential optimality of this low-rank multigrid solver for Lyapunov equations is promising though. In the next chapter, we will therefore explain its properties in more detail and see how it can be extended to a Riemannian algorithm.

4.3 Basic principles of the Riemannian method

Contrary to the existing low-rank solvers for Lyapunov equations, the method we propose is not based on a reformulation of an existing iterative method but on the principle of Riemannian optimization. We will find a low-rank approximation of X in (4.1) by minimizing a suitable objective function on the manifold $\mathbf{S}_+^{n,p}$. Recall that this manifold is the set of s.p.s.d. matrices of rank p,

$$\mathbf{S}_+^{n,p} = \{x \in \mathbf{S}^n \mid x \succeq 0,\ \operatorname{rank}(x) = p\}. \tag{4.10}$$

The objective function should quantify the error of the approximation. Let X denote the solution and $x \in \mathbf{S}_+^{n,p}$ its approximation. Then it is clear that the Frobenius norm of the error,

$$f_F : \mathbf{S}_+^{n,p} \to \mathbf{R},\ x \mapsto \|X - x\|_{\mathrm{F}}^2 = \operatorname{tr}[(X-x)(X-x)], \tag{4.11}$$

would be an ideal candidate for an objective function. It has one fatal flaw however. Since the solution X is unknown, it can not be computed.

We therefore propose the following two objective functions. The first is

$$f_E : \mathbf{S}_+^{n,p} \to \mathbf{R},\ x \mapsto \operatorname{tr}(xAxM) - \operatorname{tr}(xC).$$

We will show in Section 4.4.1 that this function is related to the energy norm of the error $X - x$. Furthermore, since it uses a norm that is derived from the matrices A and M, these matrices have to be symmetric positive definite.

The second option for the objective function is always applicable. It is related to the Frobenius norm of the residual $R(x) = AxM^T + MxA^T - C$:

$$f_R : \mathbf{S}_+^{n,p} \to \mathbf{R},\ x \mapsto \operatorname{tr}[AxM^T(MxA^T + AxM^T)] - 2\operatorname{tr}(AxM^TC).$$

Some properties of this function are discussed in Section 4.4.2.

Now that we have chosen the objective functions, we can formulate the main problem:

$$\min_x f(x) \qquad \text{subject to } x \in \mathbf{S}_+^{n,p}, \tag{4.12}$$

We will solve (4.12) by the Riemannian Trust-Region method as detailed in Section 2.8.1 and Algorithm 1, which uses a second-order model of these objective functions. This requires that we derive the Riemannian gradient and the Riemannian Hessian of f_E and f_R. We will do this in Section 4.5, but first, we discuss some properties of these objective functions.

4.4 The objective functions

In this section, we show why the objective functions defined above are a good choice to obtain low-rank approximations for the Lyapunov equation. Furthermore, we explain the applicability and the difference between the two.

4.4.1 Based on the energy norm: f_E

Recall that we defined the following objective function

$$f_E : \mathbf{S}_+^{n,p} \to \mathbf{R}, \ x \mapsto \operatorname{tr}(xAxM) - \operatorname{tr}(xC). \tag{4.13}$$

We claim that this function is related to the energy norm if (4.1) is regarded as a standard linear system.

Like we explained in Section 4.1.1, the Lyapunov equation (4.1) can be written as a standard linear equation by means of vectorization. Let $\operatorname{vec}(\cdot)$ denote the operator that makes a vector from a matrix by column-wise stacking and let $\otimes$ denote the Kronecker product, then we established that (4.1) is equivalent to

$$\mathcal{L} \operatorname{vec}(X) = \operatorname{vec}(C), \quad \text{with } \mathcal{L} = A \otimes M + M \otimes A. \tag{4.14}$$

In addition, recall that the $\operatorname{vec}(\cdot)$ operator defines an isomorphism between the Euclidean spaces $\mathbf{R}^{n\times n}$ and $\mathbf{R}^{n^2}$ where the inner products relate to each other by

$$\operatorname{tr}(X^T Y) = \operatorname{vec}(X)^T \operatorname{vec}(Y).$$

Further on, we will need the $\mathcal{L}$-norm; this is the weighted Euclidean norm

$$\|\cdot\|_{\mathcal{L}} := \sqrt{g^{\mathrm{E}}(\cdot,\cdot)_{\mathcal{L}}} \quad \text{with } g^{\mathrm{E}}(u,v)_{\mathcal{L}} := u^T \mathcal{L} v. \tag{4.15}$$

In order for this norm to make sense, $\mathcal{L}$ as defined in (4.14), should be symmetric and positive definite. This is always the case based on the following assumptions on A and M.

Proposition 4.4. *Suppose $A, M \succ 0$, then $\mathcal{L} \succ 0$.*

Proof. From the symmetry of A and M, we have that $\mathcal{L} = \mathcal{L}^T$. To prove that $\mathcal{L}$ is positive definite, we show that $\lambda(A \otimes M + M \otimes A) > 0$. From Penzl (1998), we know that the eigenvalues of $A \otimes M + M \otimes A$ consist of the set of all numbers $\mu_{i,j} := \mu_i + \mu_j$, where μ_i, μ_j are the eigenvalues of the symmetric/positive-definite pencil $A - \lambda M$. Since $A \succ 0$, the eigenvalues $\mu_{i,j}$ of this pencil are strictly positive (Stewart, 2001, Th. 3.4.2), and so $\mu_i + \mu_j > 0$. □

From now on, whenever we use f_E, we will always assume that A and M are s.p.d. matrices.

Now we can work out the $\mathcal{L}$-norm of the error $E = X - x$ of $x \in \mathbf{S}_+^{n,p}$, with X the true solution of (4.1). Since x is symmetric, E is symmetric too and using the relations from above, we get

$$\begin{aligned}\|\mathrm{vec}(E)\|_{\mathcal{L}}^2 &= \mathrm{vec}(E)^T (A \otimes M + M \otimes A)\,\mathrm{vec}(E),\\ &= \mathrm{vec}(E)^T\,\mathrm{vec}(MEA) + \mathrm{vec}(E)^T\,\mathrm{vec}(AEM),\\ &= 2\,\mathrm{tr}(EMEA),\end{aligned}$$

where we used some of the properties in B.1.2 for traces. Inserting $E = X - x$, we continue

$$\begin{aligned}\|\mathrm{vec}(E)\|_{\mathcal{L}}^2 &= 2\,\mathrm{tr}[(X-x)M(X-x)A],\\ &= 2\,\mathrm{tr}(XMXA) - 2\,\mathrm{tr}(AXMx + MXAx) + 2\,\mathrm{tr}(xMxA),\\ &= 2\,\mathrm{tr}(XMXA) - 2\,\mathrm{tr}(Cx) + 2\,\mathrm{tr}(MxAx),\\ &= 2\,\mathrm{tr}(XMXA) + 2f_E(x).\end{aligned}$$

Since $\mathrm{tr}(XMXA)$ is a constant, minimizing $f_E(x)$, as defined in (4.13), amounts to minimizing the $\mathcal{L}$-norm of the error of x.

The following is a trivial consequence of the definition of a norm.

Proposition 4.5. *Suppose $\mathcal{L} \succ 0$. Then the objective function (4.13) and the energy norm (4.15) of the error vanish only at the solution X, i.e.,*

$$f_E(x) = 0 \iff \|\mathrm{vec}(X-x)\|_{\mathcal{L}} = 0 \iff AxM + MxA = C.$$

The optimization problem (4.12) is formulated on the set of s.p.s.d. matrices of rank p. One may wonder whether this is a restriction in comparison to optimizing over the set of s.p.s.d. matrices with rank less than or equal to p, i.e.,

$$\min_x f_E(x) \quad \text{s.t. } x \in \{X \in \mathbf{S}_+^n \mid \mathrm{rank}(X) \le p\}. \tag{4.16}$$

Intuitively, we can expect that at least one of the minimizers of problem (4.16) will not be of rank lower than p if the rank of the exact solution X_* is at least p. This can be proved rigorously as follows.

Proposition 4.6. *Let $A, M \succ 0$ and $\mathrm{rank}(X_*) \ge p$, with X_* the exact solution of (4.1). Then every minimizer of (4.16) has rank p.*

Proof. The proof mimics the proof by contradiction of a similar result in Helmke & Shayman (1995, Prop. 2.4), but is based on using the $\mathcal{L}$-norm instead of the Euclidean norm.
Let $g(X) := \frac{1}{2}\|\text{vec}\,(X - X_*)\,\|^2_{\mathcal{L}} = \text{tr}[(X - X_*)M(X - X_*)A]$. Since $g(X) = f_E(X) + c$ with $c \in \mathbf{R}$, replacing f_E with g in (4.16) does not change the minimizers. Suppose $\hat{X}$ is such a minimizer and $\text{rank}(\hat{X}) = l < p$. Then the rank one perturbation $\hat{X} - \epsilon bb^T$, with any $\epsilon \geq 0$ and arbitrary $b \in \mathbf{R}^{n\times 1}$ will not have a function value lower than $g(\hat{X})$. This gives

$$\begin{aligned} g(\hat{X} - \epsilon bb^T) &= g(\hat{X}) - 2\epsilon\,\text{tr}[(\hat{X} - X_*)Abb^TM] + \epsilon^2\,\text{tr}(bb^TMbb^TA) \\ &\geq g(\hat{X}). \end{aligned}$$

Take any b such that $\text{tr}(bb^TMbb^TA) = 1$. So for all ϵ, we have that

$$2\,\text{tr}[(X_* - \hat{X})Abb^TM] \leq \epsilon$$

and this means that $\text{tr}[(X_* - \hat{X})Abb^TM] = 0$. Since $A, M \succ 0$ and since b is arbitrary (as long as $\text{tr}(bb^TMbb^TA) = 1$), we can conclude $X_* = \hat{X}$. This contradicts the assumption that $\text{rank}(\hat{X}) = l < p$. □

From Prop. 4.3, we know that for controllable systems, the solution will be of full rank. Hence for such systems, it always suffices to restrict the optimization to $\mathbf{S}^{n,p}_+$ instead of solving the bigger problem (4.16). In addition, even when the exact solution is (theoretically) singular, say of rank $k < n$, it suffices to optimize on $\mathbf{S}^{n,p}_+$ as long as $p \leq k$.

We remark that the previous observation is valid only in exact arithmetic. In most applications, the solution will have decaying eigenvalues and the algorithm may need to approximate eigenvalues close to ϵ_{mach}. This can potentially lead to unstable calculations. We later see in Section 4.5.4 that the forthcoming algorithm can cope with these small eigenvalues.

4.4.2 Based on the residual: f_R

Recall that we defined the following objective function

$$f_R : \mathbf{S}^{n,p}_+ \to \mathbf{R},\ x \mapsto \text{tr}[AxM^T(MxA^T + AxM^T)] - 2\,\text{tr}(AxM^TC). \quad (4.17)$$

We claim that this objective is related to the Frobenius norm of

$$R(x) := AxM^T + MxA^T - C, \quad (4.18)$$

the residual at x. Observe that since $x = x^T$ and, by assumption, $C = C^T$, matrix $R(x)$ is symmetric also.

Working out $\|R(x)\|_{\mathrm{F}}^2$ and manipulating the matrices inside the traces, we get

$$\begin{aligned}\|R(x)\|_{\mathrm{F}}^2 &= \operatorname{tr}(R(x)R(x)) \\ &= \operatorname{tr}[(AxM^T + MxA^T - C)(AxM^T + MxA^T - C)] \\ &= 2\operatorname{tr}[AxM^TAxM^T + AxM^TMxA^T] - 4\operatorname{tr}[AxM^TC] + \operatorname{tr}(CC) \\ &= 2f_R(x) + \operatorname{tr}(CC).\end{aligned}$$

Since $\operatorname{tr}(CC)$ is again a constant, minimizing $f_R(x)$, as defined in (4.17), amounts to minimizing the residual of the error x.

For ease of exposition, we state also the following trivial proposition.

Proposition 4.7. *Let* (4.1) *be nonsingular. Then the objective function* (4.17) *and the residual* (4.18) *vanish only at the solution, i.e.,*

$$f_R(x) = 0 \quad \Longleftrightarrow \quad R(x) = 0 \quad \Longleftrightarrow \quad AxM^T + MxA^T = C.$$

In the same way as for f_E, we can show that the bigger optimization problem

$$\min_x f_R(x) \quad \text{s.t. } x \in \{X \in \mathbf{S}_+^n \mid \operatorname{rank}(X) \leq p\}. \tag{4.19}$$

does not give better solutions when the rank of the solution is larger than p.

Proposition 4.8. *Let* (4.1) *be nonsingular and* $\operatorname{rank}(X_*) \geq p$*, with* X_* *the exact solution of* (4.1)*. Then every minimizer of* (4.19) *has rank* p*.*

Proof. We prove again by contradiction using the function $g(X) := \|R(X)\|_{\mathrm{F}}^2$. Since $g(X) = 2f_R(X) + c$ with $c \in \mathbf{R}$, replacing f_R by g in (4.19) does not change the minimizers. Suppose $\hat{X}$ is a minimizer of (4.19) with $\operatorname{rank}(\hat{X}) = l < p$. Then the rank one perturbation $\hat{X} - \epsilon bb^T$, with any $\epsilon \geq 0$ and arbitrary $b \in \mathbf{R}^{n\times 1}$ will not have a function value lower than $g(\hat{X})$. Let us denote the symmetric matrix $Z_b = Abb^TM^T + Mbb^TA^T$. We obtain

$$\begin{aligned}g(\hat{X} - \epsilon bb^T) &= g(\hat{X}) - 2\epsilon\operatorname{tr}[R(\hat{X})Z_b] + \epsilon^2\operatorname{tr}(Z_bZ_b) \\ &\geq g(\hat{X}).\end{aligned}$$

Take any b such that $\operatorname{tr}(Z_bZ_b) = 1$. So for all ϵ, we have that

$$2\operatorname{tr}[R(\hat{X})(Abb^TM^T + Mbb^TA^T)] \leq \epsilon$$

and this means that $\operatorname{tr}[R(\hat{X})(Abb^TM^T + Mbb^TA^T)] = 0$. Since $\mathcal{L}$ is nonsingular and since b is arbitrary (as long as $\operatorname{tr}(Z_bZ_b) = 1$), we can conclude $R(\hat{X}) = 0$. Hence, $\hat{X} = X_*$ and this contradicts the assumption that $\operatorname{rank}(\hat{X}) = l < p$. □

The same remarks at the end of section 4.4.1 about the rank of the exact solution are applicable here. Furthermore, it seems sensible again to restrict our optimization approach to nonsingular Lyapunov equations.

Remark 4.9. Observe that for nonsingular $\mathcal{L}$ the norm of the residual can also be written as

$$\begin{aligned}\|R(x)\|_{\mathrm{F}}^2 &= (\mathcal{L}\,\mathrm{vec}(x) - \mathrm{vec}(C))^T(\mathcal{L}\,\mathrm{vec}(x) - \mathrm{vec}(C)),\\ &= (\mathrm{vec}(x) - \mathcal{L}^{-1}\,\mathrm{vec}(C))^T\mathcal{L}^T\mathcal{L}(\mathrm{vec}(x) - \mathcal{L}^{-1}\,\mathrm{vec}(C)),\\ &= \mathrm{vec}(x - X)^T\mathcal{L}^T\mathcal{L}\,\mathrm{vec}(x - X),\end{aligned}$$

where X is the solution of $\mathcal{L}\,\mathrm{vec}(X) = \mathrm{vec}(C)$.

4.4.3 Comparison between f_E and f_R

When $\mathcal{L} \succ 0$, we have two objective functions at our disposal. Since f_R is always applicable, why would we even consider using f_E? It turns out that there are good reasons to choose f_E over f_R. One of these reasons is that minimizing f_R is numerically less stable and that the energy norm is a better estimator of the real error than the residual.

We show this for a simple Lyapunov equation, namely the benchmark problem in Penzl (1999, Ex. 2.1) corresponding to a one-dimensional heat equation with Dirichlet boundary and a special right-hand side. Discretizing with the usual three-point stencil gives a Lyapunov equation (4.1) where

$$A = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & 1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix},\ M = I,\ C = cc^T,\ c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{4.20}$$

with $h = 1/(n+1)$ the mesh size. In the experiments of the current section, the discretization is kept small to $n = 100$, such that the behavior of the algorithms would not be affected by any large-scale issues.

Summarizing the previous sections, we have three methods at our disposal to compute x_p, the rank p approximation to X. Each of these methods minimizes some measure of the error. The first method is simply a truncated eigenvalue decomposition (EVD) of the exact solution X. This results in the best approximation in the Frobenius norm of the error

$$E_0 := \|x_p - X\|_{\mathrm{F}}/\|X\|_{\mathrm{F}}.$$

The second approach is minimizing f_E. The global minimum of f_E is the best approximation in the energy norm

$$E_1 := \|x_p - X\|_{\mathcal{L}} / \|X\|_{\mathcal{L}}.$$

Finally, the third method will minimize f_R. The global minimum of f_R gives the smallest residual

$$E_2 := \|R(x_p)\|_{\mathrm{F}} / \|R(0)\|_{\mathrm{F}}.$$

with $R(x) := AxM^T + MxA^T - C$. We stress that the *global* minima of f_E, f_R minimize E_1, E_2 respectively. It remains to be seen whether the *local* optimizers of f_E, f_R are useful, i.e., whether they sufficiently decrease the error.

As a mnemonic, observe that E_i can be seen as a (scaled) Euclidean norm of the error $x_p - X$ but weighted by i times $\mathcal{L}$:

$$E_0^2 = \operatorname{vec}(x_p - X)^T \operatorname{vec}(x_p - X) / \operatorname{vec}(X)^T \operatorname{vec}(X),$$

$$E_1^2 = \operatorname{vec}(x_p - X)^T \mathcal{L} \operatorname{vec}(x_p - X) / \operatorname{vec}(X)^T \operatorname{vec}(X),$$

$$E_2^2 = \operatorname{vec}(x_p - X)^T \mathcal{L}^T \mathcal{L} \operatorname{vec}(x_p - X) / \operatorname{vec}(C)^T \operatorname{vec}(C).$$

The first two are immediate, while the last identity was shown in Remark 4.9.

In Figure 4.1, we have investigated these measures for different values of the rank p. Taking into account the definition of the measures E_0, E_1, E_2, the truncated EVD should deliver the lowest error in Fig. 4.1(a), the optimizers of f_E should be minimal in Fig. 4.1(b), while those of f_R should be the lowest in Fig. 4.1(c). One can observe that this is always the case for the truncated EVD and f_E. On the other hand, the minimizers of f_R for the higher ranks can be suboptimal. Since the numerical methods always converged to a local optimum, this was clearly due to the nonconvexity of the problem. Hence, minimizing f_E is apparently less sensitive to the nonconvexity of the optimization problem than f_R. Remark that we did not guide the minimization algorithms by choosing any special initial guesses or randomizing the algorithm over several runs. The results were simply obtained by one deterministic run of the proposed Algorithm 2.

We can observe some additional properties about minimizing f_E and f_R. In Fig. 4.1(a) we see that, despite the fact that we minimize the energy norm, the Frobenius norm of the error due to f_E is almost as good as that of the best rank-k approximation. The residual on the other hand is considerably worse. This is to be expected since the energy norm is known to be a better estimator for the real error than the residual.

Furthermore, it is clear that the numerical final accuracy for $p \to \infty$ of f_R is worse than that of f_E. The final accuracy of f_E is, in turn, worse than that of the

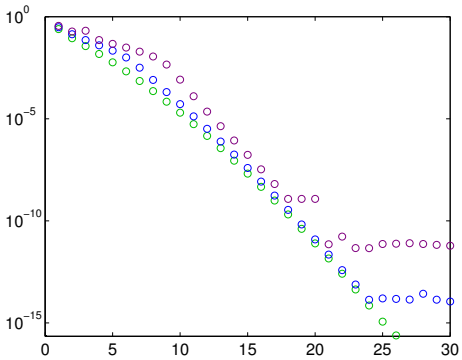
truncated EVD. This is also to be expected. The minimization of f_R is essentially based on solving the normal equations directly. Thus, the best one can hope for is an error of the order of $\sqrt{\epsilon_{\text{mach}}}$. On the other hand, the truncated EVD can achieve a solution accurate to the order of ϵ_{mach}. Apparently, the minimization of f_E is somewhere situated in between those two extremes. The residual of all three methods is essentially the same.

From the experiments above, we have seen that when $\mathcal{L} \succ 0$, there are three good reasons to prefer f_E over f_R: it is more robust to local optimizers, it gives better approximations and it is more accurate. Furthermore, we will see that, in case of f_E, the expressions for the gradient and the Hessian are more concise and that choosing a preconditioner for the iterative solver is more straightforward. All these reasons are related to the fact that the energy norm is more natural for solving positive definite systems like (4.1). Minimizing the energy norm of the error over some (sub)space is also very common in the context of Krylov solvers for s.p.d. systems, see, e.g., in a more general setting Kressner & Tobler (2009).

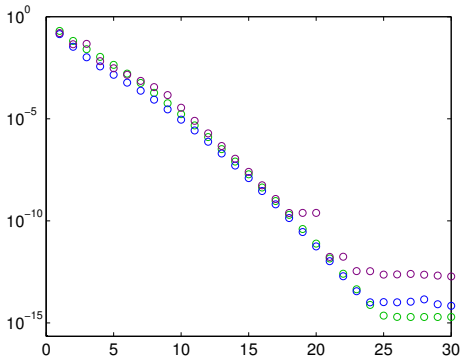
4.5 The Riemannian algorithms

In Chapter 3, we derived an embedded geometry for $\mathbf{S}_+^{n,p}$ which is supposedly well-suited for Riemannian optimization. We will clarify this now for the optimization of problem (4.12) by the Riemannian Trust-Region method of Section 2.8.1.

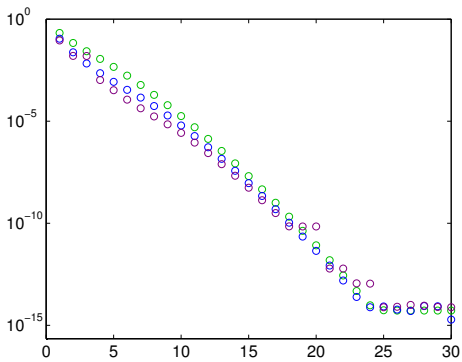
In particular, we will show how the second-order models, necessary for the Riemannian Trust-Region method, can be derived for the objective functions f_E and f_R. As explained in Section 2.8, a convenient choice for the lift-and-retract framework is to use a quadratic model m_x of the lifted objective function $\widehat{f}_x := f \circ R_x$, with f either f_E or f_R. Thanks to Th. 2.37, we can build this model by taking a classic Taylor expansion of $\widehat{f}_x := f_x \circ R_x^{(2)}$ with $R_x^{(2)}$ the second-order expansion (3.44). We will do this now for f_E and f_R.



(a) $E_0 := \|x_p - X\|_{\mathrm{F}}/\|X\|_{\mathrm{F}}$



(b) $E_1 := \|x_p - X\|_{\mathcal{L}}/\|X\|_{\mathcal{L}}$



(c) $E_2 := \|R(x_p)\|_{\mathrm{F}}/\|R(0)\|_{\mathrm{F}}$

Figure 4.1: Measures of the quality of the rank p approximations x_p obtained by a truncated EVD (○), by minimizing f_E (○) and by minimizing f_R (○), for a small Lyapunov equation. The quality is measured in the Frobenius norm (a), in the energy norm of the error (b) and in the Frobenius norm of the residual (c).

4.5.1 The second-order model of f_E

Recall that $\mathrm{P}_x^{\mathrm{t}}$ denotes the orthogonal projection on the $T_x\mathbf{S}_+^{n,p}$. Let $\xi^p := \mathrm{P}_x^{\mathrm{t,p}}(\xi)$ with $\mathrm{P}^{\mathrm{t,p}}$ the projector onto a specific subspace of $T_x\mathbf{S}_+^{n,p}$, see Section 3.3.4, then

$$\begin{aligned}
\widehat{f}_x(\xi) &:= f_E(R_x^{(2)}(\xi)) \\
&= f_E(x + \xi + \xi^p x^\dagger \xi^p + O(\xi^3)) \\
&= \operatorname{tr}\left[(x + \xi + \xi^p x^\dagger \xi^p + O(\xi^3))A(x + \xi + \xi^p x^\dagger \xi^p + O(\xi^3))M\right] \\
&\quad - \operatorname{tr}\left[(x + \xi + \xi^p x^\dagger \xi^p + O(\xi^3))C\right] \\
&= f_E(x) + \operatorname{tr}(\xi R) + \operatorname{tr}(\xi A \xi M + \xi^p R \xi^p x^\dagger) + O(\|\xi\|^3)
\end{aligned}$$

where R is the residual of x, i.e., $R := AxM + MxA - C$.

Let $g^{\mathrm{E}}(\cdot,\cdot)$ denote the Euclidean inner product (3.15). In the truncated expression we can easily recognize the terms that contribute to the gradient,

$$g^{\mathrm{E}}(\xi, \operatorname{grad} f_E(x)) = \operatorname{tr}[\xi R],$$

and the Hessian,

$$g^{\mathrm{E}}(\xi, \operatorname{Hess} f_E(x)[\xi]) = 2\operatorname{tr}[\xi A \xi M + \xi^p R \xi^p x^\dagger].$$

Next, we need to manipulate these expressions to obtain the gradient as a tangent vector of $T_x\mathbf{S}_+^{n,p}$ and the Hessian as a linear and symmetric mapping of $T_x\mathbf{S}_+^{n,p} \to T_x\mathbf{S}_+^{n,p}$. This can be done by judiciously sneaking in the orthogonal projectors $\mathrm{P}_x^{\mathrm{t}}(\xi) = \xi$ and $\mathrm{P}_x^{\mathrm{t,p}}(\xi^p) = \xi^p$.

Gradient. The computation of the gradient is almost trivial since $R = R^T$:

$$g^{\mathrm{E}}(\xi, \operatorname{grad} f_E(x)) := \operatorname{tr}(\xi R) = g^{\mathrm{E}}(\xi, R) = g^{\mathrm{E}}(\mathrm{P}_x^{\mathrm{t}}(\xi), R) = g^{\mathrm{E}}(\xi, \mathrm{P}_x^{\mathrm{t}}(R))$$

and so we obtain

$$\operatorname{grad} f_E(x) = \mathrm{P}_x^{\mathrm{t}}(AxM + MxA - C). \tag{4.21}$$

We recover the gradient as the orthogonal projection onto $T_x\mathbf{S}_+^{n,p}$ of the gradient in the full space. This is to be expected from Thm. 2.34.

Hessian. As for the Hessian, we need additionally $\xi^p := \mathrm{P}_x^{\mathrm{t,p}}(\xi)$,

$$\begin{aligned} g^{\mathrm{E}}(\xi, \operatorname{Hess} f_E(x)[\xi]) &= 2\operatorname{tr}(\xi A \xi M + \xi^p R \xi^p x^\dagger) \\ &= 2g^{\mathrm{E}}(\xi, A\xi M) + 2g^{\mathrm{E}}(\xi^p, R\xi^p x^\dagger) \\ &= 2g^{\mathrm{E}}(\xi, \mathrm{P}_x^{\mathrm{t}}(A\xi M)) + 2g^{\mathrm{E}}(\mathrm{P}_x^{\mathrm{t,p}}(\xi), R\mathrm{P}_x^{\mathrm{t,p}}(\xi)x^\dagger) \\ &= g^{\mathrm{E}}(\xi, 2\mathrm{P}_x^{\mathrm{t}}(A\xi M) + 2\mathrm{P}_x^{\mathrm{t,p}}(R\mathrm{P}_x^{\mathrm{t,p}}(\xi)x^\dagger)) \end{aligned}$$

and so

$$\begin{aligned} \operatorname{Hess} f_E(x)[\xi] &= 2\mathrm{P}_x^{\mathrm{t}}(A\xi M) + 2\mathrm{P}_x^{\mathrm{t,p}}(R\mathrm{P}_x^{\mathrm{t,p}}(\xi)x^\dagger) \\ &= \mathrm{P}_x^{\mathrm{t}}(A\xi M + M\xi A) + \mathrm{P}_x^{\mathrm{t,p}}(R\mathrm{P}_x^{\mathrm{t,p}}(\xi)x^\dagger + x^\dagger \mathrm{P}_x^{\mathrm{t,p}}(\xi)R). \end{aligned} \tag{4.22}$$

From looking at this expression, it may not be clear whether all properties of a Hessian are satisfied. Luckily, we can get a much more familiar representation of the Hessian after vectorization. Recalling the derivation of the matrices (3.21)–(3.23), we apply the $\operatorname{vec}(\cdot)$ operator to (4.22),

$$\begin{aligned} &\operatorname{vec}(\operatorname{Hess} f_E(x)[\xi]) \\ &\quad = \operatorname{vec}\left(\mathrm{P}_x^{\mathrm{t}}(A\xi M + M\xi A) + \mathrm{P}_x^{\mathrm{t,p}}(R\mathrm{P}_x^{\mathrm{t,p}}(\xi)x^\dagger + x^\dagger \mathrm{P}_x^{\mathrm{t,p}}(\xi)R)\right) \\ &\quad = P_x^{\mathrm{t}} \operatorname{vec}(A\xi M + M\xi A) + P_x^{\mathrm{t,p}} \operatorname{vec}\left(R\mathrm{P}_x^{\mathrm{t,p}}(\xi)x^\dagger + x^\dagger \mathrm{P}_x^{\mathrm{t,p}}(\xi)R\right) \\ &\quad = P_x^{\mathrm{t}}(A \otimes M + M \otimes A)\operatorname{vec}(\mathrm{P}_x^{\mathrm{t}}\xi) + P_x^{\mathrm{t,p}}(R \otimes x^\dagger + x^\dagger \otimes R)\operatorname{vec}(\mathrm{P}_x^{\mathrm{t,p}}\xi) \\ &\quad = \left[P_x^{\mathrm{t}}(A \otimes M + M \otimes A)P_x^{\mathrm{t}} + P_x^{\mathrm{t,p}}(R \otimes x^\dagger + x^\dagger \otimes R)P_x^{\mathrm{t,p}}\right]\operatorname{vec}(\xi). \end{aligned}$$

Finally, the Hessian is given by the matrix

$$\mathcal{H}_x := P_x^{\mathrm{t}} \mathcal{L} P_x^{\mathrm{t}} + P_x^{\mathrm{t,p}}(x^\dagger \otimes R + R \otimes x^\dagger)P_x^{\mathrm{t,p}}. \tag{4.23}$$

with $\mathcal{L} := A \otimes M + M \otimes A$. This matrix is clearly a linear and symmetric operator and, due to the presence of P_x^{t} and $P_x^{\mathrm{t,p}}$, it has $T_x\mathbf{S}_+^{n,p}$ as its domain and range.

If we compare this Hessian to the Euclidean Hessian of the full space, $\mathcal{L}$, we see that besides the expected projector P_x^{t} there is a "correction term" due to curvature of the low-rank constraint. These two terms can be elegantly interpreted as a combination of first- and second-order information of the objective function and the manifold as a constraint; see Absil *et al.* (2009a, Sec. 6) for the analysis. Furthermore, this correction term can make the Hessian indefinite and renders the optimization problem nonconvex (also in the Riemannian sense). This shows the need for a modification of Newton's method to a globally convergent algorithm like our TR approach.

Numerical verification. The correction term in $\mathcal{H}_x$ is necessary to have a correct second-order model, as we illustrate in Fig. 4.2. There, we have plotted the maximum relative error of two models in function of the norm of the tangent vector for 1000 random vectors for the Lyapunov equation (4.20). The model was constructed at a random $x \in \mathbf{S}_+^{100,5}$ which was not a critical point of f_E. The first model (denoted with circles) uses $\mathcal{H}_x$ as the Hessian and the second model (with squares) uses $P_x^{\mathrm{t}} \mathcal{L} P_x^{\mathrm{t}}$. In other words, the first model, being the true second-order model on the manifold, should have a third-order decay of the error. We have verified this for the four retractions $R_x^{(1)}, R_x^{(2)}, R_x^{\mathrm{proj}}, R_x^{\mathrm{ograph}}$ from Section 3.5. (Remark that in the numerical method below we will only use R_x^{proj}.)

It is clearly visible that only the model with $\mathcal{H}_x$ as the Hessian in combination with a second-order retraction gives rise to a second-order model. These retractions are indeed $R_x^{(2)}$, R_x^{proj} and R_x^{ograph}. Observe in addition that R_x^{proj} is more accurate for a fixed ξ than the other second-order retractions.

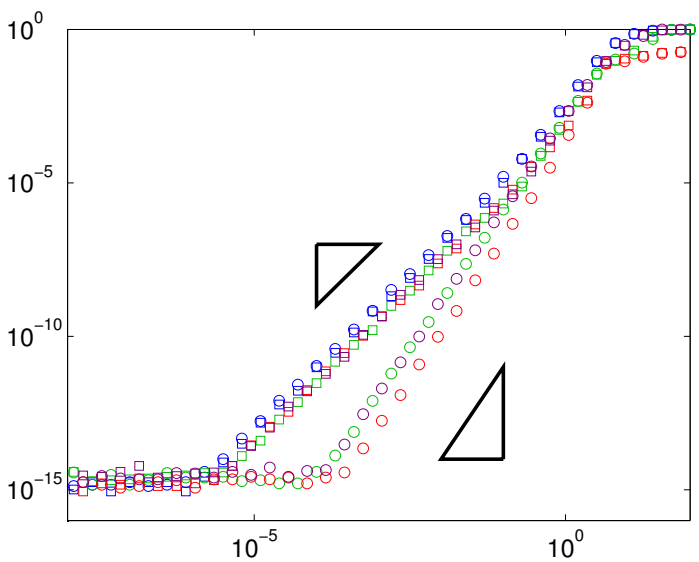


Figure 4.2: Relative error of the linear (with □) and quadratic model (with ○) for $f_E(R_x(t\xi))$ and for the retractions $R_x^{(1)}$ ○; $R_x^{(2)}$ ○; R_x^{proj} ○; and R_x^{ograph} ○. The triangles indicate the second and third order convergence of the error.

4.5.2 The second-order model of f_R

The derivation of the second-order model of f_R goes along the same lines as that of f_E. Let $\xi^p := \mathrm{P}_x^{\mathrm{t,p}}(\xi)$, then

$$\widehat{f}_x(\xi) := f_R(R_x^{(2)}(\xi)) = f_R(x + \xi + \xi^p x^\dagger \xi^p + O(\xi^3))$$

can be split into a series expansion up to second-order terms in ξ, i.e.,

$$\widehat{f}_x(\xi) = f_R(x) + f_R^{(1)}(x,\xi) + f_R^{(2)}(x,\xi) + O(\|\xi\|^3).$$

where $f_R^{(1)}(x,\xi)$ depends linearly on ξ and $f_R^{(2)}(x,\xi)$ quadratically.

Recall that $R := AxM^T + MxA^T - C$, then some straightforward, yet tedious manipulations reveal

$$f_R^{(1)}(\xi) = 2\,\mathrm{tr}[(A\xi M^T)R] \tag{4.24}$$

and

$$f_R^{(2)}(\xi) = \mathrm{tr}[(A\xi M^T)(A\xi M^T + M\xi A^T) + 2(A\xi^p x^\dagger \xi^p M^T)R]. \tag{4.25}$$

Gradient By definition $f_R^{(1)}(\xi) := g^{\mathrm{E}}(\xi, \operatorname{grad} f_R(x))$, hence we get

$$\begin{aligned} g^{\mathrm{E}}(\xi, \operatorname{grad} f_R(x)) &= 2\,\mathrm{tr}[\xi(M^T RA)] \\ &= 2g^{\mathrm{E}}(\xi, \mathrm{P}_x^{\mathrm{t}}(M^T RA)). \end{aligned}$$

If we explicitly symmetrize the argument of $\mathrm{P}_x^{\mathrm{t}}$, we obtain

$$\operatorname{grad} f_R(x) := \mathrm{P}_x^{\mathrm{t}}(M^T RA + A^T RM).$$

Again, we recover the Riemannian gradient as the orthogonal projection onto $T_x\mathbf{S}_+^{n,p}$ of the Euclidean gradient.

Hessian For the Hessian, we need to manipulate $f_R^{(2)}(\xi) := g^{\mathrm{E}}(\xi, \operatorname{Hess} f_R(x)[\xi])$:

$$\begin{aligned} g^{\mathrm{E}}(\xi, \operatorname{Hess} f_R(x)[\xi]) &= 2\,\mathrm{tr}\left[\xi(M^T A\xi M^T A + M^T M\xi A^T A)\right] \\ &\quad + 2\,\mathrm{tr}\left[\xi^p x^\dagger \xi^p (M^T RA)\right] \\ &= 2g^{\mathrm{E}}(\xi, \mathrm{P}_x^{\mathrm{t}}(M^T A\xi M^T A + M^T M\xi A^T A)) \\ &\quad + 2g^{\mathrm{E}}(\xi, \mathrm{P}_x^{\mathrm{t,p}}(x^\dagger \mathrm{P}_x^{\mathrm{t,p}}(\xi)(M^T RA))) \end{aligned}$$

and so

$$\begin{aligned}
&\operatorname{Hess} f_R(x)[\xi] \\
&\quad = 2\mathrm{P}_x^{\mathrm{t}}(M^T A\xi M^T A + M^T M\xi A^T A) + 2\mathrm{P}_x^{\mathrm{t,p}}(x^\dagger \mathrm{P}_x^{\mathrm{t,p}}(\xi) M^T RA)) \\
&\quad = \mathrm{P}_x^{\mathrm{t}}[M^T A\xi M^T A + M^T M\xi A^T A + A^T M\xi A^T M + A^T A\xi M^T M] \\
&\qquad + \mathrm{P}_x^{\mathrm{t,p}}[A^T RM\mathrm{P}_x^{\mathrm{t,p}}(\xi)x^\dagger + x^\dagger \mathrm{P}_x^{\mathrm{t,p}}(\xi) M^T RA]. \qquad (4.26)
\end{aligned}$$

Analogous to f_E, this expression for the Hessian operator can be represented as a matrix. Applying vectorization again, we obtain

$$\mathcal{H}_x := P_x^{\mathrm{t}}(\mathcal{L}^T\mathcal{L})P_x^{\mathrm{t}} + P_x^{\mathrm{t,p}}(x^\dagger \otimes M^T RA + A^T RM \otimes x^\dagger)P_x^{\mathrm{t,p}}. \qquad (4.27)$$

with $\mathcal{L} := A \otimes M + M \otimes A$.

Observe that the Riemannian Hessian consists again of two terms. The first term is the projection of the Euclidean Hessian, $\mathcal{L}^T\mathcal{L}$. This is also the system matrix of the normal equations of (4.1). The second term is a term due to the curvature of $\mathbf{S}_+^{n,p}$.

Numerical verification The second-order model for f_R is verified in the same way as that of f_E. Instead of a symmetric matrix A, we now use the three-point discretization of a one-dimensional convection-diffusion operator

$$\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} u - \alpha \frac{\mathrm{d}u}{\mathrm{d}x}.$$

Parameter α determines the amount of convection. This results in the following matrix:

$$A = \frac{1}{h^2}\begin{bmatrix}
2 & -1-\alpha h & & & & & \\
-1+\alpha h & 2 & -1-\alpha h & & & & \\
 & -1+\alpha h & 2 & -1-\alpha h & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & -1+\alpha h & 2 & -1-\alpha h & \\
 & & & & -1+\alpha h & 2 & -1-\alpha h \\
 & & & & & -1+\alpha h & 2
\end{bmatrix}, \qquad (4.28)$$

with mesh size $h = 1/(n+1)$.

In Figure 4.3, the error of the two models, based on $\mathcal{H}_x$ or $P_x^{\mathrm{t}}\mathcal{L}^T\mathcal{L}P_x^{\mathrm{t}}$, for $\alpha = 200$ are visible. At $\alpha = 200$, A has a complex spectrum, but for other values of α the

convergence behavior is essentially the same. Comparing this figure with Figure 4.2, we can draw exactly the same conclusions: only the model with $\mathcal{H}_x$ in combination with a second-order retraction is second-order accurate; and retraction R_x^{proj} is (slightly) more accurate for a fixed ξ than the other retractions.

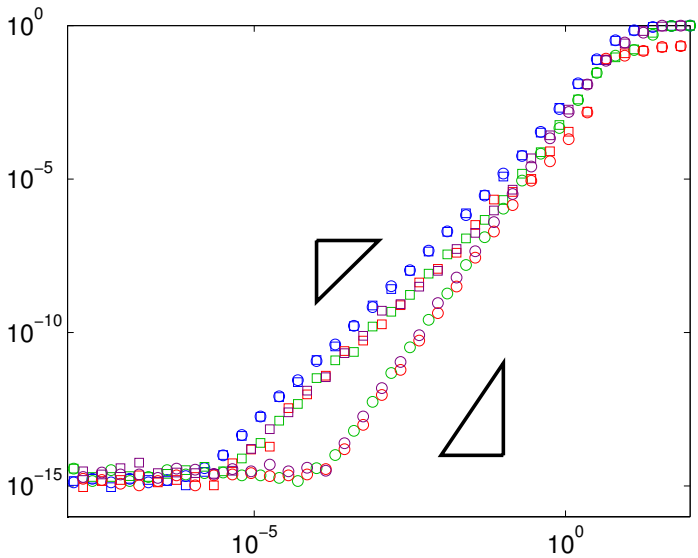


Figure 4.3: Relative error of the linear (with □) and quadratic model (with ○) for $f_R(R_x(t\xi))$ and for the retractions $R_x^{(1)}$ ○; $R_x^{(2)}$ ○; R_x^{proj} ○; and R_x^{ograph} ○. The triangles indicate the second and third order convergence of the error.

4.5.3 Implementation aspects

We shall assume that the matrices A and M have a fast matrix vector product at the cost of $O(n)$. This is a very reasonable assumption when we are dealing with large-scale applications arising from discretized PDEs.

Factorization Recall that every $x = YY^T \in \mathbf{S}_+^{n,p}$ can be represented as $x = VDV^T$, see Cor. 3.2, and that the two formats are interchangeable by Remark 3.3. For the actual implementation, we prefer the format VDV^T with $V \in \text{St}^{n\times k}$ and $D = \text{diag}(d)$, $d \in \mathbf{R}^p$, i.e., as a compact eigenvalue decomposition. The orthogonal

projections onto $T_x\mathbf{S}_+^{n,p}$, as detailed in Section 3.3.4, become

$$\mathrm{P}_x^{\mathrm{t,p}}(Z) = (I - VV^T)\frac{Z + Z^T}{2}VV^T + VV^T\frac{Z + Z^T}{2}(I - VV^T)$$

$$\mathrm{P}_x^{\mathrm{t}}(Z) = VV^T\frac{Z + Z^T}{2}VV^T + \mathrm{P}_x^{\mathrm{t,p}}(Z).$$

Tangent vectors As explained in the Remark 3.17, we can represent every tangent vector in $x = VDV^T$ as

$$\xi = VSV^T + ZV^T + VZ^T, \quad \text{with } S \in \mathbf{S}^k, Z \in \mathbf{R}^{n\times k}, V^TZ = 0.$$

So we only need to compute and store the matrices S and Z.

Objective function f_E Evaluation of $f_E(x)$ in $x = VDV^T$ can be done efficiently since

$$f_E(x) = \operatorname{tr}(xAxM) - \operatorname{tr}(xC) = \operatorname{tr}[(V^TAV)D(V^TMV)D] - \operatorname{tr}[(V^TCV)D].$$

The vectors AV, MV and CV will be useful later on, so we store them after each calculation of $f_E(x)$.

Objective function f_R In the same way, the evaluation of $f_R(x)$ in $x = VDV^T$ can be done efficiently as

$$f_R(x) = \operatorname{tr}[(V^TM^TAV)D(V^TM^TAV)D + (V^TA^TAV)D(V^TM^TMV)D] - 2\operatorname{tr}[(V^TM^TCAV)D]$$

Again, the vectors $AV, MV, A^TV, M^TV, C(AV)$ are stored for later use.

Gradient of f_E The gradient at $x = VDV^T$ is given by the projection of the residual $R = AVDV^TM + MVDV^TA - C$ onto $T_x\mathbf{S}_+^{n,p}$, see eq. (4.21). With the previously explained projectors, this becomes

$$\operatorname{grad} f_E(x) = VV^TRVV^T + (I - VV^T)RVV^T + VV^TR(I - VV^T).$$

After some manipulations and rearranging the terms for efficiency, we obtain that $\operatorname{grad} f_E(x)$ equals the tangent vector $VSV^T + ZV^T + VZ^T$ with

$$T = (AV)D(V^TMV) + (MV)D(V^TAV) - CV,$$

$$S = V^TT,$$

$$Z = T - VS.$$

Gradient of f_R The gradient at $x = VDV^T$ equals the tangent vector $\operatorname{grad} f_R(x) = VSV^T + ZV^T + VZ^T$ with

$$\begin{aligned} T &= (A^TAV)D(V^TM^TMV) + (M^TMV)D(V^TA^TAV) \\ &\quad + (A^TMV)D(V^TA^TMV) + (M^TAV)D(V^TM^TAV) \\ &\quad - A^TCMV - M^TCAV, \\ S &= V^TT, \\ Z &= T - VS. \end{aligned}$$

Hessian of f_E The Hessian at x evaluated for $\xi = VS_\xi V^T + Z_\xi V^T + VZ_\xi^T$ is given by eq. (4.22) where $x^\dagger = VD^{-1}V^T$. After similar but slightly more tedious manipulations as for the gradient, we get $\operatorname{Hess} f_E(x)[\xi] = VSV^T + ZV^T + VZ^T$ with

$$\begin{aligned} T_1 &= (AV)S_\xi(V^TMV) + (MV)S_\xi(V^TAV) \\ &\quad + (AV)(Z_\xi^TMV) + (MV)(Z_\xi^TAV) + (AZ_\xi)(V^TMV) + (MZ_\xi)(V^TAV), \\ T_2 &= (AV)D(V^TMZ_\xi) + (MV)D(V^TAZ_\xi) - CZ_\xi, \\ S &= V^TT_1, \\ Z &= T_1 - V(V^TT_1) + (T_2 - V(V^TT_2))D^{-1}. \end{aligned}$$

The dominating costs are the matrix vector products AZ_ξ, MZ_ξ and CZ_ξ.

Hessian of f_R The Hessian at x evaluated for $\xi = VS_\xi V^T + Z_\xi V^T + VZ_\xi^T$ is given $\operatorname{Hess} f_R(x)[\xi] = VSV^T + ZV^T + VZ^T$ with

$$\begin{aligned} T_1 &= (A^TAV)S_\xi(V^TM^TMV) + (M^TMV)S_\xi(V^TA^TAV) \\ &\quad + (M^TAV)S_\xi(V^TM^TAV) + (A^TMV)S_\xi(V^TA^TMV) \\ &\quad + (A^TAV)(Z_\xi^TM^TMV) + (M^TMV)(Z_\xi^TA^TAV) \\ &\quad + (A^TMV)(Z_\xi^TA^TMV) + (M^TAV)(Z_\xi^TM^TAV) \\ &\quad + (A^TAZ_\xi)(V^TM^TMV) + (M^TMZ_\xi)(V^TA^TAV), \\ &\quad + (M^TAZ_\xi)(V^TM^TAV) + (A^TMZ_\xi)(V^TA^TMV). \end{aligned}$$

$$T_2 = (A^T AV)D(V^T M^T MZ_\xi) + (M^T MV)D(V^T A^T AZ_\xi) - A^T CMZ_\xi$$
$$+ (M^T AV)D(V^T M^T AZ_\xi) + (A^T MV)D(V^T A^T MZ_\xi) - M^T CAZ_\xi,$$
$$S = V^T T_1,$$
$$Z = T_1 - V(V^T T_1) + (T_2 - V(V^T T_2))D^{-1}.$$

4.5.4 A practical algorithm

In principle, the optimization method of Alg. 1, together with the derivations from above, suffice to find a low-rank approximation if the rank is known in advance. In practice however, this rank is unknown since one is usually interested in an approximation that is better than a certain tolerance. We will take the relative residual in the Frobenius norm, i.e. $\|AXM^T + MXA^T - C\|/\|C\|$, as tolerance.

Initialization We therefore propose Algorithm 2 that computes a series of local optimizers to problem (4.12) with increasing rank p until the tolerance is satisfied. In order to ensure a monotonic decrease of the cost function, and thus the error, we can reuse the previous solution Y_i and append a zero column. Since the zero column does not increase the rank of $Y_{i+1} = \begin{bmatrix} Y_i & 0_{n\times\delta} \end{bmatrix}$, the Hessian at $x = Y_{i+1}Y_{i+1}^T$ would be singular due to $x^\dagger \notin \mathbf{S}_+^{n,p}$ in (4.22) and so we cannot use Y_{i+1} as initial guess for Algorithm 1. Instead, we perform one steepest descent step and in practice this always results in a full rank matrix Y_{i+1}. In case this Y_{i+1} would not be of full rank, one can always slightly perturb the search direction of the steepest descent step such that the new iterate is of full-rank. After that, we can find a minimizer with Algorithm 1. In our numerical experiments (see Section 4.7), we found that δ, the increment of the rank, is best kept small, say 2 to 6.

Stopping criterium Due to numerical cancellation, the residual should not be computed based on the expression $\mathrm{tr}(R_i R_i)$ with $R_i = (AY_i)(Y_i^T M^T) + (MY_i)(Y_i^T A^T) - C$. Instead, we propose two ways of computing the residual. The first is similar to Penzl (1999). Suppose $x_i = Y_i Y_i^T$ and $C = cc^T$ with $c \in \mathbf{R}^{n\times l}$. After having computed the compact QR factorization of $\begin{bmatrix} AY_i & MY_i & c \end{bmatrix} = Q_i T_i$,

we can express the relative residual as

$$\begin{aligned} r_i &= \|Ax_iM^T + Mx_iA^T - C\|/\|C\| \\ &= \left\| \begin{bmatrix} AY_i & MY_i & c \end{bmatrix} \begin{bmatrix} MY_i & AY_i & -c \end{bmatrix}^T \right\| / \|cc^T\| \\ &= \left\| T_i \begin{bmatrix} 0 & I_k & 0 \\ I_k & 0 & 0 \\ 0 & 0 & -I_l \end{bmatrix} T_i^T \right\| / \|c^Tc\|. \end{aligned} \tag{4.29}$$

To be efficient, this approach requires that C is of low rank $l \ll n$ which is fairly common in control applications. However, the residual can also be computed without any rank conditions on C. Since the matrix vector product of the residual with a given vector can be applied efficiently, it is possible to approximate r_i with a matrix-free eigensolver, like `eigs` in MATLAB. Since the dominant eigenvalues of the symmetric matrix R_i are usually well separated, this estimation converges fast.

The cost of computing r_i stays very moderate since the rank is small and we only need to compute it once the minimizer is found. This is in contrast to alternative existing methods, like CFADI and KPIK, where the computation of the residual can be the most expensive step of the whole algorithm.

Regularizing the Hessian The evaluation of the Hessians requires the computation of the pseudo-inverse $x^\dagger$. In case x has eigenvalues close to ϵ_{mach}, this will lead to unstable calculations due to blowup and the optimization algorithm will not converge. Luckily, it is fairly straightforward to avoid this blowup by regularizing the pseudo-inverse itself.

Let $x = VDV^T$ with $D = \operatorname{diag}(d)$ be an eigenvalue decomposition. Instead of calculating $x^\dagger$ as $VD^{-1}V^T$, we use instead

$$x^\dagger = V \operatorname{diag}(r) V^T, \quad \text{with } r_i = 1/\sqrt{d_i^2 + \epsilon_{\text{mach}}^2}.$$

In the numerical experiments one observes that this regularization succeeds in its goal of letting the numerical algorithms converge for any x, whether it has large or small eigenvalues. This approach is similar to Koch & Lubich (2007), where this regularization is also used and its effects are further analyzed.

4.6 Preconditioning the optimization of f_E

The computationally most expensive step of the RTR method of Section 2.8.1 is the solution of the TR subproblems (2.12). Since these problems are possibly very

Algorithm 2 Final algorithm: RLyap

Require: objective function $f \in \{f_E, f_R\}$, initial rank p_1, initial guess $x_1 = Y_1 Y_1^T$ with $Y_1 \in \mathbf{R}_*^{n \times p_1}$, residual tolerance τ, rank increase δ.

1: **for** $i = 1, 2, \ldots$ **do**
2: **Find** x_i **as a minimizer of** (4.12):
perform Algorithm 1 with f on $\mathbf{S}_+^{n,p_i}$.
3: **Compute the residual of** x_i:
calculate r_i based on (4.29) or with `eigs`.
4: **if** $r_i \leq \tau$ **then**
5: **Solution found**:
return x_i and quit.
6: **else**
7: **Increase rank**:
$p_{i+1} = p_i + \delta$.
8: **Compute initial guess** $x_{i+1} = Y_{i+1} Y_{i+1}^T$:
perform one step of steepest descent on $Y_{i+1} = \begin{bmatrix} Y_i & 0_{n \times \delta} \end{bmatrix}$.
9: **end if**
10: **end for**

large, we solve them iteratively with the truncated CG (tCG) method: in each outer step of the RTR method, the second-order model is minimized with the usual matrix-free CG algorithm. However, tCG employs two extra stopping criteria compared to CG. The algorithm terminates if CG would use a search direction of negative or zero curvature, or if the new iterate would violate the TR bound. Sufficiently close to the minimizer, these modifications are however inactive. This results in a number of inner iterations to solve (2.12) up to a certain tolerance while still guaranteeing superlinear convergence of RTR.

Like classic CG, tCG lends itself excellently to preconditioning since a well chosen preconditioner will have a good influence on the conditioning of each TR subproblem. The effect is that the number of inner iterations will be drastically lowered. It is however not directly obvious how we can define a matrix-free preconditioner that is symmetric and positive definite in all points x. In this section, we will derive such a preconditioner for the optimization of f_E.

4.6.1 Projected Euclidean Hessian

From Section 4.5.1, we know that the Riemannian Hessian of $f_E(x)$,

$$\mathcal{H}_x := P_x^{\mathrm{t}}(A \otimes M + M \otimes A)P_x^{\mathrm{t}} + P_x^{\mathrm{t,p}}(x^\dagger \otimes R + R \otimes x^\dagger)P_x^{\mathrm{t,p}},$$

consists of two parts, namely a projection of the Euclidean Hessian $\mathcal{L} = A \otimes M + M \otimes A$ and a term involving the residual. For PDE-related Lyapunov equations, this

projected Hessian $P_x^{\mathrm{t}}\mathcal{L}P_x^{\mathrm{t}}$ should make a good candidate for a preconditioner, since most of the bad conditioning of the TR subproblems can be attributed to $\mathcal{L}$, i.e. the PDE. This can be observed by solving problems of different size but with constant condition number for $\mathcal{L}$. In that case, the number of CG iterations required to solve the Newton equations is roughly independent of the size. Moreover, thanks to $\mathcal{L} \succ 0$, $P_x^{\mathrm{t}}\mathcal{L}P_x^{\mathrm{t}}$ is always symmetric and positive definite on $T_x\mathbf{S}_+^{n,p}$.

In order to see how effective this preconditioner is, we have solved a series of Lyapunov equations resulting from a five-point discretized Laplace equation on a square with zero Dirichlet boundary conditions. The right-hand side is a random symmetric matrix of rank 3. The solutions were computed with Alg. 1 and the iteration was stopped when the norm of the gradient was below 10^{-10}.

First, we check the dependence on the size n of the system. The condition number of $\mathcal{L}$ (and presumably $\mathcal{H}_x$) will grow $\sim n$ so we can expect more tCG iterations as the subproblems become larger. In Table 4.2 we see the number of outer RTR iterations and the total number of inner tCG iterations to solve for a rank $k = 15$ approximation. For this example, a rank 15 approximation has a relative residual (4.29) of about 10^{-5}, which is sufficient to assess the performance of the preconditioner. We have included the maximum number of tCG iterations as well since the last subproblems need to be solved up to high accuracy. For the unpreconditioned problem, this maximum number grows as $\sqrt{n}$, or, in other words, as the square root of the condition number of $\mathcal{L}$. This is in correspondence with the standard convergence analysis for CG. For the preconditioned problem on the other hand, this number is small for all sizes and, more importantly, it stays bounded. Furthermore, the preconditioner reduces the total number of outer and inner iterations drastically to a number almost independent of the size of the system.

prec.	n	150^2	200^2	250^2	300^2	350^2	400^2	450^2	500^2
none	n_{outer}	46	44	49	44	43	44	56	48
	$\sum n_{\text{inner}}$	1913	2173	2984	3158	4076	4185	5375	5622
	$\max n_{\text{inner}}$	414	529	624	731	757	858	1004	1080
$P_x^{\mathrm{t}}\mathcal{L}P_x^{\mathrm{t}}$	n_{outer}	39	40	42	46	47	48	47	49
	$\sum n_{\text{inner}}$	83	83	91	94	96	101	88	93
	$\max n_{\text{inner}}$	14	13	15	13	13	13	12	10

Table 4.2: Effect of preconditioning. Dependence on n for problems with fixed rank $k = 15$.

Second, in Table 4.3 we investigated the dependence on k while the size was fixed to $n = 500^2$. For the unpreconditioned problem, the maximum number of inner tCG iterations is rather independent of the rank. This seems to support the

hypothesis that most of the poor conditioning of $\mathcal{H}_x$ can be attributed to $\mathcal{L}$. For the preconditioner on the other hand, we should expect some dependence on k since we did not take the second part of the Riemannian Hessian into account. Even though we observe in Table 4.3 an increase in the total number of inner iterations with growing rank, there is still a significant reduction thanks to preconditioning. Furthermore, the maximum number of inner iterations seems to stay constant with growing rank.

precond.	k	1	4	7	10	13	16	19
none	n_{outer}	20	34	35	40	50	51	69
	$\sum n_{\text{inner}}$	4921	4949	5295	4502	6039	5682	6211
	$\max n_{\text{inner}}$	1150	1066	1050	1064	1035	1078	1066
$P_x^{\text{t}}\mathcal{L}P_x^{\text{t}}$	n_{outer}	11	28	35	35	48	48	50
	$\sum n_{\text{inner}}$	26	65	73	70	98	92	100
	$\max n_{\text{inner}}$	5	8	8	11	10	11	11

Table 4.3: Effect of preconditioning. Dependence on k for problems with fixed size $n = 500^2$.

4.6.2 Applying the preconditioner

It is obvious from the tables that preconditioning with $P_x\mathcal{L}P_x$ greatly reduces the total number of inner iterations. However, the preconditioner will only be effective if it can be computed sufficiently fast, ideally at a cost of $O(n)$. We will show that this is possible for $M = I$ and when $(A + \lambda I)x = b$ with $\lambda > 0$ can be solved for x in $O(n)$.

Applying the preconditioner in $x = VDV^T$ means solving $\xi \in T_x\mathbf{S}_+^{n,p}$ such that

$$(P_x^{\text{t}}\mathcal{L}P_x^{\text{t}})(\xi) = \eta, \quad \eta \in T_x\mathbf{S}_+^{n,p}. \tag{4.30}$$

First, we write (4.30) in matrix form:

$$P_V(A\xi M + M\xi A)P_V + P_V^{\perp}(A\xi M + M\xi A)P_V + P_V(A\xi M + M\xi A)P_V^{\perp} = \eta,$$

which decomposes into

$$P_V(A\xi M + M\xi A)P_V = P_V\eta P_V \quad \text{and} \quad P_V^{\perp}(A\xi M + M\xi A)P_V = P_V^{\perp}\eta P_V. \tag{4.31}$$

From now on, let $M = I$. With the factorizations as explained in Section 4.5.3, we can take the following matrix representations for the tangent vectors of $x = VDV^T$:

$$\xi = VS_\xi V^T + Z_\xi V^T + VZ_\xi^T \quad \text{and} \quad \eta = VS_\eta V^T + Z_\eta V^T + VZ_\eta^T.$$

System (4.31) can then be written as

$$V^T AV S_\xi + S_\xi V^T AV + V^T AZ_\xi + Z_\xi^T AV = S_\eta, \tag{4.32}$$

$$P_V^\perp (AVS_\xi + AZ_\xi + Z_\xi V^T AV) = Z_\eta, \quad \text{s.t. } V^T Z_\xi = 0. \tag{4.33}$$

where $S_\xi \in \mathbf{S}^p$ and $Z_\xi \in \mathbf{R}^{n\times p}$ are the unknown matrices. By taking the eigenvalue decomposition $V^T AV = Q\Lambda Q^T$, the previous system is equivalent to

$$\Lambda \widetilde{S}_\xi + \widetilde{S}_\xi \Lambda + \widetilde{V}^T A\widetilde{Z}_\xi + \widetilde{Z}_\xi^T A\widetilde{V} = \widetilde{S}_\eta, \tag{4.34}$$

$$P_{\widetilde{V}}^\perp (A\widetilde{V}\widetilde{S}_\xi + A\widetilde{Z}_\xi + \widetilde{Z}_\xi \Lambda) = \widetilde{Z}_\eta, \quad \text{s.t. } \widetilde{V}^T \widetilde{Z}_\xi = 0. \tag{4.35}$$

where $\widetilde{S}_\xi := Q^T S_\xi Q \in \mathbf{S}^p$ and $\widetilde{Z}_\xi := Z_\xi Q \in \mathbf{R}^{n\times p}$ are transformed unknown matrices, and $\widetilde{V} := VQ$, $\widetilde{S}_\eta := S_\eta Q$ and $\widetilde{Z}_\eta := Z_\eta Q$.

We will now eliminate $\widetilde{Z}_\xi$ from (4.35) and substitute it into (4.34). Since $\Lambda = \operatorname{diag}(\lambda_i)$, we can solve in (4.35) for each column $\widetilde{Z}_\xi(:,i)$ independently (we use the notation $(:,i)$ to denote the i-th column):

$$P_{\widetilde{V}}^\perp (A + \lambda_i I)\widetilde{Z}_\xi(:,i) = \widetilde{Z}_\eta(:,i) - P_{\widetilde{V}}^\perp A\widetilde{V}\widetilde{S}_\xi(:,i), \quad \text{s.t. } \widetilde{V}^T \widetilde{Z}_\xi(:,i) = 0. \tag{4.36}$$

By writing out the projector $P_{\widetilde{V}}^\perp$, it is straightforward to see that this system is equivalent to the following saddle-point problem

$$\begin{bmatrix} A + \lambda_i I & \widetilde{V} \\ \widetilde{V}^T & 0 \end{bmatrix} \begin{bmatrix} \widetilde{Z}_\xi(:,i) \\ y \end{bmatrix} = \begin{bmatrix} \widetilde{Z}_\eta(:,i) - P_{\widetilde{V}}^\perp A\widetilde{V}\widetilde{S}_\xi(:,i) \\ 0 \end{bmatrix} \tag{4.37}$$

with $y \in \mathbf{R}^p$. This saddle-point problem can be efficiently solved by exploiting the sparsity of A but we will postpone the details to Section 4.6.3. For now, we can formally write (4.36) as

$$\widetilde{Z}_\xi(:,i) = \mathcal{T}_i^{-1}(\widetilde{Z}_\eta(:,i)) - \mathcal{T}_i^{-1}(P_{\widetilde{V}}^\perp A\widetilde{V})\widetilde{S}_\xi(:,i), \tag{4.38}$$

where $\mathcal{T}_i^{-1}(B)$ indicates solving the i-th saddle-point problem, corresponding to (4.37), with right-hand side B. Now plugging (4.38) into (4.34), we obtain

$$\Lambda \widetilde{S}_\xi + \widetilde{S}_\xi \Lambda + \begin{bmatrix} v_1 - w_1 & \cdots & v_p - w_p \end{bmatrix} + \begin{bmatrix} v_1 - w_1 & \cdots & v_p - w_p \end{bmatrix}^T = \widetilde{S}_\eta$$

with $v_i = \widetilde{V}^T A\mathcal{T}_i^{-1}(\widetilde{Z}_\eta(:,i))$ and $w_i = \widetilde{V}^T A\mathcal{T}_i^{-1}(P_{\widetilde{V}}^\perp A\widetilde{V})\widetilde{S}_\xi(:,i)$. Let $K_i = \lambda_i I_k - \widetilde{V}^T A\mathcal{T}_i^{-1}(P_{\widetilde{V}}^\perp A\widetilde{V})$, then we can isolate $\widetilde{S}_\xi$ as

$$\begin{bmatrix} K_1\widetilde{S}_\xi(:,1) & \cdots & K_k\widetilde{S}_\xi(:,k) \end{bmatrix} + \begin{bmatrix} \widetilde{S}_\xi(1,:)K_1^T \\ \vdots \\ \widetilde{S}_\xi(k,:)K_k^T \end{bmatrix} = R, \tag{4.39}$$

with known right-hand side matrix $R = \widetilde{S}_\eta - \begin{bmatrix} v_1 & \cdots & v_p \end{bmatrix} - \begin{bmatrix} v_1 & \cdots & v_p \end{bmatrix}^T$. Equation (4.39) is a linear equation in $\widetilde{S}_\xi$ with a special block structure. By vectorizing (see App. B.1.3), it is straightforward to see that the first part of (4.39) satisfies

$$\operatorname{vec}\begin{bmatrix} K_1\widetilde{S}_\xi(:,1) & \cdots & K_p\widetilde{S}_\xi(:,p) \end{bmatrix} = \begin{bmatrix} K_1 & & \\ & \ddots & \\ & & K_p \end{bmatrix} \begin{bmatrix} \widetilde{S}_\xi(:,1) \\ \vdots \\ \widetilde{S}_\xi(:,p) \end{bmatrix} = \mathcal{K}\operatorname{vec}(\widetilde{S}_\xi),$$

where $\mathcal{K}$ denotes the p^2-by-p^2 block-diagonal matrix $\operatorname{diag}(K_i)$. For the second part, we can use $\operatorname{vec}(X) = \Pi\operatorname{vec}(X^T)$ with Π the perfect-shuffle matrix to obtain

$$\operatorname{vec}\begin{bmatrix} \widetilde{S}_\xi(1,:)L_1^T \\ \vdots \\ \widetilde{S}_\xi(p,:)L_k^T \end{bmatrix} = \Pi\operatorname{vec}\begin{bmatrix} L_1\widetilde{S}_\xi^T(1,:) & \cdots & L_k\widetilde{S}_\xi^T(p,:) \end{bmatrix} = \Pi\mathcal{K}\operatorname{vec}(\widetilde{S}_\xi^T).$$

Finally, the whole equation (4.39) can be written as a linear system of size k^2:

$$\mathcal{K}\operatorname{vec}(\widetilde{S}_\xi) + \Pi\mathcal{K}\operatorname{vec}(\widetilde{S}_\xi^T) = \operatorname{vec}(R) \quad \Longleftrightarrow \quad (\mathcal{K} + \Pi\mathcal{K}\Pi)\operatorname{vec}(\widetilde{S}_\xi) = \operatorname{vec}(R). \tag{4.40}$$

After solving (4.40) for $\widetilde{S}_\xi$ we obtain $\widetilde{Z}_\xi$ from (4.38). Undoing the transformations by Q, we get $S_\xi = Q\widetilde{S}_\xi Q^T$ and $Z_\xi = \widetilde{Z}_\xi Q^T$, and thus ξ, such that (4.30) is satisfied.

In case $M \neq I$, we simply approximate M by I and use the previous techniques. Although this is a very crude approximation of M, the obtained preconditioner can work quite well in the numerical experiments. The reason is that for generalized Lyapunov equations M usually represents the Galerkin mass matrix of a FEM discretization. For quasi-uniform meshes with shape regular elements this mass matrix is essentially a scaled identity matrix, see Elman *et al.* (2005, Ch. 1.6)

4.6.3 Cost

There are two dominating costs for applying the preconditioner, namely solving the saddle-point problems (4.37) and solving the linear system (4.40).

Regarding (4.37), there is a vast amount of literature for solving large and sparse saddle-point problems of this kind, see Benzi *et al.* (2005) for an overview. The solution technique of the previous section requires solving $\mathcal{T}_i(X) = B$ for two different right-hand sides, or equivalently, $B = \begin{bmatrix} \widetilde{Z}_\eta(:,i) & P_{\widetilde{V}}^{\perp} A\widetilde{V} \end{bmatrix} \in \mathbf{R}^{n\times(p+1)}$, see (4.38). In our case, p is rather small so we can solve $\mathcal{T}_i(X) = B$ by eliminating the (negative) Schur complement $S_i = \widetilde{V}^T(A + \lambda_i I)^{-1}\widetilde{V}$, as in Benzi *et al.* (2005,

Sec. 5). This gives

$$N = S_i^{-1}(\widetilde{V}^T(A + \lambda_i I)^{-1}B),$$

$$X = (A + \lambda_i I)^{-1}B - (A + \lambda_i I)^{-1}\widetilde{V}N.$$

For each $\mathcal{T}_i$, applying S_i^{-1} means an $O(p^3)$ cost for the Cholesky factorization of the dense matrix S_i and for the forward and back substitution to solve for $N \in \mathbf{R}^{p\times(p+1)}$. In addition, we need to apply $(A + \lambda_i I)^{-1}$ to B and $\widetilde{V}$. Assuming an optimal solver for the sparse s.p.d. matrix $A + \lambda_i I$, this implies a cost of $O(np)$. In total, we get a cost of $O(np^2) + O(p^4)$ to solve all k saddle-point problems. Usually $p \ll n$ and the $O(np^2)$ cost dominates.

Since in every inner iteration of RTR, the iterate x stays fixed, the $n \times p$ matrices $(A+\lambda_i I)^{-1}\widetilde{V}$ and $(A+\lambda_i I)^{-1}P_{\widetilde{V}}^{\perp}A\widetilde{V}$ remain the same and can be re-used. If one is willing to cache these results for all p shifts, there is a significant speedup possible. The downside is that the memory requirements grow from $O(np)$ to $O(np^2)$.

Equation (4.40) is a linear and symmetric system of size p^2. Solving the vectorized form by a dense factorization results in an $O(p^6)$ cost which is prohibitively large, even for small p. However, in practice the equation can be solved much faster with an iterative method like CG. In all problems, we have observed convergence in only $O(\log p)$ steps. Together with (4.39) as matrix-vector product with cost $O(p^3)$, this results in an empirical $O(p^3 \log p)$ cost for solving (4.40).

4.7 Numerical results

In this section we report on some properties of our Riemannian algorithm, Alg. 2, named RLyap, when solving large-scale model problems. The computations were done with MATLAB R2009b on a 64-bit Intel Pentium Xeon 2.66 GHz with $\epsilon_{\text{mach}} \simeq 2 \cdot 10^{-16}$. The RTR algorithm was implemented using GenRTR (Baker *et al.*, 2007), a generic MATLAB package for Riemannian Trust-Region. The reported timings are wall times that include all necessary computations like setting up the preconditioner, computing sparse Cholesky factors and determining parameters necessary for the solvers.

In addition, we compare the performance of RLyap with two state-of-the-art low-rank Lyapunov solvers, namely the modified CFADI method and KPIK. For CFADI we used LyaPack 1.8 (Saak *et al.*, 2008) and for KPIK the implementation by Simoncini (2007). We will also use an algebraic multigrid preconditioner with the implementation from Boyle *et al.* (2010) for iteratively solving systems. All default options were kept.

From Section 4.4.3 we know that for a Lyapunov equation with $\mathcal{L} \succ 0$, minimizing f_E is to be preferred over f_R. Furthermore, we only derived a preconditioner for f_E. Since we wish to compare with the state-of-the-art *in this section, we will focus only on the anti-stable and symmetric Lyapunov equation, hence* $\mathcal{L} \succ 0$. Any meaningful comparison for f_R would first require the derivation of a new preconditioner suited for f_R.

4.7.1 Quality of the low-rank solutions

In Fig. 4.4 we investigated the quality of the solutions from optimizing f_E compared to the best rank k approximations of the exact solution. We used RLyap (Algorithm 2) to obtain the minimizers for f_E. In addition, we also computed the solutions with KPIK and CFADI. The KPIK method is the standard algorithm from Simoncini (2007). For the CFADI method we used the implementation in LyaPack 1.8 Saak *et al.* (2008) with the heuristic shifts determination from Benner *et al.* (2008). The computation of these shifts in Saak *et al.* (2008) is determined by a triple of parameters (l_0, k_+, k_-). To compare the influence of the shifts, we used two sets: a set of good shifts, corresponding to $(10, 50, 25)$, and a set of slightly worse shifts, corresponding to $(5, 20, 10)$.

The generalized Lyapunov equation was based on the RAIL benchmark from Benner & Saak (2004) of size $n = 1\,357$. In general, problems that have a r.h.s. matrix C with $\operatorname{rank}(C) > 1$ lead to an unnecessary complicated implementation (block Krylov methods) and analysis (superlinear convergence effects) since they do not gain any substantial insight compared to the rank one case. Hence, we took the usual simplification of this benchmark to have a rank one matrix $C = B_1 B_1^T$ by selecting B_1 to be the first column of the B matrix in Benner & Saak (2004). We stress that this is only a simplification for this section and in the later experiments, we will solve problems with matrices C that have higher ranks.

Since RLyap minimizes the error in the energy norm of the residual, we should expect that the best rank k approximations always have a better accuracy measured in the Frobenius norm. This is verified in the left panel of Fig. 4.4. In addition we see that the difference stays rather small and behaves uniformly in the rank. In other words, the RLyap approximations are nearly as good as the best rank k approximations. Surprisingly, the errors of the residual of the RLyap approximations are a little better than the best rank k approximations, as seen in the right panel of Fig. 4.4.

Next, we performed the same comparison with the KPIK and CFADI algorithm. Each step of KPIK or CDFADI appends a new column to the factor Y of the solution $x = YY^T$ and thus the rank will increase with every step. Although each step of these algorithms is cheap in comparison with the RLyap method, in Fig. 4.4 we can clearly see that these low-rank solutions are far from optimal. Furthermore,

the convergence of the CFADI method seems quite sensitive to the choice of the shifts.

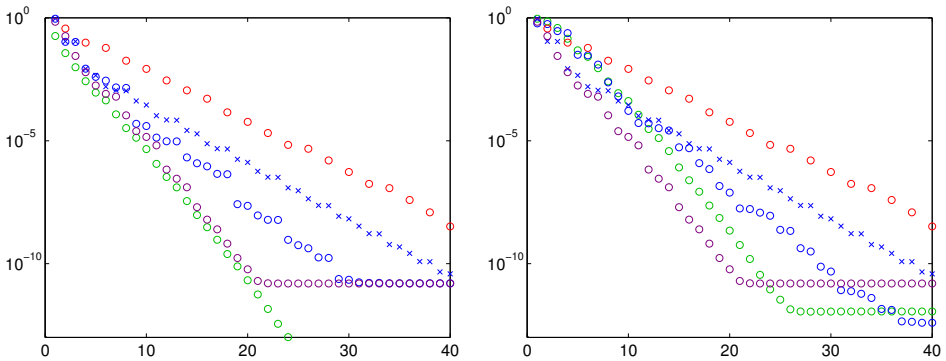


Figure 4.4: The relative error (left) and the absolute residual (right) of the minimizers of f_E (o) in function of the rank k for the simplified RAIL benchmark. In addition, the best rank k approximations (o) and the approximations computed with CFADI(5,20,10) (×), CFADI(10,50,25) (o), KPIK (o) are shown.

4.7.2 Accuracy of the linear systems

Most low-rank solvers, including RLyap with the preconditioner of Section 4.6, need to solve (shifted) linear systems. For large-scale problems these systems will need to be solved iteratively by, e.g., a Krylov method preconditioned with multigrid. We use an Algebraic Multigrid (AMG) as multigrid solver with the implementation of Boyle *et al.* (2010). Depending on the accuracy of the desired low-rank solution, KPIK and CFADI need to solve these systems quite accurately. An advantage of our method is that the AMG preconditioner can directly be used as approximate inverse for the shifted system in Section 4.6.2. So, instead of accurately solving shifted linear systems, we use AMG on the shifted system to precondition the actual TR subproblem. Since the AMG preconditioner is spectrally equivalent with the original shifted system, we get again a sound preconditioner for the TR subproblems.

We will investigate numerically how the accuracy of this inverse influences RLyap and CFADI. For RLyap we take a fixed number of V-cycles in the preconditioner. For CFADI, each linear system is solved by a fixed number of CG steps, preconditioned by AMG. We can see in Table 4.4 that RLyap converges with every choice of number of V-cycles, while CFADI stagnates if the linear system is solved too crudely. Apparently only one V-cycle in RLyap's preconditioner gives the fastest wall time. One can argue that even one V-cycle is still too costly for RLyap. We did

not pursue this further, but a more careful convergence analysis of the inner-outer tolerances could give a significant speedup.

	V-cycles	1	2	3	4	5	6	7
	avg. tol.	3e−1	5e−2	2e−3	7e−5	3e−6	1e−7	5e−9
CFADI	time (s.)	26	32	37	54	63	65	68
	rel. res.	7e−1	9e−2	5e−3	3e−4	2e−5	8e−7	5e−7
RLyap	time (s.)	42	52	70	85	114	134	155
	rel. res.	2e−7	2e−7	2e−7	2e−7	2e−7	2e−7	2e−7

Table 4.4: Effect of the accuracy when solving linear systems with different number of AMG V-cycles. The corresponding average reduction of the residual is indicated by "avg. tol.".

4.7.3 Without mass matrix

The Lyapunov equation (4.1) with A the two-dimensional (2D) Poisson problem on the square, $M = I$ and C a rank one right-hand side is a much-used benchmark. The results of the performance of RLyap, CFADI and KPIK for a relative residual of 10^{-6} is listed in Table 4.5. Solving $(A + \lambda I)x = b$ was done with a sparse direct solver (CHOLMOD with AMD reordering) or with an iterative solver (CG preconditioned with AMG). In case of KPIK and CFADI the inner tolerance for the iterative solver was 10^{-10} (a lower tolerance resulted in stagnation for the biggest problem) whereas for RLyap only one AMG V-cycle sufficed, see also Section 4.7.2.

It is clear from the table that RLyap is significantly slower with a direct solver than with an iterative solver, while the situation is exactly reversed for CFADI and KPIK. This seems to indicate that the preconditioner in RLyap is too crude to warrant solving it very accurately, that is, by a direct solver. Remark however that it is possible to get a significant speedup for the sparse direct solver since the symbolic factorization has to be done only once. For this problem, this phase actually accounts for almost half the time of the total solve.

Since our MATLAB implementation of RLyap is not competitive with a sparse direct solver, we will only compare the iterative approach. We can observe that RLyap performs quite well for this problem: it is only slightly slower than the fastest method, KPIK, and it is several times faster than CFADI. Furthermore, the difference with KPIK becomes smaller for bigger problems: while the smallest problem is 62% slower, the largest is only 25% slower. If we compare the ranks of the solutions, we observe that RLyap always delivers the smallest rank. The rank of KPIK is significantly higher and grows with problem size.

		CHOLMOD with AMD			PCG with AMG		
		RLyap	CFADI	KPIK	RLyap	CFADI	KPIK
$n = 500^2$	time (s.)	101	55	13	40	70	24
	rank X	12	20	36	12	19	36
$n = 1000^2$	time (s.)	513	104	65	175	310	118
	rank X	12	20	38	12	18	38
$n = 1500^2$	time (s.)	1495	267	189	443	811	354
	rank X	12	20	19	12	19	44

Table 4.5: Performance for the finite difference discretized 2D Poisson problem on the square. Tolerance on the relative residual was 10^{-6}.

The previous problem can be regarded as relatively easy since the grid is very isotropic. We therefore constructed a problem with a more irregular triangular mesh by discretizing the three-dimensional (3D) Poisson equation on the cube with piecewise linear finite elements. The right-hand side is $C = bb^T$ with b the FEM discretization of the unit function. This is a problem were CHOLMOD cannot be used so this shows the necessity of the iterative approach. The results of the comparison is visible in Table 4.6 and are almost similar to the previous 2D problem. Again KPIK is the fastest method, but now CFADI performs significantly better than RLyap for the bigger problem. The reason that RLyap is slower for the bigger problem is that the quality of the AMG V-cycle deteriorates drastically for the bigger problem.

		PCG with AMG		
		RLyap	CFADI	KPIK
$n = 132\,745$	time (s.)	81	115	58
	rank X	14	14	34
$n = 306\,006$	time (s.)	275	328	168
	rank X	15	15	36
$n = 1\,068\,660$	time (s.)	1750	1430	882
	rank X	16	16	46

Table 4.6: Performance for the finite element discretized 3D Poisson problem on the cube. Tolerance on the relative residual was 10^{-6}.

4.7.4 With mass matrix

We will now report how RLyap performs with a mass matrix. We took the RAIL benchmark (Benner & Saak, 2004) with the outer product of the first column of the B-matrix as right-hand side. First, we solved a Lyapunov equation by neglecting M, i.e., we only take A of the benchmark. After that, we solved the actual system with M. The results are visible in Table 4.7. We can see that if $M = I$, all solvers behave as expected. If we use the actual system with $M \neq I$, the results are very different. For the biggest problem CFADI is twice as fast as KPIK, while RLyap performs disproportionately poorly. This can be explained by the approximation of $M = I$ in RLyap's preconditioner: since the condition number of M is about 400, this approximation is apparently too crude to give an efficient solver.

		simplified $M = I$			orig. M	$M = LL^T$	
		RLyap	CFADI	KPIK	RLyap	CFADI	KPIK
$n = 5177$	time (s.)	3.9	2.6	1.4	29	5.9	5.3
	rank X	22	21	54	26	28	80
$n = 20\,209$	time (s.)	13	12	7.8	103	39	49
	rank X	22	25	70	26	31	114
$n = 79\,841$	time (s.)	76	61	46	447	249	552
	rank X	29	28	96	30	34	170

Table 4.7: Comparison for the simplified RAIL benchmark. Linear systems solved by PCG with AMG.

4.7.5 Right-hand side matrix of high rank

An advantage of the proposed method is that it does not impose conditions on the form of the right-hand side matrix C. Since all matrices will eventually be projected onto the tangent space, RLyap only requires the product of a vector with C. KPIK and CFADI on the other hand, require that C is factored as $C = BB^T$ with $B \in \mathbf{R}^{n \times l}$. Furthermore, for computational efficiency it is important that l is small since systems of the form $(A + \lambda I)^{-1}B$ have to be solved in each step. If the numerical rank of the solution X is much smaller than l these methods will presumably not be efficient.

We will now show that RLyap can indeed be more efficient in solving systems when l is large. All of the existing benchmark examples for low-rank Lyapunov solvers however are formulated for relatively low-rank C; in fact, many only use a rank one matrix. We will therefore construct an example that has a matrix C with

relatively high rank compared to rank of the approximation of X. Take A_n the $n \times n$ tridiagonal matrix of a discretized one-dimensional Laplacian. Consider the Lyapunov equation

$$A_n X + X A_n = C \quad \text{with } C := A_n^{-1} \begin{bmatrix} 0_{n_2 \times n_2} & 0_{n_2 \times n_1} \\ 0_{n_1 \times n_2} & I_{n_2 \times n_2} \end{bmatrix} A_n^{-1} \tag{4.41}$$

and $n_2 = \lfloor n/10 \rfloor$ and $n_1 = n - n_2$.

We can solve (4.41) without modification with RLyap. The experimental results for different meshes are visible in Table 4.7.5. We used a direct solver for the preconditioner and a tolerance of 10^{-6} for the relative residual. This tolerance could not be satisfied for the biggest problem, so we relaxed in addition the tolerance to $5 \cdot 10^{-5}$. Now the method succeeds in finding a low-rank approximation for all problems.

Although matrix C is by construction available in factored form, its rank will grow as $n/10$. This is clearly unsuitable for CFADI or KPIK. Thanks to the pre- and post-multiplying by A^{-1}, matrix C will have decaying eigenvalues and a reasonably good low-rank approximation. One can compute such a rank k_C approximation with a matrix-free eigenvalue solver, e.g., `eigs` in MATLAB. The decay is however slow and only algebraic so the rank of the resulting approximation C_k can still be rather high. Furthermore, since C_k is an approximation of the true right-hand side C the solution of $AX + XA = C_k$ will again be an approximation of the true solution of eq. (4.41). So it is important to take k_C sufficiently high but not too high. For the smallest problem, $k_C = 15$ turned out to be the smallest rank for which the tolerance on the residual can still be satisfied. We take in addition $k_C = 30$ to examine the effect of k_C.

With these low-rank matrices C_{15} and C_{30} at hand, we solved the same systems again with CFADI and RLyap. We did not compare with KPIK since this requires a block-Krylov implementation which is currently not available. In all cases, except the smallest problem, RLyap outperformed the modified CFADI method with respect to wall time, rank of the solution and final accuracy.

RLyap with C_{15} was as expected faster than with C_{30} and both were faster than with C. The influence on the rank k_C was not big however, and solving directly with the real C is much more user friendly. The CFADI method on the other hand is very sensitive to the rank of C_k, both in time and accuracy. Take for example the problem with size $n = 40\,000$. Here CFADI with C_{15} stagnates while RLyap succeeds in solving the problem. If the right-hand side is C_{30}, CFADI succeeds in solving the problem again, but the method became twice as slow. In addition, we see that RLyap gives more accurate final approximations than CFADI.

	solver	RLyap	CF-ADI	RLyap	CF-ADI	RLyap
	rhs	C	C_{15}	C_{15}	C_{30}	C_{30}
$n = 20\,000$	time (s.)	35.7	**15.4**	32.6	40.3	34.3
$\tau =$ 1e−6	rank X	25	35	27	49	25
	residual	9.27e−7	6.39e−7	7.03e−7	5.53e−7	9.27e−7
$n = 40\,000$	time (s.)	70.3	(38.7)	**48.9**	111.2	61.6
$\tau =$ 1e−6	rank X	23	35	25	49	27
	residual	9.87e−7	2.67e−6	9.30e−7	8.61e−7	9.86e−7
$n = 80\,000$	time (s.)	169.7	(103.1)	**116.8**	(232.1)	128.4
$\tau =$ 1e−6	rank X	25	35	25	50	25
	residual	9.89e−7	2.68e−6	9.81e−7	2.98e−6	9.90e−7
$n = 160\,000$	time (s.)	(560.6)	(183.4)	(400.1)	(404.9)	(516.3)
$\tau =$ 1e−6	rank X	27	36	31	50	30
	residual	1.74e−6	2.85e−5	1.98e−6	2.73e−5	1.56e−6
$n = 160\,000$	time (s.)	176.8	139.5	**104.7**	300.9	125.5
$\tau =$ 5e−5	rank X	12	33	12	48	12
	residual	1.44e−5	3.57e−5	3.35e−5	3.47e−5	1.44e−5

Table 4.8: Experimental results for problem (4.41) computed with RLyap and CFADI for different meshes. The right-hand side matrices were the real matrix, C, and rank 15 and 30 approximations, C_{15} and C_{30} respectively. Timings between parentheses indicate that convergence stagnated and tolerance τ on the residual could not be satisfied.

4.8 Conclusions

We proposed a new algorithm, RLyap, to solve for low-rank solutions of Lyapunov equations based on optimization on the manifold of fixed-rank matrices. The performance of RLyap seems to be between that of KPIK and ADI, although there are problems where the situation is reversed. The example in Section 4.7.5 shows that the solver can be significantly faster and yet be more user-friendly when the rank of the solution is lower than that of the right-hand side matrix. Noting that CFADI and KPIK perform already quite well for symmetric problems, this leads to think that the Riemannian approach is promising for non-symmetric problems, where, e.g., the shift determination for CFADI is much more difficult. This will require a preconditioner for the Hessian of f_R, the objective function based on the residual.

Even though RLyap performs adequately, there is still room for improvement, especially w.r.t. the preconditioner. In the current implementation, most of the time

is spent solving shifted linear systems with constantly changing shifts. Since these shifts do not always change significantly and the preconditioner does not need to be solved very accurately, there is great potential to lower the computational burden. A similar observation was made in Nong & Sorensen (2009) where the multiple shifts could be avoided by using a subspace technique and a single shift. A different technique for accelerating the convergence in case of large-scale discretizations of PDEs will be discussed in the next chapter.

5 Multilevel strategies

In the previous chapter we devised a Riemannian optimization algorithm that solves low-rank approximations for Lyapunov equations. The algorithm was made scalable to large-scale problems by the use of the projected preconditioner. It was essentially an algebraic technique that did not use the structure of the underlying PDE. In this chapter, we will employ a different technique for solving large-scale problems. One that exploits the multilevel nature of PDEs directly.

The result will be a generic multilevel Riemannian optimization method that can be used for minimizing PDE-related objective functions. We apply this algorithm to our usual problem of low-rank approximations for the Lyapunov equation on $\mathbf{S}_+^{n,p}$. Based on a Local Fourier Analysis of the Euclidean version of this algorithm, the tensor-product multigrid, we can use the typical strategies from linear multigrid and obtain a true multigrid solver on $\mathbf{S}_+^{n,p}$.

The tensor-product multigrid and its Local Fourier Analysis was published in Vandereycken & Vandewalle (2009). The multilevel Riemannian optimization method and its application to Lyapunov equations is an unpublished contribution.

5.1 Introduction

The prototypical example of a solver that exploits the multilevel structure of PDEs is multigrid (Trottenberg *et al.*, 2000). It uses a hierarchy of nested grids where

one cycles between these grids to smooth the error on a finer scale and to compute corrections on a coarser scale. It is well-known that multigrid is one of the most efficient solvers for elliptic PDEs. It is therefore natural to ask whether the idea of multigrid can be extended to Lyapunov equations while still obtaining low-rank solutions.

In this chapter we extend the multigrid technique to the Riemannian Trust-Region method of Chapter 4. First, we outline the basic principles of a classical multigrid iteration for the Lyapunov equation, termed a tensor-product multigrid, and show how its convergence can be analyzed. Then, we incorporate this multigrid algorithm into the Riemannian framework. This new method will be the generalization of recently developed multilevel optimization algorithms in Euclidean space, like MG/Opt of Lewis & Nash (2005) and the Recursive Trust-Region method of Gratton *et al.* (2008), to Riemannian manifolds. We demonstrate numerically that this combination leads to an optimization algorithm on $\mathbf{S}_{+}^{n,p}$ that is efficient (in the sense of multigrid optimality) and robust (in the sense of Trust-Region robustness).

5.1.1 Discretizations of PDEs

Since the emphasis of this chapter is the solution of PDEs, we introduce their usual discretization and notation in some more detail. Let $\Omega \subset \mathbf{R}^d$ be a bounded, open domain. Consider the linear second-order PDE

$$-\sum_{i,j=1}^{d} \frac{\partial}{\partial x_j}\left(a_{i,j}(x)\frac{\partial u}{\partial x_i}\right) + \sum_{i=1}^{d} b_i(x)\frac{\partial}{\partial u x_i} + c(x)u = f(x), \quad x \in \Omega. \tag{5.1}$$

The boundary value problem consists of solving $u(x)$ such that it satisfies (5.1) together with appropriate boundary conditions, e.g, homogeneous Dirichlet $u(x) = 0$ for all $x \in \partial\Omega$. Under suitable conditions on Ω, the boundary conditions and the coefficients, this boundary value problem has a unique solution. We assume that these conditions are satisfied.

Since operator (5.1) is defined on a function space, one usually discretizes $u(x)$ in order to obtain a finite-dimensional problem. There are several ways to discretize PDEs. Among the more popular ones are the finite difference method (FDM), finite element method (FEM) and finite volume method (FVM). The result is a finite-dimensional approximation of $u(x)$ that is parameterized by some vector of coefficients.

In the context of multi-*grid* one says that the parameters of this discretization belong to some spatial grid G. In addition, since it is called *multi*-grid, one constructs a series of grids G_l for $l = 1, 2, \ldots$ that each contain the coefficients $x_l \in \mathbf{R}^{n_l}$ for the approximations $u_l(x)$. The discretization at $l = 1$ will have the coarsest grid-size h_l, while the next levels will subsequently refine this grid. For a

quasi-uniform grid one has $h_l = O(2^{-l})$. The vector x_l can be found by solving a finite-dimensional linear system

$$A_l x_l = f_l$$

where $A_l \in \mathbf{R}^{n_l \times n_l}$ is called the system matrix and $f_l \in \mathbf{R}^{n_l}$ the right-hand side.

5.2 Tensor-product multigrid

Let A_l and f_l be the result of a discretization of a PDE at some level l as explained above. Consider then the standard Lyapunov equation associated with (5.1):

$$A_l X_l + X_l A_l^T = C_l, \quad C_l = f_l f_l^T. \tag{5.2}$$

From the previous chapter, it should be clear that since X_l is a very large and dense matrix, solving equation (5.2) is a formidable task. This shows the need for efficient iterative methods that are scalable when $l \to \infty$.

As stated in the introduction, we choose an iterative method based on multigrid since for many PDEs, multigrid is known to be an optimal solver: the amount of work and memory scales linearly with the number of unknowns. In case of a Lyapunov equation, we will show that the multigrid algorithm can be formulated as an iterative method that operates on a tensor-product space. In this manner we obtain a so-called *tensor-product multigrid*, as developed in Rosen & Wang (1995); Penzl (1997). The method in this section computes X_l in (5.2) as a full matrix, which is obviously not scalable to problems with $n_l \gg 10^4$. Computing low-rank approximations of X_l will be dealt with in the next section.

We show that the typical multigrid optimality and efficiency caries over for this tensor-product multigrid method. In Penzl (1997), this has been proved qualitatively for a specific instance of tensor-product multigrid that solves the one-dimensional Poisson equation. By means of Local Fourier Analysis (LFA), we can show the optimality for more general formulations of the multigrid algorithm and for a wider class of elliptic PDEs. LFA will allow us to make an educated guess about which combination of multigrid components will be effective for tensor-product multigrid and its Riemannian counterpart of the next section.

We have already seen in the previous chapter that equation (5.2) is a linear equation,

$$\mathcal{L}_l \operatorname{vec}(X_l) = \operatorname{vec}(C_l), \quad \mathcal{L}_l := A_l \otimes I_l + I_l \otimes A_l, \tag{5.3}$$

where $\operatorname{vec}(\cdot)$ denotes the vectorization operator (B.5). If we set $N_l := n_l^2$, then $\mathcal{L}_l \in \mathbf{R}^{N_l \times N_l}$ and $A_l \in \mathbf{R}^{n_l \times n_l}$.

5.2.1 Standard multigrid

We first briefly review the standard multigrid principle. Suppose we have coarsened a number of grids G_l for $l = 1, \ldots, l_{\max}$. Then linear multigrid is based on three components (see, e.g., Trottenberg *et al.* (2000)):

1. A hierarchy of discrete problems

   $$A_l x_l = f_l, \quad l = 1, \ldots, l_{\max},$$

   defined on each grid G_l. One can take the direct discretization of the PDE on each grid G_l, or a more complicated variation like Galerkin.

2. Transfer operators

   $$I_l^{l-1} : G_l \to G_{l-1}, \quad I_{l-1}^{l} : G_{l-1} \to G_l$$

   to go from one grid to the other. They are called the restriction and prolongation operator. Common examples are the full weighting operator for I_l^{l-1} and linear interpolation for I_{l-1}^{l}.

3. A smoother $\texttt{smooth}^{\nu}(x_l, A_l, f_l)$. Typically, this is a linear iteration based on a very cheap approximation $\widetilde{A}_l$ of A_l:

   $$x_l \leftarrow x_l + \omega_l p_l \quad \text{with } p_l \text{ the solution of } \widetilde{A}_l p_l = f_l - A_l x_l \tag{5.4}$$

   and ω_l a dampening parameter. Popular choices are damped Richardson ($\widetilde{A}_l = I_l$), damped Jacobi ($\widetilde{A}_l$ the diagonal part of A_l) and Gauss–Seidel ($\widetilde{A}_l$ the triangular part of A_l); and any banded variant, like line-Jacobi. These iterations are not necessarily good at solving $A_l x_l = f_l$, but they do reduce the high-frequency components of the error very well.

Based on these components, one can formulate the usual linear two-grid cycle, listed in Algorithm 3. Since this iteration cycles between two grids, we employ the standard notation of G_h for the fine grid and G_H for the coarse grid. Likewise, A_h denotes the discretization on the fine grid G_h and A_H on G_H. The other multigrid components are denoted analogously: the restriction I_h^H, the prolongation I_H^h, the coarse grid correction e_H and its prolongation e_h; and, the residual r_h and its restriction r_H.

The effectiveness of multigrid is highly dependent on the specific choice of these components. In specific, the coarsening and the smoother have to be complementary. Since this is a key topic in multigrid theory, we will not go into detail here and we assume that our choice of multigrid components is indeed effective and efficient.

Algorithm 3 Linear two-grid cycle

for $i = 1, 2, \ldots$ **do**
  $\overline{x}_h = \texttt{smooth}^{\nu_1}(x_h^{(i)}, A_h, f_h)$
  $r_h = f_h - A_h \overline{x}_h$
  $r_H = I_h^H r_h$
  $e_H = \texttt{cgc}(A_H, r_H) \equiv A_H^{-1} r_H$
  $e_h = I_H^h e_H$
  $\hat{x}_h = \overline{x}_h + e_h$
  $x_h^{(i+1)} = \texttt{smooth}^{\nu_2}(\hat{x}_h, A_h, f_h)$
end for

5.2.2 Tensor-product multigrid

In order to use the two-grid cycle for the Lyapunov equation (5.3), we need to generalize the previous multigrid components to the Lyapunov case. Since most classical multigrid components for two- and three-dimensional PDEs are already built using tensor products of lower dimensional variants, e.g., bilinear interpolation, we can keep on taking tensor products and obtain operators for the Lyapunov equation. This results in operations that act on the tensor-product space $\mathbf{R}^{n_l} \otimes \mathbf{R}^{n_l} \simeq \mathbf{R}^{n_l^2}$. In the following, we will denote grids and operators that belong to this tensor-product space by a calligraphic symbol, e.g., $\mathcal{G}_l$ is the grid for $\mathcal{L}_l$.

Next, we will assume that there are multigrid components available for an efficient linear multigrid iteration that solves $A_l x_l = f_l$. The aim is now to build the components for the tensor-product multigrid that solves (5.2) using the components for solving $A_l x_l = f_l$.

Hierarchy of grids. For the tensor-product multigrid method we will simply take tensor products of the hierarchy of grids from the original multigrid solver. So we get $\mathcal{G}_l := G_l \otimes G_l$ for all levels l.

Once the grids are defined, we need a discrete representation of the PDE on these grids. On the finest grid, the operator $\mathcal{L}_{\text{lmax}} = A_{\text{lmax}} \otimes I_{\text{lmax}} + I_{\text{lmax}} \otimes A_{\text{lmax}}$ was given by construction. Besides the fine grid operator $\mathcal{L}_{\text{lmax}}$, one also needs a coarse grid operator $\mathcal{L}_l$ for all $0 < l < l_{\text{lmax}}$. We will only consider direct coarsened operators A_{l-1} and the corresponding tensor-product operator $\mathcal{L}_{l-1} := A_{l-1} \otimes I_{l-1} + I_{l-1} \otimes A_{l-1}$. The Galerkin coarsening, also popular in classical multigrid, has the disadvantage that the stencils may become larger and that they have to be computed recursively. Both of which may become problematic in higher dimensions; see also bin Zubair *et al.* (2007).

Intergrid transfers. Prolongation and restriction for the tensor grid are easily constructed as tensor products of the operators on G_l. Indeed, suppose $I_l^{l-1} : G_l \to G_{l-1}$ then $I_l^{l-1} \otimes I_l^{l-1}$ defines a suitable restriction $\mathcal{I}_l^{l-1} : G_l \otimes G_l \to G_{l-1} \otimes G_{l-1}$. Prolongation is analogous. From a practical point of view, these operators can be applied directly to a matrix, e.g., for the restriction we get

$$\mathcal{I}_l^{l-1} : \mathrm{vec}(X_l) \mapsto (I_l^{l-1} \otimes I_l^{l-1})\,\mathrm{vec}(X_l) = \mathrm{vec}(I_l^{l-1} X_l (I_l^{l-1})^T). \tag{5.5}$$

This means we apply the restriction operator I_l^{l-1} on each column and each row of the matrix X_l.

Smoothing. The smoothers that are based on an approximation $\widetilde{A}_l$ of A_l can be generalized directly to $\widetilde{\mathcal{L}}_l = \widetilde{A}_l \otimes I_l + I_l \otimes \widetilde{A}_l$ as an approximation of $\mathcal{L}_l$. Formally, this only requires solving systems with $\widetilde{\mathcal{L}}_l$. From this, we get a smoother on $\mathcal{G}_l$ with error amplification matrix $\mathcal{S}_l$.

Observe that solving systems with $\widetilde{\mathcal{L}}_l$ is again solving a Lyapunov equation, namely

$$\widetilde{A}_l P_l + P_l \widetilde{A}_l^T = C_l, \tag{5.6}$$

which looks very expensive. In some cases, however, $\widetilde{A}_l$ will be of a very simple form and these systems can be solved quickly. Indeed, for Richardson ($\widetilde{A}_l = I_l$) we simply have $P_l = C_l/2$. Furthermore, when the matrices $\widetilde{A}_l$ have a constant bandwidth for all levels l, equation (5.6) can be solved in $O(n^2)$ as well. This is the case for Jacobi and line-Jacobi (after a trivial reordering) but not always for Gauss-Seidel.

Final algorithm The previous list contains all the components necessary for a tensor-product multigrid that solves the Lyapunov equation. By writing the iteration in terms of matrices, we obtain Algorithm 4.

5.2.3 Local Fourier analysis

The convergence of a multigrid method can be proved in several ways. One way is by means of Local Fourier Analysis (LFA) (Trottenberg *et al.*, 2000, Ch. 4), which assumes that the PDE has constant coefficients and is defined on an infinite domain. A general discrete operator with non-constant coefficients can be analyzed through local linearization and replacement by an operator with constant coefficients. Although this is a simplification of realistic PDEs, multigrid practice has learned that LFA can still deliver useful insight in the inner workings of a multigrid algorithm. In specific, it can guide us when choosing the coarsening and smoother. We extend LFA for the tensor-product multigrid method.

Algorithm 4 Tensor-product multigrid with V-cycle ($\gamma = 1$) or W-cycle ($\gamma = 2$).

Require: level l, initial guess $X_l^{(0)}$, rhs C_l

Pre-smoothing:
  solve P_l in $\widetilde{A}_l P_l + P_l \widetilde{A}_l^T = C_l - A_l X_l^{(0)} - X_l^{(0)} A_l^T$
  $X_l^{(1)} = X_l^{(0)} + \omega_l P_l$
Restrict residual:
  $R_{l-1} = I_l^{l-1}(C_l - A_l X_l^{(1)} - X_l^{(1)} A_l^T)(I_l^{l-1})^T$
Coarse grid correction
if $l = 1$ **then**
  Exact solve:
    solve E_{l-1} in $A_{l-1}E_{l-1} + E_{l-1}A_{l-1}^T = R_{l-1}$
else
  Recursive multigrid call:
    $E_{l-1} = \texttt{tensor-mg}((l-1, 0, R_{l-1})$.
    $E_{l-1} = \texttt{tensor-mg}^{\gamma-1}(l-1, E_{l-1}, R_{l-1})$.
end if
Prolongate coarse grid correction and correct:
  $X_l^{(2)} = X_l^{(1)} + I_{l-1}^l E_{l-1}(I_{l-1}^l)^T$
Post-smoothing:
  solve P_l in $\widetilde{A}_l P_l + P_l \widetilde{A}_l^T = C_l - A_l X_l^{(2)} - X_l^{(2)} A_l^T$
  $X_l^{(3)} = X_l^{(2)} + \omega_l P_l$

Classic LFA. Let us introduce the standard LFA notation (Trottenberg *et al.*, 2000; Wienands & Joppich, 2005) adapted to a general d-dimensional setting. Since we will only be using two levels, standard notation in LFA uses a subscript $_h$ for a fine grid and $_H$ for a coarse grid. We associate a fixed mesh width $h = (h_1, \ldots, h_d)$ with an infinite grid $G_h = \{x = (x_1, \ldots, x_d) = \kappa h = (\kappa_1 h_1, \ldots, \kappa_d h_d), \kappa \in \mathbf{Z}^d\}$ with $\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ the space of integers. On this grid, the discrete operator A_h corresponds to a difference stencil $[s_\kappa]_h$:

$$A_h w_h(x) = \sum_\kappa s_\kappa w_h(x + \kappa h),$$

where s_κ are constant coefficients. The formal eigenfunctions or Fourier modes of this operator are given by $\varphi(\theta, x) = e^{\mathrm{i}\theta x h} = e^{\mathrm{i}\theta_1 x_1/h_1} \cdots e^{\mathrm{i}\theta_1 x_d/h_d}$ for $x \in G_h$ with formal eigenvalue or symbol

$$\widetilde{A}_h(\theta) = \sum_\kappa s_\kappa e^{\mathrm{i}\theta\kappa}.$$

The frequency $\theta \in \mathbf{R}^d$ varies continuously in the analysis with $\theta \in [-\pi, \pi)^d$.

In addition, we assume a coarse grid $G_H = \{x = \kappa H, \kappa \in \mathbf{Z}^d\}$. The mesh-width H depends on the type of coarsening, e.g. standard coarsening gives $H_i = 2h_i$ for

all $i = 1, \ldots, d$. Based on the aliasing on this coarsened grid, one can classify the Fourier modes into high and low frequency components on G_h. This results in a space for the Fourier modes, called the $2h$-harmonics, which are indistinguishable on the coarse grid G_H.

It is well known that the two-grid cycle is invariant for this space of $2h$-harmonics for a wide range of smoothers, and restriction and prolongation operators. This results in an error amplification matrix M of the two-grid cycle that is similar to a block-diagonal matrix $\widetilde{M}$ with blocks $\widetilde{M}_i$ of size 2^d. Convergence factors can then be easily computed by iterating over all the blocks, or equivalently over all the low frequency Fourier modes:

$$\rho(M) = \max_{\theta^{\text{low}}} \rho(\widetilde{M}(\theta)) = \max_i \rho(\widetilde{M}_i) \tag{5.7}$$

and

$$\|M\|_2 = \max_{\theta^{\text{low}}} \sqrt{\rho(\widetilde{M}(\theta)^T \widetilde{M}(\theta))} = \max_i \sqrt{\rho(\widetilde{M}_i^T \widetilde{M}_i)}. \tag{5.8}$$

Here $\rho(M)$ denotes the asymptotic convergence factor of M, or equivalently the spectral radius of M.

LFA for tensor-product multigrid. The Fourier modes on the tensor-product grid $\mathcal{G}_h$ are simply the tensor product of the Fourier modes on G_h, $\varphi(\theta, x) = \varphi(\theta_1, x_1)\varphi(\theta_2, x_2)$, where we have used the partition $\theta = \theta_1 \otimes \theta_2$ with $\theta \in [-\pi, \pi)^{2d}$ and $\theta_1, \theta_2 \in [-\pi, \pi)^d$. Since the components of the tensor-product multigrid are tensor products of the corresponding components of a classic multigrid, one can compute the Fourier symbols accordingly. So, for the Fourier symbol of $\mathcal{L}_h$ in (5.2) we get

$$\begin{aligned}
\mathcal{L}_h\varphi(\theta, x) &= \left\{ \sum_\kappa s_\kappa \varphi(\theta_1, x_1 + \kappa h) \right\} \varphi(\theta_2, x_2) \\
&\quad + \varphi(\theta_1, x_1) \left\{ \sum_\kappa s_\kappa \varphi(\theta_2, x_2 + \kappa h) \right\} \\
&= \left(\widetilde{A}_h(\theta_1) + \widetilde{A}_h(\theta_2) \right) \varphi_1(\theta_1, x_1)\varphi_2(\theta_2, x_2) = \widetilde{\mathcal{L}}_h(\theta)\varphi(\theta, x)
\end{aligned}$$

The other tensor-product multigrid operators, i.e., the smoother, prolongation and restriction, are analogous. We remark that the Fourier symbols of a smoother on a colored grid, like RB-GS, are slightly more tedious to compute. However, there is a systematic way of deriving the symbols which can be implemented in a symbolic software package like Maple. See also bin Zubair *et al.* (2007) for a similar derivation of these symbols.

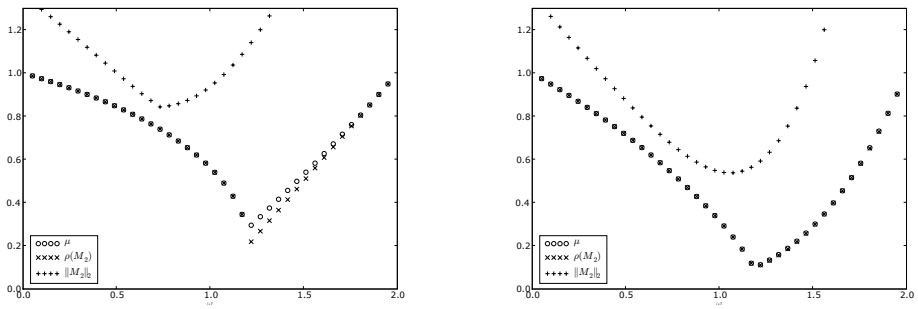


Figure 5.1: Smoothing factor μ, convergence factor $\rho(M)$ and spectral norm $\|M\|$ of the two-grid cycle with respect to the dampening factor ω. Left: one step pre-smoothing and no post smoothing and right: one step pre- and post-smoothing.

The space of $2h$-harmonics on $\mathcal{G}_h$ is again the tensor product of that on G_h. As a result, the error amplification matrix $\mathcal{M}$ of the two-grid cycle is similar to a block-diagonal matrix $\widetilde{\mathcal{M}}$ with blocks of size 2^{2d} instead on 2^d. The convergence factors are computed similarly to (5.7) and (5.8). Note however, that since $\theta \in [-\pi, \pi)^{2d}$ the total work to compute these estimates is squared compared to that of (5.7) and (5.8).

Results. We present a typical result of LFA for a Lyapunov equation stemming from an isotropic Poisson operator. For this kind of operators, Red-Black Gauss-Seidel (RB-GS) turns out to be a very cost-effective smoother. It was pointed out in Yavneh (1995) that for higher dimensional systems, RB-GS with an over-relaxation parameter ω can greatly benefit the smoothing factor, much more than for classic two- or three-dimensional systems. However, these results only consider the smoothing factor and they lack a complete analysis of the two-grid cycle. We have observed that a specific choice of multigrid components does have an influence on the total convergence factor. We show this for the influence of the number of pre- and post-smoothing steps.

In Figure 5.1 we present the convergence factors in function of this parameter ω for a 4-dimensional system resulting from LFA with RB-GS, full weighting and bilinear interpolation. This 4-dimensional system was obtained as a Lyapunov equation with a two-dimensional Poisson operator. It is clear that while the smoothing factor gives indeed a good indication of the asymptotic convergence factor $\rho(M)$, this is not so for the norm-wise reduction $\|M\|$. The reduction factor $\|M\|$ is considerably larger than the smoothing factor μ and they both have a different minimizer for ω.

5.3 Riemannian multilevel optimization

In this section, we propose a novel modification of the tensor-product multigrid from the previous section to a Riemannian algorithm on the manifold $\mathbf{S}_+^{n,p}$. This will allow us to compute low-rank solutions for Lyapunov equations based on minimizing f_E on $\mathbf{S}_+^{n,p}$ by a multilevel algorithm.

One of the advantages of this new method is that some of the typical smoothers of classic multigrid can be efficiently incorporated in this Riemannian multilevel algorithm. We will explain this generalization in three steps. Finally, we report on the numerical properties of this algorithm applied to two model problems that are typical in multigrid theory.

5.3.1 Nonlinear multigrid in Euclidean space

Anticipating the nonlinear nature of manifold $\mathbf{S}_+^{n,p}$, we switch to the framework of nonlinear multigrid; and more specifically, the Full Approximation Scheme (FAS); see, e.g., Trottenberg *et al.* (2000, Section 5.3). In case of nonlinear multigrid, the equation that needs to be solved can be nonlinear. On the fine grid G_h, this nonlinear equation is given by

$$A_h(x_h) = f_h. \tag{5.9}$$

On the coarse grid G_H, we have the usual coarse discretization of this operator, denoted by A_H. The standard[1] nonlinear FAS two-grid cycle to solve (5.9) is listed in Algorithm 5. The difference with the linear two-grid cycle of Alg. 3 is that the smoothed approximation $\overline{x}_h$ is also transferred to the coarse grid and the coarse grid equation is modified to

$$A_H(w_H) = A_H(\overline{x}_H + e_H) = r_H + A_H(\overline{x}_H).$$

Since A_H is nonlinear, one usually solves w_H in this equation by correcting the initial iterate $\overline{x}_H$ by some correction e_H. Observe that when A_h and A_H are linear operators, Alg. 5 reduces to Alg. 3.

This FAS two-grid cycle is visualized in Figure 5.2. At each grid, the iteration approximately solves an equation with a particular right-hand side vector:

$$b_h \equiv f_h, \quad \text{on } G_h, \tag{5.10}$$

$$b_H \equiv r_H + A_H(\overline{x}_H), \quad \text{on } G_H. \tag{5.11}$$

Observe that Fig. 5.2 clearly emphasizes the difference between iterates (points in Euclidean space) and updates (vectors). It is obvious that the coarse grid

[1] For simplicity we assume that the restriction operator for $\overline{x}_h$ is the same as for r_h.

Algorithm 5 Nonlinear FAS two-grid cycle

for $i = 1, 2, \ldots$ **do**
  $\overline{x}_h = \texttt{smooth}^{\nu_1}(x_h^{(i)}, A_h, f_h)$
  $r_h = f_h - A_h(\overline{x}_h)$
  $r_H = I_h^H r_h$
  $\overline{x}_H = I_h^H \overline{x}_h$
  solve $A_H(\overline{x}_H + e_H) = r_H + A_H(\overline{x}_H)$ for e_H
  $e_h = I_H^h e_H$
  $\hat{x}_h = \overline{x}_h + e_h$
  $x_h^{(i+1)} = \texttt{smooth}^{\nu_2}(\hat{x}_h, A_h, f_h)$
end for

corrections e_h and e_H are the updates of the iterates $\overline{x}_h$ and $\overline{x}_H$, respectively. In addition, the action of a smoother also constitutes an update vector. In case of a linear equation, the smoother (5.4) is already in this form:

$$\overline{x}_h = x_h^{(i)} + \omega_h p_h$$

$$x_h^{(i+1)} = \hat{x}_h + \omega_h \hat{p}_h$$

with p_h and $\hat{p}_h$ the solutions of a cheap linear system. Before generalizing this cycle to a manifold, we need to turn it into an optimization algorithm first.

5.3.2 Multilevel optimization in Euclidean space

In the previous chapter, we proposed the objective function f_E of eq. (4.13) and showed in Section 4.4.1 that minimizing this function amounts to solving a Lyapunov equation. Applied to the Lyapunov equation (5.1), the function f_E will have a natural multilevel structure. This is made explicit with the notation

$$f_l(x) := \operatorname{tr}(xA_l x) - \operatorname{tr}(xC_l). \tag{5.12}$$

Furthermore, in the two-grid context, f_h will mean the objective function f_E using the fine grid matrices A_h; and likewise for f_H for the coarse grid.

Suppose now that we are minimizing the fine grid function $f_h(x_h)$ and that at iterate $\overline{x}_h$ we have somehow smoothed the error (more on this later). How can we compute an approximation of this smooth error on the coarse grid? In other words, which coarse grid objective function do we need to minimize in $\overline{x}_H := I_h^H \overline{x}_h$, the coarse iterate? Considering the principle behind FAS, this means we need to generalize the coarse grid correction equation

$$A_H(x_H) = A_H(\overline{x}_H + e_H) = I_h^H(r_h) + A_H(\overline{x}_H)$$

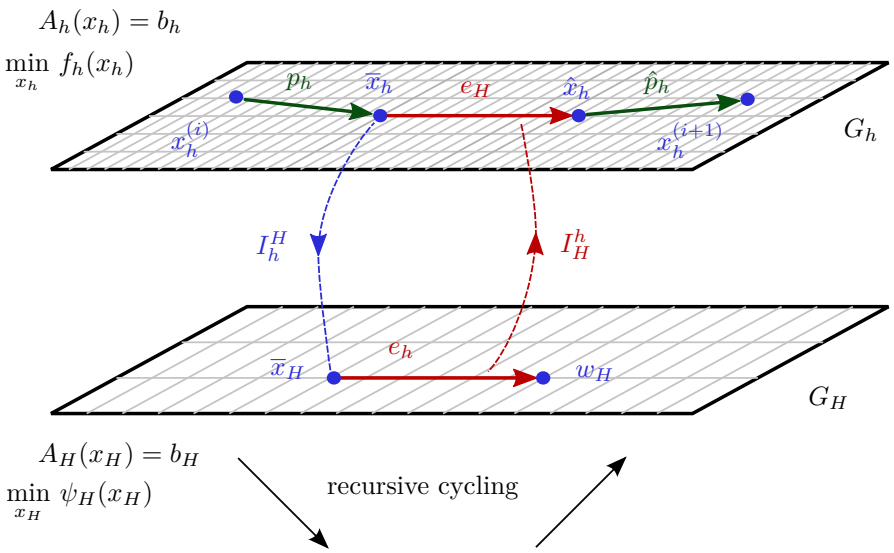


Figure 5.2: The FAS two-grid cycle for solving a nonlinear equation or minimizing an objective function.

to an appropriate coarse grid objective function $\psi_H(x_H)$.

Let $g^{\mathrm{E}}(\cdot,\cdot)$ denote the Euclidean inner product. Then it turns out that, by minimizing the objective function

$$\psi_H(x_H) := f_H(x_H) - g^{\mathrm{E}}(x_H, \operatorname{grad} f_H(\overline{x}_H) - I_h^H g_h), \quad g_h := \operatorname{grad} f_h(\overline{x}_h), \tag{5.13}$$

one obtains an effective two-grid cycle for optimizing f_h. This linear modification to f_H is one of the key ideas of multilevel optimization, as proposed in the MG/Opt method of Nash (2000) and Lewis & Nash (2005). The reasoning behind this modification is the following: if one takes as objective function the residual equation of a nonlinear equation, the minimizer of $\psi_H(x_H)$ reduces to the FAS coarse grid correction for this equation.

Remark that the minimization of (5.13) starts at the initial guess $\overline{x}_H$. Hence, one seeks an update e_H such that

$$\psi_H(\overline{x}_H + e_H) := f_H(\overline{x}_H + e_H) - g^{\mathrm{E}}(\overline{x}_H + e_H, \operatorname{grad} f_H(\overline{x}_H) - I_h^H g_h), \tag{5.14}$$

with $g_h := \operatorname{grad} f_h(\overline{x}_h)$, is sufficiently minimized.

For the coarse grid correction e_H to be useful, the error has to be representable on the coarse grid, i.e., it has to be smooth. Much like in classic multigrid, the usual cheap first-order optimization methods can be used to smooth the error. Practice has learned 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., Gratton *et al.* (2010).

Although we perform optimization, the principle behind the two-grid cycle does not change, except for the introduction of (5.13). In Figure 5.2 we therefore added the minimization of the fine and coarse scale objective functions f_h and ψ_H instead of solving equations with the operators A_h and A_H. Finally, we are in the position to generalize this to $\mathbf{S}_+^{n,p}$.

Remark 5.1. Our exposition of multilevel optimization is overly simplistic. Although the presented key concepts are generally straightforward to understand, the actual implementation as an optimization algorithm is much more involved. In contrast to linear multigrid, (multilevel) optimization methods need to be robustified with line-searches or a Trust-Region updating scheme. This is an active area of research and we refer to Borzí (2005); Gratton *et al.* (2008); Wen & Goldfarb (2009) for the actual convergence proofs of these methods. Since the aim of this chapter is to point out the proof-of-concept without convergence proofs, we skip these technicalities. In the actual implementation of the algorithm, we used a Trust-Region mechanism.

5.3.3 Generalization to Riemannian manifolds

The problem at hand is the minimization of a fine scale objective function f_l on a Riemannian manifold $\mathcal{M}_l$. As usual, we have a series of grids, in this case manifolds, denoted by $\mathcal{M}_l$. Each manifold, will have its own metric g_l and the tangent space at each point $x_l \in \mathcal{M}_l$ is denoted by $T_{x_l}\mathcal{M}_l$.

Very briefly, recall the general principle of retraction-based optimization of Figure 5.3. Given a retraction R_x, in each step on the algorithm, a lifted objective function

$$\widehat{f}_x(\xi) := f \circ R_x$$

is approximately minimized in $\xi \in T_x\mathcal{M}$ by a method from standard Euclidean optimization. In the next step, we construct a new lifted function, and the process repeats itself.

We detail how the two-grid optimization cycle from above can be generalized to this retraction-based framework. The actual Riemannian multilevel algorithm is then obtained by recursively applying this two-grid scheme. Let $\mathcal{M}_h$ denote a fine scale manifold and $\mathcal{M}_H$ a coarse scale manifold. Then in Figure 5.4, we have pictured the Riemannian two-grid cycle between the manifolds $\mathcal{M}_h$ and $\mathcal{M}_H$. This

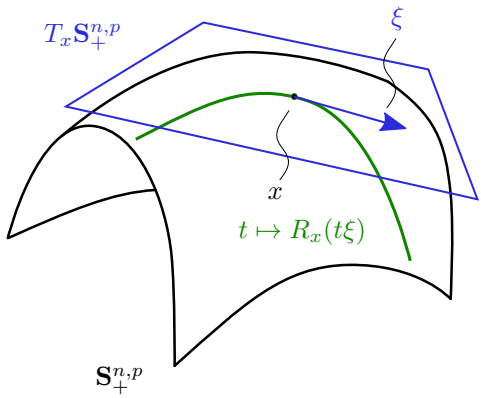


Figure 5.3: Retraction-based optimization.

two-grid cycle is based on the three typical multigrid components (as listed in Section 5.2.1).

Transfer operators. As usual, we need intergrid transfer operators between the fine and coarse manifolds:

$$I_h^H : \mathcal{M}_h \to \mathcal{M}_H, \tag{5.15}$$

$$I_H^h : \mathcal{M}_H \to \mathcal{M}_h. \tag{5.16}$$

Due to the difference between elements on a manifold and their tangent vectors, we need an additional set of intergrid transfer operators for the tangent vectors:

$$\hat{I}_h^H : T_{x_h}\mathcal{M}_h \to T_{x_H}\mathcal{M}_H, \quad \text{with } x_h \in \mathcal{M}_h \text{ and } x_H := I_h^H(x_h) \in \mathcal{M}_H, \tag{5.17}$$

$$\hat{I}_H^h : T_{x_H}\mathcal{M}_H \to T_{x_h}\mathcal{M}_h, \quad \text{with } x_H \in \mathcal{M}_H \text{ and } x_h := I_H^h(x_H) \in \mathcal{M}_h. \tag{5.18}$$

Obviously, $\hat{I}_h^H$ depends on I_h^H, and $\hat{I}_H^h$ on I_H^h.

Smoothers. The smoother on $\mathcal{M}_h$ is a cheap method that minimizes f_h: given $x_h^{(i)}$, it returns a tangent vector ξ_h such that after retraction, the error of the new iterate $\overline{x}_h = R_{x_h^{(i)}}(\xi_h)$ is smooth.

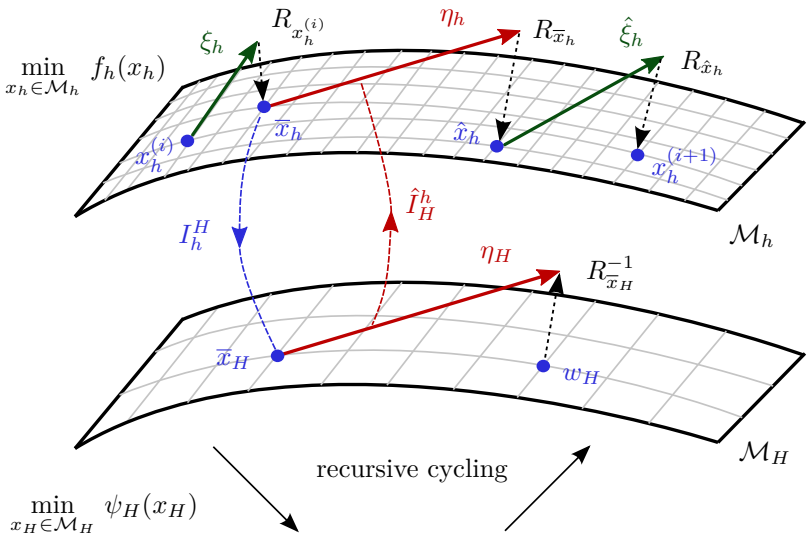


Figure 5.4: The Riemannian two-grid cycle for minimizing an objective function on a manifold.

Although there are many choices, we base our smoother on a familiar Riemannian optimization method: the truncated CG algorithm as used in RTR. The tCG method will give a tangent vector that, after a possible dampening, is retracted to get a new iterate. Observe that one successful iteration of tCG is a step of steepest descent. In case of linear systems, this update is closely related to the Richardson iteration, which is a well-known smoother in classic multigrid. We expect that this Riemannian iteration will have similar smoothing properties.

The tCG algorithm can benefit from preconditioning by a positive definite approximation of the Hessian. This allows us to use more powerful preconditioners than the basic Richardson iteration. Suppose we can approximate the Riemannian Hessian by its diagonal, then one successful application of tCG will be a step of the Jacobi iteration. It is clear that better preconditioners should lead to better smoothers. We will demonstrate that this is indeed the case in the numerical experiments of Section 5.4.

A hierarchy of discrete problems. The problem on $\mathcal{M}_h$ is the minimization of the actual objective function f_h. On the coarse level, we need to modify the objective function in a similar way as (5.13). To generalize this, we first rewrite

the Euclidean modification. Let $\kappa_H := \operatorname{grad} f_H(\overline{x}_H) - I_h^H g_h$ and $g_h := \operatorname{grad} f_h(\overline{x}_h)$. Then (5.13) satisfies

$$\begin{aligned}\psi_H(\overline{x}_H + e_H) &:= f_H(\overline{x}_H + e_H) - g^{\mathrm{E}}(\overline{x}_H + e_H, \kappa_H),\\ &= f_H(\overline{x}_H + e_H) - g^{\mathrm{E}}(e_H, \kappa_H) + c, \quad c \in \mathbf{R}.\end{aligned}$$

Since $g^{\mathrm{E}}(\overline{x}_H, \kappa_H)$ is constant, we can omit it from the objective function.

Recognizing e_H as a tangent vector in the tangent space of $\overline{x}_H$, we can lift the function ψ_H by aid of the retraction $R_{\overline{x}_H}$. This gives a coarse objective function that is suitable for Riemannian optimization:

$$\hat{\psi}_{\overline{x}_H} : T_{\overline{x}_H}\mathcal{M}_H \to \mathbf{R},\ \eta_H \mapsto f_H(R_{\overline{x}_H}(\eta_H)) - g_{\overline{x}_H}(\eta_H, \kappa_H), \tag{5.19}$$

with $\kappa_{\overline{x}_H} := \operatorname{grad} f_H(\overline{x}_H) - \hat{I}_h^H(\operatorname{grad} f_h(\overline{x}_h))$. Obviously, the gradients are the Riemannian gradients of f_H and f_h. By definition of $\hat{I}_h^H$, the subtraction of the two tangent vectors is well defined.

Vector transport of the linear modification. The above coarse grid function does not tell the whole picture. In a true *multi*-grid algorithm, the coarse grid function $\hat{\psi}_{\overline{x}_H}$ will be minimized using several intermediate steps. However, since retractions are only locally well-defined and numerically stable (see Section 3.5.4 for a discussion), one cannot simply minimize the lifted function $\hat{\psi}_{\overline{x}_H}$ completely through the tangent space $T_{\overline{x}_H}\mathcal{M}_H$ and expect that this minimizer is a good coarse grid correction. Instead, we apply the principle of moving tangent vectors by lift-and-retract through several tangent spaces.

Suppose we have obtained an η_H as approximation of the minimization of (5.19). In the new iterate, $w_H := R_{\overline{x}_H}(\eta_H)$, we construct an updated lifted function

$$\hat{\psi}_{w_H} : T_{w_H}\mathcal{M}_H \to \mathbf{R},\ \eta_H \mapsto f_H(R_{w_H}(\eta_H)) - g_{w_H}(\eta_H, \kappa_{w_H}) \tag{5.20}$$

Since f_H is defined on the whole manifold $\mathcal{M}_H$, the first term is immediately well defined. However, the second term is an inner product on $T_{w_H}\mathcal{M}_H$. As such, κ_{w_H} needs to lie in $T_{w_H}\mathcal{M}_H$. This requires transporting the vector $\kappa_{\overline{x}_H}$ from $T_{\overline{x}_H}\mathcal{M}_H$ to $T_{w_H}\mathcal{M}_H$. We will denote this operation by

$$P_{x\to y} : T_x\mathcal{M}_H \to T_y\mathcal{M}_H, \quad x, y \in \mathcal{M}_H. \tag{5.21}$$

In classic differential geometry, transporting tangent vectors from one tangent space to another is usually done by *parallel translation* (Lee, 1997, Chap. 4). However, for general manifolds the parallel translation of tangent vectors is costly to compute. Luckily, we do not need it per se. In the same way that retractions are cheap

replacements of geodesics, it suffices to approximate parallel transport locally and to first-order.

We will not go into detail here on the necessary properties of this approximation—applied to $\mathbf{S}_+^{n,p}$ this transport will be obvious—except noting that the operation $P_{x\to y}$ is in fact the *vector transport* of Absil *et al.* (2008, Section 8.1). Vector transport is a much-used technique when generalizing Euclidean optimization algorithms, see, e.g., Edelman *et al.* (1999); Qi *et al.* (2010).

Riemannian two-grid cycle. In Algorithm 6, we listed the final two-grid cycle to optimize an objective function on a Riemannian manifold. All elements should be clear, except one: the computation of the tangent vector that corresponds to the coarse grid correction by $R^{-1}_{\overline{x}_H}$. On a complete Riemannian manifold where the retraction is the exponential mapping Exp of (A.3.2), this inverse is always well-defined. However, in general, the retraction mapping is only locally invertible and its inverse may still be rather expensive to compute . Luckily, the numerical experiments indicate that this inverse does not need to be exact (first-order accuracy is again sufficient) and that locality is not a problem in the retraction-based framework.

Algorithm 6 ML-RTR: Riemannian two-grid cycle to minimize f_h

for $i = 1, 2, \ldots$ **do**
  $\xi_h = \mathtt{tCG}^{\nu_1}(x_h^{(i)}, f_h)$
  $\overline{x}_h = R_{x_h^{(i)}}(\omega \xi_h)$
  $\overline{x}_H = I_h^H \overline{x}_h$
  $\kappa_{w_H^{(1)}} = \operatorname{grad} f_H(\overline{x}_H) - \hat{I}_h^H(\operatorname{grad} f_h(\overline{x}_h))$
  for $k = 1, 2, \ldots, k_{\max}$ **do**
    define the coarse function $\hat{\psi}_{w_H^{(k)}}$ of (5.20) based on $\kappa_{w_H^{(k)}}$
    minimize $\hat{\psi}_{w_H^{(k)}}$ for $\eta_H^{(k)}$ by the RTR method
    $w_H^{(k+1)} = R_{w_H^{(k)}}(\eta_H^{(k)})$
    $\kappa_{w_H^{(k+1)}} = P_{w_H^{(k)} \to w_H^{(k+1)}}(\kappa_{w_H^{(k)}})$
  end for
  $\eta_H = R^{-1}_{\overline{x}_H}(w_H^{(k_{\max})} - \overline{x}_H)$
  $\eta_h = \hat{I}_H^h \eta_H$
  $\hat{x}_h = R_{\overline{x}_h}(\eta_h)$
  $\hat{\xi}_h = \mathtt{tCG}^{\nu_2}(\hat{x}_h, f_h)$
  $x_h^{(i+1)} = R_{\hat{x}_h}(\omega \hat{\xi}_h)$
end for

Remark 5.2. Like Remark 5.1, the actual implementation of Alg. 6 will need to be robustified with line-search or Trust-Region. Since combining the existing

robustifying techniques of multilevel and Riemannian optimization is likely to be rather technical and involved, this is beyond the scope of the current chapter. We emphasize that in our numerical experiments, we used a Trust-Region mechanism inspired by the RTR algorithm of Absil *et al.* (2007) combined with the Recursive Trust-Region method from Gratton *et al.* (2008). However, we do not prove convergence of this algorithm.

5.3.4 Multilevel Lyapunov equations on $\mathbf{S}_+^{n,p}$

In this section, we can finally specify how the ML-RTR algorithm can be used to minimize f_E of eq. (4.13) on $\mathbf{S}_+^{n,p}$ when the Lyapunov equations originate from the discretization of an elliptic PDE. The intuition will be based on the tensor-product multigrid of Section 5.2.2 for this Lyapunov equation.

Suppose we have an efficient multigrid strategy for $A_l x_l = f_l$ available on the grids G_l with corresponding transfer operators and smoothers. If we regard an an element of $G_l \otimes G_l$ as a matrix, then we can approximate it by a low-rank matrix. Suppose we do this for every level l and in addition, we assume that our approximations are symmetric and positive definite. Now we have constructed a hierarchy of manifolds $\mathcal{M}_l := \mathbf{S}_+^{n_l,p}$. For simplicity, we leave the rank p fixed and coarsen only n_l, the spatial variables.

By constructing the manifolds $\mathcal{M}_l$ in this way, many of the operators of the general ML-RTR algorithm can be chosen as the standard components of the tensor-product multigrid of Section 5.2.2. Together with our LFA analysis for this tensor-product multigrid, one can make educated guesses which combination of multigrid components are supposedly efficient. Of course, the crucial difference is that we need to account for the tangent space. We will outline this in more detail for some of these components.

Smoothers. Like we mentioned in the previous section, the tCG method can be used as a smoother, similar to Richardson. In addition, we can easily incorporate more powerful smoothers in the following way. In Section 4.6, we introduced $P\mathcal{L}_l P$, with P the projection onto the tangent space, as an effective preconditioner for the tCG method. However, computing the action of this preconditioner required solving shifted systems of A_l, which can still be expensive. Luckily, for a smoother a very crude approximation of A_l and thus $\mathcal{L}_l$ suffices.

Suppose the smoother of standard multigrid on $A_l x_l = f_l$ uses a smoother with $\widetilde{A}_l$ as approximation for A_l. Now we know from the Local Fourier Analysis of Section 5.2.3 that

$$\widetilde{\mathcal{L}}_l := \widetilde{A}_l \otimes I_l + I_l \otimes \widetilde{A}_l$$

will be an effective smoother for the tensor-product multigrid. Hence, we can use $P\widetilde{\mathcal{L}}_l P$ as a preconditioner for tCG and (hopefully) benefit from the better smoothing properties of $\widetilde{\mathcal{L}}_l$.

In addition, this smoother can be efficiently computed by the numerical procedure of Section 4.6.2. Applied to $\widetilde{\mathcal{L}}_l$, the dominating costs of the smoother will be solving p^2 shifted systems $\widetilde{A}_l$. For a smoother this is supposed to be (relatively) cheap.

Vector transport. Since the tangent vectors of $\mathbf{S}_+^{n,p}$ are embedded in $\mathbf{R}^{n\times n}$, the vector transport of (5.21) can be based on moving tangent vectors and projecting them onto the new tangent space. In other words,

$$P_{x\to y} : T_x\mathbf{S}_+^{n,p} \to T_y\mathbf{S}_+^{n,p},\ \xi_x \mapsto \mathrm{P}_y^{\mathrm{t}}(\xi_x)$$

with $\mathrm{P}_y^{\mathrm{t}}$ the orthogonal projection onto $T_y\mathbf{S}_+^{n,p}$. This a well-defined vector transport, see Absil *et al.* (2008, Section 8.3.1), and can be efficiently computed.

Transfer operators Since $x_l \in \mathcal{M}_l \in \mathcal{G}_l$ with $\mathcal{G}_l$ the grids of tensor product multigrid, we can simply take the intergrid transfer operators of tensor-product multigrid for the operators (5.15)–(5.16). In addition, these operators can be efficiently applied to the factored representation of $x_l = V_l D_l V_l^T$.

For the transfer operators $\hat{I}_h^H$ and $\hat{I}_H^h$ of (5.17)–(5.18), we also use the intergrid transfer operators of tensor-product multigrid. However, this transfer has to be followed by an orthogonal projection onto the new tangent space. This is similar as the vector transport of above.

5.4 Numerical results

In this section we report on the numerical properties of Alg. 6, our Riemannian Multilevel algorithm, applied to PDE-related Lyapunov equations. The computations were done with MATLAB R2009b on a 64-bit Intel Pentium Xeon 2.66 GHz with $\epsilon_{mach} \simeq 2 \cdot 10^{-16}$. The ML-RTR algorithm was implemented using MATLAB as a generalization of the RTR algorithm of the previous chapter. The reported timings are wall clock times that include all necessary computations.

The multigrid components were all implemented in MATLAB. Due to the relatively expensive cost of recursive cycling in MATLAB, this is not the best computing environment for a multigrid type algorithm. Still, our implementation should be sufficient to illustrate the typical multigrid properties: mesh-independent convergence and a computational cost that scales linearly with the problem size.

5.4.1 Computation of the residual

The computation of the residual should account for the fact that the discretization matrices A can be very large and highly ill-conditioned. We use two different measures for the relative residual. The first one is the popular choice

$$r_1(x) := \|AxM^T + MxA^T\|_{\mathrm{F}}/\|C\|_{\mathrm{F}}, \tag{5.22}$$

which we also used in the previous chapter. This residual can be efficiently computed by equation (4.29).

Another choice, which is numerically sometimes more sensible, is the following from Barrett *et al.* (1993, Sec. 4.2.1):

$$r_2(x) := \|AxM^T + MxA^T\|_{\mathrm{F}}/(\|\mathcal{L}\|_2\|x\|_{\mathrm{F}} + \|C\|_{\mathrm{F}}), \quad \|\mathcal{L}\|_2 \simeq 2\|A\|_2. \tag{5.23}$$

In case $M = I$, we have $\|\mathcal{L}\|_2 = 2\|A\|_2$, and when $M \neq I$, this is approximately correct for discretizations with quasi-uniform meshes. This relative residual is used also in the low-rank Lyapunov solver of Simoncini (2007).

The difference with residual r_1 and r_2 becomes clear if one thinks in terms of backward errors. Let $c := \mathrm{vec}(C)$ and let $\hat{x}$ be an approximation to the system $\mathcal{L}x = c$. Recall that the backward error of $\hat{x}$ is the smallest change $\max\{\|\delta\mathcal{L}\|/\|\mathcal{L}\|, \|\delta c\|/\|c\|\}$ to the problem $\mathcal{L}x = c$ that makes $\hat{x}$ the solution of $(\mathcal{L} + \delta\mathcal{L})\hat{x} = c + \delta c$. Now the tolerance $r_2(x) < \tau$ is equivalent to demanding that the backward error satisfies $\|\delta\mathcal{L}\| < \tau\|\mathcal{L}\|$ and $\|\delta c\| < \tau\|c\|$. On the other hand, the tolerance $r_1(x) < \tau$ puts $\|\delta\mathcal{L}\| = 0$ and only demands $\|\delta c\| < \tau\|c\|$. Hence, r_1 is more stringent than r_2 and it can be very hard to satisfy this condition.

However, in the context of PDEs, the residual r_2 can be too optimistic as a backward error estimate since all the perturbations $\delta\mathcal{L}$ in $\mathcal{L}$ are treated equally. For PDEs, the matrices A and M typically have a fixed sparsity pattern, hence $\delta\mathcal{L}$ should also be sparse. In principle, it is possible to devise a residual which accounts for these perturbations in the right way, but this measure is not so easily generalizable to our low-rank matrices. We included both measures in our numerical experiments to compare their suitability.

5.4.2 One-dimensional diffusion

For the first example, consider the one-dimensional (1d) diffusion equation

$$\frac{\partial}{\partial x}\left(a(x)\frac{\partial u}{\partial x}\right) = b(x), \quad b(x) := e^x \sin(3\pi x), \tag{5.24}$$

with homogeneous boundary conditions on the interval $[0, 1]$. After discretization using the usual three-point stencil with a mesh of size $h = 1/(n + 1)$, one obtains

a tridiagonal matrix $A \in \mathbf{R}^{n\times n}$ and a vector $c \in \mathbf{R}^n$ that represents $b(x)$. The corresponding Lyapunov equation is then defined using this A and with $M = I$ and $C = cc^T$.

Regarding the choice of $a(x)$, we considered two examples that are a model problem for linear multigrid on $Ax = b$: one problem with constant coefficient $a(x) = 1$ and one with variable coefficient

$$a(x) = 1 + e^{x/3}\cos(5x)/10. \tag{5.25}$$

In the latter case, the coefficient $a(x)$ varies smoothly around 1 with only a 10% variation. Like in linear multigrid, this problem should behave much in the same way as the constant coefficient case. As a result, standard coarsening and a damped point Jacobi smoother is a sensible multigrid strategy for both problems. We took the diagonal part of A as preconditioner in $\mathcal{L}$ for tCG followed by a dampening of $\omega = 0.5$. This is the equivalent of a damped Jacobi smoother in standard multigrid. Equation (5.24) is discretized with meshes $h = 32 \cdot 2^{-l}$ for growing levels $l = 0, 1, 2, \ldots, l_{\text{lmax}}$. The coarsest problem of size $n = 32$, i.e., level $l = 0$, is solved to full numerical accuracy by the RTR method on $\mathbf{S}_+^{32,8}$, as outlined in the previous chapter.

We now recursively apply Alg. 6 to minimize f_E on the manifold $\mathbf{S}_+^{n,p}$ with $p = 8$. For the corresponding Lyapunov equation, the rank 8 optimizers of f_E have a relative error of about 10^{-12}, which is more than accurate enough for the mesh-sizes we will consider below. The convergence of this method is displayed in Figure 5.5.

One can observe that the chosen multigrid strategy is indeed effective since both problems show a mesh-independent convergence. However, in contrast to linear multigrid, the convergence of our multilevel algorithm has a pronounced transient behavior. While the asymptotic convergence rate is indeed mesh-independent, this behavior begins later for the bigger problems.

Take for example the problem of size $n = 262144$ in the left panel of Figure 5.5. In the beginning, the method converges fast, but around iteration 10, the Trust-Region mechanism does not make good progress and the proposed iterates are rejected. It is not until iteration 25 that convergence to the optimizer starts again.

In Table 5.1, the time per iteration and the final residuals are displayed. First notice that all problems were solved up to full numerical accuracy if the error is measured by the relative residual (5.23). On the other hand, the residual (5.22) was only reduced to about $\sqrt{\epsilon_{\text{mach}}}$. Since these problems were clearly over-solved, we displayed the time *per iteration* in Table 5.1. Despite the fact that the implemented ML-RTR algorithm has to follow a Trust-Region strategy and not a fixed V-cycle pattern, the amount of work scales almost linearly with the problem size.

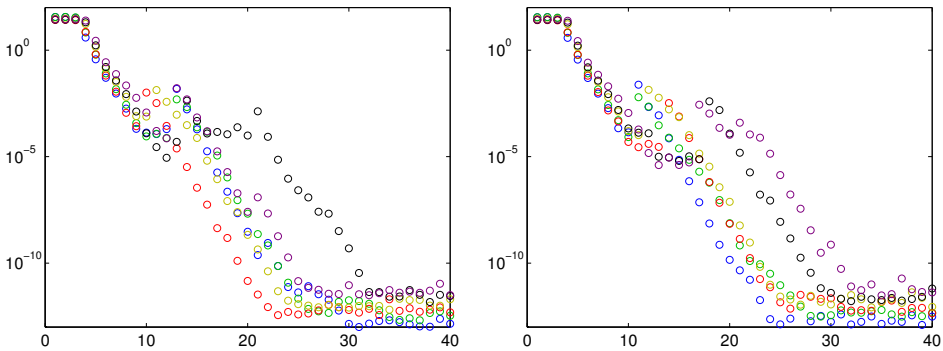


Figure 5.5: Convergence of the gradient for ML-RTR. The Lyapunov equation originates from a 1d diffusion equation with constant (left) and variable coefficients (right). The sizes of the discretizations are 16384 (○), 32768 (○), 65536 (○), 131072 (○), 262144 (○) and 524288 (○).

$a(x)$	size n	16384	32768	65536	131072	262144	524288
1	time (s.)	2.1	4.0	8.1	16.4	33.4	76.4
	$r_1(x_{40})/10^{-8}$	1.3	2.2	3.1	29	47	208
	$r_2(x_{40})/10^{-16}$	11	4.58	1.57	3.74	1.51	1.66
(5.25)	time (s.)	2.1	4.0	8.1	16.4	34.0	71.7
	$r_1(x_{40})/10^{-8}$	3.0	3.1	3.3	21	44	211
	$r_2(x_{40})/10^{-16}$	23.6	5.90	1.60	2.61	1.22	1.60

Table 5.1: Time per iteration and final residuals r_1 (5.22) and r_2 (5.23) for the problem of Figure 5.5.

5.4.3 Two-dimensional diffusion

As next example, we solve the singularly perturbed ($\epsilon \to 0$) two-dimensional (2d) diffusion equation

$$\frac{\partial^2 u}{\partial x^2} + \epsilon \frac{\partial^2 u}{\partial y^2} = b(x,y), \quad b(x,y) := e^{x+2y} \sin(3\pi x) \sin(\pi y), \tag{5.26}$$

with homogeneous boundary conditions on the square $[0,1]^2$. After discretization using the usual five-point stencil with a mesh of size $h = 1/(\sqrt{n}+1)$, one obtains the system matrix

$$A = \overline{A} \otimes I_n + \epsilon I_n \otimes \overline{A} \in \mathbf{R}^{n \times n},$$

with $\overline{A}$ the discretization of a one-dimensional Laplace equation with a three-point stencil, and a vector $c \in \mathbf{R}^n$ that represents $b(x, y)$. The corresponding Lyapunov equation is again constructed using this A and with $M = I$ and $C = cc^T$.

Classical multigrid theory tells us that a point smoother in combination with standard coarsening will not be an effective solver for equations with anisotropic diffusion. We can confirm this for the Riemannian multilevel algorithm as well. Figure 5.6 shows the convergence history for the cases $\epsilon = 1$ and $\epsilon = 10^{-6}$ when solving the Lyapunov equations again for a rank 8 minimizer of f_E. The smoother is a damped point Jacobi using $\omega = 0.75$ and implemented as a diagonal preconditioner for tCG. It is clear that this strategy is indeed effective for $\epsilon = 1$ but not for $\epsilon = 10^{-6}$.

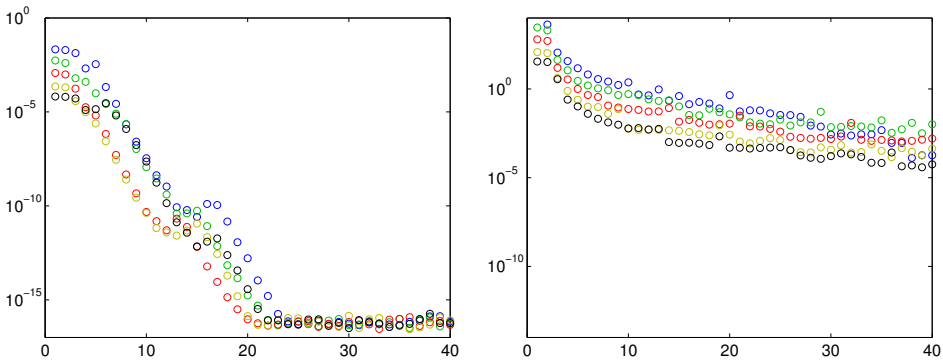


Figure 5.6: Convergence of the gradient for ML-RTR with a point smoother. The Lyapunov equation originates from a 2d diffusion equation with isotropic (left panel) and anisotropic (right panel) diffusion. The sizes of the discretizations are 4096 (○), 16384 (○), 65536 (○), 262144 (○) and 1048576 (○).

ϵ	size n	4096	16384	65536	262144	1048576
1	time (s.)	0.37	1.25	5.01	21.4	99.4
	$r_1(x_{40})/10^{-8}$	3.8	4.9	5.1	5.2	5.2
	$r_2(x_{40})/10^{-16}$	300	23	1.1	0.01	0.06
10^{-6}	time (s.)	0.37	1.29	4.98	21.1	97.8
	$r_1(x_{40})$	0.14	6.3	8.7	6.4	5.6
	$r_2(x_{40})/10^{-8}$	17	1.3	0.4	0.09	0.007

Table 5.2: Time per iteration and final residuals r_1 (5.22) and r_2 (5.23) for the problem of Figure 5.6. Observe that the residuals for $\epsilon = 10^{-6}$ are 10^8 times greater than those for $\epsilon = 1$.

It is well known that a line smoother in combination with standard coarsening is an effective strategy for multigrid on the PDE (5.26) when $\epsilon \to 0$. From the LFA analysis in Section 5.2.3, we know that a smoother which is based on the approximation

$$\widetilde{L} = \widetilde{A} \otimes I_N + I_N \otimes \widetilde{A}, \quad \text{with } \widetilde{A} := \operatorname{diag}(\overline{A}) \otimes I_n + \epsilon I_n \otimes \overline{A},$$

will be effective for tensor-product multigrid on the corresponding Lyapunov equation. In Figure 5.7, we have applied this operator as a preconditioner in tCG for ML-RTR. It is obvious that this is again an effective strategy.

Comparing the time per iteration in Table 5.3 with that of Table 5.2, we see that the method using line Jacobi is basically twice as costly as that of point Jacobi, while both methods converge equally fast (compare the left panels of Figure 5.6 and Figure 5.7). However, this extra cost of line Jacobi gives a more robust multigrid strategy.

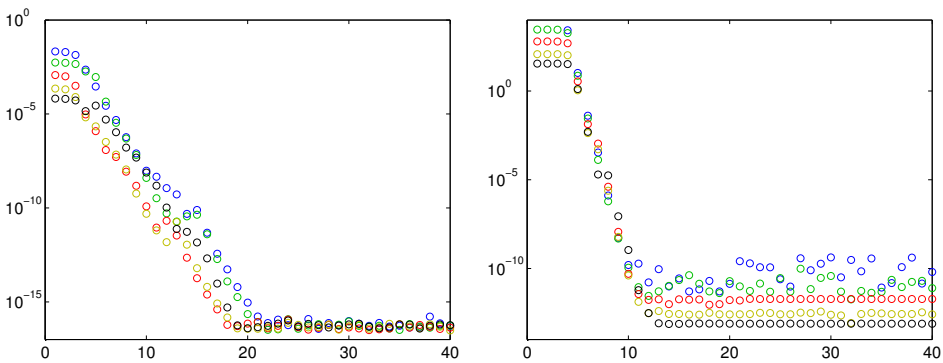


Figure 5.7: Same problem as Figure 5.6 but now with a line smoother.

ϵ	size n	4096	16384	65536	262144	1048576
1	time (s.)	0.6	2.4	9.7	41	183
	$r_1(x_{40})/10^{-8}$	3.8	4.9	5.1	5.2	5.2
	$r_2(x_{40})/10^{-16}$	300	23	1.5	0.01	0.01
10^{-6}	time (s.)	0.7	2.4	9.5	40	173
	$r_1(x_{40})/10^{-8}$	27	28	6.0	5.1	5.3
	$r_2(x_{40})/10^{-16}$	3822	243	3.1	0.16	0.01

Table 5.3: Time per iteration and final residuals r_1 (5.22) and r_2 (5.23) for the problem of Figure 5.7.

5.5 Conclusions

In this chapter, we proposed a multilevel optimization algorithm for minimizing objective functions on $\mathbf{S}_{+}^{n,p}$ that have a natural multilevel character. To apply this algorithm to the computation of low-rank solutions of a Lyapunov equation, we first discussed the tensor-product multigrid. The multigrid components for the Riemannian multilevel algorithm could then be based on a Local Fourier Analysis of this tensor-product multigrid. Finally, we showed numerically that this multilevel algorithm exhibits the two key properties of a multigrid solver: mesh-independent convergence and a cost per iteration that scales linearly with the problem size.

6

A homogeneous space geometry with complete geodesics

In this chapter we will introduce another description for the geometry of $\mathbf{S}_+^{n,p}$, namely as a homogeneous space of the general linear group $\mathbf{GL}^n$. In comparison to the description as an embedded submanifold of Chapter 3, homogeneous spaces have more structure and a richer theory. Together with the machinery of Riemannian submersions, this will lead to the main contribution of this approach: complete geodesics.

Most of the contents of this chapter is submitted as Vandereycken *et al.* (2010).

6.1 Introduction

The main reason for studying the geometry of $\mathbf{S}_+^{n,p}$ in Chapter 3 is our use of Riemannian optimization to solve rank-constrained problems, like the Lyapunov equation. However, the Riemannian geometry of $\mathbf{S}_+^{n,p}$ is interesting in its own right. While the embedded geometry is effective for the Riemannian optimization algorithms, it is not attractive as a mathematical space since it is not complete. In this chapter, we derive a geometry with complete geodesics.

We will assume a basic knowledge about homogeneous spaces. While Section 2.4.2 provides an introduction, we refer to Boothby (1986); Lee (2003) for a more detailed treatment. Before deriving this homogeneous space geometry, we commence with a rationale for studying the geometry of $\mathbf{S}_+^{n,p}$ in a more mathematically oriented way. We will explain why the Riemannian geometry of $\mathbf{S}_+^{n,p}$ is indeed interesting in its own right, why low-rank matrices are useful and why we want a geometry with complete geodesics.

6.1.1 The benefit of Riemannian geometry

Symmetric positive semidefinite matrices $\mathbf{S}_+^n$ are fundamental objects in many areas of applied mathematics. They have their use in modeling as well as in computation. Among the many fields of applications, we mention only their use as covariance matrices in statistics (Huber, 1981), as optimization variables in semidefinite programming (Boyd & Vandenberghe, 2004) and as kernels in machine learning (Lanckriet *et al.*, 2004).

In case of the full-rank matrices, the Riemannian geometry of $\mathbf{S}_{++}^n$ turned out be a useful mathematical tool since it led to certain results that would otherwise be difficult to obtain. These can be mainly theoretical, as in Smith (2005) where the author analyses the accuracy for estimation problems on $\mathbf{S}_{++}^n$; the so-called Cramér–Rao bound analysis. Since $\mathbf{S}_{++}^n$ is not a vector space, one cannot perform a classical Cramér–Rao bound analysis on this set. The approach taken in Smith (2005) relies heavily on the rich Riemannian geometry of $\mathbf{S}_{++}^n$ to derive an intrinsic analysis, suitable for the manifold.

An applications-oriented example is that of Pennec *et al.* (2006). It discusses a Riemannian framework for image processing with $\mathbf{S}_{++}^n$ used in Diffusion Tensor Imaging (DTI). In order to have meaningful results in DTI, the computational operations like filtering, interpolation, diffusion and restoration of missing data, need to incorporate the nonlinear structure of $\mathbf{S}_{++}^n$. By borrowing from the typical objects in differential geometry, the computing framework of Pennec *et al.* (2006) consists of algorithms that are again intrinsically defined on $\mathbf{S}_{++}^n$. The experimental results indicate that this produces better results than methods that deal with the nonlinear structure of $\mathbf{S}_{++}^n$ in an ad-hoc way.

Sometimes the geometric link is established in a later phase, like in Nesterov & Todd (2008) where the optimization paths in short-steps methods for semidefinite programming (SDP) are related to the geodesics on $\mathbf{S}_{++}^n$. In particular, algorithms whose iterates lie on (or close to) such geodesics are optimal and presumably efficient. The Riemannian viewpoint then provides guidance for the construction of new efficient interior-point methods for optimizing over other affine manifolds.

These (and many other) applications seem to have a common rationale for employing

Riemannian geometry. By addressing matrix problems through Riemannian eyes, one has the many classic theorems and properties from differential geometry at ones disposal "for free". This is of course beneficial for the analysis and the design of practical algorithms on matrix manifolds. Some caveat is in order though. Since the foundations of differential geometry are theoretical, it usually requires some effort to translate them to constructive theorems and efficient algorithms.

6.1.2 The benefit of low-rank matrices

The aforementioned applications exploiting the Riemannian geometry of $\mathbf{S}_+^n$ are restricted to the full-rank matrices. Since the classical matrix algorithms on $\mathbf{S}_+^n$, like the eigenvalue decomposition, have at least an $O(n^3)$ complexity, they do not scale well to problems with large n. In order to have scalable algorithms—ideally, with $O(n)$ complexity—one usually resorts to some parsimonious structure in the problem that can be exploited to reduce the complexity. Sparsity is a well-known and obvious data structure. Another much-used remedy in matrix computation is to work with low-rank matrices.

Take again the problem of computing the eigenvalue decomposition. If the eigenvalues of an s.p.s.d. matrix decay exponentially, this matrix admits a very accurate low-rank approximation. Hence, it suffices to compute only the dominant eigenspace of size $n \times p$, with $p \ll n$. From a numerical point of view, these low-rank approximations can be as accurate as the original matrix, yet they require only $O(np)$ memory. In addition, even if the eigenvalues do not decay exponentially, for many real-world applications, low-rank approximations are still useful; see, e.g., Rosipal & Girolami (2001); Jadea *et al.* (2003) in the context of machine learning with low-rank kernels.

If the matrix to approximate has a fast matrix-vector product, there exist many algorithms that can compute its dominant eigenspaces efficiently. One of the more popular algorithms is based on the implicitly restarted Lanczos method (Lehoucq *et al.*, 1997), implemented as `eigs` in MATLAB. On the other hand, when the matrix is full, it is significantly harder to compute a low-rank approximation efficiently. If its entries are known explicitly, one can for example use techniques from cross-approximations and hierarchical schemes like in Hackbusch (1999); Tyrtyshnikov (2000); Bebendorf & Rjasanow (2003), or perform an updating and downsizing strategy as in Hoegaerts *et al.* (2007); Mastronardi *et al.* (2010).

When the matrix is not explicitly known but only via an equation or an observation, the low-rank approximation problem is significantly more difficult. The rank-constrained problem central in this thesis—the Lyapunov matrix equation—belongs to this class, but there are many others, e.g., the low-rank matrix completion problem (Candès & Tao, 2009). There are of course methods that can solve specific

rank-constrained problems. However, most of them are not immediately generalized to other matrix problems or are based on ad-hoc techniques.

6.1.3 The need for a complete space

The existence of geodesics that can be extended indefinitely has a profound global implication: it guarantees that any two points on the manifold can be connected by a geodesic. The minimal length among all such connecting geodesics coincides with the usual distance function on the manifold. Furthermore, by virtue of the Hopf–Rinow Theorem 2.27, the manifold will be a complete metric space w.r.t. this distance function. Complete spaces are attractive in many ways and this is also the case when applied to manifolds.

Here, we mention only one application area that benefits greatly from these complete geodesics, namely, that of optimization and Newton algorithms on Riemannian manifolds. The prototype example of an optimization algorithm performs a line-search along some search direction. In case of a manifold, this search can be done along a geodesic. The fact that the geodesics are complete and available in an efficient closed-form makes this line-search straightforward and well-defined. Alternatively, Trust-Region methods on manifolds also benefit since every modification to the Trust-Region radius will be algorithmically possible.

We do not wish to claim that geodesics are necessary for optimizing on a manifold. Retraction-based Riemannian optimization, as introduced in Section 2.8 and used in the rest of this thesis, dispenses with geodesics in favor of their first-order approximations, called retractions. Retractions that can be extended indefinitely surely share the same practical advantages as complete geodesics. However, most global convergence theory for optimization on Riemannian manifolds assumes completeness; see, e.g., Ferreira & Svaiter (2002); Dedieu *et al.* (2003); Li & Wang (2006); Absil *et al.* (2007); Alvarez *et al.* (2008)—although Yang (2006) does not.

In this chapter, we therefore introduce another geometry which leads to complete geodesics, namely, that of a homogeneous space of the general linear group $\mathbf{GL}^n$. In comparison to the existing approaches, homogeneous spaces have more structure and a richer theory; see, e.g., Kobayashi & Nomizu (1963). Together with the machinery of Riemannian submersions as in Cheeger & Ebin (1975), O'Neill (1983) and Gallot *et al.* (2004), we can derive the complete geodesics based on the geodesics of $\mathbf{GL}^n$. Due to the importance of scalability in the applications of low-rank matrices, we give much attention to deriving efficient expressions for the typical objects of differential geometry, like these geodesics.

6.2 Manifold $\mathbf{S}_+^{n,p}$ as a homogeneous space

Recall from Chapter 3 that the set

$$\mathbf{S}_+^{n,p} = \{x \in \mathbf{S}_+^n \mid \operatorname{rank}(x) = p\} = \{YY^T \mid Y \in \mathbf{R}_*^{n\times p}\}$$

is a smooth manifold of dimension $pn - p(p-1)/2$. Furthermore, we introduced the Lie group action

$$\theta : \mathbf{GL}^n \times \mathbf{S}^n \to \mathbf{S}^n,\ (A, x) \mapsto AxA^T.$$

The orbit of this action through

$$e := \begin{bmatrix} I_p & 0_{p\times(n-p)} \\ 0_{(n-p)\times p} & 0_{(n-p)\times(n-p)} \end{bmatrix} \in \mathbf{S}_+^{n,p}$$

is denoted by

$$\theta_e : \mathbf{GL}^n \to \mathbf{S}_+^{n,p},\ A \mapsto AeA^T.$$

Observe that the image of θ_e is indeed the manifold $\mathbf{S}_+^{n,p}$.

6.2.1 Transitivity of the Lie group action

In this chapter, we exploit an important property of the Lie group action θ, namely transitivity (see A.2). This allows us to describe $\mathbf{S}_+^{n,p}$ as a so-called homogeneous space. We will assume a basic knowledge of homogeneous spaces as introduced in Section 2.4.2. For more on this fundamental topic in Riemannian geometry, we refer to Boothby (1986). The derivations of Chapter 3 already made use of the fact that mapping θ is transitive, but only in an implicit way. In this chapter, the transitivity will be used more explicitly.

Proposition 6.1. *Manifold $\mathbf{S}_+^{n,p}$ is a homogeneous space with transitive $\mathbf{GL}^n$-action*

$$\theta : \mathbf{GL}^n \times \mathbf{S}_+^{n,p} \to \mathbf{S}_+^{n,p},\ (A, x) \mapsto AxA^T. \tag{6.1}$$

Proof. Transitivity of θ means that there exists an $A \in \mathbf{GL}^n$ for every $x_1, x_2 \in \mathbf{S}_+^{n,p}$ such that $\theta(A, x_1) = x_2$. Let $x_1 = Y_1Y_1^T$ and $x_2 = Y_2Y_2^T$ with $Y_1, Y_2 \in \mathbf{R}_*^{n\times p}$, then such an A is given by

$$A = \begin{bmatrix} Y_2 & Z_2 \end{bmatrix} \begin{bmatrix} Y_1 & Z_1 \end{bmatrix}^{-1},$$

for any $Z_1, Z_2 \in \mathbf{R}_*^{n\times(n-p)}$ with $Z_1 \perp Y_1$ and $Z_2 \perp Y_2$. □

6.2.2 Quotient manifold $\mathbf{GL}^n/\mathrm{Stab}_e$

Mapping θ_e is surjective but not injective. In this section, we will construct a bijection based on θ_e that gives us an alternative description of manifold $\mathbf{S}_+^{n,p}$, namely as a quotient manifold. The general theory for quotient manifolds as explained in Section 2.4.1, can now be applied to $\mathbf{GL}^n$.

An elegant way to express the many-to-one relation of θ_e is with the so-called stabilizer or stability group, denoted by Stab_e. This is the maximal subgroup of $\mathbf{GL}^n$ that leaves the action of θ_e fixed, that is $\theta_e(L) = e$ for all $L \in \mathrm{Stab}_e$. It is not difficult to see that this group is given by

$$\mathrm{Stab}_e = \begin{bmatrix} \mathbf{O}^p & \mathbf{R}^{p\times(n-p)} \\ 0_{(n-p)\times p} & \mathbf{GL}^{n-p} \end{bmatrix},$$

with $\mathbf{O}^p$ the orthogonal group. Observe that Stab_e in indeed a closed group in $\mathbf{GL}^n$.

The many-to-one relation can now be factored out by means of the following equivalence relation for $A, B \in \mathbf{GL}^n$:

$$A \sim B \quad \Longleftrightarrow \quad B = AL \text{ for some } L \in \mathrm{Stab}_e.$$

The equivalence class containing A is denoted by $[A] := \{B \in \mathbf{GL}^n : A \sim B\}$. Now, let

$$\mathbf{GL}^n/\mathrm{Stab}_e := \{[A] : A \in \mathbf{GL}^n\} \tag{6.2}$$

denote the quotient of $\mathbf{GL}^n$ by this equivalence relation. It will be the set of all the equivalence classes of $\sim$. Mapping

$$\pi : \mathbf{GL}^n \to \mathbf{GL}^n/\mathrm{Stab}_e, \quad A \mapsto [A] \tag{6.3}$$

is the canonical projection or *quotient map*. In order to avoid ambiguity regarding to which space $[A]$ belongs, we will use $\pi(A)$ to denote $[A]$ viewed as an element of $\mathbf{GL}^n/\mathrm{Stab}_e$, and $\pi^{-1}(\pi(A))$ for $[A]$ as a subset of $\mathbf{GL}^n$.

Since Stab_e is a closed subgroup in $\mathbf{GL}^n$, it is a Lie subgroup. By virtue of the Homogenous Space Construction Theorem 2.20, the left coset space $\mathbf{GL}^n/\mathrm{Stab}_e$ is a smooth quotient manifold. It is a standard result for homogeneous spaces that this quotient manifold is diffeomorphic to $\mathbf{S}_+^{n,p}$.

Proposition 6.2. *Mapping*

$$\Theta_e : \mathbf{GL}^n/\mathrm{Stab}_e \to \mathbf{S}_+^{n,p},\ \pi(A) \mapsto \theta_e(A)$$

is a diffeomorphism with $\theta_e = \Theta_e \circ \pi$. *In other words,* $\mathbf{GL}^n/\mathrm{Stab}_e \simeq \mathbf{S}_+^{n,p}$ *with dimension* $np - p(p-1)/2$.

Proof. Since $\mathbf{S}_+^{n,p}$ is a smooth manifold, we can use e.g. Boothby (1986, Th. IV.9.3). Alternatively, the fact that Stab_e is a closed Lie subgroup of $\mathbf{GL}^n$ makes the set $\mathbf{S}_+^{n,p}$ a smooth manifold, see Boothby (1986, Th. IV.9.6). The dimension of $\mathbf{GL}^n/\mathrm{Stab}_e$ equals $\dim(\mathbf{GL}^n) - \dim(\mathrm{Stab}_e)$. □

The following commuting diagram summarizes the construction of this section.

$$\begin{array}{ccc} \mathbf{GL}^n & & \\ \downarrow{\scriptstyle \pi} & \searrow^{\theta_e} & \\ \mathbf{GL}^n/\mathrm{Stab}_e & \xrightarrow[\Theta_e]{} & \mathbf{S}_+^{n,p} \end{array} \tag{6.4}$$

So $\theta_e = \Theta_e \circ \pi$ is the typical decomposition of a surjective function as a projection followed by a bijection.

6.2.3 Representatives for equivalence classes

Manifold $\mathbf{GL}^n/\mathrm{Stab}_e$ is a quotient space that contains abstract equivalence classes as elements. Since we prefer to work with concrete matrices as elements, we will use a representative $A \in \mathbf{GL}^n$ to represent the equivalence class $[A]$. Apart from the non-uniqueness of this representative, this poses no real problems. (In fact, this freedom will be advantageous later on.) Let us see how these representatives look like.

First note that throughout the paper we will use the following partitioning for an $A \in \mathbf{GL}^n$:

$$A = \begin{bmatrix} Y & Z \end{bmatrix}, \quad Y \in \mathbf{R}_*^{n\times p},\, Z \in \mathbf{R}_*^{n\times(n-p)}. \tag{6.5}$$

By using the notation Y and Z we will implicitly always mean matrices that satisfy the form of eq. (6.5). For this A, the equivalence class $\pi^{-1}(\pi(A))$ can be written as

$$\begin{aligned} \pi^{-1}(\pi(A)) &= A\,\mathrm{Stab}_e, \\ &= \begin{bmatrix} Y\,\mathbf{O}^p & Y\,\mathbf{R}^{(n-p)\times p} + Z\,\mathbf{GL}^{n-p} \end{bmatrix}, \\ &= \begin{bmatrix} Y\,\mathbf{O}^p & Y\,\mathbf{R}^{(n-p)\times p} + Y_\perp\,\mathbf{GL}^{n-p} \end{bmatrix}, \end{aligned}$$

where $Y_\perp \in \mathbf{R}^{n\times(n-p)}$ is any orthonormal basis for the orthogonal complement of Y in $\mathbf{GL}^n$. As long as $A \in \mathbf{GL}^n$, the last $n-p$ columns of A are all equivalent. So only the first p columns have an influence on π and these columns belong to the equivalence class $Y\,\mathbf{O}^p$.

6.2.4 Reductive space $\mathbf{GL}^n/\mathbf{O}^n$

When $p = n$, the homogeneous space geometry in Prop. 6.1 has an important additional property, namely that of a *reductive space*, see Smith (2005). We will not go into detail why reductiveness is an important property (see, e.g., Nomizu (1954); Kobayashi & Nomizu (1963)), since it is not necessary for the rest of the thesis, except noting that reductiveness leads to nice properties. One of these is that there is a natural choice for a metric on $\mathbf{S}_+^{n,n} \simeq \mathbf{GL}^n/\mathbf{O}^n$, which will be invariant to the $\mathbf{GL}^n$-action. This invariance is a powerful property to exploit (e.g., in deriving geodesics) and has been used in most of the literature on the Riemannian geometry of $\mathbf{S}_+^{n,n}$.

When $p < n$, it is reasonable to try to find a metric on $\mathbf{GL}^n/\mathrm{Stab}_e$ which is also invariant to the $\mathbf{GL}^n$-action. This is however not possible, as was shown in Bonnabel & Sepulchre (2009) for $\mathbf{S}_+^{2,1}$ by a continuity argument on this metric. In fact, we will show that $\mathbf{GL}^n/\mathrm{Stab}_e$ is never a reductive space for all rank deficient cases, i.e., $p < n$.

Reductiveness is usually formulated in terms of Lie algebras, which are, in our case, the tangent spaces at the identity of Lie groups. In addition, we denote these Lie algebras by the usual Fraktur letters. Let $\mathfrak{g} = \mathbf{R}^{n\times n}$ denote such a Lie algebra of $\mathbf{GL}^n$ and let

$$\mathfrak{h} = \begin{bmatrix} \mathrm{skew}(p) & \mathbf{R}^{p\times(n-p)} \\ 0 & \mathbf{R}^{(n-p)\times(n-p)} \end{bmatrix}$$

be the Lie algebra of Stab_e. The homogeneous space $\mathbf{GL}^n/\mathrm{Stab}_e$ is reductive if and only if there exists a subspace $\mathfrak{p}$ of $\mathfrak{gl}$, complementary to $\mathfrak{h}$, such that $H\mathfrak{p}H^{-1} \subseteq \mathfrak{p}$ for all $H \in \mathrm{Stab}_e$, see Kobayashi & Nomizu (1963, Chap. X). Differentiating, this implies that

$$\mathfrak{h}\mathfrak{p} - \mathfrak{p}\mathfrak{h} \subseteq \mathfrak{p}. \tag{6.6}$$

All subspaces of $\mathfrak{gl}$ complementary to $\mathfrak{h}$ are of the form

$$\mathfrak{p} = \left\{ \begin{bmatrix} S + \mathcal{W}(B,S) & \mathcal{D}(B,S) \\ B & \mathcal{E}(B,S) \end{bmatrix} : B \in \mathbf{R}^{(n-p)\times p}, S \in \mathbf{S}^p \right\},$$

where

$$\mathcal{W} : \mathbf{R}^{(n-p)\times p} \times \mathbf{S}^p \to \mathrm{skew}(p)$$

$$\mathcal{D} : \mathbf{R}^{(n-p)\times p} \times \mathbf{S}^p \to \mathbf{R}^{p\times(n-p)}$$

$$\mathcal{E} : \mathbf{R}^{(n-p)\times p} \times \mathbf{S}^p \to \mathbf{R}^{(n-p)\times(n-p)}$$

are linear maps. Since map $\mathcal{E}$ is linear we can decompose it as $\mathcal{E} = \mathcal{E}_1 + \mathcal{E}_2$ with $\mathcal{E}_1 : \mathbf{R}^{(n-p)\times p} \to \mathbf{R}^{(n-p)\times(n-p)}$, $\mathcal{E}_2 : \mathbf{S}^p \to \mathbf{R}^{(n-p)\times(n-p)}$, and similarly for $\mathcal{W}$ and $\mathcal{D}$.

Observe that the l.h.s. of condition (6.6) reduces to

$$[\mathfrak{h},\mathfrak{p}] = \begin{bmatrix}\Sigma & K\\ 0 & L\end{bmatrix}\begin{bmatrix}S+\mathcal{W} & \mathcal{D}\\ B & \mathcal{E}\end{bmatrix} - \begin{bmatrix}S+\mathcal{W} & \mathcal{D}\\ B & \mathcal{E}\end{bmatrix}\begin{bmatrix}\Sigma & K\\ 0 & L\end{bmatrix}$$
$$= \begin{bmatrix}\Sigma S+\Sigma\mathcal{W}+KB-S\Sigma-\mathcal{W}\Sigma & \Sigma\mathcal{D}+K\mathcal{E}-SK-\mathcal{W}K-\mathcal{D}L\\ LB-B\Sigma & L\mathcal{E}-BK-\mathcal{E}L\end{bmatrix}, \tag{6.7}$$

with $\Sigma \in \mathrm{skew}(p)$, $K \in \mathbf{R}^{p\times(n-p)}$ and $L \in \mathbf{R}^{(n-p)\times(n-p)}$. The proof is now as follows:

Proposition 6.3. *If $p < n$, then there are no linear maps $\mathcal{W},\mathcal{D},\mathcal{E}$ in (6.7) such that (6.6) is satisfied for all Σ, S, K, L, B arbitrary matrices of suitable form. Hence the homogeneous space $\mathbf{GL}^n/\mathrm{Stab}_e$ is reductive only for $p = n$.*

Proof. First, if $p = n$, linear maps $\mathcal{D},\mathcal{E}$ and matrices K, L, B are void. We immediately get that $\mathcal{W} = 0$ satisfies (6.6), as is known from Smith (2005).
Next, we proof the case $p < n$ by contradiction. Take $K = 0$ and $\Sigma = 0$, then condition (6.6) with (6.7) implies

$$\begin{bmatrix}0 & -\mathcal{D}(B,S)L\\ LB & L\mathcal{E}(B,S)-\mathcal{E}(B,S)L\end{bmatrix} = \begin{bmatrix}\mathcal{W}(LB,0) & \mathcal{D}(LB,0)\\ LB & \mathcal{E}(LB,0)\end{bmatrix}. \tag{6.8}$$

So one of the conditions is $\mathcal{E}(LB,0) = L\mathcal{E}(B,S)-\mathcal{E}(B,S)L$, or $\mathcal{E}_1(LB) = L\mathcal{E}_1(B)+L\mathcal{E}_2(S)-\mathcal{E}_1(B)L-\mathcal{E}_2(S)L$ for all L, B and S. Hence $L\mathcal{E}_2(S)-\mathcal{E}_2(S)L = 0$, or in other words, $\mathcal{E}_2(S)$ commutes with L. For general L and S, this can only be if $\mathcal{E}_2(S) = \alpha(S)I_{n-p}$ where α is a scalar-valued function.
Now take $L = 0$ and $\Sigma = 0$, then condition (6.6) with (6.7) satisfies

$$\begin{bmatrix}KB & K\mathcal{E}(B,S)-SK-\mathcal{W}(B,S)K\\ 0 & -BK\end{bmatrix} = \begin{bmatrix}X+\mathcal{W}(0,X) & \mathcal{D}(0,X)\\ 0 & \mathcal{E}(0,X)\end{bmatrix}, \tag{6.9}$$

where

$$X = (KB+B^TK^T)/2.$$

We get the condition $\mathcal{E}(0,(KB+B^TK^T)/2) = -BK$. From above, we already know that $\mathcal{E}(0,(KB+B^TK^T)/2) = \alpha((KB+B^TK^T)/2)I_{n-p}$. This leads to a contradiction if BK is not a scalar, i.e., $n-p > 1$. If $n-p = 1$, then $\alpha(S) = -\operatorname{tr}(S)$ satisfies the condition.
We continue with proving the case $n-p = 1$. Take again (6.8), since the lower-right corner of the l.h.s. is now zero, it implies $\mathcal{E}(LB,0) = \mathcal{E}_1(LB) = 0$ for all L, B. Hence $\mathcal{E}_1 = 0$ and $\mathcal{E}(B,S) = -\operatorname{tr}(S)$. In addition (6.8) also implies $\mathcal{D}(LB,0) = -\mathcal{D}(B,S)L$, and, since L is a scalar, also $L\mathcal{D}_1(B) = -L\mathcal{D}_1(B)-L\mathcal{D}_2(S)$. From this we have $\mathcal{D} = 0$. From (6.8) we have also $\mathcal{W}(LB,0) = 0$, so $\mathcal{W}_1 = 0$. Finally, from (6.9) again, we get that $-\operatorname{tr}(S)K-SK-\mathcal{W}_2(S)K = 0$ for all K. Since $\mathcal{W}_2(S)$ is always skew-symmetric and $\operatorname{tr}(S)I_p+S$ symmetric, their sum can never be zero and we have a contradiction. □

6.2.5 The Riemannian metric

Since for $p < n$, manifold $\mathbf{S}_+^{n,p}$ is no longer reductive, there is not a natural metric anymore. The typical choice to equip $\mathbf{GL}^n$ with the Euclidean metric

$$\overline{g}^{\text{Eucl}}(\xi_A, \zeta_A) := \text{tr}(\xi_A^T \zeta_A), \quad \forall \xi_A, \zeta_A \in T_A\mathbf{GL}^n,\ A \in \mathbf{GL}^n,$$

turns out to be less then ideal when one is concerned about geodesics. Take e.g. curve $t \mapsto I_n - tI_n$. It is obviously a length minimizing geodesic, but it is not complete since at $t = 1$ its image is zero. Since the rationale of this chapter is a metric with complete geodesics, we disregard the Euclidean metric as candidate for a metric on $\mathbf{S}_+^{n,p}$.

Since all left- and right-invariant vector fields on $\mathbf{GL}^n$ are complete (Boothby, 1986, Cor. V.5.8), it is reasonable to have a left- or right-invariant metric also. (Note that both left- and right-invariant is not possible on $\mathbf{GL}^n$.) In this chapter, we therefore choose the *right-invariant metric* $\overline{g}$, defined as

$$\overline{g}_A(\eta_A, \nu_A) := \overline{g}_I(\eta_A A^{-1}, \nu_A A^{-1}) = \text{tr}(A^{-T}\eta_A^T \nu_A A^{-1}),$$
$$\text{for all } \eta_A, \nu_A \in T_A\mathbf{GL}^n,\ A \in \mathbf{GL}^n. \quad (6.10)$$

In Section 6.3, we will explicitly show that, although the geodesics of $(\mathbf{GL}^n, \overline{g})$ are not always right-invariant, they are still complete.

At first sight, an equally logical choice would be the left-invariant metric for which all left-invariant vector fields are complete. We will later see in Section 6.6.3 however that the left-invariant metric does not allow us to reuse the geodesics of $\mathbf{GL}^n$ in the same way as the right-invariant does.

6.2.6 The tangent space

Similar to the representatives of Section 6.2.3, we want to use the tangent space of $\mathbf{GL}^n$ to represent tangent vectors of $\mathbf{GL}^n/\text{Stab}_e$. This requires the notion of the vertical and horizontal space as detailed in Section 2.4.3: a specific decomposition of $T_A\mathbf{GL}^n \simeq \mathbf{R}^{n\times n}$.

Since Stab_e is closed, $\pi^{-1}(\pi(A))$ is a sub-manifold embedded in $\mathbf{GL}^n$ (Boothby, 1986, Lemma IV.9.7). This means that its tangent space is a subspace of the embedding space $\mathbf{R}^{n\times n}$ and it is called the *vertical space* $\mathcal{V}_A$. In $A = \begin{bmatrix} Y & Z \end{bmatrix}$ it is given by

$$\mathcal{V}_A = \begin{bmatrix} Y\,\text{skew}(p) & \mathbf{R}^{n\times(n-p)} \end{bmatrix} = A \begin{bmatrix} \text{skew}(p) & \mathbf{R}^{p\times(n-p)} \\ 0 & \mathbf{R}^{(n-p)\times(n-p)} \end{bmatrix}. \quad (6.11)$$

The *horizontal space* $\mathcal{H}_A$ is any complementary subspace of $\mathcal{V}_A$ in $\mathbf{R}^{n\times n}$ and we will take the orthogonal complement of $\mathcal{V}_A$ w.r.t. the right-invariant metric. So, using the right-invariant metric $\overline{g}$ as in eq. (6.10), the horizontal space becomes

$$\mathcal{H}_A = A^{-T}\begin{bmatrix}\mathbf{S}^p & 0\\ \mathbf{R}^{(n-p)\times p} & 0\end{bmatrix}A^T A. \tag{6.12}$$

The tangent space of $\mathbf{GL}^n/\mathrm{Stab}_e$ can now be represented uniquely by tangent vectors from the horizontal space. These are called the *horizontal lifts* and we will consistently denote this lift of $\xi_{\pi(A)} \in T_{\pi(A)}\mathbf{GL}^n/\mathrm{Stab}_e$ by $\overline{\xi}_A \in T_A\mathbf{GL}^n$. Let $\overline{\xi}$ denote such a unique horizontal lift on $\mathbf{GL}^n$ of a vector field ξ on $\mathbf{GL}^n/\mathrm{Stab}_e$. It satisfies

$$\mathrm{D}\pi(A)[\overline{\xi}_A] = \xi_{\pi(A)}, \quad \overline{\xi}_A \in \mathcal{H}_A. \tag{6.13}$$

We have in addition that the lifts are related along the equivalence class $\pi^{-1}(\pi(A))$ in the following way:

Proposition 6.4. *A horizontal vector field $\overline{\xi}$ of $\mathbf{GL}^n$ is the horizontal lift of a vector field ξ on $\mathbf{GL}^n/\mathrm{Stab}_e$ if and only if, for all $A \in \mathbf{GL}^n$, it holds that*

$$\overline{\xi}_{AL} = \overline{\xi}_A L, \quad \forall L \in \mathrm{Stab}_e.$$

Proof. First observe that $\mathcal{H}_I = L^{-T}\mathcal{H}_I L^T$ for all $L \in \mathrm{Stab}_e$, from which it follows that $\mathcal{H}_{AL} = \mathcal{H}_A L$. Thus, we have constructed a horizontal space that satisfies a connection (Kobayashi & Nomizu, 1963, Ch. II) on the principle bundle $\mathbf{GL}^n(\mathbf{GL}^n/\mathrm{Stab}_e, \mathrm{Stab}_e)$ and the proof follows by (Kobayashi & Nomizu, 1963, Prop. II.1.2). □

6.2.7 The Riemannian submersion

In the previous section we have seen that the tangent space of $\mathbf{GL}^n/\mathrm{Stab}_e$ can be represented by the horizontal space, which is embedded into $\mathbf{R}^{n\times n}$. We would like to reuse the right-invariant metric of $(\mathbf{GL}^n, \overline{g})$ in the same way by restricting it to the horizontal space. There is a caveat however. In order to have a well-defined metric, one has to make sure that the non-uniqueness of the horizontal lifts does not pose a problem. In particular, we will demand that the metric is invariant along the fiber.

This construction will make

$$\pi : \mathbf{GL}^n \to \mathbf{GL}^n/\mathrm{Stab}_e$$

into a *Riemannian submersion*. This type of submersion was first axiomatically proposed by B. O'Neill in O'Neill (1966) as a convenient property for general

submersions on fiber bundles, see also O'Neill (1983); Gallot *et al.* (2004); Absil *et al.* (2008). The key axiom (O'Neill, 1966, S2) is that $\mathrm{D}\pi(A)|_{\mathcal{H}_A}$, the differential of π restricted to $\mathcal{H}_A$ for fixed A, is an isometry. This allows one to derive several differential geometric objects on $\mathbf{GL}^n/\mathrm{Stab}_e$ based on the properties of $\mathbf{GL}^n$ restricted to $\mathcal{H}_A$ (and $\mathcal{V}_A$). Intuitively, one can state that since $\mathrm{D}\pi(A)|_{\mathcal{H}_A}$ is an isometry, it is sufficient to base derivations on concrete lifted tangent vectors in $\mathcal{H}_A$ instead of on abstract elements in $T_A\mathbf{GL}^n/\mathrm{Stab}_e$.

In Section 6.3 we will see that the geodesics are easily derived by virtue of this Riemannian submersion. The only thing we still need to show is that for our case π is indeed Riemannian when $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ has as metric the right-invariant metric of $(\mathbf{GL}^n, \overline{g})$ restricted to the horizontal space.

Proposition 6.5. *Let $\overline{g}$ be the right-invariant metric (6.10). Then the relation*

$$g_{\pi(A)}(\eta_{\pi(A)}, \nu_{\pi(A)}) = \overline{g}_A(\overline{\eta}_A, \overline{\nu}_A) = \mathrm{tr}(A^{-T}\overline{\eta}_A^T\overline{\nu}_A A^{-1}) \tag{6.14}$$

defines a Riemannian metric g on $\mathbf{GL}^n/\mathrm{Stab}_e$. The metric g turns the quotient map

$$\pi : \mathbf{GL}^n \to \mathbf{GL}^n/\mathrm{Stab}_e$$

into a Riemannian submersion and $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ is a Riemannian quotient manifold of $(\mathbf{GL}^n, \overline{g})$.

Proof. The lifted metric is invariant on each fiber

$$\begin{aligned}\overline{g}_{AL}(\overline{\eta}_{AL}, \overline{\nu}_{AL}) &= \overline{g}_I(\overline{\eta}_{AL}L^{-1}A^{-1}, \overline{\nu}_{AL}L^{-1}A^{-1}) \\ &= \overline{g}_I(\overline{\eta}_A A^{-1}, \overline{\nu}_A A^{-1}) = \overline{g}_A(\overline{\eta}_A, \overline{\nu}_A),\end{aligned}$$

hence $\mathrm{D}\pi(A)|_{\mathcal{H}_A}$ is an isometry for each A. This makes π a Riemannian submersion (O'Neill, 1983, Def. 7.44). □

6.2.8 Some useful expressions

We will derive some relations regarding the horizontal space which will be convenient later on. Take $A = \begin{bmatrix} Y & Z \end{bmatrix}$ as in (6.5). We have the identity

$$A^{-T} = \begin{bmatrix} \mathrm{P}_Z^\perp Y(Y^T\mathrm{P}_Z^\perp Y)^{-1} & \mathrm{P}_Y^\perp Z(Z^T\mathrm{P}_Y^\perp Z)^{-1} \end{bmatrix}.$$

So, every vector $\overline{\xi}_A \in \mathcal{H}_A$ can be written as

$$\begin{aligned}\overline{\xi}_A &= A^{-T}\begin{bmatrix} \mathbf{S}^p & 0 \\ \mathbf{R}^{(n-p)\times p} & 0 \end{bmatrix} A^T A \\ &= \begin{bmatrix} \mathrm{P}_Z^\perp & \mathrm{P}_Y^\perp Z \end{bmatrix}\begin{bmatrix} (Y^T\mathrm{P}_Z^\perp Y)^{-1}\mathbf{S}^p \\ (Z^T\mathrm{P}_Y^\perp Z)^{-1}\mathbf{R}^{(n-p)\times p} \end{bmatrix}\begin{bmatrix} Y^TY & Y^TZ \end{bmatrix}.\end{aligned}$$

The choice $Z = Y_\perp$, i.e.,

$$A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}, \tag{6.15}$$

with $Y \in \mathbf{R}_*^{n\times p}$, $Y_\perp \in \mathbf{R}_*^{n\times(n-p)}$, $Y^T Y_\perp = 0$, $Y_\perp^T Y_\perp = I_{n-p}$ allows to simplify the horizontal space to

$$\mathcal{H}_A = \begin{bmatrix} Y(Y^TY)^{-1}\mathbf{S}^p(Y^TY) + Y_\perp \mathbf{R}^{(n-p)\times p} & 0_{n\times(n-p)} \end{bmatrix}.$$

Still using $Z = Y_\perp$, the inner product of two tangent vectors $\overline{\xi}_1, \overline{\xi}_2 \in \mathcal{H}_A$ can also be written succinctly. Suppose $\overline{\xi}_1 = \begin{bmatrix} Y(Y^TY)^{-1}H_1(Y^TY) + Y_\perp K_1 & 0 \end{bmatrix}$ and analogously for $\overline{\xi}_2$, then

$$\overline{g}_A(\overline{\xi}_1, \overline{\xi}_2) = \operatorname{tr}((Y^TY)^{-1}(H_1(Y^TY)H_2 + K_1^T K_2)). \tag{6.16}$$

6.2.9 The orthogonal projections

Now that we have defined the metric, we can specify the projection onto the horizontal space orthogonal w.r.t. this metric. Since $T_A\mathbf{GL}^n \simeq \mathbf{R}^{n\times n} = \mathcal{V}_A \oplus \mathcal{H}_A$ for all $A \in \mathbf{GL}^n$, we can define for every $A \in \mathbf{GL}^n$ the following orthogonal projections:

$$\mathrm{P}^{\mathrm{h}} : \mathbf{R}^{n\times n} \to \mathcal{H}_A, \tag{6.17}$$

with $\mathrm{P}^{\mathrm{h}}(V) = 0$ for all $V \in \mathcal{V}_A$ and $\mathrm{P}^{\mathrm{h}}(H) = H$ for all $H \in \mathcal{H}_A$; and

$$\mathrm{P}^{\mathrm{v}} : \mathbf{R}^{n\times n} \to \mathcal{V}_A, \tag{6.18}$$

with $\mathrm{P}^{\mathrm{v}}(H) = 0$ for all $H \in \mathcal{H}_A$ and $\mathrm{P}^{\mathrm{v}}(V) = V$ for all $V \in \mathcal{V}_A$. So we can decompose every $Z \in \mathbf{R}^{n\times n}$ into a horizontal term $\mathrm{P}^{\mathrm{h}}(Z)$ and a vertical term $\mathrm{P}^{\mathrm{v}}(Z)$ for which $\overline{g}_A(\mathrm{P}^{\mathrm{h}}(Z), \mathrm{P}^{\mathrm{v}}(Z)) = 0$ and $Z = \mathrm{P}^{\mathrm{h}}(Z) + \mathrm{P}^{\mathrm{v}}(Z)$.

These projections can be computed for all A and Z as oblique projections onto $\mathcal{H}_A$ along $\mathcal{V}_A$, and vice versa. However this is very inefficient since it involves the whole $n \times n$ matrices A and Z and it requires a basis for the range and null spaces of $\mathcal{V}_A$ and $\mathcal{H}_A$. We will derive an efficient expression for these projections by exploiting the equivalence along the fiber $[A]$.

First, we need a technical Lemma regarding the so-called anti-stable and symmetric generalized Lyapunov equation. This equation was featured more prominently in Chapter 4, but we reformulate the results about its solvability here for convenience.

Lemma 6.6. *Let $S_1, S_2 \in \mathbf{S}_{++}^n$ be given and define the generalized Lyapunov operator*

$$\mathcal{L}_{S_1,S_2} : \mathbf{S}^n \to \mathbf{S}^n, \quad X \mapsto S_1 X S_2 + S_2 X S_1. \tag{6.19}$$

Then operator (6.19) is linear and bijective. Furthermore, equation $\mathcal{L}_{S_1,S_2}(X) = B$, can be solved in $O(n^3)$ time and memory.

Proof. See Theorem 4.1 and Section 4.2.2. □

The choice $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix} \in \mathbf{GL}^n$ allows for a direct computation of the projections.

Lemma 6.7. *Let $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix} \in \mathbf{GL}^n$ be as in (6.15). The horizontal projection of the tangent vector $\xi_A = \begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix}$, with the same partitioning as A, is given by*

$$\mathrm{P}^{\mathrm{h}}(\xi_A) = \begin{bmatrix} Y(Y^TY)^{-1}H(Y^TY) + \mathrm{P}_Y^\perp \xi_1 & 0_{n\times(n-p)} \end{bmatrix}$$

with $H \in \mathbf{S}^p$ the solution of the Lyapunov equation

$$(Y^TY)^{-1}H(Y^TY) + (Y^TY)H(Y^TY)^{-1} = (Y^TY)^{-1}(Y^T\xi_1) + (\xi_1^TY)(Y^TY)^{-1}. \quad (6.20)$$

The vertical projection is given by

$$\mathrm{P}^{\mathrm{v}}(\xi_A) = \begin{bmatrix} \mathrm{P}_Y\xi_1 - Y(Y^TY)^{-1}H(Y^TY) & \xi_2 \end{bmatrix}.$$

Proof. We will give a constructive proof. The vertical and horizontal space are given by (6.11) and (6.12) respectively:

$$\mathcal{V}_A = \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} \Omega & M \\ 0 & N \end{bmatrix}, \quad \mathcal{H}_A = \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} (Y^TY)^{-1}H(Y^TY) & 0 \\ K(Y^TY) & 0 \end{bmatrix}.$$

with Ω, M, N, H and K coefficient matrices of suitable form. The horizontal projection of ξ_A must satisfy $\mathrm{P}^{\mathrm{h}}(\xi_A) = \xi_A - v$, $v \in \mathcal{V}_A$. Writing the tangent vector as

$$\xi_A = \begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix} = \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix},$$

this condition reduces to

$$\begin{bmatrix} (Y^TY)^{-1}H(Y^TY) & 0 \\ K(Y^TY) & 0 \end{bmatrix} = \begin{bmatrix} X_{11} - \Omega & X_{12} - M \\ X_{21} & X_{22} - N \end{bmatrix},$$

which immediately gives that $K = X_{21}(Y^TY)^{-1}$, $M = X_{12}$ and $N = X_{22}$. The condition $(Y^TY)^{-1}H(Y^TY) = X_{11} - \Omega$ with $H = H^T$ and $\Omega = -\Omega^T$ is solved by adding it to its transpose. This gives $(Y^TY)^{-1}H(Y^TY) + (Y^TY)H(Y^TY)^{-1} = X_{11} + X_{11}^T$. Since $Y^TY \succ 0$, Lemma 6.6 guarantees a unique and symmetric H. This way, we have determined all the coefficients H, K of the projected matrix $\mathrm{P}^{\mathrm{h}}(\xi_A)$. Since $\xi_1 = YX_{11} + Y_\perp X_{21}$, it is a straightforward exercise to express the obtained matrix for $\mathrm{P}^{\mathrm{h}}(\xi_A)$ into the form of the Lemma. □

Using the previous Lemma, we can compute the horizontal projection of a tangent vector ξ_A at an arbitrary $A = \begin{bmatrix} Y & Z \end{bmatrix}$ by transporting ξ_A along the fiber $[A]$ to a point $\widetilde{A} = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$. Since $\widetilde{A} = AL$ for some $L \in \mathrm{Stab}_e$, it suffices to compute the projection of the tangent vector $\xi_{\widetilde{A}} = \xi_A L$ and transporting $\mathrm{P}^{\mathrm{h}}(\xi_{\widetilde{A}})$ back to A by $\mathrm{P}^{\mathrm{h}}(\xi_{\widetilde{A}})L^{-1}$.

Proposition 6.8. *Let $A = \begin{bmatrix} Y & Z \end{bmatrix} \in \mathbf{GL}^n$ be as in (6.5). The horizontal projection of the tangent vector $\xi_A = \begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix}$, with the same partitioning as A, is given by*

$$\mathrm{P}^{\mathrm{h}}(\xi_A) = \begin{bmatrix} \xi_1^h & \xi_1^h (Y^T Y)^{-1} Y^T Z \end{bmatrix}$$

with $\xi_1^h = Y(Y^TY)^{-1}H(Y^TY) + \mathrm{P}_Y^\perp \xi_1$ and H the solution of the Lyapunov equation (6.20).

Proof. Suppose we fix $\widetilde{A} = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$ for some orthonormal $Y_\perp$, then we have $\widetilde{A}L = A$ with

$$L = \begin{bmatrix} I_k & (Y^TY)^{-1}Y^TZ \\ 0 & Y_\perp^T Z \end{bmatrix}. \tag{6.21}$$

The transported tangent vector $\xi_{\widetilde{A}} = \xi_A L^{-1}$ has the same first p columns as ξ_A. By Lemma 6.7, we have as horizontal projection $\mathrm{P}^{\mathrm{h}}(\xi_{\widetilde{A}}) = \begin{bmatrix} \xi_1^h & 0 \end{bmatrix}$. Transporting this back to A gives the desired projection $\mathrm{P}^{\mathrm{h}}(\xi_A) = \mathrm{P}^{\mathrm{h}}(\xi_{\widetilde{A}})L$ □

Notice that we can compute the projections based only on Y, the first p columns of A, and we do not need to construct or use $Y_\perp$.

6.2.10 The Levi–Civita connection

We will use the Levi–Civita connection on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$, denoted by ∇. Since this connection can be related to the one on $(\mathbf{GL}^n, \overline{g})$, denoted by $\overline{\nabla}$, we will first derive this connection for two arbitrary vector fields.

Proposition 6.9. *Let η, ν be two vector fields on $\mathbf{GL}^n$, then the Levi–Civita connection of $(\mathbf{GL}^n, \overline{g})$ in $A \in \mathbf{GL}^n$ satisfies*

$$(\overline{\nabla}_\nu \eta)(A) = \mathrm{D}\eta(A)[\nu] + \frac{1}{2}\{[A^{-T}\eta^T, \nu A^{-1}]A + [A^{-T}\nu^T, \eta A^{-1}]A - \eta A^{-1}\nu - \nu A^{-1}\eta\}. \tag{6.22}$$

Proof. Let η, ν, λ be vector fields on $\mathbf{GL}^n$. Notice that since $\mathbf{GL}^n$ is a vector space, one has $[\nu, \eta] = \mathrm{D}\eta[\nu] - \mathrm{D}\nu[\eta]$, and likewise for all permutations between η, ν and λ. Furthermore, we have the identity

$$\mathrm{D}\overline{g}_A(\eta,\lambda)[\nu] = \operatorname{tr}(A^{-T}\eta^T \mathrm{D}\lambda[\nu]\, A^{-1}) + \operatorname{tr}(A^{-T}\lambda^T \mathrm{D}\eta[\nu]\, A^{-1}) \\ - \operatorname{tr}(A^{-T}\eta^T \lambda A^{-1}\nu A^{-1}) - \operatorname{tr}(A^{-T}\lambda^T \eta A^{-1}\nu A^{-1})$$

and again for all permutations. Substituting these identities in Koszul's formula (Lee, 1997, (5.1)),

$$2\overline{g}_A(\overline{\nabla}_\nu \eta, \lambda) = \mathrm{D}\overline{g}_A(\eta,\lambda)[\nu] + \mathrm{D}\overline{g}_A(\nu,\lambda)[\eta] - \mathrm{D}\overline{g}_A(\eta,\nu)[\lambda] \\ + \overline{g}_A(\lambda,[\nu,\eta]) + \overline{g}_A(\eta,[\lambda,\nu]) - \overline{g}_A(\nu,[\eta,\lambda]),$$

we obtain

$$2\operatorname{tr}(A^{-T}\lambda^T \nabla_\nu \eta A^{-1}) = 2\operatorname{tr}(A^{-T}\lambda^T \mathrm{D}\eta[\nu]\, A^{-1} - A^{-T}\eta^T \lambda A^{-1}\nu A^{-1} \\ - A^{-T}\lambda^T \eta A^{-1}\nu A^{-1} - A^{-T}\nu^T \lambda A^{-1}\eta A^{-1} \\ - A^{-T}\lambda^T \nu A^{-1}\eta A^{-1} + A^{-T}\eta^T \nu A^{-1}\lambda A^{-1} + A^{-T}\nu^T \eta A^{-1}\lambda A^{-1}).$$

This identity holds for all λ, hence we recover (6.22). □

Remark 6.10. Despite the widespread use of $\mathbf{GL}^n$, to the best of our knowledge, the expression of the connection for arbitrary vector fields on $(\mathbf{GL}^n, \overline{g})$ is new. For left- or right-invariant vector fields however, the connection simplifies considerably since it is again left- or right-invariant. This has been observed by many authors; see, e.g., Cheeger & Ebin (1975, Prop. 3.18) and Mahony (1994, Ex. 5.8.3).

Since $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ is a Riemannian quotient manifold, the connection on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ can be expressed in terms of horizontal lifts. Recall that the unique horizontal lift of a vector field η is denoted by $\overline{\eta}$.

Proposition 6.11. *Let ∇ and $\overline{\nabla}$ denote the connections on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ and $(\mathbf{GL}^n, \overline{g})$ respectively. Then for all vector fields ν, η on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$, the horizontal lift of $\nabla_\nu \eta$ satisfies*

$$\overline{\nabla_\nu \eta} = \overline{\nabla}_{\overline{\nu}}\overline{\eta} - \frac{1}{2}\mathrm{P}^{\mathrm{v}}[\overline{\nu}, \overline{\eta}] = \mathrm{P}^{\mathrm{h}}(\overline{\nabla}_{\overline{\nu}}\overline{\eta}),$$

where P^{v} and P^{h} denote the orthogonal projections (6.18)–(6.17).

Proof. The first identity follows from (3.25) in Cheeger & Ebin (1975), while the second is stated in O'Neill (1983, Lemma 7.45). □

6.3 Geodesics

According to Prop. 6.5, the quotient map $\pi : \mathbf{GL}^n \to \mathbf{GL}^n/\mathrm{Stab}_e$ is a Riemannian submersion. This means that we can identify all the geodesics on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ as the geodesics on $\mathbf{GL}^n$ for the right-invariant metric $\overline{g}$ that stays horizontal; see, e.g., O'Neill (1983, Cor. 7.46) or Gallot *et al.* (2004, Prop. 2.109). In this section, we will therefore first derive the geodesics of $(\mathbf{GL}^n, \overline{g})$, restrict them to the horizontal space $\mathcal{H}_A$ and finally formulate and solve them in a closed form.

6.3.1 Geodesics of $(\mathbf{GL}^n, \overline{g})$

Despite the widespread occurrence of $\mathbf{GL}^n$ as a Lie group, to the best of our knowledge, geodesics for the left- or right-invariant metric are not much studied in the literature. In this section, we therefore treat their derivation with some more detail.

To begin, we derive an initial value problem (IVP) for the geodesics of $(\mathbf{GL}^n, \overline{g})$.

Proposition 6.12. *Let $A_0 \in \mathbf{GL}^n$ and $U_0 \in \mathbf{R}^{n\times n}$ be given. Then the geodesic in $(\mathbf{GL}^n, \overline{g})$ through A_0 along $U_0 A_0$ is the solution $A(t)$ of the IVP*

$$\dot{A}(t) = U(t)A(t), \qquad A(0) = A_0, \tag{6.23}$$

$$\dot{U}(t) = U(t)U(t)^T - U(t)^T U(t), \qquad U(0) = U_0, \tag{6.24}$$

with $U(t) \in \mathbf{R}^{n\times n}$. Furthermore, $A(t)$ is defined for all $t \in \mathbf{R}$.

Proof. We apply the Euler–Lagrange formalism to the length functional (or strictly speaking, the energy Lagrangian). Since we can (formally) write $A(t)$ as the solution of the initial value problem $\dot{A}(t) = U(t)A(t)$ with $A(0) = A_0$ for some $U(t) \in \mathbf{R}^{n\times n}$, this functional is given by

$$S(U) := \int_0^1 \overline{g}_{A(t)}(\dot{A}(t), \dot{A}(t))\,\mathrm{d}t = \int_0^1 \overline{g}_I(U(t), U(t))\,\mathrm{d}t.$$

By virtue of 2.28, the extremals of S will be exactly the geodesics $A(t)$. A calculus of variations then gives that $U(t)$ has to satisfy (6.24) in order for $S(U)$ to be stationary.
Since $\mathrm{d}(\mathrm{tr}(UU^T))/\,\mathrm{d}t = 2\,\mathrm{tr}(\dot{U}U) = 0$, matrix $U(t)$ has constant Frobenius norm. The differential equations are thus Lipschitz on $\mathbf{R}$ and the solution of the IVP exists and is unique on the whole real line for all initial conditions (Stoer & Bulirsch, 1992, Th. 7.1.1.). □

Alternative proof. By Definition 2.26, curve $A(t)$ is a geodesic if and only if $(\overline{\nabla}_{\dot{A}}\dot{A})(A(t)) = 0$ for all t. Since $\mathrm{D}\dot{A}(A(t))[\dot{A}] = \mathrm{d}^2A/\,\mathrm{d}t^2$, formula (6.22) for the connection becomes

$$(\overline{\nabla}_{\dot{A}}\dot{A})(A(t)) = \mathrm{d}^2A/\,\mathrm{d}t^2 + \left[A^{-T}\dot{A}^T,\, \dot{A}A^{-1}\right]A - \dot{A}A^{-1}\dot{A}.$$

Taking the derivative of (6.23) and using (6.24), we obtain

$$(\overline{\nabla}_{\dot{A}}\dot{A})(A(t)) = \left[U,\, U^T\right]A + UUA + \left[U^T,\, U\right]A - UUA = 0. \qquad \square$$

Remark 6.13. In the same way geodesics for the left-invariant metric are proved in Lee *et al.* (2007). Another proof based on a more general framework can be found in Miller *et al.* (2003).

We derive the closed-form solution of the IVP in Prop 6.12. First observe that IVP (6.24) is actually a so-called *Lax pair* with solution

$$U(t) := Q(t)U_0Q(t)^T, \quad \text{with } Q(t) := \exp(t(U_0 - U_0^T)) \in \mathbf{O}^n,$$

where exp denotes the matrix exponential (A.2). This is easily verified using the relation $\dot{U}(t) = Q(U_0 - U_0^T)U_0Q^T + QU_0(U_0^T - U_0)Q^T$. Hence, $A(t)$ is the solution of the IVP

$$\dot{A}(t) = Q(t)U_0Q(t)^TA(t), \quad A(0) := A_0 \in \mathbf{GL}^n.$$

Take $B(t) := Q(t)^TA(t)$, then

$$\dot{B}(t) = (U_0^T - U_0)Q(t)^TA(t) + U_0Q(t)^TA(t) = U_0^TB(t).$$

Together with $B(0) := A_0$, this gives $B(t) = \exp(tU_0^T)A_0$, and so we have basically proved the following result.

Proposition 6.14. *The geodesics of* $(\mathbf{GL}^n, \overline{g})$ *are given by*

$$A(t) = \exp(t(U_0 - U_0^T))\exp(tU_0^T)A_0,$$

for all $A_0 \in \mathbf{GL}^n$ *and* $U_0 \in \mathbf{R}^{n\times n}$. *They are complete.*

Proof. The fact that $A(t)$ is a geodesic follows by the construction above. Since the image of the matrix exponential is always a full-rank matrix (Boothby, 1986, Th. IV.6.2), matrix $A(t)$ is well-defined and will be in $\mathbf{GL}^n$ for all t. $\square$

6.3.2 Lack of one-parameter subgroups and right-invariance

Geodesics of compact Lie groups can typically be represented by the usual matrix exponential which gives them two interesting properties: they are right- (or left)-invariant and they form a one-parameter subgroup. We show that the geodesics of $(\mathbf{GL}^n, \overline{g})$ do not share these properties.

Let $\gamma_I(t)$ be a geodesic of Prop. 6.14 with foot $\gamma_I(0) = I$. Then for any $A \in \mathbf{GL}^n$, the curve $\gamma_A(t) := \gamma_I(t)A$ is also a geodesic. Furthermore, their velocity vector fields are related by right-translation $R_A : \mathbf{GL}^n \to \mathbf{GL}^n$, $X \mapsto XA$ since

$$dR_A(\dot{\gamma}_I(t)) = \dot{\gamma}_I(t)A = \dot{\gamma}_A(t),$$

with $dR_A := R_A$ the differential of R_A. Such vector fields are called *right-related*. They are however not always *right-invariant*, or in other words, their flow does not need to be same after right translation. Take again $\gamma_I(t)$ at $t = 0$. Now we right-translate $\gamma_I(t)$ to itself, i.e., $R_A := R_{\gamma_I(T)}$ for some $t = T$. Right-invariance would mean equality for each T in

$$dR_A(\dot{\gamma}_I(0)) = \dot{\gamma}_I(0)\gamma_I(T) = U_0\gamma_I(T) \stackrel{?}{=} \dot{\gamma}_I(T) = U(T)\gamma_I(T).$$

We have equality when $U(t) = U_0$ stays constant, e.g., when U_0 is a normal matrix. In this case the geodesic is simply a matrix exponential $t \mapsto \exp(tU_0)A_0$, which is obviously right-invariant.

There is another property which depends on the normality of U_0. Let $n > 1$ and let $\gamma_I(t)$ again be a geodesic with $\gamma_I(0) = I$. Since there are initial conditions U_0 for which

$$\gamma_I(t + s) \neq \gamma_I(t)\gamma_I(s) \neq \gamma_I(s)\gamma_I(t),$$

the geodesics do not form a one-parameter subgroup in general. However, when U_0 is a normal matrix, the matrix exponentials in Prop. 6.14 commute and the geodesics can be written as

$$t \mapsto \exp(tU_0)A_0. \tag{6.25}$$

This is e.g. the case for (skew-)symmetric and orthogonal matrices U_0. In fact, one can show, together with the formula for the Levi–Civita connection, that the normality condition $[U_0, U_0^T] = 0$ captures all the cases for which the geodesics are one-parameter subgroups (Cheeger & Ebin, 1975, Prop. 3.18). Since there is a one-to-one correspondence between one-parameters subgroups in $\mathbf{GL}^n$ and exp (Boothby, 1986, Cor. IV.6.3), the curves (6.25) will be geodesics if and only if U_0 is normal. We will later see in Prop. 6.17 that the case for general U_0 is necessary to have meaningful horizontal geodesics on $\mathbf{GL}^n/\mathrm{Stab}_e$.

Remark 6.15. There are other affine connections for which the curves (6.25) describe all the geodesics, most notably the Cartan connections (Postnikov, 2001, §6.4). However, these connections do not share some important properties of the Levi–Civita connection like geodesics that are length-minimizing. For these reasons, we prefer to work with the Levi–Civita connection.

Remark 6.16. Geodesics on Lie groups for the right-invariant metric are widespread in Lagrangian and symplectic dynamics; see, e.g., Marsden & Tudor

(1999). In this field, the geodesics are usually derived based on the Euler–Arnold formalism: the Euler–Poincaré equations lead to an IVP for the intrinsic velocity $U(t)$ as the Lax pair (6.24); the IVP for the geodesic follows directly by the definition of $U(t) = \dot{A}(t)A(t)^{-1}$. Since these derivations require some more involved differential geometry, we prefer our more constructive approach.

6.3.3 Horizontal geodesics of $(\mathrm{GL}^n, \overline{g})$

By restricting a geodesic on $(\mathbf{GL}^n, \overline{g})$ to stay horizontal we obtain a representative of a geodesic on the abstract manifold $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$. At first sight, these geodesics on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ appear to be easy to find, since the geodesics of $(\mathbf{GL}^n, \overline{g})$ are right-related and available in closed form. This is however not as useful as it seems, since the horizontal space does not share the same right relation and the matrix exponentials involve large $n \times n$ matrices.

We will therefore derive, in this and the next two sections, an alternative IVP for the horizontal geodesics that allows us to solve the first p columns of a geodesic only. Observe that these first p columns, called Y, are sufficient to determine the geodesic as $x = YY^T$.

Proposition 6.17. *Let $A_0 \in \mathbf{GL}^n$, $H_0 \in \mathbf{S}^p$ and $K_0 \in \mathbf{R}^{(n-p)\times p}$ be given and let $A(t)$ be the solution of the initial value problem*

$$\dot{A}(t) = A(t)^{-T} \begin{bmatrix} H_0 & 0 \\ K_0 & 0 \end{bmatrix} A(t)^T A(t), \quad A(0) = A_0. \tag{6.26}$$

Then the complete geodesic on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ through $\pi(A_0)$ along $\xi_{\pi(A_0)}$ with horizontal lift $\overline{\xi}_{A_0} := \dot{A}(0)$ is given by $\pi(A(t))$ for all $t \in \mathbf{R}$.

Proof. First observe that since $\dot{A}(t) \in \mathcal{H}_{A(t)}$, the curve $A(t)$ stays horizontal in $(\mathbf{GL}^n, \overline{g})$. By Gallot *et al.* (2004, Prop. 2.109), $\pi(A(t))$ will be a complete geodesic on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ if $A(t)$ is a complete geodesic in $(\mathbf{GL}^n, \overline{g})$. Identifying $A(t)$ in (6.26) as the curve $A(t)$ of (6.23) we see that $U(t)$ has to satisfy

$$U(t) = A(t)^{-T} \begin{bmatrix} H_0 & 0 \\ K_0 & 0 \end{bmatrix} A(t)^T$$

in order that $A(t)$ satisfies the ODE of Prop. 6.12. Taking the derivative,

$$\dot{U}(t) = -A(t)^{-T}\dot{A}(t)^T A(t)^{-T} \begin{bmatrix} H_0 & 0 \\ K_0 & 0 \end{bmatrix} A(t)^T + A(t)^{-T} \begin{bmatrix} H_0 & 0 \\ K_0 & 0 \end{bmatrix} \dot{A}(t)^T,$$

we see that $\dot{U}(t)$ also satisfies (6.24). Thus, $A(t)$ is a geodesic in $(\mathbf{GL}^n, \overline{g})$. □

The following corollary is a simple consequence of the partitioning in (6.5).

Corollary 6.18. *Let* $A(t) := \begin{bmatrix} Y(t) & Z(t) \end{bmatrix}$ *be a geodesic as defined in Prop. 6.17 and be partitioned like eq. (6.5). Then the matrices* $Y(t)$ *and* $Z(t)$ *are solutions of the IVP*

$$\dot{Y} = P_Z^{\perp} Y (Y^T P_Z^{\perp} Y)^{-1} H_0 (Y^T Y) + P_Y^{\perp} Z (Z^T P_Y^{\perp} Z)^{-1} K_0 (Y^T Y),$$

$$\dot{Z} = P_Z^{\perp} Y (Y^T P_Z^{\perp} Y)^{-1} H_0 (Y^T Z) + P_Y^{\perp} Z (Z^T P_Y^{\perp} Z)^{-1} K_0 (Y^T Z),$$

with $Y(0) = Y_0$ *and* $Z(0) = Z_0$.

6.3.4 Moving the geodesics along the fiber

Although a representative of a geodesic on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ is given by Prop. 6.17 and the closed-form solution as Prop. 6.14 this formulation is not very satisfactory since it involves $n \times n$ matrices. We therefore derive another and more efficient formulation by exploiting the equivalence along $\pi(A(t))$.

Take the usual partitioning $A(t) = \begin{bmatrix} Y(t) & Z(t) \end{bmatrix}$. Since in the end we are only interested in Y, we are free to pick another representative of $\pi(A)$ such that matrix Z is of a more suitable form. We choose Z to be orthogonal to Y. This can always be done, since A is of full rank. The aim of this section is to transform the IVP of Cor. 6.18 into a more suited "triangular" $Z(t)$ along the equivalence class $[A(t)]$. This will introduce a new variable, called $W(t)$.

At $t = 0$, we can simply take an $A(0) := \begin{bmatrix} Y_0 & Z_0 \end{bmatrix}$ such that $Y_0 \perp Z_0$. For $t > 0$ however, $Z(t)$ will not stay orthogonal to $Y(t)$. We will therefore introduce a new variable

$$W := P_Y^{\perp} Z = Z - Y(Y^T Y)^{-1} Y^T Z \tag{6.27}$$

and derive equations of motion in terms of only Y and W. Since $W \perp Y$, it will turn out that the projectors in Cor. 6.18 disappear and that the new ODE is of more suited form.

Let us first introduce the operator $E : (\mathbf{S}^p, \mathbf{R}) \to \mathbf{S}^p$ defined as

$$E(S,t) = Q \operatorname{diag}(d_i)\, Q^T, \quad \text{with} \quad d_i = \begin{cases} t, & \lambda_i = 0, \\ (e^{t\lambda_i} - 1)/\lambda_i, & \lambda_i \neq 0, \end{cases}$$

where $S = Q \operatorname{diag}(\lambda_i)\, Q^T$ is an eigenvalue decomposition. Observe that $\lambda_i = 0$ is a removable singularity and $\mathrm{d}E(S,t)/\,\mathrm{d}t = e^{tS}$. The following identities can be proved directly from Cor. 6.18.

Lemma 6.19. *Suppose* $Y_0 \perp Z_0$, *then* $Y(t)$ *and* $Z(t)$ *of Cor. 6.18 satisfy*

$$Y(t)^T Y(t) = e^{tH_0} Y_0^T Y_0\, e^{tH_0}, \tag{6.28}$$

$$Y(t)^T Z(t) = e^{tH_0} Y_0^T Y_0\, E(H_0, t) K_0^T. \tag{6.29}$$

Proof. First observe that from Cor. 6.18 we get that $Y^T\dot{Y} = H_0 Y^T Y$, $Y^T\dot{Z} = H_0 Y^T Z$ and $\dot{Y}^T Z = Y^T Y K_0^T$. Next, we have

$$\frac{\mathrm{d}(Y^T Y)}{\mathrm{d}t} = \dot{Y}^T Y + Y^T \dot{Y} = (Y^T Y)H_0 + H_0(Y^T Y),$$

with $Y^T(0)Y(0) = Y_0^T Y_0$. So eq. (6.28) is indeed the solution of this initial value problem. In addition, we get

$$\frac{\mathrm{d}(Y^T Z)}{\mathrm{d}t} = \dot{Y}^T Z + Y^T \dot{Z} = e^{tH_0} Y_0^T Y_0\, e^{tH_0} K_0^T + H_0(Y^T Z),$$

with $Y(0)^T Z(0) = 0$. After substituting $X := e^{-tH_0} Y^T Z$ we get $\dot{X} = Y_0^T Y_0 e^{tH_0} K_0^T$ with $X(0) = 0$. Its solution is $X(t) = Y_0^T Y_0 E(H_0, t) K_0^T$ and we recover (6.29) as solution. □

The equation of motion for W can be found by taking the derivative of eq. (6.27). This gives

$$\dot{W} = \dot{Z} - \dot{Y}(Y^T Y)^{-1} Y^T Z - Y \frac{\mathrm{d}}{\mathrm{d}t}((Y^T Y)^{-1} Y^T Z).$$

According to Cor. 6.18, we have that $\dot{Y}(Y^T Y)^{-1}(Y^T Z) = \dot{Z}$. Together with Lemma 6.19, we can simplify the expression for $\dot{W}$ to

$$\begin{aligned} \dot{W} &= -Y \frac{\mathrm{d}}{\mathrm{d}t}(e^{-tH_0} E(H_0, t) K_0^T) \qquad (6.30)\\ &= Y(H_0 e^{-tH_0} E(H_0, t) - I_p) K_0^T = -Y e^{-tH_0} K_0^T. \end{aligned}$$

We have in addition the identity

$$Z = W + Y e^{-tH_0} E(H_0, t) K_0^T. \qquad (6.31)$$

Next, we rewrite $\dot{Y}$ in terms of W instead of Z. For this we need the following technical result.

Lemma 6.20. *Suppose $Y_0 \perp Z_0$, then $Y(t)$ and $Z(t)$ of Cor. 6.18 satisfy*

$$\begin{aligned} &P_Z^{\perp} Y (Y^T P_Z^{\perp} Y)^{-1} H_0 + P_Y^{\perp} Z (Z^T P_Y^{\perp} Z)^{-1} K_0 \\ &\qquad = Y(Y^T Y)^{-1} H_0 + W(W^T W)^{-1} K_0 e^{-tH_0} \qquad (6.32) \end{aligned}$$

with $W := P_Y^{\perp} Z$.

Proof. Matrix $A := \begin{bmatrix} Y & Z \end{bmatrix}$ will be full rank for any Y and Z of Cor. 6.18, so it suffices to verify that

$$A^T \left(P_Z^\perp Y (Y^T P_Z^\perp Y)^{-1} H_0 + P_Y^\perp Z (Z^T P_Y^\perp Z)^{-1} K_0 \right) \\ \stackrel{?}{=} A^T \left(Y (Y^T Y)^{-1} H_0 + W (W^T W)^{-1} K_0 e^{-tH_0} \right). \quad (6.33)$$

Working out A^T in partitioned form and using that $Y \perp W$, this gives

$$Y^T P_Z^\perp Y (Y^T P_Z^\perp Y)^{-1} H_0 \stackrel{!}{=} Y^T Y (Y^T Y)^{-1} H_0,$$

$$Z^T P_Y^\perp Z (Z^T P_Y^\perp Z)^{-1} K_0 \stackrel{?}{=} Z^T Y (Y^T Y)^{-1} H_0 + Z^T W (W^T W)^{-1} K_0 e^{-tH_0}.$$

The first equality is trivially satisfied. Plugging in Z as given by (6.31), the second expression becomes

$$K_0 \stackrel{?}{=} K_0 E(H_0, t) e^{-tH_0} H_0 + K_0 e^{-tH_0} = K_0 (E(H_0, t) e^{-tH_0} H_0 + e^{-tH_0}).$$

It is straightforward to check that $E(H_0, t) e^{-tH_0} H_0 + e^{-tH_0} \stackrel{!}{=} I_p$. □

Now we can rewrite $\dot{Y}$ directly as

$$\dot{Y} = Y (Y^T Y)^{-1} H_0 Y^T Y + W (W^T W)^{-1} K_0 e^{-tH_0} Y^T Y. \quad (6.34)$$

Summarizing all the transformations, we have almost proved the following proposition.

Proposition 6.21. *Let $A(t) = \begin{bmatrix} Y(t) & Z(t) \end{bmatrix}$ be defined as in Prop. 6.17 with the partitioning (6.5). In addition, assume that $Y(0) \perp Z(0)$, then $Y(t)$ is the solution of the initial value problem*

$$\dot{Y}(t) = \left(Y(t)\, e^{-tH_0} (Y_0^T Y_0)^{-1} H_0 + W(t) (Z_0^T Z_0)^{-1} K_0 \right) Y_0^T Y_0\, e^{tH_0}, \quad (6.35)$$

$$\dot{W}(t) = -Y(t)\, e^{-tH_0} K_0^T, \quad (6.36)$$

with $Y(0) = Y_0$, $W(0) = Z_0$ and $W(t) \in \mathbf{R}^{n \times (n-p)}$. Furthermore, $Z(t)$ is given by

$$Z(t) = W(t) + Y(t)\, e^{-tH_0} E(H_0, t) K_0^T. \quad (6.37)$$

Proof. Equations (6.36) and (6.37) follow from the construction above. Observe that from (6.30) it follows that $\mathrm{d}(W^T W)/\,\mathrm{d}t = 0$ and so $W^T(t) W(t) = Z_0^T Z_0$. Plugging this and (6.28) into (6.34), we obtain (6.35). □

6.3.5 Closed-form solution

Now we are ready to solve for the geodesics in closed form. First, we make the assumption that Z_0 is the normalized orthogonal component of Y_0, so $Z_0 := Y_{\perp 0}$ and $Z_0^T Z_0 = I_{n-p}$. This can always be done by transporting the foot of the geodesic along the equivalence class $[A]$. Next, we introduce the new variable

$$\widetilde{Y}(t) := Y(t) e^{-tH_0} (Y_0^T Y_0)^{-1/2}.$$

With $(Y_0^T Y_0)^{1/2}$ we denote the unique principal matrix square root of $Y_0^T Y_0$. Since $Y_0^T Y_0 \succ 0$, this square root is symmetric positive definite and it satisfies $(Y_0^T Y_0)^{1/2}(Y_0^T Y_0)^{1/2} = Y_0^T Y_0$. It can easily be computed by means of an eigenvalue decomposition, see Higham (2008). Taking the derivative of $\widetilde{Y}(t)$, we obtain an IVP equivalent to the one in Prop. 6.21. Let Ω_0 denote the following skew-symmetric matrix:

$$\Omega_0 := (Y_0^T Y_0)^{-1/2} H_0 (Y_0^T Y_0)^{1/2} - (Y_0^T Y_0)^{1/2} H_0 (Y_0^T Y_0)^{-1/2} \in \mathrm{skew}(p).$$

The new IVP is homogeneous and linear with constant coefficients:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\widetilde{Y} & W\end{bmatrix} = \begin{bmatrix}\widetilde{Y} & W\end{bmatrix}\Sigma_0, \qquad \begin{aligned}\widetilde{Y}(0) &= Y_0 (Y_0^T Y_0)^{-1/2},\\ W(0) &= Y_{\perp 0}.\end{aligned} \tag{6.38}$$

with

$$\Sigma_0 := \begin{bmatrix}\Omega_0 & -(Y_0^T Y_0)^{1/2} K_0^T \\ K_0 (Y_0^T Y_0)^{1/2} & 0\end{bmatrix} \in \mathrm{skew}(n).$$

Its solution is

$$\begin{bmatrix}\widetilde{Y}(t) & W(t)\end{bmatrix} = \begin{bmatrix}Y_0 (Y_0^T Y_0)^{-1/2} & Y_{\perp 0}\end{bmatrix} \exp(t\Sigma_0).$$

Although this closed-form solution is straightforward to compute, it is rather expensive since it involves the matrix exponential of an $n \times n$ matrix. Furthermore, we would like to avoid having to use matrices $Y_{\perp 0}$ and K_0 explicitly.

Thanks to the choice $Z_0 := Y_{\perp 0}$, the horizontal lift of the initial velocity $\xi_{\pi(A_0)}$ of a geodesic can be written like in Section 6.2.8:

$$\overline{\xi}_{A_0} = \begin{bmatrix}T & 0\end{bmatrix}, \quad T = Y_0 (Y_0^T Y_0)^{-1} H_0 (Y_0^T Y_0) + Y_p \in \mathbf{R}^{n \times p},\ Y_p \perp Y_0. \tag{6.39}$$

We can identify matrix T as $\dot{Y}(0)$ in Prop. 6.21. This results in the identity $Y_p = Y_{\perp 0} K_0 Y_0^T Y_0$. Substituting $\widetilde{W}(t) := W(t) K_0 (Y_0^T Y_0)^{1/2}$ in (6.38) we get an IVP without $Y_{\perp 0}$ and K_0:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\widetilde{Y} & \widetilde{W}\end{bmatrix} = \begin{bmatrix}\widetilde{Y} & \widetilde{W}\end{bmatrix}\begin{bmatrix}\Omega_0 & -S_0 \\ I_p & 0\end{bmatrix}, \qquad \begin{aligned}\widetilde{Y}(0) &= Y_0 (Y_0^T Y_0)^{-1/2},\\ \widetilde{W}(0) &= Y_p (Y_0^T Y_0)^{-1/2}.\end{aligned} \tag{6.40}$$

where $S_0 := (Y_0^T Y_0)^{-1/2} Y_p^T Y_p (Y_0^T Y_0)^{-1/2} \in \mathbf{S}^p$. Now we can summarize all the transformations in the following proposition. For ease of exposition, we formulate it stand-alone.

Proposition 6.22. *Let $Y_0 \in \mathbf{R}_*^{n\times p}$, $H_0 \in \mathbf{S}^p$ and $Y_p \in \mathbf{R}^{n\times p}$ with $Y_p \perp Y_0$ be given and define*

$$A_0 := \begin{bmatrix} Y_0 & Y_{\perp 0} \end{bmatrix}, \quad \overline{\xi}_{A_0} := \begin{bmatrix} Y_0 (Y_0^T Y_0)^{-1} H_0 (Y_0^T Y_0) + Y_p & 0_{n\times(n-p)} \end{bmatrix},$$

$$\Omega_0 := (Y_0^T Y_0)^{-1/2} H_0 (Y_0^T Y_0)^{1/2} - (Y_0^T Y_0)^{1/2} H_0 (Y_0^T Y_0)^{-1/2},$$

$$S_0 := (Y_0^T Y_0)^{-1/2} Y_p^T Y_p (Y_0^T Y_0)^{-1/2}.$$

Then the geodesic on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ through $\pi(A_0)$ along $\xi_{\pi(A_0)}$ with horizontal lift $\overline{\xi}_{A_0}$ is the curve $t \mapsto \pi\left(\begin{bmatrix} Y(t) & Y_\perp(t) \end{bmatrix}\right)$ with

$$Y(t) := \left(Y_0 (Y_0^T Y_0)^{-1/2} X_{11}(t) + Y_p (Y_0^T Y_0)^{-1/2} X_{21}(t) \right) (Y_0^T Y_0)^{1/2} \exp(tH_0)$$

and

$$\begin{bmatrix} X_{11}(t) & X_{12}(t) \\ X_{21}(t) & X_{22}(t) \end{bmatrix} := \exp\left(t \begin{bmatrix} \Omega_0 & -S_0 \\ I_p & 0 \end{bmatrix} \right).$$

Proof. The solution of (6.40) is

$$\begin{bmatrix} \widetilde{Y}(t) & \widetilde{W}(t) \end{bmatrix} = \begin{bmatrix} Y_0 (Y_0^T Y_0)^{-1/2} & Y_p (Y_0^T Y_0)^{-1/2} \end{bmatrix} \exp\left(t \begin{bmatrix} \Omega_0 & -S_0 \\ I_p & 0 \end{bmatrix} \right).$$

Undoing the transformation of $\widetilde{Y}$ gives $Y(t) = \widetilde{Y}(t)(Y_0^T Y_0)^{1/2} \exp(tH_0)$. By the construction above, matrix $Y(t)$ represents the first p columns of a representative of a geodesic on $(\mathbf{GL}^n/\mathrm{Stab}_e, \overline{g})$ and this is sufficient to define the whole the geodesic. □

Since we have a closed-from expression for the geodesics in Prop. 6.22, we do not need to integrate any ODEs. Furthermore, this can be done cheaply when $p \ll n$, which is for low-rank applications the most prominent case.

Corollary 6.23. *The geodesics on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ can be evaluated in closed-form for any t in $O(p^3) + O(np^2)$ work.*

6.3.6 Metric space

By the Hopf–Rinow theorem, $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ is a complete metric space since the geodesics can be extended indefinitely. This means that given two s.p.s.d. matrices in

$\mathbf{S}_+^{n,p}$, we can always construct a minimal geodesic that connects these two matrices. The length of this geodesic defines the distance function on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$ and, as we will see in the next section, on $\mathbf{S}_+^{n,p}$ also.

A practical use of this distance requires an efficient algorithm to construct these connecting geodesics, preferably in closed form. We have not found such a closed-form solution in the general case. As alternative, one can numerically solve a relatively simple boundary value problem since the ODE can be integrated efficiently as in Prop. 6.22. Since this is beyond the scope of the thesis, we do not explore this further and let it be as a possibility for future research.

6.4 Isometric embedding in $\mathrm{R}^{n\times n}$

In this section, we will give an interpretation of the homogeneous space structure with the right-invariant metric. In Helmke & Moore (1994) it has been shown that $\mathbf{S}_+^{n,p}$ is also an embedded sub-manifold of $\mathbf{R}^{n\times n}$. Since the structure of a sub-manifold is easier to understand than that of a quotient manifold, we will see how the two relate. Specifically, we will construct an isometry between the two manifolds.

Since we are dealing with three manifolds now, we will fix some notation in order to avoid confusion. As depicted in the table below, we use $\bar{\cdot}$ for objects of $\mathbf{GL}^n$ and $\widetilde{\cdot}$ for $\mathbf{S}_+^{n,p}$, except for $A \in \mathbf{GL}^n$ which simply denotes a matrix.

Manifold	$\mathbf{GL}^n/\mathrm{Stab}_e$	$\mathbf{GL}^n$	$\mathbf{S}_+^{n,p}$
Elements	$\pi(A)$	A	$\widetilde{S}$
Vector fields	ξ	$\overline{\xi}$	$\widetilde{\xi}$
Metric	g	$\overline{g}$	$\widetilde{g}$
Connection	∇	$\overline{\nabla}$	$\widetilde{\nabla}$

6.4.1 Related elements

In Prop. 6.2 we have constructed a diffeomorphism θ_e between $\mathbf{GL}^n/\mathrm{Stab}_e$ and $\mathbf{S}_+^{n,p}$: there is a one-to-one correspondence between $\pi(A) \in \mathbf{GL}^n/\mathrm{Stab}_e$ and fixed rank matrix $\theta_e(\pi(A)) \in \mathbf{S}_+^{n,p}$. The meaning of this mapping is not difficult to understand. Suppose we select $A = \begin{bmatrix} Y & Z \end{bmatrix}$ as a representative of $\pi(A)$. Now $\theta_e(\pi(A)) = \theta_e(A)$ gives a fixed rank matrix on $\mathbf{S}_+^{n,p}$ by selecting the first p columns of A, i.e. Y, and forming the matrix $\widetilde{S} = YY^T$, an s.p.s.d. matrix of rank p. The equivalence along the fibers is also clear: matrix $A_* = \begin{bmatrix} YQ & Z_* \end{bmatrix}$, $Q \in \mathbf{O}^p$ belongs

to the same fiber $\pi^{-1}(\pi(A))$ as A and $\theta_e(A_*)$ gives indeed the same s.p.s.d. matrix $\widetilde{S} = YY^T$.

This equivalence along the fiber was a useful property to exploit when deriving the expressions of the geodesics in Section 6.3. In what follows we will continue using it.

6.4.2 Related tangent vectors

One can relate a vector field on $\mathbf{GL}^n/\mathrm{Stab}_e$ to a vector field on $\mathbf{S}_+^{n,p}$ by means of the differential of θ_e. Suppose we take $A = \begin{bmatrix} Y & Z \end{bmatrix}$ as representative of $\pi^{-1}(\pi(A))$ with corresponding s.p.s.d. matrix $\widetilde{S} = YY^T \in \mathbf{S}_+^{n,p}$. Then the differential is

$$F_{\pi(A)} := \mathrm{D}\theta_e(\pi(A)) : \ T_{\pi(A)}\mathbf{GL}^n/\mathrm{Stab}_e \to T_{\widetilde{S}}\mathbf{S}_+^{n,p},$$

where we introduced the notation $F_{\pi(A)}$. A tangent vector $\xi_{\pi(A)}$ is so-called F-related to a tangent vector $\widetilde{\xi}_{\widetilde{S}}$ if it satisfies

$$F_{\pi(A)}(\xi_{\pi(A)}) = \widetilde{\xi}_{\widetilde{S}}.$$

This relation is unique because θ_e is a diffeomorphism (see, e.g., Boothby (1986, Th. IV.2.7.)). In other words, $F_{\pi(A)}$ is an isomorphism between vector spaces. By slight abuse of notation, this relation carries over directly to vector fields, i.e., we say that the vector fields ξ and $\widetilde{\xi}$ are F-related if $F(\xi) = \widetilde{\xi}$.

Before we proceed, we recall the following characterization of Section 3.3.1. The tangent space of $\mathbf{S}_+^{n,p}$ at $S = YY^T$ is given by

$$T_{YY^T}\mathbf{S}_+^{n,p} = \left\{ \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} H & K^T \\ K & 0 \end{bmatrix} \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} \;\middle|\; H \in \mathbf{S}^p,\ K \in \mathbf{R}^{(n-p)\times p} \right\} \quad (6.41)$$

with $Y_\perp \in \mathbf{R}_*^{n\times(n-p)}$ an orthonormal basis for the orthogonal complement of Y in $\mathbf{GL}^n$.

Since $\mathbf{GL}^n/\mathrm{Stab}_e$ contains abstract elements, we worked with the horizontal lift of a tangent vector. We can relate these horizontal lifts in a similar way to the tangent vectors of $\mathbf{S}_+^{n,p}$. If we take the derivative of the identity $\theta_e = \theta_e \circ \pi$ and use property (6.13) for horizontal lifts, we get for all $\overline{\xi}_A \in \mathcal{H}_A$ that

$$\mathrm{D}\theta_e(A)[\overline{\xi}_A] = \mathrm{D}\theta_e(\pi(A))[\mathrm{D}\pi(A)[\overline{\xi}_A]] = \mathrm{D}\theta_e(\pi(A))[\xi_{\pi(A)}]. \quad (6.42)$$

We see that the differential of θ_e can be computed by taking a classical derivative of the matrix valued function θ_e where the directions lie in the horizontal space. From Prop. 6.5 we know that

$$\overline{F}_A := \mathrm{D}\pi(A)|_{\mathcal{H}_A} : \ \mathcal{H}_A \to T_{\pi(A)}\mathbf{GL}^n/\mathrm{Stab}_e$$

is a bijection. So, by restricting the domain of $\mathrm{D}\theta_e$ to the horizontal space, mapping

$$\widetilde{F}_A := \mathrm{D}\theta_e(A)|_{\mathcal{H}_A} : \ \mathcal{H}_A \to T_{\widetilde{S}}\mathbf{S}^{n,p}_+. \tag{6.43}$$

is again an isomorphism. The corresponding diagram is

$$\begin{array}{ccc} \mathcal{H}_A & & \\ \Big\downarrow{\scriptstyle \overline{F}_A} & \searrow^{\widetilde{F}_A} & \\ T_{\pi(A)}\mathbf{GL}^n/\mathrm{Stab}_e & \xrightarrow[F_{\pi(A)}]{} & T_{\widetilde{S}}\mathbf{S}^{n,p}_+ \end{array} \tag{6.44}$$

In the following proposition, we show how these relations can be computed explicitly in case of $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$.

Proposition 6.24. *Let $\overline{\xi}$ be the horizontal lift of a vector field ξ on $\mathbf{GL}^n/\mathrm{Stab}_e$. Then $\overline{\xi}$ is $\widetilde{F}$-related to a unique vector field $\widetilde{\xi}$ on $\mathbf{S}^{n,p}_+$, i.e., $\widetilde{F}(\overline{\xi}) = \widetilde{\xi}$, where $\widetilde{F} = F \circ \overline{F}$. For $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$, this relation can be computed as*

$$\begin{aligned} \widetilde{F}_A : \begin{bmatrix} Y(Y^TY)^{-1}H(Y^TY) + Y_\perp K & 0_{n\times(n-p)} \end{bmatrix} \\ \mapsto Y\mathcal{L}(H)Y^T + Y_\perp K Y^T + Y K^T Y_\perp^T, \end{aligned} \tag{6.45}$$

and

$$\begin{aligned} \widetilde{F}_A^{-1} : YHY^T + Y_\perp K Y^T + Y K^T Y_\perp^T \\ \mapsto \begin{bmatrix} Y(Y^TY)^{-1}\mathcal{L}^{-1}(H)(Y^TY) + Y_\perp K & 0_{n\times(n-p)} \end{bmatrix}, \end{aligned} \tag{6.46}$$

with $\mathcal{L}$ the bijection

$$\mathcal{L} : \mathbf{S}^p \to \mathbf{S}^p, X \mapsto (Y^TY)^{-1}X(Y^TY) + (Y^TY)X(Y^TY)^{-1}.$$

Proof. The relation $\widetilde{F} = F \circ \overline{F}$ was already shown above. Take $A = \begin{bmatrix} Y & Z \end{bmatrix}$ and $\widetilde{S} = YY^T$. Since $\dim(\mathcal{H}_A) = \dim(T_{\widetilde{S}}\mathbf{S}^{n,p}_+)$, we have that $\widetilde{F}_A$ is a bijection for every A. Now restrict A to $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$. The horizontal lift of a tangent vector $\xi_{\pi(A)}$ will be of the form $\overline{\xi}_A = \begin{bmatrix} Y(Y^TY)^{-1}H(Y^TY) + Y_\perp K & 0_{n\times(n-p)} \end{bmatrix}$. Working out

the derivative of $\theta_e(A)$, we get

$$\begin{aligned}\mathrm{D}\theta_e(A)[\overline{\xi}_A] &= \overline{\xi}_A E A^T + A E \overline{\xi}_A^T \\ &= Y(Y^TY)^{-1}H(Y^TY)Y^T + Y_\perp K Y^T \\ &\quad + Y(Y^TY)H(Y^TY)^{-1}Y^T + Y K^T Y_\perp^T \\ &= Y\mathcal{L}(H)Y^T + Y_\perp K Y^T + Y K^T Y_\perp^T = \widetilde{\xi}_{\widetilde{S}}.\end{aligned}$$

Based on (6.41), it is clear that $\widetilde{F}_A(\overline{\xi}_1) := \mathrm{D}\theta_e(A)[\overline{\xi}_1]$ corresponds to only one tangent vector in $T_{\widetilde{S}}\mathbf{S}_+^{n,p}$. Likewise, the inverse $\widetilde{F}_A^{-1}$ is also unique: given

$$\widetilde{\xi}_{\widetilde{S}} = YHY^T + Y_\perp K Y^T + Y K^T Y_\perp^T \in T_{\widetilde{S}}\mathbf{S}_+^{n,p},$$

matrix

$$\overline{\xi}_1 = \begin{bmatrix} Y(Y^TY)^{-1}\mathcal{L}^{-1}(H)(Y^TY) + Y_\perp K & 0_{n\times(n-p)}\end{bmatrix}$$

represents the horizontal lift at $A = \begin{bmatrix} Y & Y_\perp\end{bmatrix}$. Since $Y^TY \succ 0$, we know know from Lemma 6.6 that $\mathcal{L}$ is a bijection. Hence equation $\mathcal{L}(X) = H$ has a unique and symmetric solution. □

It is illustrative to verify that mappings (6.45)–(6.46) are indeed invariant along the fiber $\pi^{-1}(\pi(A))$. Instead of using $A = \begin{bmatrix} Y & Y_\perp\end{bmatrix}$ as base point, we can also choose $B = \begin{bmatrix} YQ & Y_\perp P\end{bmatrix}$ with $Q \in \mathbf{O}^p$, $P \in \mathbf{O}^{n-p}$. Now the horizontal lift of tangent vector $\xi_{\pi(A)}$ becomes

$$\overline{\xi}_B = \overline{\xi}_A L, \quad L = \begin{bmatrix} P & 0 \\ 0 & Q\end{bmatrix}.$$

It is a straightforward exercise to show that $\widetilde{F}_A(\overline{\xi}_A) = \widetilde{F}_B(\overline{\xi}_B)$.

6.4.3 Related metrics

Next, we relate the metrics. Suppose we have two vector fields $\overline{\xi}$, $\overline{\eta}$ that are the horizontal lifts of two vector fields ξ, η on $\mathbf{GL}^n/\mathrm{Stab}_e$. In addition, let $\widetilde{\xi}$, $\widetilde{\eta}$ be the corresponding $\widetilde{F}$-related field on $\mathbf{S}_+^{n,p}$. The aim is to find an inner product $\widetilde{g}$ on $\mathbf{S}_+^{n,p}$ such that

$$\widetilde{g}(\widetilde{\xi}, \widetilde{\eta}) = g(\xi, \eta) = \overline{g}(\overline{\xi}, \overline{\eta})$$

for all related vector fields. We call the inner products g and $\widetilde{g}$ then again F-related.

Since $\mathbf{S}_+^{n,p}$ is embedded in the Euclidean space $\mathbf{R}^{n\times n} \simeq \mathbf{R}^{n^2}$, we will construct $\widetilde{g}$ as a weighted Euclidean metric:

$$\widetilde{g}_{\widetilde{S}}(\xi_1, \xi_2) := \operatorname{vec}(\xi_1)^T \mathcal{W} \operatorname{vec}(\xi_2), \quad \xi_1, \xi_2 \in T_{\widetilde{S}}\mathbf{S}_+^{n,p},$$

for some matrix $\mathcal{W} \in \mathbf{S}^{n^2}$. We have used the vectorization operator $\operatorname{vec} : \mathbf{R}^{n\times n} \to \mathbf{R}^{n^2}$, which makes a vector from a matrix by column-wise stacking. In order that $\widetilde{g}$ is an inner product, $\mathcal{W}$ should be symmetric and positive definite for all vectors in $T_{\widetilde{S}}\mathbf{S}_+^{n,p}$. However, $\mathcal{W}$ may be indefinite on $\mathbf{R}^{n^2\times n^2}$.

The following proposition gives such a matrix $\mathcal{W}$. It makes use of the Kronecker product $\otimes$ as explained in App. B.1.3.

Proposition 6.25. *Let $\widetilde{S} = YY^T \in \mathbf{S}_+^{n,p}$ and $M = Y^TY$. Let $\widetilde{g}$ be the inner product on $\mathbf{S}_+^{n,p}$ that satisfies*

$$\widetilde{g}_{\widetilde{S}}(\widetilde{\xi}_1, \widetilde{\xi}_2) := \frac{1}{2} \operatorname{vec}(\widetilde{\xi}_1)^T \mathcal{W} \operatorname{vec}(\widetilde{\xi}_2), \quad \forall \widetilde{\xi}_1, \widetilde{\xi}_2 \in T_{\widetilde{S}}\mathbf{S}_+^{n,p}, \tag{6.47}$$

with

$$\begin{aligned}\mathcal{W} = (Y \otimes Y)(M^3 \otimes M + M \otimes M^3)^{-1}(Y^T \otimes Y^T) \\ + YM^{-3}Y^T \otimes Y_\perp Y_\perp^T + Y_\perp Y_\perp^T \otimes YM^{-3}Y^T.\end{aligned} \tag{6.48}$$

Then $\widetilde{g}$ is F-related to g.

Proof. We first check that (6.47) fulfills the three axioms of an inner product and that it is well defined. Linearity is trivial. Next, observe that we can write $\mathcal{W} = \mathcal{Y}\mathcal{C}\mathcal{Y}^T$ with $\mathcal{Y} = \begin{bmatrix} Y \otimes Y & Y \otimes Y_\perp & Y_\perp \otimes Y & Y_\perp \otimes Y_\perp \end{bmatrix}$ and

$$\mathcal{C} = \begin{bmatrix} (M^3 \otimes M + M \otimes M^3)^{-1} & 0 & 0 & 0 \\ 0 & M^{-3} \otimes I_{n-p} & 0 & 0 \\ 0 & 0 & I_{n-p} \otimes M^{-3} & 0 \\ 0 & 0 & 0 & 0_{(n-p)^2\times(n-p)^2} \end{bmatrix}.$$

Since $M \succ 0$, one can verify that $\mathcal{C} \succeq 0$, from which, by Sylvester's law of inertia, $\mathcal{W} \succeq 0$. This immediately gives symmetry. Positive-definiteness follows from the fact that $\widetilde{g}_{\widetilde{S}}(X, X)$ vanishes only for matrices $X \in \mathbf{R}^{n\times n}$ of the form $Y_\perp L Y_\perp^T$ for some $L \in \mathbf{R}^{(n-p)\times(n-p)}$. Since these matrices do not belong to the tangent space, $\widetilde{g}_{\widetilde{S}}(\widetilde{\xi}, \widetilde{\xi}) = 0$ implies $\widetilde{\xi} = 0$. Furthermore, the metric is independent of the choice of Y and $Y_\perp$. Suppose we take $Z = YQ$, for some $Q \in \mathbf{O}^p$ and $Z_\perp = Y_\perp P$ for some $P \in \mathbf{O}^{n-p}$. Then it is straightforward to show that matrix

$$(Z\otimes Z)(W^3\otimes W+W\otimes W^3)^{-1}(Z^T\otimes Z^T)+ZW^{-3}Z^T\otimes Z_\perp Z_\perp^T+Z_\perp Z_\perp^T\otimes ZW^{-3}Z^T$$

with $W = Z^TZ = Q^TMQ$ is equal to $\mathcal{W}$. Hence, the inner product stays the same.

We will proof by construction that (6.47) coincides with the right-invariant metric (6.14). Let us take the usual related base points $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$ and $\widetilde{S} = YY^T$ and related tangent vectors $\widetilde{\xi}_1 = \widetilde{F}(\overline{\xi}_1)$ and $\widetilde{\xi}_2 = \widetilde{F}(\overline{\xi}_2)$. All these tangent vectors can be parameterized by matrices $H_1, H_2 \in \mathbf{S}^p$ and $K_1, K_2 \in \mathbf{R}^{(n-p)\times p}$ satisfying the relations in Prop. 6.24. In case of $\widetilde{\xi}_1$ and $\overline{\xi}_1$, we have

$$\widetilde{\xi}_1 = YH_1Y^T + Y_\perp K_1 Y^T + YK_1^TY_\perp, \quad \overline{\xi}_1 = \begin{bmatrix} YM^{-1}\mathcal{L}^{-1}(H_1)M + Y_\perp K_1 & 0 \end{bmatrix}$$

and similarly for $\widetilde{\xi}_2$ and $\overline{\xi}_2$. First we work out $\widetilde{g}_{\widetilde{S}}(\widetilde{\xi}_1, \widetilde{\xi}_2)$. Using the property of (B.6), we get that

$$\begin{aligned} \operatorname{vec}(\widetilde{\xi}_1) &= (Y \otimes Y)\operatorname{vec}(H_1) + (Y \otimes Y_\perp)\operatorname{vec}(K_1) + (Y_\perp \otimes Y)\operatorname{vec}(K_1^T) \\ &= \mathcal{Y} \begin{bmatrix} \operatorname{vec}(H_1) \\ \operatorname{vec}(K_1) \\ \operatorname{vec}(K_1^T) \\ 0_{(n-p)^2} \end{bmatrix}. \end{aligned}$$

Now we have that

$$\begin{aligned} 2\widetilde{g}_{\widetilde{S}}(\widetilde{\xi}_1, \widetilde{\xi}_2) &= \begin{bmatrix} \operatorname{vec}(H_2) \\ \operatorname{vec}(K_2) \\ \operatorname{vec}(K_2^T) \\ 0_{(n-p)^2} \end{bmatrix}^T \mathcal{Y}^T\mathcal{Y}\mathcal{C}\mathcal{Y}^T\mathcal{Y} \begin{bmatrix} \operatorname{vec}(H_1) \\ \operatorname{vec}(K_1) \\ \operatorname{vec}(K_1^T) \\ 0_{(n-p)^2} \end{bmatrix} \\ &= \operatorname{vec}(H_2)^T(M^{-1} \otimes M + M \otimes M^{-1})^{-1}\operatorname{vec}(H_1) \\ &\quad + \operatorname{vec}(K_2)^T(M^{-1} \otimes I)\operatorname{vec}(K_1) \\ &\quad + \operatorname{vec}(K_2^T)^T(I \otimes M^{-1})\operatorname{vec}(K_1^T) \\ &= \operatorname{vec}(H_2)^T\operatorname{vec}(\mathcal{L}^{-1}H_1) + \operatorname{vec}(K_2)^T\operatorname{vec}(K_1M^{-1}) \\ &\quad + \operatorname{vec}(K_2^T)^T\operatorname{vec}(M^{-1}K_1^T). \end{aligned}$$

Changing to matrices and using the definition of $\mathcal{L}$ from Prop. 6.24 gives

$$\begin{aligned} 2\widetilde{g}_{\widetilde{S}}(\widetilde{\xi}_1, \widetilde{\xi}_2) &= \operatorname{tr}(\mathcal{L}(\mathcal{L}^{-1}(H_2))\mathcal{L}^{-1}(H_1)) + \operatorname{tr}(M^{-1}K_2^TK_1) + \operatorname{tr}(K_2M^{-1}K_1^T) \\ &= 2\operatorname{tr}(M\mathcal{L}^{-1}(H_2)M^{-1}\mathcal{L}^{-1}(H_2)) + 2\operatorname{tr}(M^{-1}K_2^TK_1). \end{aligned}$$

Now applying formula (6.16) for $\overline{g}_A(\overline{\xi}_1, \overline{\xi}_2)$, we get immediately that

$$\overline{g}_A(\overline{\xi}_1, \overline{\xi}_2) = \operatorname{tr}(M\mathcal{L}^{-1}(H_2)M^{-1}\mathcal{L}^{-1}(H_1) + M^{-1}K_2^TK_1) = \widetilde{g}_{\widetilde{S}}(\widetilde{\xi}_1, \widetilde{\xi}_2).$$

Since the metrics are independent of the choice of Y and A, this concludes the proof. □

The following is immediate:

Corollary 6.26. *Mapping*

$$\theta_e : (\mathbf{GL}^n/\mathrm{Stab}_e, g) \to (\mathbf{S}_+^{n,p}, \widetilde{g})$$

is an isometry.

Observe that any change to $\mathcal{W}$ that is restricted to the normal space of $\widetilde{S}$ will have no influence on (6.47). Specifically, we can choose any $\mathcal{W} = \mathcal{Y}\mathcal{C}\mathcal{Y}^T$ with $\mathcal{Y} = \begin{bmatrix} Y \otimes Y & Y \otimes Y_\perp & Y_\perp \otimes Y & Y_\perp \otimes Y_\perp \end{bmatrix}$ and

$$\mathcal{C} = \begin{bmatrix} (M^3 \otimes M + M \otimes M^3)^{-1} & 0 & 0 & X_1 \\ 0 & M^{-3} \otimes I_{n-p} & 0 & X_2 \\ 0 & 0 & I_{n-p} \otimes M^{-3} & X_3 \\ X_1^T & X_2^T & X_3^T & X_4 \end{bmatrix},$$

with $X_1 \in \mathbf{R}^{p^2 \times p^2}$, $X_2 \in \mathbf{R}^{p(n-p)\times p(n-p)}$, $X_3 \in \mathbf{R}^{p(n-p)\times p(n-p)}$ and $X_4 \in \mathbf{S}^{(n-p)^2}$.

6.4.4 Related connection

By virtue of the naturality of the Levi–Civita connection (Lee, 1997, Prop. 5.6), isometry θ_e of Prop. 6.1 induces a $\mathrm{D}\theta_e$-related connection $\widetilde{\nabla}$ on $(\mathbf{S}_+^{n,p}, \widetilde{g})$. In other words, $\widetilde{\nabla} := \mathrm{D}\theta_e \circ \nabla$ with ∇ the connection on $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$.

We would like to express this again in terms of horizontal lifts. Recall from Prop. 6.24 that the bijection $\overline{F}_A^{-1}$ allows us to lift each vector field on $\mathbf{S}_+^{n,p}$ to a unique vector field on $\mathbf{GL}^n$ with $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$ the θ_e-related element of $\widetilde{S} = YY^T$. Furthermore, this lifted vector field can be parameterized by a classic matrix function:

$$\overline{\eta}(A) := \overline{F}_A^{-1}(\widetilde{\eta}(\widetilde{S})) = \begin{bmatrix} Z_\eta(Y) & 0_{n\times(n-p)} \end{bmatrix} \quad \text{with } Z_\eta : \mathbf{R}^{n\times p} \to \mathbf{R}^{n\times p}. \tag{6.49}$$

Now, Prop. 6.11 allows us to express the connection $\widetilde{\nabla}$ of $(\mathbf{S}_+^{n,p}, \widetilde{g})$ in terms of the horizontal lifts as

$$\widetilde{\nabla}_{\widetilde{\nu}}\widetilde{\eta} = \overline{F}(\mathrm{P}^{\mathrm{h}}(\overline{\nabla}_{\overline{\nu}}\overline{\eta})) = \overline{F}(\overline{\nabla}_{\overline{\nu}}\overline{\eta}),$$

where we used the property that P^{v} belongs to the null space of $\overline{F}$. Summarizing, this results in the following proposition.

Proposition 6.27. *Let $\widetilde{\eta}, \widetilde{\nu}$ be two vector fields on $\mathbf{S}^{n,p}_+$. Let Z_η, Z_ν be the parameterization (6.49) of the related vector fields $\overline{\eta}, \overline{\nu}$ on $\mathbf{GL}^n/\mathrm{Stab}_e$. Then the Levi–Civita connection in $\widetilde{S} = YY^T$ w.r.t. metric $\widetilde{g}$ of Prop. 6.25 is given by*

$$\widetilde{\nabla}_{\widetilde{\nu}}\widetilde{\eta}(\widetilde{S}) = (\mathrm{D}Z_\eta(Y)[Z_\nu] + W)\,Y^T + Y\,(\mathrm{D}Z_\eta(Y)[Z_\nu] + W)^T \tag{6.50}$$

where

$$W := \frac{1}{2}\{Y(Y^TY)^{-1}(Z_\eta^TZ_\nu + Z_\nu^TZ_\eta) - Z_\nu(Y^TY)^{-1}(Z_\eta^TY + Y^TZ_\eta) - Z_\eta(Y^TY)^{-1}(Z_\nu^TY + Y^TZ_\nu)\}. \tag{6.51}$$

6.4.5 Related geodesics

The geodesics on $(\mathbf{S}^{n,p}_+, \widetilde{g})$ are simply the image of F of the geodesics of $(\mathbf{GL}^n/\mathrm{Stab}_e, g)$:

Proposition 6.28. *Let $\widetilde{S}_0 = Y_0Y_0^T \in \mathbf{S}^{n,p}_+$ with $Y_0 \in \mathbf{R}^{n\times p}_*$ and define $N_0 := (Y_0^TY_0)^{1/2}$. Let*

$$Z(t) := Y_0N_0^{-1}X_{11}(t) + Y_pN_0^{-1}X_{21}(t)$$

with

$$\begin{bmatrix} X_{11}(t) & X_{12}(t) \\ X_{21}(t) & X_{22}(t) \end{bmatrix} := \exp\left(t \begin{bmatrix} N_0^{-1}H_0N_0 - N_0H_0N_0^{-1} & -N_0^{-1}Y_p^TY_pM_0^{-1/2} \\ I_p & 0_{p\times p} \end{bmatrix}\right).$$

Then the geodesic in $\widetilde{S}_0$ along

$$\widetilde{\xi}_0 = Y_0(N_0^{-1}H_0N_0 + N_0H_0N_0^{-1})Y_0^T + Y_pY_0^T + Y_0Y_p^T,$$

with $H_0 \in \mathbf{S}^p$ and $Y_0 \perp Y_p \in \mathbf{R}^{n\times p}$ is the curve

$$t \mapsto \widetilde{S}(t) := Z(t)N_0\exp(2tH_0)N_0Z(t)^T.$$

Proof. All geodesics on $(\mathbf{S}^{n,p}_+, \widetilde{g})$ will be of the form $\widetilde{S}(t) = Y(t)Y(t)^T$ with $Y(t)$ from Prop. 6.22. The initial tangent vector is identified by taking the derivative in $t = 0$. □

6.5 Special geodesics and retractions

Observe that every $Y \in \mathbf{R}^{n\times p}_*$ can be written as $Y = UC$ with orthonormal $U \in \mathbf{R}^{n\times p}_*$ and $C \in \mathbf{GL}^p$. The corresponding s.p.s.d. matrix then satisfies $\widetilde{S} = UWU^T$,

$W = CC^T \succ 0$. In this section we will see whether we can derive a geodesic as a decomposed curve also: one curve for the orthonormal part $U(t)$ and one for the small matrix $W(t) \succ 0$. The general answer will turn out to be negative although the resulting curve will be a retraction.

6.5.1 The case $K_0 = 0$.

Take $K_0 = 0$ in eq. (6.38), or equivalently $Y_p = 0$ in Prop. 6.22, then the geodesic, which we denote by $Y_K(t)$, is given by

$$Y_K(t) = Y_0(Y_0^T Y_0)^{-1/2}\exp(t\Omega_0)(Y_0^T Y_0)^{1/2}\exp(tH_0). \tag{6.52}$$

Observe that the curve does not change the column span of Y_0. This geodesic is in fact directly related to the geodesics on $(\mathbf{S}_+^{p,p}, \overline{g}) \simeq (\mathbf{GL}^p/\mathbf{O}^p, g)$. Since we can always write $Y_0 = U_0 C_0$ with orthonormal $U_0 \in \mathbf{R}_*^{n\times p}$ and $C_0 \in \mathbf{GL}^p$, we have $Y_K(t) = U_0 C(t)$ with

$$C(t) = C_0(C_0^T C_0)^{-1/2}\exp(t\Omega_0)(C_0^T C_0)^{1/2}\exp(tH_0).$$

and

$$\Omega_0 = (C_0^T C_0)^{-1/2} H_0 (C_0^T C_0)^{1/2} - (C_0^T C_0)^{1/2} H_0 (C_0^T C_0)^{-1/2}.$$

Suppose we take C_0 as a representative in $\mathbf{GL}^p/\mathbf{O}^p$ for matrix $C_0 C_0^T \in \mathbf{S}_+^{p,p}$. Looking at Prop. 6.22, we see that $\pi(C(t))$ is indeed a geodesic on $(\mathbf{GL}^p/\mathbf{O}^p, g)$.

6.5.2 The case $H_0 = 0$.

Let $L_0 := K_0(Y_0^T Y_0)^{1/2}$. If $H_0 = 0$ and $p < n$, we can write the matrix exponential in expression (6.38) as follows

$$\exp\left(t\begin{bmatrix} 0 & -L_0^T \\ L_0 & 0 \end{bmatrix}\right) = \begin{bmatrix} V & 0 & 0 \\ 0 & U_1 & U_2 \end{bmatrix} \begin{bmatrix} \cos(t\Sigma) & -\sin(t\Sigma) & 0 \\ \sin(t\Sigma) & \cos(t\Sigma) & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} V^T & 0 \\ 0 & U_1^T \\ 0 & U_2^T \end{bmatrix}$$

with

$$L_0 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T$$

a partitioned singular value decomposition. The geodesic, now denoted $Y_H(t)$, satisfies

$$Y_H(t) = (Y_0(Y_0^T Y_0)^{-1/2} V \cos(t\Sigma) + Y_0^{\perp} U_1 \sin(t\Sigma))V^T (Y_0^T Y_0)^{1/2}. \tag{6.53}$$

Observe that colspan $Y_H(t)$ is no longer constant along the geodesic.

In this case, $Y_H(t)$ is related to a geodesic on the Grassmann manifold of linear subspaces equipped with the natural metric. To see this observe that colspan $Y_H(t)$ coincides with the expression for a geodesic in the Grassmann manifold, see Absil *et al.* (2004, Th. 3.6). Contrary to the Grassmann manifold where $Y_H(t)$ belongs to the equivalence class $Y\mathbf{GL}^p$, the equivalence of $Y_H(t)$ in $\mathbf{GL}^n/\mathrm{Stab}_e$ is restricted to the orthogonal group only, i.e., $Y\mathbf{O}^p$, see also Section 6.2.3. As a consequence, the eigenvalues of the geodesic $Y_H(t)Y_H(t)^T$ remain constant. To see this, first take $Y_0 = U_0C_0$ with orthonormal $U_0 \in \mathbf{R}_*^{n\times p}$ and $C_0 \in \mathbf{GL}^p$, then curve (6.53) becomes $Y_H(t) = U(t)(C_0^TC_0)^{1/2}$ with

$$U(t) = (U_0C_0(C_0^TC_0)^{-1/2}V\cos(t\Sigma) + Y_0^\perp U_1\sin(t\Sigma))V^T.$$

Since $C_0(C_0^TC_0)^{-1/2}$ is an orthogonal matrix, one can verify that $U(t)$ stays orthonormal for all t. Hence the eigenvalues of the geodesic $Y_H(t)Y_H(t)^T = U(t)(C_0^TC_0)U(t)^T$ on $\mathbf{S}_+^{n,p}$ will remain constant.

6.5.3 A retraction

We have seen that we can classify the geodesics into two disjoint types, depending on whether they change colspan $Y(t)$ or not. These two types correspond to either H_0 or K_0 zero. In case H_0 and K_0 are both non-zero, we can construct a curve by composition of these two geodesics. Take again $Y_0 = U_0C_0$. In order that the composition of the geodesics match, we rewrite the geodesic by isolating the constant orthonormal matrix and obtain

$$Y_K(t) = \underbrace{U_0C_0(C_0^TC_0)^{-1/2}}_{=:\widetilde{U}_0,\ \widetilde{U}_0^T\widetilde{U}_0=I_p}\gamma_K(t),$$

with $\gamma_K(t) := \exp(t\Omega_0)(C_0^TC_0)^{1/2}\exp(tH_0)$. In the same way, we obtain

$$Y_H(t) = \gamma_H(t)(C_0^TC_0)^{1/2},$$

with $\gamma_H(t) := (U_0C_0(C_0^TC_0)^{-1/2}V\cos(t\Sigma) + Y_0^\perp U_1\sin(t\Sigma))V^T$. Now we have $Y_K(t) = \gamma_H(0)\gamma_K(t)$ and $Y_H(t) = \gamma_H(t)\gamma_K(0)$. The new curve for $H_0 \neq 0$ and $K_0 \neq 0$ is then

$$\gamma_{HK}(t) := \gamma_H(t)\gamma_K(t)$$

which is complete, i.e., it stays in $\mathbf{GL}^n/\mathrm{Stab}_e$ for all t.

This curve is obviously not a geodesic, but it is a first-order approximation of it in the following sense: take $Y(t)Y(t)^T$ the geodesic on $(\mathbf{S}_+^{n,p}, \widetilde{g})$ with $Y(0) = \gamma_{HK}(0)$, then

$$\mathrm{dist}_{\widetilde{g}}(\gamma_{HK}(t)\gamma_{HK}(t)^T, Y(t)Y(t)^T) = O(t^2), \quad t \to 0$$

Here $\operatorname{dist}_{\widetilde{g}}$ is the distance on the metric space $(\mathbf{S}_+^{n,p}, \widetilde{g})$ defined by the length of the minimal geodesic. In Section6.4.3 metric $\widetilde{g}$ was shown to be a weighted Euclidean metric, so we can bound this distance by the usual Frobenius norm of the embedding space. This gives the following equivalent property of quasi-geodesics

$$\|\gamma_{HK}(t)\gamma_{HK}(t)^T - Y(t)Y(t)^T\|_F = O(t^2), \quad t \to 0 \tag{6.54}$$

Curves of this type are called quasi-geodesics of order one (Nishimori & Akaho, 2005) or first-order retractions (Absil *et al.*, 2008, Def. 4.1.1). Amongst other uses, they appear in optimization on manifolds as a cheap but equally good substitute of the exponential map, see Absil *et al.* (2008) for a general overview.

To verify that $\gamma_{HK}(t)\gamma_{HK}(t)^T$ is indeed of first-order, apply the Baker–Campbell–Hausdorff formula (Hairer *et al.*, 2006, III.4.2) to split an exponential:

$$\begin{aligned}
\exp(tA)\exp(tB) &= \exp\left(t(A+B) + t^2(AB - BA) + \ldots\right) \\
&= I + \sum_{k=1}^{\infty} \frac{1}{k!} \left(t(A+B) + O(t^2)\right)^k, \quad t \to 0 \\
&= I + \sum_{k=1}^{\infty} \frac{1}{k!} \left(t^k(A+B)^k + O(t^{k+1})\right), \quad t \to 0 \\
&= \exp\left(t(A+B)\right) + O(t^2), \quad t \to 0
\end{aligned}$$

Using this expansion for the exponential of (6.38), we obtain with $L_0 := K_0(Y_0^T Y_0)^{1/2}$

$$\exp\left(t\begin{bmatrix}\Omega_0 & -L_0^T \\ L_0 & 0\end{bmatrix}\right) = \exp\left(t\begin{bmatrix}0 & -L_0^T \\ L_0 & 0\end{bmatrix}\right)\exp\left(t\begin{bmatrix}\Omega_0 & 0 \\ 0 & 0\end{bmatrix}\right) + O(t^2)$$

Now working out each exponential like in the cases above, we arrive at $Y(t) = \gamma_{HK}(t) + O(t^2), t \to 0$.

Curve $\gamma_{HK}(t)$ of (6.54) shows some resemblance to the curves in Bonnabel & Sepulchre (2009). In Section 6.6.4, we will come back to these curves and compare them more thoroughly.

6.6 Comparison with other metrics and geometries

In Section 6.2.4 we have already hinted that for $p < n$ there is no longer a natural choice for the metric as in the case $p = n$. In this chapter, we have chosen the right-invariant metric since this turned π into a Riemannian submersion, a useful

property to exploit for deriving geodesics. In the literature there exist other metrics however. Each choice has advantages and disadvantages. Below we will briefly compare them with our choice and focus most attention on the properties of the geodesics.

6.6.1 Embedded submanifold with the Euclidean metric.

One can regard $\mathbf{S}_+^{n,p}$ as a sub-manifold embedded in $\mathbf{R}^{n\times n}$ and equip it with the usual Euclidean metric, as was done in, e.g., Helmke & Moore (1994); Helmke & Shayman (1995); Orsi *et al.* (2004, 2006); Vandereycken *et al.* (2009); Vandereycken & Vandewalle (2010). The advantage is that this familiar metric allows us to interpret many geometric objects in a straightforward way. Take e.g. two vector fields ν, ξ on $\mathbf{S}_+^{n,p}$, then the Levi–Civita connection in $S \in \mathbf{S}_+^{n,p}$ satisfies

$$\nabla_{\nu_S}\xi = \mathrm{P}_S(\mathrm{D}\xi(S)[\nu_S]),$$

with P_S the usual orthogonal projection onto the tangent space $T_S\mathbf{S}_+^{n,p}$ and D a classic Euclidean directional derivative of a matrix-valued function.

Hence, it may seem natural to use this metric, but it is not the most appealing from a theoretical perspective since $\mathbf{S}_+^{n,p}$ is not a complete metric space. In Vandereycken *et al.* (2009), the authors derive the equations of motion of the geodesics and clearly show that they can not always be extended indefinitely. Still, the simplicity of many expressions makes this an interesting geometry.

We have seen in Section 6.4 that our homogeneous space geometry also coincides with an embedded sub-manifold but now with a much more involved and varying metric. It is however still a weighted Euclidean metric. In principle this allows for the same interpretation of the Levi–Civita connection involving a projection but now w.r.t. to the metric of Prop. 6.25.

6.6.2 Quotient manifold $\mathrm{R}_*^{n\times p}/\mathrm{O}^p$ with the Euclidean metric

One can factor every $S \in \mathbf{S}_+^{n,p}$ as $S = YY^T$ with orthonormal $Y \in \mathbf{R}_*^{n\times p}$. This factorization is unique up to the action of the orthogonal group, i.e., transformation $Y \mapsto YQ$ gives the same matrix S for all $Q \in \mathbf{O}^p$. This allows one to the describe $\mathbf{S}_+^{n,p}$ as the quotient manifold

$$\mathbf{S}_+^{n,p} \simeq \mathbf{R}_*^{n\times p}/\mathbf{O}^p.$$

In Journée *et al.* (2010); Meyer *et al.* (2010); Bonnabel *et al.* (2010); Sepulchre *et al.* (2010), the authors equip this quotient manifold with the Euclidean metric. The horizontal space at $Y \in \mathbf{R}_*^{n\times p}$ satisfies, see Journée *et al.* (2010, eq. (15)),

$$\mathcal{H}_Y^{\mathrm{Eucl}} = \left\{ Z \in \mathbf{R}_*^{n\times p} : Z^TY = Y^TZ \right\},$$

which, by counting dimensions, is equivalent to

$$\mathcal{H}_Y^{\mathrm{Eucl}} = \left[Y(Y^TY)^{-1}\mathbf{S}^p + Y_\perp \mathbf{R}^{(n-p)\times p} \right].$$

It is not difficult to see that this description is equivalent to equipping $\mathbf{GL}^n/\mathrm{Stab}_e$ with the Euclidean metric $\overline{g}^{\mathrm{Eucl}}$. Reiterating the steps of the derivation in Section 6.2, but now with $\overline{g}^{\mathrm{Eucl}}$, we get as horizontal space

$$\mathcal{H}_A^{\mathrm{Eucl}} = A^{-T} \begin{bmatrix} \mathbf{S}^p & 0 \\ \mathbf{R}^{(n-p)\times p} & 0 \end{bmatrix}.$$

In $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$, the horizontal lifts are of the form

$$\mathcal{H}_A^{\mathrm{Eucl}} = \left[Y(Y^TY)^{-1}\mathbf{S}^p + Y_\perp \mathbf{R}_*^{(n-p)\times p} \quad 0_{n\times(n-p)} \right].$$

Since $\pi_{\mathrm{Eucl}} : (\mathbf{GL}^n, \overline{g}_{\mathrm{Eucl}}) \to (\mathbf{GL}^n/\mathrm{Stab}_e, g_{\mathrm{Eucl}})$ is again a Riemannian submersion, we have that the geodesics are the projection of horizontal geodesics on $(\mathbf{GL}^n, \overline{g}_{\mathrm{Eucl}})$. These will be straight curves

$$t \mapsto Y_0 + t\dot{Y}_0, \quad \forall\, Y_0 \in \mathbf{R}_*^{n\times p},\ \dot{Y}_0 \in \mathcal{H}_{Y_0}^{\mathrm{Eucl}},$$

which are obviously not complete; the underlying reason being that $(\mathbf{GL}^n, \overline{g}_{\mathrm{Eucl}})$ is not complete. This is the primary reason why we disregarded the Euclidean metric in Section 6.2.5.

6.6.3 Quotient manifold $\mathbf{R}_*^{n\times p}/\mathbf{O}^p$ with a special metric

In Absil *et al.* (2009b), the authors equip $\mathbf{R}_*^{n\times p}/\mathbf{O}^p$ with the following specially chosen metric on $\mathbf{R}_*^{n\times p}$:

$$g_Y^{\mathrm{Left}}(Z_1, Z_2) = \mathrm{tr}(Z_1^T \mathrm{P}_Y^\perp(Z_2) + Z_1^T Y(Y^TY)^{-2}Y^TZ_2)$$

where $P_Y^\perp := I - \mathrm{P}_Y$ denotes the usual orthogonal projection. In this case, the horizontal space in $Y \in \mathbf{R}_*^{n\times p}$ satisfies

$$\mathcal{H}_Y^{\mathrm{Left}} = \left[Y\mathbf{S}^p + Y_\perp \mathbf{R}_*^{(n-p)\times p} \right].$$

The reason for choosing this metric, is that it allows to pick a specific affine (not the Levi–Civita) connection which results in a particularly lean expression for a Newton equation on $\mathbf{R}_*^{n\times p}/\mathbf{O}^p$. We refer to Absil *et al.* (2009b) for details. Here, we only want to point out that this metric coincides with equipping $\mathbf{GL}^n/\mathrm{Stab}_e$ with the left-invariant metric

$$\overline{g}_A^{\mathrm{Left}}(\xi_A, \eta_A) = \overline{g}_I^{\mathrm{Left}}(A^{-1}\xi_A, A^{-1}\eta_A) = \mathrm{tr}(\xi_A^T A^{-T}A^{-1}\eta_A). \tag{6.55}$$

Analogously as in the previous section, quotient manifold $(\mathbf{GL}^n/\mathrm{Stab}_e, g_{\mathrm{Left}})$ has as horizontal space

$$\mathcal{H}^{\mathrm{Left}}_A = A \begin{bmatrix} \mathbf{S}^p & 0 \\ \mathbf{R}^{(n-p)\times p} & 0 \end{bmatrix}.$$

In $A = \begin{bmatrix} Y & Y_\perp \end{bmatrix}$, this gives

$$\mathcal{H}^{\mathrm{Left}}_A = \begin{bmatrix} Y\mathbf{S}^p + Y_\perp \mathbf{R}^{(n-p)\times p} & 0_{n\times(n-p)} \end{bmatrix} = \begin{bmatrix} \mathcal{H}^{\mathrm{Left}}_Y & 0_{n\times(n-p)} \end{bmatrix}.$$

For the previous choice of A, let $\xi = \begin{bmatrix} Z_\xi & 0 \end{bmatrix} \in \mathcal{H}^{\mathrm{Left}}_A$ with $Z_\xi = YS_\xi + Y_\perp K_\xi \in \mathcal{H}^{\mathrm{Left}}_Y$ and similarly for η and Z_η. Then metric g_{Left} satisfies

$$g^{\mathrm{Left}}_A(\xi_A, \eta_A) = \mathrm{tr}\left(\begin{bmatrix} S_\xi & K_\xi^T \\ 0 & 0 \end{bmatrix} A^T A^{-T} A^{-1} A \begin{bmatrix} S_\eta & 0 \\ K_\eta & 0 \end{bmatrix} \right) = \mathrm{tr}(S_\xi S_\eta + K_\xi^T K_\eta),$$

which is obviously the same as

$$\begin{aligned} g^{\mathrm{Left}}_Y(Z_\xi, Z_\eta) = {} & \mathrm{tr}\left((S_\xi Y^T + K_\xi^T Y_\perp^T) Y (Y^T Y)^{-2} Y^T (YS_\eta + Y_\perp K_\eta) \right) \\ & + \mathrm{tr}\left((S_\xi Y^T + K_\xi^T Y_\perp^T) Y_\perp Y_\perp^T (YS_\eta + Y_\perp K_\eta) \right). \end{aligned}$$

The downside of this description is that this horizontal space does not define a connection on the principal bundle $\mathbf{GL}^n(\mathbf{GL}^n/\mathrm{Stab}_e, \mathrm{Stab}_e)$ since $\mathcal{H}^{\mathrm{Left}}_{AH} \neq \mathcal{H}^{\mathrm{Left}}_A H$ for all $H \in \mathrm{Stab}_e$. One can show that

$$\pi^{\mathrm{Left}}_a : (\mathbf{GL}^n, \overline{g}^{\mathrm{Left}}) \to (\mathbf{GL}^n/\mathrm{Stab}_e, g^{\mathrm{Left}})$$

is not a Riemannian submersion anymore, so the geodesics (for the Levi–Civita connection) are not simply the projections of those on $(\mathbf{GL}^n, \overline{g}^{\mathrm{Left}})$. We do have however $\mathcal{H}^{\mathrm{Left}}_{YQ} = \mathcal{H}^{\mathrm{Left}}_Y Q$ for all $Q \in \mathbf{O}^p$. So projection

$$\pi^{\mathrm{Left}}_b : (\mathbf{R}^{n\times p}_*, \overline{g}^{\mathrm{Left}}) \to (\mathbf{GL}^n/\mathrm{Stab}_e, g^{\mathrm{Left}})$$

is Riemannian. However, the geodesics of $(\mathbf{R}^{n\times p}_*, g^{\mathrm{Left}})$ are most likely rather difficult to find, and it is not known whether they are complete or not.

6.6.4 Quotient manifold $(\mathrm{St}^{n,p} \times \mathbf{S}^{n,p}_+)/\mathbf{O}^p$ with a polar metric

We can factor every $S \in \mathbf{S}^{n,p}_+$ as $S = UPU^T$ with orthonormal $U \in \mathbf{R}^{n\times p}_*$ and $P \in \mathbf{S}^{p,p}_+$. This factorization is unique up to the action of the orthogonal group, i.e., transformation

$$U \mapsto UQ, \quad P \mapsto Q^T P Q, \quad \text{for all } Q \in \mathbf{O}^p$$

gives the same matrix S. This allows one to the describe $\mathbf{S}_+^{n,p}$ as the quotient manifold

$$\mathbf{S}_+^{n,p} \simeq (\mathrm{St}^{n,p} \times \mathbf{S}_+^{p,p})/\mathbf{O}^p,$$

with $\mathrm{St}^{n,p}$ the Stiefel manifold of n-by-p orthonormal matrices. In Bonnabel & Sepulchre (2009), the authors equip this manifold with a so-called *polar metric* which is a linear combination of the natural metrics on $\mathrm{St}^{n,p}$ and $\mathbf{S}_+^{p,p}$. Since $\mathrm{St}^{n,p}$ and $\mathbf{S}_+^{p,p}$ both have a rich geometry, the metric on $(\mathrm{St}^{n,p} \times \mathbf{S}_+^{p,p})/\mathbf{O}^p$ inherits most of the useful invariance properties (but not all) of the reductive space $\mathbf{S}_+^{n,n}$.

A difficulty regarding this approach is that the quotient map

$$\pi^{\mathrm{Polar}} : (\mathrm{St}^{n,p} \times \mathbf{S}_+^{p,p}) \to (\mathrm{St}^{n,p} \times \mathbf{S}_+^{p,p})/\mathbf{O}^p$$

is not a Riemannian submersion. Therefore the projection of the closed-form (horizontal) geodesics of $(\mathrm{St}^{n,p} \times \mathbf{S}_+^{p,p})$ will not be geodesics of $(\mathrm{St}^{n,p} \times \mathbf{S}_+^{p,p})/\mathbf{O}^p$; they will only be quasi-geodesics. In addition, it is not obvious if this description will eventually lead to efficient geodesics. These curves do however have nice properties: they are complete, available in closed-form and it is straightforward to construct a curve connecting to s.p.s.d. matrices, see, e.g., Meyer *et al.* (2009, 2010); Bonnabel *et al.* (2010); Bonnabel & Sepulchre (2010); Sepulchre *et al.* (2010).

These quasi-geodesics of Bonnabel & Sepulchre (2009) show some resemblance to the quasi-geodesics $\gamma_{HK}(t)$ of Section 6.5.3: compare curve Bonnabel & Sepulchre (2009, eq. (5.4)) to $\gamma_{HK}(t)\gamma_{HK}(t)^T$. While the curve for the orthonormal part, i.e., $\gamma_H(t)$, is the same, we do not recover a geodesic on $\mathbf{S}_+^{p,p}$ with the natural metric, but only with the right-invariant metric. This makes it harder to construct a connecting curve between two s.p.s.d. matrices. On the other hand, since curve $\gamma_{HK}(t)$ is not a geodesic anyway, there is probably little point in trying to find such a connecting curve.

The approach of this paper is in some sense the opposite of this in Bonnabel & Sepulchre (2009). Our primary aim was finding a geometry which allowed for an easy formulation of the geodesics. The property that π is a Riemannian submersion is of central importance. The downside of this approach is that it involves some work to derive efficient closed-form expressions of the geodesics. On the other hand, the aim in Bonnabel & Sepulchre (2009) was to use a geometry which allowed to reuse much of the rich geometry of $\mathrm{St}^{n,p}$ and $\mathbf{S}_+^{p,p}$. This results in geodesics in the structure space that have more symmetry and finding the connecting geodesics can be done in closed-form. However, the projected horizontal geodesics are only quasi-geodesics in $\mathbf{S}_+^{n,p}$ since Π is not Riemannian.

6.7 Conclusions

We introduced a homogenous space geometry for $\mathbf{S}_+^{n,p}$, the symmetric positive semidefinite matrices of fixed rank. By choosing the right-invariant metric on $\mathbf{GL}^n$ we made the canonical projection onto $\mathbf{GL}^n/\mathrm{Stab}_e \simeq \mathbf{S}_+^{n,p}$ a Riemannian submersion. This had the appealing property that the complete horizontal geodesics of $\mathbf{GL}^n$ could be used as pre-image of the geodesics on $\mathbf{GL}^n/\mathrm{Stab}_e$. The derivation of an efficient closed-form expression of these geodesics opens the door to the practical application of this complete space to rank-constrained problems involving positive semidefinite matrices.

Since the quotient space $\mathbf{GL}^n/\mathrm{Stab}_e$ consists of abstract equivalence classes as elements, we embedded it isometrically in the space of real matrices. This should allow for a more concrete understanding of the vector fields, the metric and the geodesics in terms of classic matrices.

7 Conclusions

This chapter presents the main results of this thesis and gives an overview of possible future research directions. The main contributions can be summarized as follows:

- a systematic derivation of an embedded geometry of the set of symmetric and positive definite matrices of fixed rank;
- a comparison of the geodesics and retractions for this embedded geometry;
- a new Riemannian approach to compute low-rank approximations of solutions of large-scale Lyapunov equations;
- an efficient preconditioner that allows the Riemannian Trust-Region to be competitive with state-of-the-art low-rank Lyapunov solvers for the symmetric positive definite case;
- a generalization of multilevel optimization to Riemannian manifolds;
- a Local Fourier Analysis that can be used as a guideline for choosing the multigrid components when applying this multilevel Riemannian algorithm to Lyapunov equations;
- a homogeneous space geometry with complete geodesics;
- the derivation of efficient and scalable expressions for the geometric objects of this homogeneous space geometry.

7.1 Future research

We present four broad ideas for future research that can be based on this thesis.

7.1.1 New matrix problems

In this thesis we applied the Riemannian approach only to Lyapunov equations. An obvious extension of the current research would be to apply this approach to general matrix problems. This can be equations of the form

$$\sum_{i=1}^{k} A_i X B_i^T = C$$

and eigenvalues problems that satisfy

$$\sum_{i=1}^{k} A_i X B_i^T = \lambda X, \quad \text{with } \lambda \text{ an eigenvalue.}$$

Such matrix problems arise from the Galerkin discretization of PDEs with nearly separable coefficients and boundary conditions on the product of two domains. For example, the radiative transport equation is posed on the product of a three- with a two-dimensional domain (Widmer *et al.*, 2008). Other applications include the control and stabilization of linear stochastic dynamical systems (Damm, 2004) and the computation of similarity measures between graphs (Blondel *et al.*, 2004).

Much of this extension is straightforward, with the exception of finding preconditioners that yield good performance on the one hand and are compatible with the low-rank structure of the manifold on the other hand. As we have seen in the context of Lyapunov equations, an efficient preconditioner is key to the performance of the Riemannian approach.

7.1.2 Low-rank tensor formats for high-dimensional problems

Low-rank matrices are low-rank tensors in two dimensions. An obvious extension is applying Riemannian optimization for solving high-dimensional problems with low-rank tensor formats. The Riemannian optimization framework had a sound convergence theory. This is in contrast to the more ad-hoc application of low-rank tensor methods that are used to solve high-dimensional equations. While these method certainly have potential (see, e.g., Kressner & Tobler (2010); Khoromskij & Schwab (2010); Ballani & Grasedyck (2010)), they lack robustness and/or generality.

Since there exist many low-rank tensor formats, this requires that one first studies the suitability of a tensor format in light of Riemannian optimization. For tensor trains, this has recently been done already in Holtz *et al.* (2010), but this geometry is not yet used for optimization. Furthermore, there exist other formats that may be preferable over tensor trains, like the hierarchical formats of Grasedyck (2010).

7.1.3 Multilevel Riemannian optimization on other manifolds

The multilevel Riemannian optimization algorithm was presented for general manifolds but we did not give a convergence proof. A rigorous convergence proof of this method is the topic of possible future research. This requires merging the retraction-based proofs for Riemannian manifolds with the recursive proofs of multilevel optimization. At first sight, the obvious candidates are the Riemannian Trust-Region framework of Absil *et al.* (2008) and the Recursive Trust-Region proof of Gratton *et al.* (2008). Both proofs are based on optimization within the Trust-Region framework which seems to allow for more flexibility than line-search based optimization.

Another idea for research is the application of the multilevel Riemannian optimization to other manifolds. Multilevel problems on the Grassmann manifold seem ideal since this manifold has a long history with respect to Riemannian optimization. In specific, the classic second-order geometric objects like the exponential mapping (and its inverse) and parallel transport are available in an efficient and closed-form (Absil *et al.*, 2004). As large-scale application, one can take the problem of finding invariant subspaces for PDE-related problems. There already exists multilevel eigensolvers for a long time, see, e.g., Hackbusch (1979); McCormick (1981); Brandt *et al.* (1983); Mandel & McCormick (1989). However, to the best of our knowledge, there are not many *robust* multilevel algorithms for invariant subspace calculation that exploit the multilevel character of the PDE in the outer loop, as a multilevel Trust-Region scheme would do.

7.1.4 Matrix means for fixed-rank matrices

The main reason for our homogeneous space was the derivation of complete geodesics. A typical application that needs complete geodesics is the computation of the Karcher mean on a manifold. Such a mean is the midpoint of a connecting geodesic between two elements on a Riemannian manifold. This problem has a closed-form solution in the case of full-rank matrices (Moakher, 2005), but for fixed-rank the problem is still open. Although there are attempts at generalizing the mean to fixed-rank matrices, see, e.g., Bonnabel & Sepulchre (2009), they are not based on the midpoint of a geodesic on this manifold.

Applied to our homogeneous space geometry, this would require an efficient procedure for computing connecting geodesics. It is far from certain that this can be done in closed-form, which necessitates the use of a numerical method to compute the underlying boundary value problem. Since geodesics behave very much like exponentials this is a highly ill-conditioned problem. Direct multiple shooting (Bock & Plitt, 1984) implemented as a Riemannian optimization algorithm would be a possibility for an algorithm that can cope with this ill conditioning.

A

Lie groups and their actions

This appendix is about Lie groups and actions based on Lie groups. All the statements are well known and can be found in introductory books like Lee (2003). The last section deals with semialgebraic actions that originate from a Lie group. Its main theorem from Gibson (1979) is not so standard.

A.1 Lie groups and Lie algebras

Definition A.1. A (matrix) *Lie group* is a smooth manifold $\mathcal{G}$ equipped with a product rule $\mathcal{G} \times \mathcal{G} \to \mathcal{G}, (g_1, g_2) \mapsto g_1 g_2$ that satisfies the following properties.

(a) Associativity: $p(qr) = (pq)r$ for all $p, q, r \in \mathcal{G}$.

(b) Identity element: there exists an $e \in \mathcal{G}$ such that $ep = pe = p$, for all $p \in \mathcal{G}$.

(c) Inverse element: for every $p \in \mathcal{G}$, there exists an $p^{-1} \in \mathcal{G}$ such that $p^{-1}p = e$.

(d) Smoothness: the maps $(p, r) \mapsto pr$ and $p \mapsto p^{-1}$ are smooth.

Hence, a Lie group is a smooth manifold that, as a set, is a classic group by properties (a)–(b)–(c) with the additional smoothness condition (d). We will only consider matrix Lie groups where the product rule is the usual matrix multiplication.

Example A.2. We need the following finite-dimensional matrix Lie groups.

(a) The *general linear group* is defined as

$$\mathbf{GL}^n = \{X \in \mathbf{R}^{n\times n} \mid \operatorname{rank}(X) = n\}.$$

It is not compact. It has dimension n^2 and has two connected components: $\mathbf{GL}^n_+ = \{X \in \mathbf{GL}^n \mid \det(X) > 0\}$ and $\mathbf{GL}^n_- = \{X \in \mathbf{GL}^n \mid \det(X) < 0\}$.

(b) The *orthogonal group* is defined as

$$\mathbf{O}^n = \{X \in \mathbf{GL}^n \mid X^T X = I_n\}.$$

It is compact. It has dimension $n(n-1)/2$ and has two connected components: $\mathbf{O}^n_+ = \{X \in \mathbf{O}^n \mid \det(X) = 1\}$ and $\mathbf{O}^n_- = \{X \in \mathbf{O}^n \mid \det(X) = -1\}$.

Definition A.3. A *Lie subgroup* $\mathcal{H} \subset \mathcal{G}$ is a subset of $\mathcal{G}$ which is

(a) a group with respect to the product rule of $\mathcal{G}$,

(b) an embedded submanifold of $\mathcal{G}$.

Theorem A.4 (Lee (2003, Thm. 20.11))**.** *Suppose $\mathcal{G}$ is a Lie group and $\mathcal{H} \subset \mathcal{G}$ is a subgroup that is also a closed subset of $\mathcal{G}$. Then $\mathcal{H}$ is a Lie subgroup.*

The tangent space of a Lie group $\mathcal{G}$ at the identity is a so-called *Lie algebra*. It is denoted by $\mathfrak{g}$.

Definition A.5. A Lie algebra $\mathfrak{g}$ is a real vector space with a product rule $[\ ,\ ] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, called the *Lie bracket*, that satisfies the following properties.

(a) Skew-symmetric. For all $x, y \in \mathfrak{g}$: $[x, y] = -[y, x]$.

(b) Bilinear. For all $\alpha, \beta \in \mathbf{R}$ and $x, y, z \in \mathfrak{g}$: $[\alpha x + \beta y, z] = \alpha[x, z] + \beta[y, z]$.

(c) The Jacobi identity. For all $x, y, z \in \mathfrak{g}$: $[x, [y, z]] + [z, [x, y]] + [y, [z, x]] = 0$.

Example A.6. For the Lie groups from Example A.2, the Lie algebras are:

(a) The general linear group: $T_I\mathbf{GL}^n = \mathbf{R}^{n\times n}$.

(b) The orthogonal group: $T_I\mathbf{O}^n = \operatorname{skew}(n) = \{X \in \mathbf{R}^{n\times n} \mid X^T = -X\}$.

The Lie bracket is given by $[X, Y] := XY - YX$.

A.2 Actions of Lie groups and their orbits

Definition A.7. An action of a Lie group $\mathcal{G}$ on a manifold $\mathcal{M}$ is a smooth map $\delta : \mathcal{G} \times \mathcal{M} \to \mathcal{M}$ satisfying

(a) $\delta(e, x) = x$, for all $x \in \mathcal{M}$.

(b) $\delta(p, \delta(r, x)) = \delta(pr, x)$, for all $p, r \in \mathcal{G}$ and $x \in \mathcal{M}$.

An action is called *transitive*, if for any $x, y \in \mathcal{M}$, there exists a $p \in \mathcal{G}$ such that $\delta(p, x) = y$.

Definition A.8. The *orbit* of an action is the range of the map

$$\delta_x : \mathcal{G} \to \mathcal{M},\ p \mapsto \delta(p, x)$$

for some $x \in \mathcal{M}$.

The actions δ from above have to be *smooth*. Since actions on matrix Lie groups typically involve matrix operations, it is useful to know that the standard matrix multiplication, addition and subtraction are smooth operations. Furthermore, the inverse of a full-rank matrix is also smooth in the neighborhood of this matrix.

A.3 Exponential map

There are two conventions for the exponential map in the literature. In order to make a distinction, we will denote them differently as "exp" and "Exp". In some special but important cases, they are the same. Both definitions directly carry over to exponential mappings defined on smooth manifold, not only Lie groups; see, e.g., Section 2.7.2.

A.3.1 Based on left-invariant flows: exp

Definition A.9. Let $\mathcal{G}$ be a matrix Lie group and $\mathfrak{g}$ its Lie algebra. The mapping $\exp : \mathfrak{g} \to \mathcal{G}$ is defined as $\exp(a) = \sigma(1)$ where $\sigma(t) \in \mathcal{G}$ satisfies the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\sigma(t) = a\sigma(t), \quad \sigma(0) = e,\ a \in \mathfrak{g}. \tag{A.1}$$

In general, exp is only a local diffeomorphism around e. When $\mathcal{G}$ is compact and connected, exp is a global diffeomorphism. For $\mathbf{GL}^n$, the range of exp is a subset of $\mathbf{GL}^n$.

For matrix Lie groups, like the ones in Example A.2, the map exp is given by the classical matrix exponential

$$\exp(X) = \sum_{i=0}^{\infty} \frac{1}{i!} X^i, \quad \text{for all } X \in \mathfrak{g}. \tag{A.2}$$

Take $\mathcal{G} = \mathbf{GL}^n$, then the differential equation (A.1) becomes

$$\frac{\mathrm{d}}{\mathrm{d}t} X(t) = AX(t) \in \mathbf{R}^{n \times n}, \quad X(0) = I_n \in \mathbf{GL}^n,$$

Its solution is indeed given by $X(t) = \exp(tA) \in \mathbf{GL}^n$.

Unlike the scalar case $\mathcal{G} = \mathbf{R}$, exp does not commute in general, that is

$$\exp(X)\exp(Y) \neq \exp(Y)\exp(X) \neq \exp(X+Y), \quad \text{for all } X, Y \in \mathfrak{g}.$$

It only commutes when its arguments commute, i.e.,

$$\exp(X)\exp(Y) = \exp(Y)\exp(X) = \exp(X+Y) \quad \Longleftrightarrow \quad XY = YX.$$

We do have the following series expansion, called the *Baker–Campbell–Hausdorff* formula:

$$\exp(tX)\exp(tY) = \exp(tC_1 + t^2 C_2 + t^3 C_3 + t^4 C_4 + \cdots),$$

with

$$C_1 = X + Y \qquad C_2 = \frac{1}{2}[X, Y]$$

$$C_3 = \frac{1}{12}[X,[X,Y]] + \frac{1}{12}[Y,[Y,X]] \qquad C_4 = \frac{1}{24}[X,[Y,[Y,X]]].$$

From C_5 on, the terms quickly become very complicated; see, e.g., Hairer *et al.* (2006, Section III.4).

An alternative definition of this exponential mapping, is by means of one-parameter subgroups.

Definition A.10. A one-parameter subgroup of a Lie group $\mathcal{G}$ is a continuous function $\phi : \mathbf{R} \to \mathcal{G}$ which is differentiable at zero and also satisfies

$$\phi(\alpha + \beta) = \phi(\alpha)\phi(\beta) = \phi(\beta)\phi(\alpha), \quad \text{for all } \alpha, \beta \in \mathbf{R}.$$

We always have $\phi(0) = e$. To stress the initial condition $x \in \mathfrak{g}$ at zero, we use the notation ϕ_x, i.e., $\frac{\mathrm{d}}{\mathrm{d}t}\phi_x(0) = x$.

Definition A.11. Let $\phi_x(t)$ denote the unique one-parameter subgroup of a Lie group $\mathcal{G}$ with initial condition $\frac{\mathrm{d}}{\mathrm{d}t}\phi_x(0) = x$. Then the exponential map is defined as

$$\exp : \mathfrak{g} \to \mathcal{G},\ x \mapsto \phi_x(1).$$

A.3.2 Based on geodesics: Exp

The second definition of the exponential map is defined in terms of geodesics; see Section 2.7.1 for the definition of geodesics w.r.t. an affine connection ∇. It is defined on the whole tangent bundle.

Definition A.12. Let ∇ be the connection on the Lie group $\mathcal{G}$. Then the exponential map at $p \in \mathcal{G}$ is defined as

$$\mathrm{Exp}_p : T_p\mathcal{G} \to \mathcal{G},\ \xi \mapsto \gamma_p(1),$$

where γ_p is the unique geodesic for ∇ with foot $p = \gamma_p(0)$ and in the direction of $\xi = \dot{\gamma}_p(0)$.

In general, Exp_p is only a local diffeomorphism; if the geodesics are complete, it is a global one.

A.4 Semialgebraic group actions

When the action of a Lie group on $\mathbf{R}^n$ satisfies an additional property, the orbits of this action are nicely behaved. They are embedded submanifolds on $\mathbf{R}^n$. This is expressed in the following Theorem from Gibson (1979, Thm. B4).

Theorem A.13. *Let $\delta : \mathcal{G} \times \mathbf{R}^n \to \mathbf{R}^n$ be a smooth action of a Lie group $\mathcal{G}$. Suppose that δ is a semialgebraic mapping, then for each $x \in \mathbf{R}^n$, the orbit of δ through x is a smooth embedded submanifold of $\mathbf{R}^n$.*

An extended version of this theorem is repeatedly used in Helmke & Moore (1994). Since Gibson (1979) is not a wide-spread reference, we repeat the proof almost verbatim, but before proving this theorem, we need some basic facts about algebraic geometry. We refer to Bierstone & Milman (1988), Bochnak *et al.* (1998) for an introduction.

Definition A.14 (Semialgebraic set; Bierstone & Milman (1988, Def. 1.1))**.** A set $A \subset \mathbf{R}^n$ is semialgebraic if it can be obtained by finitely many applications of the operations of intersection, union and set difference starting from sets of the form $\{x \in \mathbf{R}^n \mid f(x) > 0\}$ with f a polynomial on $\mathbf{R}^n$.

Definition A.15 (Semialgebraic mapping; Bierstone & Milman (1988, Cor. 1.6))**.** A mapping $f : A \subset \mathbf{R}^n \to \mathbf{R}^p$ is semialgebraic if its graph $\{(x, f(x)) \mid x \in A\}$ is a semialgebraic set in $\mathbf{R}^n \times \mathbf{R}^p$.

Theorem A.16 (Gibson (1979, Thm. B1))**.** *Every rational mapping defined on a semialgebraic set is a semialgebraic mapping.*

Theorem A.17 (Tarski–Seidenberg; Bierstone & Milman (1988, Cor. 1.8))**.** *Let $A \subset \mathbf{R}^n$ be a semialgebraic set, and let $f : A \to \mathbf{R}^p$ be a semialgebraic mapping. Then the image $f(A)$ is a semialgebraic set in $\mathbf{R}^p$.*

Theorem A.18 (Bochnak *et al.* (1998, Prop. 2.9.10))**.** *Let $A \subset \mathbf{R}^n$ be a non-void semialgebraic set. Then A has at least one neighborhood $U \subset \mathbf{R}^n$ such that $A \cap U$ is a smooth embedded submanifold of $\mathbf{R}^n$.*

Now we can prove the main results.

Proof of Theorem A.13. Let $x \in \mathbf{R}^n$. The orbit through x,

$$\delta_x(G) := G \mapsto \delta(G, x),$$

is the image under δ of the semi-algebraic set $G \times \{x\}$, hence semialgebraic by the Tarski–Seidenberg theorem. By Theorem A.18, this subset has at least one neighborhood that is a smooth embedded submanifold of $\mathbf{R}^n$. Let some $g \in G$ be in this neighborhood. Consider then the mapping

$$\delta_g : \delta_x(G) \to \delta_x(G),\ x \mapsto \delta(g, x).$$

It is not difficult to see that it is a bijection. Hence, the previous neighborhood can be extended to the whole orbit and $\delta_x(G)$ is a smooth embedded submanifold of $\mathbf{R}^n$. □

A.5 Equivalence relations

Definition A.19. An *equivalence relation* on a space $\mathcal{M}$, denoted by $\sim$, is a relation which is

(a) reflexive: $x \sim x$ for all $x \in \mathcal{M}$,

(b) symmetric: $x \sim y$ if and only if $y \sim x$ for all $x, y \in \mathcal{M}$,

(c) transitive: if $x \sim y$ and $y \sim z$, then $x \sim z$ for all $x, y, z \in \mathcal{M}$.

The set

$$[\, x \,] := \{\, y \in \mathcal{M} \mid x \sim y \,\}$$

is called the *equivalence class* of $x \in \mathcal{M}$.

B

Elements of Linear Algebra and Calculus

B.1 Linear algebra

B.1.1 Eigenvalues and eigenvectors

The set of *eigenvalues* of a matrix $A \in \mathbf{R}^{n\times n}$, denoted $\lambda(A)$, consists of all complex numbers for which $A - \lambda I_n$ is singular. Let $\lambda_i \in \lambda(A)$, then $Av_i = \lambda_i v_i$ for the *eigenvector* $v_i \neq 0$.

The *inertia* of a symmetric matrix A, denoted inertia(A), is the triplet of nonnegative integers (m, z, p) where m, z, and p are respectively the number of negative, zero, and positive elements of $\lambda(A)$. Sylvester's law of inertia states that the inertia is invariant under congruence.

Theorem B.1 (Sylvester's law of inertia (Golub & Van Loan, 1996, Th. 8.1.17))**.** *Let $A \in \mathbf{S}^n$ and $X \in \mathbf{R}^{n\times n}$ be nonsingular. Then* inertia$(A) =$ inertia(XAX^T).

Given $A, B \in \mathbf{R}^{n\times n}$, a linear *matrix pencil* is the mapping

$$\mathbf{C} \to \mathbf{R}^{n\times n},\ \lambda \mapsto A - \lambda B.$$

It is called *regular* if there exists at least one value for λ such that $A - \lambda B \in \mathbf{GL}^n$.

The set of *generalized eigenvalues* of a matrix pencil $A - \lambda B$, denoted $\lambda(A, B)$, consists of all complex numbers for which $A - \lambda B$ is singular. Let $\lambda_i \in \lambda(A, B)$, then $Av_i = \lambda_i B v_i$ for the *generalized eigenvector* $v_i \neq 0$. If B is nonsingular, there are n generalized eigenvalues. If B is singular, the number of finite values in $\lambda(A, B)$ may be empty, finite or infinite. If $\lambda(A, B)$ contains $m < n$ finite eigenvalues, there are $n - m$ infinite eigenvalues $\lambda_i = \infty$.

Observe that $\lambda(A) = \lambda(A, I_n)$. If B is nonsingular, then $\lambda(A, B) = \lambda(B^{-1}A) = \lambda(AB^{-1}) = \lambda(B^{-1/2}AB^{-1/2})$.

B.1.2 Trace

The *trace* of a matrix $A \in \mathbf{R}^{n\times m}$, denoted by $\mathrm{tr}(A)$, is defined as the sum of the diagonal elements of A,

$$\mathrm{tr}(A) = \sum_{i=1}^{\min(m,n)} A(i,i).$$

The trace satisfies the following properties

$$\mathrm{tr}(A) = \mathrm{tr}(A^T), \quad \text{for all } A \in \mathbf{R}^{n\times n} \tag{B.1}$$

$$\mathrm{tr}(AB) = \mathrm{tr}(BA), \quad \text{for all } A \in \mathbf{R}^{n\times m} \text{ and } B \in \mathbf{R}^{m\times n} \tag{B.2}$$

$$\mathrm{tr}(AB) = 0, \quad \text{for all } A \in \mathbf{S}^n \text{ and } B \in \mathrm{skew}(n) \tag{B.3}$$

B.1.3 The Kronecker product

The (left) *Kronecker product* is denoted by $\otimes$. (The right Kronecker product is an alternative, less-used convention). Let $A \in \mathbf{R}^{m\times n}$ and $B \in \mathbf{R}^{p\times q}$, then the Kronecker product is given as the block matrix

$$A \otimes B = \begin{bmatrix} A_{1,1}B & A_{1,2}B & \cdots & A_{1,n}B \\ A_{2,1}B & A_{2,2}B & \cdots & A_{2,n}B \\ \vdots & \vdots & & \vdots \\ A_{m,1}B & A_{m,2}B & \cdots & A_{m,n}B \end{bmatrix} \in \mathbf{R}^{mp\times nq}.$$

Some properties of the Kronecker product:

$$A \otimes (B + C) = A \otimes B + A \otimes C$$

$$(A + B) \otimes C = A \otimes C + B \otimes C$$

$$(cA) \otimes B = A \otimes (cB) = c(A \otimes B)$$

$$(A \otimes B) \otimes C = A \otimes (B \otimes C)$$

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \quad \text{provided } A \text{ and } B \text{ are nonsingular}$$

$$(A \otimes B)^T = A^T \otimes B^T$$

$$\operatorname{rank}(A \otimes B) = \operatorname{rank}(A) \operatorname{rank}(B)$$

$$\operatorname{tr}(A \otimes B) = \operatorname{tr}(A) \operatorname{tr}(B)$$

for matrices A, B, C, D of suitable size and scalar c.

The Kronecker product is not commutative in general, i.e., $A \otimes B \neq B \otimes A$ for arbitrary square matrices A, B. However, the two products are related by permutation matrices, called the *perfect-shuffle matrices*

$$A \otimes B = P(B \otimes A)Q \tag{B.4}$$

In case A and B are both square and of the same size, we have $P = Q$. See Van Loan (2000) for a concrete expression of thee matrices.

B.1.4 Vectorization

Define the *vectorization* operator $\operatorname{vec}(\cdot)$ as the operation that stacks the columns of a matrix under each other, from left to right, i.e.,

$$\operatorname{vec} : \mathbf{R}^{m \times n} \to \mathbf{R}^{mn}, \begin{bmatrix} X_{1,1} & X_{1,2} & & X_{1,n} \\ \vdots & \vdots & & \vdots \\ X_{m,1} & X_{m,2} & & X_{m,n} \end{bmatrix} \mapsto \begin{bmatrix} X_{1,1} \\ \vdots \\ X_{m,1} \\ X_{1,2} \\ \vdots \\ \vdots \\ X_{m,n} \end{bmatrix} \tag{B.5}$$

The vectorization satisfies the following property. Let $A \in \mathbf{R}^{m\times n}$ and $B \in \mathbf{R}^{p\times q}$, then

$$\operatorname{vec}(AXB^T) = (B \otimes A)\operatorname{vec}(X) \tag{B.6}$$

$$\operatorname{vec}(A)^T \operatorname{vec}(B) = \operatorname{tr}(A^T B) \tag{B.7}$$

$$\operatorname{vec}(A^T) = \Pi \operatorname{vec}(A), \tag{B.8}$$

where Π is a particular perfect-shuffle matrix like in (B.4). See Lancaster & Tismenetsky (1985) and Horn & Johnson (1991) for more on these properties.

B.1.5 Control theory

Let $A \in \mathbf{R}^{n\times n}$ and $B \in \mathbf{R}^{n\times p}$. The *controllability matrix* is the $n \times np$ matrix given by

$$\begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}. \tag{B.9}$$

The pair (A, B) is called *controllable*, if the controllability matrix (B.9) has rank n.

B.2 Derivatives and differentials

We present the concept of derivatives for functions between normed vector spaces. The results are all standard. Most of this section is taken verbatim from Absil *et al.* (2008, App. A.5).

B.2.1 General concepts

Let $\mathcal{U}$ and $\mathcal{V}$ be two finite-dimensional vector spaces of $\mathbf{R}$. A mapping $F : \mathcal{U} \to \mathcal{V}$ is *Fréchet differentiable* at $x \in \mathcal{U}$ if there exists a linear operator

$$\mathrm{D}F(x) :\ \mathcal{U} \to \mathcal{V},\ h \mapsto \mathrm{D}F(x)[h],$$

called the *Fréchet differential* of F at x, such that

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

The element $\mathrm{D}F(x)[h]$ is called the *directional derivative* of F at x along h. The derivative obeys the *chain rule*:

$$\mathrm{D}(f \circ g)(x)[h] = \mathrm{D}f(g(x))[\mathrm{D}g(x)[h]], \tag{B.10}$$

where $D(f \circ g)(x)$ denotes the derivative of the composite function $f \circ g$ at x and $\mathrm{D}f(g(x))$ denotes the derivative of f at $g(x)$.

Let $f : \mathcal{U} \to \mathbf{R}$ be a smooth real-valued function. Then the *gradient* of f at $x \in \mathcal{U}$, denoted by $\operatorname{grad} f(x)$, is the unique element of $\mathcal{U}$ that satisfies

$$\langle \operatorname{grad} f(x), h \rangle_x = \mathrm{D}f(x)[h], \quad \text{for all } h \in \mathcal{U}.$$

Here $\langle \cdot, \cdot \rangle_x$ is the inner product on $\mathcal{U}$. Obviously, the gradient depends on this inner product.

The derivative of the differential of a smooth function $f : \mathcal{U} \to \mathcal{V}$ is called the *second derivative*. It is denoted by $\mathrm{D}^2 f(x)$ and can be represented as a bilinear symmetric map

$$\mathrm{D}^2 f(x) : \mathcal{U} \times \mathcal{U} \to \mathcal{U}, \quad (h_1, h_2) \mapsto \mathrm{D}^2 f(x)[h_1, h_2].$$

with the property

$$\mathrm{D}^2 f(x)[h_1, h_2] = \mathrm{D}^2 f(x)[h_2, h_1], \quad \text{for all } h_1, h_2 \in \mathcal{U}.$$

The *Hessian* of f at x, denoted by $\operatorname{Hess} f(x) : \mathcal{U} \to \mathcal{U}$, is the unique symmetric operator that satisfies

$$\langle \operatorname{Hess} f(x)[h_1], h_2 \rangle_x = \mathrm{D}^2 f(x)[h_1, h_2], \quad \text{for all } h_1, h_2 \in \mathcal{U}.$$

We have the relation

$$\operatorname{Hess} f(x)[h] = \mathrm{D}(\operatorname{grad} f)(x)[h]$$

for all $h \in \mathcal{U}$.

B.2.2 Explicit expressions for some matrix-valued functions

If F is a linear function, we have $\mathrm{D}F(x)[h] = F(h)$. In particular, the derivative of the trace function is given by

$$\mathrm{D} \operatorname{tr}(X)[H] = \operatorname{tr}(H).$$

Let $\operatorname{inv} : \mathbf{GL}^n \to \mathbf{GL}^n$, $X \mapsto X^{-1}$ denote the inverse function. Then we have that (Dehaene, 1995, Lemma 3.1)

$$\mathrm{D} \operatorname{inv}(X)[H] = -X^{-1} H X^{-1}.$$

In other words, for a smooth curve $t \mapsto X(t)$ of invertible matrices, we get

$$\frac{\mathrm{d}X^{-1}}{\mathrm{d}t} = -X^{-1} \frac{\mathrm{d}X}{\mathrm{d}t} X^{-1}. \tag{B.11}$$

The derivative of the matrix exponential exp of (A.2) satisfies

$$\frac{\mathrm{d}\exp(tX)}{\mathrm{d}t} = \exp(tX)X = X\exp(tX).$$

Bibliography

Absil, P.-A., Mahony, R. & Sepulchre, R. (2004) Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, **80**, 199–220. 171, 181

Absil, P.-A., Baker, C. & Gallivan, K. (2007) Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, **7**, 303–330. xix, 4, 12, 33, 34, 128, 140

Absil, P.-A., Mahony, R. & Sepulchre, R. (2008) *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press. 3, 11, 15, 18, 19, 21, 29, 30, 32, 35, 36, 58, 127, 129, 148, 172, 181, 193

Absil, P.-A., Trumpf, J., Mahony, R. & Andrews, B. (2009a) All roads lead to Newton: Feasible second-order methods for equality-constrained optimization. *UCL-INMA-2009.024*. Université catholique de Louvain, INMA. 88

Absil, P.-A., Ishteva, M., Lathauwer, L. D. & Huffel, S. V. (2009b) A geometric Newton method for Oja's vector field. *Neural Comput.*, **21**, 1415–1433. 6, 174

Absil, P.-A. & Malick, J. (2010) Projection-like retractions on matrix manifolds. *Technical Report*. Department of Mathematical Engineering, Université catholique de Louvain. 58, 63, 64

Alvarez, F., Bolte, J. & Munier, J. (2008) A unifying local convergence result for Newton's method in Riemannian manifolds. *Found. Comput. Math.*, **8**, 197–226. 140

Antoulas, A. C., Sorensen, D. C. & Zhou, Y. (2002) On the decay rate of Hankel singular values and related issues. *Systems & Control Letters*, **46**, 323–342. 75

Antoulas, A. C. (2005) *Approximation of Large-Scale Dynamical Systems*. Adv. Des. Control. SIAM, Philadelphia. 73

Baker, C., Absil, P.-A. & Gallivan, K. (2007). http://www.math.fsu.edu/~cbaker/GenRTR. 102

Ballani, J. & Grasedyck, L. (2010) A projection method to solve linear systems in tensor format. *Preprint 46*. DFG-Schwerpunktprogramm 1324. 180

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J. M., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. & Van der Vorst, H. (1993) *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics. 130

Bartels, R. H. & Stewart, G. W. (1972) Solution of the matrix equation $AX + XB = C$. *Communications of the ACM*, **15**, 820–826. 74

Bebendorf, M. & Rjasanow, S. (2003) Adaptive low-rank approximation of collocation matrices. *Computing*, **70**, 1–24. 139

Bellman, R. E. (1957) *Dynamic Programming*. Princeton University Press. 74

Benner, P. (2006) *Control Theory*. Handbook of Linear Algebra. Chapman & Hall/CRC. 73

Benner, P., Peter, H. & J. Saak, J. (2008) On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations. *Electron. Trans. Numer. Anal.*, **29**. 103

Benner, P. & Saak, J. (2004) Efficient numerical solution of the LQR-problem for the heat equation. *Proc. Appl. Math. Mech.*, **4**, 648–649. 103, 107

Benner, P. & Saak, J. (2010) A Galerkin-Newton-ADI method for solving large-scale algebraic Riccati equations. *SPP1253-090*. T.U. Chemnitz. 76

Benzi, M., Golub, G. H. & Liesen, J. (2005) *Acta Numerica*. Cambridge University Press, chapter Numerical Solution of Saddle Point Problems, pp. 1–137. 101

Bierstone, E. & Milman, P. D. (1988) Semianalytic and subanalytic sets. *Publications Mathèmatiques De L'ihés*, **67**, 5–42. 187, 188

BIN ZUBAIR, H., OOSTERLEE, C. W. & WIENANDS, R. (2007) Multigrid for high-dimensional elliptic partial differential equations on non-equidistant grids. *SIAM Journal on Scientific Computing*, **29**, 1613–1636. 115, 118

BLONDEL, V., GAJARDO, A., HEYMANS, M., SENELLART, P. & VAN DOOREN, P. (2004) A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Review*, **46**, 647–666. 180

BOCHNAK, J., COSTE, M. & ROY, M.-F. (1998) *Real algebraic geometry*. Springer. 187, 188

BOCK, H. G. & PLITT, K. J. (1984) A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings 9th IFAC World Congress Budapest*. Pergamon, pp. 243—247. 182

BONNABEL, S., MEYER, G. & SEPULCHRE, R. (2010) Adaptive filtering for estimation of a low-rank positive semidefinite matrix. *Proc. of the 19th International Symposium on Mathematical Theory of Networks and Systems, Budapest, 2010*. MTNS2010. 173, 176

BONNABEL, S. & SEPULCHRE, R. (2009) Geometric distance and mean for positive semi-definite matrices of fixed rank. *SIAM J. Matrix Anal. Appl.*, **31**, 1055–1070. 8, 144, 172, 176, 181

BONNABEL, S. & SEPULCHRE, R. (2010) Rank-preserving geometric means of positive semi-definite matrices. *Proc. of the 19th International Symposium on Mathematical Theory of Networks and Systems, Budapest, 2010*. MTNS2010. 8, 176

BOOTHBY, W. M. (1986) *An Introduction to Differentiable Manifolds and Riemannian Geometry*, second edn. Academic Press. 11, 30, 32, 138, 141, 143, 146, 154, 155, 163

BORZÍ, A. (2005) On the convergence of the MG/OPT method. *PAMM*, **5**, 735–736. 123

BOYD, S. & VANDENBERGHE, L. (2004) *Convex Optimization*. Cambridge University Press. 138

BOYLE, J., MIHAJLOVIC, M. D. & SCOTT, J. A. (2010) HSL_MI20: an efficient AMG preconditioner for finite element problems in 3d. *Internat. J. Numer. Methods Engrg.*, **82**, 64–98. 102, 104

BRANDT, A., MCCORMICK, S. & RUGE, J. (1983) Multigrid methods for differential eigenproblems. *SIAM Journal on Scientific Computing*, **4**, 244–260. 181

Burer, S. & Monteiro, R. D. (2003) A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Programming*, **95**, 329–357. 6

Cai, J.-F., Candès, E. J. & Shen, Z. (2010) A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, **20**, 1956–1982. 7

Candès, E. J. & Tao, T. (2009) The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory*, **56**, 2053–2080. 7, 139

Cheeger, J. & Ebin, D. G. (1975) *Comparison Theorems in Riemannian Geometry*. Amsterdam: North-Holland Publishing Co., pp. viii+174. 140, 152, 155

Chu, K. E. (1987) The solution of the matrix equations $AXB - CXD = E$ and $(YA - DZ, YC - BZ) = (E, F)$. *Linear Algebra Appl.*, **93**, 93–105. 70

Damm, T. (2004) *Rational Matrix Equations in Stochastic Control*. Lecture Notes In Control And Information Sciences. Springer. 180

Daniilidis, A., Malick, J. & Sendov, H. (2009) Locally symmetric submanifolds lift to spectral manifolds. *UAB 23/2009*. Departament de Matematiques, Universitat Autònoma de Barcelon. 41

Dedieu, J.-P., Priouret, P. & Malajovich, G. (2003) Newton's method on Riemannian manifolds: Covariant alpha theory. *IMA J. Numer. Anal.*, **23**, 395–419. 140

Dehaene, J. (1995) Continuous-time matrix algorithms, systolic algorithms and adaptive neural networks. *Ph.D. thesis*, Department of Electrical Engineering, Katholieke Universiteit Leuven. 194

Edelman, A., Arias, T. A. & Smith, S. T. (1999) The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.* 127

Elman, H., Silvester, D. & Wathen, A. (2005) *Finite Element and Fast Iterative Solvers*. Oxford University Press. 101

Fazel, M. (2002) Matrix rank minimization with applications. *Ph.D. thesis*, Elec. Eng. Dept, Stanford University. 6

Ferreira, O. P. & Svaiter, B. F. (2002) Kantorovich's theorem on Newton's method in Riemannian manifolds. *J. Complexity*, **18**. 140

Gallot, S., Hulin, D. & Lafontaine, J. (2004) *Riemannian Geometry*. Universitext, third edn. Berlin: Springer-Verlag. 140, 148, 153, 156

Gibson, C. G. (1979) *Singular Points of Smooth Mappings*. Research Notes in Mathematics. Pitman. 46, 183, 187, 188

Goldfarb, D. & Ma, S. (2010) Convergence of fixed point continuation algorithms for matrix rank minimization. *Found. Comput. Math.* 7, 58

Golub, G. H. & Van Loan, C. F. (1996) *Matrix Computations*, 3rd edn. Johns Hopkins Studies in Mathematical Sciences. 62, 74, 190

Grasedyck, L. (2004) Existence of a low rank or $\mathcal{H}$-matrix approximant to the solution of a Sylvester equation. *Numer. Linear Algebra Appl.*, **11**, 371–389. 75, 76

Grasedyck, L. (2010) Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, **31**, 2029–2054. 181

Grasedyck, L. & Hackbusch, W. (2007) A multigrid method to solve large scale Sylvester equations. *SIAM J. Matrix Anal. Appl.*, **29**, 870–894. 58, 77

Gratton, S. ., Mouffe, M., Sartenaer, A., Toint, P. L. & Tomanos, D. (2010) Numerical experience with a recursive trust-region method for multilevel nonlinear optimization. *Optimization Methods and Software*, **25**, 359–386. 123

Gratton, S., Sartenaer, A. & Toint, P. L. (2008) Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, **19**, 414–448. 112, 123, 128, 181

Gugercin, S., Sorensen, D. & Antoulas, A. (2003) A modified low-rank Smith method for large-scale Lyapunov equations. *Numer. Algorithms*, **32**, 27–55. 58, 77

Guttman, L. (1946) Enlargement methods for computing the inverse matrix. *The Annals of Mathematical Statistics*, **17**. 43

Hackbusch, W. (1979) On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method. *SIAM Journal on Numerical Analysis*, **16**. 181

Hackbusch, W. (1999) A sparse matrix arithmetic based on $\mathcal{H}$-matrices. part i: Introduction to $\mathcal{H}$-matrices. *Computing*, **62**, 89–108. 139

Hairer, E., Lubich, C. & Wanner, G. (2006) *Geometric Numerical Integration*, second edn. Springer-Verlag. 52, 172, 186

Helmke, U. & Moore, J. B. (1994) *Optimization and Dynamical Systems*. Communications and Control Engineering Series. London: Springer-Verlag London Ltd. 41, 42, 45, 46, 59, 162, 173, 187

Helmke, U. & Shayman, M. A. (1995) Critical points of matrix least squares distance functions. *Linear Algebra Appl.*, **215**, 1–19. 42, 45, 46, 81, 173

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, pp. xx+425. 160

Hodel, A. S., Tenison, B. & Poolla, K. R. (1996) Numerical solution of the Lyapunov equation by approximate power iteration. *Linear Algebra Appl.*, **236**, 205–230. 76

Hoegaerts, L., De Lathauwer, L., Goethals, I., Suykens, J. A. K., Vandewalle, J. & De Moor, B. (2007) Efficiently updating and tracking the dominant kernel principal components. *Neural Networks*, **20**. 139

Holtz, S., Rohwedder, T. & Schneider, R. (2010) On manifolds of tensors of fixed TT-rank. *Preprint 61*. DFG-SPP 1324. 181

Horn, R. A. & Johnson, C. R. (1991) *Topics in Matrix Analysis*. Cambridge University Press, Cambridge. 193

Huber, P. J. (1981) *Robust Statistics*. Wiley. 138

Jadea, A. M., Srikantha, B., Jayaramana, V. K., Kulkarni, B. D., Jogb, J. P. & Priyab, L. (2003) Feature extraction and denoising using kernel PCA. *Chemical Engineering Science*, **58**, 4441–4448. 139

Jaimoukha, I. & Kasenally, E. (1994) Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Numer. Anal.*, **31**, 227–251. 76

Jbilou, J. (2010) ADI preconditioned Krylov methods for large Lyapunov matrix equations. *Linear Algebra Appl.*, **432**. 76

Jbilou, K. & Riquet, A. J. (2006) Projection methods for large Lyapunov matrix equations. *Linear Algebra Appl.*, **415**, 344–358. 76

Journée, M., Bach, F., Absil, P.-A. & Sepulchre., R. (2010) Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, **20**, 2327–2351. 6, 8, 173

Khoromskij, B. N. & Schwab, C. (2010) Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *Technical report 2010-04*. Seminar for applied mathematics, Zürich. 180

Kobayashi, S. & Nomizu, K. (1963) *Foundations of Differential Geometry*. Interscience Publishers, a division of John Wiley & Sons, New York-London, pp. xi+329. 31, 140, 144, 147

Koch, O. & Lubich, C. (2007) Dynamical low-rank approximation. *SIAM. J. Matrix Anal.*, **29**, 434–454. 6, 96

Koecher, M. (1957) Positivitätsbereiche im R^n. *Amer. J. of Math.*, **79**, 575—596. 5

Kressner, D. & Tobler, C. (2009) Krylov subspace methods for linear systems with tensor product structure. *SIAM J. Matrix Anal. Appl.* to appear. 85

Kressner, D. & Tobler, C. (2010) Low-rank tensor Krylov subspace methods for parametrized linear systems. *Technical report 2010-16*. Seminar for applied mathematics, ETH Zürich. 180

Lancaster, P. & Tismenetsky, M. (1985) *The Theory of Matrices*, 2nd edn. Academic Press, Orlando. 193

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E. & Jordan, M. I. (2004) Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, **5**, 27–72. 138

Lee, J. M. (1997) *Riemannian Manifolds: An Introduction to Curvature*. Graduate Texts in Mathematics, vol. 176. Springer-Verlag, pp. xv+224. 26, 29, 31, 126, 152, 168

Lee, J. M. (2003) *Introduction to smooth manifolds*. Graduate Texts in Mathematics, vol. 218. New York: Springer-Verlag. 11, 12, 15, 17, 19, 23, 24, 29, 42, 45, 138, 183, 184

Lee, S., Choi, M., Kim, H. & Park, F. C. (2007) Geometric direct search algorithms for image registration. *IEEE Transactions on Image Processing*, **16**, 2215–2224. 154

Lehoucq, R. B., Sorensen, D. C. & Yang, C. (1997) *ARPACK users' guide: solution of large-scale eigenvalue problems with Implicitly Restarted Arnoldi Methods*. SIAM. 139

Lewis, A. S. & Malick, J. (2008) Alternating projections on manifolds. *Math. Oper. Res.*, **33**, 216–234. 61

Lewis, R. M. & Nash, S. G. (2005) Model problems for the multigrid optimization of systems governed by differential equations. *SIAM Journal on Scientific Computing*, **26**, 1811–1837. 112, 122

Li, C. & Wang, J. (2006) Newton's method on Riemannian manifolds: Smale's point estimate theory under the γ-condition. *IMA J. Numer. Anal.*, **26**, 228–251. 140

Li, J.-R. & White, J. (2004) Low-rank solution of Lyapunov equations. *SIAM Rev.*, **46**, 693–713. 76

Liu, Z. & Vandenberghe, L. (2009) Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*. 7

Mahony, R. E. (1994) Optimization algorithms on homogeneous spaces: with applications in linear systems theory. *Ph.D. thesis*, Department of Systems Engineering, Canberra, Australia. 152

Mandel, J. & McCormick, S. (1989) A multilevel variational method for $Au = \lambda Bu$ on composite grids. *J. Comput. Phys.*, **80**. 181

Manton, J. H. (2002) Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Process.*, **50**, 635–650. 58

Marsden, J. E. & Tudor, T. S. (1999) *Introduction to Mechanics and Symmetry*, 2nd ed. corr. 2nd printing edn. Springer. 155

Mastronardi, N., Tyrtyshnikov, E. E. & Van Dooren, P. (2010) A fast algorithm for updating and downsizing the dominant kernel principal components. *SIAM J. Matrix Anal. Appl.*, **31**, 2376–2399. 139

McCormick, S. (1981) A mesh refinement method for $Ax = \lambda Bx$. *Mathematics of Computation*, **36**, 485–498. 181

Meka, R., Jain, P. & Dhillon, I. S. (2010) Guaranteed rank minimization via singular value projection. *Proceedings of the Neural Information Processing Systems Conference (NIPS)*. NIPS. 58

Meyer, G., Journée, M., Bonnabel, S. & Sepulchre, R. (2009) From subspace learning to distance learning: a geometrical optimization approach. *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*. IEEE/SP, pp. 385–388. 176

Meyer, G., Bonnabel, S. & Sepulchre, R. (2010) Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *Submitted. Available at arXiv:1006.1288v1*. 8, 173, 176

Miller, M. I., Trouve, A. & Younes, L. (2003) The metric spaces, Euler equations, and normal geodesic image motions of computational anatomy. *Proceedings of the 2003 International Conference on Image Processing*, vol. 2. ICIP2003, pp. II – 635–638 vol.3. 154

Miller, S. A. & Malick, J. (2005) Newton methods for nonsmooth convex minimization: connections among $\mathcal{U}$-Lagrangian, Riemannian Newton and SQP methods. *Math. Programming*, **104**, 609–633. 61

Moakher, M. (2005) A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.*, **26**, 735–747. 55, 181

Moore, B. (1981) Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, **26**, 17–32. 73, 74

Nash, S. G. (2000) A multigrid approach to discretized optimization problems. *Journal of Optimization Methods and Software*, **14**, 99–116. 122

Nesterov, Y. & Todd, M. (2008) On the riemannian geometry defined by self-concordant barriers and interior-point methods. *Foundations of Computational Mathematics*, **2**, 333–361. 138

Nishimori, Y. & Akaho, S. (2005) Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing*, **67**, 106–135. 172

Nocedal, J. & Wright, S. J. (1999) *Numerical Optimization*. Springer Ser. Oper. Res. Springer-Verlag, New York. xix, 34, 60

Nomizu, K. (1954) Invariant affine connections on homogeneous spaces. *American Journal of Mathematics*, **76**, 33–65. 5, 144

Nong, R. & Sorensen, D. C. (2009) A parameter free ADI-like method for the numerical solution of large scale Lyapunov equations. *CAAM* TR09-16. Computational and Applied Mathematics, Rice University. 76, 110

O'Neill, B. (1966) The fundamental equations of a submersion. *Michigan Math. J.*, **13**, 459–469. 27, 147, 148

O'Neill, B. (1983) *Semi-Riemannian Geometry*. Pure and Applied Mathematics, vol. 103. New York: Academic Press Inc., pp. xiii+468. 140, 148, 152, 153

Orsi, R., Helmke, U. & Moore, J. B. (2004) A Newton–like method for solving rank constrained linear matrix inequalities. *43rd IEEE Conference on Decision and Control*. IEEE. 173

Orsi, R., Helmke, U. & Moore, J. B. (2006) A Newton-like method for solving rank constrained linear matrix inequalities. *Automatica*, **42**, 1875–1882. 173

Pennec, X., Fillard, P. & Ayache, N. (2006) A Riemannian framework for tensor computing. *International Journal of Computer Vision*, **66**, 41–66. 138

Penzl, T. (1997) A multi-grid method for generalized Lyapunov equations. *SFB393/97-24*. Technische Universität Chemnitz. 113

Penzl, T. (1998) Numerical solution of generalized Lyapunov equations. *Adv. Comput. Math.*, **8**, 33–48. 71, 79

Penzl, T. (1999) A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, **4**, 1401–1418. 76, 83, 95

Penzl, T. (2000) Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Systems Control Lett.*, **40**, 139–144. 75

Postnikov, M. M. (2001) *Geometry VI: Riemannian geometry*. Springer. 31, 155

Qi, C., Gallivan, K. A. & Absil, P.-A. (2010) Riemannian BFGS algorithm with applications. *Recent Advances in Optimization and its Applications in Engineering*. Springer, pp. 183–192. 127

Recht, B., Fazel, M. & Parrilo, P. (2010) Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, **52**, 471–501. 6

Rosen, I. G. & Wang, C. (1995) A multilevel technique for the approximate solution of operator Lyapunov and algebraic Riccati equations. *SIAM J. Numer. Anal.*, **32**, 514–541. 113

Rosipal, R. & Girolami, M. (2001) An expectation-maximization approach to nonlinear component analysis. *Neural Comput.*, **13**, 505–510. 139

Saad, Y. (1990) Numerical solution of large Lyapunov equations. *Signal Processing, Scattering, Operator Theory, and Numerical Methods* (M. A. Kaashoek, J. H. V. Schuppen & A. C. M. Ran eds). Birkhäuser, Boston, MA, pp. 503–511. 76

Saak, J., Mena, H. & Benner, P. (2008). http://www-user.tu-chemnitz.de/~saak/Software/mess.php. 102, 103

Scheerlinck, N., Verboven, P., Stigter, J. D., Baerdemaeker, J. D., Impe, J. F. V. & Nicolai, B. M. (2001) A variance propagation algorithm for stochastic heat and mass transfer problems in food processes. *Internat. J. Numer. Methods Engrg.*, **51**, 961–983. 73

Sepulchre, R., Absil, P.-A. & Bonnabel, S. (2010) Géométrie des matrices semi-définies positives de rang fixé: un peu de théorie, et beaucoup d'applications. *Proceedings of the Sixième Conférence Internationale Francophone d'Automatique*. CIFA2010. 173, 176

Simoncini, V. (2007) A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, **29**, 1268–1288. 76, 77, 102, 103, 130

Skovgaard, L. T. (1984) A Riemannian geometry of the multivariate normal model. *Scandinavian Journal of Statistics*, **11**, 211–223. 5

Smith, S. T. (2005) Covariance, subspace, and intrinsic Cramér-Rao bounds. *IEEE Trans. Signal Process.*, **53**, 1610–1630. 138, 144, 145

Snyders, J. & Zakai, M. (1970) On nonnegative solutions of the equation $AD + DA^T = -C$. *SIAM J. Appl. Math.*, **18**, 704–714. 71, 72

Sorensen, D. C. & Zhou, Y. (2002) Bounds on eigenvalue decay rates and sensitivity of solutions to Lyapunov equations. *Technical Report* Technical Report TR02-07. Computational and Applied Mathematics, Rice University. 75

Steihaug, T. (1983) The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, **20**, 626–637. 33

Stewart, G. W. (2001) *Matrix Algorithms. Volume II: Eigensystems*. Adv. Des. Control. SIAM, Philadelphia. 43, 79

Stoer, J. & Bulirsch, R. (1992) *Introduction to Numerical Analysis*, second edn. Springer-Verlag. 153

Stykel, T. (2002a) Analysis and numerical solution of generalized Lyapunov equations. *Ph.D. thesis*, Institut für Mathematik, Technische Universität Berlin. 72

Stykel, T. (2002b) Stability and inertia theorems for generalized Lyapunov equations. *Linear Algebra Appl.*, **355**. 72

Toint, P. L. (1981) *Sparse Matrices and Their Uses*. Academic Press, London, New York, chapter Towards an efficient sparsity exploiting Newton method for minimization, pp. 57–88. 33

Trottenberg, U., Oosterlee, C. W. & Schuller, A. (2000) *Multigrid*. Academic Press. 111, 114, 116, 117, 120

Tyrtyshnikov, E. (2000) Incomplete cross approximation in the mosaic-skeleton method. *Computing*, **64**, 367–380. 139

Van Loan, C. F. (2000) The ubiquitous Kronecker product. *J. Comput. Appl. Math.*, **123**, 85–100. 51, 192

Vandereycken, B., Absil, P.-A. & Vandewalle, S. (2009) Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank. *Proceedings of the IEEE 15th Workshop on Statistical Signal Processing*. IEEE, pp. 389–392. 8, 37, 173

Vandereycken, B., Absil, P.-A. & Vandewalle, S. (2010) A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *Technical Report* TW572. Department of Computer Science, Katholieke Universiteit Leuven. 9, 137

Vandereycken, B. & Vandewalle, S. (2009) Local fourier analysis for tensor-product multigrid. *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics 2009*. IAP, pp. 354–356. 9, 111

Vandereycken, B. & Vandewalle, S. (2010) A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, **31**, 2553–2579. 8, 37, 69, 173

Vasilyev, D. & White, J. (2005) A more reliable reduction algorithm for behavioral model extraction. *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. IEEE/ACM, pp. 813–820. 76

Wen, Z. & Goldfarb, D. (2009) A line search multigrid method for large-scale nonlinear optimization. *SIAM Journal on Optimization*, **3**, 1478–1503. 123

Widmer, G., Hiptmair, R. & Schwab, C. (2008) Sparse adaptive finite elements for radiative transfer. *Journal of Computational Physics*, **227**, 6071–6105. 180

Wienands, R. & Joppich, W. (2005) *Practical Fourier Analysis for Multigrid Methods*. Chapman and Hall/CRC Press. 117

Yang, Y. (2006) Globally convergent optimization algorithms on Riemannian manifolds: Uniform framework for unconstrained and constrained optimization. *J. Optim. Theory Appl.*, **132**, 245–265. 140

Yavneh, I. (1995) Multigrid smoothing factors of red-black Gauss–Seidel applied to a class of elliptic operators. *SIAM J. Numer. Anal.*, **32**. 119

Curriculum vitae

Personalia

Name: Bart Vandereycken
Date of birth: 25 February 1982
Nationality: Belgian

Higher education

2010 Ph.D. in Engineering, K.U.Leuven, Leuven, Belgium
Thesis: *Riemannian and multilevel optimization for rank-constrained matrix problems*

2005 Engineer in Computer Science (Burgerlijk Ingenieur in de Computerwetenschappen), K.U.Leuven, Leuven, Belgium
Thesis: *Three-dimensional simulation of electromagnetic waves using wavelets*
Graduated summa cum laude

Awards and fellowships

2008 Best poster award for [2], SIAM Conference on Optimization

2005 PhD fellowship of the Research Foundation Flanders (FWO)

2005 Jos Schepens Memorial Fund for best Master's thesis in the Computer Science department, K.U.Leuven

Publications

[1] *A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank*
B. Vandereycken, P.-A. Absil and S. Vandewalle
Submitted to IMA Journal on Numerical Analysis (also as Technical report TW572, Department of Computer Science, K.U.Leuven)

[2] *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*
B. Vandereycken and S. Vandewalle
SIAM Journal on Matrix Analysis and Applications, Volume 31, Issue 5, pp. 2553-2579 (2010)

[3] *Local Fourier analysis for tensor-product multigrid*
B. Vandereycken, S. Vandewalle
Proceedings of the International Conference on Numerical Analysis and Applied Mathematics 2009, pp. 354-356 (2009)

[4] *Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank*
B. Vandereycken, P.-A. Absil and S. Vandewalle
Proceedings of the IEEE 15th Workshop on Statistical Signal Processing, pp. 389-392 (2009)

[5] *The smoothed spectral abscissa for robust stability optimization*
J. Vanbiervliet, B. Vandereycken, W. Michiels, S. Vandewalle and M. Diehl
SIAM Journal on Optimization, Volume 20, Issue 1, pp. 156-171 (2009)

Conference presentations

1. B. Vandereycken, S. Vandewalle. *Multilevel Riemannian optimisation for low-rank solutions of Lyapunov equations*, ICCAM, 5-9 July 2010
2. B. Vandereycken, S. Vandewalle. *Local Fourier Analysis for Tensor-Product Multigrid*, ICNAAM 18-22 September, 2009
3. B. Vandereycken, P.-A. Absil, S. Vandewalle. *Optimization on the homogeneous space of fixed-rank positive semidefinite matrices*, Belgian-French-German Conference on Optimization, Leuven, Belgium, 14-18 September 2009

4. B. Vandereycken, S. Vandewalle. *Riemannian optimization algorithms for matrix completion*, SIAM Conference on Applied Linear Algebra, Monterey, CA, USA, 26-29 October 2009

5. B. Vandereycken, P.-A. Absil, S. Vandewalle. *Embedded Geometry of the Set of Symmetric Positive Semidefinite Matrices of Fixed Rank*, IEEE Workshop on Statistical Signal Processing, Cardiff, Wales, UK, 31 August-3 September 2009

6. B. Vandereycken, S. Vandewalle. *Geometric Optimization on the Manifold of Rank Constrained SPD Matrices*, SIAM conference on Optimization, Boston, MA, USA, 10-13 May 2008

7. B. Vandereycken, J. Vanbiervliet, W. Michiels, S. Vandewalle, M. Diehl. *Robust Control Design with the Smoothed Spectral Abscissa*, ICCAM, 7-11 July 2008

8. B. Vandereycken, S. Vandewalle. *Geometric optimization on the manifold of rank constrained matrices*, OPTEC SAB meeting, Leuven, 17 March 2008

9. B. Vandereycken, S. Vandewalle. *Geometric optimization on the manifold of rank constrained matrices*, ICCAM, 7-11 July 2008

10. B. Vandereycken, S. Vandewalle. *Solving large-scale Lyapunov equations with multigrid*, IMA Conference on Numerical Linear Algebra and Optimisation, Birmingham, UK, 13-15 September 2007

11. B. Vandereycken, S. Vandewalle. *Robust multigrid methods for large-scale matrix equations*, Workshop Matrix Equations, Chemnitz, 17 June 2007

12. B. Vandereycken, S. Vandewalle. *Solving large-scale Lyapunov equations with multigrid*, Benelux Meeting on Systems and Control, Lommel, Belgium, 13-15 March 2007

13. B. Vandereycken, S. Vandewalle. *Multigrid for large-scale time dependent Lyapunov equations*, Thirty-first Conference of the Dutch-Flemish Numerical Analysis Communities, Zeist, The Netherlands, October 11-13, 2006

14. B. Vandereycken, S. Vandewalle. *A multigrid method for matrix differential equations*, ECCOMAS CFD 2006, Egmond aan Zee, The Netherlands, September 5-8, 2006

15. B. Vandereycken, S. Vandewalle. *Multigrid for large scale time-dependent Sylvester equations*, 9th Copper Mountain Conference on Iterative Methods, Copper Mountain, Colorado, USA, April 2-7, 2006

Seminars

1. *Solving Lyapunov equations by geometric optimization on the manifold of low-rank matrices*, Doctoral Seminar, Katholieke Universiteit Leuven, Belgium, October 3, 2008.

2. *A multigrid method for large-scale matrix differential equations*, Doctoral Seminar, Katholieke Universiteit Leuven, Belgium, October 27, 2006.

Teaching

1. 2007, 2008, 2009, 2010: Teaching assistant for P&O (Tag game with Lego Mindstorms)

2. 2005, 2006: Teaching assistant for P&O (Wifi positioning with PDAs).