KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Riemannian Optimization for Averaging Positive Definite Matrices

Applications, Matrix Structures, and Algorithms

Ben Jeuris

Supervisor:
Prof. dr. R. Vandebril
Prof. dr. J. Nicaise

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

June 2015

Riemannian Optimization for Averaging Positive Definite Matrices

Applications, Matrix Structures, and Algorithms

Ben JEURIS

Examination committee:
Em. prof. dr. ir. P. Van Houtte, chair
Prof. dr. R. Vandebril, supervisor
Prof. dr. J. Nicaise, supervisor
Prof. dr. ir. W. Michiels
Prof. dr. ir. S. Vandewalle
Prof. dr. ir. P.-A. Absil
(Université Catholique de Louvain)
Prof. dr. B. Iannazzo
(Università degli Studi di Perugia)

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

June 2015

Preface

> *My dear, here we must run as fast as we can, just to stay in place. And if you wish to go anywhere you must run twice as fast as that.*
>
> - Lewis Carroll, *Alice in Wonderland*

This thesis is the result of the research that I performed over the past years under the guidance of Prof. R. Vandebril and Prof. J. Nicaise. It could not have been realized without the financial support of the Research Foundations – Flanders (FWO), which I gratefully acknowledge.

My research actually started slightly over a year before I officially started my PhD. During a gathering for master students choosing a thesis topic, I had a conversation with Raf Vandebril about a paper containing some "funny" algorithms. Soon after, I started my master's thesis under his supervision, with Bart Vandereycken to help guide me through the more abstract concepts of optimization on manifolds. After the completion of my master's thesis, it was clear that there were still many extensions of the research topic to explore. I gladly accepted Raf's offer to continue researching these possibilities during a PhD, in which the mathematical aspects would be monitored by Johannes Nicaise. Bart, thank you for your advice during, but also after my master thesis. Raf and Johannes, thank you for giving me the opportunity to pursue a doctoral degree and for your support throughout these years.

I would like to thank all the members of the examination committee for agreeing to read my thesis and providing me with useful feedback. I am grateful to Stefan Vandewalle and Pierre-Antoine Absil for their evaluation of my PhD during these four years and for sharing their expertise on different aspects of the thesis. I also thank Bruno Iannazzo, not only for the pleasant collaboration, but also for allowing me to visit the university in Perugia several times. The trips with you, Ivan, and/or Filippo were always very enjoyable. Furthermore,

thanks to Wim Michiels, for adding your expertise to the jury and for your stimulating questions. Finally, thanks go to Paul Van Houtte for chairing the defense.

During these fours years, I have had the pleasure of talking with many other researchers, on my own thesis topic and many related aspects. Some of these discussions resulted in a collaboration and, after a while, in published papers. In addition to Raf, Bruno, and Bart, I would like thank my coauthors Dario Bini, Yves Moreau, and Pooya Zakeri for sharing their expertise.

Of course, being a PhD student is not all about doing research. During my first year, I learned that an important part of the job description also consists of visiting the coffee machine. Thank you Dan, Dirk, Dong, Hendrik, Nick, Piers, Przemyslaw, Stephanie, Thomas, Yaidel, for teaching me this valuable lesson. I was happy to share an office with Nele, Paul, and during the final year also with Sathish. Some of you I got to know better during the participation in and organization of PhDays. Thanks for some good times Annelies, Clara, Korneel, Laurent, Matthias, Micol, Nele, Peter, Pieter, Roel, Roel, and Ward. A big thank you also goes to Margot for organizing all practical aspects of administration, conferences, and other visits.

To my family, especially my parents and my brother, thank you for stimulating and supporting me in everything I do. For trying to understand what I was talking about when I practiced my first conference presentation. For helping me out with a car ride back to Leuven late in the evening. For too many things to mention here. Thank you.

Thanks to all my friends outside of work, for helping me put my research in perspective with the occasional puzzled look. Thank you Astrid, Ben, Daan, Dries, Ine, Inge, Karel, Kevin, Lies, Marijke, Nele, Rob, Steven, Thomas, Vincent, and Yoeri. Finally, a very special thank you goes out to Elwin, especially for everything for which I fail to find the words.

Ben Jeuris

Abstract

Large data collections often need to be represented by an average value which upholds certain properties, such as reducing the noise level of repeated measurements or representing the central location of the data in case of clustering. Averaging operations are applicable for a wide variety of data types and structures. In this thesis, we focus on the set of positive definite matrices as a whole and on subsets containing all matrices of a desired structure.

The geometric mean of positive numbers possesses various useful properties, which stimulated the search for a generalization of the mean towards positive definite matrices. Since several possible generalizations have been suggested during the development of the matrix geometric mean, an extensive overview of the instances is presented, along with a summary of related means and approximations.

Among the multiple instances of the geometric mean, specific attention is given to the Karcher mean, which is defined as the minimizer of an optimization problem on the set of positive definite matrices. The smooth manifold structure of the set can be exploited in the theory of Riemannian optimization. We apply this rich theory to the setting of the Karcher mean and investigate a large number of first- and second-order optimization techniques. Numerical experiments indicate that the computational cost of an iteration in a second-order method nullifies the advantage of its quadratic convergence. An application in bioinformatics is considered, where we obtain an improvement over the state-of-the-art kernel fusion methods in protein fold classification.

While the Karcher mean is considered as the most natural geometric mean of positive definite matrices, it does not preserve any additional structure present in the given matrices. To account for such structures, we discuss two adaptations of the Karcher mean.

The first adaptation can be defined for any structure imposed on the positive definite matrices and is referred to as the structured geometric mean. We perform

a theoretical analysis of this adaptation, proving that many of the geometric mean properties remain valid in a suitably adjusted form. In the case of linear structures, two steepest descent algorithms are designed by endowing the set of positive definite matrices with either the Euclidean geometry or its natural geometry. A convergence analysis and numerical experiments demonstrate the superiority of the algorithm based on the natural geometry.

The second adaptation of the Karcher mean is designed specifically for Toeplitz matrices and is later generalized to Toeplitz-Block Block-Toeplitz matrices. We refer to this adaptation as the Kähler mean. For Toeplitz matrices, the mean can be found using a simple expression for real matrices and a quick algorithm in the complex case. For Toeplitz-Block Block-Toeplitz matrices, the generalization of the Kähler mean is presented, resulting in the construction of a steepest descent algorithm. The significant computational cost of the algorithm leads us to design a greedy approximation. Numerical experiments show this approximation closely resembles the actual mean and can serve as an efficient initial guess in its computation.

Beknopte samenvatting

Grote dataverzamelingen moeten vaak voorgesteld worden door een gemiddelde waarde die aan bepaalde eigenschappen voldoet, zoals het reduceren van ruis bij herhaalde metingen of het voorstellen van de centrale locatie in een datacluster. Uitmiddelingsoperaties bestaan voor allerlei datatypes en -structuren. In deze thesis focussen we enerzijds op de volledige verzameling van de positief definiete matrices en anderzijds op deelverzamelingen die alle matrices van een gewenste structuur bevatten.

Het geometrisch gemiddelde van positieve getallen bezit verschillende nuttige eigenschappen, wat de zoektocht naar een veralgemening van dat gemiddelde naar positief definiete matrices stimuleerde. Aangezien al verscheidene mogelijke veralgemeningen gesuggereerd zijn tijdens de ontwikkeling van het geometrisch gemiddelde voor matrices, geven we een overzicht van deze veralgemeningen en een samenvatting van de gerelateerde gemiddelden en benaderingen.

Tijdens de bespreking van de vele instanties van het geometrisch gemiddelde wordt er specifieke aandacht besteed aan het Karcher-gemiddelde, dat gedefinieerd is als de oplossing van een optimalisatieprobleem op de verzameling van de positief definiete matrices. De gladde variëteitstructuur van de verzameling kan worden uitgebuit in de theorie van Riemannse optimalisatie. We passen deze theorie toe op de probleemstelling van het Karcher-gemiddelde en onderzoeken een groot aantal optimalisatietechnieken van de eerste en tweede orde. Numerieke experimenten tonen aan dat de berekeningskost van een iteratie in tweede orde methode het voordeel van kwadratische convergentie teniet doet. We beschouwen een toepassing in bio-informatica, waarbij een verbetering wordt behaald ten opzichte van de meest geavanceerde kernelfusiemethoden in de classificatie van proteïnevouwen.

Hoewel het Karcher-gemiddelde beschouwd wordt als het meest natuurlijke geometrisch gemiddelde van positief definiete matrices, bewaart het geen bijkomende structuren van de gegeven matrices. Om rekening te houden met

zulke structuren bespreken we twee aanpassingen van het Karcher-gemiddelde.

De eerste aanpassing kan gedefinieerd worden voor elke bijkomende structuur op de positief definiete matrices en wordt het gestructureerde geometrisch gemiddelde genoemd. We voeren een theoretische analyse van deze aanpassing uit, waarin bewezen wordt dat veel eigenschappen van het geometrisch gemiddelde geldig blijven in een licht aangepaste vorm. In het geval van lineaire structuren worden twee steilste-afdalingsalgoritmes ontworpen: de verzameling van de positief definiete matrices wordt ofwel met de Euclidische geometrie ofwel met zijn natuurlijke geometrie beschouwd. Een convergentieanalyse en numerieke experimenten tonen de superioriteit van het algoritme dat gebaseerd is op de natuurlijke geometrie.

De tweede aanpassing van het Karcher-gemiddelde is specifiek voor Toeplitz-matrices ontworpen en wordt later veralgemeend naar Toeplitz-Block Block-Toeplitz matrices. We noemen deze aanpassing het Kähler-gemiddelde. Voor Toeplitz-matrices kan dit gemiddelde berekend worden met, voor reële matrices, een eenvoudige uitdrukking of, voor complexe matrices, een snel algoritme. Voor Toeplitz-Block Block-Toeplitz matrices wordt de veralgemening van het Kähler-gemiddelde gegeven, met als resultaat de constructie van een steilste-afdalingsalgoritme. Door de significante berekeningskost van het algoritme ontwerpen we ook een *greedy* benadering. Numerieke experimenten tonen aan dat deze benadering veel gelijkenissen vertoont met het echte gemiddelde en dienst kan doen als een efficiënt startpunt in de berekening van dit echte gemiddelde.

Abbreviations

ALM	Ando–Li–Mathias
BFGS	Broyden–Fletcher–Goldfarb–Shanno
BT	Block-Toeplitz
CG	Conjugate Gradient
CLE	Continuous Lyapunov Equation
FFT	Fast Fourier Transform
HPFP	Hierarchical Protein Fold Prediction
MKL	Multiple Kernel Learning
NBMP	Nakamura–Bini–Meini–Poloni
PD	Positive Definite
RBF	Radial Basis Function
SCOP	Structural Classification of Proteins
SD	Steepest Descent
SVM	Support Vector Machine
TBBT	Toeplitz-Block Block-Toeplitz
TR	Trust Region
vpa	Variable-precision arithmetic

List of Symbols

Mathematical notation

$x, y, z, \ldots$	Scalars or vectors
$X, Y, Z, \ldots$	Matrices
$\mathbb{X}, \mathbb{Y}, \mathbb{Z}, \ldots$	Sets of scalars
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \ldots$	Sets of matrices
$\mathfrak{Re}(x)$ ($\mathfrak{Re}(X)$)	The (elementwise) real part of x (X)
$\mathfrak{Im}(x)$ ($\mathfrak{Im}(X)$)	The (elementwise) imaginary part of x (X)
$\mathcal{O}(\cdot)$	Big O notation
I, I_n	The identity matrix of unspecified size and of size n
J, J_n	The counter-identity matrix of unspecified size and of size n (ones on the anti-diagonal)

Matrix operations

X^T	The transpose of X
X^H	The conjugate transpose of X
X^*	The elementwise conjugate of X
$\overline{X}$	$:= JX^*J$
$\otimes$	The Kronecker product
vec	The vectorization operator

vech	The half-vectorization operator, which stacks the upper triangular part of a matrix columnwise
$\mathfrak{B}$	The approximation to the Hessian operator in the BFGS method
$\mathfrak{L}$	The CLE operator $X = \mathfrak{L}(A, Q) \Leftrightarrow AX + XA^H = Q$
∇_F	The displacement operator $:= X \mapsto X - FXF^H$

Matrix measures

tr	The matrix trace
det	The matrix determinant
$\lambda(X)$	The spectrum, or eigenvalues, of X
$\rho(X)$	The spectral radius of X
$\mu(X)$	The spectral condition number of X

Scalar and matrix spaces

$\mathbb{R}, \mathbb{C}$	The set of real or complex numbers
$\mathbb{R}^n, \mathbb{C}^n$	The set of real or complex vectors of length n
$\mathbb{R}^{m\times n}, \mathbb{C}^{m\times n}$	The set of $m \times n$ real or complex matrices
$\mathbb{R}^{++}$	The set of strictly positive, real numbers
$\mathbb{D}$	The complex unit disk
$\mathcal{M}$	A general smooth Riemannian manifold
$\mathcal{H}_n$	The set of $n \times n$ symmetric (Hermitian) matrices
$\mathcal{P}_n$	The set of $n \times n$ positive definite matrices
$\mathcal{T}_n$	The set of $n \times n$ Toeplitz matrices
$\mathcal{B}_{n,N}$	The set of Block-Toeplitz matrices, consisting of n by n blocks, each block an $N \times N$ matrix
$\mathcal{T}_{n,N}$	The set of Toeplitz-Block Block-Toeplitz matrices, consisting of n by n blocks, each block an $N \times N$ matrix

$\mathcal{T}_n^+, \mathcal{B}_{n,N}^+, \mathcal{T}_{n,N}^+$	The sets containing all positive definite elements in $\mathcal{T}_n$, $\mathcal{B}_{n,N}$, and $\mathcal{T}_{n,N}$ respectively
$\mathcal{D}_n$	$:= \left\{X \in \mathbb{C}^{n\times n} \;\middle|\; I_n - X\overline{X} > 0\right\}$
$\mathcal{SD}_n$	The Siegel disk $:= \left\{X \in \mathbb{C}^{n\times n} \;\middle|\; I_n - XX^H > 0\right\}$
$\mathcal{U}^{-1}$	$:= \left\{X^{-1} \;\middle|\; X \in \mathcal{U}, \det(X) \neq 0\right\}$
$\mathcal{U}_S$	$:= \left\{S^H X S \;\middle|\; X \in \mathcal{U}\right\}$

Matrix means

$\mathbf{G}$	The geometric mean
$\mathbf{H}$	The harmonic mean
$\mathbf{A}$	The arithmetic mean
$\mathbf{K}$	The Karcher mean
$\mathbf{G}_{\mathcal{U}}$	The structured geometric mean for the matrix structure indicated by set $\mathcal{U}$
$\mathbf{LE}$	The log-Euclidean mean
$\mathbf{P}_t$	The power mean with power $t \in [-1, 1] \setminus \{0\}$
$\mathbf{S}_k$	The inductive mean of k positive definite matrices
$\mathbf{B}_{\mathcal{S}}, d_{\mathcal{S}}, f_{\mathcal{S}}$	The barycenter of matrices in the set $\mathcal{S}$ for distance measure $d_{\mathcal{S}}$ and corresponding cost function $f_{\mathcal{S}}$

Differential geometry and Riemannian optimization

δ	A distance measure
γ	A geodesic
$T_X\mathcal{M}$	The tangent space to the manifold $\mathcal{M}$ at point X
$\xi_X, \eta_X, \ldots$	Tangent vectors at point X
$\xi, \eta, \ldots$	Vector fields
$\mathrm{D}f(X)[\xi_X]$	The classical Fréchet derivative of the function f at point X in the direction of the tangent vector ξ_X

J_f	The Jacobian matrix of $\operatorname{vec}(f(X))$ w. r. t. the variable $\operatorname{vec}(X)$
∇	The Levi–Civita connection
$\langle \cdot , \cdot \rangle_X$	An inner product on the tangent space at point X
$\operatorname{grad} f(X)$	The Riemannian gradient of the function f at point X
$\operatorname{Hess} f(X)[\xi_X]$	The Riemannian Hessian of the function f at point X evaluated for the tangent vector ξ_X
R_X	A retraction mapping at point X
$\mathfrak{T}_{\xi_X}$	A vector transport in the direction of tangent vector ξ_X

Contents

List of Figures

List of Tables

Chapter 1

Introduction

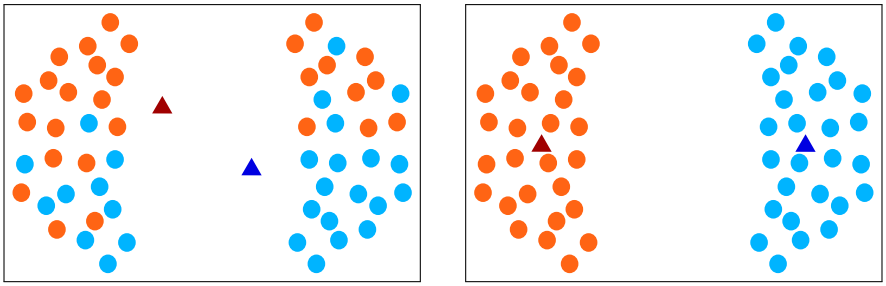
1.1 Context

Large collections of data elements often require a representative to encompass the group as a whole. The desired properties of this representative determine the way it should be chosen. In general, such a representative is referred to as a *center size*. As the name suggests, a center size is designed to act as a central value with respect to the data collection.

The two most well-known examples of a center size are the *mean* and the *median*. Typically, when the idea of a mean is considered, most people identify this concept with the arithmetic mean, given by the sum of the elements divided by their number. Similarly, the median is often identified with the sample median, given by the middle value in an ordered sequence of the elements. However, these are only two examples in a very extensive collection of definitions.

Despite the many possibilities for both means and medians, some intrinsic properties transcend the separate definitions and define the fundamental difference between these two popular center sizes. While a median is known to be more robust to outliers, a mean will incorporate as much as possible the information in the data elements. In this thesis, we will focus on means rather than medians.

A classical application of a mean is found in the averaging of repeated measurements to reduce measurement noise. Depending on the type of noise on the data, an appropriate mean can be chosen to maintain the desired signal while canceling the imposed noise structure. From a statistical point of view,



(a) An initial guess for the clusters. (b) The final clusters.

Figure 1.1: The first and final step in a clustering algorithm based on the cluster means (triangles).

these means are often referred to as unbiased estimators. A more advanced application is found in the field of image segmentation and is illustrated in Figure 1.1. Suppose the points in the figure represent some features of the various pixels in an image, where we want to separate an object from the background. A typical approach is to split the feature points into two groups of which the means are sufficiently different. Note that a slight change in this approach results in the well-known k-means clustering, where the data is also divided into clusters, but now based on optimal cluster concentration around its mean.

The concept of a mean is applicable to a wide variety of data types and structures, such as scalars, vectors, matrices, or more abstract constructions. In this thesis, we focus on a special matrix property, known as *positive definite* matrices [65, 135]. This matrix property has proven its usefulness in various applications, such as complex analysis, the theory of vibrations of mechanical systems, and other areas of applied matrix theory.

Some concrete examples of positive definite matrices include the Hessian matrix at a minimizer of the corresponding cost function, covariance matrices, and the matrix representation of the algebraic or trigonometric moments of nonnegative functions. Often, such matrices also exhibit some additional structure besides positive definiteness. For example, the matrix representations of the algebraic and trigonometric moments of a nonnegative function will possess the Hankel and Toeplitz (Section 1.2.4) matrix structure, respectively.

On the set of positive definite matrices, the matrix geometric mean will be our main averaging operation. This matrix mean is an extension of the geometric mean of positive numbers, defined for k numbers as the k-th root of their

product. The geometric mean is known to have interesting properties when averaging ratios or exponential growth, as indicated in the following example.

Example 1.1. Suppose an exponentially growing variable is estimated from separate experiments, resulting in k growth predictions $f_i(t) = e^{\lambda_i t}$, $i = 1, \ldots, k$. The separate predictions are combined using the arithmetic and geometric means to obtain, respectively,

$$\mathbf{A}(f_1(t), \ldots, f_k(t)) = \frac{e^{\lambda_1 t} + \cdots + e^{\lambda_k t}}{k},$$

$$\mathbf{G}(f_1(t), \ldots, f_k(t)) = \left(e^{\lambda_1 t} \cdots e^{\lambda_k t}\right)^{\frac{1}{k}} = e^{\frac{\lambda_1 + \cdots + \lambda_k}{k} t}.$$

While the expression for the arithmetic mean does not give a clear view on the averaged growth rate, the geometric mean returns an exponentially growing function with an intuitive growth rate.

Concerning the example of covariance matrices, we will discuss an *application in bioinformatics* in Section 3.7. The application consists of a number of proteins, for which several covariance matrices are constructed originating from separate data features. These covariance matrices are then combined in a geometric mean, in an effort to unite the separate information with as little loss as possible.

1.2 Some basics

1.2.1 Positive definite matrices

Given a matrix $A \in \mathbb{C}^{n \times n}$, we call the scalars $\lambda_i \in \mathbb{C}$, $i = 1 \ldots, n$, the *eigenvalues* of the matrix A if $A - \lambda_i I_n$ is singular, with I_n the identity matrix of size n. Any vector $v_i \neq 0 \in \mathbb{C}^n$, which satisfies $Av_i = \lambda_i v_i$, is called an eigenvector of A corresponding to the eigenvalue λ_i.

A *positive semidefinite* matrix $A \in \mathbb{C}^{n \times n}$ is denoted as $A \geq 0$ and is defined as a matrix for which

$$x^H A x \geq 0, \qquad \forall x \neq 0 \in \mathbb{C}^n,$$

where x^H denotes the conjugate transpose of x. A *positive definite* matrix $A \in \mathbb{C}^{n \times n}$ satisfies the same definition but with strict inequality, and is accordingly denoted as $A > 0$. A partial order is placed on the set of positive (semi)definite matrices as follows: for positive (semi)definite matrices A and B, we write $A \geq B$ if $A - B \geq 0$, and similar for strict inequalities. The definition of a positive definite matrix is equivalent to the condition that it is a Hermitian matrix with positive real eigenvalues.

Note that a positive definite matrix being Hermitian is an inherent consequence of this definition. If the above definition would be adapted for real matrices in $\mathbb{R}^{n\times n}$ by only considering real vectors $x \in \mathbb{R}^n$, a real positive definite matrix would no longer be inherently symmetric. We demonstrate this in the following example.

Example 1.2. Consider the matrix

$$A = \begin{bmatrix} 5 & 4 \\ 1 & 2 \end{bmatrix},$$

which has positive real eigenvalues $\lambda_1 = 6$ and $\lambda_2 = 1$, but is not symmetric. For any non-zero real vector $x = [a, b]^T \in \mathbb{R}^2$, we find that

$$x^T Ax = 5a^2 + 5ab + 2b^2 = \left(\sqrt{5}a + \frac{\sqrt{5}}{2}b\right)^2 + \frac{3}{4}b^2 > 0,$$

with x^T the transpose of x. However, entering a complex vector for x into the product will in general not result in a real value, let alone a positive real one.

Many applications working with real positive definite matrices explicitly impose symmetry. In this thesis, we will only consider the complex definition of positive definite matrices, hence any real or complex positive definite matrix will always implicitly be assumed to be symmetric or Hermitian, respectively. The set of all positive definite $n \times n$ matrices will be denoted by $\mathcal{P}_n$ and that of the symmetric (Hermitian) $n \times n$ matrices by $\mathcal{H}_n$.

1.2.2 Matrix functions and derivatives

In general, a function of a matrix could refer to any kind of operation involving matrices, such as performing operations on its elements, computing the matrix trace or determinant, taking a matrix norm, etc. We are mostly interested in two types of *matrix functions*:

- real-valued matrix functions $f : \mathbb{C}^{n\times n} \to \mathbb{R}$, which include cost functions in optimization problems and matrix norms;
- matrix functions $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$, which usually are the matrix equivalent of some scalar function. For a diagonalizable matrix $A = UDU^{-1}$, with D a diagonal matrix having diagonal elements d_i, these functions are defined as $f(A) = Uf(D)U^{-1}$, where $f(D)$ is diagonal having elements $f(d_i)$. If the matrix is not diagonalizable, it can at least

be written in the Jordan canonical form $A = ZKZ^{-1}$ [63, 64]. In this case, the function is defined as $f(A) = Zf(K)Z^{-1}$, where the operation $f(K)$ is split over the separate Jordan blocks such that

$$K_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \mapsto f(K_i) = \begin{bmatrix} f(\lambda_i) & f'(\lambda_i) & \cdots & \frac{f^{(n_i-1)}(\lambda_i)}{(n_i-1)!} \\ & f(\lambda_i) & \ddots & \vdots \\ & & \ddots & f'(\lambda_i) \\ & & & f(\lambda_i) \end{bmatrix},$$

with n_i the size of Jordan block K_i.

In both cases, the classical *Fréchet derivative* [64] of the function f at a matrix $X \in \mathbb{C}^{n\times n}$ in the direction of a matrix $E \in \mathbb{C}^{n\times n}$ can be defined as

$$\mathrm{D}f(X)[E] = \lim_{t\to 0} \frac{f(X+tE) - f(X)}{t} = \frac{\mathrm{d}}{\mathrm{d}t} f(X+tE)\bigg|_{t=0}.$$

Since Fréchet derivatives will be computed throughout the thesis, we recall some of their basic properties and also present the derivatives of some basic matrix functions. Suppose $f, g : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ are given matrix functions, $h(X) = X^{-1}$ the *matrix inverse* function, and $s(P) = P^{1/2}$ the *matrix square root* function, with X invertible and P positive definite.

$$\begin{aligned}
&\mathrm{D}(fg)(X)[E] = \mathrm{D}f(X)[E]g(X) + f(X)\,\mathrm{D}g(X)[E], && \text{product rule,}\\
&\mathrm{D}(f\circ g)(X)[E] = \mathrm{D}f\big(g(X)\big)\big[\mathrm{D}g(X)[E]\big], && \text{chain rule,}\\
&\mathrm{D}h(X)[E] = -X^{-1}EX^{-1}, && \text{inversion [43],}\\
&\mathrm{D}s(P)[E]\,P^{\frac{1}{2}} + P^{\frac{1}{2}}\,\mathrm{D}s(P)[E] = E, && \text{square root.}
\end{aligned}$$

The derivative of the matrix square root is obtained by applying the product rule to the definition $P^{1/2}P^{1/2} = P$ and can be recognized (and solved) as a *continuous Lyapunov equation*.

Finally, since cost functions involving matrices are often based on some type of matrix norm, we discuss two important examples and their derivative. Note that a norm can be seen as a matrix function $\|\cdot\| : \mathbb{C}^{m\times n} \to \mathbb{R}$, which implies that its (directional) derivatives should also be mapped onto $\mathbb{R}$.

The *Frobenius norm* of a matrix $X \in \mathbb{C}^{m\times n}$ is defined as $\|X\|_F = \sqrt{\mathrm{tr}(X^HX)}$. Its derivative can be found as

$$\mathrm{D}(\|\cdot\|_F)(X)[E] = \frac{\mathfrak{Re}\left(\mathrm{tr}(E^HX)\right)}{\|X\|_F},$$

where $\mathfrak{Re}$ returns the real part of a complex number.

The *spectral norm* of $X \in \mathbb{C}^{m\times n}$ is given by $\|X\|_2 = \sigma_1$, with σ_1 the largest singular value of X. The derivative of this norm is no longer well-defined for all matrices X because of the presence of the maximum function in its definition (when choosing the *largest* singular value). However, in case the largest singular value of X is strictly larger than all its other singular values, the derivative is well-defined and given by

$$\mathrm{D}(\|\cdot\|_2)(X)[E] = \mathfrak{Re}\left(U_1^H E V_1\right),$$

where $X \in \mathbb{C}^{m\times n}$ is a matrix with a strictly largest singular value, and U_1 and V_1 respectively are the left and right singular vector of X corresponding to this largest singular value. For real matrices, the derivative is identical and presented by Giles [62].

1.2.3 Matrix manifolds and Riemannian optimization

From a differential geometric point of view, the set of positive definite matrices $\mathcal{P}_n$ is known to be a *smooth manifold* [21, 77, 114]. Such a manifold is one of the most well-behaved generalizations of a classical, flat Euclidean space, since it closely resembles this Euclidean space locally through its definition.

For a general d-dimensional manifold $\mathcal{M}$, this local resemblance of the Euclidean space is formalized using *charts*, which are defined as bijections ϕ from an open subset $\mathcal{U} \subset \mathcal{M}$ onto an open subset of $\mathbb{R}^d$. The charts are gathered in an *atlas* $\{(\mathcal{U}_i, \phi_i)\}$ in order to get a full description of $\mathcal{M}$. To obtain a smooth manifold $\mathcal{M}$, a *smooth atlas* $\mathcal{A}$ of the space is required, which is defined as a countable collection $\{(\mathcal{U}_i, \phi_i)\}$ with the following properties,

- The open sets $\mathcal{U}_i$ cover $\mathcal{M}$,
$$\bigcup_i \mathcal{U}_i = \mathcal{M},$$
- For all i, the chart
$$\phi_i : \mathcal{U}_i \to \mathbb{R}^d$$
is a homeomorphism onto an open subset of $\mathbb{R}^d$,
- For any pair i, j with $\mathcal{U}_i \cap \mathcal{U}_j \neq \emptyset$, the sets $\phi_i(\mathcal{U}_i \cap \mathcal{U}_j)$ and $\phi_j(\mathcal{U}_i \cap \mathcal{U}_j)$ are open in $\mathbb{R}^d$ and the mapping
$$\phi_j \circ \phi_i^{-1} : \mathbb{R}^d \to \mathbb{R}^d$$
is smooth (differentiable for all degrees of differentiation) on $\phi_i(\mathcal{U}_i \cap \mathcal{U}_j)$.

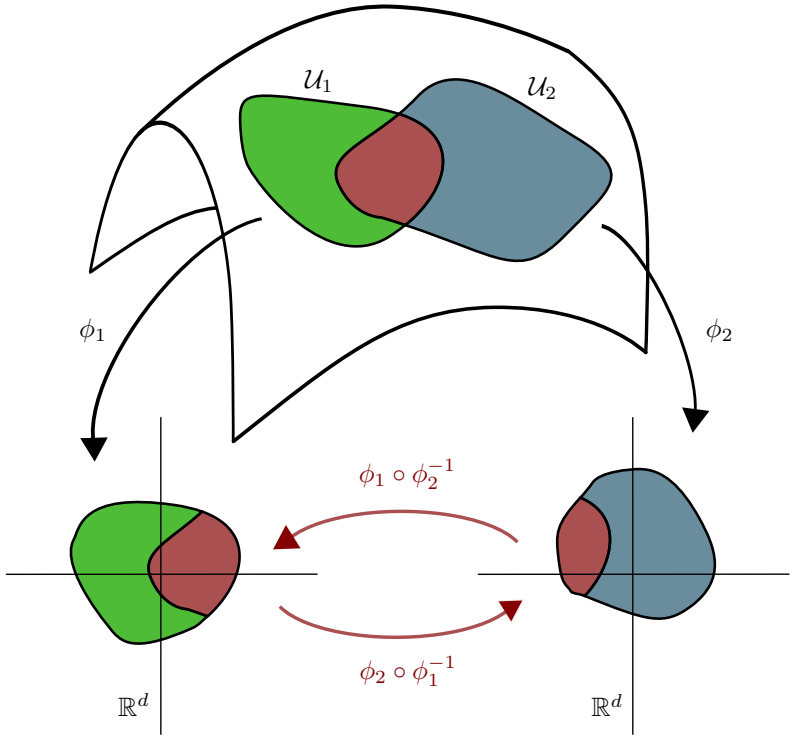


Figure 1.2: The charts of a manifold.

This definition of a smooth atlas is illustrated in Figure 1.2. A smooth atlas $\mathcal{A}$ can be extended by adding charts $(\mathcal{U}, \phi)$ such that $\mathcal{A} \cup \{(\mathcal{U}, \phi)\}$ is still a smooth atlas. By gathering all such charts, the smooth atlas $\mathcal{A}$ is extended to a *maximal atlas* $\mathcal{A}^+$. Finally, for $\mathcal{M}$ to be a smooth manifold, it should be accompanied by a maximal atlas $\mathcal{A}^+$, of which the induced topology on $\mathcal{M}$ is Hausdorff and second-countable. The manifold structure of $\mathcal{P}_n$ and its associated concepts will be discussed in Section 3.3.

The manifold $\mathcal{P}_n$ is not only smooth, it is also typically associated with a *natural geometry* different from the classical Euclidean geometry. The geometry is considered natural because it is inherently linked to the congruence invariance of $\mathcal{P}_n$, which means that any congruence transformation $\mathcal{P}_n \to \mathcal{P}_n : A \mapsto S^H A S$, with $S \in \mathbb{C}^{n \times n}$ invertible, is an automorphism of $\mathcal{P}_n$. Furthermore, the corresponding geodesics, or lines of shortest distance, can be extended infinitely on the manifold. The foundation of this geometry is determined by a smoothly varying inner product, defined on the tangent space at each point

of the manifold. From this construction, other concepts, such as a distance measure, geodesics, and more can be derived.

While $\mathcal{P}_n$ has a naturally associated geometry available, this is not the case for any general matrix manifold. A general matrix manifold can be associated with a wide range of geometries, of which none may present itself as most natural. A smooth manifold accompanied by a smoothly varying inner product on its tangent space is often referred to as a *Riemannian manifold*.

Once we have determined the manifold containing a specific matrix structure of interest and have endowed this manifold with a geometry, optimization can be performed on the manifold. In fact, the structure of the manifold will be exploited in the so-called *Riemannian optimization*. Specifically, a constrained optimization problem

$$\min f(X), \qquad s.t. \;\; X \in \mathcal{M},$$

with $\mathcal{M}$ a general Riemannian manifold, is treated as an unconstrained optimization problem on the manifold

$$\min_{X \in \mathcal{M}} f(x).$$

This concept is further clarified in Figure 1.3, where the Riemannian version of the classical steepest descent algorithm is presented. The figure shows a simplified representation of a manifold on which the contour lines of the cost function f are indicated by thin red lines. At each iteration, the Riemannian version of the gradient of the cost function is computed (the blue arrows indicate the opposite direction), after which a path on the manifold corresponding to the indicated direction is followed (green lines). By following this procedure, optimization is performed while remaining on the manifold at each step. The local operation of translating a tangent vector (blue arrow) to the manifold (green line) is called a *retraction*, hence this type of optimization is sometimes referred to as *retraction-based optimization*.

The steepest descent method represented in Figure 1.3 is an example of a first-order optimization technique. More advanced second-order optimization methods can also be developed using generalized constructions such as a vector transport, affine connections, the Riemannian Hessian, ...

1.2.4 Additional matrix structure

A possible, but overly simplistic view of matrices is to consider them as simply a collection of numbers presented in an ordered grid. More interesting is the

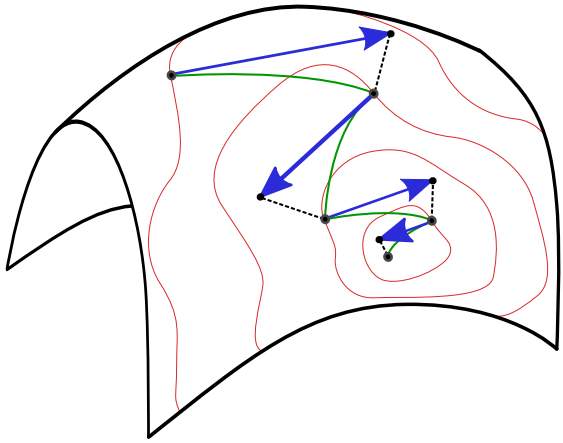


Figure 1.3: Riemannian optimization.

recognition and exploitation of some structure present in the elements of these matrices.

As discussed in Section 1.2.1, positive definite matrices already exhibit the symmetric (Hermitian) matrix structure. However, this is only one example from an extremely wide variety of possible structures. In fact, when using matrices for data representation, the appearance of a specific matrix structure is usually inherently linked to some characteristic of the data itself. As such, maintaining the structure throughout matrix operations could prove vital to the interpretation of the result.

Apart from being necessary for interpretation in an application, structured matrices also present some computational advantages. The structure often allows one to represent the matrices using only a small number of parameters. This reduced representation can also be exploited when performing matrix operations, such as multiplication by a vector.

In our applications, the structure of interest will often be a combination of positive definiteness with some additional structure. We discuss a few popular matrix structures, both linear and non-linear.

Linear matrix structures When referring to a matrix structure as linear, this indicates that any linear combination of two matrices of the specified structure

will maintain that structure. A basic example of a linear matrix structure is a *symmetric* or *Hermitian* matrix.

Two more advanced matrix structures are the so-called *Toeplitz* and *Hankel* matrices, which have fixed elements on the diagonals and anti-diagonals respectively. A general Toeplitz matrix T and Hankel matrix H are given by

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}, \qquad H = \begin{bmatrix} t_{-n+1} & t_{-n+2} & \cdots & t_0 \\ t_{-n+2} & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & t_{n-2} \\ t_0 & \cdots & t_{n-2} & t_{n-1} \end{bmatrix},$$

with $t_{-n+1}, \ldots, t_{n-1} \in \mathbb{C}$.

These matrix structures are among the most studied structures, as they have numerous areas of application, such as polynomial and power series computation [102], representation of correlation matrices, Markov chains and queuing theory [29], and the mathematical modeling of any problem involving some sort of shift invariance in terms of space or time.

Non-linear matrix structures Another structure which is often mentioned in the same context as the Toeplitz and Hankel matrices, but is no longer linear, is the *Vandermonde* structure. A general Vandermonde matrix V is given by

$$V = \begin{bmatrix} 1 & t_0 & \cdots & t_0^{n-1} \\ 1 & t_1 & \cdots & t_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_{n-1} & \cdots & t_{n-1}^{n-1} \end{bmatrix},$$

with $t_0, \ldots, t_{n-1} \in \mathbb{C}$. Note that a positive definite Vandermonde matrix V must have $t_i = \alpha^i$, $i = 0 \ldots, n-1$, for a real positive $\alpha > 1$. Hence such a matrix depends only on a single parameter.

A basic, but widely used example of a non-linear matrix structure is *orthogonality*. An orthogonal (or unitary) matrix U satisfies the equation $U^H U = I$, with I the identity matrix.

In the context of efficient matrix representations, we also encounter the *low rank* matrices. Suppose $X \in \mathbb{C}^{n \times n}$ a matrix of rank $r \ll n$. This indicates the existence of a factorization $X = CR^H$, with $C, R \in \mathbb{C}^{n \times r}$. While X itself contains n^2 elements, its factorization can be described using $2nr$ parameters only.

Unfortunately, a positive definite matrix is always of full rank, hence the intersection with low rank matrices would be void (although a special adaptation of the geometric mean was considered by Bonnabel, Collard, and Sepulchre [31]). However, a variation of the matrices of low rank is given by the matrices of *low displacement rank*. Such a displacement can be defined in several ways, depending on which displacement structure should be emphasized [72]. For example, the displacement operator $\nabla_F : X \mapsto X - FXF^H$, with F the matrix with ones on the first subdiagonal and zeros everywhere else, is inspired by the Toeplitz structure, since

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix} \text{ becomes } \nabla_F T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & 0 & \cdots & 0 \end{bmatrix}.$$

The displaced matrix $\nabla_F T$ is of rank 2 for any Toeplitz matrix T. Interestingly, the displacement rank of the inverse of any Toeplitz matrix also equals 2. As a consequence, matrices of low displacement rank under displacement operator ∇_F are sometimes referred to as *Toeplitz-like* matrices.

1.2.5 A convenient notation

Throughout this thesis, expressions will be presented containing a multitude of variables. We aim to clearly indicate the difference between main and auxiliary variables by using the following notation. We denote a function f, defined as $f(X) = g(A, B, C)$, with auxiliary variables $A = g_1(X)$, $B = g_2(X)$, and $C = g_3(X)$, as

$$f(X) = g(A, B, C),$$

$$\begin{bmatrix} A = g_1(X), \\ B = g_2(X), \\ C = g_3(X), \end{bmatrix}$$

indicating that f is the main object of interest.

1.3 Goals and contributions

The main topic of this thesis can be summarized as the averaging of positive definite matrices, with or without the preservation of additional matrix structure,

e. g., positive definite Toeplitz matrices. Concerning averaging techniques, we start by studying the matrix geometric mean, which extends the geometric mean of positive numbers, resulting in an elaborate collection of definitions. Afterwards, newly developed means for structured matrices are discussed using the adaptive concept of a barycenter, which acts as a center of mass in the matrix space according to the associated geometry.

The matrix geometric mean Generalizing the scalar geometric mean to a matrix setting is not straightforward because of the lack of commutativity for the matrix product. Instead, a list of defining properties was derived [6], which was soon shown to allow an extensive collection of definitions [6, 21, 25, 74, 87, 97, 103]. Therefore, our first objective is the presentation of an overview of the existing algorithms for the geometric mean.

During the development of the geometric mean, a great number of related means have also been proposed [5, 6, 7, 28, 89], which do not satisfy all required properties of a geometric mean, but display some interesting features nonetheless. Finally, some approximations of the geometric mean have been suggested [86, 88, 100, 101, 110], aiming for a close estimate to a geometric mean with a low computational cost. We will provide an overview and analysis of the most interesting related means and approximations, and we introduce a new approximation to the geometric mean based on arithmetic and harmonic iterations.

One particular instance of the geometric mean, known as the Karcher mean [21, 74], is defined as the barycenter of the given matrices with respect to the natural geometry of $\mathcal{P}_n$. Its computation is performed using the theory of Riemannian optimization [1], resulting in a collection of first- and second-order optimization techniques [27, 54, 53, 100, 105, 109, 111]. We present a survey of existing algorithms, along with a first-time application of the Riemannian BFGS method to this optimization problem. Furthermore, an explicit expression of the Riemannian Hessian of the cost function will be introduced. Finally, we apply the Karcher mean and our arithmetic-harmonic approximation in the protein fold recognition problem from the field of bioinformatics, improving the state-of-the-art protein fold classification methods.

The matrix geometric mean of positive definite matrices will be discussed in Chapters 2 and 3. The content of these chapters was published as part of

> Jeuris, B., Vandebril, R., and Vandereycken, B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis 39*, 1 (2012), pp. 379–402.

Jeuris, B., and Vandebril, R. Geometric mean algorithms based on harmonic and arithmetic iterations. *Lecture Notes in Computer Science 8085* (2013), pp. 785–793.

Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Protein fold recognition using geometric kernel data fusion. *Bioinformatics 30*, 13 (2014), pp. 1850–1857.

Adaptations for additional structure While desirable from an application point of view, preservation of additional matrix structure had not yet found its way into the definition of the geometric mean. As a first attempt, we discuss an adaptation of the Karcher mean by restricting the search space in the corresponding optimization problem, or equivalently, by optimizing over a submanifold. A proof of existence is provided under certain conditions, along with a convergence analysis of two suggested algorithms for linear structures.

Special attention is paid to the set of positive definite Toeplitz matrices, not only in the case of the adapted Karcher mean, but also in case of a different, application-inspired geometry. This new geometry stems from a transformation of positive definite Toeplitz matrices in signal processing [11, 13, 73, 127], and is introduced as the natural geometry in the transformed space. We present the origin of the transformation, the derivation of the mean, and its resulting properties. An explicit formula is provided for real matrices and a fast algorithm in the complex case.

Finally, the application-inspired geometry for positive definite Toeplitz matrices will be generalized towards positive definite Toeplitz-Block Block-Toeplitz matrices [16, 44, 48, 69, 73, 118]. We present some possible generalizations of the transformation, discuss the natural geometry in the transformed space, and derive optimization algorithms for the computation of the mean. A global and a greedy optimization algorithm will be constructed, where the greedy version has similar properties as the global version, but is computationally more efficient.

The different approaches to the averaging of positive definite matrices with additional structure will be discussed in Chapters 4 and 5. The content of these chapters was published or submitted as part of

Bini, D., Iannazzo, B., Jeuris, B., and Vandebril, R. Geometric means of structured matrices. *BIT Numerical Mathematics 54*, 1 (2014), pp. 55–83.

Jeuris, B., and Vandebril, R. The Kähler mean of Block-Toeplitz matrices with Toeplitz structured blocks. Submitted to *SIAM journal on Matrix Analysis and Applications.*

Throughout the thesis, our theoretical analysis will be accompanied by numerical experiments. All these experiments are performed on an Intel® Core™i5-2540M CPU @ 2.60 GHz using MATLAB R2012a–R2014a. When the computational speed of algorithms is tested, we avoid the use of MATLAB built-in functions to obtain interpretable and comparable timings.

1.4 Outline

An overview of this thesis is given, briefly indicating the content of each chapter. The majority of the content has been subjected to peer-reviewing at international journals, which is pointed out for each chapter by mentioning the corresponding publications.

Chapter 1 In this first chapter, we have provided a general introduction to the context of the thesis, along with some basic theory of the main concepts. Afterwards, the goals and corresponding contributions of the thesis were stated.

Chapter 2 We start with a description of the historical generalization of the scalar geometric mean to the matrix geometric mean and present some of the first instances thereof. Additionally, some interesting means, closely related to the geometric mean, will be presented. Finally, we analyze some existing and newly developed approximations, based both on the expression for the mean of two matrices and on its computation using the arithmetic and harmonic means. The content of this chapter was published as part of Jeuris et al. [71] and Jeuris et al. [70].

Chapter 3 In this chapter, the Karcher mean will be discussed, a natural instance of the geometric mean which is stated as an optimization problem. Both first- and second-order optimization techniques for its computation are presented along with the underlying Riemannian optimization concepts. We finish the chapter with an application in bioinformatics, where the state-of-the-art in protein fold recognition is improved. The content of this chapter was published as part of Jeuris et al. [71] and Zakeri et al. [134].

Chapter 4 We discuss an adaptation of the Karcher mean which accounts for additional matrix structure by restriction of the search space. Special attention is paid to linear structures, with the Toeplitz matrices as a working example. We provide a theoretical analysis for general structures and a convergence analysis

of two newly developed algorithms for linear structures. The content of this chapter was published as a part of Bini et al. [24].

Chapter 5 In this chapter, we present the computation of a least squares mean of positive definite Toeplitz and Toeplitz-Block Block-Toeplitz matrices using a different geometry (the so-called Kähler metric). Aside from the actual least squares mean, a greedy approximation will also be discussed for Toeplitz-Block Block-Toeplitz matrices. This approximation satisfies the same properties as the global mean, but significantly reduces the computational cost. The content of this chapter concerning the mean of positive definite Toeplitz matrices was published as part of Bini et al. [24].

Chapter 6 We formulate the conclusions of the thesis, indicate the main contributions, and speculate on possible future research.

Chapter 2

The matrix geometric mean

This chapter describes the generalization of the geometric mean for positive scalars to the matrix geometric mean. Because of the lack of a unique definition, we present an overview of the most important instances proposed over time. One of the main instances of the matrix geometric mean, called the Karcher mean, is computed using the expansive theory of Riemannian optimization. The definition of this mean and various Riemannian optimization algorithms will therefore be presented separately in Chapter 3.

Apart from these instances of the geometric mean, a summary of closely related means is given. These related means were often designed with the geometric mean in mind, but do not satisfy all its desired properties. Additionally, we present some approximations to the matrix geometric mean based on its known value for two matrices. This known value can be obtained using an explicit expression or by iteratively taking the arithmetic and harmonic mean of the two matrices. Both approaches are used to obtain approximations to the matrix geometric mean for more than two matrices.

In Chapter 3, some of the related means and the geometric mean approximations will be further investigated in terms of proximity to the Karcher mean and their appropriateness as an initial guess for the optimization algorithms.

The content of this chapter was published as a part of

> Jeuris, B., Vandebril, R., and Vandereycken, B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis 39*, 1 (2012), pp. 379–402.

Jeuris, B., and Vandebril, R. Geometric mean algorithms based on harmonic and arithmetic iterations. *Lecture Notes in Computer Science 8085* (2013), pp. 785–793.

2.1 Introduction

A mean is, in general, simply a center size subjected to certain generic properties such as idempotency (the mean of $(A, \ldots, A)$ equals A), invariance under a permutation of the elements and homogeneity (the mean of $(\lambda A_1, \ldots, \lambda A_k)$ equals λ times the mean of $(A_1, \ldots, A_k)$). However, these generic properties alone do not uniquely define a mean, so there can be many different types of means. Two common examples of means are the arithmetic mean and harmonic mean, classically defined for positive scalars $(a_1, \ldots, a_k)$ as

$$\mathbf{A}(a_1, \ldots, a_k) = \frac{a_1 + \cdots + a_k}{k}, \qquad \mathbf{H}(a_1, \ldots, a_k) = \left(\frac{a_1^{-1} + \cdots + a_k^{-1}}{k} \right)^{-1},$$

respectively.

We discuss the *geometric mean*, which for positive real numbers $(a_1, \ldots, a_k)$ is defined as

$$\mathbf{G}(a_1, \ldots, a_k) = (a_1 \cdots a_k)^{\frac{1}{k}}. \tag{2.1}$$

When conveying this definition to the set of positive definite (PD) matrices $\mathcal{P}_n$, we see that the formula above can not be readily extended to matrices due to their non-commutativity. However, a list of desired properties for the general geometric mean can be derived from this scalar expression.

These properties (listed in Section 2.2) have proven to be useful in various applications, e. g., radar technology [12], medical imaging [56], mechanics [94] and image processing [107]. All these areas display situations in which the information about the current system is being represented in a collection of PD matrices. In order to perform calculations on these matrices, such as averaging and interpolation, we need algorithms that preserve the positive definiteness. This preservation is one of the useful properties of the geometric mean. Another property of the mean provides advantages in the area of elasticity calculations of structures [94]. In these calculations, both a PD elasticity matrix and its inverse, the compliance matrix, are used. Hence, given a collection of these elasticity matrices and a collection consisting of the corresponding compliance matrices, the geometric means of both matrix collections will again be each others inverse (as stated in property 8 in Section 2.2).

Thanks to the wide range of practical and theoretical applications, matrix means receive a lot of attention from present-day scientists. A consequence of the diversity of application areas is the wide variety of approaches to define and compute the matrix geometric mean. Some constructions are based on intuitive interpretations of the geometric mean (Section 2.3), while others prefer to think of it as an optimization problem. The latter approach results in the Karcher mean, which is discussed separately in Chapter 3 and is often considered to be the most natural matrix geometric mean.

The main contribution of this chapter (as well as the next chapter) is to present a survey of algorithms for computing a matrix geometric mean. We recall the theoretical foundation for the analytically known mean of two matrices and the interpretations of the algorithms based on intuitive approaches. Afterwards, we also discuss a number of approximations of the matrix geometric mean. Some of these are based on the exact expression for the mean of two matrices, while others generalize the theorem which states that iteratively taking the arithmetic and harmonic mean of two matrices results in their geometric mean [8].

The organization of this chapter is as follows: we start by listing the desired properties of the geometric mean and the resulting unique definition in case of two matrices in Section 2.2. Next, in Section 2.3, we discuss some intuitively appealing algorithms based on planar approaches: the ALM, NBMP and the CHEAP mean. However, these appealing interpretations will not always lead to very efficient numerical algorithms. Afterwards, some other means which have been presented in the context of PD matrices and which exhibit some relation to the geometric mean are presented in Section 2.4. Finally, in Section 2.5, we discuss a number of basic approximations to the matrix geometric mean. Throughout the chapter we compare the performance of the algorithms discussed. The analysis of the geometric mean approximations is continued in Chapter 3, where the proximity to the Karcher mean and the appropriateness as an initial guess for the optimization algorithms is investigated.

2.2 The geometric mean of two matrices

The scalar geometric mean (2.1) can not be readily extended to PD matrices because the matrix product is not commutative. Indeed, $(A_1 \cdots A_k)^{1/k}$ is not invariant under permutation, which is one of the most basic properties of means. Hence a list of desired properties has been composed instead, often referred to as the ALM (Ando–Li–Mathias) list [6, 25]. Because of the importance of these properties, we summarize them here, using the partial ordering of symmetric (Hermitian) matrices: a positive semidefinite matrix A is denoted by $A \geq 0$.

Similarly, $B \geq C$ is a simplified notation for $B - C \geq 0$. The same approach is used for positive definiteness with the strict inequality. The ALM list, using PD matrices $A_1, \ldots, A_k$, where we denote the *geometric mean* by $\mathbf{G}(A_1, \ldots, A_k)$, is given by the following properties:

1. Consistency: if $A_1, \ldots, A_k$ commute, then $\mathbf{G}(A_1, \ldots, A_k) = (A_1 \cdots A_k)^{\frac{1}{k}}$.
2. Joint homogeneity:
$$\mathbf{G}(\alpha_1 A_1, \ldots, \alpha_k A_k) = (\alpha_1 \ldots \alpha_k)^{\frac{1}{k}} \mathbf{G}(A_1, \ldots, A_k), \qquad \alpha_1, \ldots, \alpha_k > 0.$$
3. Invariance under permutation: $\mathbf{G}(A_{\pi(1)}, \ldots, A_{\pi(k)}) = \mathbf{G}(A_1, \ldots, A_k)$ with π a permutation of $(1, \ldots, k)$.
4. Monotonicity: if $A_i \geq B_i$, for all i, then $\mathbf{G}(A_1, \ldots, A_k) \geq \mathbf{G}(B_1, \ldots, B_k)$.
5. Continuity from above: if for all fixed i, $A_i^{(\ell)}$ is a monotonically decreasing sequence of matrices converging to $A_i^{(*)}$ for $\ell \to \infty$, then $\mathbf{G}(A_1^{(\ell)}, \ldots, A_k^{(\ell)})$ converges to $\mathbf{G}(A_1^{(*)}, \ldots, A_k^{(*)})$.
6. Congruence invariance: for all invertible matrices $S \in \mathbb{R}^{n \times n}$,
$$\mathbf{G}(S^H A_1 S, \ldots, S^H A_k S) = S^H \mathbf{G}(A_1, \ldots, A_k) S.$$
7. Joint concavity:
$$\begin{aligned} &\mathbf{G}(\lambda A_1 + (1-\lambda)B_1, \ldots, \lambda A_k + (1-\lambda)B_k) \\ &\geq \lambda \mathbf{G}(A_1, \ldots, A_k) + (1-\lambda)\mathbf{G}(B_1, \ldots, B_k), \qquad 0 < \lambda < 1. \end{aligned}$$
8. Invariance under inversion: $\mathbf{G}(A_1^{-1}, \ldots, A_k^{-1}) = (\mathbf{G}(A_1, \ldots, A_k))^{-1}$.
9. Determinant equality: $\det \mathbf{G}(A_1, \ldots, A_k) = (\det A_1 \cdots \det A_k)^{\frac{1}{k}}$.
10. Arithmetic-Geometric-Harmonic inequality:
$$\frac{1}{k} \sum_{i=1}^{k} A_i \geq \mathbf{G}(A_1, \ldots, A_k) \geq \left(\frac{1}{k} \sum_{i=1}^{k} A_i^{-1} \right)^{-1}.$$

Unfortunately, these properties do not result in a unique definition for the geometric mean. For the case of two matrices, however, the geometric mean is uniquely defined from properties 1 to 10 and is given by the following expressions [21]

$$\mathbf{G}(A, B) = A(A^{-1}B)^{1/2} = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}. \tag{2.2}$$

An intuitively attractive interpretation for this result can be obtained from the manifold structure of the set of PD matrices $\mathcal{P}_n$. Specifically, the association of such a manifold with a smooth inner product results in *Riemannian manifold* with its corresponding geometry (Section 3.3.2). Besides the classical Euclidean geometry, $\mathcal{P}_n$ can also be endowed with its so-called natural geometry, for which the intrinsic distance between $A, B \in \mathcal{P}_n$ is given by

$$\delta^{(\mathrm{pd})}(A, B) = \left\| \log(A^{-1/2} B A^{-1/2}) \right\|_F, \tag{2.3}$$

with $\| \cdot \|_F$ the Frobenius norm. Using this distance measure, the geodesic between A and B, i. e., the curve of shortest distance on the manifold between A and B, can be written as

$$\begin{aligned} \gamma^{(\mathrm{pd})}(t) &= A(A^{-1}B)^t = A^{1/2}(A^{-1/2}BA^{-1/2})^t A^{1/2} \\ &= A \#_t B, \quad t \in [0, 1]. \end{aligned} \tag{2.4}$$

This shows that the geometric mean is exactly the midpoint on the geodesic,

$$\mathbf{G}(A, B) = \gamma^{(\mathrm{pd})}(1/2) = A \#_{1/2} B.$$

The subscript in the last term is often dropped when $t = 1/2$.

2.3 Geometric means based on planar approaches

While the properties in the ALM list result in an explicit, unique definition for the geometric mean of two matrices, this is not the case when dealing with more matrices. Considering the simplified case of a space with planar Euclidean geometry, the arithmetic mean of three matrices is the centroid of the triangle they form. Various intuitively appealing techniques to determine this centroid have been generalized to the non-planar (non-Euclidean) geometry of $\mathcal{P}_n$ [6, 25, 28, 97, 103], where the arithmetic mean is naturally generalized to the geometric mean. In this section, we discuss the geometric interpretation of the ALM [6], NBMP [25, 97] and CHEAP [28] mean and end with a comparison of these algorithms.

It has been shown by Poloni [103] that these existing matrix geometric means can be combined into new ones. However, he continues to prove that for more than four matrices, no computational advantage can be obtained from the combination techniques, hence we simply mention the possibility without further discussion.

2.3.1 ALM mean

The ALM mean [6] is a geometric mean which, as the name suggests, satisfies the desired properties enumerated in the ALM list. When taking the ALM mean of k matrices, recursion is used to define the iterations in which we replace $\left(A_1^{(j)}, ..., A_k^{(j)}\right)$ by

$$\left(A_1^{(j+1)}, ..., A_k^{(j+1)}\right) = \left(\mathbf{G}_{ALM}((A_i^{(j)})_{i\neq 1}), ..., \mathbf{G}_{ALM}((A_i^{(j)})_{i\neq k})\right),$$

where $\mathbf{G}_{ALM}$ denotes the recursively defined ALM mean of $k-1$ matrices with the known geometric mean of two matrices (2.2) as its base. All terms in these iterations are proven to converge towards the same limit [6] and in Figure 2.1(a), a planar simplification of this algorithm for three matrices is depicted.

2.3.2 NBMP mean

The NBMP mean [25, 97], just as the ALM mean, satisfies all properties in the ALM list. To compute the NBMP mean of k matrices, we use recursion to define the iterations in which we replace $\left(A_1^{(j)}, ..., A_k^{(j)}\right)$ by $\left(A_1^{(j+1)}, ..., A_k^{(j+1)}\right)$, which equals

$$\left(A_1^{(j)} \#_{\frac{k-1}{k}} \mathbf{G}_{NBMP}((A_i^{(j)})_{i\neq 1}), ..., A_k^{(j)} \#_{\frac{k-1}{k}} \mathbf{G}_{NBMP}((A_i^{(j)})_{i\neq k})\right),$$

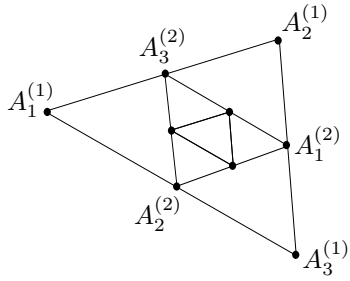
where $\mathbf{G}_{NBMP}$ denotes the recursively defined NBMP mean of $k-1$ matrices, with the geometric mean of two matrices (2.2) as its base. The notation from (2.4) was used to denote the point on the geodesic representing the weighted mean of the terms involved. All terms in these iterations are again proven to converge towards the same limit [25] and in Figure 2.1(b) we show a simplified representation of how the algorithm operates on three matrices. Note that while this planar representation reaches the centroid of the triangle in one step, the lack of matrix commutativity causes the need for iterations. For example, for three PD matrices A_1, A_2, and A_3, we find numerically that

$$A_1 \#_{\frac{2}{3}} \left(A_2 \#_{\frac{1}{2}} A_3\right) \neq A_2 \#_{\frac{2}{3}} \left(A_3 \#_{\frac{1}{2}} A_1\right),$$

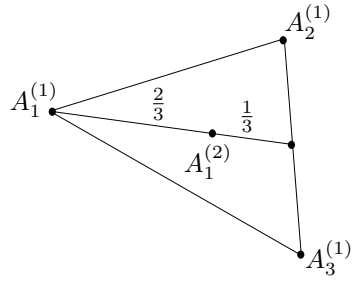
indicating that the results are not the same (after a single step).

2.3.3 General class

We have encountered two means, both satisfying all properties in the ALM list, but yielding different results, as shown in the next example.



(a) The ALM mean.



(b) The NBMP mean.

Figure 2.1: Simplified representations of the algorithms for three matrices.

Example 2.1. If we consider the matrices

$$\begin{bmatrix} 25 & 4 \\ 4 & 1 \end{bmatrix}, \begin{bmatrix} 20 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 20 \end{bmatrix},$$

the results for the ALM and NBMP algorithm are respectively

$$\begin{bmatrix} 7.6943 & 0.9919 \\ 0.9919 & 2.0528 \end{bmatrix} \text{ and } \begin{bmatrix} 7.7139 & 0.9719 \\ 0.9719 & 2.0425 \end{bmatrix},$$

which clearly shows that the results differ.

In fact, it is shown by Bini, Meini, and Poloni [25] and Lim [87] that the ALM and NBMP mean are two instances of an entire class of means, all satisfying the required ALM properties but with possibly different results. For k matrices this *general mean* $\mathbf{G}_{s_1,\ldots,s_{k-1}}$ depends on $k-1$ parameters $(s_1,\ldots,s_{k-1}) \in (0,1]^{k-1}$ and again recursion is used to define the iterations, in which we replace $\left(A_1^{(j)},...,A_k^{(j)}\right)$ by $\left(A_1^{(j+1)},...,A_k^{(j+1)}\right)$, given by

$$\left(A_1^{(j)} \#_{s_1} \mathbf{G}_{s_2,\ldots,s_{k-1}}((A_i^{(j)})_{i\neq 1}),...,A_k^{(j)} \#_{s_1} \mathbf{G}_{s_2,\ldots,s_{k-1}}((A_i^{(j)})_{i\neq k})\right). \tag{2.5}$$

The recursive dependence of these iterations is visualized in Figure 2.2. At each successive recursion level i, the first of the remaining parameters $(s_i,\ldots,s_{k-1})$ is used as a weight (as indicated on the branches), resulting in the weighted geometric mean $A_p \#_{s_{k-1}} A_q$ $(p,q=1,\ldots,k)$ at the foundation of the recursion. Note that one iteration of the means at any recursion level requires the computation until convergence of all means at the level below.

All terms in the iteration (2.5) converge to the same limit, except when $s_{k-1}=1$, since this will result in the computation of the mean $\mathbf{G}_{1,\ldots,1}$ at some recursion

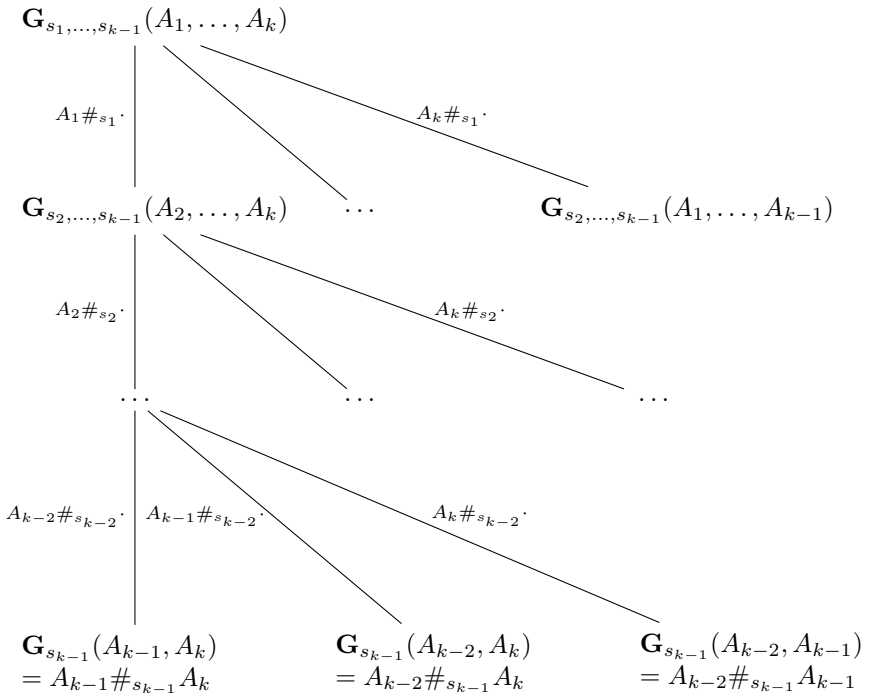


Figure 2.2: Structural diagram of the recursion levels of the general class of means

level. In the iterations of this mean, the order of the matrices is simply permuted, and the separate terms do not converge.

For the ALM and NBMP mean, the parameters $(s_1,\ldots,s_{k-1})$ become respectively $(1,1,\ldots,1,1/2)$ and $((k-1)/k,(k-2)/(k-1),\ldots,1/2)$. This illustrates that for a general number of matrices the geometric mean is not uniquely defined, not even starting from the ten desired properties. In Chapter 3, we investigate the Karcher mean, which also satisfies all properties but has a more appealing analogy with the arithmetic mean.

2.3.4 CHEAP mean

The CHEAP mean [28], unlike the previous algorithms, is no longer recursively defined. It also no longer satisfies all properties present in the ALM list, but as we will notice later, this will be compensated by its cheap computational

cost. Although the mean is only an approximation to a geometric mean, since not all desired properties are satisfied, we still opted to discuss it because of its appealing interpretation as a planar approach. The underlying idea is again computing the centroid of a triangle (with vertices A, B, and C, see Figure 2.3) by the formula

$$A + \frac{1}{3}\big((B - A) + (C - A)\big).$$

The expression above can be interpreted as a step in a Euclidean space from vertex A, in the direction of the tangent vector $\frac{1}{3}\left((B - A) + (C - A)\right)$, which is the arithmetic mean of the directions of vertex A to the three vertices A, B, and C (where direction $A - A$ is trivially omitted).

This notion of a direction is naturally linked to that of a geodesic, or a path of shortest distance. In a Euclidean setting, such a path between vertices A and B is described by $\gamma(t) = A + t(B - A)$, of which the corresponding direction at vertex A can readily be obtained as $B - A$. In the setting of the natural geometry of the manifold of PD matrices, this geodesic from vertex A to vertex B was given by (2.4). The direction of this path at A can be expressed by the tangent vector $A\log(A^{-1}B)$. By again combining the directions of the vertex A to the three vertices A, B, and C, and using the notion of a retraction (Section 3.3.3) to take a step on a manifold, we obtain the expression [28]

$$A\exp\left(\frac{1}{3}\left(\log(A^{-1}B) + \log(A^{-1}C)\right)\right).$$

In the general case of k matrices we replace in each iteration the matrices $(A_1^{(j)}, ..., A_k^{(j)})$ by $(A_1^{(j+1)}, ..., A_k^{(j+1)})$ in which

$$A_i^{(j+1)} = A_i^{(j)} \exp\left(\frac{1}{k}\sum_{\ell=1,\ell\neq i}^{k} \log\left(\left(A_i^{(j)}\right)^{-1} A_\ell^{(j)}\right)\right).$$

We iterate until convergence, although convergence is not always guaranteed for this algorithm, i. e., when the matrices are not sufficiently close to each other (see Bini and Iannazzo [28, Theorem 2.1] for an exact bound). In practice, we do observe convergence of the algorithm in all tests.

For the ALM and NBMP algorithms, the mean of two matrices is by definition known to be the analytical geometric mean, since they are recursively defined, starting with this analytical expression. For the CHEAP mean this consistency is less obvious but it is nonetheless still present. If we examine the CHEAP

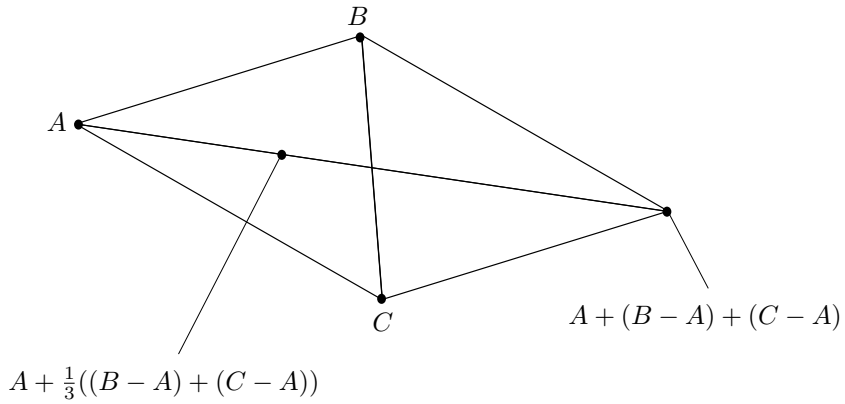


Figure 2.3: Simplified representation of the CHEAP mean for three matrices.

mean of two matrices by applying one iteration of the algorithm, we get

$$A_1^{(0)} \to A_1^{(1)} = A_1^{(0)} \exp\left(\frac{1}{2}\log\left((A_1^{(0)})^{-1}A_2^{(0)}\right)\right) = A_1^{(0)}\left((A_1^{(0)})^{-1}A_2^{(0)}\right)^{\frac{1}{2}},$$

$$A_2^{(0)} \to A_2^{(1)} = A_2^{(0)} \exp\left(\frac{1}{2}\log\left((A_2^{(0)})^{-1}A_1^{(0)}\right)\right) = A_2^{(0)}\left((A_2^{(0)})^{-1}A_1^{(0)}\right)^{\frac{1}{2}},$$

which are two equivalent expressions for the geometric mean of $A_1^{(0)}$ and $A_2^{(0)}$.

2.3.5 Numerical experiments

In Figure 2.4(a), we show the required computational time of all the above algorithms as the number of 30×30 well-conditioned matrices in the mean increases. The random positive definite matrices throughout this thesis are constructed in MATLAB as follows, with n the size of the matrix, k the number of matrices, and f the order of magnitude of the condition number:

```
for i=1:k
  [Q,~]=qr(rand(n)); D=diag([[rand(1,n-1)+1],10^(-f)]);
  A{i}=Q*D*Q';
end
```

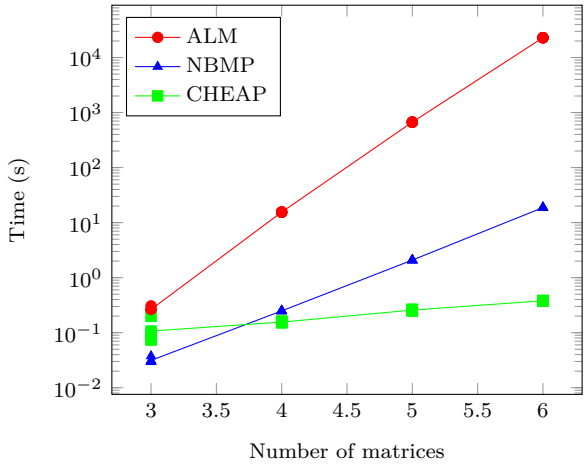
The stopping criterion for all three algorithms is activated when the relative difference between two consecutive iteration points becomes less than a specific tolerance.

While the ALM mean is proven to converge linearly [6] and the NBMP mean superlinearly of order 3 [25], both have rapidly increasing computational time as the number of matrices increases. The number of operations for both algorithms equals $\mathcal{O}(n^3 k! \prod_{i=3}^{k} p_i)$, in which n denotes the size of the matrices, k the number of matrices and p_i the average amount of iterations required to compute the ALM and NBMP mean of i matrices. We observe from experiments that the numbers p_i do not change significantly as the size of the matrices increases. The advantage of the superlinear convergence of the NBMP algorithm over the linear convergence of the ALM algorithm is found in these p_i factors, since they will be much smaller for the first. The problem for both, however, lies in the significant $k!$ factor, which grows tremendously fast as k increases. Despite their performance, it is still interesting to examine these means since they were the first algorithms devised to compute the matrix geometric mean of a general number of matrices.

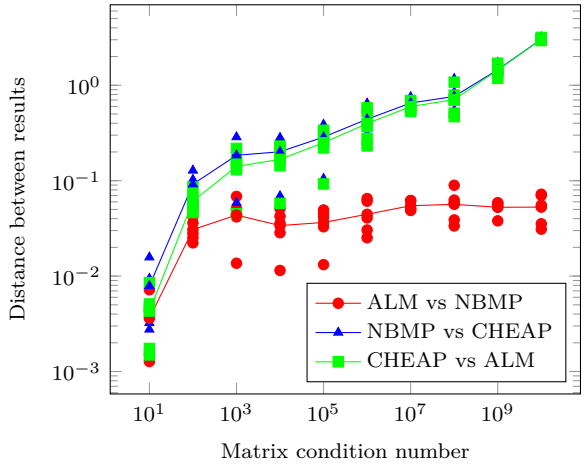
For the CHEAP mean, however, the number of operations equals $\mathcal{O}(n^3 k^2 p_k)$, in which k^2 is a vast improvement over $k!$. Of course, this increased speed of the CHEAP mean comes at a price. It no longer satisfies all properties in the ALM list, and can therefore no longer be considered to be an actual geometric mean. We compare the results of the different algorithms by taking the means of three 30×30 matrices, for which the condition number of each matrix is varied. In Figure 2.4(b), the intrinsic distances (2.3) between the results are shown and it is clear that the ALM and NBMP mean are more similar to each other than to the CHEAP mean, especially as the condition number of the matrices increases. However, the CHEAP mean can still be found in the vicinity of the other means when the condition number of the matrices is acceptable. Combined with its lower computational cost, this vicinity causes the CHEAP mean to be an appealing initial guess for the optimization algorithms in Chapter 3.

A similar figure could be obtained by displaying the classical Euclidean distance between the results, but the distances are consistently smaller than for the intrinsic distance (2.3). This difference is explained by the fact that the intrinsic distance is measured on a curved manifold, while the Euclidean one measures the distance on a straight line in the vector space $\mathcal{H}_n$ of symmetric (Hermitian) matrices, of which $\mathcal{P}_n$ is a submanifold.

The accuracy of the methods is harder to verify since we need a reference solution to compare the results of the algorithms with. As a first test, we construct a set of simultaneously diagonalizable, and hence commuting, matrices. Of these we know the exact geometric mean using the first property in the ALM list, so we can use this as our reference solution. However, the CHEAP mean is shown [28] to converge in one iteration to the exact solution when the matrices commute. Hence, this test is only meaningful for the ALM and NBMP mean, of which we



(a) Required time for different amounts of matrices.



(b) Distance between the different means for the intrinsic measure (2.3). The mean of the samples is indicated by the connecting lines.

Figure 2.4: Computational time and proximity of the different planar algorithms.

show the results in Figure 2.6(a). The relative intrinsic distance

$$\frac{\left\|\log(A^{-1/2}GA^{-1/2})\right\|_F}{\|G\|_F}, \tag{2.6}$$

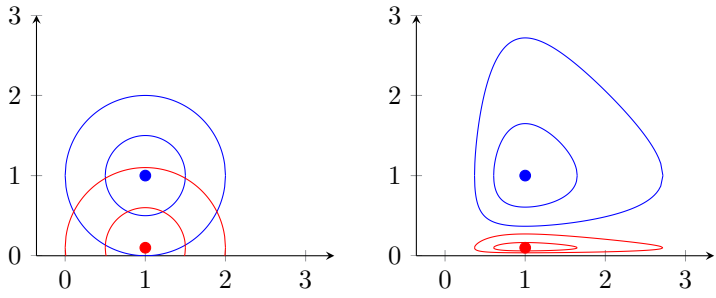
with A the result of one of the algorithms and G the exact solution, is used to display the deviation between both for different condition numbers of the matrices. Recall that the numerator is the intrinsic distance (2.3) between A and G. The accuracy of both algorithms is very similar and deteriorates steadily as the condition number of the matrices increases.

We note that when using the classical Euclidean distance, the deviations are almost at machine precision for all condition numbers. This difference is again related to the curvature of the manifold, but also to the construction of our test data. The conditioning of the matrices was deteriorated by introducing small eigenvalues. On the manifold of PD matrices, this corresponds to working with elements close to the edge of the manifold, where the intrinsic distance becomes larger because of the curvature of the manifold. To illustrate, we display an example of geometric balls for the Euclidean geometry and the natural geometry of $\mathcal{P}_n$ in Figure 2.5. The balls for the natural geometry are compressed as they move closer to the edge of the manifold, confirming that the intrinsic distance on the manifold becomes much larger than the straightforward Euclidean distance.

For matrices which are not simultaneously diagonalizable, the geometric mean is not uniquely defined, hence we need a different model solution. A high precision version of each of the algorithms is designed using the `vpa` functionality of MATLAB with 32 digits of accuracy. Even combined with the largest magnitude of the conditioning of the matrices ($\mathcal{O}(10^{10})$), we still expect a fully accurate solution of 16 digits. The relative intrinsic distance (2.6) between results from the original algorithms and the high precision versions is displayed in Figure 2.6(b) (using three random 10×10 PD matrices). Again the steady deterioration of the accuracy can be seen as the condition number of each of the matrices increases. However, it is clear that the CHEAP algorithm is more sensitive to this condition number than the ALM and NBMP algorithms.

2.4 Related means

Throughout the study of the geometric mean, other definitions have been proposed, which were not necessarily inspired by some intuitive planar interpretation. Some of these were based on a transformation of the set of PD matrices, while others were derived from applications.



(a) Euclidean geometry. (b) Natural geometry.

Figure 2.5: Simplified representation of geometric balls for the Euclidean and natural geometry on $\mathcal{P}_2$. The 2×2 matrices are represented using their eigenvalues, and the balls of radius 1 and 0.5 are shown with centers I (blue) and $A = \text{diag}(1, 10^{-1})$ (red).

Although constructed with the geometric mean in mind, most of these related means can only serve as an approximation because they lack some of the properties described in Section 2.2. However, as they were designed to resemble the geometric mean, it is worthwhile to mention some of them.

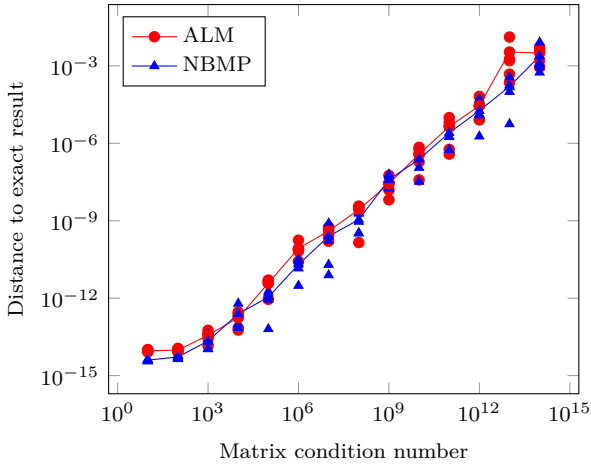
2.4.1 Log-Euclidean mean

A one-to-one relation can be constructed between the manifold of PD matrices $\mathcal{P}_n$ and the vector space of symmetric (Hermitian) matrices $\mathcal{H}_n$ using the matrix logarithm and matrix exponential functions. Moreover, considering the geometry of the sets, the geometric mean is considered the most natural mean on $\mathcal{P}_n$, while the classical arithmetic mean is most natural on the vector space $\mathcal{H}_n$.

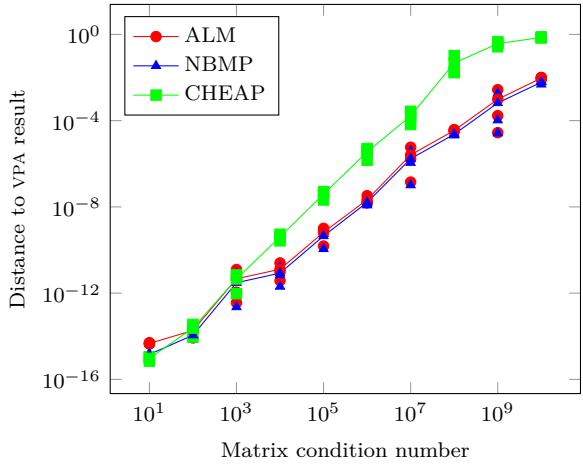
Combined, the *log-Euclidean mean*, or ExpLog mean, of some PD matrices is obtained by mapping all involved matrices to $\mathcal{H}_n$ using the matrix logarithm, taking their arithmetic mean, and mapping the result back to $\mathcal{P}_n$ using the matrix exponential [6, 7, 28]. For matrices $A_1, \ldots, A_k \in \mathcal{P}_n$, we write

$$\mathbf{LE}(A_1, \ldots, A_k) = \exp\left(\frac{1}{k}\sum_{i=1}^{k} \log\left(A_i\right)\right). \tag{2.7}$$

However, this mean lacks some of the desired properties for a geometric mean, such as monotonicity and congruence invariance. Moreover, the log-Euclidean



(a) Accuracy for different condition numbers of the three simultaneously diagonalizable matrices.



(b) Accuracy for different condition numbers of the three random matrices. The mean of the samples is indicated by the connecting lines.

Figure 2.6: Accuracy of the different planar algorithms.

mean of two matrices is different from the geometric mean of two matrices (2.2). Finally, it can be shown that the results of the planar approaches from the previous section are positioned very close to each other, while the log-Euclidean mean is located far from these means.

2.4.2 Symmetric function means

We discuss two more means, which are derived based on an electrical networks approach [5] and are referred to as the *T-mean* and *P-mean*. Each mean is associated with its own positive functions $T_{i,k} : \mathcal{P}_n^k \to \mathcal{P}_n$ and $P_{i,k} : \mathcal{P}_n^k \to \mathcal{P}_n$, $i = 1, \ldots, k$, where the parameter k is equal to the number of matrices in the mean. For positive definite matrices $A_1, \ldots, A_k$, an iterative process is initialized by $\left(A_1^{(0)}, \ldots, A_k^{(0)}\right) = (A_1, \ldots, A_k)$ and defined by

$$A_i^{(j+1)} = T_{i,k}(A_1^{(j)}, \ldots, A_k^{(j)}), \quad i = 1, \ldots, k,$$

for the T-mean, and similar for the P-mean. The matrices in the k-tuples will converge to the same value, which defines the mean.

What makes these functions interesting in our context is that the functions $T_{1,k}$ and $P_{1,k}$ are the same and are equal to the arithmetic mean. Likewise, the functions $T_{k,k}$ and $P_{k,k}$ are both equal to the harmonic mean. Furthermore, each of the functions is constructed using operations reminiscent of the arithmetic and harmonic mean (serial and parallel sums). Considering the Arithmetic-Geometric-Harmonic inequality in Section 2.2 and the upcoming Algorithm 2.5.1, these means also appear closely related to the geometric mean. For the definition of the functions $T_{i,k}$ and $P_{i,k}$ we refer to Anderson, Morley, and Trapp [5].

2.4.3 Power means

A class of means was presented by Lim and Palfia [89] as a generalization of the scalar power mean

$$\mathbf{P}_t(a_1, \ldots, a_k) = \left(\frac{a_1^t + \cdots + a_k^t}{k}\right)^{\frac{1}{t}},$$

with $a_1, \ldots, a_k > 0$, $t \in [-1, 1] \setminus \{0\}$. Note that $\mathbf{P}_1$ corresponds to the arithmetic mean, while $\mathbf{P}_{-1}$ corresponds to the harmonic mean. Alternatively, the scalar power mean of $a_1, \ldots, a_k$ for power t can be defined as the unique positive

solution x of the equation $x = \frac{1}{k}\sum_{i=1}^{k} x^{1-t}a_i^t$, where the summation terms can be recognized as weighted geometric means of x and a_i.

The generalization towards PD matrices $A_1, \ldots, A_k$ of the *power mean* was defined as the solution X of the matrix equation

$$X = \frac{1}{k}\sum_{i=1}^{k} X\#_t A_i,$$

with $X\#_t A$ as in (2.4). This matrix equation was shown to have a unique PD solution for $t \in (0, 1]$, while for $t \in [-1, 0)$ the definition $\mathbf{P}_{-t}(A_1, \ldots, A_k) = \mathbf{P}_t(A_1^{-1}, \ldots, A_k^{-1})^{-1}$ was used. The arithmetic and harmonic means emerge again for $t = 1$ and $t = -1$ respectively.

These power means satisfy all properties of the geometric mean in an adapted form, including monotonicity and joint concavity. Moreover, it was proven by Lim and Palfia [89] that as t converges to zero, the corresponding power means converge to the Karcher mean (Chapter 3), an important instance of the geometric mean which satisfies all desired properties.

2.5 Approximations based on the two-variable mean

The various properties of the matrix geometric mean make it appealing in many applications. Some applications require only a limited amount of accurate digits. In this section, we experiment with a number of approximations to the geometric mean based on the exact expression of the mean of two matrices and on arithmetic and harmonic mean iterations. Note that we already discussed an approximation of the matrix geometric mean in Section 2.3.4 (the CHEAP mean), which sacrificed some of the properties in the ALM list for computational efficiency.

In Chapter 3, the approximations will be evaluated in terms of accuracy by considering their proximity to the Karcher mean, which we will use as a reference for the matrix geometric mean. Additionally, their appropriateness as an initial guess in the optimization algorithms for the Karcher mean will be evaluated. In this context, the related means in the previous section could also be interpreted as approximations to the Karcher mean.

2.5.1 Arithmetic-harmonic mean for two elements

Previously, we mentioned a link between the geometric mean and iterating the arithmetic and harmonic means. Here, we formalize this link for both scalars and matrices.

Let a_1, a_2 be positive numbers and A_1, A_2 be two PD matrices. Recall that the *arithmetic* and *harmonic mean*, $\mathbf{A}$ and $\mathbf{H}$, and the *geometric mean* $\mathbf{G}$ are given by

$$\mathbf{H}(a_1, a_2) = \left(\frac{a_1^{-1} + a_2^{-1}}{2}\right)^{-1}, \quad \mathbf{H}(A_1, A_2) = \left(\frac{A_1^{-1} + A_2^{-1}}{2}\right)^{-1},$$

$$\mathbf{A}(a_1, a_2) = \left(\frac{a_1 + a_2}{2}\right), \qquad \mathbf{A}(A_1, A_2) = \left(\frac{A_1 + A_2}{2}\right),$$

$$\mathbf{G}(a_1, a_2) = \sqrt{a_1 a_2}, \qquad \mathbf{G}(A_1, A_2) = A_1^{1/2}\left(A_1^{-1/2} A_2 A_1^{-1/2}\right)^{1/2} A_1^{1/2}.$$

By taking the arithmetic and harmonic means of a_1 and a_2 and iteratively reapplying the means to the results, the two numbers will converge to the same value, which coincides with their geometric mean [34, 57].

Algorithm 2.5.1 Arithmetic-harmonic iterations

Let a_1, a_2 be two positive numbers

- **while** $(|a_1 - a_2| > tol)$
  - $b_1 = \mathbf{A}(a_1, a_2)$;
  - $b_2 = \mathbf{H}(a_1, a_2)$;
  - $a_1 = b_1$; $a_2 = b_2$;
- **end while**

Return: $a_1 (= a_2)$, equal to the geometric mean of the original numbers

It has been proven that Algorithm 2.5.1 converges to the geometric mean, for both scalars [34, 57] and matrices [8]. In the next section, we discuss a number of possibilities for its generalization and their performance.

2.5.2 Beyond two variables

In this section, some straightforward, as well as more advanced iterative algorithms to approximate the matrix geometric mean will be proposed. A new

approximation based on the arithmetic-harmonic iterations from the previous section is introduced. Their performance will be compared in the next section.

Inductive mean A very straightforward approach to generalizing the geometric mean of two matrices (2.2) is given by the inductive mean [86, 88, 110]. For matrices $A_1, \ldots, A_k \in \mathcal{P}_n$, it is defined as

$$\begin{aligned} \mathbf{S}_k(A_1, \ldots, A_k) &= \mathbf{S}_{k-1}(A_1, \ldots, A_{k-1}) \#_{\frac{1}{k}} A_k, \qquad \mathbf{S}_1(A_1) = A_1, \\ &= \Big(\Big((A_1 \#_{\frac{1}{2}} A_2) \#_{\frac{1}{3}} A_3 \Big) \cdots \Big) \#_{\frac{1}{k}} A_k, \end{aligned} \tag{2.8}$$

where the notation $A \#_t B$ from (2.4) was used. Surprisingly, this simple mean satisfies all desired properties of a geometric mean except for the permutation invariance. Combined with its fixed expression (k applications of the geometric mean of two matrices), these properties turn the inductive mean into an appealing approximation.

Crude midpoint guess A very simplistic generalization of Algorithm 2.5.1 is one where we reduce any k number of matrices to two from the very start, instead of starting an iterative process. The result is the somewhat crude approximation to a geometric mean presented in Algorithm 2.5.2. However, it is very cheap to compute and can be seen to preserve some of the desired properties of the matrix geometric mean, such as the invariance under inversion. Its value as a mean may be limited, but it is worthwhile to consider this cheap approximation as an initial guess in the optimization algorithms for the Karcher mean (Chapter 3).

Algorithm 2.5.2 Crude midpoint guess

Let $A_1, \ldots, A_k$ be k PD matrices

- $B_1 = \mathbf{A}(A_1, \ldots, A_k)$;
- $B_2 = \mathbf{H}(A_1, \ldots, A_k)$;

Return: $\mathbf{G}(B_1, B_2)$

Circular mean The known expression for the geometric mean of two matrices is an appealing start when approximating the mean of more matrices. One such approach [101] describes a method to generalize any mean of two variables. By imposing an order on the matrices and iteratively updating them by their mean

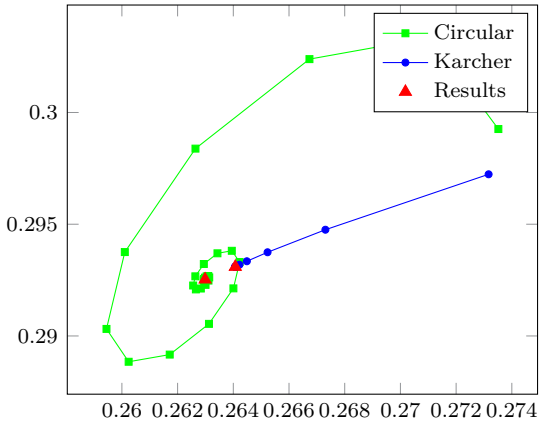


Figure 2.7: Eigenvalue representation of the first matrix in the iterations of the Circular mean (Algorithm 2.5.3) and of the iterations of the Karcher mean (see Chapter 3).

with the next matrix in line, an effective and convergent procedure is obtained, which is presented in Algorithm 2.5.3. Note that for three matrices the Circular mean coincides with the ALM mean (Section 2.3.1).

When this procedure was first proposed by Palfia [100], a conjecture was made that this mean was possibly converging to the Karcher mean (an instance of the geometric mean, see Chapter 3). However, this was later proven to be false, as shown in Figure 2.7.

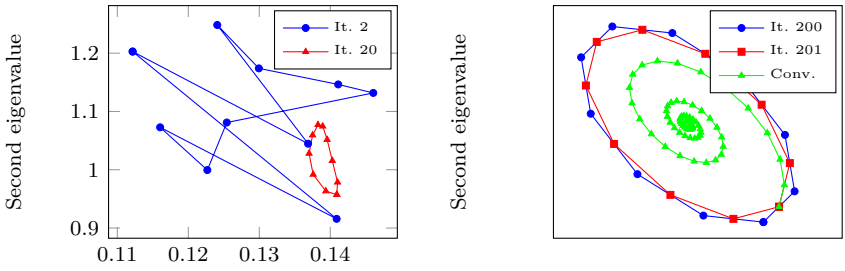
Algorithm 2.5.3 Circular mean

Let $A_1, \ldots, A_k$ be k PD matrices

- **while** (not converged)
  - For $i = 1, \ldots, k$, set $B_i = \mathbf{G}(A_i, A_{(i \mod k)+1})$;
  - For $i = 1, \ldots, k$, set $A_i = B_i$;
- **end while**

Return: Circular mean

Although the algorithm neatly converges to an approximate mean, its convergence slows down quickly because of an interesting phenomenon. It has been proven [52] that an iteration such as Algorithm 2.5.3, which operates on the corners and edges of a random polygon, converges to an *elliptic setting*.



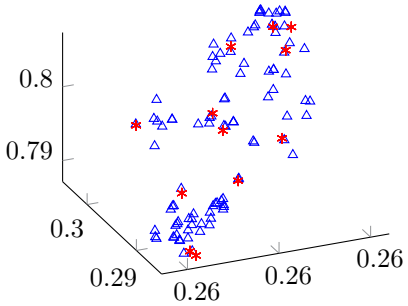
(a) Iteration 2 and 20.

(b) Iteration 200, 201, and the convergence of one of the matrices.

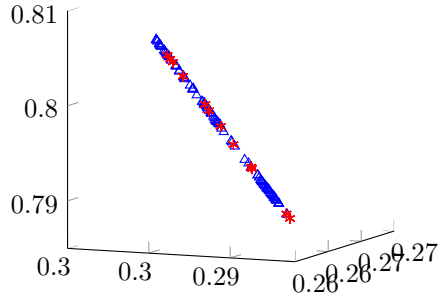
Figure 2.8: The evolution of the eigenvalues of ten 2×2 matrices over the iterations of the Circular mean (Algorithm 2.5.3).

In our context, this random polygon is a more general construction, since it has the matrices entered in the Circular mean as vertices and the geodesics (2.4) between two successive matrices as edges. Once such an elliptic setting is reached, size reduction of the ellips, and hence convergence, becomes significantly slow. Let us graphically illustrate this behavior when considering Algorithm 2.5.3. The convergence towards the elliptic behavior is displayed in Figure 2.8 where the k-tuple of matrices in each iteration is represented by a polygon with the matrix eigenvalues as vertices. Figure 2.8(b) clearly shows this slow behavior with iterations 200 and 201. Another indication of the elliptic phenomenon is the spiraling convergence of the Circular mean in Figure 2.7 and 2.8(b), which represents the evolution of the eigenvalues of the first matrix in the k-tuple throughout the iterations of the Circular mean.

One straightforward, but effective solution to the slow convergence of the Circular mean is the permutation of the matrices during each iteration. This *randomized version* of the Circular mean is presented in Algorithm 2.5.4. The randomization introduced into the algorithm will cause it to give varying results. Moreover, we note that the fixed order Circular mean (Algorithm 2.5.3) is dependent on the order of the matrices as well, of which we show an example in Figure 2.9. In fact, Figure 2.9(a) presents the results of applying the Circular mean to all possible permutations of the order of the matrices, combined with a number of results from the randomized Circular mean. As can be seen, the distribution of the randomized results is very similar to the distribution of the fixed order results. Another interesting observation is that all the results, from both the randomized and fixed order algorithm, appear in a plane (Figure 2.9b), as has been proven for fixed order results by Elmachtoub and Van Loan [52].



(a) The results of the randomized algorithm and the results of the fixed order algorithm have a similar distribution.



(b) The results (both randomized and fixed order) lie within the plane mentioned by Elmachtoub and Van Loan [52].

Figure 2.9: Various results of the Circular mean (Algorithms 2.5.3 and 2.5.4) for the same ten 2×2 matrices. The results of the fixed order algorithm ($*$) are indicated for all initial permutations of the matrix order, along with the results of 100 runs of the randomized algorithm ($\triangle$). The two figures display the same results, but from a different angle of view.

Hence the results of the randomized Circular mean behave very similar to the results of the fixed order version, while the randomized computation results in rapid convergence.

Algorithm 2.5.4 Randomized Circular mean

Let $A_1, \ldots, A_k$ be k PD matrices

- **while** (not converged)
  - For $i = 1, \ldots, k$, set $B_i = \mathbf{G}(A_i, A_{(i \mod k)+1})$;
  - For $i = 1, \ldots, k$, set $A_{\pi(i)} = B_i$, with π a random permutation of $(1, \ldots, k)$;
- **end while**

Return: Randomized Circular mean

A harmonic and arithmetic circular iteration In the original Algorithm 2.5.1, arithmetic and harmonic means were combined to obtain the geometric mean of two matrices. Later, we discussed the Circular mean, which continuously used the geometric mean of two matrices to obtain an approximate mean for

more matrices. By combining these, we obtain Algorithm 2.5.5, which acts as a more advanced generalization of the algorithm for two matrices and to which we will refer as the *HA mean*. We start by duplicating the set of matrices, one focusing on the harmonic and one on the arithmetic mean. As with the Circular mean, an order is imposed of the matrices, after which each matrix is iteratively updated by its harmonic (or arithmetic) mean with the next matrix in line in the complementary set.

When experimenting with Algorithm 2.5.5, we notice that the vertices of the two sets B_i and C_i converge fast towards each other. Afterwards, the nearly equal sets tend to exhibit the elliptic behavior we noticed for the Circular mean, resulting in a slow convergence. As before, we try to resolve this using a random permutation of the order of the matrices (see Algorithm 2.5.4). These results display the same behavior we noticed for Algorithms 2.5.3 and 2.5.4, meaning that the results of the randomized algorithm all appear in the same plane and in the vicinity of the fixed order results.

Algorithm 2.5.5 HA mean

Let $A_1, \ldots, A_k$ be k PD matrices

- For $i = 1, \ldots, k$, set $B_i = A_i$ and $C_i = A_i$;
- **while** (not converged)
  - For $i = 1, \ldots, k$, set $\tilde{B}_i = \mathbf{H}(B_i, C_{(i \mod k)+1})$;
  - For $i = 1, \ldots, k$, set $\tilde{C}_i = \mathbf{A}(B_i, C_{(i \mod k)+1})$;
  - For $i = 1, \ldots, k$, set $C_i = \tilde{C}_i$, $B_i = \tilde{B}_i$.
- **end while**

Return: HA mean

2.5.3 Numerical experiments

We would like to test the performance of these various approximations to the geometric mean, but for now, we will focus only on the computational speed of the algorithms. In Section 3.6 of the next chapter, we will continue our evaluation of these approximations by investigating their proximity to the Karcher mean and their usefulness as an initial guess in the optimization algorithms. We include the log-Euclidean mean (2.7) in our discussion, as this is a popular approximation of the geometric mean given by an explicit expression.

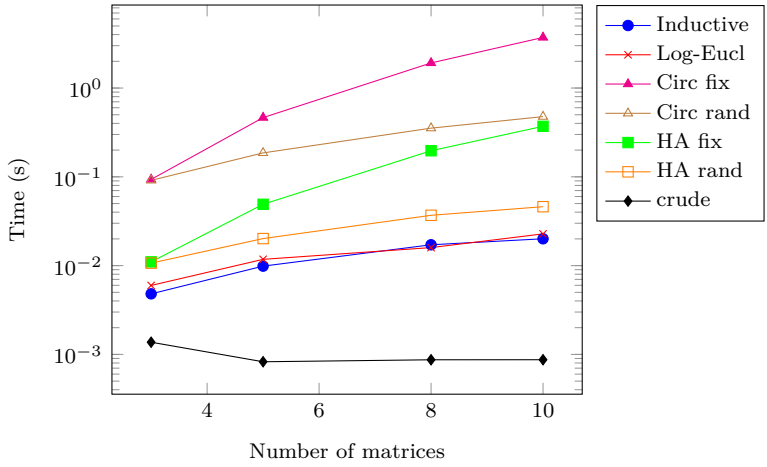


Figure 2.10: Computational time of the approximations to the geometric mean for a varying number of 10×10 matrices. The figure displays only an average over 20 runs of such timings to avoid clutter. In the legend, **Inductive** indicates the inductive mean (2.8), **Log-Eucl** the log-Euclidean mean (2.7), **Circ fix** and **Circ rand** the fixed order and randomized Circular means (Algorithms 2.5.3 and 2.5.4), **HA fix** the mean in Algorithm 2.5.5 with **HA rand** the randomized version, and **crude** the one in Algorithm 2.5.2.

In these experiments, random positive definite matrices were constructed as described in Section 2.3.5.

Computational speed We start by examining the performance of the fixed order and randomized form of the Circular mean and HA mean. As expected, when applying a randomization to the order of the matrices during each iteration, the average computational time reduces as the number of matrices increases. This can be observed in Figure 2.10, where the fixed order and randomized algorithms are indicated by **Circ fix**, **HA fix** and **Circ rand**, **HA rand**, respectively. Note that for three matrices, there is no difference in computational time (and result) since permuting the order of the matrices can not change the (two) neighbors adjacent to each matrix.

When comparing the Circular and HA mean themselves, we observe a significantly lower computational time for the HA mean. Figure 2.10 displays this difference for both the fixed order and the randomized versions of the algorithms.

Since both the inductive and log-Euclidean mean are given by an explicit expression, their computational cost is fixed for a given number of matrices. However, the required matrix operations are more involved than those present in the crude midpoint guess from Algorithm 2.5.2, explaining the difference in computational time. This crude midpoint guess itself has a low computational time, but as we will see in the next chapter, it also has a bad proximity to the Karcher mean.

Summary of the interaction with the Karcher mean For convenience, we indicate the most important conclusions when comparing the approximations with the Karcher mean. For a full discussion, we refer to Section 3.6.

In terms of proximity to the Karcher mean, the Circular and HA mean tend to give good results, for both their fixed order and randomized versions. The inductive mean also displays a decent proximity, while the results of the crude midpoint guess and the log-Euclidean mean are located farther from the Karcher mean.

When the approximations are used as an initial guess in the optimization algorithms for the Karcher mean, many of them result in a similar number of iteration steps. Most noticeable is the increased number of required iterations when starting from the arithmetic mean, harmonic mean, and crude midpoint guess.

2.6 Conclusions

In this chapter, various techniques to compute a matrix geometric mean or an approximation thereof were presented.

The ALM mean, NBMP mean, and the other planar approaches to define the geometric mean of more than two matrices already demonstrated the lack of a unique definition. Hence, as will be further discussed in the next chapter, there will not be one correct matrix geometric mean, but rather a most natural one.

Similar to the exact expression for the geometric mean of two matrices, its computation using harmonic and arithmetic mean iterations did not have a straightforward generalization. We have examined some existing approaches and newly developed algorithms, judging their performance based on computational speed. For the Circular and HA mean, we noticed a similar distribution of the results for both the fixed order and randomized order versions, while the second ones provided a significant decrease in the number of required iterations and consequently, in computational time. Combined with the results in Section

3.6 of the next chapter, the inductive mean and randomized version of the HA mean appear to give optimal results as geometric mean approximations.

Chapter 3

The Karcher mean and Riemannian optimization on positive definite matrices

In this chapter, a specific instance of the matrix geometric mean, referred to as the Karcher mean, least squares mean, etc., is investigated in further detail. Different algorithms for the computation of this mean, both previously existing and newly developed, are provided and compared in experiments.

We exploit the Riemannian geometry of the set of positive definite matrices extensively in the development of the algorithms. A general version of all the required Riemannian constructions is presented and the definitions are applied to the Euclidean and natural geometry of the positive definite matrices. For a more complete discussion, we refer to Boothby [33] and Lee [84] on the topic of Riemannian geometry and to Absil, Mahony, and Sepulchre [1] concerning Riemannian optimization.

Next, the discussion of the approximate means of the previous chapter is continued and a practical application of the Karcher mean is examined.

The content of this chapter was published as a part of

> Jeuris, B., Vandebril, R., and Vandereycken, B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis 39*, 1 (2012), pp. 379–402.

Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Protein fold recognition using geometric kernel data fusion. *Bioinformatics 30*, 13 (2014), pp. 1850–1857.

3.1 Introduction

Many instances of the matrix geometric mean have been discussed in the previous chapter, except for its perhaps most important instance, the Karcher mean. It has been shown to be a proper geometric mean, satisfying all properties in the ALM list (Section 2.2). Monoticity, for example, is a property only recently proven for the Karcher mean [23, 83].

This mean is also often referred to as the least squares mean, a name which originates from its definition as the minimizer of the sum of squared (intrinsic) distances on the manifold of positive definite (PD) matrices $\mathcal{P}_n$. To obtain this minimizer, we use Riemannian optimization, a generalization of classical optimization techniques towards manifolds.

This chapter is organized in the following way. In Section 3.2, we start with a formal definition of the Karcher mean. We provide a basic framework for the Riemannian optimization based on Absil et al. [1] in Section 3.3, which we apply immediately to the set of PD matrices $\mathcal{P}_n$. The use of a different inner product is also considered and a new, explicit expression for the Riemannian Hessian of the Karcher cost function (3.1) is introduced. Afterwards, we apply the Riemannian BFGS method to the optimization problem, which is unprecedented for the computation of the Karcher mean. The resulting first- and second-order optimization techniques are discussed and compared in Section 3.4 and 3.5. Section 3.6 continues the discussion of the approximate means of the previous chapter and we conclude the chapter with an application of the Karcher mean in Section 3.7.

3.2 Definition

As mentioned in the previous chapter, the properties in the ALM list fail to specify a unique definition for the matrix geometric mean. The ALM and NBMP mean are only two examples of a general class of means satisfying all these properties (Section 2.3.1–2.3.3). Another mean that satisfies all the necessary

properties is the *Karcher mean*, which is defined as the minimizer

$$\mathbf{K}(A_1, ..., A_k) = \arg\min_{X \in \mathcal{P}_n} \sum_{i=1}^{k} \delta^2(A_i, X),$$

where $\mathcal{P}_n$ represents the set of PD matrices, $\delta := \delta^{(\mathrm{pd})}$ is the intrinsic distance on this manifold as given in (2.3) and $A_i \in \mathcal{P}_n$ are the matrices of which we want to find the Karcher mean. In terms of an optimization problem this translates to a cost function f, given by

$$f(X; A_1, \ldots, A_k) = \sum_{i=1}^{k} \left\| \log(A_i^{-1/2} X A_i^{-1/2}) \right\|_F^2. \tag{3.1}$$

When the involved matrices $(A_1, \ldots, A_k)$ are clear from context, we will simplify the notation of the cost function to $f(X)$. For this mean to be well-defined, the minimizer of the cost function should be unique. When the manifold $\mathcal{P}_n$ is endowed with its natural inner product (3.3) (Section 3.3.2), the cost function f is strictly geodesically convex, which is a generalization of the classical convexity as follows: let $X, Y \in \mathcal{P}_n$, and $t \in [0, 1]$, then

$$f(X \#_t Y) \leq (1 - t) f(X) + t f(Y),$$

with $X \#_t Y$ as in (2.4). Combining this with the convexity of the set $\mathcal{P}_n$ itself, the minimizer can be proven to be unique [21, 30].

The uniqueness of this minimizer can also be verified in a different manner. It is known [36, 85] that $\mathcal{P}_n$ with the natural inner product (3.3) forms a Cartan–Hadamard manifold, which is a complete and simply connected Riemannian manifold with non-positive sectional curvature everywhere. On this type of manifold, the so-called Riemannian center-of-mass (or barycenter), which in this case is exactly the Karcher mean, is known to be unique [37, 76].

The Karcher mean satisfies all properties in the ALM list, of which the monotonicity has been proven only recently [23, 83]. Moreover, this mean is considered to be the most natural generalization of the geometric mean because of its analogy with the arithmetic mean, which can be defined as a similar minimizer by using the standard Euclidean distance.

Since the Karcher mean is computed iteratively as the solution of an optimization problem, we require a good initial guess. We will use the CHEAP mean in our experiments, since it possesses good computational speed and reasonable accuracy.

3.3 Differential geometry

Calculating the Karcher mean involves solving an optimization problem on a manifold, which requires a more general approach than in the traditional environment of vector spaces. We need to introduce some new concepts to perform this generalization, but we only briefly discuss these matters here; for a more thorough discussion of the subject we refer to any introductory book on differential geometry, such as Boothby [33] or Lee [84], and to Absil et al. [1] for the optimization perspective. After the introduction, we will use these generalized concepts to implement a number of optimization techniques, more specifically, the steepest descent, conjugate gradient, BFGS, and Newton algorithms. The type of generalized optimization used here is often referred to as retraction-based optimization [1, 2], indicating that the concept of retractions (Section 3.3.3) lies at the foundation of these techniques.

The general concepts discussed here were found in Absil et al. [1], and many of these structures were already derived for $\mathcal{P}_n$ endowed with its natural geometry [53, 95, 100, 109]. We added an explicit expression for the Levi–Civita connection on this manifold (and consequently for the Riemannian Hessian) and a derivation of all these structures for the manifold endowed with the inner product inherited from $\mathcal{H}_n$.

3.3.1 Manifold and tangent space

So far we have referred to the set of PD matrices $\mathcal{P}_n$ as a manifold. Intuitively, a *manifold* is defined as a set which can locally be mapped one-to-one to $\mathbb{R}^d$ (where d is the dimension of the manifold). In order to get a smooth (C^∞) manifold, we also require these mappings, or charts, to transition smoothly onto each other in case their domain overlaps (Section 1.2.3). The space $\mathcal{P}_n$ is well-known as a smooth manifold [93, 121], and can be represented using a single chart. It can also be seen as an open submanifold of $\mathcal{H}_n$, which is referred to as the *enveloping* or *embedding space*.

Another important concept is the *tangent space* to a manifold at a certain point, which is basically a first-order (vector space) approximation of the manifold at this point. For $\mathcal{P}_n$, the tangent space at each point X, denoted by $T_X\mathcal{P}_n$, can be identified with the vector space of symmetric (Hermitian) matrices $\mathcal{H}_n$,

$$T_X\mathcal{P}_n \simeq \mathcal{H}_n.$$

Applying a *tangent vector* $\xi_X \in T_X\mathcal{P}_n$ at a point $X \in \mathcal{P}_n$ to a differentiable function $f : \mathcal{P}_n \to \mathbb{R}$ is defined as

$$\xi_X f = \mathrm{D}f(X)[\xi_X],$$

where ξ_X in the right-hand side is simply seen as a symmetric (Hermitian) matrix and $\mathrm{D}f(X)$ denotes the classical Fréchet derivative of f at X. For the cost function f in (3.1), this derivative is given by [21, Theorem 6.3.3]

$$\mathrm{D}f(X)[\xi_X] = 2\sum_{i=1}^{k} \mathrm{tr}\left(X^{-1}\log(XA_i^{-1})\xi_X\right), \tag{3.2}$$

with A_i the matrices of which we want to find the mean and $\mathrm{tr}(\cdot)$ the matrix trace.

A *vector field* is a construction that associates with each point on a manifold a tangent vector in its tangent space. Suppose ξ is a vector field on the manifold $\mathcal{P}_n$ and f is a real-valued function on this manifold, then ξf is again a real-valued function on the manifold defined by

$$\xi f : \mathcal{P}_n \to \mathbb{R} : X \mapsto \xi_X f,$$

with $\xi_X \in T_X\mathcal{P}_n$ the tangent vector associated with the point X.

3.3.2 Inner product and gradient

Gradient-based optimization requires the notions of a gradient and inner product, which will be introduced here for $\mathcal{P}_n$. In fact, we consider two inner products. The first is the inner product most often associated with $\mathcal{P}_n$: for $\xi_X, \eta_X \in T_X\mathcal{P}_n$, we have

$$\langle \xi_X, \eta_X \rangle_X^{(\mathrm{pd})} = \mathrm{tr}(\xi_X X^{-1} \eta_X X^{-1}), \tag{3.3}$$

which leads to the intrinsic distance measure (2.3) and geodesics, or paths of shortest distance, of the form (2.4). We recall these here, for $X, Y \in \mathcal{P}_n$,

$$\delta^{(\mathrm{pd})}(X,Y) = \left\| \log(X^{-1/2} Y X^{-1/2}) \right\|_F,$$

$$\gamma^{(\mathrm{pd})}(t) = X(X^{-1}Y)^t = X^{1/2}(X^{-1/2}YX^{-1/2})^t X^{1/2}.$$

Another benefit of this inner product is that the corresponding geodesics are complete, meaning that any geodesic segment can be extended infinitely. For

the second inner product we take the same one as the enveloping space $\mathcal{H}_n$: suppose again $\xi_X, \eta_X \in T_X\mathcal{P}_n$, then

$$\langle \xi_X, \eta_X \rangle_X^{(\mathrm{sym})} = \mathrm{tr}(\xi_X \eta_X). \tag{3.4}$$

As a consequence, the intrinsic distance and expression of the geodesics become the same as in $\mathcal{H}_n$:

$$\delta^{(\mathrm{sym})}(X, Y) = \|Y - X\|_F, \tag{3.5}$$

$$\gamma^{(\mathrm{sym})}(t) = X + t(Y - X), \tag{3.6}$$

with $X, Y \in \mathcal{P}_n$. These geodesics are no longer infinitely extendable since it is possible for some X, Y, t that the matrix $\gamma^{(\mathrm{sym})}(t)$ in (3.6) is no longer PD and thus not an element of $\mathcal{P}_n$. However, for sufficiently small t, $\gamma^{(\mathrm{sym})}(t)$ is in $\mathcal{P}_n$ and it could prove to be computationally more efficient than the more involved expression $\gamma^{(\mathrm{pd})}$ (2.4). We will refer to inner product (3.3) as the Riemannian inner product and to inner product (3.4) as the Euclidean inner product. Note that these are only two inner products out of infinitely many possibilities, as will be demonstrated in the numerical experiments in Section 3.4.3.

Next, the gradient of a cost function gives the direction of steepest ascent. It can be defined in each point X on a general manifold $\mathcal{M}$ as the tangent vector $\mathrm{grad}\, f(X) \in T_X\mathcal{M}$ such that

$$\langle \mathrm{grad}\, f(X), \xi_X \rangle_X = \mathrm{D}f(X)[\xi_X], \quad \forall \xi_X \in T_X\mathcal{M}. \tag{3.7}$$

Using (3.2) we find for our current setting

$$\mathrm{grad}^{(\mathrm{pd})} f(X) = 2 \sum_{i=1}^{k} X^{\frac{1}{2}} \log(X^{\frac{1}{2}} A_i^{-1} X^{\frac{1}{2}}) X^{\frac{1}{2}}, \tag{3.8}$$

when using the Riemannian inner product (3.3) and

$$\mathrm{grad}^{(\mathrm{sym})} f(X) = 2 \sum_{i=1}^{k} X^{-\frac{1}{2}} \log(X^{\frac{1}{2}} A_i^{-1} X^{\frac{1}{2}}) X^{-\frac{1}{2}}. \tag{3.9}$$

when using the Euclidean one (3.4). Note the slight difference between both in the sign of the power of the outer X-factors.

From the geodesic convexity of f with the Riemannian inner product, we know that the optimization problem, or equivalently the equation $\mathrm{grad}^{(\mathrm{pd})} f(X) = 0$, always admits a unique solution. Since (3.9) is obtained from (3.8) after a full-rank transformation, setting this gradient to zero gives the same unique solution. However, the path taken towards this solution in the optimization algorithms will depend on the chosen inner product and corresponding geometry.

3.3.3 Retraction and vector transport

At some point during an optimization algorithm, one obtains a tangent vector at the current iteration point X which describes the direction in which one wants to take a step towards the next iteration point. Hence we require a mapping, $R_X : T_X\mathcal{P}_n \to \mathcal{P}_n$, called *retraction*, that locally maps $T_X\mathcal{P}_n$ onto the manifold $\mathcal{P}_n$ itself, and which is the generalized operation of taking a step on a manifold. An interpretation of these retractions is taking a unit step along a geodesic, or an approximation thereof, on the manifold in the direction specified by the argument (see Figure 3.1(a)). The condition for such a mapping to be called a retraction is the preservation of the first-order information of the tangent space $T_X\mathcal{P}_n$, meaning that a step of size zero stays at the same point X and the differential of the retraction in this origin (the zero matrix $0_X \in T_X\mathcal{P}_n$) is the identity mapping. We consider three retractions:

$$R_X^{(\mathrm{sym})}(\xi_X) = X + \xi_X, \tag{3.10}$$

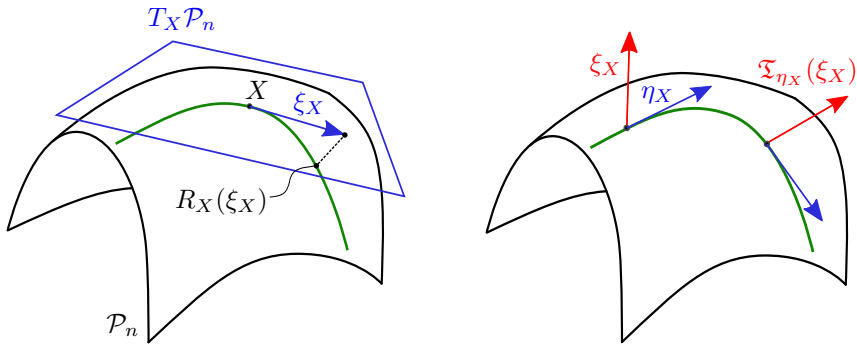
$$R_X^{(\mathrm{pd})}(\xi_X) = X^{\frac{1}{2}} \exp\left(X^{-\frac{1}{2}} \xi_X X^{-\frac{1}{2}}\right) X^{\frac{1}{2}}, \tag{3.11}$$

$$R_X^{(\mathrm{pd}')}(\xi_X) = X + \xi_X + \frac{1}{2}\xi_X X^{-1} \xi_X, \tag{3.12}$$

for $X \in \mathcal{P}_n$, $\xi_X \in T_X\mathcal{P}_n$. The first of these is a unit step along the geodesic (3.6) and can thus be considered to be a natural retraction with respect to the Euclidean inner product (3.4). When the manifold is endowed with the Riemannian inner product (3.3), $R_X^{(\mathrm{sym})}$ is still a valid first-order retraction. As mentioned before, precaution has to be taken to assure that the result of the retraction is still PD. We do this by reducing the step size when necessary. The retraction $R_X^{(\mathrm{pd})}$ is the one that naturally arises when the manifold is endowed with the Riemannian inner product. To see this, recall that the geodesic between $X, Y \in \mathcal{P}_n$ is given by

$$\begin{aligned}\gamma^{(\mathrm{pd})}(t) &= X^{1/2}\left(X^{-1/2} Y X^{-1/2}\right)^t X^{1/2} \\ &= X^{1/2} \exp\left(t \log\left(X^{-1/2} Y X^{-1/2}\right)\right) X^{1/2}, \quad t \in [0,1].\end{aligned}$$

Such a geodesic is completely determined (locally in general, but globally for $\mathcal{P}_n$ [1]) by its starting point and direction, which are given by X and $\xi_X = X^{1/2} \log(X^{-1/2} Y X^{-1/2}) X^{1/2}$, respectively. Now, entering ξ_X into the expression for $\gamma^{(\mathrm{pd})}$ and taking a unit step ($t = 1$), we obtain the retraction given in (3.11). The last retraction is the second-order approximation to $R_X^{(\mathrm{pd})}$,



(a) Retraction. (b) Vector transport.

Figure 3.1: Simplified representations of a retraction and a vector transport.

which can easily be seen by using the identity

$$\exp(X) = I + X + \frac{1}{2}X^2 + \mathcal{O}(X^3), \quad X \to 0.$$

Next, in order to perform, among others, the conjugate gradient algorithm we need to somehow transfer a tangent vector at a point $X \in \mathcal{P}_n$ to the tangent space at another point $Y \in \mathcal{P}_n$. This operation is performed by a *vector transport* (Figure 3.1(b)). We consider two vector transports: for $X \in \mathcal{P}_n, \xi_X, \eta_X \in T_X\mathcal{P}_n$,

$$\mathfrak{T}^{(\mathrm{sym})}_{\eta_X}(\xi_X) = \xi_X, \tag{3.13}$$

$$\mathfrak{T}^{(\mathrm{pd})}_{\eta_X}(\xi_X) = X^{\frac{1}{2}} \exp\left(\frac{X^{-\frac{1}{2}}\eta_X X^{-\frac{1}{2}}}{2}\right) X^{-\frac{1}{2}}\xi_X X^{-\frac{1}{2}} \exp\left(\frac{X^{-\frac{1}{2}}\eta_X X^{-\frac{1}{2}}}{2}\right) X^{\frac{1}{2}}, \tag{3.14}$$

where $\mathfrak{T}^{(.)}_{\eta_X}(\xi_X)$ denotes the vector transport of ξ_X over η_X. The definition of a vector transport [1, definition 8.1.1] states that it needs to be linear in ξ_X and if η_X is the zero element, the vector transport must be the identity mapping. Both these conditions are easily checked for the expressions above. The definition also states that a vector transport has an associated retraction, meaning that the tangent vector $\mathfrak{T}_{\eta_X}(\xi_X)$ should be an element of the tangent space at $R_X(\eta_X)$, for some retraction R_X. Vector transport (3.13) is associated with retraction $R^{(\mathrm{sym})}_X$, since these structures naturally arise when $\mathcal{P}_n$ is endowed with the Euclidean inner product of the enveloping vector space $\mathcal{H}_n$. Such a natural vector transport is often referred to as parallel transport. The structure of

vector transport (3.14) suggests that it is associated with retraction $R_X^{(\mathrm{pd})}$ [53]. Note that it is also possible to find a vector transport associated with $R_X^{(\mathrm{pd}')}$, but we will restrict our attention to the two vector transports mentioned above.

3.3.4 Levi–Civita connection and Riemannian Hessian

Some of our optimization methods require second-order information about the system, which is provided by the Hessian operator. The *Riemannian Hessian* of a real-valued function f at a point X on the manifold is a linear, symmetric mapping from the tangent space into itself, given by

$$\operatorname{Hess} f(X) : T_X\mathcal{P}_n \to T_X\mathcal{P}_n : \xi_X \mapsto \operatorname{Hess} f(X)[\xi_X] = \nabla_{\xi_X} \operatorname{grad} f, \tag{3.15}$$

where ∇ is the so-called *Levi-Civita connection*, which depends on the inner product, hence the Hessian also depends on the inner product.

When endowed with the Euclidean inner product (3.4), the manifold is an open Riemannian submanifold of $\mathcal{H}_n$, which is a vector space. Hence the Levi–Civita connection is given by

$$\nabla_{\zeta_X}^{(\mathrm{sym})}\xi = \mathrm{D}(\xi)(X)[\zeta_X],$$

with ξ a vector field. To compute the Fréchet derivative in the right-hand side, we recall that the vector field ξ assigns to each $X \in \mathcal{P}_n$ a tangent vector $\xi_X \in T_X\mathcal{P}_n \simeq \mathcal{H}_n$. The derivative is performed by considering ξ as a function of X with images in $\mathcal{H}_n$.

For the Riemannian inner product (3.3), however, this connection is more complicated. It can be shown that

$$\nabla_{\zeta_X}^{(\mathrm{pd})}\xi = \mathrm{D}(\xi)(X)[\zeta_X] - \frac{1}{2}\left(\zeta_X X^{-1}\xi_X + \xi_X X^{-1}\zeta_X\right)$$

satisfies all properties of the Levi–Civita connection. A straightforward way to do this, is by checking that it satisfies the Koszul formula [1], which at a point $X \in \mathcal{P}_n$ is given by

$$\begin{aligned} 2\langle \nabla_{\zeta_X}\eta, \xi_X\rangle_X =& \zeta_X\langle\eta,\xi\rangle + \eta_X\langle\xi,\zeta\rangle - \xi_X\langle\zeta,\eta\rangle \\ & - \langle\zeta_X, [\eta,\xi]_X\rangle_X + \langle\eta_X, [\xi,\zeta]_X\rangle_X + \langle\xi_X, [\zeta,\eta]_X\rangle_X. \end{aligned}$$

The actual computation of the Hessian will be discussed separately for each second-order method in Section 3.5.

3.4 First-Order Implementations

We can use all the building blocks of the previous section to assemble a number of optimization methods. In this section, we restrict our attention to first-order optimization algorithms. Some more advanced second-order methods will be discussed later in Section 3.5.

The basic methods in this section have already been derived in various papers [27, 53, 55, 100, 109], both in their standard form and in some approximated way. As an example of these approximated approaches, we found a Richardson-like iteration by Bini and Iannazzo [27], which is based on the standard steepest descent method with the Euclidean inner product (3.3), except a linearization of the natural retraction (3.11) is used. But this approximation is exactly the steepest descent algorithm using retraction (3.10), hence it can still be interpreted as a steepest descent method on the manifold.

3.4.1 Steepest descent method

First, we combine the elements of the previous section into the steepest descent algorithm, which in each iteration takes a step in the direction of $-\operatorname{grad} f(x)$, the direction of steepest descent. The step size is determined using Armijo line search [1], which is a standard backtracking technique, starting from a given step size and iteratively reducing it with a constant factor until an acceptable decrease of the cost function, relative to the corresponding step size, is obtained. Algorithm 3.4.1 contains the steepest descent method when setting the parameter β to 0. This parameter β will be used in the conjugate gradient method to provide a notion of momentum between the iterations. The stopping criteria in the algorithm check whether the Armijo step size or the absolute or relative difference between two consecutive iteration points are smaller than their respective tolerances. We consider all three retractions (3.10), (3.11) and (3.12).

Remark In Chapter 4, we will consider an adaptation of the Karcher mean for matrices containing some additional structure. For the case of linear matrix structures, two steepest descent algorithms are developed, accompanied by a convergence analysis. Since the Karcher mean of unstructured PD matrices arises as a special case from this more general analysis, we present a summary of the results.

- The convergence speed of the steepest descent method based on the Euclidean inner product (3.4) becomes worse as the conditioning of the

matrices A_i, $i = 1, \ldots, k$, in the mean or the conditioning of the matrices $A_i^{-1/2}\mathbf{K}A_i^{-1/2}$ deteriorates, where $\mathbf{K} = \mathbf{K}(A_1, \ldots, A_k)$. Even when all matrices A_i are close to the Karcher mean, the algorithm remains inefficient while some A_i is ill-conditioned.

- The convergence speed of the steepest descent method based on the Riemannian inner product (3.3) only depends on the conditioning of the matrices $A_i^{-1/2}\mathbf{K}A_i^{-1/2}$. Therefore, if the matrices A_i are not too far from each other, a very fast convergence is expected.

For the complete analysis and discussion, we refer to Sections 4.4.1–4.4.3 of the next chapter.

3.4.2 Conjugate gradient method

When the steepest descent method is applied to an ill-conditioned optimization problem, meaning that the Hessian of the cost function at the minimizer is ill-conditioned, a characteristic phenomenon emerges. Near this minimizer, we can find a direction in which the cost function has a steep descent towards the minimizer and another in which the descent is very slow. As a consequence, the steepest descent method mainly follows the steep direction and can suffer from iteration steps which are nearly perpendicular to the slow descent direction. The convergence of the algorithm becomes very slow and is characterized by the iteration steps which move back and forth. We refer to this movement as the zigzag-pattern.

In Figure 3.2(a), we show this typical zigzag-pattern that arises for the steepest descent method. The pattern was activated by taking five 3×3 random matrices for which the smallest eigenvalue is roughly 10^3 times smaller than the others. The conjugate gradient method counteracts this pattern by allowing the new search direction to be influenced by the previous one, introducing a notion of momentum. The amount of influence of the previous search direction in the conjugate gradient algorithm is determined by the β factor, for which we consider three different formulas [98], denoted by $\beta^{(\mathrm{fr})}$ (Fletcher–Reeves), $\beta^{(\mathrm{pr})}$ (Polak–Ribière), and $\beta^{(\mathrm{hs})}$ (Hestenes–Stiefel). Algorithm 3.4.1 shows the details of this method and Figure 3.2(b) displays the improvement of the zigzag-pattern.

To accurately determine its influence, the previous search direction must be transported towards the new iteration point using vector transports. To get the best affinity between the vector transports and the retractions, we work with the natural retractions $R^{(\mathrm{sym})}$ (with corresponding vector transport $\mathfrak{T}^{(\mathrm{sym})}$) and $R^{(\mathrm{pd})}$ (with $\mathfrak{T}^{(\mathrm{pd})}$).

Algorithm 3.4.1 The Karcher mean using the conjugate gradient method

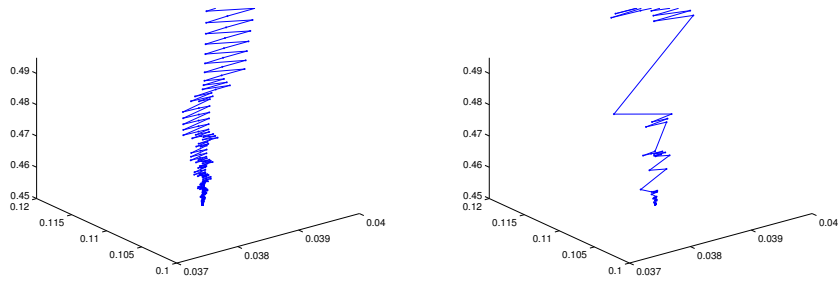
Let $A_1, ..., A_k$, $k > 2$, be PD matrices, X_0 a PD initial guess, $R^{(i)}$ and $\mathfrak{T}^{(i)}$ ($R^{(\mathrm{sym})}$ and $\mathfrak{T}^{(\mathrm{sym})}$ or $R^{(\mathrm{pd})}$ and $\mathfrak{T}^{(\mathrm{pd})}$) a retraction and a vector transport, and $\beta^{(\ell)}$ ($\beta^{(\mathrm{fr})}$, $\beta^{(\mathrm{pr})}$ or $\beta^{(\mathrm{hs})}$ or 0) the given β type.

- $j = 0$;
- $grad_0 = \operatorname{grad} f(X_0)$; $\{f$ is our cost function $\}$
- $\xi_0 = -grad_0$; $\{\xi_j$ is the search direction$\}$
- **while** (not converged)
  - $X_{j+1} = R^{(i)}_{X_j}(t^A \xi_j)$; $\{$with t^A the Armijo step size [1]$\}$
  - $grad_{j+1} = \operatorname{grad} f(X_{j+1})$;
  - $\xi_{old} = \mathfrak{T}^{(i)}_{t^A \xi_j}(\xi_j)$; $\{$vector transport of the old search direction$\}$
  - Determine $\beta^{(\ell)}$ according to the given type; $\{\beta = 0$ corresponds to steepest descent$\}$
  - $\xi_{j+1} = -grad_{j+1} + \beta^{(\ell)} \xi_{old}$;
  - **if** (ξ_{j+1} not a descent direction)
    - $\xi_{j+1} = -grad_{j+1}$;
  - **end if**
  - $j = j + 1$;
- **end while**

Return: Karcher mean $\mathbf{K}(A_1, ..., A_k)$

3.4.3 Numerical experiments

When comparing the overall performance of the steepest descent and conjugate gradient algorithms, we notice that the influence of choosing inner product (3.3) or (3.4) is far greater than the impact of the chosen retraction or, in case of the conjugate gradient algorithm, the β type. In fact, when only varying the retraction and β type, the results are all very similar. The speed-up of the conjugate gradient technique over steepest descent is also hardly noticeable in general, which is explained by the presence of a sufficiently good initial guess, the CHEAP mean. When this initial point is sufficiently close to the solution of the problem, the cost function will behave nicely in this neighbourhood and the zigzag-pattern mentioned before is less likely to occur.



(a) Using steepest descent.

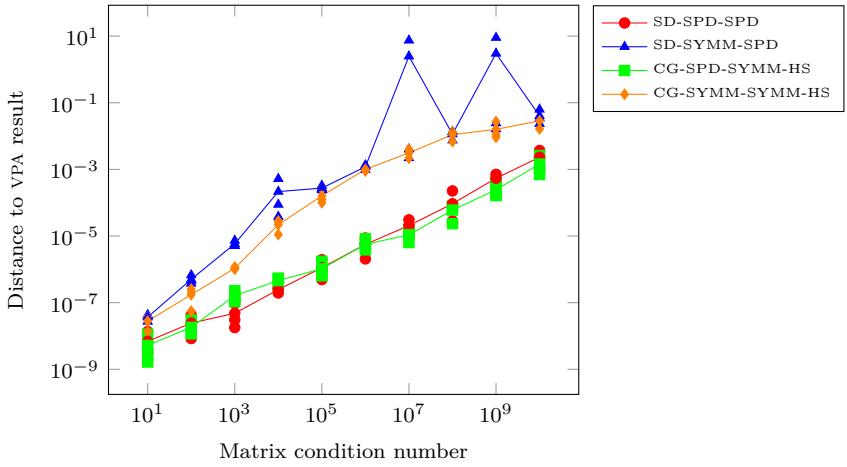
(b) Using conjugate gradient.

Figure 3.2: Evolution of the eigenvalues of the consecutive iteration points in the computation of the Karcher mean of five 3×3 matrices for which the zigzag-pattern appears for steepest descent.

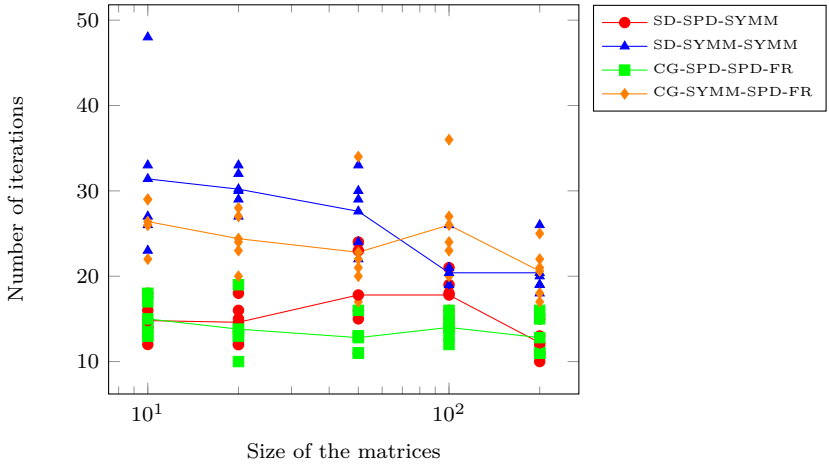
To investigate the accuracy of these first-order methods, the results of the algorithms are compared with a high precision computation of the Karcher mean (using the `vpa` functionality of MATLAB with 32 digits of accuracy). We expect a fully accurate result of 16 digits from the high precision computation, even when working with the largest magnitude of the conditioning of the matrices ($\mathcal{O}(10^{10})$). In Figure 3.3(a), the relative intrinsic distance between the results of the algorithms (for three 10×10 matrices) and this high precision solution is given as a function of the condition number of the matrices. The algorithms using the Riemannian inner product (3.3) display better results in general than those using the Euclidean one (3.4). Note that the steepest descent method starts having convergence problems when the Euclidean inner product and very ill conditioned matrices are used.

The speed of the algorithms is tested both for an increasing number of matrices (with the size fixed to 10×10 matrices) and for varying sizes of the matrices in the mean (with the amount fixed to five). Again we notice the advantage of the geometric constructions (gradient, etc.) associated with the Riemannian inner product, as these result in algorithms requiring less iterations. The computational cost of these constructions, however, is usually higher than those related to the Euclidean inner product, which causes the overall computational time to be very similar. In Figure 3.3(b), the number of iterations is displayed for the algorithms as a function of the size of the matrices, which shows the distinction between the two inner products.

We can conclude that geometric constructions, such as the gradient, associated



(a) Accuracy for different condition numbers.



(b) Required iterations for different sizes.

Figure 3.3: Comparison of the accuracy and computational speed of Algorithm 3.4.1 using the steepest descent and conjugate gradient algorithm. In the legends, we first indicate whether the Steepest Descent (**SD**) or Conjugate Gradient (**CG**) technique is used, next, which of the inner products (3.3) (**SPD**) or (3.4) (**SYMM**) is used, and finally whether retraction $R_X^{(\mathrm{sym})}$ (**SYMM**) or $R_X^{(\mathrm{pd})}$ (**SPD**) was taken. In case of CG, we also indicate which β type is used (**HS** or **FR**, see Section 3.4.2). The mean of the samples is indicated by the connecting lines.

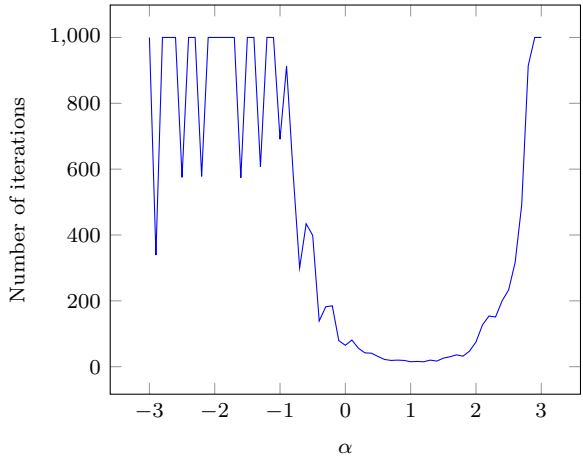


Figure 3.4: Number of iterations required for a steepest descent algorithm when using geometric constructions related to inner product (3.16) for different values of α.

with the Riemannian inner product are best suited for our optimization problem since the resulting algorithms require less iterations and provide a better accuracy. It therefore seems interesting to consider the performance of a steepest descent algorithm when the inner product

$$\langle \xi_X, \eta_X \rangle_X^{(\alpha)} = \operatorname{tr}\left(\xi_X X^{-\alpha} \eta_X X^{-\alpha}\right) \tag{3.16}$$

and its related geometric constructions are used, which reduces to the previous cases when $\alpha = 0$ or $\alpha = 1$. In Figure 3.4, we display the number of iterations such an algorithm requires for different values of α. The figure displayed has been constructed using the retraction $R_X^{(\mathrm{pd})}$, but a nearly identical figure was obtained using $R_X^{(\mathrm{sym})}$, again indicating that the influence of the inner product is far greater than that of the retraction. It is obvious from the figure that the Riemannian inner product, corresponding to $\alpha = 1$, is most natural to the manifold and results in the best performance.

3.5 Second-Order Implementations

The goal of second-order optimization techniques is to use (an approximation of) the Hessian of the cost function to obtain a quadratically (or at least superlinearly) converging algorithm. In this section, we discuss a number

of methods to accomplish this and compare their performance in terms of computational speed and accuracy. However, computing the Hessian typically has a higher computational complexity, which means it is yet to be determined whether these algorithms are more efficient than the first-order techniques. We first consider a Newton algorithm which is embedded in a trust region method, where the trust region subproblem is solved using a truncated conjugate gradient technique (Algorithms 10 and 11 in Absil et al. [1]). In the discussion of this Newton-based trust region method, we focus on the computation of the Hessian. Afterwards, the Riemannian BFGS method is used in a line-search algorithm.

In existing literature, the Riemannian Hessian of the Karcher cost function is computed using a decomposition as defined in Section 3.5.2 [54, 109]. We derive the Hessian according to the classical definition (3.15) (using the Levi–Civita connection) in Section 3.5.1 and an approximation in Section 3.5.3. The Riemannian BFGS method described in Section 3.5.4 will be the application of the existing generalized algorithm of Qi [105] and Savas and Lim [111] to $\mathcal{P}_n$.

3.5.1 Trust region method: Exact Hessian

In Section 3.3.4, we derived all the components needed to determine the Hessian of our cost function f and noticed that the result depends on the inner product as well. Hence, using the definition of the Hessian for the Euclidean inner product (3.4), with connection $\nabla^{(\mathrm{sym})}$, gives

$$\mathrm{Hess}^{(\mathrm{sym})} f(X)[\xi_X] = \mathrm{D}(\mathrm{grad}^{(\mathrm{sym})} f)(X)[\xi_X].$$

Entering the expression for $\mathrm{grad}^{(\mathrm{sym})} f(X)$ (3.9) into this equation leads to

$$\begin{aligned}\mathrm{Hess}^{(\mathrm{sym})} f(X)[\xi_X] =& 2\sum_{i=1}^{k} \mathrm{D}(\log)(A_i^{-1}X)[A_i^{-1}\xi_X]X^{-1} \\ &- 2\sum_{i=1}^{k} \log(A_i^{-1}X)X^{-1}\xi_X X^{-1},\end{aligned} \tag{3.17}$$

where we used the product and chain rules of differentiation as well as the differential of the matrix inverse function. We also recognize the differential of the matrix logarithm function at point $A_i^{-1}X$ in the direction of the tangent vector $A_i^{-1}\xi_X$, of which the computation has been extensively researched [3, 4, 64]. Note that some computational advantage can be obtained by using a similarity transformation to compute the derivative at the Hermitian matrix $A_i^{-1/2}XA_i^{-1/2}$ in the direction of the Hermitian matrix $A_i^{-1/2}\xi_X A_i^{-1/2}$.

When using the Riemannian inner product (3.3) with connection $\nabla^{(\mathrm{pd})}$, the Hessian becomes

$$\begin{aligned}\operatorname{Hess}^{(\mathrm{pd})} f(X)[\xi_X] = {} & \mathrm{D}(\operatorname{grad}^{(\mathrm{pd})} f)(X)[\xi_X] \\ & - \frac{1}{2}\left(\xi_X X^{-1} \operatorname{grad}^{(\mathrm{pd})} f(X) + \operatorname{grad}^{(\mathrm{pd})} f(X) X^{-1} \xi_X\right).\end{aligned} \tag{3.18}$$

In this case entering the expression for $\operatorname{grad}^{(\mathrm{pd})} f(X)$ (3.8) leads to

$$\begin{aligned}\operatorname{Hess}^{(\mathrm{pd})} f(X)[\xi_X] = {} & 2\sum_{i=1}^{k} \xi_X \log(A_i^{-1}X) + 2\sum_{i=1}^{k} X\, \mathrm{D}(\log)(A_i^{-1}X)[A_i^{-1}\xi_X] \\ & - \left(\sum_{i=1}^{k} \xi_X \log(A_i^{-1}X) + \sum_{i=1}^{k} \log(XA_i^{-1})\xi_X\right) \\ = {} & \sum_{i=1}^{k} \xi_X \log(A_i^{-1}X) - \sum_{i=1}^{k} \log(XA_i^{-1})\xi_X \\ & + 2\sum_{i=1}^{k} X\, \mathrm{D}(\log)(A_i^{-1}X)[A_i^{-1}\xi_X],\end{aligned} \tag{3.19}$$

where again the differential of the matrix logarithm is needed. Note that the first two terms in (3.19) are each other's (conjugate) transpose (except the minus sign), which can be exploited in the computations. At first sight this can seem somewhat peculiar, since this subtraction produces a skew-symmetric matrix while the result of the Hessian is supposed to be symmetric (its image is $T_X\mathcal{P}_n \simeq \mathcal{H}_n$). However, looking at (3.18), we can see that the differential of the gradient is symmetric and the second part, as the sum of a matrix and its transpose, is symmetric as well, proving that $\operatorname{Hess}^{(\mathrm{pd})} f(X)[\xi_X]$ is an element of $\mathcal{H}_n$ (where symmetric should be replaced with Hermitian in the complex setting).

3.5.2 Trust region method: Hessian by decomposition

Another way to compute the Hessian of a least squares cost function is described by Ferreira et al. [54], and more explicitly for the current cost function (3.1) by Rentmeesters and Absil [109]. The proposed procedure determines the Hessian by computing all components of a decomposition thereof, after which these are combined to obtain the actual Hessian. A downside, however, is that when determining the Karcher mean of k $n \times n$ matrices, this reconstruction consists

of $kn(n+1)/2$ terms, which grows rapidly as n increases (in Rentmeesters and Absil [109], only 3×3 matrices were considered). Two other important remarks are that this technique is derived for the manifold endowed with the Riemannian inner product (3.3) and that the computation of the terms as in Rentmeesters and Absil [109] is designed specifically for the Hessian taken at the identity matrix $X = I$. This causes the need to translate the problem in each iteration step to ensure this position for the current iteration point. Note that the Karcher cost function (3.1) remains invariant if all given matrices A_i and the argument X are transformed using the mapping $\mathcal{P}_n \to \mathcal{P}_n : Y \mapsto Z^{-1/2} Y Z^{-1/2}$, for all matrices $Z \in \mathcal{P}_n$. By choosing $Z = X_j$, with X_j the newly found iteration point, the problem setting can be translated to the identity matrix without changing the cost function. After convergence, we apply the inverse mapping to translate the identity matrix to the actual Karcher mean of the original matrices. Theoretically, this need for a translation is not a downside, in fact, it can even simplify notation and required constructions. Computationally, however, it could cause problems when working, e. g., with ill-conditioned matrices.

The expression for the Hessian is given by

$$\begin{aligned}\operatorname{Hess}^{(\mathrm{pd})} f(X)[\xi_X] =& 2 \sum_{\ell=1}^{m} \langle \xi_X, E_\ell \rangle_X^{(\mathrm{pd})} w_\ell(1) E_\ell \\ =& 2 \sum_{\ell=1}^{m} \operatorname{tr}(\xi_X X^{-1} E_\ell X^{-1}) w_\ell(1) E_\ell,\end{aligned}$$

where $m = kn(n+1)/2$ and E_ℓ, w_ℓ are defined as in Rentmeesters and Absil [109]. Explicit expressions for the components of the decomposition are given at $X = I$, hence the expression can be further simplified to

$$\operatorname{Hess}^{(\mathrm{pd})} f(I)[\xi_I] = 2 \sum_{\ell=1}^{m} \operatorname{tr}(\xi_I E_\ell) w_\ell(1) E_\ell. \tag{3.20}$$

When computing the Hessian, the rank one structure of E_ℓ can be exploited to limit the required amount of operations.

Remark We note that a more efficient implementation of the Newton method using this Hessian was brought to our attention by Rentmeesters [108] later on. In the numerical experiments of Section 3.5.5 we have used our own implementation, but we will discuss the consequences of using the alternative implementation.

3.5.3 Trust region method: Hessian by approximation

In the calculations of the exact Hessian, determining the differential of the matrix logarithm function appears to be a serious computational cost. A perhaps less elegant, but sometimes advantageous solution is to replace the matrix logarithm by (a truncation of) its Taylor series, at $X = I$ given by

$$\sum_{\ell=1}^{\infty} \frac{(-1)^{\ell+1}}{\ell}(X-I)^{\ell},$$

which converges to $\log(X)$ for all $\rho(X-I) < 1$, where ρ denotes the spectral radius. Truncating this series after the second term and entering the result into the expressions for the gradient (3.8) and (3.9), corresponding to inner products (3.3) and (3.4), we obtain

$$\operatorname{grad}_2^{(\mathrm{pd})} f(X) = 2\sum_{i=1}^{k}(2XA_i^{-1}X - \frac{3}{2}X - \frac{1}{2}XA_i^{-1}XA_i^{-1}X),$$

$$\operatorname{grad}_2^{(\mathrm{sym})} f(X) = 2\sum_{i=1}^{k}(2A_i^{-1} - \frac{3}{2}X^{-1} - \frac{1}{2}A_i^{-1}XA_i^{-1}),$$

respectively as an approximation to the gradient. Note that we only use this approximate gradient to derive the approximated Hessian, while the actual gradient will be used in the final algorithm. Applying the definition of the Hessian to these expressions results in

$$\begin{aligned}
\operatorname{Hess}_2^{(\mathrm{pd})} f(X)[\xi_X] &= \mathrm{D}(\operatorname{grad}_2^{(\mathrm{pd})} f)(X)[\xi_X] \\
&\quad - \frac{1}{2}\left(\xi_X X^{-1} \operatorname{grad}_2^{(\mathrm{pd})} f(X) + \operatorname{grad}_2^{(\mathrm{pd})} f(X) X^{-1}\xi_X\right) \\
&= \sum_{i=1}^{k}\Bigg(2XA_i^{-1}\xi_X + 2\xi_X A_i^{-1}X - \frac{1}{2}\xi_X A_i^{-1}XA_i^{-1}X \\
&\quad - XA_i^{-1}\xi_X A_i^{-1}X - \frac{1}{2}XA_i^{-1}XA_i^{-1}\xi_X\Bigg), \qquad (3.21)
\end{aligned}$$

$$\begin{aligned}
\operatorname{Hess}_2^{(\mathrm{sym})} f(X)[\xi_X] &= \mathrm{D}(\operatorname{grad}_2^{(\mathrm{sym})} f)(X)[\xi_X] \\
&= \sum_{i=1}^{k}\left(3X^{-1}\xi_X X^{-1} - A_i^{-1}\xi_X A_i^{-1}\right). \qquad (3.22)
\end{aligned}$$

The finite convergence radius of the Taylor series indicates that when the matrices in the mean lie close to each other (and to the current estimate), a trust region method using one of these Hessians is expected to work well. However, since these are only approximations to the Hessian, quadratic convergence of the algorithm is no longer guaranteed.

3.5.4 Riemannian BFGS method

Finally, we test a Riemannian generalization of the classical BFGS method [105, 111], which, instead of working with the exact Hessian, uses an estimate of the Hessian that is updated throughout the algorithm. The main point of interest is how this *update* is performed. Suppose we know the estimate $\mathfrak{B}_j$ of the Hessian at iteration point X_j, where it is assumed to be a linear operator from $T_{X_j}\mathcal{P}_n$ onto itself and can thus be represented by an $(n(n+1)/2) \times (n(n+1)/2)$ matrix B_j. Then the linear operator $\mathfrak{B}_{j+1} : T_{X_{j+1}}\mathcal{P}_n \to T_{X_{j+1}}\mathcal{P}_n$, corresponding to the new iteration point X_{j+1}, is defined by

$$\mathfrak{B}_{j+1}p = \widetilde{\mathfrak{B}}_j p - \frac{\langle s_j, \widetilde{\mathfrak{B}}_j p\rangle_{X_{j+1}}}{\langle s_j, \widetilde{\mathfrak{B}}_j s_j\rangle_{X_{j+1}}}\widetilde{\mathfrak{B}}_j s_j + \frac{\langle y_j, p\rangle_{X_{j+1}}}{\langle y_j, s_j\rangle_{X_{j+1}}} y_j, \quad \forall p \in T_{X_{j+1}}\mathcal{P}_n, \tag{3.23}$$

$$\left[\widetilde{\mathfrak{B}}_j = \mathfrak{T}_{t^A\xi_j} \circ \mathfrak{B}_j \circ \left(\mathfrak{T}_{t^A\xi_j}\right)^{-1}, \right. \tag{3.24}$$

in which ξ_j is the search direction at X_j, t^A is the Armijo step size taken to arrive at X_{j+1}, s_j is the vector transport of the old search direction ξ_j to the new iteration point, y_j is a measure for the change of the gradient over the iteration step (formal expressions can be found in Algorithm 3.5.1), and $\mathfrak{T}_{t^A\xi_j}$ is a vector transport. Since an inner product and a vector transport are present in these expressions, there are again different situations to investigate. We will test the algorithm for each of the two inner products, combined with their natural retraction and vector transport. Note that to evaluate $\mathfrak{B}_{j+1}p$ using the matrix representation B_{j+1}, we need an $n(n+1)/2$ vector representation of p. This is done by the half-vectorization operator vech, which stacks the elements of the upper triangular part of p columnwise. Using this representation, the matrix-vector product $B_{j+1}\,\mathrm{vech}(p)$ returns a vector of length $n(n+1)/2$ which is the half-vectorization of the matrix $\mathfrak{B}_{j+1}p$.

In the easier case of the Euclidean inner product (3.4), expressions (3.23) and (3.24) become

$$\begin{aligned} B_{j+1}\operatorname{vech}(p) =& \widetilde{B}_j \operatorname{vech}(p) - \widetilde{B}_j \operatorname{vech}(s_j)\frac{\operatorname{tr}(s_j\widetilde{\mathfrak{B}}_j p)}{\operatorname{tr}(s_j\widetilde{\mathfrak{B}}_j s_j)} \\ &+ \operatorname{vech}(y_j)\frac{\operatorname{tr}(y_j p)}{\operatorname{tr}(y_j s_j)}, \qquad \forall p \in T_{X_{j+1}}\mathcal{P}_n \\ \Big[& \widetilde{\mathfrak{B}}_j = \mathfrak{B}_j. \end{aligned}$$

To remove p from this expression, we split the matrix traces using an adaptation of the property $\operatorname{tr}(AB) = \operatorname{vec}(A)^H \operatorname{vec}(B)$, with A and B symmetric (Hermitian) matrices, vec the operator which stacks the columns of a matrix into a vector, and A^H denotes the conjugate transpose of a matrix A. Since the second matrix should be half-vectorized, this needs to be compensated for in the first vectorization. To this end, we also change the vectorization of the first matrix to half-vectorization, but with the adaptation that each off-diagonal element is doubled, an operation we denote by vech_2. This yields $\operatorname{tr}(AB) = \operatorname{vech}_2(A)^H \operatorname{vech}(B)$ and our update formula becomes

$$\begin{aligned} B_{j+1} =& B_j - \frac{1}{\operatorname{tr}(s_j\mathfrak{B}_j s_j)} B_j \operatorname{vech}(s_j)\operatorname{vech}_2(s_j)^H B_j \\ &+ \frac{1}{\operatorname{tr}(y_j s_j)} \operatorname{vech}(y_j)\operatorname{vech}_2(y_j)^H. \end{aligned} \tag{3.25}$$

Note that this update differs in only two rank-1 terms from the previous estimate of the Hessian, which can be exploited in the implementation.

For the Riemannian inner product (3.3), the calculation of $\widetilde{\mathfrak{B}}_j$ is no longer so straightforward. Entering vector transport $\mathfrak{T}^{(\mathrm{pd})}$ (3.14) into equation (3.24), we obtain

$$\begin{aligned} \widetilde{\mathfrak{B}}_j p &= Q\mathfrak{B}_j(Q^{-1}pQ^{-H})Q^H, \\ \left[Q \right. &= \left. X_j^{\frac{1}{2}} \exp\left(\frac{X_j^{-\frac{1}{2}} t^A \xi_j X_j^{-\frac{1}{2}}}{2} \right) X_j^{-\frac{1}{2}}, \right. \end{aligned}$$

with $(\cdot)^{-H}$ a contraction of $(\cdot)^{-1^H}$. To extract p from this expression, we want to use the property $\operatorname{vec}(ABC) = (C^T \otimes A)\operatorname{vec}(B)$ for general matrices A, B, and C [66], with $\otimes$ the Kronecker product, but this property cannot be used when half-vectorization is applied. Therefore, let us pretend for a moment that

$\mathfrak{B}_j$ and $\widetilde{\mathfrak{B}}_j$ are represented by $n^2 \times n^2$ matrices and apply this rule to the above expression:

$$\begin{aligned}\widetilde{B}_j \operatorname{vec}(p) =& \operatorname{vec}(Q\mathfrak{B}_j(Q^{-1}pQ^{-H})Q^H)\\ =&(Q^* \otimes Q)B_j \operatorname{vec}(Q^{-1}pQ^{-H})\\ =&(Q^* \otimes Q)B_j(Q^{-*} \otimes Q^{-1}) \operatorname{vec}(p),\end{aligned}$$

with $(\cdot)^*$ the elementwise complex conjugate of a matrix. Changing back to half-vectorization can be accomplished by using the so-called duplication and elimination matrices D_n and E_n, which are simple matrices for transforming respectively a half-vectorization into a normal vectorization and vice versa (under the assumption of working with symmetric/Hermitian matrices). Equation (3.23) can be tackled in the same fashion as before, where we only need to pay attention to the extra factors in the current inner product. The total update procedure now becomes

$$\begin{aligned}B_{j+1} =&\widetilde{B}_j - \frac{1}{\operatorname{tr}(X_{j+1}^{-1}s_jX_{j+1}^{-1}\widetilde{\mathfrak{B}}_j s_k)}\widetilde{B}_j \operatorname{vech}(s_j) \operatorname{vech}_2(X_{j+1}^{-1}s_jX_{j+1}^{-1})^H \widetilde{B}_j\\ &+ \frac{1}{\operatorname{tr}(X_{j+1}^{-1}y_jX_{j+1}^{-1}s_j)} \operatorname{vech}(y_j) \operatorname{vech}_2(X_{j+1}^{-1}y_jX_{j+1}^{-1})^H. \qquad (3.26)\\ \left[\widetilde{B}_j = E_n(Q^* \otimes Q)D_nB_jE_n(Q^{-*} \otimes Q^{-1})D_n,\right.\end{aligned}$$

Note that there are again two rank-1 terms present in the update. However, the update from B_j to $\widetilde{B}_j$ is no longer the identity and can in this case be seen as an update of the rank-1 terms which originated in the previous iterations.

Now that the techniques to update the estimate of the Hessian are specified, we show the entire Riemannian BFGS method in Algorithm 3.5.1. As for the first-order optimization techniques, the stopping criteria in this algorithm check whether the Armijo step size or the absolute or relative difference between two consecutive iteration points are smaller than their respective tolerances.

3.5.5 Numerical experiments

We discuss the accuracy and computational speed of the different second-order optimization techniques and compare the results with those of the first-order optimization methods. As before, the optimization algorithms are initiated using the CHEAP mean.

Algorithm 3.5.1 The Karcher mean using the Riemannian BFGS method

Let $A_1, ..., A_k$, $k > 2$, be PD matrices, X_0 a PD initial guess, B_0 an initial matrix approximation to the Hessian at X_0, and $R^{(i)}$ and $\mathfrak{T}^{(i)}$ ($R^{(\mathrm{sym})}$ and $\mathfrak{T}^{(\mathrm{sym})}$ or $R^{(\mathrm{pd})}$ and $\mathfrak{T}^{(\mathrm{pd})}$) a retraction and a vector transport

- $j = 0$;
- $grad_0 = \operatorname{grad} f(X_0)$; {$f$ is our cost function}
- **while** (not converged)
  - Obtain ξ_j: Solve the system: {The search direction}

$$B_j \operatorname{vech}(\xi_j) = -\operatorname{vech}(grad_j);$$

  - **if** (ξ_j not a descent direction)
    - $\xi_j = -grad_j$;
  - **end if**
  - $X_{j+1} = R^{(i)}_{X_j}(t^A \xi_j)$; {with t^A the Armijo step size [1]}
  - $grad_{j+1} = \operatorname{grad} f(X_{j+1})$;
  - $s_j = \mathfrak{T}^{(i)}_{t^A \xi_j}(t^A \xi_j)$;
  - $y_j = grad_{j+1} - \mathfrak{T}^{(i)}_{t^A \xi_j}(grad_j)$;
  - Update B_j to B_{j+1} using (3.25) or (3.26), depending on the inner product;
  - $j = j + 1$;
- **end while**

Return: Karcher mean $\mathbf{K}(A_1, ..., A_k)$

Accuracy We start by applying the second-order techniques to three 10×10 matrices and compare the results with a high precision computation of the Karcher mean, which is obtained using the `vpa` functionality of MATLAB with 32 digits of accuracy. The results are shown in Figure 3.5. In general, the Newton-based trust region methods from Sections 3.5.1 and 3.5.2 display a better accuracy than the previous first-order techniques, even though occasional convergence problems shift the mean values in Figure 3.5. The Newton-based trust region method using the approximated Hessian from Section 3.5.3 shows very bad accuracy, hence we will not discuss this method any further.

The Riemannian BFGS line-search method on the other hand displays an accuracy similar to the first-order methods, which steadily deteriorates as the condition number of the matrices increases. In general, we notice that the optimization methods, which use the Armijo line search technique to determine the step size, stop when the norm of the gradient is of the order of the square root of machine precision. This is caused by the use of the squared norm of the gradient in the Armijo condition, and can be resolved by switching to a different method to choose the step size, e. g., a fixed step size.

Overall, we again notice a slightly better accuracy for the techniques corresponding to the Riemannian inner product (3.3).

Computational speed To test the speed of the algorithms, the number of matrices in the mean is varied in Figure 3.6 (where we fix the size to 10×10 matrices) as well as the size of the matrices in Figure 3.7(a) (when taking 5 matrices). Figure 3.6 once more displays the advantages of the Riemannian inner product, resulting in fewer iterations (Figure 3.6(a)) and a lower computational time (Figure 3.6(b)), even though its corresponding geometric constructions, such as the gradient and Hessian, will in general be more expensive to compute.

The advantage of the smaller amount of iterations when using the Riemannian inner product can be seen when comparing the computational time of the two resulting Newton-based trust region methods (Figure 3.6(b): TR-SPD and TR-SYMM). In fact, the Newton-based trust region method using the exact Hessian corresponding to the Riemannian inner product (TR-SPD) gives good results, approaching the computational time of the first-order methods. Although not indicated in the figures, an even faster computational speed can be obtained for the Newton-based trust region method with the Hessian by decomposition (Section 3.5.2) by using the efficient implementation of Rentmeesters [108] (the method TR-DECOMP in the figures refers to our own implementation of this technique). As indicated by Rentmeesters and Absil [109], this algorithm is faster than the first-order optimization methods for small matrices, but will lose its computational advantage as the size of the matrices increases. Consequently,

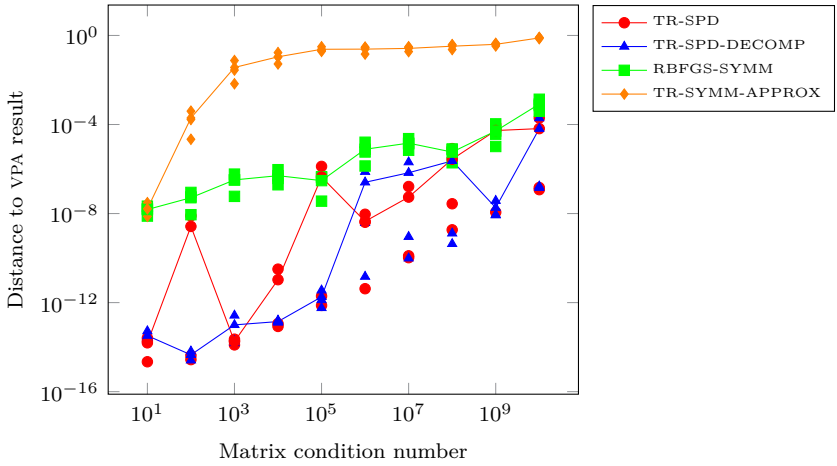
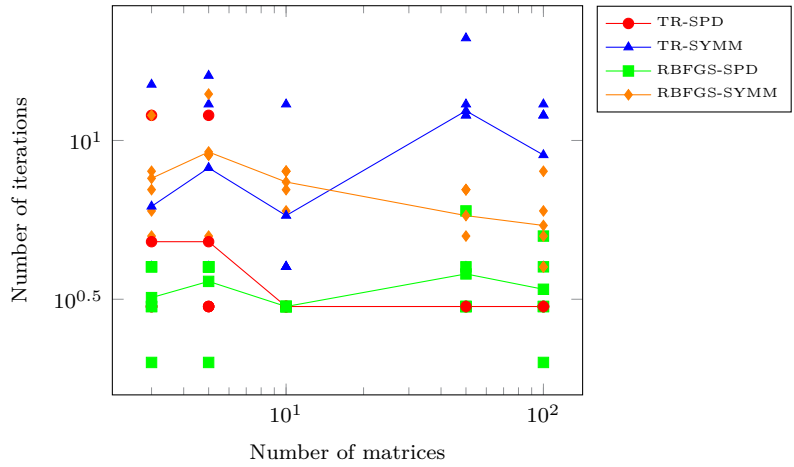


Figure 3.5: Comparison of the accuracy of the Karcher mean using various Newton-based trust region methods and the Riemannian BFGS line-search algorithm. In the legends, we first indicate whether the Newton-based trust region (**TR**) or Riemannian BFGS (**RBFGS**) technique is used, next, which of the inner products (3.3) (**SPD**) or (3.4) (**SYMM**) is used and finally an extra term if a non-standard method was used (**DECOMP** for the technique in Section 3.5.2 and **APPROX** for those in Section 3.5.3). The mean of the samples is in both figures indicated by the connecting lines.

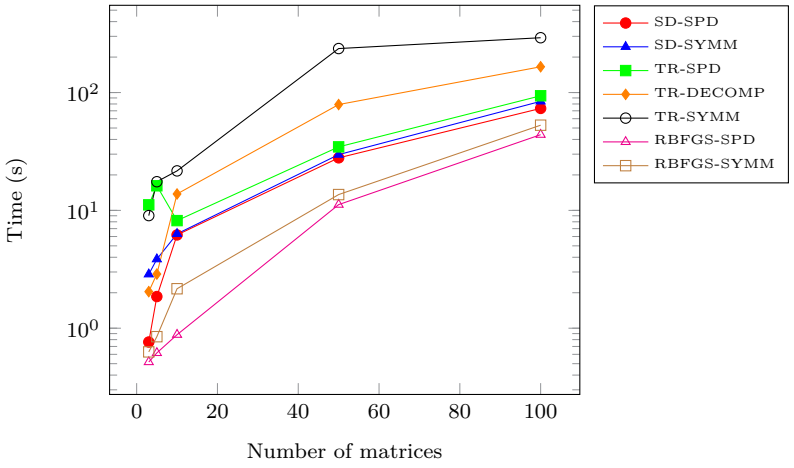
the gradient descent method again becomes computationally more efficient for larger matrices, as also observed and concluded in our experiments.

Another remarkable result in Figure 3.6(b) is the performance of the Riemannian BFGS line-search method, which displays a lower computational time than the steepest descent algorithm. We do note that this test is performed for a varying number of 10×10 matrices, and as the size of the matrices starts to increase, the Riemannian BFGS line-search method, as well as the other superlinear techniques, are outperformed by the steepest descent and conjugate gradient algorithm (see Figure 3.7(a)).

Figure 3.7(b) shows the evolution of the gradient for all algorithms based on the Riemannian inner product. The quadratic convergence of the Newton-based trust region algorithm is clearly visible, as well as a superlinear convergence for the Riemannian BFGS line-search method. The steepest descent and conjugate gradient algorithms display very similar (linear) convergence since the problem is well-conditioned, eliminating the need for the conjugate gradient technique to be activated. Finally, the Newton-based trust region algorithm using the



(a) Required iterations in the second-order methods.



(b) Required time for different methods using the two inner products.

Figure 3.6: Comparison of all the discussed optimization algorithms for a varying number of 10×10 matrices. In the legends, the abbreviations **SD** (Steepest Descent), **TR** (Newton-based Trust Region) and **RBFGS** (Riemannian BFGS) are used to denote the applied technique, after which the inner product is indicated with **SPD** (3.3) or **SYMM** (3.4). For TR the suffix **DECOMP** is added to indicate the technique in Section 3.5.2.

approximated Hessian (Section 3.5.3) has lost all quadratic convergence and displays an even slower convergence than the steepest descent method.

3.6 Performance of the approximate means

As mentioned, we resume our analysis of the algorithms proposed in Section 2.5 as approximations to the matrix geometric mean.

We discuss their accuracy in terms of proximity to the Karcher mean. Afterwards, we examine their usefulness as an initial guess for the Riemannian optimization algorithms previously discussed for the Karcher mean. As a comparison, we also consider some standard initializers of the optimization algorithms, namely the arithmetic, harmonic and CHEAP mean [28].

3.6.1 Distance to the Karcher mean

When comparing the Circular mean (Algorithm 2.5.3) and HA mean (Algorithm 2.5.5), both in their fixed order and randomized order versions, we notice nearly identical proximity to the Karcher mean for all four algorithms. Figure 3.8 displays these results for the two versions of the HA mean. Their proximity to the Karcher mean decreases only slightly as the conditioning of the matrices deteriorates.

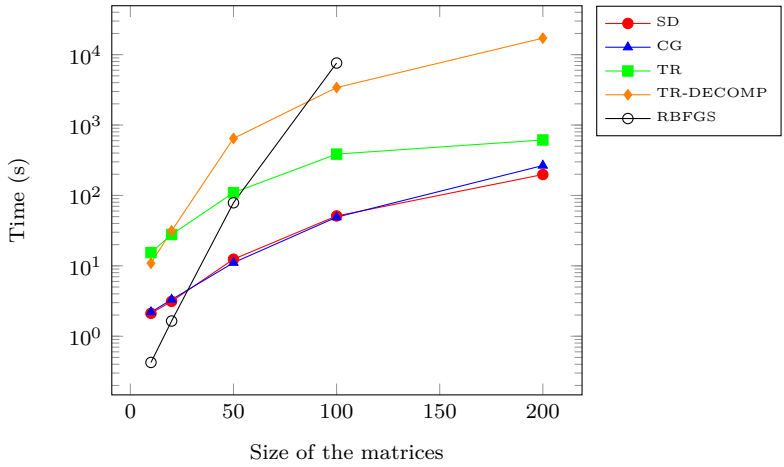
Combining this with the analysis of the computational cost of the four algorithms (Section 2.5.3), we conclude that the randomized version of the HA mean is most suited as an approximation to the Karcher mean.

In Section 2.4.1, we already mentioned that the log-Euclidean mean is located far from the small cluster of planar approaches to a geometric mean. As can be seen in Figure 3.8, the same holds true when comparing the log-Euclidean to the Karcher mean.

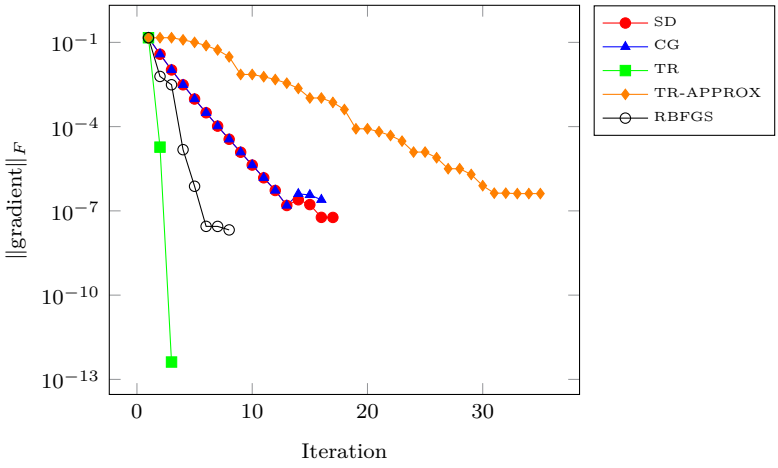
The inductive mean exhibits a similar proximity to the Karcher mean as the HA mean for well-conditioned matrices. However, as the condition of the matrices becomes worse, its proximity deteriorates slightly faster than that of the HA mean.

Concerning the crude guess from Algorithm 2.5.2, its expected bad proximity to the Karcher mean can be observed in Figure 3.8.

In Figure 3.9, we display the location of the various approximations with respect to the original matrices for a simple example of three 2×2 PD matrices. We



(a) Required time for the different methods for 5 matrices of varying size.



(b) Norm of the gradient throughout the iterations.

Figure 3.7: Comparison of all the discussed algorithms for the Riemannian inner product (3.3). In the legends, the abbreviations **SD** (Steepest Descent), **CG** (Conjugate Gradient), **TR** (Newton-based Trust Region) and **RBFGS** (Riemannian BFGS) are used to denote the applied technique. For TR the suffix **DECOMP** or **APPROX** is added to indicate the techniques in Section 3.5.2 and 3.5.3 respectively.

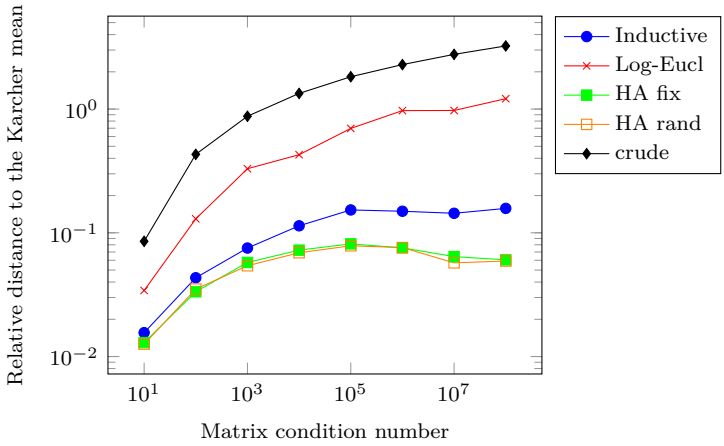


Figure 3.8: Relative distance of the approximate means to the Karcher mean for a varying condition number of the given matrices. The figure displays only an average of such distances to avoid clutter. In the legend, **Inductive** indicates the inductive mean (2.8), **Log-Eucl** the log-Euclidean mean (2.7), **crude** the mean in Algorithm 2.5.2, and **HA** the mean of Algorithm 2.5.5, both for fixed order and randomized order.

have added the geodesics, the lines of shortest distance in the Riemannian space, between these original matrices to demonstrate the Riemannian curvature of the manifold. As can be seen, the various means appear very close to the Karcher mean in this simple example of well-conditioned matrices.

3.6.2 Initial guesses for the Karcher mean algorithm

After analyzing the various proposed means in terms of performance, we now test their appropriateness as an initial guess to the conjugate gradient (CG) algorithm for computing the Karcher mean. In Table 3.1, the average number of iterations the CG algorithm requires until convergence is shown for well-conditioned and ill-conditioned matrices. These results are rounded averages of a number of repetitions of the test.

After inspecting the table, we can see that in general both the CHEAP mean and the two fixed order means (Circular and HA) result in a low number of iterations. The inductive and log-Euclidean mean also serve as a good initial guess for the Karcher mean. On the other hand, the arithmetic mean, the harmonic mean, and the crude guess appear to give the worst results, and are therefore least

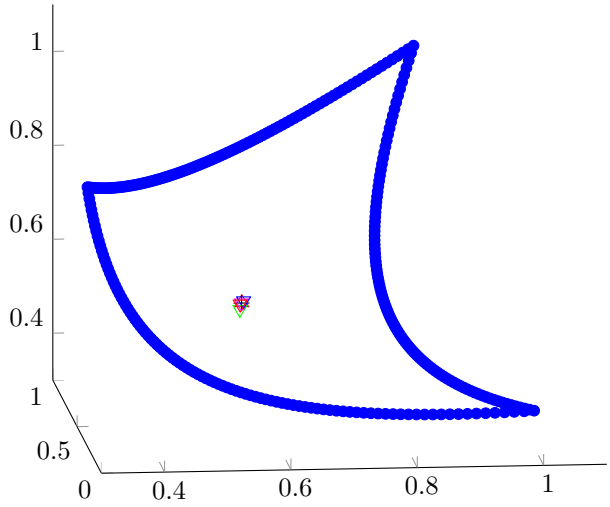


Figure 3.9: Representation in 3D space of three 2×2 PD matrices using the two diagonal elements and the off-diagonal element. The curved triangle represents the geodesics, the lines of shortest distance, between the three matrices, which are the vertices.

suited as an initial guess. If we look at the ill-conditioned matrices separately, we also notice that the HA mean (Algorithm 2.5.5) performs especially well, both with fixed order and randomized graph.

3.7 Geometric kernel data fusion

Throughout our discussion of the Karcher mean of PD matrices and its various approximations and related means, it has become apparent that the topic of geometric means is theoretically very appealing. In this section, however, we turn our attention towards an application in bioinformatics, where the classification of protein folds is improved by fusing multiple data features in a geometrically inspired manner.

We present an introduction to the application and indicate how the geometric mean is introduced into this context. For full technical details, we refer to Zakeri et al. [134].

Table 3.1: Number of iterations required for the conjugate gradient algorithm for the Karcher mean to converge, using the specified approximation as an initial guess (averaged over some repetitions). The condition number indicates the conditioning of the matrices averaged by the mean.

Condition number	1e1	1e8
Inductive	15	16
Circular, fixed	15	14
Circular, random	15	15
Crude guess	18	17
HA, fixed	14	11
HA, random	16	13
log-Euclidean	14	15
Arithmetic mean	19	16
Harmonic mean	21	17
CHEAP mean	15	16

3.7.1 Protein fold classification

Knowledge on the tertiary, or 3-dimensional, structure of proteins can provide information on a number of their properties, such as the biological function of the proteins and protein-protein interaction. Hence, determining this structure is among the essential objectives in molecular biology, cell biology, proteomics, and bioinformatics.

While recent developments in genome sequencing projects have greatly increased the number of protein coding sequences, the identification of the 3-dimensional structure of proteins is still expensive and time-consuming. Instead, a taxonomic approach is considered which classifies the proteins based on some available information on the folds in their 3-dimensional structure, where the number of these protein domain folds are assumed to be restricted [50, 96].

In practice, information on protein folds can be obtained through various representative models of protein features from separate data sources, such as primary structural information [39, 49, 132], local pairwise sequence alignment-based feature spaces [42], physicochemical properties of constituent amino acids [49, 90], and sequence evolution information [39]. As these features arise from different research fields, the resulting data is of a heterogeneous nature. A popular approach is therefore to represent the various data sources using kernel matrices, after which kernel-based data fusion can be applied. Since these kernel

matrices are PD by construction, we exploit this property by fusing the kernels using the Karcher mean or one of its approximations. Finally, a one-against-others support vector machine (SVM) classifier is trained and applied on the fused data.

3.7.2 Kernel construction

As a first step of the protein fold recognition, we need to construct the kernels corresponding to the separate protein features.

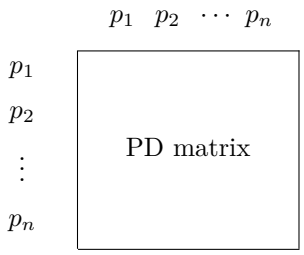
Suppose we are considering n proteins $p_1, \ldots, p_n$ and some protein feature f. This protein feature can be measured for each of the proteins, resulting in the samples $f_1, \ldots, f_n$. The construction of the corresponding kernel matrix is performed using a radial basis function (RBF) kernel function. A simple example of a Gaussian RBF kernel for the samples $f_1, \ldots, f_n$ is given by the matrix PD K with entries

$$K_{i,j} = \exp\left(-\frac{\|f_i - f_j\|^2}{2\sigma^2}\right),$$

where $\|\cdot\|$ is an appropriate norm corresponding to the data type of the samples, such as scalars or vectors, and σ a free parameter.

In practice, however, multiple measurements $f_{i,1}, \ldots, f_{i,m}$, $i = 1, \ldots, n$, are performed on all proteins, where the number of measurements can differ according to the protein feature f. Afterwards, the kernel matrices are constructed using a more advanced RBF kernel function, which is defined for any number m of repeated measurements and again results in a PD matrix.

In the end, the kernel matrices can be seen as a non-linear extension of covariance/correlation matrices and encode the similarity between the proteins based on a specific protein feature. In general, we can think of a kernel matrix as follows,



where the (i, j)-th element in the matrix represents the similarity between proteins p_i and p_j based on some protein feature f.

3.7.3 Geometric kernel fusion

Classically, kernel fusion is performed by combining the kernel matrices in a weighted arithmetic mean. Several involved convex optimization-based approaches exist which try to optimize the kernel weights based on different optimization criteria [10, 78, 79, 106, 115, 116]. These weights have the intuitive interpretation of representing the importance of the corresponding kernels and the information they hold. However, such linear convex combinations often fail to capture all information for kernels containing complementary, non-redundant information.

Moreover, a desirable property emerges when the kernel matrices are considered as an extension of the covariance matrices of Gaussian distributions. The inverse of such a covariance matrix K is referred to as a precision matrix P, which encodes the independence relations between the variables (or samples) in the form of partial correlations. We now denote by $\mathbf{G}$ a geometric mean, which satisfies the property of invariance under inversion, and $K = \mathbf{G}(K_1, K_2)$, with K_1 and K_2 covariance matrices. The precision matrices corresponding to K_1 and K_2 are denoted by P_1 and P_2, with $P = \mathbf{G}(P_1, P_2)$. Then it follows that

$$P = \mathbf{G}(P_1, P_2) = \mathbf{G}(K_1^{-1}, K_2^{-1}) = K^{-1}.$$

Hence, computing the mean of the covariance matrices is equivalent to the computation of the mean of the precision matrices, a duality which is geometrically and computationally appealing.

Now, since we are working with PD kernels for which invariance under inversion is an intuitively appealing property of the averaging process, we consider performing the kernel fusion using a geometric mean instead of a linear combination of the kernels. Specifically, we fuse the kernels using the Karcher mean, the log-Euclidean mean (2.7), and the randomized version of the HA mean (Algorithm 2.5.5) as an approximation. Some other means and methods were also considered as a comparison.

3.7.4 Improving the state-of-the-art kernel fusion methods

Once the kernels of the separate data feature sources have been combined using some fusion method, e. g., the Karcher mean, the fused data can be used to train an SVM classifier of the type “one-against-others”, or “one-vs-the-rest”. Such

Table 3.2: Performance of some protein fold classifiers based on different methods for kernel fusion.

Kernel fusion	**Performance (%)**
Arithmetic mean	60.57
Harmonic mean	65.80
MKLdiv-dc [133]	75.19
HPFP [90]	74.21
Karcher mean	86.16
randomized HA mean	86.68
log-Euclidean mean	81.72

a classifier relies on the assumption of a restricted number of protein domain folds, since it trains a partial classifier for each possible protein structure, or class, using examples of the specified protein structure as positive samples and all other as negative ones. After the training phase, the overall classifier can be applied to new data, where the partial classifiers will return a probability of class membership, leading to the protein structure of highest probability.

To test the performance of the new kernel fusion methods, we also consider some approaches which perform kernel fusion using weighted arithmetic means. The process of finding appropriate weights from training data and taking the weighted mean is also referred to as multiple kernel learning (MKL).

The resulting classifiers are trained and tested using the well-established SCOP (Structural Classification of Proteins) benchmark dataset [96]. We focus on combining 26 RBF kernels, corresponding to 26 protein features. First, the fusion is performed on a training set containing 311 proteins, which corresponds to combining 26 PD matrices of size 311. Next, the result of this fusion is used to train a one-against-others SVM classifier. Finally, we fuse the 26 kernel matrices corresponding to the 383 proteins in the test set, and apply the SVM classifier to the result. In Table 3.2, we present the performance of some of the most promising predictors (MKLdiv-dc [133], HPFP [90]) and some predictors based on recognizable fusion methods (arithmetic mean, harmonic mean), along with the results of the new geometric kernel fusion methods. Performance is indicated by the percentage of correct protein fold classification on the test dataset.

Using one of the geometrically inspired means, we notice an improved protein fold recognition on the test set, with highest performance for the randomized HA mean (Algorithm 2.5.5) where we have 86.7% correct classification. Moreover,

the randomized HA mean has a low computational cost, resulting in a method for protein fold classification which is both performant and efficient.

Our initial intention in this application was the recognition of the structure or folds of a protein in order to obtain information on their biological function and other properties. However, some (parts of) protein sequences can already be associated with clear functional descriptions. Incorporating the known functional information of these protein sequences into the model allows us to further identify the possible function and structure of other protein sequences. Doing so yields an increased 89.3% correct classification on the test set.

3.8 Conclusions

In this chapter, we have discussed the Karcher mean, one of the main instances of the matrix geometric mean with an appealing analogy to the arithmetic mean. As a consequence, it is widely considered as the most natural generalization of the geometric mean. Various optimization algorithms, both first- and second-order techniques, were presented for its computation. The convergence of these algorithms was not discussed explicitly, as it is examined more generally by Absil et al. [1]. For the first-order optimization techniques, the linear convergence can easily be verified using Corollary 4.3.2 and Theorems 4.4.1 and 4.4.2 in Absil et al. [1] and exploiting the convexity of the problem. The superlinear convergence of the second-order optimization methods, although predicted by the experiments, is theoretically not so easily guaranteed. The convergence of the Riemannian BFGS line-search method was considered by Gallivan, Qi, and Absil [60], where a Riemannian version of the Dennis-Moré condition [46] was derived. For the Newton-based trust region methods, a local and global convergence analysis was performed by Absil et al. [1].

We noticed that while the second-order techniques required less iterations, the computational cost associated with each of these iterations was higher than that of the first-order algorithms, which in general nullified the advantage of superlinear convergence. Hence we conclude that for the current algorithms on the manifold $\mathcal{P}_n$, it is more advantageous to work with first-order optimization techniques when the size of the matrices increases.

It could be possible to produce more efficient second-order optimization algorithms if we were to reduce our search space, meaning the manifold of interest, to a certain subset of which the structure can be further exploited. For example, the geometry of the manifold of larger matrices of fixed, low rank has already been extensively researched [32, 122, 123], and can be used in the Riemannian optimization techniques presented in this chapter.

Concerning the approximate means, especially the Circular and HA means, we noticed that a randomization of the matrix order in every iteration provides a significant speed-up of convergence, while maintaining proximity to the Karcher mean. Considering proximity to the Karcher mean, computational speed, and appropriateness as an initial guess to the Karcher mean optimization problem, the inductive mean and the randomized version of the HA mean appear to provide optimal results.

Chapter 4

The geometric mean of structured matrices

In the previous chapter, we discussed the Karcher mean of positive definite matrices, which is considered as a most natural generalization of the matrix geometric mean. While this mean returns a positive definite matrix, any additional structure present in the matrices will be ignored by the optimization algorithms and will be destroyed in general.

In this chapter, we present an adaptation of the Karcher mean which accounts for additional structure. After a theoretical exploration of this structured mean, we develop and analyze some algorithms for its computation.

The content of this chapter was published as a part of

> Bini, D., Iannazzo, B., Jeuris, B., and Vandebril, R. Geometric means of structured matrices. *BIT Numerical Mathematics 54*, 1 (2014), pp. 55–83.

4.1 Introduction

The wish to generalize the concept of the geometric mean to positive definite (PD) matrices and, on the other hand, the need to average quantities expressed by PD matrices in certain applications have led to the definition and the study of the Karcher mean [21, 22, 93]. This mean and its accompanying geometry

have been extensively discussed in the previous chapter. If not mentioned, we will assume throughout this chapter that the manifold of PD matrices $\mathcal{P}_n$ is endowed with its natural inner product (3.3), along with the corresponding intrinsic distance. For convenience, we recall these here,

$$\langle \xi_X, \eta_X \rangle_X = \operatorname{tr}(\xi_X X^{-1} \eta_X X^{-1}),$$

$$\delta(X, Y) = \left\| \log(X^{-1/2} Y X^{-1/2}) \right\|_F, \tag{4.1}$$

with $X, Y \in \mathcal{P}_n$, $\xi_X, \eta_X \in T_X \mathcal{P}_n$.

An important feature of the Karcher mean is that it possesses all the properties desired by a geometric mean, like the ten axioms summarized in the ALM list [6]. For this reason, it is a viable tool in applications requiring some of these properties [19, 94]. For instance, a geometric mean should be permutation invariant, monotone, joint concave, and should satisfy the arithmetic-geometric-harmonic inequality (see Section 2.2 for the precise statements of the properties). In particular, one of the most characteristic properties of a geometric mean is its invariance under inversion:

$$\mathbf{G}(A_1^{-1}, \ldots, A_k^{-1}) = \mathbf{G}(A_1, \ldots, A_k)^{-1}. \tag{4.2}$$

In certain applications, however, besides the positive definiteness, the data matrices have some further structure in the sense that they belong to some special subset $\mathcal{S}$, say a linear space. For instance, in the design and analysis of certain radar systems, the matrices to be averaged are correlation matrices, which are PD Toeplitz matrices or are PD Toeplitz block matrices with Toeplitz blocks [12, 14, 15, 81, 130]. In these cases, one would like the geometric mean to belong to the same class $\mathcal{S}$ as the data. Unfortunately, the Karcher mean does not preserve many structures, in particular the Karcher mean of Toeplitz and/or band matrices is typically not of Toeplitz and/or band form anymore, as illustrated by the following simple example.

Example 4.1. Let $\mathcal{S}$ be the set of tridiagonal Toeplitz matrices and choose $A_1, A_2 \in \mathcal{S}$, where $A_1 = I$ (the identity matrix) and $A_2 = \operatorname{tridiag}(1, 2, 1)$ (the matrix with 2's on the main, and 1's appearing on sub- and superdiagonals). We have $A_1 A_2 = A_2 A_1$, thus the Karcher mean equals $(A_1 A_2)^{1/2}$. For $n = 3$, we get

$$(A_1 A_2)^{1/2} = \frac{\sqrt{2}}{4} \begin{bmatrix} \sqrt{2+\sqrt{2}}+2 & \sqrt{2}\sqrt{2-\sqrt{2}} & \sqrt{2+\sqrt{2}}-2 \\ \sqrt{2}\sqrt{2-\sqrt{2}} & \sqrt{2+\sqrt{2}} & \sqrt{2}\sqrt{2-\sqrt{2}} \\ \sqrt{2+\sqrt{2}}-2 & \sqrt{2}\sqrt{2-\sqrt{2}} & \sqrt{2+\sqrt{2}}+2 \end{bmatrix}$$

which is neither tridiagonal nor Toeplitz.

In this chapter we introduce the concept of a structured geometric mean of PD matrices in such a way that if $A_1, \ldots, A_k \in \mathcal{S}$, their mean also belongs to $\mathcal{S}$. Given a subset $\mathcal{S}$ of $\mathcal{P}_n$ and matrices $A_1, \ldots, A_k \in \mathcal{S}$, we say that $\widehat{\mathbf{G}} \in \mathcal{S}$ is a *structured geometric mean* with respect to $\mathcal{S}$ of $A_1, \ldots, A_k$ if the Karcher cost function $f(X; A_1, \ldots, A_k)$ (3.1) takes its minimum value over $\mathcal{S}$ at $\widehat{\mathbf{G}}$. However, for a general subset $\mathcal{S}$, this minimizer is not necessarily unique anymore. The set of all structured geometric means of $A_1, \ldots, A_k$ (all minimizers of f) with respect to $\mathcal{S}$ is denoted by $\mathbf{G}_{\mathcal{S}} = \mathbf{G}_{\mathcal{S}}(A_1, \ldots, A_k)$.

We show that if $\mathcal{S}$ is closed (and nonempty) then $\mathbf{G}_{\mathcal{S}}$ is nonempty and the matrices $\widehat{\mathbf{G}} \in \mathbf{G}_{\mathcal{S}}$ satisfy most of the ALM axioms in a suitably adjusted form. For instance, the invariance under inversion property (4.2) turns into

$$\mathbf{G}_{\mathcal{S}}(A_1, \ldots, A_k) = \mathbf{G}_{\mathcal{S}^{-1}}\left(A_1^{-1}, \ldots, A_k^{-1}\right)^{-1},$$

where for a set $\mathcal{U} \subseteq \mathcal{P}_n$ we denote $\mathcal{U}^{-1} = \{X^{-1} : X \in \mathcal{U}\}$. That is, the inverse of any structured geometric mean of the matrices $A_1, \ldots, A_k \in \mathcal{S}$ with respect to $\mathcal{S}$ coincides with a structured mean of the inverses $A_1^{-1}, \ldots, A_k^{-1}$ with respect to the set $\mathcal{S}^{-1}$ where these inverses reside.

Moreover, we show that, in many interesting cases, structured geometric means can be characterized in terms of the PD solutions of a suitable vector equation and we provide algorithms for their computation.

The chapter is organized as follows. In Section 4.2 the Karcher cost function is examined with special focus on the existence of the minimizer over a closed set. The structured matrix mean itself is the subject of study in Section 4.3, where the theoretical properties it should satisfy are examined. Section 4.4 proposes two algorithms for computing a structured mean $\widehat{\mathbf{G}}$ in a linear space together with their convergence analysis. For one algorithm, it is shown that the convergence speed is independent of the condition numbers of the given matrices A_i and is faster when the condition numbers of the matrices $A_i^{-1/2}\widehat{\mathbf{G}}A_i^{-1/2}$ are smaller, for $i = 1, \ldots, k$. Because of its nature and its convergence properties, this algorithm can be viewed as the natural extension to the structured case of the Richardson-like algorithm for the computation of the Karcher mean, introduced and analyzed by Bini and Iannazzo [27]. Section 4.5 shows numerical experiments related to accuracy and speed for computing the structured matrix mean.

Here we introduce (and recall) some basic notation and properties that will be used in the rest of the chapter. Given a matrix A, we define $\lambda(A)$ the spectrum of A, that is, the set of all the eigenvalues of A, and $\rho(A) = \max_{\lambda \in \lambda(A)} |\lambda|$ the spectral radius of A. By A^H we denote the conjugate transpose of A. Moreover we denote by $\|A\|_F := (\mathrm{tr}(A^H A))^{1/2} = (\sum_{i,j} |a_{ij}|^2)^{1/2}$ the Euclidean (Frobenius) norm of A, and $\|A\|_2 = \rho(A^H A)^{1/2}$ is the spectral norm. Given a

matrix $A \in \mathbb{R}^{n\times n}$ (or $\mathbb{C}^{n\times n}$), the vec-operator is used to build $\text{vec}(A) \in \mathbb{R}^{n^2}$ (or $\mathbb{C}^{n^2}$), a long vector obtained by stacking the columns of A. Finally, in our analysis, we use the interaction between the vec-operator and the classical Kronecker product, given by $\text{vec}(ABC) = (C^T \otimes A)\,\text{vec}(B)$.

4.2 Existence of structured geometric means

In this section, we recall the geodesic convexity of the Karcher cost function (3.1), which was a key ingredient in the proof of its uniqueness. Afterwards, the existence of the structured geometric mean is proven, followed by an example which displays the important influence of geodesical convexity on the uniqueness of the structured mean.

4.2.1 Uniqueness of the Karcher mean for PD matrices and geodesical convexity

As we recall, the Riemannian geometry on $\mathcal{P}_n$ given by the inner product (3.3) turns out to be complete and every two PD matrices X and Y can be connected by a unique geodesic [21, 82]. The midpoint $X\#_{1/2}Y$ (2.4) of this geodesic coincides with the geometric mean of the two matrices [68, 82].

Previously (Section 3.2), we discussed two approaches to prove the existence of a unique minimizer of the Karcher cost function $f(X; A_1, \ldots, A_k)$ (3.1) over $\mathcal{P}_n$, with $A_1, \ldots, A_k \in \mathcal{P}_n$. One proof was obtained using the fact that $\mathcal{P}_n$, with the inner product (3.3), forms a Cartan-Hadamard manifold [36, 80, 82, 85], which is a Riemannian manifold, complete, simply connected and with non-positive sectional curvature everywhere. On such a Cartan-Hadamard manifold the Karcher mean[1] exists and is unique [37, 76, 80]. A second proof was based on the strict geodesical convexity of the cost function f, which for any two different matrices $X, Y \in \mathcal{P}_n$ is defined by

$$f(X\#_t Y) < (1-t)f(X) + tf(Y), \qquad 0 < t < 1. \tag{4.3}$$

For the function $f(X) = f(X; A)$ (corresponding to $k = 1$ in (3.1)), this property follows from Bhatia [21, Exercise 6.1.13], where it is stated that this function is strictly geodesically convex. The case $k > 1$ is obtained by summing up the k inequalities obtained by applying (4.3) to the functions $f(X) = f(X; A_i)$, for $i = 1, \ldots, k$, respectively.

[1] The Karcher mean corresponds to the center-of-mass as it is referred to in [74, 76, 80].

The notion of geodesical convexity in $\mathcal{P}_n$ is different from the customary convexity in the Euclidean space where one requires that

$$f((1-t)X + tY) \leq (1-t)f(X) + tf(Y), \qquad 0 \leq t \leq 1.$$

In fact, the Karcher cost function f is not convex in the traditional sense as the following example shows.

Example 4.2. Consider the set made of the unique matrix $A = 1$, and $x, y \in \mathbb{R}^{++} = \mathcal{P}_1$. We have $f(x) = \delta^2(x, A) = \log^2(x)$ which is not convex. On the other hand the function $\log^2(x)$ is strictly geodesically convex and this can be shown by an elementary argument: in fact, it is continuous and for $t = 1/2$, we can show

$$\begin{aligned}\delta^2(\sqrt{xy}, 1) = \log^2(\sqrt{xy}) &= \frac{1}{4}\left(\log^2 x + \log^2 y + 2\log x \log y\right) \\ &= \frac{1}{2}\left(\log^2 x + \log^2 y\right) - \frac{1}{4}\left(\log x - \log y\right)^2 \\ &< \frac{1}{2}\left(\log^2 x + \log^2 y\right) = \frac{1}{2}\left(\delta^2(x,1) + \delta^2(y,1)\right),\end{aligned}$$

which proves the scalar version of (4.3) at $t = 1/2$. By iteratively proving the inequality at the midpoints of the resulting intervals (e. g., $t = 1/4$ or $t = 3/4$ as the first next step) and using a continuity argument, the strict geodesical convexity can be shown on $0 < t < 1$.

Since f is strictly geodesically convex, it can be proved that it has a unique minimizer over any closed, geodesically convex subset $\mathcal{S}$ of $\mathcal{P}_n$, where we say that a subset $\mathcal{S} \subseteq \mathcal{P}_n$ is geodesically convex if for any $X, Y \in \mathcal{S}$, the entire geodesic $X\#_t Y$, $t \in [0,1]$ belongs to $\mathcal{S}$. Indeed, if X_1 and X_2 were two different matrices in $\mathcal{S}$ where f takes its minimum, then from (4.3) it would follow that $f(X_1\#_t X_2) < f(X_1) = f(X_2)$ for any $0 < t < 1$ which contradicts the assumption.

4.2.2 Existence of structured geometric means on a closed set

For a generic closed subset $\mathcal{U}$ of $\mathcal{P}_n$, which is not necessarily geodesically convex, we can prove the existence of a minimum by using the fact that our cost function $f(X)$ is continuous.

Theorem 4.3. *Let $\mathcal{U} \subseteq \mathcal{P}_n$ be a closed subset. Then for any $A_1, \ldots, A_k \in \mathcal{P}_n$ the function $f(X) = f(X; A_1, \ldots, A_k)$ has a minimum in $\mathcal{U}$.*

Proof. Consider the distance measures $\delta(X,Y)$, given in (2.3) (and recalled in Section 4.1), and $d(X,Y)$ (the Thompson metric [119]), given by

$$d(X,Y) = \left\| \log(Y^{-1/2} X Y^{-1/2}) \right\|_2 . \tag{4.4}$$

From the inequality between the Frobenius and the spectral norm, we have

$$d(X,Y) \leq \delta(X,Y) \leq \sqrt{n} d(X,Y)$$

for all $X, Y \in \mathcal{P}_n$. We also define the order interval $[X,Y]$ as

$$[X,Y] = \{ Z \in \mathcal{P}_n : X \leq Z \leq Y \},$$

for all $X, Y \in \mathcal{P}_n$ with $X \leq Y$ [61]. The order interval $[e^{-r}X, e^{r}X]$ turns out to be the closed ball of radius $r > 0$ around X with respect to the Thompson metric, thus the order interval $[e^{-r}I, e^{r}I]$ is a compact subset of $\mathcal{P}_n$.

Next, consider the set $Q_t = \{X \in \mathcal{P}_n : f(X) \leq t\}$. Because of the continuity of f, each Q_t is closed in $\mathcal{P}_n$. The definition of f also guarantees the boundedness of Q_t for each t with respect to the Riemannian metric δ. Since we have the above inequality of the distance measures, this implies that each Q_t is also bounded with respect to the Thompson metric. Hence it is possible to find some r such that Q_t is contained in the closed ball of radius r around I with respect to the Thompson metric, that is $Q_t \subset [e^{-r}I, e^{r}I]$. Since Q_t is now a closed subset of a compact set (in $\mathcal{P}_n$), it is also compact in $\mathcal{P}_n$.

Finally, define $B_t = Q_t \cap \mathcal{U}$, which is again a compact subset in $\mathcal{P}_n$ as it is the intersection of a compact and a closed set. The collection of all nonempty B_t is a descending family of nonempty, compact subsets of $\mathcal{P}_n$, which has a nonempty intersection. The elements in this intersection are the minimizers of f in $\mathcal{U}$.

□

In general, uniqueness of the point where $f(X)$ takes its minimum cannot be guaranteed. For instance, if both A and A^{-1} belong to $\mathcal{U}$ while $I = A\#_{1/2}A^{-1}$ does not (hence the set is not geodesically convex), then the function $f_1(X) := \delta^2(X,A) + \delta^2(X,A^{-1})$ reaches its minimum at a point $\widehat{\mathbf{G}} \neq I \in \mathcal{U}$. Clearly, $f_1(\widehat{\mathbf{G}}^{-1}) = f_1(\widehat{\mathbf{G}})$ and if $\widehat{\mathbf{G}}^{-1} \neq \widehat{\mathbf{G}}$ belongs to $\mathcal{U}$, then we have at least two distinct points of minimum. A more concrete example is the following.

Example 4.4. Consider the 2×2 matrices $A = I$ and $B = \begin{bmatrix} a & 0 \\ 0 & a^{-1} \end{bmatrix}$, where $a > 1$. Define the segment $\mathcal{U} = \{G(t) = A + t(B - A), \;\; t \in [0,1]\}$, which is closed and convex, but not geodesically convex. The function $f(t) = \delta^2(G(t), A) + \delta^2(G(t), B)$ takes the form $f(t) = \log^2((1-t)/a + t) + \log^2(a(1 -$

$t) + t) + \log^2((1-t) + t/a) + \log^2((1-t) + at)$ and is symmetric with respect to $t = 1/2$. For $a = 200$ the function has the graph shown in Figure 4.1 with a local maximum at $t = 1/2$ and two global minima close to the edges of the segment.

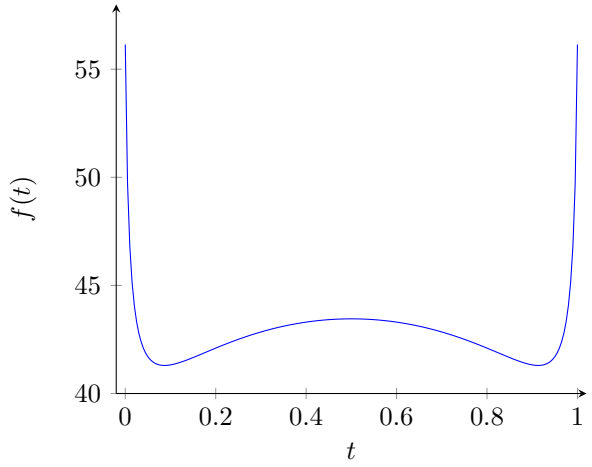


Figure 4.1: Graph of $f(t) = \delta^2(G(t), A) + \delta^2(G(t), B)$ for $G(t) = A + t(B - A)$ with $A = I$ and $B = \mathrm{diag}(200, 1/200)$.

4.3 A theoretical exploration of the structured geometric mean

In this section we discuss the relation between the structured and generic geometric mean, together with the adaptation of the generic properties to the structured setting. Afterwards, we derive a vector equation for the structured mean, which will serve as the foundation for the algorithms in the next section.

For convenience of notations and parametrizations, we will restrict our attention to the real case in the remainder of this chapter, hence the set $\mathcal{P}_n$ stands for the manifold of real PD matrices whose tangent space is the set of real, symmetric matrices $\mathcal{H}_n$. However, the results naturally generalize to the set of Hermitian PD matrices.

4.3.1 The geometric and structured geometric mean relation

The properties shown in Section 4.2 imply that a structured geometric mean with respect to $\mathcal{U}$, as defined in Section 4.1, always exists for any closed subset $\mathcal{U}$ of $\mathcal{P}_n$. In particular, this holds in the cases where $\mathcal{U} = \mathcal{S} \cap \mathcal{P}_n$, for any vector space $\mathcal{S}$, and also for $\mathcal{U}^{-1} := \mathcal{S}^{-1} \cap \mathcal{P}_n$, where we define $\mathcal{S}^{-1} = \left\{X^{-1} \;\middle|\; X \in \mathcal{S}, \det X \neq 0\right\}$. This captures a wide class of interesting structures emerging in applications, e. g., Toeplitz and band matrices, as well as their inverses.

More general structures are given in terms of a parametrization $\sigma(t) : \mathcal{V} \to \mathbb{R}^{n\times n}$, with σ a differentiable function defined in the open subset $\mathcal{V} \subseteq \mathbb{R}^q$, which we will refer to as the *parameter space*. The set $\mathcal{T} = \sigma(\mathcal{V})$ is the structure determined by σ. If σ is linear and $\mathcal{V} = \mathbb{R}^q$, then $\mathcal{T}$ is a linear space. Examples of sets $\mathcal{T}$ of interest which generally do not form a linear space are the set of matrices with a given displacement rank [26], the set of semiseparable [120], and quasiseparable matrices [51]. For an $n \times n$ symmetric Toeplitz matrix, a possible parametrization is given by

$$\sigma(t) = \sigma\big((t_0, t_1, \ldots, t_{n-1})\big) = \begin{bmatrix} t_0 & t_1 & \ldots & t_{n-1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{n-1} & \ldots & t_1 & t_0 \end{bmatrix}. \tag{4.5}$$

For a band matrix, we can, e. g., just store the nonzero-elements in a long vector and map them onto their exact locations. In the following, given a closed set $\mathcal{T}$ we let $\mathcal{U} = \mathcal{T} \cap \mathcal{P}_n$.

In Example 4.4 we illustrated that the minimum of the cost function restricted to a closed subset $\mathcal{U} \subseteq \mathcal{P}_n$ is not necessarily unique. For this reason, we consider the structured geometric mean $\mathbf{G}_{\mathcal{U}} = \mathbf{G}_{\mathcal{U}}(A_1, \ldots, A_k)$ of $A_1, \ldots, A_k \in \mathcal{U}$ as the set of matrices in $\mathcal{U}$ where the function $f(X)$ attains its minimum. Formally speaking, for $A_1, \ldots, A_k \in \mathcal{U}$, let $\widehat{\mathbf{g}} \in \mathbb{R}^q$ be such that $\widehat{\mathbf{G}} = \sigma(\widehat{\mathbf{g}}) \in \mathbf{G}_{\mathcal{U}}(A_1, \ldots, A_k)$, then

$$f(\sigma(\widehat{\mathbf{g}}); A_1, \ldots, A_k) = \min_{t \in \sigma^{-1}(\mathcal{U})} f(\sigma(t); A_1, \ldots, A_k).$$

Since $\mathcal{U} \subseteq \mathcal{P}_n$, the minimum over $\mathcal{P}_n$ is less than or equal to the minimum over $\mathcal{U}$. In general it will often happen that $\widehat{\mathbf{G}} \neq \mathbf{K}(A_1, \ldots, A_k)$ like in Example 4.1 (where $\mathbf{K}$ denotes the unstructured Karcher mean). However, if the set $\mathcal{U}$ is closed and geodesically convex then

$$\mathbf{G}_{\mathcal{U}}(A_1, \ldots, A_k) = \mathbf{K}(A_1, \ldots, A_k),$$

so the structured geometric mean coincides with the Karcher mean. An interesting case of a geodesically convex set is given by $\mathcal{U} = \mathcal{T} \cap \mathcal{P}_n$, when $\mathcal{T}$ is an algebra, i. e., a linear space closed under multiplication and inversion.

4.3.2 Properties of the geometric mean conveyed to the structured mean setting

Some desired properties for a matrix geometric mean were stated by Ando, Li and Mathias [6], and were listed in Section 2.2. These included, among others, consistency with scalars, permutation invariance, joint homogeneity, monotonicity, invariance under inversion, and invariance under congruence. Yet another property naturally desired of a geometric mean, but not required in the list of Ando, Li and Mathias, is the *repetition invariance*, that is, for any set of PD matrices $A_1, \ldots, A_k \in \mathcal{P}_n$,

$$\mathbf{G}(A_1, \ldots, A_k, A_1, \ldots, A_k) = \mathbf{G}(A_1, \ldots, A_k).$$

Now, we consider the properties of the structured geometric mean. Some properties such as the permutation invariance trivially hold, while others should be restated. In fact, in the generic case the structures we consider are neither invariant under inversion nor under congruence. That is because if $A \in \mathcal{U}$ then it is not necessarily true that $A^{-1} \in \mathcal{U}$ or $S^H A S \in \mathcal{U}$, for some invertible matrix S.

We start with the invariance under inversion as this is one of the most characteristic properties of the geometric mean. To this end we consider the set $\mathcal{T}^{-1} = \left\{ X^{-1} \;\middle|\; X \in \mathcal{T}, \det X \neq 0 \right\}$, parametrized with the function $\sigma(t)^{-1}$. Clearly, the intersection $\mathcal{U}$ of $\mathcal{T}$ with $\mathcal{P}_n$ always yields invertible matrices, so that $\mathcal{T}^{-1} \cap \mathcal{P}_n = \mathcal{U}^{-1}$.

According to our definition, the structured geometric mean of $A_1^{-1}, \ldots, A_k^{-1} \in \mathcal{U}^{-1}$ is given by the set $\mathbf{G}_{\mathcal{U}^{-1}}(A_1^{-1}, \ldots, A_k^{-1})$. For any $\widehat{\mathbf{G}} \in \mathbf{G}_{\mathcal{U}^{-1}}$, we have $\widehat{\mathbf{G}} = \sigma(\widehat{\mathbf{g}})^{-1}$ such that

$$f\left(\sigma\left(\widehat{\mathbf{g}}\right)^{-1}; A_1^{-1}, \ldots, A_k^{-1}\right) = \min_{t \in \sigma^{-1}(\mathcal{U})} f\left(\sigma(t)^{-1}; A_1^{-1}, \ldots, A_k^{-1}\right).$$

Since $\delta(A, B) = \delta\left(A^{-1}, B^{-1}\right)$, we get $f(X; A_1, \ldots, A_k) = f\left(X^{-1}; A_1^{-1}, \ldots, A_k^{-1}\right)$ so that

$$f\left(\sigma(\widehat{\mathbf{g}}); A_1, \ldots, A_k\right) = \min_{t \in \sigma^{-1}(\mathcal{U})} f\left(\sigma(t); A_1, \ldots, A_k\right)$$

and thus $\widehat{\mathbf{G}}^{-1} \in \mathbf{G}_{\mathcal{U}}(A_1, \ldots, A_k)$. Since $\widehat{\mathbf{G}}$ was chosen arbitrarily, and since $\mathcal{U}$ can be interchanged with $\mathcal{U}^{-1}$, we have the analogue of the invariance under

inversion for the structured geometric mean:

$$\mathbf{G}_{\mathcal{U}}(A_1,\ldots,A_k)^{-1} = \mathbf{G}_{\mathcal{U}^{-1}}\left(A_1^{-1},\ldots,A_k^{-1}\right).$$

In a similar manner we can restate the invariance under congruence in a structured style by defining, for any nonsingular S, the set $\mathcal{U}_S := S^H\mathcal{U}S = \{S^HXS \;\big|\; X \in \mathcal{U}\}$. The invariance under congruence is then understood as

$$\mathbf{G}_{\mathcal{U}_S}(S^HA_1S,\ldots,S^HA_kS) = S^H\mathbf{G}_{\mathcal{U}}(A_1,\ldots,A_k)S.$$

Joint homogeneity, in order to be defined, requires that the set $\mathcal{T}$ satisfies the following property:

$$X \in \mathcal{T} \quad \Rightarrow \quad \alpha X \in \mathcal{T}$$

for any scalar $\alpha > 0$. This property clearly holds if $\mathcal{T}$ is a linear space or the set formed by the inverses of the nonsingular matrices of a linear space. For these sets, the joint homogeneity holds.

Repetition invariance holds true as well by (3.1), since

$$f(X; A_1,\ldots,A_k,A_1,\ldots,A_k) = 2f(X; A_1,\ldots,A_k),$$

and a scaling of the cost function does not change the location of the minimizers over a (sub)set.

Regarding the remaining properties, we observe that the consistency with scalars is violated, as Example 4.1 shows. Nevertheless, weaker consistency properties hold, such as idempotency, namely $\mathbf{G}_{\mathcal{U}}(A,A,\ldots,A) = A$ for each structure $\mathcal{U}$ and $A \in \mathcal{U}$.

Finally, the properties related to the ordering of PD matrices such as monotonicity are not true as shown by the following numerical example.

Example 4.5. We consider the four Toeplitz matrices

$$T_1 = \begin{bmatrix} 1 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 \\ 1/2 & 1/2 & 1 \end{bmatrix}, \quad T_2 = T_1,$$

$$T_3 = \begin{bmatrix} 3/4 & 1/2 & 0 \\ 1/2 & 3/4 & 1/2 \\ 0 & 1/2 & 3/4 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix},$$

and, using the algorithms presented in the next sections, we compute a structured geometric mean $\mathbf{G}^{(\varepsilon)}_{\mathcal{T}_3^+}$ of the three matrices T_1, T_2 and $T_3 + \varepsilon S$ for various $\varepsilon \geq 0$,

where $\mathcal{T}_3^+$ is the set of 3×3 PD Toeplitz matrices. The norm of $\mathbf{G}^{(\varepsilon)}_{\mathcal{T}_3^+} - \mathbf{G}^{(0)}_{\mathcal{T}_3^+}$ becomes small as ε tends to 0 and we observe that $\mathbf{G}^{(\varepsilon)}_{\mathcal{T}_3^+} - \mathbf{G}^{(0)}_{\mathcal{T}_3^+}$ is not positive (semi)definite, while $T_3 + \varepsilon S \geq T_3$. This gives numerical evidence of the lack of monotonicity of a structured geometric mean. Furthermore, computing the arithmetic mean $\mathbf{A}$ of T_1, T_2 and T_3, one observes also that the expected inequality $\mathbf{A} \geq \mathbf{G}^{(0)}_{\mathcal{T}_3^+}$ does not hold in this case.

4.3.3 The structured mean as the solution(s) of a vector equation

We start from the Karcher mean, which for matrices $A_1, \ldots, A_k \in \mathcal{P}_n$ is obtained as the unique solution in $\mathcal{P}_n$ of the matrix equation

$$\sum_{i=1}^{k} \log\left(X A_i^{-1}\right) = 0. \tag{4.6}$$

Equation (4.6) is obtained using the fact that f is differentiable and has a minimum at the Karcher mean. Thus the Karcher mean satisfies the condition $\operatorname{grad}^{(\mathrm{sym})} f(X) = 0$, or equivalently, $\operatorname{grad}^{(\mathrm{pd})} f(X) = 0$, where both gradient expressions are given in Section 3.3.2 [74, 93].

In the general case, the restriction of f to a structure given by $\sigma(t),\ t \in \mathcal{V} \subseteq \mathbb{R}^q$, is investigated. For any minimum $\widehat{\mathbf{g}}$ (with corresponding $\sigma(\widehat{\mathbf{g}})$) not located at the boundary of the parameter space, the Euclidean gradient $\operatorname{grad}(f \circ \sigma)(t)$ (defined as in the classical, vector space setting) of the function with respect to t must be zero, so we are interested in the solutions of the vector equation $\operatorname{grad}(f \circ \sigma)(t) = 0$.

From the chain rule of derivation, we obtain

$$\operatorname{grad}(f \circ \sigma)(t) = \left(\sum_{i,j} \frac{\partial f(\sigma(t))}{\partial \sigma_{i,j}} \frac{\partial \sigma_{i,j}(t)}{\partial t_s} \right)_{s=1,\ldots,q} = 0$$

which leads to the vector equation

$$\sum_{i,j} \big(\Gamma(\sigma(t))\big)_{i,j} \frac{\partial \sigma_{i,j}(t)}{\partial t_s} = 0, \quad s = 1, \ldots, q, \tag{4.7}$$

where $\Gamma(X) := \frac{1}{2} \operatorname{grad}^{(\mathrm{sym})} f(X)$.

In the case where $\mathcal{T}$ is a linear space, the parametrization $\sigma(t)$ is linear and can be written in matrix form as

$$\operatorname{vec}(\sigma(t)) = Ut, \quad U \in \mathbb{R}^{n^2 \times q},$$

such that equation (4.7) turns into

$$U^T \operatorname{vec}(\Gamma(\sigma(t))) = 0, \tag{4.8}$$

$$\left[\Gamma(X) = X^{-1} \sum_{i=1}^{k} \log\left(X A_i^{-1}\right).\right.$$

When $\mathcal{T}$ denotes the set of symmetric Toeplitz matrices, the parametrization (4.5) leads to a matrix U having orthogonal columns, with $U^T U = D = \operatorname{diag}(n, 2(n-1), 2(n-2), \ldots, 2)$. In particular, for $n = 3$ we have

$$U^T = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right].$$

For $\mathcal{T}$ being the set of symmetric tridiagonal matrices the parametrization

$$\sigma(t) = \begin{bmatrix} t_1 & t_{n+1} & & & & \\ t_{n+1} & t_2 & t_{n+2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & t_{2n-2} & t_{n-1} & t_{2n-1} & \\ & & & t_{2n-1} & t_n & \end{bmatrix}$$

also leads to a matrix U having orthogonal columns. Moreover, $U^T U = \operatorname{diag}(I_n, 2I_{n-1})$. For $n = 3$, e. g., we have

$$U^T = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right].$$

4.4 Algorithms for structured geometric means in the linear case

We will give two algorithms for the computation of structured geometric means when they are defined as solutions $\widehat{\mathbf{g}}$ of a vector equation of the form (4.8), i. e., for linear matrix structures.

We first provide a general definition of a class of algorithms based on a preconditioned functional iteration, then we specialize to two algorithms characterized by two different preconditioners. From a linear algebraic point of view, the two algorithms are identical except for the preconditioning of the gradient. From a differential geometric point of view, the algorithms are derived similarly as Riemannian steepest descent methods by endowing $\mathcal{P}_n$ with different geometries.

The first algorithm, provided in Section 4.4.2 is derived by relying on the projection of the gradient with respect to the Euclidean geometry. The second algorithm, presented in Section 4.4.3, is obtained through projection with respect to the Riemannian inner product (3.3) on $\mathcal{P}_n$ described in Section 3.3.2.

4.4.1 A preconditioned functional iteration

Throughout this section we assume that $A_1, \ldots, A_k \in \mathcal{U}$, where $\mathcal{U} = \mathcal{T} \cap \mathcal{P}_n$ and $\mathcal{T}$ is a linear space with a parametrization $\sigma(t)$ such that $\operatorname{vec}(\sigma(t)) = Ut$, and $D = U^T U$.

The structured geometric mean $\mathbf{G}_{\mathcal{U}}$ is defined as the set of minimizers of the function $f(X; A_1, \ldots, A_k)$ over $\mathcal{U}$. These minimizers must be sought among the stationary points of the function f, that is, among the solutions to the vector equation (4.8).

Therefore, a way to design algorithms for computing structured means $\mathbf{G}_{\mathcal{U}}$ is to apply numerical techniques to solve the vector equation (4.8). We consider a preconditioned Richardson-like iteration constructed in the spirit of [27]. Let $V(X)$ be a nonsingular and sufficiently differentiable matrix function and define

$$\begin{aligned} &t_{j+1} = \varphi(t_j), \quad j = 0, 1, \ldots, \\ &\left[\begin{array}{l} \varphi(t) = t - \theta S(t), \\ S(t) = V(\sigma(t))^{-1} U^T \operatorname{vec}(\Gamma(\sigma(t))), \end{array} \right. \end{aligned} \tag{4.9}$$

where θ is a parameter introduced to enhance convergence, $V(\sigma(t))$ is a preconditioner and t_0 is a given vector such that $\sigma(t_0)$ is PD. Observe that the fixed points of $\varphi(t)$ are the solutions of the vector equation (4.8) and, if convergent, such a solution can be found as the limit of the sequence t_j.

In the following, given a matrix function $h(X)$, where X and $h(X)$ are $n \times n$ matrices, we denote by $J_h(Y)$ the $n^2 \times n^2$ Jacobian matrix of $\operatorname{vec}(h(X))$ with respect to the variable $\operatorname{vec}(X)$ computed at $X = Y$. Similarly we denote by $J_{h\circ\sigma}(t_Y)$ the $n^2 \times q$ Jacobian of the composed function $\operatorname{vec}(h(\sigma(t)))$ with respect to the variables $(t_1, \ldots, t_q)$ at $t = t_Y$. The function indicated by the subscript

as well as the variable between parentheses will specify whether the derivatives are taken w. r. t. the matrix variable X or the vector variable t.

Observe that if $V(\sigma(t))$ is chosen as the Jacobian of $U^T \operatorname{vec}(\Gamma(\sigma(t)))$, then (4.9) coincides with Newton's iteration.

If $t_{\widehat{\mathbf{G}}}$ is a solution of (4.8), corresponding to $\sigma(t_{\widehat{\mathbf{G}}}) = \widehat{\mathbf{G}} \in \mathbf{G}_{\mathcal{U}}$, and if t_j is sufficiently near to $t_{\widehat{\mathbf{G}}}$, then

$$t_{j+1} - t_{\widehat{\mathbf{G}}} = J_\varphi(t_{\widehat{\mathbf{G}}})(t_j - t_{\widehat{\mathbf{G}}}) + O\left(\|t_j - t_{\widehat{\mathbf{G}}}\|^2\right),$$

so that in order to study the local convergence of this sequence it is sufficient to estimate the spectral radius ρ or any induced norm of $J_\varphi(t_{\widehat{\mathbf{G}}})$ and determine θ in such a way that $\rho(J_\varphi(t_{\widehat{\mathbf{G}}})) < 1$. Notice that the Jacobian of $\varphi(t)$ at $t = t_{\widehat{\mathbf{G}}}$ is given by $I - \theta K$ where $K = J_S(t_{\widehat{\mathbf{G}}})$ is the Jacobian of $S(t)$ at $t = t_{\widehat{\mathbf{G}}}$. Therefore, if we can find a preconditioner $V(t)$ such that K has real positive eigenvalues with minimum and maximum eigenvalues $\kappa_{\min}$ and $\kappa_{\max}$ respectively, then the choice $\theta = 2/(\kappa_{\min} + \kappa_{\max})$ ensures local convergence and provides the minimum spectral radius of $J_\varphi(t_{\widehat{\mathbf{G}}})$ given by

$$\rho(J_\varphi(t_{\widehat{\mathbf{G}}})) = \frac{\kappa_{\max} - \kappa_{\min}}{\kappa_{\max} + \kappa_{\min}} = \frac{\mu - 1}{\mu + 1} < 1,$$

where $\mu = \kappa_{\max}/\kappa_{\min}$. Moreover, any values $\hat{\kappa}_{\min} \leq \hat{\kappa}_{\max}$ such that $\hat{\kappa}_{\min} \leq \kappa_{\min} \leq \kappa_{\max} \leq \hat{\kappa}_{\max}$ can be used instead of $\kappa_{\min}$ and $\kappa_{\max}$ to determine a value $\hat{\theta} = 2/(\hat{\kappa}_{\min} + \hat{\kappa}_{\max})$ which ensures convergence. Also notice that if μ is closer to 1, the iterations in (4.9) will converge faster.

Therefore, our goal is to perform a spectral analysis of K and to find an upper bound to the ratio $\mu = \kappa_{\max}/\kappa_{\min}$, assuming that all the eigenvalues of K are real positive. From the composition rule of derivatives we find that

$$K = V(\sigma(t_{\widehat{\mathbf{G}}}))^{-1} U^T J_\Gamma(\widehat{\mathbf{G}}) U + \left(\left(U^T \operatorname{vec}\left(\Gamma(\sigma(t_{\widehat{\mathbf{G}}}))\right)\right)^T \otimes I_q\right) J_{V(\sigma(\cdot))^{-1}}(t_{\widehat{\mathbf{G}}}),$$

and since $U^T \operatorname{vec}(\Gamma(\sigma(t_{\widehat{\mathbf{G}}}))) = 0$, it follows that

$$K = V(\sigma(t_{\widehat{\mathbf{G}}}))^{-1} U^T J_\Gamma(\widehat{\mathbf{G}}) U. \tag{4.10}$$

To evaluate $J_\Gamma(\widehat{\mathbf{G}})$, we recall that $\Gamma(X) = \sum_{i=1}^k X^{-1} \log(X A_i^{-1})$, so that it is sufficient to determine the formal expression of $J_\psi(\widehat{\mathbf{G}})$ for $\psi(X; A) = X^{-1} \log(X A^{-1})$ for a generic A and then to write $J_\Gamma(\widehat{\mathbf{G}}) = \sum_{i=1}^k J_{\psi_i}(\widehat{\mathbf{G}})$, where $\psi_i(X) := \psi(X; A_i)$. In order to evaluate $J_\psi(\widehat{\mathbf{G}})$, we rely on the definition

of the Fréchet derivative of a matrix function $f(X)$ at X in the direction E,

$$\mathrm{D}f(X)[E] = \lim_{t\to 0} \frac{f(X+tE)-f(X)}{t} = \frac{\mathrm{d}}{\mathrm{d}t} f(X+tE)\Big|_{t=0}.$$

In fact, the $n^2 \times n^2$ Jacobian matrix $J_f(X)$ of the vector function $\mathrm{vec} \circ f \circ \mathrm{vec}^{-1}$ at $\mathrm{vec}(X)$ is related to the Fréchet derivative by the equation

$$\mathrm{vec}(\mathrm{D}f(X)[E]) = J_f(X)\,\mathrm{vec}(E). \tag{4.11}$$

In the following, we will use a number of properties of the Fréchet derivative of matrix functions, as well as the derivatives of some specific functions (see Section 1.2.2). For the derivative of the exponential function we have [64, Eq. 10.17a]

$$J_{\exp}(Y) = (I \otimes \exp Y)\, \beta\left(Y^T \otimes I - I \otimes Y\right),$$

$$\left[\beta(z) = (e^z - 1)/z, \quad \beta(0) = 1.\right.$$

Since $J_{\log}(X) = J_{\exp}(Y)^{-1}$ for $Y = \log X$, we find that

$$J_{\log}(X) = \gamma\left(\log\left(X^T\right) \otimes I - I \otimes \log X\right)\left(I \otimes X^{-1}\right), \tag{4.12}$$

$$\left[\gamma(z) = z/(e^z - 1), \quad \gamma(0) = 1.\right.$$

We are now ready to provide an explicit expression of the Fréchet derivative of the function $\psi(X) = X^{-1}\log\left(XA^{-1}\right)$ and of the Jacobian $J_\psi(X)$.

Lemma 4.6. *Let $\psi(X) = X^{-1}\log\left(XA^{-1}\right)$. Assume that A, X are PD. For the matrix $J_\psi(X)$ such that $\mathrm{vec}\left(\mathrm{D}\psi(X)[E]\right) = J_\psi(X)\,\mathrm{vec}(E)$ we have*

$$J_\psi(X) = -X^{-1}\log\left(XA^{-1}\right) \otimes X^{-1} + (A^{-1} \otimes X^{-1})\gamma(W)(I \otimes AX^{-1}),$$

$$\left[\begin{array}{l} W = \log\left(XA^{-1}\right) \otimes I - I \otimes \log\left(XA^{-1}\right), \\ \gamma(z) = z/(e^z - 1),\ \gamma(0) = 1. \end{array}\right.$$

Proof. Since $h(X) := \log\left(XA^{-1}\right)$ is the composition of $h_1(X) = \log(X)$ and $h_2(X) = XA^{-1}$, we get by the chain rule of differentiation

$$\mathrm{D}h(X)[E] = \mathrm{Dlog}\left(XA^{-1}\right)[EA^{-1}].$$

As $\psi(X)$ is the product of $g(X) = X^{-1}$ and $h(X)$, the product rule of differentiation gives us

$$\mathrm{D}\psi(X)[E] = -X^{-1}EX^{-1}\log\left(XA^{-1}\right) + X^{-1}\,\mathrm{D}h(X)[E].$$

Combining the latter two equations yields

$$\mathrm{D}\psi(X)[E] = -X^{-1}EX^{-1}\log\left(XA^{-1}\right) + X^{-1}\,\mathrm{Dlog}\left(XA^{-1}\right)[EA^{-1}].$$

By (4.11) and the interaction between the vec-operator and the Kronecker product, we find that the matrix $J_\psi(X)$ representing $\mathrm{D}\psi(X)$ is given by

$$J_\psi(X) = -\left(X^{-1}\log\left(XA^{-1}\right)\right)^T \otimes X^{-1} + \left(I \otimes X^{-1}\right) J_{\log}\left(XA^{-1}\right)\left(A^{-T} \otimes I\right).$$

Replacing (4.12) in the equation above and using the fact that $A = A^T$ and $X = X^T$ yields

$$\begin{aligned} J_\psi(X) = &-\log\left(A^{-1}X\right)X^{-1} \otimes X^{-1} \\ &+ (I \otimes X^{-1})\gamma\left(\log\left(A^{-1}X\right) \otimes I - I \otimes \log\left(XA^{-1}\right)\right)\left(A^{-1} \otimes AX^{-1}\right). \end{aligned}$$

Using the fact that $W\log(V)W^{-1} = \log\left(WVW^{-1}\right)$, the first term can be written as $-X^{-1}\log\left(XA^{-1}\right) \otimes X^{-1}$. The second term can be written as $(I \otimes X^{-1})(A^{-1} \otimes I)\gamma\left(\log\left(XA^{-1}\right) \otimes I - I \otimes \log\left(XA^{-1}\right)\right)\left(I \otimes AX^{-1}\right)$, which completes the proof. □

Recall that $\Gamma(X) = \sum_{i=1}^k \psi\left(X, A_i\right)$ and $\widehat{\mathbf{G}}^{-1}\sum_{i=1}^k \log(\widehat{\mathbf{G}}A_i^{-1}) = 0$, for $\widehat{\mathbf{G}} = \sigma(t_{\widehat{\mathbf{G}}})$. Then by Lemma 4.6, we obtain the following formula for the Jacobian $J_\Gamma(\sigma(t_{\widehat{\mathbf{G}}}))$:

$$J_\Gamma(\widehat{\mathbf{G}}) = \left(I \otimes \widehat{\mathbf{G}}^{-1}\right) H \left(I \otimes \widehat{\mathbf{G}}^{-1}\right),$$

$$\left[\begin{array}{l} H = \sum_{i=1}^k H_i, \\ H_i = \left(A_i^{-1} \otimes I\right)\gamma\left(\log\left(\widehat{\mathbf{G}}A_i^{-1}\right) \otimes I - I \otimes \log\left(\widehat{\mathbf{G}}A_i^{-1}\right)\right)(I \otimes A_i). \end{array}\right.$$

Moreover, by using the properties of the Kronecker product and the relation $\log(\widehat{\mathbf{G}}A^{-1}) = A^{1/2}\log(A^{-1/2}\widehat{\mathbf{G}}A^{-1/2})A^{-1/2}$, we can write

$$H_i = \left(A_i^{-1/2} \otimes A_i^{1/2}\right)\gamma\left(\log M_i \otimes I - I \otimes \log M_i\right)\left(A_i^{-1/2} \otimes A_i^{1/2}\right),$$

$$\left[M_i = A_i^{-1/2}\widehat{\mathbf{G}}A_i^{-1/2}.\right.$$

From this expression it turns out that H_i is PD, and from (4.10) we find that $J_S(t_{\widehat{\mathbf{G}}})$ is the product of the matrices $V(\sigma(t_G))^{-1}$ and the PD matrix $U^T(I \otimes \widehat{\mathbf{G}}^{-1}) \sum_{i=1}^k H_i (I \otimes \widehat{\mathbf{G}}^{-1})U$.

Thus we may conclude with the following

Theorem 4.7. *The Jacobian K of the function $S(t)$ in (4.10) at $\sigma(t_{\widehat{\mathbf{G}}}) = \widehat{\mathbf{G}}$ is given by*

$$K = V^{-1}U^T \left(I \otimes \widehat{\mathbf{G}}^{-1}\right) H \left(I \otimes \widehat{\mathbf{G}}^{-1}\right) U,$$

$$\left[\begin{array}{l} H = \sum_{i=1}^k H_i, \\ H_i = \left(A_i^{-1/2} \otimes A_i^{1/2}\right)\gamma\left(\log M_i \otimes I - I \otimes \log M_i\right)\left(A_i^{-1/2} \otimes A_i^{1/2}\right), \\ M_i = A_i^{-1/2}\widehat{\mathbf{G}}A_i^{-1/2}, \\ \gamma(z) = z/(e^z - 1),\ \gamma(0) = 1. \end{array}\right.$$

Moreover, the eigenvalues of K are the solutions κ of the equation

$$\det\left(\kappa V - U^T \left(I \otimes \widehat{\mathbf{G}}^{-1}\right) H \left(I \otimes \widehat{\mathbf{G}}^{-1}\right) U\right) = 0.$$

4.4.2 A basic preconditioner

The simplest choice for the preconditioner $V(t)$ in (4.9) is $V(t) = U^TU = D$. This corresponds to projecting the gradient of the cost function $f(X; A_1, \ldots, A_k)$ onto the tangent space of the set $\mathcal{U}$ according to the Euclidean inner product. The eigenvalue problem $\det(\kappa I - K) = 0$ turns into the generalized q-dimensional symmetric eigenvalue problem

$$\det\left(U^T\left(\kappa I - (I \otimes \widehat{\mathbf{G}}^{-1})H(I \otimes \widehat{\mathbf{G}}^{-1})\right)U\right) = 0.$$

This problem is the projection on the space spanned by the columns of U of the problem $\det(\nu I - (I \otimes \widehat{\mathbf{G}}^{-1})H(I \otimes \widehat{\mathbf{G}}^{-1})) = 0$, which has real positive solutions.

Now we recall the following result [20], valid for general PD matrices A, B, which relates the generalized eigenvalues of the pair (A, B) to the ones of the projected pair (U^TAU, U^TBU).

Lemma 4.8. *Let A, B be $n \times n$ PD matrices and U an $n \times m$ matrix. Then the generalized eigenvalues of the pair (U^TAU, U^TBU), which solve the equation $\det\big(U^T(A - \kappa B)U\big) = 0$, are real positive and lie in between the maximum and minimum eigenvalues λ of the pair (A, B), which satisfy $\det(A - \lambda B) = 0$. Moreover, the extreme eigenvalues $\lambda_{\min}, \lambda_{\max}$ of the pair (A, B) are such that $\alpha_{\min}/\beta_{\max} \le \lambda_{\min} \le \lambda_{\max} \le \alpha_{\max}/\beta_{\min}$, where $\alpha_{\min}, \alpha_{\max}, \beta_{\min}, \beta_{\max}$ are the minimum and maximum eigenvalues of the matrices A and B, respectively.*

Proof. The condition $\det(\lambda B - A) = 0$ is equivalent to $\det(\lambda I - B^{-1/2}AB^{-1/2}) = 0$, which has real positive solutions since $B^{-1/2}AB^{-1/2}$ is a PD matrix. The remaining part of the lemma follows from the fact that maximum and minimum eigenvalues of the larger and smaller problems coincide with maximum and minimum value of the Rayleigh quotient x^TAx/x^TBx for $x \in \mathbb{R}^n$, and for $x \in \text{span}(U)$, respectively. □

A first consequence of the above lemma is that the extreme eigenvalues $\kappa_{\min}$ and $\kappa_{\max}$ of K are in between the maximum and the minimum eigenvalue of the n^2-dimensional symmetric PD matrix $Y = (I \otimes \widehat{\mathbf{G}}^{-1})H(I \otimes \widehat{\mathbf{G}}^{-1})$, so that the ratio μ between the maximum and minimum eigenvalue of K is less than or equal to the condition number $\mu(Y)$ of the symmetric matrix Y. We write $Y = \sum_{i=1}^{k} Y_i$, where

$$Y_i = \big(A_i^{-1/2} \otimes A_i^{-1/2}M_i^{-1}\big)\gamma\left(\log M_i \otimes I - I \otimes \log M_i\right)\big(A_i^{-1/2} \otimes M_i^{-1}A_i^{-1/2}\big),$$

with $M_i, i = 1, \ldots, k$, and γ as defined in Theorem 4.7. We can now combine Lemma 4.8 with the property that states that the sum of the minimum (maximum) eigenvalues of a set of matrices is smaller (larger) than the minimum (maximum) eigenvalue of their matrix sum. Doing so, we find that $\widehat{k}_{\min} := \sum_{i=1}^{k} \lambda_{\min}^{(i)} \leq \kappa_{\min}$ and $\widehat{k}_{\max} := \sum_{i=1}^{k} \lambda_{\max}^{(i)} \geq \kappa_{\max}$, where $\lambda_{\min}^{(i)}$ and $\lambda_{\max}^{(i)}$ are the minimum and the maximum eigenvalues of Y_i.

Moreover, from Lemma 4.8 and from the expression above for Y_i it follows that $\lambda_{\min}^{(i)} \geq \gamma_{\min}^{(i)}/(\alpha_{\max}^{(i)})^2$, $\lambda_{\max}^{(i)} \leq \gamma_{\max}^{(i)}/(\alpha_{\min}^{(i)})^2$, where $\alpha_{\min}^{(i)}$, $\alpha_{\max}^{(i)}$ are the minimum and the maximum eigenvalues of A_i, respectively, while $\gamma_{\min}^{(i)}$ and $\gamma_{\max}^{(i)}$ are the minimum and maximum eigenvalues of $(I \otimes M_i^{-1})\gamma(\log M_i \otimes I - I \otimes \log M_i)(I \otimes M_i^{-1})$, respectively.

From the properties of the matrix function $\gamma(\cdot)$ and from the properties of the Kronecker product we find that the eigenvalues of the latter matrix can be given explicitly in terms of the eigenvalues $\nu_r^{(i)}$ of the matrix M_i. In fact, they coincide with $\frac{1}{(\nu_s^{(i)})^2}\big(\log t_{r,s}^{(i)}\big)/\big(t_{r,s}^{(i)} - 1\big)$, where $t_{r,s}^{(i)} = \frac{\nu_r^{(i)}}{\nu_s^{(i)}}$, $r, s = 1, \ldots, n$.

Since the function $(\log t)/(t-1)$ is monotonically decreasing, its minimum and maximum are

$$\eta_{\min}^{(i)} = \big(\log \mu^{(i)}\big)/\big(\mu^{(i)} - 1\big),$$

$$\eta_{\max}^{(i)} = \log\big(1/\mu^{(i)}\big)/\big(1/\mu^{(i)} - 1\big) = \mu^{(i)}\big(\log \mu^{(i)}\big)/\big(\mu^{(i)} - 1\big),$$

for $\mu^{(i)} = \mu(M_i)$ the spectral condition number of M_i. Additionally, taking the factor $(\nu^{(i)})^{-2}$ into consideration gives

$$\gamma_{\min}^{(i)} \geq \eta_{\min}^{(i)} \, (\nu_{\max}^{(i)})^{-2},$$

$$\gamma_{\max}^{(i)} \leq \eta_{\max}^{(i)} \, (\nu_{\min}^{(i)})^{-2} \leq \mu^{(i)} (\nu_{\min}^{(i)})^{-2},$$

where $\nu_{\min}^{(i)}$ and $\nu_{\max}^{(i)}$ represent respectively the minimum and maximum eigenvalue of M_i.

Therefore, we may conclude that the eigenvalues of K are bounded by $\widetilde{\kappa}_{\min} := \sum_{i=1}^{k} \eta_{\min}^{(i)} / (\nu_{\max}^{(i)} \, \alpha_{\max}^{(i)})^2$ and $\widetilde{\kappa}_{\max} := \sum_{i=1}^{k} \eta_{\max}^{(i)} / (\nu_{\min}^{(i)} \, \alpha_{\min}^{(i)})^2$.

Observe that this bound gets worse when either some matrix A_i is ill-conditioned or if some matrix $A_i^{-1/2} \widehat{\mathbf{G}} A_i^{-1/2}$ is ill-conditioned. The latter case cannot occur if the matrices A_i do not differ much from $\widehat{\mathbf{G}}$. However, since this bound depends on the conditioning of the matrices A_i, the algorithm remains very inefficient as long as some A_i is ill-conditioned. This drawback is overcome in the next section, where we design a more effective preconditioner.

4.4.3 A preconditioner based on differential geometry

The Karcher mean for PD matrices inherits a beautiful interpretation in terms of differential geometry, as discussed in Chapter 3. It can be considered as the center of mass for a well chosen inner product on the manifold of PD matrices [76, 80]. In this section, we consider two approaches inspired by this idea.

When considering a manifold optimization approach, the intersection $\mathcal{U}$ of a linear space $\mathcal{T}$ with the manifold of PD matrices $\mathcal{P}_n$ can be viewed as a Riemannian submanifold of $\mathcal{P}_n$ itself, which in turn is called the *enveloping space*. This entails that the inner product from this enveloping space is induced on the submanifold. An immediate consequence is that the gradient of the cost function for the submanifold is given by the orthogonal projection (with respect to the inner product) of the gradient for the enveloping space. Similar to the space of symmetric matrices being the tangent space to the manifold of PD matrices, the intersection $\mathcal{V}$ of the linear space $\mathcal{T}$ with the space of symmetric matrices is the tangent space to $\mathcal{U}$.

First consider the manifold of PD matrices endowed with the Euclidean inner product $\langle \xi_X, \eta_X \rangle_X = \operatorname{tr}(\xi_X \eta_X)$, with ξ_X and η_X symmetric, and X a PD matrix. Note that even though this inner product is independent of X, the subscript notation is kept for consistency. In this case, the orthogonal projection of a

symmetric matrix ξ_X onto $\mathcal{T}$ gives a matrix T_X, with

$$\operatorname{vec}(T_X) = U\left(U^T U\right)^{-1} U^T \operatorname{vec}(\xi_X),$$

or $\operatorname{vec}(T_X) = U t_X$, with

$$t_X = \left(U^T U\right)^{-1} U^T \operatorname{vec}(\xi_X). \tag{4.13}$$

The expression for the gradient of the Karcher cost function, corresponding to the Euclidean inner product, is known for the manifold of PD matrices (Section 3.3.2) and is given by

$$\begin{aligned}\operatorname{grad}^{(\mathrm{sym})} f(X; A_1, \ldots, A_k) =& 2X^{-1} \sum_{i=1}^{k} \log\left(X A_i^{-1}\right) \\ =& 2X^{-1/2} \sum_{i=1}^{k} \log\left(X^{1/2} A_i^{-1} X^{1/2}\right) X^{-1/2}.\end{aligned} \tag{4.14}$$

The gradient naturally defines the direction of steepest ascent. Iteratively moving in the opposite direction results in the steepest descent algorithm, discussed in Section 3.4.1 and graphically depicted in Figure 4.2. In this figure, the thin red lines depict the contour lines, the blue arrows the gradients, and the green curves the retractions to the manifold. Recall from Section 3.3.3 that a retraction is the manifold equivalent of taking a step in a given direction.

Observe that for $\mathcal{P}_n$ immersed in the set of symmetric matrices, the tangent space at a point is the whole set of symmetric matrices. So one can consider the basic retraction $R_X(\xi_X) = X + \xi_X$ for a sufficiently small symmetric matrix ξ_X.

Entering now the gradient (4.14) in projection (4.13) and applying a gradient descent method with the basic retraction $R_X(\xi_X) = X + \xi_X$, we arrive exactly at the previous Richardson-like algorithm for finding the fixed points of the function φ (4.9).

However, since the function f to be minimized is defined through the distance (4.1), it is more natural to consider the manifold of PD matrices endowed with the corresponding inner product $\langle \xi_X, \eta_X \rangle_X = \operatorname{tr}\left(\xi_X X^{-1} \eta_X X^{-1}\right)$, with ξ_X, η_X, and X as before. In this case, the gradient for the enveloping space is known to

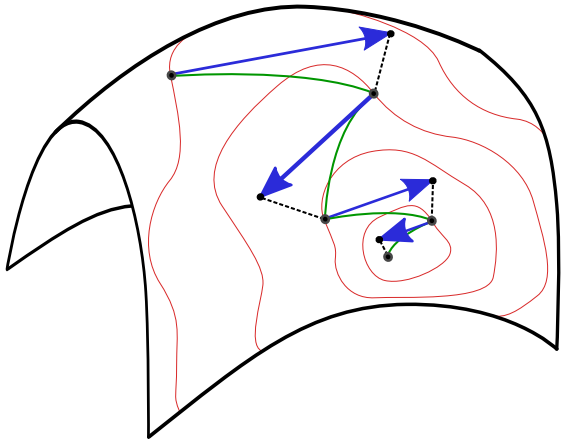


Figure 4.2: Simplified representation of the steepest descent algorithm.

be (Section 3.3.2)

$$\begin{aligned}\operatorname{grad}^{(\mathrm{pd})} f(X; A_1, \ldots, A_k) =& 2X \sum_{i=1}^{k} \log\left(A_i^{-1} X\right) \\ =& 2X^{1/2} \sum_{i=1}^{k} \log\left(X^{1/2} A_i^{-1} X^{1/2}\right) X^{1/2}.\end{aligned}$$

Note the difference with (4.14) in the sign of the power of the outer X-factors in the two bottom expressions.

The orthogonal projection T_X onto the intersection $\mathcal{V}$ (of $\mathcal{T}$ and the space of symmetric matrices) of this gradient, with respect to the Riemannian scalar product, can be found as the solution of the equations

$$\begin{aligned}&\operatorname{grad}^{(\mathrm{pd})} f(X) = T_X + C_X, \\ &\langle C_X, K_X \rangle_X = \operatorname{tr}\left(C_X X^{-1} K_X X^{-1}\right) = 0, \quad \text{for every } K_X \in \mathcal{V}.\end{aligned}$$

Writing again $\operatorname{vec}(T_X) = U t_X$, we find in parameter space

$$t_X = \left(U^T \left(X^{-1} \otimes X^{-1}\right) U\right)^{-1} U^T \left(X^{-1} \otimes X^{-1}\right) \operatorname{vec}(\operatorname{grad}^{(\mathrm{pd})} f(X)). \quad (4.15)$$

The factor $U^T \left(X^{-1} \otimes X^{-1}\right) U$ is recurring and is abbreviated as D_X, where the subscript refers to the intrinsic variable X. Observe that this Riemannian

orthogonal projection can be seen as a Euclidean oblique projection where the two bases of the subspace are the columns of U and $(X^{-1}\otimes X^{-1})U$, respectively.

Using this expression, it is possible to define another gradient descent method where we are now searching the fixed points of the function

$$\varphi(t) = t - \theta D_{\sigma(t)}^{-1}\, U^T \left(\sigma(t)^{-1}\otimes\sigma(t)^{-1}\right) \operatorname{vec}\Big(\sigma(t)\sum_{i=1}^{k}\log\left(A_i^{-1}\sigma(t)\right)\Big). \quad (4.16)$$

Relying on the interaction between the Kronecker product and the vectorization, we find that $(\sigma^{-1}\otimes\sigma^{-1})\operatorname{vec}(\sigma\sum_{i=1}^{k}\log(A_i^{-1}\sigma)) = \operatorname{vec}(\sum_{i=1}^{k}\log(A_i^{-1}\sigma)\sigma^{-1})$. Applying a property of the matrix logarithm we may rewrite the latter expression as $\operatorname{vec}(\sigma^{-1}\sum_{i=1}^{k}\log(\sigma A_i^{-1}))$. This way, equation (4.16) takes the form of (4.9) with

$$V(\sigma(t)) = U^T\left(\sigma(t)^{-1}\otimes\sigma(t)^{-1}\right)U.$$

To analyze the convergence of (4.9) with this choice for $V(\sigma(t))$, we have to analyze the eigenvalues of the Jacobian $K = J_S(t_{\widehat{\mathbf{G}}})$ of $S(t)$ in (4.9) where the equation $\det(\kappa I - K) = 0$ takes the form of the following generalized eigenvalue problem

$$\det\Big(U^T\Big(\kappa\big(\widehat{\mathbf{G}}^{-1}\otimes\widehat{\mathbf{G}}^{-1}\big) - \big(I\otimes\widehat{\mathbf{G}}^{-1}\big)H\big(I\otimes\widehat{\mathbf{G}}^{-1}\big)\Big)U\Big) = 0. \quad (4.17)$$

Since the two matrices in equation (4.17) are PD, and because of Lemma 4.8, the solutions of this generalized eigenvalue problem are real positive and are located in between the minimum and the maximum solution of the larger problem

$$\det\Big(\lambda\big(\widehat{\mathbf{G}}^{-1}\otimes\widehat{\mathbf{G}}^{-1}\big) - \big(I\otimes\widehat{\mathbf{G}}^{-1}\big)H\big(I\otimes\widehat{\mathbf{G}}^{-1}\big)\Big) = 0,$$

which in turn can be rewritten as a standard eigenvalue problem

$$\det\Big(\lambda I - \big(\widehat{\mathbf{G}}^{1/2}\otimes\widehat{\mathbf{G}}^{-1/2}\big)H\big(\widehat{\mathbf{G}}^{1/2}\otimes\widehat{\mathbf{G}}^{-1/2}\big)\Big) = 0.$$

Since $H = \sum_{i=1}^{k} H_i$, and the matrices H_i are real symmetric, the eigenvalues of this problem are located in between the sum of the minimum and the sum of the maximum eigenvalues of each subproblem

$$\det\Big(\lambda I - \big(\widehat{\mathbf{G}}^{1/2}\otimes\widehat{\mathbf{G}}^{-1/2}\big)H_i\big(\widehat{\mathbf{G}}^{1/2}\otimes\widehat{\mathbf{G}}^{-1/2}\big)\Big) = 0, \quad (4.18)$$

which is equivalent to the expressions $\det\big(\lambda(\widehat{\mathbf{G}}^{-1}\otimes\widehat{\mathbf{G}}) - H_i\big) = 0$ and $\det\big(\lambda I - (\widehat{\mathbf{G}}\otimes I)H_i(I\otimes\widehat{\mathbf{G}}^{-1})\big) = 0$. The matrix in the latter expression is similar to

$\left(A_i^{-1/2} \otimes A_i^{-1/2}\right)\left(\widehat{\mathbf{G}} \otimes I\right) H_i \left(I \otimes \widehat{\mathbf{G}}^{-1}\right)\left(A_i^{1/2} \otimes A_i^{1/2}\right)$, which, using the expression of H_i provided in Theorem 4.7, can be written as

$$(M_i \otimes I)\, \gamma\, (\log M_i \otimes I - I \otimes \log M_i)\, \left(I \otimes M_i^{-1}\right).$$

This way, the eigenvalues of (4.18) can be explicitly given in terms of the eigenvalues $\nu_r^{(i)}$ of the matrix M_i. In fact, they coincide with the values $t_{r,s}^{(i)}\left(\log t_{r,s}^{(i)}\right)/\left(t_{r,s}^{(i)} - 1\right)$ where $t_{r,s}^{(i)} = \frac{\nu_r^{(i)}}{\nu_s^{(i)}}$, $r, s = 1, \ldots, n$.

Since the function $t(\log t)/(t-1)$ is monotone, we find for the minimum and maximum solution to (4.18) the values

$$\eta_{\min}^{(i)} = \left(1/\mu^{(i)}\right) \log\left(1/\mu^{(i)}\right)/\left(1/\mu^{(i)} - 1\right) = \left(\log \mu^{(i)}\right)/\left(\mu^{(i)} - 1\right),$$

$$\eta_{\max}^{(i)} = \mu^{(i)} \left(\log \mu^{(i)}\right)/\left(\mu^{(i)} - 1\right),$$

respectively, for $\mu^{(i)} = \mu(M_i)$ the spectral condition number of M_i. Therefore, we may conclude that the eigenvalues of K are in between $\sum_{i=1}^k \eta_{\min}^{(i)}$ and $\sum_{i=1}^k \eta_{\max}^{(i)}$. This way, we find for the optimal value of θ and for the optimal spectral radius (as discussed in Section 4.4.1) the estimates

$$\theta = \frac{2}{\sum_{i=1}^k \frac{\mu^{(i)}+1}{\mu^{(i)}-1} \log \mu^{(i)}},$$

$$\rho = \frac{\sum_{i=1}^k \log \mu^{(i)}}{\sum_{i=1}^k \frac{\mu^{(i)}+1}{\mu^{(i)}-1} \log \mu^{(i)}}.$$

It is interesting to point out that in this case the convergence speed is related neither to the condition number of the structured geometric mean $\widehat{\mathbf{G}}$ nor to those of the matrices A_i but is related only to the relative distances of $\widehat{\mathbf{G}}$ from each A_i measured by the quantities $\mu^{(i)} = \mu(M_i)$, $M_i = A_i^{-1/2} \widehat{\mathbf{G}} A_i^{-1/2}$. The closer they are to 1, the faster is the convergence. Therefore, if the matrices A_i are not too far from each other, so that the quantities $\mu(M_i)$ are close to 1, the optimal value of θ is close to $1/k$ and a very fast convergence is expected. This analysis is confirmed by the numerical experiments.

4.4.4 The case of Toeplitz matrices

From the computational point of view, at each step of the iteration (4.9) we have to compute $U^T \operatorname{vec}(\Gamma(\sigma(t)))$ and then to solve a linear system with the matrix

$V(\sigma(t))$. The former computation, based on (4.8), requires $O(kn^3)$ arithmetic operations (ops), while the cost of the latter depends on the structure of $V(\sigma(t))$.

In this section we examine the case where $\mathcal{U} := \mathcal{T}_n^+$, the class of PD Toeplitz matrices, and where $\sigma(t)$ is the symmetric Toeplitz matrix having t as its first column. We describe a way to make the algorithm of Section 4.4.3 more efficient by exploiting the Toeplitz structure.

For the iteration analyzed in Section 4.4.2, V is the diagonal matrix with diagonal entries $(n, 2n-2, ..., 2)$ and the cost of solving a system with matrix V amounts to n divisions.

The iteration examined in Section 4.4.3 has a faster convergence speed but at each step an $n \times n$ system with $V(X) = U^T\big(X^{-1} \otimes X^{-1}\big)U$ must be solved, where X is a PD Toeplitz matrix. We split the computation in two steps. In the first, the n^2 entries of V are computed, in the second step a standard $O(n^3)$ ops linear system solver is used. Concerning the first step we discuss two approaches.

In both approaches the inverse of the Toeplitz matrix X needs to be computed, which can be done efficiently using the Gohberg Semencul formula [26]. Here, vectors v_1, v_2, v_3, and v_4 are determined such that $X^{-1} = L(v_1)L(v_2)^T - L(v_3)L(v_4)^T$, where $L(v)$ is the lower triangular Toeplitz matrix whose first column is v. From these, the n^2 entries of X^{-1} can be found. The overall cost is $O(n^2)$ ops.

1. As a first attempt, the entries of V are computed in a straightforward manner using the entries of X^{-1}:

$$V = \begin{bmatrix} \gamma_{1,1} & 2\gamma_{1,2} & \cdots & 2\gamma_{1,n} \\ 2\gamma_{1,2} & 2\gamma_{2,2} & \cdots & 2\gamma_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 2\gamma_{1,n} & 2\gamma_{2,n} & \cdots & 2\gamma_{n,n} \end{bmatrix},$$

$$\begin{bmatrix} \gamma_{1,j} = \sum_{i=1}^{n} \sum_{k=1}^{n-j+1} (X^{-1})_{i,k}(X^{-1})_{i,k+j-1}, \\ \gamma_{j,p} = \sum_{i=1}^{n-j+1} \sum_{k=1}^{n-p+1} \Big(\big(X^{-1}\big)_{i,k} \big(X^{-1}\big)_{i+j-1,k+p-1} \\ \qquad\qquad + \big(X^{-1}\big)_{i,k+p-1} \big(X^{-1}\big)_{i+j-1,k} \Big). \end{bmatrix}$$

   The cost of this approach in terms of arithmetic operations is of the order $O(n^4)$.

2. In the second approach, we show that the cost of this computation can be kept at the level of $O(n^3 \log n)$ ops by combining the Gohberg Semencul formula and the FFT. For a given i, the product vector $w_i = (X^{-1} \otimes X^{-1})Ue_i$, where e_i is the ith vector of the canonical basis, is such that $w_i = \mathrm{vec}(X^{-1}E_iX^{-1})$, with E_i being the symmetric Toeplitz matrix whose first column is e_i. Therefore, compute first the columns of E_iX^{-1} by performing $O(n^2)$ additions, and then multiply X^{-1} by these columns, stacking the results to obtain w_i. This computation is performed in $O(n^2 \log n)$ operations for each i by using the Gohberg Semencul formula, since the multiplication of a lower triangular Toeplitz matrix and a vector can be performed in $O(n \log n)$ operations by means of the FFT [26]. Therefore the overall computation of this stage for $i = 1, \ldots, n$ is $O(n^3 \log n)$ ops. Finally, compute for any i the vector $U^T w_i$ at a cost of $O(n^2)$ additions.

The performance and accuracy of these methods will be compared in the next section.

4.5 Numerical experiments

In this section, the different algorithms proposed in Sections 4.4.2 and 4.4.3 will be compared w. r. t. speed and accuracy. The numerical experiments are confined to Toeplitz matrices, because of applicational interest in computing their structured matrix mean [12, 14, 15, 81, 130]. These matrices are constructed randomly, but with chosen condition number, using a Newton-type iteration described by Chu and Golub [40] for structured (Toeplitz) inverse eigenvalue problems. Performance, accuracy and computational distance are subjects of the forthcoming investigations. For clarity we remind the reader that the Richardson-iteration corresponds to a projection technique on a manifold, with the classical Euclidean inner product. For all algorithms, the stopping criteria are based on checking the step size determined by the Armijo line search method [1] and on comparing two consecutive iteration points.

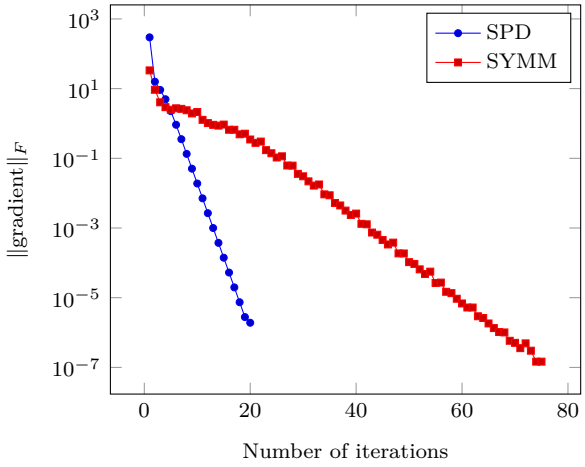
Despite the lack of proof that the structured geometric mean in the Toeplitz case is unique, it is worth pointing out that our experiments did indicate this. For any fixed set of given matrices, any initial guess, and any algorithm, we always obtained the same structured geometric mean. This suggests the conjecture that in the Toeplitz case there is a unique structured geometric mean.

Performance The performance of the projection methods explained in Section 4.4 can be compared by looking at both the number of iterations the methods require and the total amount of computational time they need.

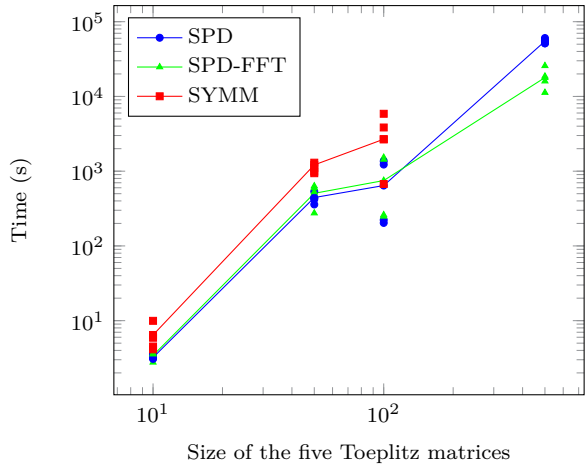
In Figure 4.3(a), the evolution of the gradient over the iterations is displayed for both techniques (and hence also the number of iterations). Using the projection method introduced in Section 4.4.3 gives a faster decrease of the gradient and results in fewer iteration steps. The number of iterations remains almost constant for this method as the size of the matrices increases. On the other hand, for the projection technique from Section 4.4.2, this number starts to increase when the matrix size grows.

However, comparing expression (4.13) and (4.15), it can be seen that the second one is computationally more expensive and hence the advantage of requiring fewer iterations could be nullified. Therefore, Figure 4.3(b) displays the total computational time of both methods for varying sizes of the matrices (both approaches from Section 4.4.4 are shown). The two methods based on Section 4.4.3 maintain an advantage despite their larger computational cost per iteration. Note that for the largest matrix size the computational time of the Euclidean based method is no longer depicted. Because of the increasing number of required iterations for larger matrices, the algorithm reaches the maximum amount of iterations and fails to converge. Concerning the operation count in Section 4.4.4, the advantage of the method based on FFT starts to appear when the matrices become sufficiently large.

Accuracy In order to analyze the accuracy of the projection methods, they are compared with a high precision version of the first algorithm in Section 4.4.4 using the `vpa` functionality of MATLAB. The relative distance, based on the intrinsic distance (4.1), between this high precision computation and the result of the actual algorithms is shown in Figure 4.4. For small condition numbers, the accuracy of all methods is similar in average, but as the condition of the matrices becomes worse, the accuracy of the projection method based on Euclidean geometry deteriorates much faster than that of the method based on the Riemannian geometry. This indicates a lower stability of the Euclidean based method, which even fails to converge when the condition number of the matrices becomes significantly large. The accuracy of the two approaches in Section 4.4.4 is similar and deteriorates steadily as the condition numbers of the matrices increase.



(a) Gradient evolution for five 50×50 matrices.



(b) Computational time for various matrix sizes. The mean of the samples is connected by a line.

Figure 4.3: Comparison of the projection methods for Toeplitz matrices. In the legends, **SYMM** indicates the method of Section 4.4.2, **SPD** indicates the first approach described in Section 4.4.4, and **SPD-FFT** the second.

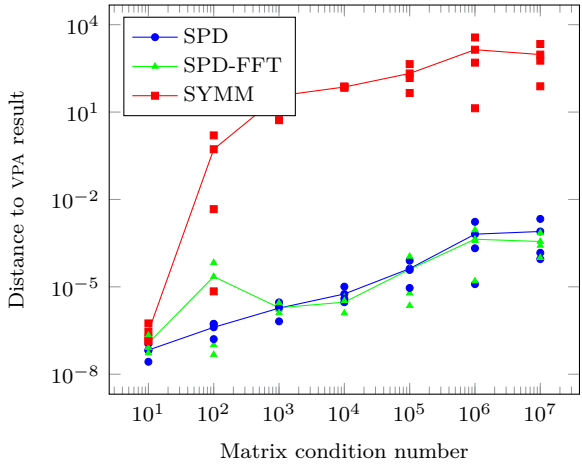


Figure 4.4: Accuracy of the projection methods when compared to a high precision version for three 10×10 matrices. The mean of the samples is connected by a line. In the legends, **SYMM** indicates the method of Section 4.4.2, **SPD** indicates the first approach described in Section 4.4.4, and **SPD-FFT** the second.

4.6 Conclusions

In this chapter we discussed an adaptation of the Karcher mean for PD matrices to structured PD matrices. Besides a theoretical investigation and adaptation of the desired properties of such a mean, algorithms were proposed. In the design of the algorithms, two trajectories were put forward, one relying mostly on linear algebra, and one based on differential geometry. A convergence analysis has been performed showing the superiority of the algorithm based on differential geometry. Numerical experiments compared the accuracy and speed of the various techniques and confirmed the theoretical analysis.

Chapter 5

The Kähler mean

The previous chapter was concerned with an adaptation of the Karcher mean of positive definite matrices which accounts for additional structure present in the matrices. While specific attention was paid to the set of positive definite Toeplitz matrices, the structured mean consistently emphasized the positive definiteness of the matrices through its definition using the intrinsic distance of $\mathcal{P}_n$.

In this chapter, we start again by focusing on the set of positive definite Toeplitz matrices. Using a transformation originating in signal processing theory, an existing geometry and distance measure are given specifically for this set. Both the computation and properties of the resulting barycenter are discussed.

Next, we discuss a generalization of the barycenter for the set of positive definite Toeplitz matrices towards the set of positive definite Toeplitz-Block Block-Toeplitz matrices, along with possible generalizations of the transformation, geometry, and distance measure.

The results in this chapter concerning the computation of the Kähler mean of positive definite Toeplitz matrices (Section 5.2.3 and 5.2.4) were published as part of

> Bini, D., Iannazzo, B., Jeuris, B., and Vandebril, R. Geometric means of structured matrices. *BIT Numerical Mathematics 54*, 1 (2014), pp. 55–83.

The presentation of the supporting theory for Toeplitz matrices and the discussion concerning Toeplitz-Block Block-Toeplitz matrices were submitted as part of

> Jeuris, B., and Vandebril, R. The Kähler mean of Block-Toeplitz matrices with Toeplitz structured blocks. Submitted to *SIAM journal on Matrix Analysis and Applications*.

5.1 Introduction

In radar theory and other signal processing applications [12, 14, 15, 81, 130], autocorrelation matrices are very popular to represent a window of some discrete or continuous signal.

For a signal $x(k)$, the element at position (t_1, t_2) in such an autocorrelation matrix is obtained from an averaging operation $E[x(k+t_1)x(k+t_2)^*] = E[x(k+t)x(k)^*]$, with $t = t_1 - t_2$ referred to as the lag. Note that $E[x(k-t)x(k)^*] = E[x(k)x(k+t)^*] = (E[x(k+t)x(k)^*])^*$. Theoretically, this averaging operation is taken over the entire signal, resulting in an infinite sum (for a discrete signal) or integral (for a continuous signal). In practice, the sum/integral is taken over the finite window of interest, where as many entries in the sum/integral as possible are taken considering the lag and size of the window.

For a finite window, the resulting autocorrelation matrix will be a positive definite (PD) Toeplitz matrix. A popular detection technique in radar theory consists of comparing a certain window in a signal with an average of the signal in the neighboring windows. Translated to the autocorrelation matrices, this means that a PD Toeplitz matrix is compared with an average of its neighboring PD Toeplitz matrices.

The previous chapter already proposed an approach to the averaging of PD Toeplitz matrices using the structured geometric mean. The mean was obtained by emphasizing the PD structure of the matrices in a restricted minimization of the Karcher cost function. An alternative could be to focus on the natural geometry of the Toeplitz matrices. But, as a vector space, the set of Toeplitz matrices is naturally endowed with Euclidean geometry, with the arithmetic mean as its corresponding center of mass.

On the other hand, from the applications mentioned above, a transformation of the autocorrelation matrices based on signal processing operations can be found [11, 13]. The transformed space can be endowed with a natural geometry and the corresponding averaging operation shows appealing results in the applications.

We analyze the associated barycenter and its properties in detail, obtaining a simple expression in the real case and a fast algorithm in the complex case.

When the basic signal $x(k)$ is replaced with a multichannel signal $X(k)$, the corresponding autocorrelation matrix can be constructed as a block matrix. Specifically, we obtain a PD Block-Toeplitz (BT) matrix, which is a PD block matrix with identical blocks along the block diagonals. In some applications, the blocks themselves will also have the Toeplitz structure, resulting in autocorrelation matrices which are PD Toeplitz-Block Block-Toeplitz (TBBT). Similar to the case of PD Toeplitz matrices, an average of these generalized matrices is required in the detection application. While this generalization has been explored for PD BT matrices, the Toeplitz structure of the individual blocks is lost in the existing methods. We introduce an adaptation for PD TBBT matrices based on the natural geometry of the generalized transformed space. Using the generalization of the transformation of the matrices, we derive first-order optimization techniques for the computation of the associated generalized means and analyze their properties.

This chapter is organized in the following way. In Section 5.2, the transformation of PD Toeplitz matrices and its underlying interpretation are discussed. Afterwards, the natural geometry of the resulting transformed space is presented, along with the computation and properties of a barycenter in this setting. The barycenter is referred to as the Kähler mean. Two possible generalizations for the transformation of PD Toeplitz matrices towards PD BT matrices are investigated in Section 5.3. Moreover, we also discuss two different distance measures for the second generalized transformation. The generalized Kähler means for PD BT matrices and PD TBBT matrices are presented in Section 5.4 and 5.5, respectively. If not clear from the context, we will explicitly mention whether the generalized mean is considered w. r. t. PD BT matrices or PD TBBT matrices. Finally, in Section 5.6 we compare the resulting algorithms in numerical experiments.

5.1.1 Definitions and notation

We will define and recall the most important basic constructions and notations used in this chapter.

For convenience, we restate the natural distance measure and inner product on the set of PD matrices $\mathcal{P}_n$,

$$\delta(X, Y) = \left\| \log\left(X^{-1/2} Y X^{-1/2}\right) \right\|_F, \tag{5.1}$$

$$\langle \xi_X, \eta_X \rangle_X = \operatorname{tr}\left(\xi_X X^{-1} \eta_X X^{-1}\right), \tag{5.2}$$

where $X, Y \in \mathcal{P}_n$, $\xi_X, \eta_X \in T_X\mathcal{P}_n \simeq \mathcal{H}_n$.

The vector space of Toeplitz matrices consists of all matrices having identical elements along the diagonals,

$$\mathcal{T}_n = \left\{ \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix} \;\middle|\; t_{-n+1}, \ldots, t_{n-1} \in \mathbb{C} \right\}. \tag{5.3}$$

The intersection of this set of Toeplitz matrices with the Hermitian matrices $\mathcal{H}_n$ is given by the elements in (5.3) for which $t_{-i} = t_i^*, i = 0, \ldots, n-1$. The set of PD Toeplitz matrices will be denoted as $\mathcal{T}_n^+ := \mathcal{T}_n \cap \mathcal{P}_n$.

We denote by $\mathcal{B}_{n,N}$ the vector space of BT matrices, where the indices n and N indicate that the matrices consist of n by n blocks and each block is an $N \times N$ matrix. As for the Toeplitz matrices, the set containing all PD elements in $\mathcal{B}_{n,N}$ will be denoted by $\mathcal{B}_{n,N}^+$. A simplified representation of the elements in $\mathcal{B}_{n,N}^+$ is given in Figure 5.1(a).

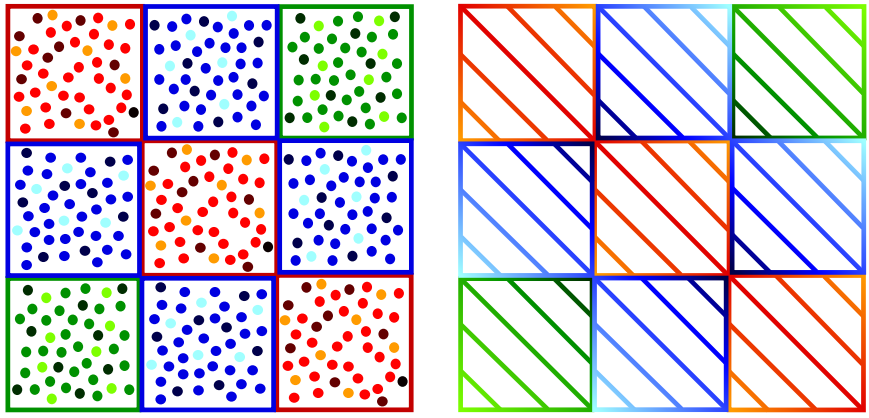
The subspace of $\mathcal{B}_{n,N}$ where the matrix blocks themselves are also Toeplitz matrices is the vector space of TBBT matrices, which we denote by $\mathcal{T}_{n,N}$. The intersection with the manifold of PD matrices is denoted by $\mathcal{T}_{n,N}^+$ and visualized in Figure 5.1(b).

Several instances of (un)structured matrices can be combined in a least squares approach, and the result is in general referred to as the barycenter. For a number of elements $A_1, \ldots, A_k$ in a set $\mathcal{S}$ with given distance measure $d_{\mathcal{S}}$, the barycenter is defined as the minimizer of the sum of squared distances to these given elements,

$$\mathbf{B}_{\mathcal{S}}(A_1, \ldots, A_k) = \arg\min_{X \in \mathcal{S}} \frac{1}{2} \sum_{i=1}^{k} d_{\mathcal{S}}^2(X, A_i). \tag{5.4}$$

This concept is known to be a natural method for combining elements, e. g., the barycenter corresponding to the classical Euclidean geometry is the arithmetic mean. Also note that when $\mathcal{S} := \mathcal{P}_n$ and $d_{\mathcal{S}} := \delta$, the barycenter becomes the Karcher mean[1] discussed in Chapter 3.

[1]Note the difference in scaling of the cost functions by a factor of 1/2. While this scaling does not influence the minimizer of the cost function, we mention that both choices were made for compatibility with the research community. In Chapter 3 and 4, our choice was based on Bhatia [21], while in this chapter, the choice is based on the classical interpretation as center-of-mass [74, 76, 80].



(a) Hermitian or PD Block-Toeplitz matrix.

(b) Hermitian or PD Toeplitz-Block Block-Toeplitz matrix.

Figure 5.1: Simplified representation of the elements of $\mathcal{B}^+_{3,N}$ and $\mathcal{T}^+_{3,N}$.

In what follows, the matrix I_n will represent the $n \times n$ identity matrix, and J_n the so-called counter-identity, the $n \times n$ matrix with ones on the anti-diagonal and zeros everywhere else. For both matrices, the index might be omitted if the size is clear from the context. The transpose of a matrix A will be denoted by A^T, its conjugate transpose by A^H, and its elementwise conjugate by A^*. Finally, we write $\overline{A}$ to represent the form JA^*J. Note that this operation corresponds to taking the conjugate transpose of A and reflecting the result over the anti-diagonal.

5.2 The Kähler mean for Toeplitz matrices

The set of Toeplitz matrices $\mathcal{T}_n$ is a linear space of matrices and is therefore traditionally associated with Euclidean geometry. However, we are interested in the intersection of $\mathcal{T}_n$ with the set of PD matrices $\mathcal{P}_n$. Applying the geometry of the latter to the intersected set resulted in the structured geometric mean, as discussed in Chapter 4. Here, we will discuss a different geometry on $\mathcal{T}_n^+$ [11, 13], along with its underlying interpretation. We introduce a simple expression for the barycenter in case of real matrices and a fast algorithm for complex matrices, after which the properties of the barycenter are studied.

5.2.1 The transformation

The interpretation of the Kähler mean heavily depends on the linear autoregressive model from signal processing theory,

$$x(k) + \sum_{j=1}^{n} a_j^n x(k-j) = w(k),$$

where x is the signal of interest and w represents its prediction error. Our interest now goes to the so-called prediction coefficients a_j^n, and the intermediate factors that arise in their computation.

By applying autocorrelation to the signal $x(k)$, its autocorrelation coefficients $r_t = E\left[x(k+t)x(k)^*\right]$ can be obtained for different lags t. If this autocorrelation is performed on the above autoregressive model, the following system is found:

$$\begin{aligned} R_n \widetilde{a}_n &= -\widetilde{r}_n, \\ \widetilde{a}_n &= [a_1^n, \ldots, a_n^n]^T, \\ \widetilde{r}_n &= [r_1, \ldots, r_n]^T, \end{aligned} \tag{5.5}$$

where R_n is the PD Toeplitz matrix of size n with elements $[R]_{i,j} = [R]_{j,i}^* = r_{i-j}$, $i, j = 0, \pm 1, \ldots, \pm(n-1)$. Note that the prediction error $w(k)$ is assumed to be uncorrelated to the signal $x(k)$. A recursive method known as the Levinson algorithm [73, 127] can be used to find the solution to system (5.5) by solving the system for $n = 1$, and sequentially obtaining the prediction coefficients $\widetilde{a}_n$ for increasing n. The Levinson recurrence relation for the prediction coefficients is given by

$$\begin{aligned} \widetilde{a}_1 &= a_1^1 = -\frac{r_1}{r_0}, \\ a_\ell^\ell &= -\frac{r_\ell + \sum_{j=1}^{\ell-1} r_{\ell-j} a_j^{\ell-1}}{r_0 + \sum_{j=1}^{\ell-1} r_j \left(a_j^{\ell-1}\right)^*}, \\ \widetilde{a}_\ell &= \begin{bmatrix} a_1^\ell \\ \vdots \\ a_{\ell-1}^\ell \\ a_\ell^\ell \end{bmatrix} = \begin{bmatrix} a_1^{\ell-1} \\ \vdots \\ a_{\ell-1}^{\ell-1} \\ 0 \end{bmatrix} + a_\ell^\ell \begin{bmatrix} a_{\ell-1}^{\ell-1*} \\ \vdots \\ a_1^{\ell-1*} \\ 1 \end{bmatrix}, \end{aligned} \tag{5.6}$$

with $\ell = 2, \ldots, n$. It can be shown that the factors a_ℓ^ℓ all lie within the complex unit disk $\mathbb{D}$, $|a_\ell^\ell| < 1, \forall \ell = 1, \ldots, n$.

Our main interest in the above is the one-to-one relation between the PD Toeplitz matrix R_n and the scalars $\left(r_0, a_1^1, \ldots, a_{n-1}^{n-1}\right)$. Note that indices of the prediction coefficients only reach $n-1$, since the computation of a_n^n would require the autocorrelation coefficient r_n, which is only given as an element of the right-hand side of (5.5), but not of R_n.

The *transformation* of the matrix R_n is the following:

$$\begin{array}{ccc} \mathcal{T}_n^+ & \rightarrow & \mathbb{R}^{++} \times \mathbb{D}^{n-1} \\ R_n & \mapsto & (p_0, \mu_1, \ldots, \mu_{n-1}), \end{array} \tag{5.7}$$

where we use the notation $p_0 := r_0$, $\mu_\ell := a_\ell^\ell$, and $\mathbb{R}^{++}$ represents the set of strictly positive real numbers. This transformation creates a one-to-one mapping between the PD Toeplitz matrices and the parameter space $\mathbb{R}^{++} \times \mathbb{D}^{n-1}$. Note that increasing the size of R_n by 1 (increasing n by 1) only requires the computation of 1 additional parameter $\mu_n := a_n^n$, while all other parameters remain fixed. This corresponds to the recursive construction of the Levinson algorithm.

A more concise notation of the transformation (5.7) (which uses (5.6)) of $R_n \in \mathcal{T}_n^+$ is given as follows [131],

$$p_0 := r_0, \quad \mu_\ell := (-1)^\ell \frac{\det(S_\ell)}{\det(R_\ell)}, \quad \ell = 1, \ldots, n-1,$$

with r_0 the main diagonal element of R_n, R_ℓ the principal submatrix of size ℓ of R_n (the upper left $\ell \times \ell$ submatrix) and S_ℓ obtained by shifting R_ℓ down one row, or equivalently, by removing the first row and last column of $R_{\ell+1}$ (the inverse transformation can be found in [131]).

5.2.2 The potential, the metric, and the cost function

In order to define the Kähler metric, the set of PD Toeplitz matrices is considered to be a Kähler manifold [11, 13]. Such a manifold is associated with the concept of a Kähler potential, of which the Hessian form defines the inner product, and hence the geometry, imposed on the manifold. In the field of signal processing (and information geometry in general), the *Kähler potential* is often chosen to be the process entropy $\Phi(R_n)$ [16], defined as follows:

$$\Phi(R_n) = \log\left(\det R_n^{-1}\right) - \log(\pi \mathrm{e}), \tag{5.8}$$

where π and e are the well-known mathematical constants. Applying some decomposition rules on the determinant of R_n and by recognizing the components

of the transformation (5.7) of R_n, the process entropy $\Phi(R_n)$ can be rewritten as a function of the parameter space $\mathbb{R}^{++} \times \mathbb{D}^{n-1}$:

$$\Phi(R_n) = -n \log(p_0) - \sum_{\ell=1}^{n-1} (n-\ell) \log\left(1 - |\mu_\ell|^2\right) - \log(\pi \mathrm{e}),$$

where R_n is identified with its transformation $(p_0, \mu_1, \ldots, \mu_{n-1})$. This decomposition of the determinant of R_n is discussed in more detail for the block matrix case in Section 5.3.1.

The *Kähler metric* can now be obtained by determining the Hessian of the Kähler potential where complex differentiation should be used for the components $\mu_\ell \in \mathbb{D}$. If we denote $\xi^{(n)} = [p_0, \mu_1, \ldots, \mu_{n-1}]^T$, then

$$[H]_{i,j} = \frac{\partial^2 \Phi}{\partial \xi_i^{(n)} \overline{\partial} \xi_j^{(n)}}.$$

The desired metric can be found as

$$\begin{aligned} \mathrm{d}s^2 &= \mathrm{d}\xi^{(n)^H} H \, \mathrm{d}\xi^{(n)} \\ &= n \frac{\mathrm{d}p_0^2}{p_0^2} + \sum_{\ell=1}^{n-1} (n-\ell) \frac{|\,\mathrm{d}\mu_\ell|^2}{\left(1 - |\mu_\ell|^2\right)^2}. \end{aligned} \tag{5.9}$$

By examining this differential metric, a natural geometry and distance measure can be found for (each of the components of) the parameter space $\mathbb{R}^{++} \times \mathbb{D}^{n-1}$. The geometry on $\mathbb{R}^{++}$ is that of the positive numbers, which is given by the scalar analog of (5.1) and (5.2) (up to a scaling with factor $\sqrt{n}$ and n respectively). For the complex unit disk $\mathbb{D}$, the hyperbolic metric of the Poincaré disk can be recognized (up to a scaling of a factor $(n-\ell)/4$). We summarize:

$$\begin{aligned} \forall a, b \in \mathbb{R}^{++}, \forall e, f \in \mathbb{R}: \quad & \langle e, f \rangle_a = n \frac{ef}{a^2}, \\ & d_{\mathbb{R}^{++}}(a, b) = \sqrt{n} \left| \log \frac{b}{a} \right|; \\ \forall \mu, \nu \in \mathbb{D}, \forall \varepsilon, \varsigma \in \mathbb{C}: \quad & \langle \varepsilon, \varsigma \rangle_\mu = \frac{n-\ell}{2} \frac{\varepsilon \varsigma^* + \varsigma \varepsilon^*}{\left(1 - |\mu|^2\right)^2}, \\ & d_{\mathbb{D}}(\mu, \nu) = \frac{\sqrt{n-\ell}}{2} \log \left(\frac{1 + \left| \frac{\mu - \nu}{1 - \mu \nu^*} \right|}{1 - \left| \frac{\mu - \nu}{1 - \mu \nu^*} \right|} \right), \end{aligned} \tag{5.10}$$

where ℓ is chosen corresponding to the coordinate (μ_ℓ, $\ell = 1, \ldots, n-1$, from (5.7)) to which it relates.

Combined, we define the *Kähler distance* $d_{\mathcal{T}_n^+}$ between two PD Toeplitz matrices T_1 and T_2 as

$$\begin{aligned} d^2_{\mathcal{T}_n^+}(T_1, T_2) &= d^2_{\mathcal{T}_n^+}\Big((p_{0,1}, \mu_{1,1}, \ldots, \mu_{n-1,1}), (p_{0,2}, \mu_{1,2}, \ldots, \mu_{n-1,2})\Big) \\ &= n \log^2\left(\frac{p_{0,2}}{p_{0,1}}\right) + \sum_{\ell=1}^{n-1} \frac{n-\ell}{4} \log^2\left(\frac{1 + \left|\frac{\mu_{\ell,1} - \mu_{\ell,2}}{1 - \mu_{\ell,1}\mu^*_{\ell,2}}\right|}{1 - \left|\frac{\mu_{\ell,1} - \mu_{\ell,2}}{1 - \mu_{\ell,1}\mu^*_{\ell,2}}\right|}\right). \end{aligned} \tag{5.11}$$

By entering this distance measure into definition (5.4), the Kähler mean is obtained as the barycenter $\mathbf{B}_{\mathcal{T}_n^+}$. Endowing the manifold $\mathcal{T}_n^+$ with the Kähler metric (5.9) results in a complete, simply connected manifold with non-positive sectional curvature everywhere, or a Cartan–Hadamard manifold. Hence, existence and uniqueness are guaranteed for the barycenter with respect to this metric [37, 76].

5.2.3 The computation

When computing the barycenter $\mathbf{B}_{\mathcal{T}_n^+}$ of the matrices $T_1, \ldots, T_k \in \mathcal{T}_n^+$, we denote the transformation (5.7) of T_i by $(p_{0,i}, \mu_{1,i}, \ldots, \mu_{n-1,i})$, and the coordinates of the barycenter by $(p_{0,\mathbf{B}}, \mu_{1,\mathbf{B}}, \ldots, \mu_{n-1,\mathbf{B}})$. Note that when the distance measure $d_{\mathcal{T}_n^+}$ (5.11) is entered into the definition of a barycenter (5.4), the optimization problem over $\mathbb{R}^{++} \times \mathbb{D}^{n-1}$ can be decoupled into n separate optimization problems, which is convenient for its computation.

The first optimization problem handles the minimization of the function $\varphi_0(x) = \sum_{i=1}^k \log^2(p_{0,i}/x)$ over $\mathbb{R}^{++}$, of which the solution can be found as the scalar geometric mean $p_{0,\mathbf{B}} = (p_{0,1} \cdots p_{0,k})^{1/k}$.

In the other $n-1$ optimization problems we want to minimize the cost function

$$\begin{aligned} \varphi_\ell(z) &= \sum_{i=1}^k \frac{1}{4} \log^2\left(\frac{1 + \left|\frac{z - \mu_{\ell,i}}{1 - z\mu^*_{\ell,i}}\right|}{1 - \left|\frac{z - \mu_{\ell,i}}{1 - z\mu^*_{\ell,i}}\right|}\right) \\ &= \sum_{i=1}^k \operatorname{atanh}^2\left(\left|\frac{z - \mu_{\ell,i}}{1 - z\mu^*_{\ell,i}}\right|\right), \end{aligned}$$

over $\mathbb{D}$, for $\ell = 1, \ldots, n-1$. By definition, the minimizer $\mu_{\ell,\mathbf{B}}$ of $\varphi_\ell(z)$ is the barycenter of $\mu_{\ell,1}, \ldots, \mu_{\ell,k}$ with respect to the customary Poincaré metric on

the complex unit disk. Using (a scaled version of) inner product (5.10) and definition (3.7), we find the minimizer where the Riemannian gradient

$$\operatorname{grad}\varphi_\ell(z) = 2(|z|^2-1)\sum_{i=1}^{k}\operatorname{sign}(c_{\ell,i})\operatorname{atanh}(|c_{\ell,i}|),$$

$$\left[c_{\ell,i} = \frac{\mu_{\ell,i}-z}{1-z^*\mu_{\ell,i}},\right.$$

with $\operatorname{sign}(z) = z/|z|$ the complex sign function, is equal to zero.

In the real case we are able to find an explicit expression for this barycenter as well, since in this case $\operatorname{sign}(c)\operatorname{atanh}(|c|) = \operatorname{atanh}(c)$ and some additional manipulations lead to

$$\mu_{\ell,\mathbf{B}} = \mathcal{C}\left(\left(\mathcal{C}(\mu_{\ell,1})\cdots\mathcal{C}(\mu_{\ell,k})\right)^{1/k}\right),$$

where $\mathcal{C}(z) = (1-z)/(1+z)$ is the Cayley transform.

In the complex case we are not able to find such an explicit formula but a quick numerical method can be devised using a gradient descent algorithm. We recall that the tangent space to the Poincaré disk can be identified with the complex plane and thus for a sufficiently small tangent vector $\varepsilon \in \mathbb{C}$, one can consider the retraction $R_z(\varepsilon) = z+\varepsilon$, which captures the fact that the manifold is an open subset of the complex plane. The resulting algorithm to find the barycenter of any $\mu_1,\ldots,\mu_k \in \mathbb{D}$ is given by the iteration

$$z_{j+1} = z_j + t_j\varepsilon_j, \tag{5.12}$$

$$\left[\begin{array}{l}\varepsilon_j = (1-|z_j|^2)\sum_{i=1}^{k}\operatorname{sign}(c_{j,i})\operatorname{atanh}(|c_{j,i}|),\\ c_{j,i} = \frac{\mu_i - z_j}{1-z_j^*\mu_i},\end{array}\right.$$

for a suitable initial value z_0 and a sufficiently small step length t_j.

Another possibility is to consider the retraction

$$R_z(\varepsilon) = \frac{z+e^{i\theta}+(z-e^{i\theta})e^{-s}}{1+z^*e^{i\theta}+(1-z^*e^{i\theta})e^{-s}},$$

$$\left[\theta = \arg\varepsilon, \quad s = \frac{2|\varepsilon|}{1-|z|^2},\right.$$

which corresponds to moving along the geodesics of the Poincaré disk [131]. Hence, this retraction is naturally associated with the geometry of the disk, but it is slightly more expensive. The corresponding gradient descent method is

$$z_{j+1} = R_{z_j}(t_j\varepsilon_j),$$

with the same ε_j as in (5.12).

5.2.4 The properties

Regarding the properties of the barycenter $\mathbf{B}_{\mathcal{T}_n^+}$, it can easily be seen that it is *permutation invariant*, *repetition invariant*, and *idempotent* (this holds for any barycenter). Moreover, if we denote the transformation (5.7) of a matrix $T_i \in \mathcal{T}_n^+$ again by $(p_{0,i}, \mu_{1,i}, \ldots, \mu_{n-1,i})$, then, for any $\alpha_i > 0$, the transformation of $\alpha_i T_i$ is $(\alpha_i p_{0,i}, \mu_{1,i}, \ldots, \mu_{n-1,i})$. Hence, from the explicit expression of the first coordinate $p_{0,\mathbf{B}}$ of the barycenter $\mathbf{B}_{\mathcal{T}_n^+}$ we get

$$\mathbf{B}_{\mathcal{T}_n^+}(\alpha_1 T_1, \alpha_2 T_2, \ldots, \alpha_k T_k) = (\alpha_1 \cdots \alpha_k)^{1/k}\, \mathbf{B}_{\mathcal{T}_n^+}(T_1, \ldots, T_k),$$

that is, *joint homogeneity* holds.

Unfortunately, this barycenter does not possess other properties of a geometric mean as shown by the following example.

Example 5.1. From the explicit expression for the mean in the real case we get a simple formula for the Kähler barycenter of two 2×2 matrices

$$T_1 = \begin{bmatrix} x_1 & y_1 \\ y_1 & x_1 \end{bmatrix}, \qquad T_2 = \begin{bmatrix} x_2 & y_2 \\ y_2 & x_2 \end{bmatrix},$$

namely

$$\mathbf{B}_{\mathcal{T}_n^+}(T_1, T_2) = \sqrt{x_1 x_2} \begin{bmatrix} 1 & \frac{a-b}{a+b} \\ \frac{a-b}{a+b} & 1 \end{bmatrix}, \text{ with } \begin{cases} a = \sqrt{(x_1 + y_1)(x_2 + y_2)} \\ b = \sqrt{(x_1 - y_1)(x_2 - y_2)} \end{cases}.$$

Now consider the following matrices

$$T_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \qquad \widetilde{T}_1 = \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix}, \qquad T_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix},$$

with $\widetilde{T}_1 \geq T_1$. By symbolic computation, one gets that

$$\mathbf{B}_{\mathcal{T}_n^+}(\widetilde{T}_1, T_2) = \begin{bmatrix} 2\sqrt{2} & \sqrt{2}(\sqrt{5} - 3) \\ \sqrt{2}(\sqrt{5} - 3) & 2\sqrt{2} \end{bmatrix} \not\geq \mathbf{B}_{\mathcal{T}_n^+}(T_1, T_2) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},$$

in fact one eigenvalue of $\mathbf{B}_{\mathcal{T}_n^+}(\widetilde{T}_1, T_2) - \mathbf{B}_{\mathcal{T}_n^+}(T_1, T_2)$ is $\lambda = \sqrt{10} - 2 - \sqrt{2} < 0$. Thus, we have shown that the Kähler barycenter is not *monotonic*. Moreover,

$$\mathbf{B}_{\mathcal{T}_n^+}(T_1, T_2) \neq (T_1 T_2)^{1/2} = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{3} \end{bmatrix},$$

and hence the Kähler barycenter does not coincide with the Karcher mean for commuting matrices. In particular, it is not a structured geometric mean as defined in Chapter 4. The relative Euclidean distance between $\mathbf{B}_{\mathcal{T}_n^+}(T_1, T_2)$ and $(T_1 T_2)^{1/2}$ is 0.134, and the relative Kähler distance (based on (5.11)) is 0.208. In fact, when comparing the structured geometric mean with the Kähler barycenter, we obtain in general a relative difference of the order 10^{-1}, indicating a clear difference between the means.

Observe that in the previous example $\mathbf{B}_{\mathcal{T}_n^+}(T_1, T_2)$ surprisingly coincides with the arithmetic mean of T_1 and T_2. It is not difficult to construct examples where it is not true that $\mathbf{B}_{\mathcal{T}_n^+}(T_1, T_2) \leq (T_1 + T_2)/2$ as it should be for a geometric mean.

Despite lacking some properties of a geometric mean, experimental results have shown that the averaging properties of the Kähler mean cooperate very well with the application from which it was derived [13, 15, 81, 130]. This makes sense since at every step of the derivation, the most natural geometries and concepts, related to this particular model, were chosen from information theory.

Furthermore, the mean also has a computational advantage through its *separation of optimization*. The separate coordinates of the matrices can be grouped and averaged independently:

$$\begin{array}{rcl}
\begin{matrix} T_1 \\ \vdots \\ T_k \end{matrix} & \mapsto & \left(\begin{array}{|c|c|c|c|} \hline p_{0,1}, & \mu_{1,1}, & \cdots, & \mu_{n-1,1} \\ \vdots & \vdots & & \vdots \\ p_{0,k}, & \mu_{1,k}, & \cdots, & \mu_{n-1,k} \\ \hline \end{array} \right) \\
& & \downarrow \quad \downarrow \qquad \downarrow \\
\mathbf{B}_{\mathcal{T}_n^+}(T_1, \ldots, T_k) & \leftarrow & \left(\; p_{0,\mathbf{B}}, \;\; \mu_{1,\mathbf{B}}, \;\; \cdots, \;\; \mu_{n-1,\mathbf{B}} \; \right)
\end{array}$$

This results in two main advantages. First, each coordinate group can be averaged in parallel since they have no influence on any of the other coordinate groups. Second, the means we end up computing contain elements of much smaller sizes than the original data (from matrices of size n to scalars), and additional computational time is saved.

5.3 Generalization of the Toeplitz structure

Our interest goes out to the linear autoregressive model for multichannel signals [91], given by

$$X(k) + \sum_{j=1}^{n} A_j^n X(k-j) = W(k),$$

with X and W vectors of signals and the factors A_j^n square matrices. Taking the normal equations of the multichannel model, the so-called Yule-Walker equations [75] are obtained,

$$\begin{aligned}
\widetilde{A}_n \widetilde{R}_n &= -U_n, \\
\widetilde{A}_n &= \left[A_1^n, \ldots, A_n^n\right], \\
U_n &= \left[R_1, \ldots, R_n\right], \\
\widetilde{R}_n &= \begin{bmatrix} R_0 & R_1 & \cdots & R_{n-1} \\ R_1^H & R_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & R_1 \\ R_{n-1}^H & \cdots & R_1^H & R_0 \end{bmatrix},
\end{aligned} \tag{5.13}$$

where $\widetilde{R}_n \in \mathcal{B}_{n,N}^+$ is a PD BT matrix of n by n blocks. The size of the blocks (N) is equal to the length of the multichannel signal vectors X and W.

Some interesting cases of the multichannel model (such as a 2D signal, when interpreted as a multichannel signal) result in a matrix $\widetilde{R}_n$ which is not only PD BT, but also has the Toeplitz structure in the individual blocks [69, 73, 118]. Hence it will become a PD TBBT matrix. In practice, these Toeplitz blocks will often be Hermitian themselves, $R_\ell = R_\ell^H, \ell = 0, \ldots, n-1$, but we will develop our theory for the more general case in which only the entire matrix $\widetilde{R}_n$ is Hermitian. The results remain valid in the more specified setting.

5.3.1 A first generalized transformation

The transformation With $\widetilde{R}_n$ now defined as a PD TBBT matrix, we would like to generalize the transformation (5.7) to $\mathcal{T}_{n,N}^+$. Similar to the link between the recursion (5.6) and the transformation (5.7), this generalization is obtained using a recursive computation of the prediction matrices in $\widetilde{A}_n$. This recursive

computation goes as follows [73, 91, 126, 128],

$$A_1^1 = -R_1 R_0^{-1}, \tag{5.14}$$

$$A_\ell^\ell = -\Delta_\ell P_{\ell-1}^{-1}, \tag{5.15}$$

$$\left[\begin{array}{l} \Delta_\ell = R_\ell + \sum_{j=1}^{\ell-1} A_j^{\ell-1} R_{\ell-j}, \\ P_{\ell-1} = R_0 + \sum_{j=1}^{\ell-1} J \left(A_j^{\ell-1}\right)^* J R_j = R_0 + \sum_{j=1}^{\ell-1} \overline{A_j^{\ell-1}} R_j, \end{array}\right. \tag{5.16}$$

$$\widetilde{A}_\ell = \left[\widetilde{A}_{\ell-1}, 0\right] + A_\ell^\ell \left[\overline{A_{\ell-1}^{\ell-1}}, \ldots, \overline{A_1^{\ell-1}}, I\right], \tag{5.17}$$

with $\ell = 2, \ldots, n$. Similar to the prediction coefficients a_ℓ^ℓ from before, the factors A_ℓ^ℓ will be the matrices of interest for the generalized transformation. To properly define this transformation, the set in which these matrices lie is investigated.

First of all, note that if all blocks in $\widetilde{R}_n$ (5.13) are assumed to be Toeplitz matrices, we have $\overline{R_\ell} = R_\ell^H, \ell = 0, \ldots, n-1$, and even stronger, $\overline{R_0} = R_0$, since this block is also a PD matrix and hence Hermitian.

Next, we mention the following formula, based on the notion of Schur complement, for the inversion of block matrices,

$$\widetilde{R}_{\ell+1}^{-1} = \begin{bmatrix} \alpha_\ell & -\alpha_\ell U_\ell \widetilde{R}_\ell^{-1} \\ -\widetilde{R}_\ell^{-1} U_\ell^H \alpha_\ell & \widetilde{R}_\ell^{-1} + \widetilde{R}_\ell^{-1} U_\ell^H \alpha_\ell U_\ell \widetilde{R}_\ell^{-1} \end{bmatrix},$$

with $\alpha_\ell = \left(R_0 - U_\ell \widetilde{R}_\ell^{-1} U_\ell^H\right)^{-1}$. Note that α_ℓ is a principal submatrix of the PD matrix $\widetilde{R}_{\ell+1}^{-1}$ and is therefore also PD.

Now, the auxiliary matrix P_ℓ in the recursive computation (5.16) can be written as

$$\overline{P_\ell} = R_0 + \widetilde{A}_\ell U_\ell^H = R_0 - U_\ell \widetilde{R}_\ell^{-1} U_\ell^H = \alpha_\ell^{-1},$$

hence $\overline{P_\ell}$ (and P_ℓ) is also a PD matrix. Using the recursion expression (5.17), an updating rule can be found for $\overline{P_\ell}$ (and consequently for α_ℓ^{-1}),

$$\overline{P_\ell} = \overline{P_{\ell-1}} - \Delta_\ell P_{\ell-1}^{-1} \overline{\Delta_\ell} = \left(I - A_\ell^\ell \overline{A_\ell^\ell}\right) \overline{P_{\ell-1}}, \tag{5.18}$$

where $\overline{P_0} = R_0$.

Finally, we show that the matrices A_ℓ^ℓ belong to the set

$$\mathcal{D}_N = \left\{\Gamma \in \mathbb{C}^{N \times N} \;\middle|\; I - \Gamma\overline{\Gamma} > 0\right\}.$$

Note that for $N = 1$, this set reduces to the complex numbers γ for which $\gamma\overline{\gamma} = \gamma\gamma^* < 1$, which is exactly the complex unit disk $\mathbb{D}$. To prove that all matrix factors A_ℓ^ℓ belong to $\mathcal{D}_N$, we start from the positive definiteness of $\overline{P_\ell}$:

$$\begin{aligned} & \overline{P_\ell} = \overline{P_{\ell-1}} - \Delta_\ell P_{\ell-1}^{-1}\overline{\Delta_\ell} > 0, \\ \xrightarrow{congruence} \quad & I - \overline{P_{\ell-1}^{-1/2}}\Delta_\ell P_{\ell-1}^{-1}\overline{\Delta_\ell}\,\overline{P_{\ell-1}^{-1/2}} > 0, \\ \xrightarrow{similarity} \quad & I - \Delta_\ell P_{\ell-1}^{-1}\overline{\Delta_\ell}\,\overline{P_{\ell-1}^{-1}} = I - A_\ell^\ell\overline{A_\ell^\ell} > 0. \end{aligned}$$

The resulting transformation will be a mapping of the PD TBBT matrices onto the new parameter space, and it is defined as

$$\begin{array}{rcl} \mathcal{T}_{n,N}^+ & \to & \mathcal{P}_N \times \mathcal{D}_N^{n-1} \\ \widetilde{R}_n & \mapsto & (P_0, \Gamma_1, \ldots, \Gamma_{n-1}), \end{array} \tag{5.19}$$

where the notation $P_0 := R_0$, $\Gamma_\ell := A_\ell^\ell$ is used, and N denotes the size of the matrix blocks. Note that the inverse transformation of a random point $(P_0, \Gamma_1, \ldots, \Gamma_{n-1}) \in \mathcal{P}_N \times \mathcal{D}_N^{n-1}$ does not necessarily have the Toeplitz structure in the individual blocks and might not even be a PD matrix. The latter issue will be resolved by the second generalized transformation (Section 5.3.2).

The metric To define the generalized metric, the Kähler potential is examined as in the scalar case. Note the following possible factorization of the determinant of $\widetilde{R}_n$ [104]:

$$\begin{aligned} \det\left(\widetilde{R}_n\right) &= \det\left(\widetilde{R}_{n-1}\right)\det\left(R_0 - U_{n-1}\widetilde{R}_{n-1}^{-1}U_{n-1}^H\right) \\ &= \det\left(\widetilde{R}_{n-1}\right)\det\left(\alpha_{n-1}^{-1}\right) \\ &= \det\left(\widetilde{R}_{n-1}\right)\det\left(I - A_{n-1}^{n-1}\overline{A_{n-1}^{n-1}}\right)\ldots\det\left(I - A_1^1\overline{A_1^1}\right)\det\left(R_0\right) \\ &= \det\left(I - A_{n-1}^{n-1}\overline{A_{n-1}^{n-1}}\right)\ldots\det\left(I - A_1^1\overline{A_1^1}\right)^{n-1}\det\left(R_0\right)^n, \end{aligned} \tag{5.20}$$

where the recursive updating rule (5.18) for α_ℓ^{-1} (and $\overline{P_\ell}$) is used. The resulting factorization of the Kähler potential (5.8) becomes (in parameter space $\mathcal{P}_N \times \mathcal{D}_N^{n-1}$):

$$\Phi\left(\widetilde{R}_n\right) = -n\log\left(\det P_0\right) - \sum_{\ell=1}^{n-1}(n-\ell)\log\left(\det\left(I - \Gamma_\ell\overline{\Gamma_\ell}\right)\right) - \log(\pi e),$$

where $\widetilde{R}_n$ is identified with $(P_0, \Gamma_1, \ldots, \Gamma_{n-1})$ under transformation (5.19).

As before, we use complex differentiation to determine the Hessian of the Kähler potential and obtain the *generalized metric*:

$$\begin{aligned} \mathrm{d}s^2 =& n \operatorname{tr}\left(P_0^{-1}\,\mathrm{d}P_0 P_0^{-1}\,\mathrm{d}P_0\right) \\ &+ \sum_{\ell=1}^{n-1} (n-\ell) \operatorname{tr}\left(\left(I - \Gamma_\ell \overline{\Gamma_\ell}\right)^{-1} \mathrm{d}\Gamma_\ell \left(I - \overline{\Gamma_\ell}\Gamma_\ell\right)^{-1} \overline{\mathrm{d}\Gamma_\ell}\right). \end{aligned}$$

From the metric it can be seen that the desired geometry on $\mathcal{P}_N$ is (up to a scalar $\sqrt{n}$ and n respectively) given by (5.1) and (5.2). Unfortunately, the set $\mathcal{D}_N$ with the geometry described in the above metric does not correspond to any known manifold, nor does a natural distance measure present itself intuitively. However, the set $\mathcal{D}_N$ does bear a close resemblance to the set

$$\mathcal{SD}_N = \left\{\Omega \in \mathbb{C}^{N\times N} \;\middle|\; I - \Omega\Omega^H > 0\right\},$$

which is (almost) the Siegel disk [112] and which has been well-studied along with the Siegel upper halfplane. In the next section we present the slight adaptation to the transformation in order to obtain elements in the parameter space $\mathcal{P}_N \times \mathcal{SD}_N^{n-1}$ and we will also discuss the geometry of the Siegel disk.

5.3.2 A second generalized transformation

In this section, we present a different generalized transformation, where the set $\mathcal{D}_N$ in transformation (5.19) is replaced by the Siegel disk $\mathcal{SD}_N$. Next, we show the relation between both sets and discuss how the new transformation is also a natural extension of the scalar Kähler metric. Finally, the geometry of the Siegel disk will be discussed.

The transformation A different approach to the transformation of a PD (TB)BT matrix can be derived from a link with Verblunsky coefficients [124, 125] as follows.

In the previous setting of Toeplitz matrices, a one-to-one correspondence exists between a PD Toeplitz matrix and a probability measure on the complex unit circle, where the elements in the Toeplitz matrix are found as the moments (or Fourier coefficients) of the corresponding probability measure [35, 44, 47, 67]. The concept of orthogonality for polynomials on the unit circle is linked to the specified probability measure, and thus indirectly to the specific Toeplitz

matrix. Finally, the computation of an orthonormal basis of polynomials on the unit circle can be performed using the Szegő's recursion [117], in which the Verblunsky coefficients arise. It turns out that these coefficients are equal to the prediction coefficients a_ℓ^ℓ (5.6) used in transformation (5.7) [17].

By generalizing the scalar probability measure on the complex unit circle to a nonnegative matrix measure, the collection of its moments into a matrix becomes a PD BT matrix [44, 48]. On the other hand, constructing orthogonal matrix polynomials on the unit circle w. r. t. the matrix measure results in a generalization of the Szegő recursion, with corresponding generalized Verblunsky coefficients [41, 48, 113].

We use the proposed generalization of the Verblunsky coefficients [48] to define a new *transformation* of a PD BT matrix as follows,

$$\begin{array}{rcl} \mathcal{B}_{n,N}^{+} & \rightarrow & \mathcal{P}_N \times \mathcal{SD}_N^{n-1} \\ \widetilde{R}_n & \mapsto & (P_0, \Omega_1, \ldots, \Omega_{n-1}), \end{array} \tag{5.21}$$

where P_0 is still equal to R_0, but now

$$\Omega_\ell := L_{\ell-1}^{-\frac{1}{2}} \left(R_\ell - M_{\ell-1}\right) K_{\ell-1}^{-\frac{1}{2}}, \tag{5.22}$$

$$\left[\begin{array}{l} L_{\ell-1} = R_0 - \left[R_1, \ldots, R_{\ell-1}\right] \widetilde{R}_{\ell-1}^{-1} \left[R_1, \ldots, R_{\ell-1}\right]^H, \\ K_{\ell-1} = R_0 - \left[R_{\ell-1}^H, \ldots, R_1^H\right] \widetilde{R}_{\ell-1}^{-1} \left[R_{\ell-1}^H, \ldots, R_1^H\right]^H, \\ M_{\ell-1} = \left[R_1, \ldots, R_{\ell-1}\right] \widetilde{R}_{\ell-1}^{-1} \left[R_{\ell-1}^H, \ldots, R_1^H\right]^H, \end{array}\right.$$

for $\ell = 1, \ldots, n-1$. Comparing this transformation to the previous one, the following relations can be found in the case of PD TBBT matrices for the auxiliary matrices P_ℓ and Δ_ℓ (5.16): $K_{\ell-1} = P_{\ell-1}$, $L_{\ell-1} = \overline{P_{\ell-1}}$, and $R_\ell - M_{\ell-1} = \Delta_\ell$. Hence we can also write the new transformation as

$$\Omega_\ell = \overline{P_{\ell-1}^{-1/2}} \Delta_\ell P_{\ell-1}^{-1/2},$$

which demonstrates the close relation between both transformations. The absence of the minus sign is not a problem as will become clear from the geometry of the Siegel disk (5.23).

It still remains to show that the coordinate matrices Ω_ℓ actually are elements of the Siegel disk. In fact, this was proven for the transformation of a general PD BT matrix by Dette and Wagener [48] and Fritzsche and Kirstein [59]. We will discuss this for the transformation of elements in the set of PD TBBT matrices $\mathcal{T}_{n,N}^{+}$. Our interest goes specifically to PD TBBT matrices, but we will briefly revisit the PD BT matrices in Section 5.4.

Suppose we have $\widetilde{R}_\ell \in \mathcal{T}^+_{n,N}$, then by exploiting the Toeplitz structure of the blocks and $\overline{\widetilde{R}_\ell} = \widetilde{R}_\ell$, we can show that

$$\begin{aligned}
\overline{\Delta_\ell} &= \overline{R_\ell} - \overline{M_{\ell-1}} \\
&= R_\ell^H - J_N \left[R_1, \ldots, R_{\ell-1}\right]^* \widetilde{R}_{\ell-1}^{-1^*} \left[R_{\ell-1}^H, \ldots, R_1^H\right]^{H^*} J_N \\
&= R_\ell^H - J_N \left[R_1, \ldots, R_{\ell-1}\right]^* J_{nN} \widetilde{R}_{\ell-1}^{-1} J_{nN} \left[R_{\ell-1}^H, \ldots, R_1^H\right]^{H^*} J_N \\
&= R_\ell^H - \left[R_{\ell-1}^H, \ldots, R_1^H\right] \widetilde{R}_{\ell-1}^{-1} \left[R_1, \ldots, R_{\ell-1}\right]^H \\
&= \Delta_\ell^H,
\end{aligned}$$

after which we can again start from the positive definiteness of $\overline{P_\ell}$,

$$\begin{aligned}
& \overline{P_\ell} = \overline{P_{\ell-1}} - \Delta_\ell P_{\ell-1}^{-1} \overline{\Delta_\ell} > 0, \\
\xrightarrow{congruence} \quad & I - \overline{P_{\ell-1}^{-1/2}} \Delta_\ell P_{\ell-1}^{-1} \Delta_\ell^H \overline{P_{\ell-1}^{-1/2}} > 0, \\
& I - \left(\overline{P_{\ell-1}^{-1/2}} \Delta_\ell P_{\ell-1}^{-1/2}\right) \left(P_{\ell-1}^{-1/2} \Delta_\ell^H \overline{P_{\ell-1}^{-1/2}}\right) = I - \Omega_\ell \Omega_\ell^H > 0,
\end{aligned}$$

which proves $\Omega_\ell \in \mathcal{SD}_N$.

The metric We want to define the generalized metric by starting from the Kähler potential, where we continue from (5.20) using the following,

$$\begin{aligned}
\det\left(I - A_\ell^\ell \overline{A_\ell^\ell}\right) &= \det\left(I - \Delta_\ell P_{\ell-1}^{-1} \overline{\Delta_\ell} \overline{P_{\ell-1}^{-1}}\right) \\
&= \det\left(I - \Delta_\ell P_{\ell-1}^{-1} \Delta_\ell^H \overline{P_{\ell-1}^{-1}}\right) \\
&= \det\left(I - \overline{P_{\ell-1}^{-1/2}} \Delta_\ell P_{\ell-1}^{-1} \Delta_\ell^H \overline{P_{\ell-1}^{-1/2}}\right) \\
&= \det\left(I - \Omega_\ell \Omega_\ell^H\right).
\end{aligned}$$

The expression for the Kähler potential and resulting *generalized metric* are

$$\Phi\left(\widetilde{R}_n\right) = -\, n \log\left(\det P_0\right) - \sum_{\ell=1}^{n-1} (n-\ell) \log\left(\det\left(I - \Omega_\ell \Omega_\ell^H\right)\right) - \log(\pi e),$$

$$\begin{aligned}\mathrm{d}s^2 =& n \operatorname{tr}\left(P_0^{-1}\,\mathrm{d}P_0 P_0^{-1}\,\mathrm{d}P_0\right)\\ &+\sum_{\ell=1}^{n-1}(n-\ell)\operatorname{tr}\left(\left(I-\Omega_\ell\Omega_\ell^H\right)^{-1}\mathrm{d}\Omega_\ell\left(I-\Omega_\ell^H\Omega_\ell\right)^{-1}\mathrm{d}\Omega_\ell^H\right).\end{aligned} \tag{5.23}$$

The geometry on $\mathcal{P}_N$ remains the same as for the first transformation. For the Siegel disk $\mathcal{SD}_N$, the natural geometry can be derived from the geometry of the Siegel upper halfplane described by Siegel himself [112], using the link

$$\Omega = (B - iI)(B + iI)^{-1},$$

$$B = i(I + \Omega)(I - \Omega)^{-1},$$

where B is an element of the Siegel upper halfplane ($\mathfrak{Im}(B) > 0$). We should note that this link and the Siegel disk itself are classically only defined for symmetric matrices (in order for the positive definiteness of $\mathfrak{Im}(B)$ to make sense). However, removing the symmetry restriction only disrupts the link and the definition of the Siegel upper halfplane, while the Siegel disk and its geometry remain well-defined.

The resulting (scaled) geometry on $\mathcal{SD}_N$ and a reminder of the (scaled) geometry on $\mathcal{P}_N$ are

$$\forall X, Y \in \mathcal{P}_N, \forall \xi_X, \eta_X \in T_X\mathcal{P}_N \simeq \mathcal{H}_N :$$

$$\begin{aligned}\langle \xi_X, \eta_X\rangle_X =& n \operatorname{tr}\left(X^{-1}\xi_X X^{-1}\eta_X\right),\\ d_{\mathcal{P}_N}(X,Y) =& \sqrt{n}\left\|\log\left(X^{-1/2}YX^{-1/2}\right)\right\|_F;\end{aligned} \tag{5.24}$$

$$\forall \Omega, \Psi \in \mathcal{SD}_N, \forall \upsilon_\Omega, \omega_\Omega \in T_\Omega\mathcal{SD}_N \simeq \mathbb{C}^{N\times N} :$$

$$\begin{aligned}\langle \upsilon_\Omega, \omega_\Omega\rangle_\Omega =& \frac{n-\ell}{2}\operatorname{tr}\left(\left(I-\Omega\Omega^H\right)^{-1}\upsilon_\Omega\left(I-\Omega^H\Omega\right)^{-1}\omega_\Omega^H\right)\\ &+\frac{n-\ell}{2}\operatorname{tr}\left(\left(I-\Omega\Omega^H\right)^{-1}\omega_\Omega\left(I-\Omega^H\Omega\right)^{-1}\upsilon_\Omega^H\right),\end{aligned} \tag{5.25}$$

$$d^2_{\mathcal{SD}_N}(\Omega,\Psi) = \frac{n-\ell}{4}\operatorname{tr}\left(\log^2\left(\frac{I+C^{\frac{1}{2}}}{I-C^{\frac{1}{2}}}\right)\right),$$

$$\left[C = (\Psi-\Omega)\left(I-\Omega^H\Psi\right)^{-1}\left(\Psi^H-\Omega^H\right)\left(I-\Omega\Psi^H\right)^{-1},\right.$$

where ℓ is chosen corresponding to the coordinate matrix (Ω_ℓ, $\ell = 1\ldots, n-1$, from (5.21)) to which it relates. Note that both inner products and distance

measures reduce to the scalar expressions (Section 5.2.2) when $N = 1$. We also point out that the distance measure $d_{\mathcal{SD}_N}$ on the Siegel disk can be written using a Frobenius norm. This is accomplished by performing the similarity transformation $(I - \Omega\Omega^H)^{-1/2} C (I - \Omega\Omega^H)^{1/2}$, which results in a Hermitian matrix (as shown below in (5.28)) and does not change the distance measure since only the eigenvalues of C matter.

The *Kähler distance* d_{BT} between two PD (TB)BT matrices $\widetilde{T}_1$ and $\widetilde{T}_2$, with transformations $(P_{0,1}, \Omega_{1,1}, \ldots, \Omega_{n-1,1})$ and $(P_{0,2}, \Omega_{1,2}, \ldots, \Omega_{n-1,2})$, is defined as

$$d^2_{BT}(\widetilde{T}_1, \widetilde{T}_2) = d^2_{BT}\Big((P_{0,1}, \Omega_{1,1}, \ldots, \Omega_{n-1,1}), (P_{0,2}, \Omega_{1,2}, \ldots, \Omega_{n-1,2})\Big) \tag{5.26}$$

$$= n \left\| \log\left(P_{0,1}^{-1/2} P_{0,2} P_{0,1}^{-1/2}\right) \right\|_F^2 + \sum_{\ell=1}^{n-1} \frac{n-\ell}{4} \operatorname{tr}\left(\log^2\left(\frac{I + C_\ell^{\frac{1}{2}}}{I - C_\ell^{\frac{1}{2}}}\right)\right),$$

$$\left[C_\ell = (\Omega_{\ell,2} - \Omega_{\ell,1}) \left(I - \Omega_{\ell,1}^H \Omega_{\ell,2}\right)^{-1} \left(\Omega_{\ell,2}^H - \Omega_{\ell,1}^H\right) \left(I - \Omega_{\ell,1} \Omega_{\ell,2}^H\right)^{-1}.\right.$$

Using the definition of a barycenter (5.4), the generalized Kähler mean can now be found as $\mathbf{B}_{BT}$.

5.3.3 An alternative for the distance measure on $\mathcal{SD}_N$

The distance measure discussed in the previous section was proposed by Siegel as a possible natural generalization to scalar distance measure on the Poincaré disk. Other generalizations have also been investigated, and among these, the one we will refer to as the *Kobayashi distance measure* d_K has some interesting properties.

For $\Omega, \Psi \in \mathcal{SD}_N$, it is defined as [16, 18, 58]

$$d_K(\Omega, \Psi) = \frac{1}{2} \log\left(\frac{1 + \|\phi_\Omega(\Psi)\|_2}{1 - \|\phi_\Omega(\Psi)\|_2}\right),$$

$$\left[\phi_\Omega(\Psi) = \left(I - \Omega\Omega^H\right)^{-\frac{1}{2}} (\Psi - \Omega) \left(I - \Omega^H \Psi\right)^{-1} \left(I - \Omega^H \Omega\right)^{\frac{1}{2}}, \right. \tag{5.27}$$

which, up to scaling, reduces exactly to the scalar distance measure on the Poincaré disk. The 2-norm $\|\cdot\|_2$ in this expression represents the spectral norm of a matrix, given by its largest singular value.

Unfortunately, the Kobayashi distance measure is not naturally associated to the metric on the Siegel disk with which we are working. We show this by examining

the differential metric at the zero matrix. By entering $\Omega = 0$ in (5.23), our differential metric on the Siegel disk becomes $\mathrm{d}s^2 = \mathrm{tr}(\mathrm{d}\Omega\ \mathrm{d}\Omega^H) = \|\mathrm{d}\Omega\|_F^2$. The differential metric corresponding to the Kobayashi distance measure at the zero matrix is given by $\mathrm{d}s^2 = \|\mathrm{d}\Omega\|_2^2$ [58, Theorem IV.1.8 and Lemma V.1.5], which is clearly not the same. We note that this distance measure could work well when used in combination with the Thompson metric [119] on $\mathcal{P}_N$, mentioned in (4.4), since both are defined using the spectral norm. However, this would lead us even further from the natural geometry, hence we only mention the connection.

The main advantage of the Kobayashi distance measure lies in the transformation ϕ_Ω (5.27), which acts as an automorphism on the Siegel disk. The distance between two matrices and between their transformations under ϕ_Ω remains the same, for both the Siegel distance $d_{\mathcal{SD}_N}$ and the Kobayashi distance d_K, and this can be exploited in the computations. During each step of the optimization process, the current iteration point is translated to the origin (the zero matrix) while the original matrices of the mean are translated accordingly. Working at the origin will simplify the computation of optimization constructions such as the gradient, retractions, etc.

We note already that this translation to the origin is no longer practical once we enforce the Toeplitz structure on the individual blocks R_ℓ, $\ell = 0, \ldots, n-1$, i. e., when we go from PD BT matrices to PD TBBT matrices. As will be fully explained in the next section, once an iteration step ω at the translated origin is computed, the actual iteration point Ω_ℓ (with respect to the original matrices) should be updated to $\phi_{(-\Omega_\ell)}(\omega)$. Imposing the Toeplitz structure on the blocks R_ℓ now results in a very involved condition for the step ω. The process of exploiting the translation itself is further explained in Section 5.4.

5.4 The generalized mean for PD BT matrices

The presence of the underlying Toeplitz structure in the blocks greatly influences the computation of the generalized Kähler mean. Therefore, we first discuss the situation in which the structure is not required, and in the next section, the necessary changes and resulting implications of imposing the Toeplitz condition are presented.

In the general case of PD BT matrices, all advantages of the scalar version are still valid. The optimization of the coordinate matrices under transformation (5.21) can be performed separately, resulting in n parallel optimization processes involving $N \times N$ matrices (instead of a single process involving $nN \times nN$ matrices).

The optimization in the first coordinate matrix results in the Karcher mean $\mathbf{B}_{\mathcal{P}_N}(P_{0,1},\ldots,P_{0,k}) = \mathbf{K}(P_{0,1},\ldots,P_{0,k})$ of the involved PD matrices, which has been discussed in Chapter 3.

For the other coordinates ($\Omega_{\ell,i} \in \mathcal{SD}_N$), the optimization at each level of ℓ ($= 1,\ldots,n-1$) can be formulated in the same way, hence we omit the dependence on ℓ in the definition of the barycenter

$$\mathbf{B}_{\mathcal{SD}_N}(\Omega_1,\ldots,\Omega_k) = \arg\min_{X\in\mathcal{SD}_N} \frac{1}{2}\sum_{i=1}^{k}\left\|\log\left(\frac{I+C_i^{\frac{1}{2}}}{I-C_i^{\frac{1}{2}}}\right)\right\|_F^2,$$

$$\left[C_i = I-(I-\Omega_i\Omega_i^H)^{\frac{1}{2}}(I-X\Omega_i^H)^{-1}(I-XX^H)(I-\Omega_iX^H)^{-1}(I-\Omega_i\Omega_i^H)^{\frac{1}{2}}, \right. \tag{5.28}$$

where the cost function has been rescaled and C_i is written in the Hermitian form which was mentioned in Section 5.3.2. The cost function in this optimization problem will be denoted as $f_{\mathbf{B}_{\mathcal{SD}_N}}(X)$.

A first-order optimization algorithm requires us to determine the (Riemannian) gradient of the cost function, defined in (3.7) and restated for $\mathcal{SD}_N$ as

$$\mathrm{D}f_{\mathbf{B}_{\mathcal{SD}_N}}(X)[\omega_X] = \langle \mathrm{grad}\, f_{\mathbf{B}_{\mathcal{SD}_N}}(X), \omega_X\rangle_X, \tag{5.29}$$

with the inner product (5.25). After some calculations, the emerging gradient is

$$\mathrm{grad} f_{\mathbf{B}_{\mathcal{SD}_N}}(X) = \left(I-XX^H\right)\sum_{i=1}^{k}\left(V_i\left(X-\Omega_i\right)\left(I-X^H\Omega_i\right)^{-1}\right)\left(I-X^HX\right), \tag{5.30}$$

$$\left[\begin{array}{l} V_i = \left(I-\Omega_iX^H\right)^{-1}\left(I-\Omega_i\Omega_i^H\right)^{\frac{1}{2}}Z_i\left(I-\Omega_i\Omega_i^H\right)^{\frac{1}{2}}\left(I-X\Omega_i^H\right)^{-1}, \\ Z_i = \mathfrak{L}\left(C_i^{\frac{1}{2}}, (I-C_i)^{-1}\log\left(\frac{I+C_i^{\frac{1}{2}}}{I-C_i^{\frac{1}{2}}}\right)\right), \end{array}\right.$$

where C_i is defined as in (5.28) and $\mathfrak{L}(A,Q)$ stands for the solution X of the continuous Lyapunov equation $AX+XA^H = Q$. Note that the second argument in the Lyapunov operator $\mathfrak{L}$ is a Hermitian matrix, hence the continuous Lyapunov equation (CLE) is well-defined. This gradient can be used to design a basic steepest descent or conjugate gradient method in order to obtain the barycenter.

Translation to the origin Using the translation ϕ (5.27), computations can be greatly simplified. Suppose the initial guess for the barycenter $\mathbf{B}_{\mathcal{SD}_N}$ is given

by a matrix X_0. The translation ϕ_{X_0} maps the matrix X_0 exactly onto the origin and by applying the same transformation to the original matrices Ω_i, the distances and hence the barycenter cost function do not change. The gradient of the (translated) cost function can now be computed at the origin and used in a basic descent method to obtain a new iteration point, denoted by Ψ_1. We can translate this new point again to the origin using the next translation ϕ_{Ψ_1}. However, in order to keep track of the barycenter approximations with respect to the original matrices, we need to keep in mind that Ψ_1 is an improvement over the origin for the translated matrices $\phi_{X_0}(\Omega_i)$. The new barycenter approximation with respect to the original matrices is hence given by $X_1 = \phi_{-X_0}(\Psi_1)$ (Note that $\phi_{X_0}^{-1} = \phi_{-X_0}$).

The resulting procedure is summarized Algorithm 5.4.1. Note that $\Omega_i^{(j+1)}$ can also be computed as $\phi_{\Psi_{j+1}}(\Omega_i^{(j)})$ [16]. However, in both this formula and the one mentioned in the algorithm, a translation needs to be performed, but by always restarting from the original matrices, the update formula mentioned in the algorithm is less sensitive to the accumulation of round-off errors.

Algorithm 5.4.1 Procedure for translating to the origin

Let $\Omega_1, \ldots, \Omega_k$ be k matrices in $\mathcal{SD}_N$, $X_0 \in \mathcal{SD}_N$ an initial guess

- **for** $j = 0, 1, \ldots$
  - Compute the translated matrices:
$$(\Omega_1^{(j)}, \ldots, \Omega_k^{(j)}) = (\phi_{X_j}(\Omega_1), \ldots, \phi_{X_j}(\Omega_k));$$
  - Compute the gradient of the translated cost function at the origin (5.31):
$$\operatorname{grad} f_{\mathbf{B}_{\mathcal{SD}_N}}(0; \Omega_1^{(j)}, \ldots, \Omega_k^{(j)}),$$
    and perform a basic descent step to obtain Ψ_{j+1};
  - Obtain the next iteration point by returning to the original matrices:
$$X_{j+1} = \phi_{-X_j}(\Psi_{j+1});$$
- **end for**

Return: $\mathbf{B}_{\mathcal{SD}_N}(\Omega_1, \ldots, \Omega_k)$

Finally, we present the simplified form of the gradient at the origin,

$$\text{grad} f_{\mathbf{B}_{\mathcal{SPD}_N}}(0; \Omega_1, \ldots, \Omega_k) = -\sum_{i=1}^{k} V_i \Omega_i, \tag{5.31}$$

$$\left[V_i = \mathfrak{L}\left(\left(\Omega_i \Omega_i^H\right)^{\frac{1}{2}}, \log\left(\frac{I + \left(\Omega_i \Omega_i^H\right)^{\frac{1}{2}}}{I - \left(\Omega_i \Omega_i^H\right)^{\frac{1}{2}}} \right) \right) ,$$

where V_i is now obtained directly as the solution of a CLE.

5.5 The generalized mean for PD TBBT matrices

As mentioned, in some applications the Toeplitz structure is not only present in the block structure, but also in the individual blocks themselves. To investigate the implications of this restriction, we have another look at the transformation (5.21) of the matrices, with the $n-1$ coordinate matrices in the Siegel disk given by (5.22).

At first sight, imposing the Toeplitz structure requires the matrix R_ℓ in each Ω_ℓ to be Toeplitz. However, the matrices $L_{\ell-1}$, $K_{\ell-1}$, and $M_{\ell-1}$ depend on the matrices $R_0, \ldots, R_{\ell-1}$, which should also be Toeplitz matrices now. All these Toeplitz restrictions are translated in an involved way to the search space in which each Ω_ℓ is located. By taking the involved connections into account, we will derive the general Kähler mean for PD TBBT matrices. Afterwards, we present an approximation to this general Kähler mean which again allows us to perform the optimization of the coordinate matrices separately, but now sequentially in the given order of the variables as in transformation (5.21) ($P_0 \to \Omega_1 \to \ldots \to \Omega_{n-1}$).

5.5.1 Global version of the mean

Instead of translating the Toeplitz restriction towards involved conditions on the coordinate matrices $(P_0, \Omega_1, \ldots, \Omega_{n-1})$, we consider the barycenter cost function $f_{\mathbf{B}_{BT}}$, based on the total Kähler distance function d_{BT} (5.26), as a function of the blocks $R_0, \ldots, R_{n-1}$ of the matrix $\widetilde{R}_n$. Doing so will result in a more involved gradient, but it allows us to enforce the Toeplitz structure directly onto its components.

The complexity of this differentiation 'throughout' the coordinate matrices is caused by the dependence on the original blocks. While the first coordinate

matrix P_0 only depends on R_0, each coordinate matrix Ω_ℓ depends on the blocks $R_0, \ldots, R_\ell$, for $\ell = 1, \ldots, n-1$. Or reversely, R_0 will influence all coordinate matrices, and for each $\ell = 1, \ldots, n-1$, block R_ℓ is present in coordinate matrices $\Omega_\ell, \ldots, \Omega_{n-1}$.

The gradient As shown in (5.29), the gradient of the cost function depends on its derivative and the inner product on the search space. Because of the intricate connections between the variables, the gradient is now defined on the product space of the blocks as follows

$$\begin{aligned}
&\mathrm{D}f_{\mathbf{B}_{BT}}\big((R_0,\ldots,R_{n-1})\big)\big[(E_0,\omega_1,\ldots,\omega_{n-1})\big] \\
&\quad= \big\langle \operatorname{grad} f_{\mathbf{B}_{BT}}\big((R_0,\ldots,R_{n-1})\big), (E_0,\omega_1,\ldots,\omega_{n-1})\big\rangle_{(R_0,\ldots,R_{n-1})} \\
&\quad:= \langle \operatorname{grad} f_{\mathbf{B}_{BT}}\big((R_0,\ldots,R_{n-1})\big)_0, E_0\rangle_{P_0} \\
&\qquad+ \sum_{\ell=1}^{n-1} \langle L_{\ell-1}^{-\frac{1}{2}} \operatorname{grad} f_{\mathbf{B}_{BT}}\big((R_0,\ldots,R_{n-1})\big)_\ell K_{\ell-1}^{-\frac{1}{2}}, L_{\ell-1}^{-\frac{1}{2}} \omega_\ell K_{\ell-1}^{-\frac{1}{2}}\rangle_{\Omega_\ell},
\end{aligned} \tag{5.32}$$

where $(P_0, \Omega_1, \ldots, \Omega_{n-1})$ is the image of $\widetilde{R}_n$ under transformation (5.21) with $L_{\ell-1}$ and $K_{\ell-1}$ the matrices formed during the transformation. The inner products $\langle .,. \rangle_{P_0}$ and $\langle .,. \rangle_{\Omega_\ell}$ are given by (5.24) and (5.25), respectively, and $\operatorname{grad} f_{\mathbf{B}_{BT}}\big((R_0,\ldots,R_{n-1})\big)_\ell$ represents the $(\ell+1)$th component of the gradient. The left and right multiplication by $L_{\ell-1}^{-1/2}$ and $K_{\ell-1}^{-1/2}$ in the last inner products is a consequence of the relation between the tangent space at R_ℓ versus the tangent space at Ω_ℓ.

To demonstrate the complexity of the relations, we present the gradient below. The point at which the gradient is computed is denoted by $\widetilde{R}_n$, with blocks $(R_0, \ldots, R_{n-1})$ and transformation $(P_0, \Omega_1, \ldots, \Omega_{n-1})$, while the PD TBBT matrices of which the barycenter is computed will be denoted by $\widetilde{R}_{n,i}$, with blocks $(R_{0,i}, \ldots, R_{n-1,i})$ and transformation $(P_{0,i}, \Omega_{1,i}, \ldots, \Omega_{n-1,i})$, $i = 1, \ldots, k$.

In the expressions, the matrices $A_j^{\ell-1}$ (5.14–5.17), associated with the creation of Δ_ℓ and $P_{\ell-1}$ (and therefore $L_{\ell-1}$, $K_{\ell-1}$, and $M_{\ell-1}$) in the transformation of $\widetilde{R}_n$, are used to increase readability and computational efficiency. The *first*

component of the gradient becomes the following

$$\begin{aligned}&\operatorname{grad} f_{\mathbf{B}_{BT}}\big(\,(R_0,\dots,R_{n-1})\,\big)_0\\ &\qquad = P_0\sum_{i=1}^{k}\left(P_0^{-1}\log\left(P_0P_{0,i}^{-1}\right)+\sum_{\ell=1}^{n-1}\frac{n-\ell}{2n}G_{\ell,i}\right)P_0,\end{aligned}\tag{5.33}$$

$$\left[\begin{array}{l}
G_{\ell,i} = -D^L_{\ell.i} - D^K_{\ell,i} + \sum_{j=1}^{\ell-1}\Big(- {A_j^{\ell-1}}^H D^L_{\ell,i} A_j^{\ell-1} - \overline{A_j^{\ell-1}}^H D^K_{\ell,i}\overline{A_j^{\ell-1}}\\
\qquad\qquad + {A_j^{\ell-1}}^H L_{\ell-1}^{-\frac{1}{2}} {V_{\ell,i}^{(1)}}^H K_{\ell-1}^{-\frac{1}{2}}\overline{A_{\ell-j}^{\ell-1}} + \overline{A_{\ell-j}^{\ell-1}}^H K_{\ell-1}^{-\frac{1}{2}} V_{\ell,i}^{(1)} L_{\ell-1}^{-\frac{1}{2}} A_j^{\ell-1}\Big),\\
D^L_{\ell,i} = \mathfrak{L}\left(L_{\ell-1}^{\frac{1}{2}}, \Omega_\ell V_{\ell,i}^{(1)} L_{\ell-1}^{-\frac{1}{2}} + L_{\ell-1}^{-\frac{1}{2}} {V_{\ell,i}^{(1)}}^H \Omega_\ell^H\right),\\
D^K_{\ell,i} = \mathfrak{L}\left(K_{\ell-1}^{\frac{1}{2}}, \Omega_\ell^H {V_{\ell,i}^{(1)}}^H K_{\ell-1}^{-\frac{1}{2}} + K_{\ell-1}^{-\frac{1}{2}} V_{\ell,i}^{(1)} \Omega_\ell\right),\\
V_{\ell,i}^{(1)} = (I-\Omega^H_{\ell,i}\Omega_\ell)^{-1}(\Omega_\ell^H - \Omega^H_{\ell,i})V_{\ell,i},\\
V_{\ell,i} = (I-\Omega_{\ell,i}\Omega_\ell^H)^{-1}(I-\Omega_{\ell,i}\Omega^H_{\ell,i})^{\frac{1}{2}} Z_{\ell,i}(I-\Omega_{\ell,i}\Omega^H_{\ell,i})^{\frac{1}{2}}(I-\Omega_\ell\Omega^H_{\ell,i})^{-1},\\
Z_{\ell,i} = \mathfrak{L}\left(C_{\ell,i}^{\frac{1}{2}}, (I-C_{\ell,i})^{-1}\log\left(\frac{I+C_{\ell,i}^{\frac{1}{2}}}{I-C_{\ell,i}^{\frac{1}{2}}}\right)\right),\\
C_{\ell,i} = I - \Big((I-\Omega_{\ell,i}\Omega^H_{\ell,i})^{\frac{1}{2}}(I-\Omega_\ell\Omega^H_{\ell,i})^{-1}(I-\Omega_\ell\Omega_\ell^H)\dots\\
\qquad\qquad (I-\Omega_{\ell,i}\Omega_\ell^H)^{-1}(I-\Omega_{\ell,i}\Omega^H_{\ell,i})^{\frac{1}{2}}\Big),
\end{array}\right.$$

where $D^L_{\ell,i}$, $D^K_{\ell,i}$, and $Z_{\ell,i}$ are obtained by solving a CLE. The *other components of the gradient* are, for $q=1,\dots,n-1$, given by

$$\begin{aligned}\operatorname{grad} f_{\mathbf{B}_{BT}}\big(\,(R_0,\dots,R_{n-1})\,\big)_q &= L_{q-1}^{\frac{1}{2}}\left(I-\Omega_q\Omega_q^H\right)\sum_{i=1}^{k}{V_{q,i}^{(1)}}^H\left(I-\Omega_q^H\Omega_q\right)K_{q-1}^{\frac{1}{2}}\\ &+ L_{q-1}^{\frac{1}{2}}\left(I-\Omega_q\Omega_q^H\right)L_{q-1}^{\frac{1}{2}}\sum_{i=1}^{k}\left(\sum_{\ell=q+1}^{n-1}\frac{n-\ell}{n-q}W_{\ell,i}^{(q)}\right)K_{q-1}^{\frac{1}{2}}\left(I-\Omega_q^H\Omega_q\right)K_{q-1}^{\frac{1}{2}},\end{aligned}\tag{5.34}$$

$$\left[\begin{array}{l}
W_{\ell,i}^{(q)} = -D^L_{\ell,i}A_q^{\ell-1} - \overline{A_q^{\ell-1}}^H D^K_{\ell,i}\\
\quad + {A_{\ell-q}^{\ell-1}}^H L_{\ell-1}^{-\frac{1}{2}}{V_{\ell,i}^{(1)}}^H K_{\ell-1}^{-\frac{1}{2}} + L_{\ell-1}^{-\frac{1}{2}}{V_{\ell,i}^{(1)}}^H K_{\ell-1}^{-\frac{1}{2}}\overline{A_{\ell-q}^{\ell-1}}\\
\quad + \sum_{j=q+1}^{\ell-1}\Big(- {A_{j-q}^{\ell-1}}^H D^L_{\ell,i}A_j^{\ell-1} - \overline{A_j^{\ell-1}}^H D^K_{\ell,i}\overline{A_{j-q}^{\ell-1}}\\
\qquad + {A_{j-q}^{\ell-1}}^H L_{\ell-1}^{-\frac{1}{2}}{V_{\ell,i}^{(1)}}^H K_{\ell-1}^{-\frac{1}{2}}\overline{A_{\ell-j}^{\ell-1}} + \overline{A_{\ell-j+q}^{\ell-1}}^H K_{\ell-1}^{-\frac{1}{2}}V_{\ell,i}^{(1)}L_{\ell-1}^{-\frac{1}{2}}A_j^{\ell-1}\Big),
\end{array}\right.$$

where $D^L_{\ell,i}$, $D^K_{\ell,i}$, and $V^{(1)}_{\ell,i}$ are the same as for the first component.

What we have done so far is to compute the gradient of $f_{\mathbf{B}_{BT}}$ as a function of the matrix blocks $(R_0, \ldots, R_{n-1})$ instead of the coordinate matrices $(P_0, \Omega_1, \ldots, \Omega_{n-1})$. Finally, we can impose the Toeplitz structure on the blocks.

Projection onto the Toeplitz structure According to manifold optimization theory, computing the gradient of a cost function on some submanifold is equivalent to computing the gradient in the embedding manifold and applying the orthogonal projection onto the submanifold [1]. In our case, the embedding manifold is the set $\left(\mathbb{C}^{N\times N}\right)^n_+$ containing all tuples $(R_0, \ldots, R_{n-1})$ which represent the blocks of an element in $\mathcal{B}^+_{n,N}$. The submanifold is given by the set $(\mathcal{T}_N)^n_+$ which contains all tuples $(R_0, \ldots, R_{n-1})$ holding the blocks of an element in $\mathcal{T}^+_{n,N}$.

Above, we have computed the gradient of the cost function $f_{\mathbf{B}_{BT}}$ for the embedding manifold $\left(\mathbb{C}^{N\times N}\right)^n_+$ since no additional structure was imposed on the blocks. Hence, we need an orthogonal projection of this gradient at any point $(R_0, \ldots, R_{n-1}) \in (\mathcal{T}_N)^n_+ \subset \left(\mathbb{C}^{N\times N}\right)^n_+$ from $T_{(R_0,\ldots,R_{n-1})}\left(\mathbb{C}^{N\times N}\right)^n_+$ onto $T_{(R_0,\ldots,R_{n-1})}\left(\mathcal{T}_N\right)^n_+$. This *projection* should be orthogonal with respect to the inner product (5.32) and, for $(E_0, \omega_1, \ldots, \omega_{n-1}) \in T_{(R_0,\ldots,R_{n-1})}\left(\mathbb{C}^{N\times N}\right)^n_+$, is given by

$$E_0 \mapsto \operatorname{vec}^{-1}\left(U_{\mathrm{H}}\left(U_{\mathrm{H}}^H\left(P_0^{-1^T}\otimes P_0^{-1}\right)U_{\mathrm{H}}\right)^{-1}U_{\mathrm{H}}^H \operatorname{vec}\left(P_0^{-1}E_0P_0^{-1}\right)\right), \quad (5.35)$$

$$\omega_\ell \mapsto \operatorname{vec}^{-1}\left(U_{\mathrm{T}}\left(U_{\mathrm{T}}^H\left(S_\ell^{K^T}\otimes S_\ell^L\right)U_{\mathrm{T}}\right)^{-1}U_{\mathrm{T}}^H \operatorname{vec}\left(S_\ell^L\omega_\ell S_\ell^K\right)\right), \quad (5.36)$$

$$\begin{cases} S_\ell^L = L_{\ell-1}^{-\frac{1}{2}}\left(I-\Omega_\ell\Omega_\ell^H\right)^{-1}L_{\ell-1}^{-\frac{1}{2}}, \\ S_\ell^K = K_{\ell-1}^{-\frac{1}{2}}\left(I-\Omega_\ell^H\Omega_\ell\right)^{-1}K_{\ell-1}^{-\frac{1}{2}}, \end{cases}$$

for $\ell = 1, \ldots, n-1$, where $(P_0, \Omega_1, \ldots, \Omega_{n-1})$ is the transformation (5.21) of $\widetilde{R}_n$, the BT matrix containing blocks $(R_0, \ldots, R_{n-1})$, with associated matrices $L_{\ell-1}$ and $K_{\ell-1}$. The vec-operator is the columnwise vectorization of a matrix, and the matrices U_{H} and U_{T} are parametrization matrices for Hermitian Toeplitz and general Toeplitz matrices, respectively. Hence, e. g., we write $\operatorname{vec}(T_1) = U_{\mathrm{H}}t_1$, with $t_1 \in \mathbb{R}^{2N-1}$ the parametrization of $T_1 \in \mathcal{T}_N \cap \mathcal{H}_N$, and $\operatorname{vec}(T_2) = U_{\mathrm{T}}t_2$, with $t_2 \in \mathbb{C}^{2N-1}$ or $t_2 \in \mathbb{R}^{4N-2}$ a parametrization of $T_2 \in \mathcal{T}_N$.

Note that when the projection is combined with the gradient above, some cancellations occur within the vec operator of the projection. This is a

consequence of the consistent use of inner product (5.32) for both the Riemannian gradient and the orthogonal projection.

5.5.2 Greedy version of the mean

It is obvious that even a basic construction such as the gradient is expensive for the generalized Kähler mean with Toeplitz structure imposed on the blocks. Here we discuss an approximation to this mean which is obtained as an attempt to regain the separated optimization of the coordinate matrices.

Remember from the previous section that the coordinate matrix P_0 only depends on the block R_0, coordinate matrix Ω_1 depends on the blocks R_0 and R_1, etc. The main idea of our approximation is to perform the optimization of the barycenter cost function $f_{\mathbf{B}_{BT}}$ in a greedy manner.

We start by minimizing the part of the cost function which only depends directly on P_0, while imposing the Toeplitz structure on R_0. This results in the computation of the structured Karcher mean of the given coordinate matrices $(P_{0,1}, \ldots, P_{0,k})$ as described in Chapter 4.

When this optimization process is completed, we assume R_0 (and P_0) to be fixed. Next, we continue with the optimization of $\Omega_1 = L_0^{-1/2}\,(R_1 - M_0)\,K_0^{-1/2}$, with the Toeplitz structure imposed on R_1. Note that since R_0 is assumed to be fixed, L_0, K_0, and M_0 are fixed as well, making the relation between Ω_1 and R_1 straightforward. When the optimization process on R_1 is finished, assume both R_0 and R_1 to be fixed and continue this method sequentially.

The optimization at the level of Ω_ℓ, $\ell = 1, \ldots, n-1$ is performed using a combination of constructions which have already been derived. From Section 5.4, we remember the barycenter cost function $f_{\mathbf{B}_{S\mathcal{D}_N}}$ with associated gradient (5.30). Because of the Toeplitz restriction and the assumption that all previously optimized coordinate matrices are fixed, the tangent space at Ω_ℓ is given by

$$T_{\Omega_\ell}\left(L_{\ell-1}^{-1/2}\,(\mathcal{T}_N - M_{\ell-1})\,K_{\ell-1}^{-1/2}\right) \simeq \left\{ L_{\ell-1}^{-\frac{1}{2}} T K_{\ell-1}^{-\frac{1}{2}} \;\middle|\; T \in \mathcal{T}_N \right\}.$$

We are now working directly on the level of Ω_ℓ instead of R_ℓ, hence the projection of the gradient onto this tangent space slightly differs from the one presented in (5.36) as follows

$$\omega_\ell \mapsto \operatorname{vec}^{-1}\left(U_{\mathrm{T}}\left(U_{\mathrm{T}}^H\left(S_\ell^{K^T} \otimes S_\ell^L\right)U_{\mathrm{T}}\right)^{-1} U_{\mathrm{T}}^H \operatorname{vec}\left(S_\ell^L L_{\ell-1}^{\frac{1}{2}} \omega_\ell K_{\ell-1}^{\frac{1}{2}} S_\ell^K\right)\right),$$

where U_{T}, S_ℓ^K, and S_ℓ^L are the same as in (5.36).

This greedy Kähler mean is only an approximation to the generalized Kähler mean since by assuming the previous blocks to be fixed, the search space during the optimization of the current block is more restricted than in the general case. The approximation does allow us to partially return to the situation of separated optimization, since the optimization is performed separately on the blocks, even though they have to be computed sequentially.

5.5.3 Properties of the generalized Kähler mean

When considering the properties of the generalized Kähler mean, an intuitive approach is to start from the properties of the Kähler mean for Toeplitz matrices (Section 5.2.4).

The generalized Kähler mean of PD BT matrices and both the global and greedy version of the Kähler mean of PD TBBT matrices will be *permutation invariant*, *repetition invariant*, and *idempotent*, since all of them are defined as barycenters.

As for the property of joint homogeneity, we start by discussing the *change of transformation* (5.21) when a PD (TB)BT matrix $\widetilde{R}_n$ is replaced with $\alpha\widetilde{R}_n$, for any real $\alpha > 0$. We denote the transformation of $\widetilde{R}_n$ by $(P_0, \Omega_1, \ldots, \Omega_{n-1})$, with corresponding prediction matrices A_j^ℓ and auxiliary matrices $P_{\ell-1}$ and Δ_ℓ, and that of $\alpha\widetilde{R}_n$ by $(P_0', \Omega_1', \Omega_{n-1}')$, now with corresponding prediction matrices $A_j^{\ell\prime}$ and auxiliary matrices $P_{\ell-1}'$ and Δ_ℓ'.

First, the change of the prediction matrices $A_j^{\ell\prime}$, and auxiliary matrices $P_{\ell-1}'$ and Δ_ℓ', $\ell = 1, \ldots, n-1$, $j = 1, \ldots, \ell$, can be found using induction. Considering (5.14)–(5.17), it is clear to see that $A_1^{1\prime} = A_1^1$, $P_0' = \alpha P_0$, and $\Delta_1' = \alpha\Delta_1$. Now assuming $\widetilde{A}_{\ell-1}' = \widetilde{A}_{\ell-1}$, we find $P_{\ell-1}' = \alpha P_{\ell-1}$, $\Delta_\ell' = \alpha\Delta_\ell$, and $A_\ell^{\ell\prime} = A_\ell^\ell$. As a consequence of (5.17), $\widetilde{A}_\ell' = \widetilde{A}_\ell$, which closes the induction.

By writing the coordinate matrices Ω_ℓ' in the form $\overline{P_{\ell-1}'^{-1/2}}\Delta_\ell' P_{\ell-1}'^{-1/2}$, we now find that $\Omega_\ell' = \Omega_\ell$, $\ell = 1, \ldots, n-1$. Summarized, the transformation of $\alpha\widetilde{R}_n$ is given by $(\alpha P_0, \Omega_1, \ldots, \Omega_{n-1})$, which is consistent with the Kähler transformation of PD Toeplitz matrices. Note that transformation (5.19) behaves in the same way for positive scaling.

Now, as for *joint homogeneity*, suppose we have k PD (TB)BT matrices $\widetilde{T}_i$, $i = 1, \ldots, k$, with a corresponding transformation $(P_{0,i}, \Omega_{1,i}, \ldots, \Omega_{n-1,i})$, and k positive scalars α_i. The generalized Kähler mean for PD BT matrices (Section 5.4) is computed separately on the coordinate matrices. Combining this with

the joint homogeneity of the Karcher mean (Chapter 3) is sufficient to prove the property in this case.

The global version of the Kähler mean for PD TBBT matrices (Section 5.5.1) can be seen to satisfy the property by studying the gradient of the cost function. If this gradient becomes the zero matrix for some matrix $\widetilde{R}_n$ with given matrices $\widetilde{T}_i$, $i = 1, \ldots, k$, it can be checked that the same happens for $(\alpha_1 \cdots \alpha_k)^{1/k}\widetilde{R}_n$ with given matrices $\alpha_i\widetilde{T}_i$, $i = 1, \ldots, k$.

Moreover, the greedy approximation (Section 5.5.2) also satisfies the property, which can be seen as follows. We will denote the transformation of the greedy Kähler mean of the unscaled $\widetilde{T}_1, \ldots, \widetilde{T}_k$ by $(P_0, \Omega_1, \ldots, \Omega_{n-1})$. The greedy Kähler mean of the scaled matrices $\alpha_1\widetilde{T}_1, \ldots, \alpha_k\widetilde{T}_k$ now starts by averaging the first coordinate matrices, resulting in $\mathbf{B}_{\mathcal{T}_N^+}(\alpha_1 P_{0,1}, \ldots, \alpha_k P_{0,1}) = (\alpha_1 \cdots \alpha_k)^{1/k} P_0$ because of the joint homogeneity of the structured geometric mean (Chapter 4) for linear structures. As mentioned before, the search space for the coordinates of this greedy mean is dependent on the ones that have already been computed. Hence, for the next coefficients $(\Omega_{1,1}, \ldots, \Omega_{1,k})$ we still minimize the cost function $f_{\mathbf{B}_{\mathcal{SPD}_N}}(X; \Omega_{1,1}, \ldots, \Omega_{1,k})$. However, the search space has changed from

$$\overline{P_0^{-1/2}}\mathcal{T}_N P_0^{-1/2} \cap \mathcal{SPD}_N \qquad \text{to} \qquad (\alpha_1 \cdots \alpha_k)^{-1/k}\overline{P_0^{-1/2}}\mathcal{T}_N P_0^{-1/2} \cap \mathcal{SPD}_N,$$

from which it can be seen that the resulting coordinate matrix Ω_1 remains the same as in the unscaled setting (since a scaling of vector space $\mathcal{T}_N$ does not change the space). The other coordinate matrices Ω_ℓ, $\ell = 2, \ldots, n-1$, similarly do not change. Finally, this results in coordinate matrices $((\alpha_1 \cdots \alpha_k)^{1/k} P_0, \Omega_1, \ldots, \Omega_{n-1})$ for the greedy Kähler mean of the scaled matrices, corresponding to the correct matrix for joint homogeneity to hold.

As for the Kähler mean of PD Toeplitz matrices, it is not difficult to find examples which contradict the property of *monotonicity* and the *Arithmetic-Geometric-Harmonic inequality* (Section 2.2). In fact, the counterexamples found for the Kähler mean of PD Toeplitz matrices can again be used to contradict the properties, since this mean arises as a special example of the generalized Kähler mean for blocks of size 1.

5.6 Numerical experiments

In this section, we will analyze the various algorithms that were discussed for the generalized Kähler mean.

First of all, we will have a closer look at the Siegel disk and compare the barycenters that arise when using the Siegel distance measure $d_{\mathcal{SD}_N}$ and the Kobayashi distance measure d_K.

Afterwards, a comparison of the global and greedy version of the generalized Kähler mean for PD TBBT matrices is presented, where we also combine the methods by using the greedy version as an initial guess for the global mean algorithm.

5.6.1 The Siegel and Kobayashi barycenter on $\mathcal{SD}_N$

In this chapter, we have endowed the Siegel disk $\mathcal{SD}_N$ with the Siegel distance measure $d_{\mathcal{SD}_N}$ (Section 5.3.2) and with the Kobayashi distance measure d_K (Section 5.3.3). Since each distance measure can be used to define a barycenter ($\mathbf{B}_{\mathcal{SD}_N}$ and $\mathbf{B}_K$ respectively) on the Siegel disk, we compare the computational time and results of both.

When investigating the distance between the barycenters, a relative distance of the order $\mathcal{O}(10^{-1})$ can be found consistently for varying matrix sizes. Note that the diameter of the Siegel disk becomes infinity for both distance measures.

As for computational time, we display some results of both barycenters for varying sizes of matrices in Figure 5.2. The Siegel barycenter $\mathbf{B}_{\mathcal{SD}_N}$ requires less computational time, which also increases more slowly.

Perhaps even more interesting is the fact that when we further increase the size of the matrices, the steepest descent method to compute the Kobayashi barycenter starts exhibiting convergence problems and a lack of a unique minimizer. These problems can be ascribed to the presence of the spectral norm in the Kobayashi distance measure. This norm is given by the largest singular value of a matrix, and its derivative (Section 1.2.2) is only well-defined when this largest value is strictly greater than the other singular values. During the computation of the barycenter $\mathbf{B}_K$, it is possible that a matrix with almost equal largest singular values is entered into this derivative, causing convergence problems. This problem is further discussed in Section 6.2.1 of the future research. Furthermore, the derivative of the spectral norm can only contribute a rank one matrix to the gradient of the barycenter cost function for each given matrix in the barycenter. Consequently, this will start causing problems when the number of matrices in the barycenter becomes too small compared to the size of the matrices.

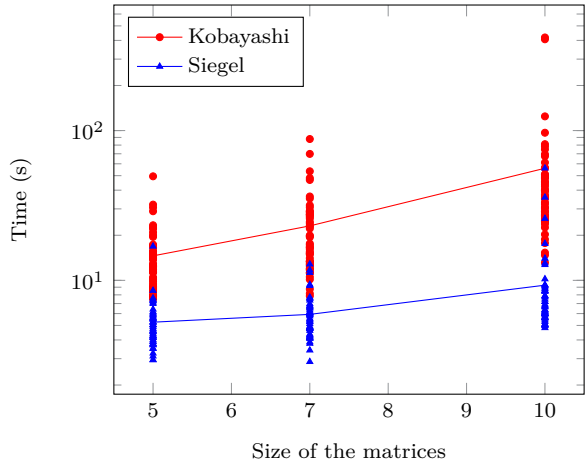


Figure 5.2: Required time for the computation of the Kobayashi and Siegel barycenters $\mathbf{B}_K$ and $\mathbf{B}_{\mathcal{SD}_N}$ of 50 matrices of varying sizes.

5.6.2 The generalized Kähler mean

We have suggested a steepest descent algorithm for the generalized Kähler mean of PD TBBT matrices, followed by a greedy approximation. Here we analyze how close this approximation is to the actual mean and we investigate the computational advantage of the approximation.

First of all, in terms of computational time the greedy version has a clear advantage over the global mean, as illustrated in Figure 5.3. This was expected, since the gradient for the greedy optimization problem can be found in the gradient of the global optimization problem (5.33)–(5.34) by setting the factors $G_{\ell,i}$ (for the first component) and $W_{\ell,i}^{(q)}$ (for the other components) to zero.

In fact, while the basic operations for the terms in the individual blocks of the gradient depend on the size of the matrices (N), the number of terms in each block in the global gradient is dependent on the block size (n) of the matrix. For the gradient in the greedy algorithm, changing the block size of the matrices from n to $n+1$ corresponds to computing one additional block in the gradient, independent of all previous blocks. On the other hand, the gradient in the global algorithm will gain an additional term in each of the previous blocks of the gradient. Hence, the greedy algorithm is linearly dependent on the number of blocks n in the matrices, while for the global algorithm this dependence is quadratic.

Table 5.1: Some averaged comparative values concerning the global and greedy version of the generalized Kähler mean of 20 PD TBBT matrices. The global algorithm is initiated by a random matrix (R), one of the original matrices in the mean (O), or the greedy approximation (G).

Number of blocks n (n by n blocks)	10	20	50
Iterations for Global (R)	24	24	23
Iterations for Global (O)	25	23	23
Iterations for Global (G)	13	12	13
Relative distance Greedy vs. Global	2.28e-04	1.36e-04	8.24e-05
Size global gradient at Greedy	2.34	2.44	3.14

Moreover, in Table 5.1, the (averaged) relative distance between the global version of the generalized Kähler mean and its greedy approximation is shown for a number of block sizes. The observed relative proximity between both versions and the computational advantage of the greedy algorithm suggests that it could work well as an approximation. In fact, many applications require only a limited amount of significant digits, in which case the greedy approximation can replace the actual mean.

The greedy approximation as initial guess for the global algorithm Next, for those applications where the global version of the generalized Kähler mean is required, we analyze the influence of the initial guess on the algorithm. Specifically, the appropriateness of the greedy version as an initial guess is investigated.

In Figure 5.3, the computational time of the global version of the mean is displayed when using a random initial guess and the greedy mean. As can be seen, using the greedy approximation results in a faster algorithm. Note that the time to compute the greedy mean was included in these results. Table 5.1 also displays the advantage of the greedy initial guess, as the required number iterations of the global algorithm are reduced by half. Hence, we can conclude that the greedy approximation works well as an initializer to the global algorithm.

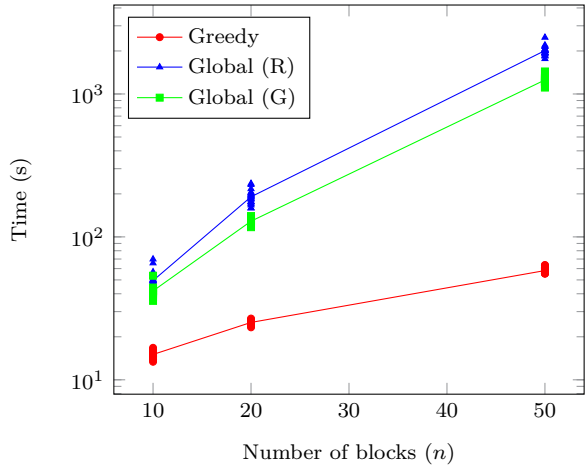


Figure 5.3: Required time for the greedy and global versions of the generalized Kähler mean for 20 PD TBBT matrices as the number of blocks varies (n by n blocks). The global algorithm is initiated by a random matrix (R), or the greedy approximation (G). For initialization with the greedy mean, the combined computational time of the greedy and global mean is shown.

5.7 Conclusions

In this chapter, we have focused on a geometry for positive definite Toeplitz matrices and a generalization thereof towards positive definite (Toeplitz-Block) Block-Toeplitz matrices.

In the case of Toeplitz matrices, the Kähler mean and its properties have been investigated, providing an explicit expression in the real case and a fast algorithm in the complex case. While this mean did not satisfy many properties relating to the ordering of matrices, such as monotonicity and the arithmetic-geometric inequality, it does cooperate well with the application from which it was derived [13, 15, 81, 130].

Afterwards, two possible generalizations of the Kähler transformation towards positive definite (Toeplitz-Block) Block-Toeplitz matrices were presented, of which the second was discussed in further detail. Two possible geometries on the Siegel disk were investigated, where one corresponded naturally with the manifold and the other was based on a useful automorphism of the set. For Toeplitz-Block Block-Toeplitz matrices, a global mean and a greedy approximation were derived, which were compared in numerical experiments. The greedy version of the generalized mean was a close approximation of

the global mean, with a significantly lower computational cost. The greedy approximation was also shown to work well as an initializer for the global optimization algorithm, effectively reducing the number of iterations by half.

Chapter 6

Conclusions

In this chapter, we present the main results of this thesis, accompanied by our main contributions. Afterwards, some possible future extensions of the research topics are discussed.

6.1 Conclusions and contributions

In this thesis, a large number of methods to *average positive definite matrices*, both with and without additional structure, were discussed.

On the set of positive numbers or positive definite matrices, a natural averaging operation is given by the *geometric mean*, which was our first topic of interest. Considering the vast number of definitions for the geometric mean and its related means, we introduced an extensive overview of the various approaches.

Among the several instances of the geometric mean, the *Karcher mean* was presented as the most natural one. Its definition as the minimizer of an optimization problem on the smooth manifold of positive definite matrices led us to an exposition of the rich theory of Riemannian optimization. In this context, various first- and second-order optimization techniques were investigated and compared. It became apparent that while the second-order techniques required less iterations, the computational cost associated with each of these iterations was higher than that of the first-order algorithms, nullifying the advantage of their quadratic convergence as the size of the matrices increased.

Besides the numerous variations of the geometric mean, we also discussed and constructed some *approximations*, which usually lacked one or more of the desired properties. These approximations were based on the explicit expression for the mean of two matrices, and on a combination of harmonic and arithmetic mean iterations. The latter approach resulted in the HA mean, which delivered promising results when combined with a randomization of the matrix order in each iteration. Specifically, it was shown to have a close proximity to the Karcher mean and a low computational time, and it served as a good initial guess in the optimization algorithms for the Karcher mean.

The Karcher mean, often considered as the most natural geometric mean of positive definite matrices, does not preserve any additional structure present in the given matrices. We have presented an adaptation of the Karcher mean to account for such structure. A theoretical analysis of this adaptation proved that many of the geometric mean properties remain valid in a suitably adjusted form. Afterwards, two steepest descent algorithms were designed for the computation of the *structured mean* in case of linear structures, where an appealing interaction between linear algebra and differential geometry could be observed. From a linear algebraic point of view, the two algorithms are identical except for their choice of a preconditioner of the gradient. From a differential geometric point of view, the algorithms were derived as Riemannian steepest descent methods on the set endowed with either the Euclidean geometry or the natural geometry of positive definite matrices. A convergence analysis was performed, showing the superiority of the algorithm based on the natural geometry.

Next, by thinking of the Karcher mean as a barycenter on the set of positive definite matrices, we presented a second adaptation specifically for positive definite Toeplitz matrices. This so-called *Kähler mean* is obtained by applying an application-inspired transformation to the matrices, after which the mean is computed as a barycenter in the transformed space. An explicit expression of the mean was provided for real matrices and a fast algorithm in the complex case. Theoretically, the mean does not satisfy many geometric mean properties relating to the ordering of matrices. However, it does deliver good results in the application from which its transformation was derived.

Finally, we considered two possible generalizations of the Kähler transformation towards *positive definite (Toeplitz-Block) Block-Toeplitz matrices*. In the second generalization, we encountered the Siegel disk, a generalization of the complex unit disk or Poincaré disk. We presented two possible geometries on the Siegel disk, one based on a useful automorphism of the set and another which corresponded naturally with the manifold. Using the generalized transformation which involved the Siegel disk with its natural geometry, we derived a global mean and a greedy approximation for Toeplitz-Block Block-Toeplitz matrices. An analysis of their properties, combined with numerical experiments, displayed

a close resemblance between the global and greedy versions of the mean, while the computational complexity was significantly lower for the greedy approximation.

We highlight our contributions in the chapters of the thesis:

Chapter 2

- An overview of the various instances of the geometric mean, its related means and approximations;
- The introduction of geometric mean approximations based on arithmetic and harmonic iterations. The randomized version of the HA mean was shown to have a close proximity to the Karcher mean and a low computational time, and it serves as a good initial guess in the optimization algorithms for the Karcher mean;

Chapter 3

- An explicit expression for the Levi–Civita connection and Riemannian Hessian of the Karcher cost function under the natural geometry of $\mathcal{P}_n$;
- The application of the Riemannian BFGS method to the computation of the Karcher mean;
- Using the Karcher mean and randomized HA mean as kernel fusion methods in a bioinformatics application, resulting in an improvement over the state-of-the-art protein fold classification methods;

Chapter 4

- The introduction of the structured geometric mean;
- A theoretical analysis of the structured mean, covering both its existence and its properties;
- The development and a convergence analysis of algorithms for the computation of the structured mean in case of linear structures;
- A beautiful link between linear algebra and differential geometry by using a Riemannian approach to develop two preconditioners for the gradient in a steepest descent algorithm;

Chapter 5

- An explicit expression for the Kähler mean of positive definite Toeplitz matrices in the real case and a fast algorithm to compute the Kähler mean in the complex case;

– The combination of the generalized transformation with the natural geometry of the Siegel disk, resulting in the global version of the generalized Kähler mean of Toeplitz-Block Block-Toeplitz matrices as a barycenter;
– A greedy approximation to the generalized Kähler mean, satisfying the same properties at a reduced computational cost.

6.2 Future research

We present some possibilities for future research that can be based on the topics mentioned in this thesis.

6.2.1 Derivative of the spectral norm

In Section 5.3.3, the Kobayashi distance measure d_K was introduced for the Siegel disk $\mathcal{SD}_N$. The computation of a barycenter based on this distance measure through gradient-based optimization would at some point require its derivation. Using the chain rule of derivation, we notice the appearance of the derivative of the spectral norm.

In Chapter 1, we presented the derivative of the spectral norm for any matrix X with a strictly largest singular value.When the largest singular value is no longer unique, the derivative becomes ill-defined. We demonstrate this by assuming the matrix X now has a largest singular value of multiplicity k, $\sigma_1 = \cdots = \sigma_k$. Using the classical definition of the Fréchet derivative, we find

$$\begin{aligned} \mathrm{D}(\|\cdot\|_2)(X)[E] &= \lim_{t\to 0} \frac{\|X+tE\|_2 - \|X\|_2}{t} \\ &= \lim_{t\to 0} \frac{\|I_k+tF\|_2 - 1}{t}, \end{aligned}$$

where $F = U_{1,k}^H E V_{1,k}$, and $U_{1,k}$ and $V_{1,k}$ are column matrices containing respectively the k left and right singular vectors of X corresponding to the singular values $\sigma_1, \ldots, \sigma_k$. In case the above limit exists, Weyl's theorem [45] dictates that it should lie within the interval $[-\delta_k, \delta_1]$, where δ_1 and δ_k are the largest and smallest singular value of F, respectively. In general, however, different results are obtained when considering the limit from above and the limit from below, indicating that the limit itself is not well-defined.

Hence, the implications concerning existence, uniqueness, and computation of a barycenter based on the Kobayashi distance are interesting, but non-trivial.

This computation was tested in Section 5.6.1, where convergence problems were observed for the Kobayashi barycenter.

6.2.2 Low displacement rank matrices

In Chapter 1, we briefly discussed matrices of low displacement rank, a structure which has proven its usefulness in various applications [72].

Since the displacement operator ∇_F is a linear transformation of matrices, the geometry of low displacement rank matrices is closely linked to that of low rank matrices. By working in the displaced setting, optimization can be performed in the same way as for low rank matrices.

The study of low rank matrices can be divided into two main branches. The first considers matrices of a fixed rank k, which are sometimes also assumed to be symmetric with fixed inertia [92, 122]. The second considers all matrices of a given rank k and lower, which is no longer a classical smooth manifold [38]. Losing the manifold structure requires the generalization of some Riemannian concepts, e. g., from tangent space to tangent cone [9, 99].

In case of positive definite low displacement rank matrices, the set of matrices of a given rank and lower might prove most useful. Consider for example the set of Toeplitz-like matrices discussed in the introduction, where we examine the mean of a Toeplitz matrix T and its inverse T^{-1}. Both matrices have a displacement rank 2 under ∇_F. The classical geometric mean of these two matrices is given by the identity matrix I, which has displacement rank 1. Hence, by considering all matrices of a given displacement rank and lower, this property of the geometric mean can be preserved.

We have analyzed the set of positive definite matrices of a given displacement rank or lower, and we have developed an algorithm for the computation of the structured geometric mean of low displacement rank matrices. This research will be submitted as part of

> Jeuris, B., and Vandebril, R. The structured geometric mean of low displacement rank matrices. *In preparation.*

6.2.3 Uniqueness of the structured geometric mean

The structured geometric mean was shown to exist for any closed subset $\mathcal{U}$ of $\mathcal{P}_n$ in Chapter 4. Uniqueness, however, was only shown in case of a geodesically

convex subset, in which case the structured geometric mean coincides with the Karcher mean.

In our numerical experiments involving linear structures, we did notice that the algorithms always returned the same solution, independent of any specifically constructed or random initial guess. This leads us to conjecture that the structured geometric mean is unique when the additional structure on $\mathcal{P}_n$ is linear.

6.2.4 A generalization of nonnegative matrix factorization

A matrix with positive (nonnegative) entries is often referred to as a positive (nonnegative) matrix. In nonnegative matrix factorization, the goal is to write a nonnegative matrix $M \in \mathbb{R}^{m\times n}$ as the product of two nonnegative matrices $U \in \mathbb{R}^{m\times k}, V \in \mathbb{R}^{k\times n}$, with k typically small.

A generalization of this concept is obtained by replacing the positive (nonnegative) entries in M and U with positive definite matrices. Hence, M and U can be represented as a grid containing positive definite matrices as entries, and we write $M \in \mathcal{P}_N^{m\times n}$ and $U \in \mathcal{P}_N^{m\times k}$. The factorization itself was generalized [129] by finding the minimizer of the cost function

$$f_E(U,V) = \sum_{i=1}^{m}\sum_{j=1}^{n} \|M_{ij} - \sum_{\ell=1}^{k} U_{i\ell}v_{\ell j}\|_F^2, \tag{6.1}$$

over $U \in \mathcal{P}_N^{m\times k}$ and positive $V \in \mathbb{R}^{k\times n}$. The generalization of the matrix product of U and V in the above expression ($\sum_{\ell=1}^{k} U_{i\ell}v_{\ell j}$) can be interpreted as a weighted arithmetic mean of the matrices in U, while the applied distance measure $\|A-B\|_F$ also stems from Euclidean geometry.

Replacing both this weighted averaging operation and the distance measure with their counterparts found in the natural geometry of $\mathcal{P}_N$ (Chapter 3), we obtain a new cost function

$$f_R(U,V) = \sum_{i=1}^{m}\sum_{j=1}^{n} \left\| \log\left(M_{ij}^{-\frac{1}{2}} K(v_{1j},\ldots,v_{kj};U_{i1},\ldots,U_{ik}) M_{ij}^{-\frac{1}{2}} \right) \right\|_F^2,$$

where $K(w_1,\ldots,w_k;A_1,\ldots,A_k)$ represents the weighted Karcher mean of matrices $A_1,\ldots,A_k \in \mathcal{P}_N$ with corresponding weights $w_1,\ldots,w_k \geq 0$.

By endowing the product manifold $\mathcal{P}_N^{m\times n}$ with the product geometry of $\mathcal{P}_N$, the optimization problem can be tackled using Riemannian optimization. An algorithm for the computation of the decomposition has been developed and was tested on artificial datasets. This research will be submitted as part of

Iannazzo, B., Jeuris, B., and Pompili, F. The Karcher decomposition of tensor grids. *In preparation.*

Bibliography

[1] Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. pages 12, 43, 44, 46, 49, 50, 51, 52, 54, 58, 65, 77, 103, 133

[2] Adler, R. L., Dedieu, J.-P., Margulies, J. Y., Martens, M., and Shub, M. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA Journal of Numerical Analysis 22*, 3 (2002), pp. 359–390. pages 46

[3] Al-Mohy, A. H., and Higham, N. J. Improved inverse scaling and squaring algorithms for the matrix logarithm. *SIAM Journal on Scientific Computing 34*, 4 (2012), pp. 153–169. pages 58

[4] Al-Mohy, A. H., Higham, N. J., and Relton, S. D. Computing the Fréchet derivative of the matrix logarithm and estimating the condition number. *SIAM Journal on Scientific Computing 35*, 4 (2013), pp. C394–C410. pages 58

[5] Anderson, W., Morley, T., and Trapp, G. Symmetric function means of positive operators. *Linear Algebra and its Applications 60* (1984), pp. 129–143. pages 12, 32

[6] Ando, T., Li, C., and Mathias, R. Geometric means. *Linear Algebra and its Applications 385* (2004), pp. 305–334. pages 12, 19, 21, 22, 27, 30, 80, 87

[7] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications 29*, 1 (2007), pp. 328–347. pages 12, 30

[8] Atteia, M., and Raissouli, M. Self dual operators on convex functionals; geometric mean and square root of convex functionals. *Journal of Convex Analysis 8*, 1 (2001), pp. 223–240. pages 19, 34

[9] Aubin, J. *Applied Functional Analysis.* Wiley-Interscience, 2000. pages 147

[10] Bach, F. R., Lanckriet, G. R., and Jordan, M. I. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the* 21*th International Conference on Machine Learning (ICML)* (2004), Omnipress, p. 6. pages 75

[11] Barbaresco, F. Information intrinsic geometric flows. *American Institute of Physics Conference Series 872* (2006), pp. 211–218. pages 13, 108, 111, 113

[12] Barbaresco, F. Innovative tools for radar signal processing based on Cartan's geometry of SPD matrices and information geometry. In *RADAR '08, IEEE International Radar Conference, Rome* (2008), pp. 1–6. pages 18, 80, 103, 108

[13] Barbaresco, F. Interactions between symmetric cone and information geometries: Bruhat-Tits and Siegel spaces models for high resolution autoregressive Doppler imagery. *Lecture Notes in Computer Science 5416* (2009), pp. 124–163. pages 13, 108, 111, 113, 118, 140

[14] Barbaresco, F. Robust statistical radar processing in Fréchet metric space: OS-HDR-CFAR and OS-STAP processing in Siegel homogeneous bounded domains. In *IRS '11, International Radar Conference, Leipzig* (2011), pp. 639–644. pages 80, 103, 108

[15] Barbaresco, F. *Information Geometry of Covariance Matrix: Cartan-Siegel Homogeneous Bounded Domains, Mostow/Berger Fibration and Fréchet Median.* Springer, 2012, ch. 9, pp. 199–255. pages 80, 103, 108, 118, 140

[16] Barbaresco, F. Information geometry manifold of Toeplitz Hermitian positive definite covariance matrices: Mostow/Berger fibration and Berezin quantization of Cartan-Siegel domains. *International Journal of Emerging Trends in Signal Processing 1*, 3 (2013), pp. 1–11. pages 13, 113, 126, 129

[17] Barbaresco, F. *Eidetic Reduction of Information Geometry Through Legendre Duality of Koszul Characteristic Function and Entropy: From Massieu–Duhem Potentials to Geometric Souriau Temperature and Balian Quantum Fisher Metric.* Springer, 2014, ch. 7, pp. 141–217. pages 123

[18] Bassanelli, G. On horospheres and holomorphic endomorfisms of the Siegel disc. *Rendiconti del Seminario Matematico della Università di Padova 70* (1983), pp. 147–165. pages 126

[19] Batchelor, P., Moakher, M., Atkinson, D., Calamante, F., and Connelly, A. A rigorous framework for diffusion tensor calculus. *Magnetic Resonance in Medicine 53* (2005), pp. 221–225. pages 80

[20] Bellman, R. *Introduction to Matrix Analysis*. McGraw–Hill, 1970. pages 95

[21] Bhatia, R. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, 2007. pages 6, 12, 20, 45, 47, 79, 82, 110

[22] Bhatia, R., and Holbrook, J. Riemannian geometry and matrix geometric means. *Linear Algebra and its Applications 413*, 2-3 (2006), pp. 594–618. pages 79

[23] Bhatia, R., and Karandikar, R. L. Monotonicity of the matrix geometric mean. *Mathematische Annalen 353*, 4 (2012), pp. 1453–1467. pages 44, 45

[24] Bini, D., Iannazzo, B., Jeuris, B., and Vandebril, R. Geometric means of structured matrices. *BIT Numerical Mathematics 54*, 1 (2014), pp. 55–83. pages 15

[25] Bini, D., Meini, B., and Poloni, F. An effective matrix geometric mean satisfying the Ando–Li–Mathias properties. *Mathematics of Computation 79*, 269 (2010), pp. 437–452. pages 12, 19, 21, 22, 23, 27

[26] Bini, D., and Pan, V. *Polynomial and Matrix Computations*. Birkhäuser, 1994. pages 86, 102, 103

[27] Bini, D. A., and Iannazzo, B. Computing the Karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications 438*, 4 (2011), pp. 1700–1710. pages 12, 52, 81, 91

[28] Bini, D. A., and Iannazzo, B. A note on computing matrix geometric means. *Advances in Computational Mathematics 35*, 2–4 (2011), pp. 175–192. pages 12, 21, 24, 25, 27, 30, 69

[29] Bini, D. A., Latouche, G., and Meini, B. *Numerical Methods for Structured Markov Chains*. Oxford University Press, 2005. pages 10

[30] Bishop, R. L., and O'Neill, B. Manifolds of negative curvature. *Transactions of the American Mathematical Society 145* (1969), pp. 1–49. pages 45

[31] Bonnabel, S., Collard, A., and Sepulchre, R. Rank-preserving geometric means of positive semi-definite matrices. *Linear Algebra and its Applications 438*, 8 (2013), pp. 3202–3216. pages 11

[32] Bonnabel, S., and Sepulchre, R. Geometric distance and mean for positive semi-definite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications 31*, 3 (2009), pp. 1055–1070. pages 77

[33] Boothby, W. M. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, 1975. pages 43, 46

[34] Borwein, J., and Borwein, P. *Pi and the AGM: a Study in the Analytic Number Theory and Computational Complexity.* Wiley-Interscience, 1987. pages 34

[35] Böttcher, A., and Grudsky, S. M. *Spectral Properties of Banded Toeplitz Matrices.* SIAM, 2005. pages 122

[36] Bridson, M., and Haefliger, A. *Metric Spaces of Non-Positive Curvature.* Springer, 1999. pages 45, 82

[37] Cartan, E. *Leçons sur la Géométrie des Espaces de Riemann.* Gauthier-Villars, 1928. pages 45, 82, 115

[38] Cason, T., Absil, P.-A., and Van Dooren, P. Iterative methods for low rank approximation of graph similarity matrices. *Linear Algebra and its Applications 438*, 4 (2013), pp. 1863–1882. pages 147

[39] Chen, K., and Kurgan, L. PFRES: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics 23*, 21 (2007), pp. 2843–2850. pages 73

[40] Chu, M., and Golub, G. Structured inverse eigenvalue problems. *Acta Numerica 11* (2002), pp. 1–71. pages 103

[41] Damanik, D., Pushnitski, A., and Simon, B. The analytic theory of matrix orthogonal polynomials. *Surveys in Approximation Theory 4* (2008), pp. 1–85. pages 123

[42] Damoulas, T., and Girolami, M. A. Pobabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics 24*, 10 (2008), pp. 1264–1270. pages 73

[43] Dehaene, J. *Continuous-time matrix algorithms, systolic algorithms and adaptive neural networks.* PhD thesis, Department of Electrical Engineering, KU Leuven, Belgium, 1995. pages 5

[44] Delsarte, P., Genin, Y. V., and Kamp, Y. G. Orthogonal polynomial matrices on the unit circle. *IEEE Transactions on Circuits and Systems 25*, 3 (1978), pp. 149–160. pages 13, 122, 123

[45] Demmel, J. W. *Applied Numerical Linear Algebra*, 2 ed. SIAM, 1997. pages 146

[46] Dennis, J., and Schnabel, R. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Springer, 1983. pages 77

[47] Dette, H., and Studden, W. J. *The Theory of Canonical Moments with Applications in Statistics, Probability, and Analysis*. Wiley and Sons, 1997. pages 122

[48] Dette, H., and Wagener, J. Matrix measures on the unit circle, moment spaces, orthogonal polynomials and the Geronimus relations. *Linear Algebra and its Applications 432*, 7 (2010), pp. 1609–1626. pages 13, 123

[49] Ding, C. H., and Dubchak, I. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics 17*, 4 (2001), pp. 349–358. pages 73

[50] Dubchak, I., Muchnik, I., Holbrook, S. R., and Kim, S.-H. Prediction of protein folding class using global description of amino acid sequence. *Proceedings of the National Academy of Sciences of the United States of America 92*, 19 (1995), pp. 8700–8704. pages 73

[51] Eidelman, Y., and Gohberg, I. On a new class of structured matrices. *Integral Equations and Operator Theory 34*, 3 (1999), pp. 293–324. pages 86

[52] Elmachtoub, A., and Van Loan, C. From random polygon to ellipse: an eigenanalysis. *SIAM Review 52*, 1 (2010), pp. 151–170. pages 36, 37, 38

[53] Ferreira, R., Xavier, J., Costeira, J., and Barroso, V. Newton method for Riemannian centroid computation in naturally reductive homogeneous spaces. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (2006), IEEE, p. p. 3. pages 12, 46, 51, 52

[54] Ferreira, R., Xavier, J., Costeira, J., and Barroso, V. Newton algorithms for Riemannian distance related problems on connected locally symmetric manifolds. *IEEE Selected Topics in Signal Processing 7*, 4 (2013), pp. 634–645. pages 12, 58, 59

[55] Fiori, S. Learning the Fréchet mean over the manifold of symmetric positive-definite matrices. *Cognitive Computation 1*, 4 (2009), pp. 279–291. pages 52

[56] Fletcher, P., and Joshi, S. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing 87*, 2 (2007), pp. 250–262. pages 18

[57] Foster, D., and Phillips, G. The arithmetic-harmonic mean. *Mathematics of Computation 42* (1984), pp. 183–191. pages 34

[58] Franzoni, T., and Vesentini, E. *Holomorphic Maps and Invariant Distances*, vol. 40 of *Mathematics studies*. North-Holland, 1980. pages 126, 127

[59] Fritzsche, B., and Kirstein, B. An extension problem for non-negative Hermitian block Toeplitz matrices. *Mathematische Nachrichten 131*, 1 (1987), pp. 287–297. pages 123

[60] Gallivan, K. A., Qi, C., and Absil, P.-A. A Riemannian Dennis–Moré condition. In *High-Performance Scientific Computing - Algorithms and Applications*. Springer, 2012, pp. 281–293. pages 77

[61] Gaubert, S., and Qu, Z. The contraction rate in Thompson's part metric of order-preserving flows on a cone – Application to generalized Riccati equations. *Journal of Differential Equations 256*, 8 (2014), pp. 2902–2948. pages 84

[62] Giles, M. An extended collection of matrix derivative results for forward and reverse mode algorithmic differentiation. Tech. rep., Oxford University Computing Laboratory, Oxford, England, 2008. pages 6

[63] Golub, G. H., and Van Loan, C. F. *Matrix Computations*, 4 ed. Johns Hopkins University Press, 2013. pages 5

[64] Higham, N. J. *Functions of Matrices: Theory and Computation*. SIAM, 2008. pages 5, 58, 93

[65] Horn, R. A., and Johnson, C. R. *Matrix Analysis*. Cambridge University Press, 1985. pages 2

[66] Horn, R. A., and Johnson, C. R. *Topics in Matrix Analysis*. Cambridge University Press, 1991. pages 63

[67] Humet, M., and Van Barel, M. Algorithms for the Geronimus transformation for orthogonal polynomials on the unit circle. *Journal of Computational and Applied Mathematics 267* (2014), pp. 195–217. pages 122

[68] Iannazzo, B. The geometric mean of two matrices from a computational viewpoint, 2011. Available on arXiv - http://arxiv.org/abs/1201.0101v1. pages 82

[69] Jakobsson, A., Marple, S. L., and Stoica, P. Computationally efficient two-dimensional Capon spectrum analysis. *IEEE Transactions on Signal Processing 48*, 9 (2000), pp. 2651–2661. pages 13, 119

[70] Jeuris, B., and Vandebril, R. Geometric mean algorithms based on harmonic and arithmetic iterations. *Lecture Notes in Computer Science 8085* (2013), pp. 785–793. pages 14

[71] Jeuris, B., Vandebril, R., and Vandereycken, B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis 39*, 1 (2012), pp. 379–402. pages 14

[72] Kailath, T., and Sayed, A. *Fast Reliable Algorithms for Matrices with Structure*. SIAM, 1999. pages 11, 147

[73] Kanhouche, R. A modified Burg algorithm equivalent in results to Levinson algorithm, 2003. Available on arXiv - http://arxiv.org/abs/math/0309384v1. pages 13, 112, 119, 120

[74] Karcher, H. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics 30*, 5 (1977), pp. 509–541. pages 12, 82, 89, 110

[75] Kay, S. M. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice–Hall, 1993. pages 119

[76] Kendall, W. Probability, convexity, and harmonic maps with small image I: uniqueness and fine existence. *Proceedings London Mathematical Society 61*, 2 (1990), pp. 371–406. pages 45, 82, 97, 110, 115

[77] Koecher, M. Positivitätsbereiche im R^n. *American Journal of Mathematics 79* (1957), pp. 575–596. pages 6

[78] Lanckriet, G. R. G., Cristianini, N., Bartlett, P., El Ghaoui, L., and Jordan, M. I. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research 5* (2004), pp. 27–72. pages 75

[79] Lanckriet, G. R. G., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S. A statistical framework for genomic data fusion. *Bioinformatics 20*, 16 (2004), pp. 2626–2635. pages 75

[80] Lang, S. *Fundamentals of Differential Geometry*, vol. 191 of *Graduate Texts in Mathematics*. Springer, 1999. pages 82, 97, 110

[81] Lapuyade-Lahorgue, J., and Barbaresco, F. Radar detection using Siegel distance between autoregressive processes, application to HF and X-band radar. In *RADAR '08, IEEE International Radar Conference, Rome* (2008). pages 80, 103, 108, 118, 140

[82] Lawson, J., and Lim, Y. The geometric mean, matrices, metrics, and more. *American Mathematical Monthly 108*, 9 (2001), pp. 797–812. pages 82

[83] Lawson, J., and Lim, Y. Monotonic properties of the least squares mean. *Mathematische Annalen 351*, 2 (2011), pp. 267–279. pages 44, 45

[84] Lee, J. M. *Manifolds and Differential Geometry*. Graduate Studies in Mathematics. American Mathematical Society, 2009. pages 43, 46

[85] Lim, Y. Riemannian and Finsler structures of symmetric cones. *Trends in Mathematics 4*, 2 (2001), pp. 111–118. pages 45, 82

[86] Lim, Y. Stopping criteria for the Ando-Li-Mathias and Bini-Meini-Poloni geometric means. *Linear Algebra and its Applications 434* (2001), pp. 1884–1892. pages 12, 35

[87] Lim, Y. Multi-variable weighted geometric means of positive definite matrices. *Linear Algebra and its Applications 435*, 2 (2011), pp. 307–322. pages 12, 23

[88] Lim, Y. Factorizations and geometric means of positive definite matrices. *Linear Algebra and its Applications 437*, 9 (2012), pp. 2159–2172. pages 12, 35

[89] Lim, Y., and Palfia, M. Matrix power means and the Karcher mean. *Journal of Functional Analysis 262*, 4 (2012), pp. 1498–1514. pages 12, 32, 33

[90] Lin, C., Zou, Y., Qin, J., Liu, X., Jiang, Y., Ke, C., and Zou, Q. Hierarchical classification of protein folds using a novel ensemble classifier. *PLoS ONE 8*, 2 (2013). e56499. doi:10.1371/journal.pone.0056499. pages 73, 76

[91] Marple, S. L. *Digital Spectral Analysis with Applications*. Prentice-Hall. Englewood Cliffs, 1980. pages 119, 120

[92] Mishra, B., Meyer, G., Bonnabel, S., and Sepulchre, R. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics 29*, 3–4 (2014), pp. 591–621. pages 147

[93] Moakher, M. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications 26*, 3 (2005), pp. 735–747. pages 46, 79, 89

[94] Moakher, M. On the averaging of symmetric positive-definite tensors. *Journal of Elasticity 82*, 3 (2006), pp. 273–296. pages 18, 80

[95] Moakher, M., and Zéraï, M. The Riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data. *Journal of Mathematical Imaging and Vision 40*, 2 (2011), pp. 171–187. pages 46

[96] Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology 247*, 4 (1995), pp. 536–540. pages 73, 76

[97] Nakamura, K. Geometric means of positive operators. *Kyunpook Mathematical Journal 49* (2009), pp. 167–181. pages 12, 21, 22

[98] Nocedal, J., and Wright, S. J. *Numerical Optimization*. Springer, 1999. pages 53

[99] O'Shea, D., and Wilson, L. Limits of tangent spaces to real surfaces. *American Journal of Mathematics 126*, 5 (2004), pp. 951–980. pages 147

[100] Palfia, M. The Riemannian barycenter computation and means of several matrices. *International Journal of Computational and Mathematical Sciences 3*, 3 (2009), pp. 128–133. pages 12, 36, 46, 52

[101] Palfia, M. A multivariable extension of two-variable matrix means. *SIAM Journal on Matrix Analysis and Applications 32*, 2 (2011), pp. 385–393. pages 12, 35

[102] Pan, V. Y. *Structured Matrices and Polynomials*. Birkhäuser, Springer, 2001. pages 10

[103] Poloni, F. Constructing matrix geometric means. *Electronic Journal of Linear Algebra*, 20 (2010), pp. 419–435. pages 12, 21

[104] Powell, P. D. Calculating determinants of block matrices, 2011. Available on arXiv - http://arxiv.org/abs/1112.4379v1. pages 121

[105] Qi, C. *Numerical optimization methods on Riemannian manifolds*. PhD thesis, Florida State University, College of arts and sciences, United States of America, 2011. pages 12, 58, 62

[106] Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. SimpleMKL. *Journal of Machine Learning Research 9* (2008), pp. 2491–2521. pages 75

[107] Rathi, Y., Tannenbaum, A., and Michailovich, O. Segmenting images on the tensor manifold. In *IEEE Conference on Computer Vision and Pattern Recognition* (2007), IEEE, pp. 1–8. pages 18

[108] Rentmeesters, Q. *Algorithms for data fitting on some common homogeneous spaces*. PhD thesis, Université Catholique de Louvain (UCL), Belgium, 2013. pages 60, 66

[109] Rentmeesters, Q., and Absil, P.-A. Algorithm comparison for Karcher mean computation of rotation matrices and diffusion tensors. In *Proceedings of the 19th European Signal Processing Conference* (2011), pp. 2229–2233. pages 12, 46, 52, 58, 59, 60, 66

[110] Sagae, M., and Tanabe, K. Upper and lower bounds for the arithmetic-geometric-harmonic means of positive definite matrices. *Linear and Multilinear Algebra 37*, 4 (1993), pp. 279–282. pages 12, 35

[111] Savas, B., and Lim, L.-H. Quasi-Newton methods on Grassmannians and multilinear approximations of tensors. *SIAM Journal on Scientific Computing 32* (2010), pp. 3352–3393. pages 12, 58, 62

[112] Siegel, C. L. Symplectic geometry. *American Journal of Mathematics 65*, 1 (1943), pp. 1–86. pages 122, 125

[113] Sinap, A., and Van Assche, W. Orthogonal matrix polynomials and applications. *Journal of Computational and Applied Mathematics 66*, 1–2 (1996), pp. 27–52. pages 123

[114] Skovgaard, L. T. A Riemannian geometry of the multivariate normal model. *Scandinavian Journal of Statistics 11* (1984), pp. 211–223. pages 6

[115] Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. Large scale multiple kernel learning. *Journal of Machine Learning Research 7* (2006), pp. 1531–1565. pages 75

[116] Sun, Z., Ampornpunt, N., Varma, M., and Vishwanathan, S. Multiple kernel learning and the SMO algorithm. In *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 2361–2369. pages 75

[117] Szegő, G. *Orthogonal Polynomials*. American Mathematical Society, 1939. pages 123

[118] Therrien, C. Relations between 2-D and multichannel linear prediction. *IEEE Transactions on Acoustics, Speech and Signal Processing 29*, 3 (1981), pp. 454–456. pages 13, 119

[119] Thompson, A. On certain contraction mappings in a partially ordered vector space. *Proceedings of the American Mathematical Society 14*, 3 (1963), pp. 438–443. pages 84, 127

[120] Vandebril, R., Van Barel, M., and Mastronardi, N. *Matrix Computations and Semiseparable Matrices I: Linear Systems*. Johns Hopkins University Press, 2008. pages 86

[121] Vandereycken, B., Absil, P.-A., and Vandewalle, S. Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank. In *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on* (2009), IEEE, pp. 389–392. pages 46

[122] Vandereycken, B., Absil, P.-A., and Vandewalle, S. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *IMA Journal of Numerical Analysis 33* (2013), pp. 481–514. pages 77, 147

[123] Vandereycken, B., and Vandewalle, S. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications 31*, 5 (2010), pp. 2553–2579. pages 77

[124] Verblunsky, S. On positive harmonic functions: a contribution to the algebra of Fourier series. *Proceedings London Mathematical Society 38* (1935), pp. 125–157. pages 122

[125] Verblunsky, S. On positive harmonic functions (second paper). *Proceedings London Mathematical Society 40* (1936), pp. 290–320. pages 122

[126] Whittle, P. On the fitting of multivariate autoregressions, and the approximate canonical factorization of a spectral density matrix. *Biometrika 50*, 1/2 (1963), pp. 129–134. pages 120

[127] Wiener, N. *The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction*, 1 ed. MIT Press, 1949, pp. 129–148. pages 13, 112

[128] Wiggins, R. A., and Robinson, E. A. Recursive solution to the multichannel fitering problem. *Journal of Geophysical Research 70*, 8 (2012), pp. 1885–1891. pages 120

[129] Xie, Y., Ho, J., and Vemuri, B. Nonnegative Factorization of Diffusion Tensor Images and Its Applications. In *International Conference on Information Processing in Medical Imaging (IPMI)* (2011). pages 148

[130] Yang, L. *Medians of probability measures in Riemannian manifolds and applications to radar target detection*. PhD thesis, Université de Poitiers, France, 2011. pages 80, 103, 108, 118, 140

[131] Yang, L., Arnaudon, M., and Barbaresco, F. Geometry of covariance matrices and computation of median. *American Institute of Physics Conference Series 1305* (2011), pp. 479–486. pages 113, 116

[132] Yang, T., Kecman, V., Cao, L., Zhang, C., and Huang, J. Z. Margin-based ensemble classifier for protein fold recognition. *Expert Systems with Applications 38*, 10 (2011), pp. 12348–12355. pages 73

[133] Ying, Y., Huang, K., and Campbell, C. Enhanced protein fold recognition through a novel data integration approach. *BMC Bioinformatics 10* (2009), p. 267. pages 76

[134] Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Protein fold recognition using geometric kernel data fusion. *Bioinformatics 30*, 13 (2014), pp. 1850–1857. pages 14, 72

[135] Zhang, F. *Matrix Theory: Basic Results and Techniques*, 2 ed. Springer, 2011. pages 2

Curriculum vitae

Personalia

Name: Ben Jeuris
Date of birth: August 16, 1988
Place of birth: Diest, Belgium

Education

2011 – 2015
Ph.D. in Engineering, *Doctoraatsopleiding in de Ingenieurswetenschappen*,
KU Leuven, Leuven, Belgium.

2009 – 2011
Master of Science in Engineering: Mathematical Engineering, *Master in de Ingenieurswetenschappen: Wiskundige Ingenieurstechnieken*,
KU Leuven, Leuven, Belgium.

2006 – 2009
Bachelor in Mathematics, *Bachelor in de Wiskunde*,
UHasselt, Diepenbeek, Belgium.

Research stays abroad

March 16, 2012 – April 15, 2012
Università degli Studi di Perugia, Perugia, Italy,
visiting Prof. B. Iannazzo

May 26, 2013 – June 22, 2013
Università degli Studi di Perugia, Perugia, Italy,
visiting Prof. B. Iannazzo

September 29, 2014 – October 29, 2014
Università degli Studi di Perugia, Perugia, Italy,
visiting Prof. B. Iannazzo

Teaching

Teaching assistant for Numerical Mathematics (*Numerieke Wiskunde*):
2013–2014, 2014–2015.

Teaching assistant for Numerical Modeling and Approximation (*Numerieke Modellering en Benadering*):
2011–2012, 2012–2013, 2013–2014, 2014–2015.

List of publications

Publications

2015

Jeuris, B., and Vandebril, R. The Kähler mean of Block-Toeplitz matrices with Toeplitz structured blocks. Submitted to *SIAM journal on Matrix Analysis and Applications*.

2014

Bini, D., Iannazzo, B., Jeuris, B., and Vandebril, R. Geometric means of structured matrices. *BIT Numerical Mathematics 54*, 1 (2014), 55–83.

Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Protein fold recognition using geometric kernel data fusion. *Bioinformatics 30*, 13 (2014), 1850–1857.

Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Application of geometric kernel data fusion in protein fold recognition and protein sub-nuclear localization. Accepted for publication in the proceedings of *Machine Learning in Systems Biology (MLSB)*, Strasbourg, France, 2014.

2013

Jeuris, B., and Vandebril, R. Geometric mean algorithms based on harmonic and arithmetic iterations. *Lecture Notes in Computer Science 8085* (2013), 785–793.

2012

Jeuris, B., Vandebril, R., and Vandereycken, B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis 39*, 1 (2012), 379–402.

Articles in preparation

Iannazzo, B., Jeuris, B., and Pompili, F. The Karcher decomposition of tensor grids. *In preparation.*

Jeuris, B., and Vandebril, R. The structured geometric mean of low displacement rank matrices. *In preparation.*

Seminars

2013
Jeuris, B. What is ... manifold optimization? KU Leuven, Department of Mathematics, Leuven, Belgium, December 9, 2013.

Jeuris, B., Bini, D., Iannazzo, B., and Vandebril, R. Matrix means through manifold optimization. KU Leuven, Department of Computer Science, Leuven, Belgium, December 11, 2013.

2012
Vandebril, R., Jeuris, B., Vandereycken, B., and Iannazzo, B. How differential geometry assists in computing the matrix geometric mean. KU Leuven, Department of Mathematics, Leuven, Belgium, February 28, 2012.

Jeuris, B., Vandebril, R., and Vandereycken, B. The Karcher mean: optimization on matrix manifolds. Università degli Studi di Perugia, Department of Mathematics and Informatics, Perugia, Italy, April 12, 2012.

Conference presentations

2015
Jeuris, B., Iannazzo, B., and Pompili, F. Nonnegative tensor grid decomposition. *Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM)*, Lecce, Italy, March 23–27, 2015. (Invited)

2014
Jeuris, B., Vandebril, R., Bini, D., and Iannazzo, B. The geometric matrix mean: an adaptation for structured matrices. *Householder Symposium*, Spa, Belgium, June 9–13, 2014. (Invited)

Jeuris, B., Vandebril, R., Bini, D., and Iannazzo, B. Barycenters of structured matrices and their computation. *International Linear Algebra Society (ILAS) Conference*, Seoul, Korea, August 6–9, 2014. (Invited)

Jeuris, B., and Vandebril, R. Optimization on structured matrix manifolds: Toeplitz and TBBT matrices. *IMA Conference on Numerical Linear Algebra and Optimization*, Birmingham, United Kingdom, September 3–5, 2014.

Zakeri, P., Jeuris, B., Vandebril, R., and Moreau, Y. Application of geometric kernel data fusion in protein fold recognition and protein subnuclear localization. *Machine Learning in Systems and Biology (MLSB)*, Strasbourg, France, September 7–10, 2014.

2013

Vandebril, R., Jeuris, B., Bini, D., Vandereycken, B., and Iannazzo, B. How differential geometry assists in computing the matrix geometric mean. *Structured matrix days*, Limoges, France, May 10–11, 2013.

Jeuris, B., Vandebril, R., Bini, D., and Iannazzo, B. The structured geometric mean. *European Conference on Computational Optimization (EUCCO)*, Chemnitz, Germany, July 17–19, 2013. (Invited)

Jeuris, B., and Vandebril, R. Geometric mean algorithms based on harmonic and arithmetic iterations. *Geometric Science of Information (GSI) Conference*, Paris, France, August 28–30, 2013.

2012

Jeuris, B., Vandebril, R., and Vandereycken, B. The Karcher mean: first and second order optimization techniques on matrix manifolds. *Siam Conference on Applied Linear Algebra*, Valencia, Spain, June 18–22, 2012.

Jeuris, B., Vandebril, R., and Vandereycken, B. Matrix manifold optimization: contemporary algorithms for computing the matrix geometric mean. *International Congress on Computational and Applied Mathematics (ICCAM)*, Ghent, Belgium, July 9–13, 2012.

2011

Jeuris, B., Vandebril, R., and Vandereycken, B. The matrix geometric mean and manifold optimization. *International Linear Algebra Society (ILAS) Conference*, Braunschweig, Germany, August 22–25, 2011.

Jeuris, B., Vandebril, R., and Vandereycken, B. The Karcher mean: Solving a particular matrix equation through matrix manifold

optimization. *Workshop on Matrix Equations and Tensor Techniques*, RWTH Aachen, Aachen, Germany, November 21–22, 2011.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
NUMERICAL APPROXIMATION AND LINEAR ALGEBRA GROUP
Celestijnenlaan 200A box 2402
B-3001 Leuven
ben.jeuris@kuleuven.be