Structured Matrix Nearness Problems: Theory and Algorithms

Borsdorf, Ruediger

2012

MIMS EPrint: **2012.63**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Structured Matrix Nearness Problems: Theory and Algorithms

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2012

Ruediger Borsdorf
School of Mathematics

Contents

Word count 65946

List of Tables

List of Figures

List of Algorithms

The University of Manchester

Ruediger Borsdorf **June 11, 2012**
Doctor of Philosophy
Structured Matrix Nearness Problems: Theory and Algorithms

In many areas of science one often has a given matrix, representing for example a measured data set and is required to find a matrix that is closest in a suitable norm to the matrix and possesses additionally a structure, inherited from the model used or coming from the application. We call these problems structured matrix nearness problems. We look at three different groups of these problems that come from real applications, analyze the properties of the corresponding matrix structure, and propose algorithms to solve them efficiently.

The first part of this thesis concerns the nearness problem of finding the nearest k factor correlation matrix $C(X) = \operatorname{diag}(I_n - XX^T) + XX^T$ to a given symmetric matrix, subject to natural nonlinear constraints on the elements of the $n \times k$ matrix X, where distance is measured in the Frobenius norm. Such problems arise, for example, when one is investigating factor models of collateralized debt obligations (CDOs) or multivariate time series. We examine several algorithms for solving the nearness problem that differ in whether or not they can take account of the nonlinear constraints and in their convergence properties. Our numerical experiments show that the performance of the methods depends strongly on the problem, but that, among our tested methods, the spectral projected gradient method is the clear winner.

In the second part we look at two two-sided optimization problems where the matrix of unknowns $Y \in \mathbb{R}^{n\times p}$ lies in the Stiefel manifold. These two problems come from an application in atomic chemistry where one is looking for atomic orbitals with prescribed occupation numbers. We analyze these two problems, propose an analytic optimal solution of the first and show that an optimal solution of the second problem can be found by solving a convex quadratic programming problem with box constraints and p unknowns. We prove that the latter problem can be solved by the active-set method in at most $2p$ iterations. Subsequently, we analyze the set of optimal solutions $\mathcal{C} = \{Y \in \mathbb{R}^{n\times p} : Y^TY = I_p, Y^TNY = D\}$ of the first problem for N symmetric and D diagonal and find that a slight modification of it is a Riemannian manifold. We derive the geometric objects required to make an optimization over this manifold possible. We propose an augmented Lagrangian-based algorithm that uses these geometric tools and allows us to optimize an arbitrary smooth function over $\mathcal{C}$. This algorithm can be used to select a particular solution out of the latter set $\mathcal{C}$ by posing a new optimization problem. We compare it numerically with a similar algorithm that, however, does not apply these geometric tools and find that our algorithm yields better performance.

The third part is devoted to low rank nearness problems in the Q-norm, where the matrix of interest is additionally of linear structure, meaning it lies in the set spanned by s predefined matrices $U_1, \ldots, U_s \in \{0, 1\}^{n\times p}$. These problems are often associated with model reduction, for example in speech encoding, filter design, or latent semantic indexing. We investigate three approaches that support any linear structure and examine further the geometric reformulation by Schuermans et al. (2003). We improve their algorithm in terms of reliability by applying the augmented Lagrangian method and show in our numerical tests that the resulting algorithm yields better performance than other existing methods.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://www.campus.manchester.ac.uk/medialibrary/policies/intellectual-property.pdf`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.manchester.ac.uk/library/aboutus/regulations`) and in The University's policy on presentation of Theses.

Publications

i. The material of Chapter 2 is based on the paper:

R. Borsdorf, N. J. Higham, and M. Raydan. Computing a nearest correlation matrix with factor structure. *SIAM J. Matrix Anal. Appl.*, 30(5):2603–2622, 2010.

ii. The material of Chapter 4 will be part of the paper:

R. Borsdorf. Two-sided optimization problems with orthogonal constraints arising in chemisty. In preparation.

Acknowledgements

At first I thank my supervisor Prof. Higham for his great support, helpful comments, his presence in any situation, that he was always approachable, and of course, for his endless reading of my thesis. I thank him for introducing me to the numerical analysis community and for always supporting my ideas. It has been a great pleasure to work with him over the last few years and to learn from his expertise.

A big thank you goes to the people of NAG: Sven Hammarling, Dr David Sayers, Dr Mike Dewar, John Holden, and in particular Dr Craig Lucas. He gave me great support during my PhD and he was always motivating me to continue my work. I am grateful that he introduced me to NAG, invited me whenever possible to mathematical discussions, and guided me through the process of including code in the NAG library. I also thank NAG for providing the MATLAB-NAG Toolbox and their financial support. I am also grateful for the helpful comments that my examiners Dr Françoise Tissuer and Prof. Pierre-Antoine Absil gave me during my viva. I would like to mention Prof. Marcos Raydan, Dr Bart Vandereycken, Prof. Alastair Spence, and Dr Martin Lotz and thank them for their mathematical exchange on particular sections of my thesis. I thank the Research Office and the School of Mathematics for their financial support and for making visits to all the conferences possible.

I would like to thank some very good friends with whom I spent a great time outside of the working hours: Dr Dmitry Yumashev, Dr Tiziano De Angelis, Elena Issoglio, Dr Christopher Munro, Bernd Liebig, Florian Kleinert, Dr Piotr Kowalczyk, Andreas Papayiannis, and Vasileios Tsitsopoulos. This time includes the football matches every week and the endless exhausting but great squash tournaments that I will definitely miss. A big thank goes to very special friends. Thank you Yuji for the exciting mathematical discussions and devastating squash defeats but especially for the great time during the last year here in Manchester. Thank you Lijing for reading my thesis and the hours of mathematical chats, for the amazing time, for always being a good listener and for encouraging me all the time. Thank you Kyriakos for all the time that never seems to become boring and for being such a good flatmate. Without you the time here would have not been the same. Thank you Micha for being such a good friend. Finally, I thank my family for their support in every respect.

To my parents

Chapter 1

Introduction

1.1 Emergence of Matrix Nearness Problems

1.1.1 Modelling of Complex Systems

The fundamental principle of science is to understand complex systems and their interactions that exist in nature and that influence our daily lives. In order to obtain a better insight into these systems researchers are trying to reduce the complexity by simplifying these systems through models that allow them to observe their principles and to predict eventually their behaviour. Having understood these systems one can improve current technologies that make our lives easier or open new opportunities for our human society. For instance to construct modern planes or bridges of large dimensions certainly one cannot avoid the usage of models to understand the physics behind and to eventually build them in such a way that they show stable behaviour, even under extreme weather conditions.

To find the appropriate model that is as simple as possible, but still reflects the main characteristics of these complex systems is often a hard problem. This is obviously what Albert Einstein meant when he said:

> 'Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius – and a lot of courage – to move in the opposite direction.'

1.1.2 Determination of Model Parameters

Once the underlying model has been specified the determination of the corresponding model parameters is required, which can be an equally hard problem. Not seldom these parameters are found by means of solving a structured matrix nearness problem.

The reason of using matrix nearness problems is that many problems in science can easily be represented by minimizing a distance between two matrices. Moreover, there exists a well established theory about matrices and their properties and highly efficient algorithms are available to operate on them. Hence, methods can be developed that determine the corresponding model parameters efficiently. Furthermore, matrices are flexibly usable and allow to represent large data sets in an intuitive manner, which is another reason why applications of matrix nearness problems are more than wide-ranging and are present in all areas of science as we will see later in this thesis.

The matrices that appear in these nearness problems often have an additional structure that comes from the application and needs to be considered when developing algorithms that solve these problems. The aim is to exploit the structure of the underlying system to make these algorithms highly efficient in terms of performance and storage usage. Structure refers thereby to an additional property that the matrices need to satisfy. For example matrices may be required to be symmetric or orthogonal or more general to lie in an additional constraining convex set, depending on the application.

1.1.3 Structured Matrix Nearness Problems

We generally define a structured matrix nearness problem as follows. Let $A \in \mathbb{R}^{n\times p}$ be a given matrix and $\mathcal{X} \subset \mathbb{R}^{n\times p}$ be the set that describes the structure of the matrix of interest. Then the problem is to find a matrix $X_* \in \mathcal{X}$ that is an optimal solution of

$$\min_{X\in\mathcal{X}} \|A - X\|, \tag{1.1}$$

where $\|\cdot\|$ can be any induced norm in $\mathbb{R}^{n\times p}$. In most of our problems $\|\cdot\|$ will be the Frobenius norm

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}.$$

However, we will also look at some weighted norms that are introduced in Section 1.3 and popular to take in many practical applications.

1.1.4 Examples Where Matrix Nearness Problems Occur

Structured matrix nearness problems arise for instance when one is looking for an optimal orthogonalization of a given matrix. This problem appears for example in aerospace computations where a direction cosine matrix needs to be determined, describing the rotation of a coordinate system relative to a reference system. This matrix is computed by solving an ordinary differential equation but often due to the solver used the computed matrix is not orthogonal. Therefore one is required to solve

a structured nearness matrix problem to obtain a matrix that is orthogonal and closest to the computed matrix [66, Section 2.6], [94]. Other structured matrix nearness problems arise for instance in biometrics identification where one needs to compare two data sets. This comparison is carried out by posing a Procrustes problem [58, Section 14.1.5], which is a structured matrix nearness problem. Similar problems appear also in molecular biology [34] or in image processing for point-to-point registrations [22], [85], [11]. In finance a large number of models has been proposed to analyze the financial market and to estimate the risk of financial instruments. In Chapter 2 we will introduce a model that is used to investigate asset returns [35, Section 3.5], collateralized debt obligations (CDOs) [8], and multivariate time series [88]. We will see that the determination of the model parameters is equivalent to solving a structured matrix nearness problem where in this case the matrix of interest enjoys a k factor structure. Another application is introduced in Chapter 4 that arises in atomic chemistry and requires to solve again a matrix nearness problem. In Section 1.7 we will mention more applications of structured matrix nearness problems, in particular those where the corresponding matrix is related to the sets of matrix structures that we will introduce throughout this chapter.

1.2 Goal of Thesis

In this thesis we look at different structured matrix nearness problems that all come from applications in different areas of science. Our goal is to investigate their matrix structure and propose algorithms that solve the corresponding nearness problems efficiently by exploiting the structure.

In the first part of this thesis we will consider nearness problems that lead to optimization problems over closed convex sets whereas in the second part we will move to problems where the optimization is performed over Riemannian manifolds.

Introductorily to the topic of this thesis we start in this chapter with considering the set $\mathcal{X}$ in (1.1) to be the intersection of a finite number of closed convex sets. In this case there exists a well established theory that deals with the corresponding nearness problems. Particularly we will look at two closed convex sets and their corresponding nearness problems due to their frequent occurrence in science but also because of their importance in the subsequent chapters: the set of linearly structured matrices and the set of positive semidefinite matrices. As we consider their corresponding nearness problems with respect to weighted norms that are popular to take we begin with introducing the Q-weighted norm and two important variants in Section 1.3. We will then explain in Section 1.4.1 why (1.1) is a well posed problem if $\mathcal{X}$ is an intersection of closed convex sets. In addition, we will introduce a popular method

in the subsequent Section 1.4.2 that guarantees convergence to an optimal solution of the corresponding nearness problem and is easy to implement: the alternating projections method. Thereafter we focus on the two sets that we mentioned above. We will look at the set of linearly structured matrices and discuss the corresponding nearness problem in Section 1.5. Subsequently we concentrate on the set of positive semidefinite matrices and its nearness problem in Section 1.6. Eventually, some applications of these problems are introduced in Section 1.7. In the subsequent Section 1.8 we will then explain more in detail which particular matrix structures and nearness problems we investigate in this thesis and where the applications come from. We will also mention at the end of this chapter in Section 1.9 which parts of the thesis are new contributions to science and what has been achieved throughout this thesis.

1.3 The Q-Weighted Norm

Before introducing the Q-weighted norm in the space $\mathbb{R}^{n\times p}$ with $n \geq p$ let us first define the operator vec : $\mathbb{R}^{n\times p} \mapsto \mathbb{R}^{np}$ that stacks the columns of a matrix into a long column vector. Then the Q-norm is defined for $Q \in \mathbb{R}^{np\times np}$ symmetric positive definite as

$$\|A\|_Q := \sqrt{\mathrm{vec}(A)^T Q \mathrm{vec}(A)}$$

for $A \in \mathbb{R}^{n\times p}$ and is induced by the inner product

$$\langle A, B\rangle_Q := \mathrm{vec}(A)^T Q \mathrm{vec}(B)$$

for $A, B \in \mathbb{R}^{n\times p}$. Note that for n and p large computing the Q-norm can become computationally expensive, in particular solving the corresponding nearness problem. Therefore we also consider two special choices of the Q-norm.

The first is the H-weighted Frobenius norm, which is defined for an arbitrary matrix $A \in \mathbb{R}^{n\times p}$ as

$$\|A\|_H := \|H \circ A\|_F \tag{1.2}$$

where $\circ$ denotes the Hadamard product: $A \circ B = (a_{ij}b_{ij})$ and H is a matrix in $\mathbb{R}^{n\times p}$ with $h_{ij} \neq 0$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$. The latter property ensures that (1.2) fulfills the conditions of a norm and is thus well defined. The H-norm is induced from the inner product

$$\langle A, B\rangle_H := \mathrm{trace}((H \circ B)^T (H \circ A)) \tag{1.3}$$

for $A, B \in \mathbb{R}^{n\times p}$.

The second weighted norm can only be defined for $n = p$ and is

$$\|A\|_W := \|W^{1/2} A W^{1/2}\|_F, \tag{1.4}$$

where $W \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix and $W^{1/2}$ denotes the square root of W defined as

$$W^{1/2} = P^T \operatorname{diag}(\lambda_i^{1/2}) P.$$

Here, $P^T \Lambda P$ is the spectral decomposition of W and $\Lambda = \operatorname{diag}(\lambda_i)$ is the matrix of the eigenvalues of W where $\operatorname{diag} : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$ is an operator with $\operatorname{diag}(a)$ the diagonal matrix with the elements of $a \in \mathbb{R}^n$ on its diagonal. We denote the norm in (1.4) by the W-norm. Let us now prove that both the H- and the W-norm are special cases of the Q-norm.

Lemma 1.3.1. *The H- and the W-weighted norms are special cases of the Q-norm.*

Proof. We first prove the claim for the H-weighted norm. Let $H \in \mathbb{R}^{n\times p}$ with $h_{ij} \neq 0$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, p$. Then we define Q as $Q := \operatorname{diag}(\operatorname{vec}(H \circ H))$. Since $h_{ij} \neq 0$, Q is symmetric positive definite and thus, well defined. The claim follows then for $A \in \mathbb{R}^{n\times p}$ from

$$\|A\|_H^2 = \|H \circ A\|_F^2 = \operatorname{vec}(H \circ A)^T \operatorname{vec}(H \circ A) = \operatorname{vec}(A)^T Q \operatorname{vec}(A) = \|A\|_Q^2.$$

Let us now look at the W-weighted norm. Let $W \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Then by using the properties of the Kronecker product, see Appendix A.1, it follows for $Q := W \otimes W$ and $A \in \mathbb{R}^{n\times n}$ that

$$\begin{aligned}\|A\|_W^2 &= \|W^{1/2} A W^{1/2}\|_F^2 = \operatorname{trace}(W A^T W A) = \operatorname{vec}(A)^T \operatorname{vec}(W A W) \\ &= \operatorname{vec}(A)^T (W \otimes W) \operatorname{vec}(A) = \|A\|_Q^2,\end{aligned}$$

which implies that Q is positive definite and thus, verifies the claim. ∎

The H- and W-norm are popular to take as a distance measurement between two matrices A, B. One reason is certainly that the H-norm weights the distance between A, B element-wise so that the effect of the weighting is intuitive and clear and thus, easily adjustable to the needs in applications. The advantage of the W-norm is that the weighting preserves the inertia and symmetry of the matrices A and B, which, as we will see later in Section 1.6.2, makes this weighted norm attractive to use.

1.4 Structured Matrix Nearness Problems for $\mathcal{X}$ Closed Convex

1.4.1 The Problem and its Properties

Now we look at the nearness problem in (1.1) when $\mathcal{X}$ is an intersection of closed convex sets in the Q-norm. Let $A \in \mathbb{R}^{n\times p}$ be a given matrix and $\mathcal{C}_1, \ldots, \mathcal{C}_m$ be m closed convex sets. Then we are trying to solve

$$\min_{X\in\mathcal{C}_1\cap\cdots\cap\mathcal{C}_m} \frac{1}{2}\|A - X\|_Q^2. \tag{1.5}$$

Let $\mathcal{X}$ be defined as $\mathcal{X} := \mathcal{C}_1\cap\cdots\cap\mathcal{C}_m$. Since $\mathcal{C}_1, \ldots, \mathcal{C}_m$ are closed and convex $\mathcal{X}$ is also closed and convex. If $\mathcal{X}$ is also nonempty an optimal solution of the corresponding nearness problem (1.5) exists and is also unique [68, Chapter 3]. This allows us to define an operator $\Pi_{\mathcal{X}} : \mathbb{R}^{n\times p} \mapsto \mathcal{X}$ that maps a matrix $A \in \mathbb{R}^{n\times p}$ onto the optimal solution of (1.5). This operator is called the *projection* of A onto $\mathcal{X}$ in the Q-norm.

In many applications the projection onto $\mathcal{X}$ is not known or available but the projection onto the individual sets $\mathcal{C}_1, \ldots, \mathcal{C}_m$ is readily applicable, see for example the nearest correlation matrix problem in Section 1.7. In this case the alternating projections method can be applied. We introduce this method in the next section as it is a popular method that is easy to implement, flexibly usable and thus, widely applicable, and most importantly it guarantees to converge to the optimal solution.

Looking at our two particular sets, we will see that we can project onto the set of linearly structured matrices and also onto the set of positive definite matrices in the W-norm separately but to compute the projection onto the intersection of both sets is a hard problem. In this case the alternating projection method provides us with tool to find the intersection point that minimizes the objective function in (1.5) in the W-norm.

1.4.2 Alternating Projections Method

The idea of the alternating projections method is to project the input matrix A in (1.5) iteratively onto the closed convex sets $\mathcal{C}_1, \ldots, \mathcal{C}_m$ that is repeating the operation

$$A \longleftarrow \Pi_{\mathcal{C}_1} \cdots \Pi_{\mathcal{C}_m}(A).$$

For each convex set, unless it is an affine subspace, it is necessary to incorporate a judiciously chosen correction of Boyle and Dykstra [25] to every projection to obtain convergence to optimal points. Then the method converges at best at linear rate. Note that the correction can be interpreted as a normal vector to the corresponding

convex set [65]. See Deutsch [42] for a survey regarding the alternating projections methods.

The convex sets $\mathcal{C}_1, \ldots, \mathcal{C}_m$ incorporate the desired properties of the final outcome matrix. Hence, modifications of the convex sets allow to obtain the solution of different nearness problems. The assumption is only that the convex sets have a point in common and the projections onto the individual convex sets under the norm considered exist. Therefore the alternating projections method is a powerful tool to solve difficult problems by transforming them into 'less' difficult problems.

Let us now look at a specific closed convex set that is the set of linearly structured matrices.

1.5 Set of Linearly Structured Matrices $\mathcal{L}$

1.5.1 Definition of $\mathcal{L}$

Let $U_1, \ldots, U_s \in \{0,1\}^{n \times p}$ be s given matrices. Then we define the set of *linearly structured matrices* as

$$\mathcal{L}(U_1, \ldots, U_s) := \left\{ X : X = \sum_{i=1}^{s} x_i U_i, \text{ and } x_i \in \mathbb{R} \text{ for all } i = 1, \ldots, s \right\}. \tag{1.6}$$

Note that for simplicity we will write only $\mathcal{L}$ if it is clear to which matrices $U_1, \ldots, U_s$ we refer to. Before we introduce some examples of $\mathcal{L}$ for different matrices $U_1, \ldots, U_s$ in Section 1.7 let us illustrate how this set can look by a member of $\mathcal{L}$ when $n = p = 3$

$$\begin{bmatrix} x_1 + x_2 & x_1 & 0 \\ x_2 & x_3 & x_1 \\ 0 & x_2 & x_1 + x_3 \end{bmatrix}.$$

The set $\mathcal{L}$ is a subspace of $\mathbb{R}^{n \times p}$, consisting of all linear combinations of the matrices $U_1, \ldots, U_s$, which implies that it is closed and convex.

Let U be defined as

$$U := [\text{vec}(U_1) \ \cdots \ \text{vec}(U_s)]. \tag{1.7}$$

Then $X \in \mathcal{L}$ can be written as $\text{vec}(X) = Ux$ with $x = (x_1, \ldots, x_s)^T$. We will use this notation later. Note that if $(U_1, \ldots, U_s)$ is a basis in $\mathbb{R}^{n \times p}$ then $\mathcal{L}$ describes the entire space $\mathbb{R}^{n \times p}$.

As we have seen in Section 1.4 solving the nearness problem for closed convex sets is equivalent to projecting onto the space. Since $\mathcal{L}$ is closed and convex the projection onto $\mathcal{L}$ is well defined.

1.5.2 Projection onto $\mathcal{L}$

We consider now how to compute $\Pi_{\mathcal{L}}$. Thus, we need to solve for $A \in \mathbb{R}^{n\times p}$

$$\min_{x\in\mathbb{R}^s} \quad f(x) := \tfrac{1}{2}\|A - \textstyle\sum_{i=1}^s x_i U_i\|_Q^2, \tag{1.8}$$

where $\Pi_{\mathcal{L}}(A)$ is then $\sum_{i=1}^s x_i^* U_i$ for $x_* = (x_1^*, \ldots, x_s^*)^T$ an optimal solution of (1.8). Note that the objective function $f(x)$ of (1.8) is convex but may not be strictly convex therefore the optimal solution of (1.8) might not be unique, depending on $U_1, \ldots, U_s$ as we will see below.

Due to the convexity of f it is sufficient to determine the zeros of the derivative of f to find the optimal solution. Differentiating f and setting it to zero leads to a linear system $Fx = b$ with

$$F = \begin{bmatrix} \langle U_1, U_1\rangle_Q & \ldots & \langle U_1, U_s\rangle_Q \\ \vdots & \ddots & \vdots \\ \langle U_s, U_1\rangle_Q & \ldots & \langle U_s, U_s\rangle_Q \end{bmatrix}, \; b = \begin{bmatrix} \langle U_1, A\rangle_Q \\ \vdots \\ \langle U_s, A\rangle_Q \end{bmatrix}. \tag{1.9}$$

The matrix F has the well known form of a Gram matrix so it is always positive semidefinite. Moreover, if the matrices $U_1, \ldots, U_s$ are chosen to be linearly independent the matrix is nonsingular, implying a unique solution of (1.8).

If the chosen inner product is the inner product defined in (1.3) inducing the H-weighted Frobenius norm the problem (1.8) is equivalent to the linear least squares problem

$$\min_{x\in\mathbb{R}^s} \|D(Cx - d)\|_2^2 \tag{1.10}$$

with $C = U \in \mathbb{R}^{np\times s}$ and U as defined in (1.7) with the rank of U less than or equal to $\min\{np, s\}$, $D = \operatorname{diag}(\operatorname{vec}(H))$, and $d = \operatorname{vec}(A) \in \mathbb{R}^{np}$. It is well known that (1.10) has always a unique minimal 2-norm solution

$$x = (DC)^+ Dd \tag{1.11}$$

[19, Theorem 1.2.10] with $(DC)^+$ the Moore-Penrose pseudo-inverse of DC. If the weights vary widely in size computing the solution (1.11) directly can become numerically instable. An example where these difficulties occur is given in [19]. To overcome this problem, the author in [19] proposes to transform the matrix DC first into a block triangular form by means of Gaussian elimination, resulting in a well-conditioned linear least squares problem.

The problem (1.8) can also be reformulated as a linear least squares problem when the W-weighted inner product is considered. In this case $p = n$ and D can be chosen as the identity matrix, C as the matrix

$$C = [\operatorname{vec}(W^{1/2}U_1W^{1/2}), \ldots, \operatorname{vec}(W^{1/2}U_sW^{1/2})] \in \mathbb{R}^{n^2\times s}$$

and $d = \text{vec}(W^{1/2}AW^{1/2}) \in \mathbb{R}^{n^2}$ in (1.10).

Now, we consider the set of symmetric positive semidefinite matrices.

1.6 Set of Positive Semidefinite Matrices $\mathcal{S}_n^+$

1.6.1 Definition of $\mathcal{S}_n^+$

Let $\mathcal{S}^n$ be the set of n-by-n symmetric matrices so that for all $A \in \mathcal{S}^n$ it holds that $A^T = A$. Then a matrix $A \in \mathcal{S}^n$ is called *positive semidefinite* if for all $x \in \mathbb{R}^n$: $x^T Ax \geq 0$. We denote the set of all *symmetric positive semidefinite matrices* by $\mathcal{S}_n^+$. This set is again closed and convex so that we can define the projection of square matrices onto this set.

1.6.2 Projection onto $\mathcal{S}_n^+$

Let us first consider the projection $\Pi_{\mathcal{S}_n^+}$ with respect to the W-norm.

Projection w.r.t. W-norm

A key property, which makes the W-norm popular to use and allows e.g. the fast computation of the nearest correlation matrix in this norm, see Section 1.7, is the existence of an explicit form for the projection $\Pi^W_{\mathcal{S}_n^+}$ under the W-norm. Let $(\cdot)_+ : \mathcal{S}^n \mapsto \mathcal{S}_n^+$ be the operator with $(S)_+ = P^T \text{diag}(\max\{\lambda_i, 0\})P$ where $P^T \Lambda P = S$ is the spectral decomposition of S. The projection of A onto $\mathcal{S}_n^+$ is then the solution of

$$\min_{X \in \mathcal{S}_n^+} \frac{1}{2}\|A - X\|_W^2 \tag{1.12}$$

for $A \in \mathbb{R}^{n\times n}$ and given by

$$\Pi^W_{\mathcal{S}_n^+}(A) := W^{-\frac{1}{2}}(W^{\frac{1}{2}}\text{sym}(A)W^{\frac{1}{2}})_+ W^{-\frac{1}{2}} \tag{1.13}$$

[65, Theorem 3.2], [21, Theorem 4.9.1], where $\text{sym}(A) = (A + A^T)/2$ is the symmetric part of A.

Projection w.r.t. H-norm

Unfortunately, such a closed form solution does generally not exist for the H-norm and thus also not for the Q-norm. In [75, Corollary 2.2] the authors show that for square, symmetric and positive semidefinite H a sufficient condition for a matrix $X \geq 0$ to be the projection $X = \Pi^H_{\mathcal{S}_n^+}(A)$ of $A \in \mathbb{R}^{n\times n}$ onto the set of positive semidefinite matrices is

$$H \circ X = (\text{sym}(A \circ H))_+. \tag{1.14}$$

Note that the actual statement of the corollary in [75] is misleading since it claims that (1.14) is a necessary condition for X to be the projection. However, this does not hold in general as shown in [112] and illustrated by the following counter example. Let

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \text{ and } H = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix}.$$

Then from (1.14) $X = (A \circ H)_+ \circ \widehat{H} = A$ and thus is not positive semidefinite where $\widehat{H}$ is the Hadamard inverse with $\widehat{h}_{ij} = 1/h_{ij}$ for $i, j = 1, \ldots, n$. Hence the matrix X that satisfies (1.14) is not the projection $\mathit{\Pi}^H_{\mathcal{S}_n^+}(A)$. The sufficiency result of [75, Corollary 2.2] is easily derived from (1.12) and (1.13). Let X_* be the projection in the H-norm. As X_* and the matrix H are positive semidefinite the product $H \circ X_*$ is also positive semidefinite [74, Section 3.1]. Hence, from (1.13) we have that $\|H \circ A - (\mathrm{sym}(H \circ A))_+\|_F \leq \|H \circ A - H \circ X_*\|_F = \|A - X_*\|_H$. As from (1.14) there exists a positive semidefinite X with $H \circ X = (\mathrm{sym}(H \circ A))_+$ it follows that X is the projection onto the set of positive semidefinite matrices in the H-norm.

If H is symmetric positive semidefinite with positive entries and of rank 1 then according to [117, Theorem 2.7] the Hadamard inverse $\widehat{H}$ is also positive semidefinite. Hence, the matrix computed as

$$X = (\mathrm{sym}(A \circ H))_+ \circ \widehat{H} \tag{1.15}$$

is positive semidefinite and therefore satisfies the sufficient condition to be the projection in the H-norm. This gives a condition on H so that the projection can be directly computed via (1.15). As in this case, the matrix H can be written as $h_{ij} = \sqrt{w_i w_j}$ for $i, j = 1, \ldots, n$ and w a vector with positive entries this result can also be derived by observing that the H- and W-norm coincide with $W = \mathrm{diag}(w)$.

Projection w.r.t. Q-norm

Note, to compute the projection onto $\mathcal{S}_n^+$ in the Q-norm one could apply the spectral projected gradient method [18] that we will discuss more in detail in Section 2.5. In general this method minimizes a smooth function f over a closed convex set $\mathcal{C}$ by using the projection onto this convex set with respect to the defined inner product in the space. Therefore in order to determine the projection onto $\mathcal{S}_n^+$ in the Q-norm one could apply the spectral projected gradient method by setting f to $f(X) := \frac{1}{2}\|A - X\|_Q^2$ and by using the projection in (1.13) to project onto the set $\mathcal{S}_n^+$.

1.7 Applications

Let us now introduce some applications of the nearness problems that we discussed in Section 1.5 and 1.6. As the set of linearly structured matrices can describe matrix sets with various different structures and patterns, depending on the chosen matrices $U_1, \ldots, U_s$, these problems are of interest in a large number of applications in many areas of science. Therefore we will only look at a selection of these applications.

1.7.1 Correlation Matrices

One example is the problem of the nearest correlation matrix under weighted norms, which has recently been studied [65], [110], [61]. A correlation matrix is a symmetric matrix that is positive semidefinite and has unit diagonal. In 2002 Higham [65] proposed to use the alternating projections method that we introduced in Section 1.4.2 to find the nearest correlation matrix in the W-norm. The idea is to project alternately onto the set of positive semidefinite matrices and the set of symmetric matrices having unit diagonal. This algorithm converges at best linearly. More recently faster algorithms were investigated. In [110] Qi and Sun introduced a semismooth quadratically convergent Newton algorithm which computes the nearest correlation matrix in the W-norm. Later they also dealt with the corresponding problem under the H-norm and proposed an augmented Lagrangian approach [112].

Correlation matrices often occur also in patterns that can be described by the set $\mathcal{L}$. In Section 2.2 we will come across one example, in which the correlation matrix depends only on one parameter. In this case the corresponding nearness problem enjoys many applications in finance. Other linear structures that arise in connection with correlation matrices are mentioned in [74, pp. 240-242]. In these examples the matrices are often divided into blocks where the elements of one block have the same value. Therefore these matrices are of linear structure. The corresponding nearness problem (1.5) is then of interest in combination with the alternating projections method when one needs to determine the nearest correlation matrix that is of one of these patterns.

1.7.2 Toeplitz and Hankel Matrices

Let $n = p$, $s = 2n-1$, and in MATLAB notation U_i `= diag(ones(i,1), n-i)`, and U_{n+i} `= diag(ones(n-i,1),-i)` for $i = 1, \ldots, n-1$, and U_n `= eye(n)`. Then the projection onto $\mathcal{L}$ spanned by $U_1, \ldots, U_s$ is the projection onto the set of Toeplitz matrices. Equivalently, the set of Hankel matrices can be described by flipping $U_1, \ldots, U_{2n-1}$ from left to right. In MATLAB notation this is U_n `= fliplr(eye(n))`,

U_i `= fliplr( diag(ones(i,1),n-i))`, and U_{n+i} `= fliplr( diag( ones(n-i,1), -i))` for $i = 1, \ldots, n-1$. Accordingly, band and block Toeplitz matrices can also be described by the set $\mathcal{L}$. These block Toeplitz matrices occur, e.g., in [59] where a homogeneous random field on a 2D domain with covariance function $r(x, y)$ needs to be sampled on a uniform rectilinear grid with equidistant grid spacing. Then the relevant covariance matrix R has block Toeplitz structure. Realisations with this desired covariance structure are then used to compute the expectation of nonlinear functionals of random fields using a quasi-Monte Carlo method. Such realisations are quickly computed by the FFT techniques where this covariance matrix needs first to be embedded in a larger circulant matrix to apply FFT. Suitable padding values are then determined to ensure the positive semidefiniteness of this matrix [44]. Instead of using specified padding values one could also determine the nearest positive semidefinite circulant matrix having R as the leading part. This gives the motivation for the matrix structure in the next subsection.

Toeplitz and Hankel matrices also play an important role in signal processing where the underlying matrix of the system is of that structure. The low rank approximation of such systems, which we will consider in Chapter 5, often corresponds to noise removal of incoming signals or model reduction.

Note that these matrix structures also appear in block form and similar algorithms for the low rank approximation have been developed in [96], [122]. One example is the block-row Hankel matrix structure arising in multiple-input multiple-output system identification problems [122].

1.7.3 Circulant Matrices

A special form of Toeplitz matrices are the so called circulant matrices. These matrices have the structure

$$\begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_3 & x_2 \\ x_2 & x_1 & x_2 & \cdots & x_4 & x_3 \\ x_3 & x_2 & x_1 & \ldots & x_5 & x_4 \\ \vdots & & & \ddots & & \vdots \\ x_3 & x_4 & x_5 & \cdots & x_1 & x_2 \\ x_2 & x_3 & x_4 & \cdots & x_2 & x_1 \end{bmatrix}$$

and can thus, be described by the set $\mathcal{L}$. We direct the interested reader to [36] to find out more about the properties and applications of such matrices.

1.7.4 Sylvester Matrices

Let p, q be two polynomials of degree n and m, respectively, with

$$p(x) = p_n x^n + \cdots + p_0$$

and

$$q(x) = q_m x^m + \cdots + q_0,$$

where p_n, q_m are nonzero. Then a Sylvester matrix associated with the two polynomials p, q is a matrix $S \in \mathbb{R}^{(m+n)\times(n+m)}$ with

$$S = \begin{bmatrix} p_n & p_{n-1} & \dots & p_0 & 0 & \dots & 0 & 0 \\ 0 & p_n & \dots & p_1 & p_0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \dots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & p_n & \dots & p_1 & p_0 & 0 \\ 0 & 0 & \dots & 0 & p_n & \dots & p_1 & p_0 \\ q_m & q_{m-1} & \dots & q_0 & 0 & \dots & 0 & 0 \\ 0 & q_m & \dots & q_1 & q_0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \dots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & q_m & \dots & q_1 & q_0 & 0 \\ 0 & 0 & \dots & 0 & q_m & \dots & q_1 & q_0 \end{bmatrix}.$$

This matrix is of interest as the coefficients of the greatest common divisor of p and q can be recovered from the Sylvester matrix [81, Theorem 3] within a constant factor by performing only row operations to triangularize it to row echelon form. Determining the greatest common divisor is a fundamental problem of computing theory and is needed e.g. in computer algebra systems for simplifying rational functions [52, Chapter 7]. It also arises in polynomial factorization or symbolic integration and in the area of error-control coding and quantization error-free computation [143].

Another relation between the matrix S and the greatest common divisor of the polynomials p and q is that the rank deficiency of S is equivalent to the degree of the greatest common divisor [52, Theorem 7.3], [78, Theorem 2.3]. This property is used if one is looking for two polynomials $\widehat{p}$ and $\widehat{q}$ that are closest to p and q, respectively and have a greatest common divisor with a degree greater or equal than a predefined number $k \leq \min\{m, n\}$. This is of interest if the coefficients of p, q are inexact and the aim is to find the two closest polynomials that have a nontrivial greatest common divisor. The problem can then be reformulated as a nearest low rank problem where the output matrix is of Sylvester structure [78]. Many efficient algorithms have been proposed to compute the greatest common factor by using the Sylvester matrix [52, Chapter 7], [143], [15].

1.8 Outline of Thesis

In this chapter we have introduced two closed convex sets: the set of linearly structured matrices and the set of positive semidefinite matrices and discussed projections onto these sets under different weighted norms. In particular we looked at the H- and W-norm that coincide with special choices for the matrix Q in the Q-norm. We have seen that computing the projection onto the set of linearly structured matrices corresponds to solving a linear system in all norms discussed. To determine the projection onto the set of semidefinite matrices one can use an explicit formula in the W-norm whereas in the H-norm or more general, in the Q-norm one requires to apply optimization routines.

We also looked at the alternating projections method as it allows to compute the intersection point of these sets that is nearest to a given matrix if the projections onto the individual sets can be computed. Eventually we introduced some important applications where these nearness problems arise.

In the next chapters we will look at different structured matrix nearness problems that come from real applications. In Chapter 2 we start with investigating a particular structure of a correlation matrix called k factor structure and we look at the corresponding nearness problem in the Frobenius norm. We will see that this problem mainly arises in the area of finance when one is modelling assert returns or multivariate time series. We will investigate several numerical methods for solving this problem and compare their performance numerically. Note that most of the material of Chapter 2 has already been published in [24].

As solving the structured nearness problems that we look at in the subsequent chapters requires to optimize an objective function over a Riemannian manifold we give for completeness an introduction to this topic in Chapter 3 where all geometric objects that are needed for an optimization are presented. We also discuss two algorithms that solve optimization problems over Riemannian manifolds: the nonlinear CG method and the RBFGS method. For the latter method we will look at a limited memory version.

In Chapter 4 we investigate two structured nearness problems that arise in atomic chemistry. We specify an analytical solution of the first problem and reformulate the second as a convex quadratic programming problem with inequality constraints. To solve the latter problem we apply the active-set method. As the set of optimal solutions of the first problem is generally not unique we consider to pose a new optimization problem over this set to find a particular solution out of it. By dropping a few constraints we show that the remaining constraining set of the optimization problem is a Riemannian manifold and develop all geometric objects to be able to

optimize over this set. By using the optimization tools for Riemannian manifolds we then apply the augmented Lagrangian method to consider the disregarded constraints. The result is a new algorithm that minimizes an arbitrary smooth function over the set of optimal solutions of the first problem. At the end of this chapter we investigate the performance of this algorithm numerically. Note that the material of this chapter will be part of [20], which is currently in preparation.

The subsequent chapter concerns the problem of finding the low rank matrix that is of a predefined linear structure and nearest to a given matrix. Our main interest lies thereby in algorithms that are applicable to any linear structure and to any symmetric positive definite weighting matrix Q in the Q-norm. We will see that we can reformulate the problem as an optimization problem over the Grassmannian manifold that allows to apply the augmented Lagrangian method.

Finally in the last chapter we draw the conclusion of our achievements throughout this thesis and mention some future work. We subsequently list all main symbols that we introduce throughout all chapters to give a better overview to the reader. For completeness we define the Fréchet derivative and the Kronecker product in the appendix and mention some properties of the latter as we make frequent use of it.

1.9 Main Research Contributions

To provide an easily accessible way for the reader to what is achieved throughout this thesis we are finishing this chapter by listing all our main contributions to research.

- Chapter 2: When considering matrices with k factor structure $C(X) = XX^T - \operatorname{diag}(XX^T - I_n)$ for $X \in \mathbb{R}^{n\times k}$ and $k \leq n$ we first discuss constraints on X that give necessary and sufficient conditions for $C(X)$ to be a correlation matrix. Then we look at a special one parameter case and derive an explicit solution of the nearness problem. Thereafter we consider the one factor case $k = 1$ and obtain a rank result for matrices with this structure. For the general nearness problem, which is in contrast to the one parameter problem highly nonconvex, we derive the gradient and the Hessian of the objective function so that first and second order iterative algorithms can be applied. Also, when $k = 1$ an instructive result on the positive definiteness of the Hessian is given.

  We investigate several numerical methods for solving the nearness problem: the alternating directions method; a principal factor method used by [8] which we show is equivalent to the alternating projection method, projecting onto a convex set and a nonconvex set in turn and hence lacks convergence results; the spectral projected gradient method (SPGM) of Birgin, Martínez, and Raydan; and Newton

and sequential quadratic programming methods. The methods differ in whether or not they can take account of the nonlinear constraints and in their convergence properties. Since all methods are iterative methods we look at the effect of different starting matrices, including a new rank one starting matrix, on the performance. Our numerical experiments show that the performance of the methods depends strongly on the problem, but that SPGM is the clear winner. In addition, we demonstrate empirically for this method how the performance and the optimal objective function value vary when k is increased.

- Chapter 3: The general purpose of this chapter is mainly to provide an introduction to the optimization over Riemannian manifolds. We therefore consider as contribution in this chapter only the discussion of how the algorithm of [101] that efficiently computes the approximation of the Hessian times a vector in the limited memory BFGS method can be generalized to Riemannian manifolds. We propose the corresponding algorithm in Algorithm 3.9.2. However, we do not look at convergence results for this method.

- Chapter 4: In this chapter our focus lies at two two-sided optimization problems where we give an analytical optimal solution of the first problem. We show that the second problem is equivalent to a convex quadratic programming problem with box constraints that can be solved by the active-set method in at most $2p$ iterations where p is the number of unknowns. We then concentrate on the set of optimal solutions of the first problem, which is the set of matrices $Y \in \mathbb{R}^{n\times p}$ and $p \leq n$ that have orthonormal columns and satisfy $Y^T NY = D$ for a given symmetric matrix $N \in \mathbb{R}^{n\times n}$ and a given diagonal matrix $D \in \mathbb{R}^{p\times p}$ with increasing diagonal elements. We show that a slight modification of this set is a Riemannian manifold and prove that for different input diagonal matrices D the corresponding analytical optimal solutions that we have derived for the first problem are connected on this manifold. We derive the tangent and normal space and propose a retraction for this manifold. In particular, we consider how to efficiently compute the projection onto the normal space, which corresponds to solving a linear system of order p^2. We show that it is enough to solve a linear system of order $(p-1)p/2$ and prove that the coefficient matrix of this system is sparse for p large. Numerical tests have shown that this matrix is also better conditioned than the coefficient matrix in the larger system.

  As mentioned in Section 1.8 we then apply the augmented Lagrangian method to incorporate the p dropped constraints, resulting in a new algorithm that minimizes an arbitrary smooth function over the set of optimal solution of the first problem

by using geometric optimization tools. We compare the performance of this algorithm numerically with an augmented Lagrangian method that incorporates all the constraints $Y^T NY = D$ by optimizing the corresponding augmented Lagrangian function. By means of numerical tests we demonstrate that the new algorithm outperforms the latter on all our test problems.

- Chapter 5: In this chapter we look at the problem of finding the nearest low rank matrix to a given matrix in the Q-weighted norm where this low rank matrix is required to be of a predefined linear structure. We analyze three existing algorithms in the literature that deal with this problem and look at the geometric approach proposed in [121] more in detail, which involves minimizing an objective function over the Grassmannian manifold. The authors in [121] propose an algorithms to find a solution of the problem. However, they noticed that their algorithm can break down. We discuss the reasons for these breakdowns and propose then to apply the augmented Lagrangian method to the optimization problem that tackles these breakdowns, resulting in an algorithm that is applicable for any linear structure. Unfortunately, we cannot guarantee convergence for this algorithm in general. In order to apply the existing convergence theory for augmented Lagrangian method one requirement is that the linear constraint qualification (LICQ) is satisfied at the optimal solution. We demonstrate by means of two examples that the LICQ can but also cannot be satisfied, depending on the problem. Subsequently, we investigate the performance of our algorithm by means of numerical tests and compare it with other existing algorithms.

Let us now look at our first main matrix structure and the corresponding nearness problems.

Chapter 2

Nearness Problems of Correlation Matrices with Factor Structure

2.1 Introduction

In many practical applications involving statistical modeling it is required to adjust an approximate, empirically obtained correlation matrix so that it has the three defining properties of a correlation matrix: symmetry, positive semidefiniteness, and unit diagonal. Lack of definiteness can result from missing or asynchronous data which, in the case of financial modeling, may be due to a company being formed or ceasing to trade during the period of interest or markets in different regions trading at different times and having different holidays. Furthermore, stress testing may require individual correlations to be artificially adjusted, with subsequent value-at-risk analysis breaking down if the perturbed matrix is not a correlation matrix [48], [111]. In a variety of applications it is natural to replace the given empirical matrix by the nearest correlation matrix in the (weighted) Frobenius norm [65], [113], [130], [140]. This problem has received much attention in the last few years and can be solved using the alternating projections method [65] or a preconditioned Newton method [23], [110], the latter having quadratic convergence and being the method of choice. Recently, a more general problem with additional inequality constraints has been considered in [86] and a projected semismooth Newton method was proposed, which has quadratic convergence.

In this work we are interested in the nearness problem in which factor structure is imposed on the correlation matrix. Such structure arises in factor models of asset returns [35, Section 3.5], collateralized debt obligations (CDOs) [8], [54], [73], and

multivariate time series [88]. To motivate this structure we consider the factor model[1]

$$\xi = X\eta + F\varepsilon \tag{2.1}$$

for the random vector $\xi \in \mathbb{R}^n$, where $X \in \mathbb{R}^{n\times k}$, $F \in \mathbb{R}^{n\times n}$ is diagonal, and $\eta \in \mathbb{R}^k$ and $\varepsilon \in \mathbb{R}^n$ are vectors of independent random variables having zero mean and unit variance, with η and ε independent of each other. In the terminology of factor analysis [98] the components of η are the factors and X is the loading matrix. With $\mathrm{cov}(\cdot)$ and $E(\cdot)$ denoting the covariance matrix and the expectation operator, respectively, it follows that $E(\xi) = 0$ and hence

$$\mathrm{cov}(\xi) = E(\xi\xi^T) = XX^T + F^2. \tag{2.2}$$

If we assume that the variance of ξ_i is 1 for all i then $\mathrm{cov}(\xi)$ is the correlation matrix of ξ and (2.2) gives $\sum_{j=1}^k x_{ij}^2 + f_{ii}^2 = 1$, so that

$$\sum_{j=1}^{k} x_{ij}^2 \le 1, \qquad i = 1\colon n. \tag{2.3}$$

This model produces a correlation matrix of the form

$$C(X) = D + \sum_{j=1}^{k} x_j x_j^T = D + XX^T, \tag{2.4a}$$

$$X = [x_1, \dots, x_k] = \begin{bmatrix} y_1^T \\ \vdots \\ y_n^T \end{bmatrix} = Y^T \in \mathbb{R}^{n\times k}, \tag{2.4b}$$

$$D = \mathrm{diag}(I_n - XX^T) = \mathrm{diag}(1 - y_i^T y_i), \tag{2.4c}$$

where I_n denotes the identity matrix in $\mathbb{R}^{n\times n}$. We say $C(X)$ has k *factor correlation matrix structure*. Note that $C(X)$ can be written in the form

$$C(X) = \begin{bmatrix} 1 & y_1^T y_2 & \dots & y_1^T y_n \\ y_1^T y_2 & 1 & \dots & \vdots \\ & & & \\ \vdots & & \ddots & y_{n-1}^T y_n \\ y_1^T y_n & \dots & y_{n-1}^T y_n & 1 \end{bmatrix},$$

where $y_i \in \mathbb{R}^k$. While $C(X)$ can be indefinite for general X, the constraints (2.3) ensure that XX^T has diagonal elements bounded by 1, which means that $C(X)$ is the sum of two positive semidefinite matrices and hence is positive semidefinite. In

[1] This model is referred to in [54] as the "multifactor copula model."

general, $C(X)$ is of full rank; correlation matrices of low rank, studied in [61], [99], [144], for example, form a very different set. The one factor model ($k = 1$) is widely used [35], [51].

The problem of computing a correlation matrix of k factor structure nearest to a given matrix is posed in the context of credit basket securities by Anderson, Sidenius, and Basu [8], wherein an ad hoc iterative method for its solution is described. The problem is also discussed by Glasserman and Suchintabandid [54, Section 5] and Jäckel [73]. Here, we give theoretical analysis of the problem and show how standard optimization methods can be used to tackle it.

We begin in Section 2.2 by considering a correlation matrix depending on just one parameter, for which an explicit solution to the nearness problem is available. The one factor (n parameter) case is treated in Section 2.3, where results on the representation, determinant, and rank of $C(X)$ are given, along with formulae for the gradient and Hessian of the relevant objective function and a result on the definiteness of the Hessian. In Section 2.4 we consider the general k factor problem and derive explicit formulae for the relevant gradient and Hessian.

Several suitable numerical methods are presented in Section 2.5. We show that the principal components-based method proposed in [8] is an alternating projections method and explain why it cannot be guaranteed to converge. Other methods considered are an alternating directions method, a spectral projected gradient method, and Newton and sequential quadratic programming (SQP) methods. We also derive a rank one starting matrix that yields a smaller function value than $X = 0$. In Section 2.6 we give numerical experiments to compare the performance of the methods and to investigate different starting matrices and the effect of varying k. Conclusions are given in Section 2.7.

Throughout, we will use the Frobenius norm $\|A\|_F = \langle A, A\rangle^{1/2}$ on $\mathbb{R}^{n\times n}$, where the inner product $\langle A, B\rangle = \mathrm{trace}(B^T A)$.

2.2 One Parameter Problem

We begin by considering a one parameter matrix $C(w)$ that has unit diagonal and every off-diagonal element equal to $w \in \mathbb{R}$:

$$C(w) = (1 - w)I_n + wee^T = I_n + w(ee^T - I_n), \tag{2.5}$$

where $e = [1, 1, \ldots, 1]^T$. This matrix is more general than the special case $C(\theta e)$ of the one factor matrix considered in the next section because in that case $w \equiv \theta^2$ is forced to be nonnegative. This structure corresponds to a covariance matrix

with constant diagonal and constant off-diagonal elements—a simple but frequently occurring pattern [5], [67], [76, p. 55], [80], [129], [141].

Lemma 2.2.1. *$C(w) \in \mathbb{R}^{n\times n}$ $(n \geq 2)$ is a correlation matrix if and only if*

$$\frac{-1}{n-1} \leq w \leq 1. \tag{2.6}$$

Proof. $C(w)$ is a correlation matrix precisely when it is positive semidefinite. The eigenvalues of $C(w)$ are $1+(n-1)w$ and $n-1$ copies of $1-w$, so $C(w)$ is positive semidefinite precisely when (2.6) holds. □

We can give an explicit solution to the corresponding nearness problem,

$$\min\{\, \|A - C(w)\|_F : C(w) \text{ is a correlation matrix} \,\}. \tag{2.7}$$

Theorem 2.2.2. *For $A \in \mathbb{R}^{n\times n}$,*

$$\min_w \|A - C(w)\|_F^2 = \|A - I_n\|_F^2 - \frac{(e^TAe - \mathrm{trace}(A))^2}{n^2 - n}$$

and the minimum is attained uniquely at

$$w_{\mathrm{opt}} = \frac{e^TAe - \mathrm{trace}(A)}{n^2 - n}. \tag{2.8}$$

The problem (2.7) *has a unique solution given by the projection of w_{opt} onto the interval $[-1/(n-1), 1]$.*

Proof. We want the global minimizer of

$$\begin{aligned} f(w) &:= \|A - (I_n + w(ee^T - I_n))\|_F^2 \\ &= \|A - I_n\|_F^2 + w^2\|ee^T - I_n\|_F^2 - 2\,\mathrm{trace}((A - I_n)w(ee^T - I_n)) \\ &= \|A - I_n\|_F^2 + w^2(n^2 - n) - 2w\,\mathrm{trace}(Aee^T - A - ee^T + I_n) \\ &= \|A - I_n\|_F^2 + w^2(n^2 - n) - 2w(e^TAe - \mathrm{trace}(A)). \end{aligned}$$

Since $f'(w) = 2w(n^2-n) - 2(e^TAe - \mathrm{trace}(A))$, f has a unique stationary point at w_{opt} given by (2.8). From $f''(w) = 2(n^2-n) > 0$ it follows that f is strictly convex, so w_{opt} is a local and hence global minimizer. The last part follows from the convexity of f. □

It is known [65, Theorem 2.5] that if $a_{ii} \equiv 1$ and A has t nonpositive eigenvalues then the solution to $\min\{\|A - X\|_F : X \text{ is a correlation matrix}\}$ has at least t zero eigenvalues. By contrast, from Theorem 2.2.2 we see that for $a_{ii} \equiv 1$ the solution to problem (2.7) has exactly one zero eigenvalue when $w_{\mathrm{opt}} \leq -1/(n-1)$ (i.e., $e^TAe \leq 0$), and exactly $n-1$ zero eigenvalues when $w_{\mathrm{opt}} \geq 1$ (i.e., $e^TAe \geq n^2$), and otherwise the solution is nonsingular.

A more general version of $C(w)$ arises when variables in an underlying model are grouped and separate intra- and intergroup correlations are defined [60]. The correlation matrix is now a block $m \times m$ matrix $C(\Gamma) = (C_{ij}) \in \mathbb{R}^{n\times n}$, where $\Gamma \in \mathbb{R}^{m\times m}$ and

$$C_{ij} = \begin{cases} C(\gamma_{ii}) \in \mathbb{R}^{n_i\times n_i}, & i = j, \\ \gamma_{ij} ee^T \in \mathbb{R}^{n_i\times n_j}, & i \neq j, \end{cases} \tag{2.9}$$

with $n = \sum_{i=1}^{m} n_i$. The objective function is, with $A = (A_{ij})$ partitioned conformally with C,

$$f(\Gamma) = \|A - C(\Gamma)\|_F^2 = \sum_{i=1}^{m} \|A_{ii} - C(\gamma_{ii})\|_F^2 + \sum_{i\neq j} \|A_{ij} - \gamma_{ij} ee^T\|_F^2. \tag{2.10}$$

The problem is to minimize $f(\Gamma)$ subject to C being in the intersection of the set of positive semidefinite matrices and the set $\mathcal{C}$ of all patterned matrices of the form (2.9). Both these sets are closed convex sets and hence so is their intersection. It follows from standard results in approximation theory (see, for example, [91, p. 69]) that the problem has a unique solution. This solution can be computed by the alternating projections method, by repeatedly projecting onto the two sets in question. To obtain the projection onto the set $\mathcal{C}$ we simply apply Theorem 2.2.2 to each term in the first summation in (2.10) and for $i \neq j$ set $\gamma_{ij} = \sum_{(p,q)\in S_{ij}} a_{pq}/|S_{ij}|$, where S_{ij} is the set of indices of the elements in A_{ij} and $|S_{ij}|$ is the number of elements in S_{ij}. The latter projection can trivially be incorporated into Algorithm 3.3 of [65], replacing the projection onto the unit diagonal matrices therein, without losing the algorithm's guaranteed convergence.

If the intergroup correlations are equal and nonnegative, say $\gamma_{ij} \equiv \beta \geq 0$, and additionally all intragroup correlations satisfy $\gamma_{ii} \geq \beta$, the matrix $C(\Gamma)$ can be represented as an $m + 1$ factor correlation matrix $C(X)$, with $X \in \mathbb{R}^{n\times(m+1)}$ a block $m \times (m + 1)$ matrix $X = (X_{ij})$ with $X_{ij} \in \mathbb{R}^{n_i}$, where

$$X_{ij} = \begin{cases} \sqrt{\beta} e \in \mathbb{R}^{n_i}, & \text{j=1}, \\ \sqrt{\gamma_{ii} - \beta} e \in \mathbb{R}^{n_i}, & \text{j=i+1}, \\ 0 & \text{otherwise}. \end{cases}$$

To illustrate, we consider a small example where $m = 2$ and $n_1 = n_2 = 2$. Then X is a block 2×3 matrix and

$$XX^T = \left[\begin{array}{c|c|c} \sqrt{\beta} & \sqrt{\gamma_{11} - \beta} & 0 \\ \sqrt{\beta} & \sqrt{\gamma_{11} - \beta} & 0 \\ \hline \sqrt{\beta} & 0 & \sqrt{\gamma_{22} - \beta} \\ \sqrt{\beta} & 0 & \sqrt{\gamma_{22} - \beta} \end{array}\right] \left[\begin{array}{cc|cc} \sqrt{\beta} & \sqrt{\beta} & \sqrt{\beta} & \sqrt{\beta} \\ \hline \sqrt{\gamma_{11} - \beta} & \sqrt{\gamma_{11} - \beta} & 0 & 0 \\ \hline 0 & 0 & \sqrt{\gamma_{22} - \beta} & \sqrt{\gamma_{22} - \beta} \end{array}\right],$$

which simplifies to the desired form

$$\left[\begin{array}{cc|cc} \gamma_{11} & \gamma_{11} & \beta & \beta \\ \gamma_{11} & \gamma_{11} & \beta & \beta \\ \hline \beta & \beta & \gamma_{22} & \gamma_{22} \\ \beta & \beta & \gamma_{22} & \gamma_{22} \end{array}\right].$$

2.3 One Factor Problem

We now consider the one factor problem, for which the correlation matrix has the form, taking $k = 1$ in (2.4),

$$C(x) = \operatorname{diag}(1 - x_i^2) + xx^T, \qquad x \in \mathbb{R}^n. \tag{2.11}$$

The off-diagonal part of $C(x)$ agrees with that of the rank one matrix xx^T, so $C(x)$ is of the general diagonal plus semiseparable form [136].

We first consider the uniqueness of this representation.

Theorem 2.3.1. *Let $C = C(x)$ for some $x \in \mathbb{R}^n$ with p nonzero elements ($0 \le p \le n$). If $p = 1$ then $C = I_n$ and $C = C(y)$ for any y with at least $n-1$ zero entries. If $p = 2$ and x_i, x_j are the nonzero entries of x then $C = C(y)$ for $y = \theta x_i e_i + \theta^{-1} x_j e_j$ for any $\theta \ne 0$. Otherwise, $C = C(y)$ for exactly two vectors: $y = \pm x$.*

Proof. Without loss of generality we can assume $C = \operatorname{diag}(1 - x_i^2) + xx^T$ has been symmetrically permuted so that $x_i \ne 0$ for $i = 1\colon p$ and $x_i = 0$ for $i = p+1\colon n$. If $p = 1$ then $C = I_n$ and x_1 is arbitrary, which gives the first part. Suppose $p > 1$. We can write

$$C = \begin{bmatrix} C_1 & 0 \\ 0 & I_{n-p} \end{bmatrix}, \tag{2.12}$$

where $C_1 \in \mathbb{R}^{p\times p}$ has all nonzero elements. If $p = 2$ then $c_{12} = x_1 x_2 = \theta x_1 \cdot \theta^{-1} x_2 \equiv y_1 y_2$ for any $\theta \ne 0$ and $C = C(y)$ with $y_3, \dots, y_n$ necessarily zero. Assume $p > 2$ and suppose $C = \operatorname{diag}(1 - y_i^2) + yy^T$. Then, from (2.12), $y_i \ne 0$ for $i = 1\colon p$ and $y_i = 0$ for $i = p+1\colon n$. From $C = \operatorname{diag}(1 - y_i^2) + yy^T$ we have

$$\frac{c_{i,i+1} c_{i,i+2}}{c_{i+1,i+2}} = y_i^2, \qquad 1 \le i \le p-2, \tag{2.13}$$

which determines the first $p-2$ components of y_i up to their signs, and y_p is determined by $y_{p-2} y_p = c_{p-2,p}$ and y_{p-1} by $y_{p-1} y_p = c_{p-1,p}$. Finally, the equations $c_{1j} = y_1 y_j$, $1 \le j \le p$, ensure that $\operatorname{sign}(y_j)$, $2 \le j \le p$, is determined by $\operatorname{sign}(y_1)$. ∎

Before addressing the nearness problem we develop some properties of $C(x)$.

Lemma 2.3.2. *The determinant of $C(x)$ is given by*

$$\det(C(x)) = \prod_{i=1}^{n}(1-x_i^2) + \sum_{i=1}^{n} x_i^2 \prod_{\substack{j=1\\ j\neq i}}^{n}(1-x_j^2). \tag{2.14}$$

Proof. Define the vector $z(\epsilon)$ by $z_i = x_i + \epsilon$. For sufficiently small ϵ, $z(\epsilon)$ has no element equal to 1 and $D = \mathrm{diag}(1-z_i^2)$ is nonsingular. Hence $C(z) = D + zz^T = D(I_n + D^{-1}z \cdot z^T)$, from which it follows that

$$\det(C(z)) = \det(D)(1 + z^T D^{-1} z) = \prod_{i=1}^{n}(1-z_i^2) \cdot \left(1 + \sum_{i=1}^{n} \frac{z_i^2}{1-z_i^2}\right).$$

On multiplying out, the formula takes the form (2.14) with x replaced by $z(\epsilon)$, and letting $\epsilon \to 0$ gives the result, since the determinant is a continuous function of the matrix elements. ∎

For the case $x_i \neq 1$ for all i the formula (2.14) is a special case of a result in [119, Section 2.1].

Corollary 2.3.3. *If $|x| \leq e$ with $x_i = 1$ for at most one i then $C(x)$ is nonsingular. $C(x)$ is singular if $x_i = x_j = 1$ for some $i \neq j$.*

The matrix $C(x)$ is not always a correlation matrix because it is not always positive semidefinite. We know from the discussion of the k factor case in Section 2.1 that a sufficient condition for $C(x)$ to be a correlation matrix is that $|x| \leq e$. This condition arises in the factor model described in Section 2.1 and hence is natural in the applications. The two extreme cases are when $|x| = e$, in which case $C = xx^T$ is of rank 1, and when $x = 0$, in which case $C = I_n$ has rank n. The next result shows more generally that the rank is determined by the number of elements of x of modulus 1.

Theorem 2.3.4. *For $C = C(x) \in \mathbb{R}^{n\times n}$ in (2.11) with $|x| \leq e$ we have* $\mathrm{rank}(C) = \min(p+1, n)$*, where p is the number of x_i for which $|x_i| < 1$.*

Proof. By a symmetric permutation of C we can assume, without loss of generality, that $|x_i| < 1$ for $i = 1\colon p$ and $|x_i| = 1$ for $i = p+1\colon n$. The result is true for $p = n$ by Corollary 2.3.3, so assume $p \leq n-1$. Partition $x = [y, z]^T$, where $y \in \mathbb{R}^p$; thus $|y| < e$ and $|z| = e$. Then

$$C = \begin{bmatrix} C_1 & yz^T \\ zy^T & zz^T \end{bmatrix},$$

where $C_1 \in \mathbb{R}^{p\times p}$ is positive definite. With $X^T = \begin{bmatrix} I & 0 \\ -zy^T C_1^{-1} & I \end{bmatrix}$ we have

$$X^T C X = \begin{bmatrix} C_1 & 0 \\ 0 & S \end{bmatrix},$$

where

$$S = zz^T - zy^T C_1^{-1} yz^T = zz^T - (y^T C_1^{-1} y)zz^T = (1 - y^T C_1^{-1} y)zz^T.$$

Hence $\operatorname{rank}(C) = \operatorname{rank}(C_1) + \operatorname{rank}(S) = p + \operatorname{rank}(S)$. Now $C_1 = \operatorname{diag}(1 - y_i^2) + yy^T =: D + yy^T$, where D is positive definite, and the Sherman–Morrison formula gives

$$C_1^{-1} = D^{-1} - \frac{D^{-1} yy^T D^{-1}}{1 + y^T D^{-1} y}.$$

So

$$y^T C_1^{-1} y = \frac{y^T D^{-1} y}{1 + y^T D^{-1} y} < 1.$$

Since $y^T C_1^{-1} y \neq 1$ and $z \neq 0$, S has rank 1 and the result follows. □

Now we are ready to address the nearness problem. Consider the problem of minimizing

$$f(x) = \|A - \big(\operatorname{diag}(1 - x_i^2) + xx^T\big)\|_F^2, \tag{2.15}$$

subject to $|x| \leq e$, where $A \in \mathbb{R}^{n \times n}$ is symmetric and we can assume without loss of generality that $a_{ii} = 1$ for all i. For $n = 2$, $f(x) = 0$ is the global minimum, attained at $x = [\theta a_{12},\ \theta^{-1}]^T$ for any $\theta \neq 0$. For $n = 3$, $f(x) = 0$ is again achieved; if $a_{ij} \neq 0$ for all i and j then there are exactly two minimizers. But for $n \geq 4$ there are more equations than variables in $A = \operatorname{diag}(1 - x_i^2) + xx^T$ and so the global minimum is generally positive.

Note that because of Theorem 2.3.1 we could further restrict one element of x to $[0, 1]$. We could go further and restrict all the elements of x to $[0, 1]$ in order to obtain a correlation matrix with nonnegative elements—a constraint that is imposed in [125], [137].

The function f is clearly twice continuously differentiable, and we need to find its gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$. Setting $\widehat{A} = A - I_n$ and $D = \operatorname{diag}(x_i)$, noticing that $\widehat{a}_{ii} \equiv 0$, and using properties of the trace operator, we can rewrite f as

$$\begin{aligned} f(x) &= \langle \widehat{A}, \widehat{A} \rangle + 2\langle \widehat{A}, D^2 \rangle - 2\langle \widehat{A}, xx^T \rangle \\ &\quad + \langle xx^T, xx^T \rangle - 2\langle xx^T, D^2 \rangle + \langle D^2, D^2 \rangle \\ &= \langle \widehat{A}, \widehat{A} \rangle - 2x^T \widehat{A} x + (x^T x)^2 - \sum_{i=1}^{n} x_i^4. \end{aligned} \tag{2.16}$$

Lemma 2.3.5. *For f in (2.15) we have*

$$\nabla f(x) = 4\big((x^T x)x - \widehat{A}x - D^2 x\big), \tag{2.17}$$

$$\nabla^2 f(x) = 4(2xx^T + (x^T x)I_n - \widehat{A} - 3D^2). \tag{2.18}$$

Proof. We have $\nabla(x^T\widehat{A}x) = 2\widehat{A}x$ and $\nabla^2(x^T\widehat{A}x) = 2\widehat{A}$. Similarly, $\nabla(\sum_{i=1}^n x_i^4) = 4D^2x$ and $\nabla^2(\sum_{i=1}^n x_i^4) = 12D^2$. It is straightforward to show that for $h(x) = (x^Tx)^2$ we have $\nabla h(x) = 4(x^Tx)x$ and $\nabla^2 h(x) = 8xx^T + 4(x^Tx)I_n$. The formulae follow by differentiating (2.16) and using these expressions. ∎

Notice that at $x = 0$, $\nabla f(0) = 0$ and $\nabla^2 f(0) = -4\widehat{A}$. For $A \neq I_n$, since $\widehat{A}$ is symmetric and indefinite (by virtue of its zero diagonal), $x = 0$ is a saddle point of f. Another deduction that can be made from the lemma is that if $a_{ii} = 1$ and $|a_{ij}| \leq 1$ for all i and j then $x = e$ is a solution if and only if $A = ee^T$.

Denote a global minimizer of f by $\overline{x}$. If $f(\overline{x}) = 0$ then $A = \operatorname{diag}(1 - \overline{x}_i^2) + \overline{x}\overline{x}^T$ is precisely of the sought structure and we call A reproducible. We ignore the constraint $|x| \leq e$ for the rest of this section. We now examine the properties of the Hessian matrix at $\overline{x}$ for reproducible A and will later draw conclusions about the nonreproducible case. Note that (2.18) simplifies to $\nabla^2 f(\overline{x}) = 4((\overline{x}^T\overline{x})I_n + \overline{x}\,\overline{x}^T - 2\overline{D}^2)$, where $\overline{D} = \operatorname{diag}(\overline{x}_i)$. Therefore we consider the matrix

$$H_n = H_n(x) = (x^Tx)I_n + xx^T - 2D^2, \qquad x \in \mathbb{R}^n. \tag{2.19}$$

For example,

$$H_4 = \begin{bmatrix} x_2^2 + x_3^2 + x_4^2 & x_1x_2 & x_1x_3 & x_1x_4 \\ x_2x_1 & x_1^2 + x_3^2 + x_4^2 & x_2x_3 & x_2x_4 \\ x_3x_1 & x_3x_2 & x_1^2 + x_2^2 + x_4^2 & x_3x_4 \\ x_4x_1 & x_4x_2 & x_4x_3 & x_1^2 + x_2^2 + x_3^2 \end{bmatrix}.$$

We want to determine the definiteness and nonsingularity properties of H_n. Without loss of generality we can suppose that

$$|x_1| \geq |x_2| \geq \cdots \geq |x_p| > x_{p+1} = \cdots = x_n = 0, \tag{2.20}$$

with $p \geq 1$. If $n = 4$ and $p = 3$ then H_4 has the form

$$\begin{bmatrix} x_2^2 + x_3^2 & x_1x_2 & x_1x_3 & 0 \\ x_2x_1 & x_1^2 + x_3^2 & x_2x_3 & 0 \\ x_3x_1 & x_3x_2 & x_1^2 + x_2^2 & 0 \\ 0 & 0 & 0 & x_1^2 + x_2^2 + x_3^2 \end{bmatrix} = \operatorname{diag}(H_3, x_1^2 + x_2^2 + x_3^2).$$

In general,

$$H_n = \operatorname{diag}(H_p, D_p), \qquad D_p = (x_1^2 + x_2^2 + \cdots + x_p^2)I_{n-p}.$$

D_p has positive diagonal entries and hence the definiteness properties of H_n are determined by those of H_p. So the problem has been reduced to the case of x_i nonzero.

Theorem 2.3.6. *H_n is positive semidefinite. Moreover, H_n is nonsingular if and only if at least three of $x_1, x_2, \ldots, x_n$ are nonzero.*

Proof. From the foregoing analysis we can restrict our attention to H_p and assume that (2.20) holds. Let $W = \operatorname{diag}(x_1, x_2, \ldots, x_p)$. Then $\widetilde{H}_p = W^T H_p W$ has the form illustrated for $p = 4$ by

$$\widetilde{H}_4 = \begin{bmatrix} x_1^2(x_2^2+x_3^2+x_4^2) & x_1^2x_2^2 & x_1^2x_3^2 & x_1^2x_4^2 \\ x_2^2x_1^2 & x_2^2(x_1^2+x_3^2+x_4^2) & x_2^2x_3^2 & x_1^2x_4^2 \\ x_3^2x_1^2 & x_3^2x_2^2 & x_3^2(x_1^2+x_2^2+x_4^2) & x_1^2x_4^2 \\ x_4^2x_1^2 & x_4^2x_2^2 & x_4^2x_3^2 & x_4^2(x_1^2+x_2^2+x_3^2) \end{bmatrix}.$$

Thus $\widetilde{H}_p$ is diagonally dominant with nonnegative diagonal elements and with equality in the diagonal dominance conditions for every row (or column); it is therefore positive semidefinite by Gershgorin's theorem. Suppose $\widetilde{H}_p$ is singular. Then $\lambda = 0$ is an eigenvalue lying on the boundary of the set of Gershgorin discs (in fact it is on the boundary of every Gershgorin disc). Hence by [69, Theorem 6.2.5], since $\widetilde{H}_p$ has all nonzero entries any null vector z of $\widetilde{H}_p$ has the property that $|z_i|$ is the same for all i. Hence any null vector can be taken to have elements $z_i = \pm 1$. But it is easy to see that no such vector can be a null vector of $\widetilde{H}_p$ for $p > 2$. Hence $\widetilde{H}_p$ is nonsingular for $p > 2$. Since H_p is congruent to $\widetilde{H}_p$, H_p is positive definite for $p > 2$. For $p = 1, 2$, H_p is singular. The result follows. ∎

Since $\overline{x}$ is, by definition, a global minimizer and is usually one of exactly two distinct global minimizers $\pm\overline{x}$, by Theorem 2.3.1, Theorem 2.3.6 does not provide any significant new information about $\overline{x}$. However, it does tell us something about the nonreproducible case. For general A, $\widehat{H}_n = \frac{1}{4}\nabla^2 f(x)$ can be written, using (2.18), as

$$\widehat{H}_n = \left((x^Tx)I_n + xx^T - 2D^2\right) + (xx^T - \widehat{A} - D^2) = H_n + E_n,$$

where H_n, defined in (2.19), is positive semidefinite by Theorem 2.3.6 and moreover positive definite if at least three components of x are nonzero. Now E_n has zero diagonal and in general is indefinite. Furthermore, E_n is singular at a stationary point x since $E_n x = 0$ by (2.17). We can conclude that at a stationary point x having at least three nonzero components the Hessian $\nabla^2 f(x) = 4\widehat{H}_n$ will be positive definite if $\|E_n\|_F$ is sufficiently small, that is, if $|(E_n)_{ij}| = |x_i x_j - a_{ij}|$ is sufficiently small for all i and j. In this case x is a local minimizer of f.

2.4 *k* Factor Problem

Now we consider the general k factor problem, for which $C(X) = D + \sum_{j=1}^{k} x_j x_j^T$ as in (2.4). We require that (2.3) holds, so that $C(X)$ is positive semidefinite and hence

is a correlation matrix.

As noted by Lawley and Maxwell [83], the representation (2.4) is far from unique as we can replace X by XQ for any orthogonal matrix $Q \in \mathbb{R}^{k\times k}$ without changing $C(X)$. This corresponds to a rotation of the factors in the terminology of factor analysis. Some approaches to determining a unique representation are described in [76], [83]. Probably the most popular one is the varimax method of Kaiser [77]. Given an X defining a matrix $C(X)$ with k factor structure, this method maximizes the function

$$V(P) = \left\| \left(I_n - \frac{1}{n} ee^T \right) (XP \circ XP) \right\|_F$$

over all orthogonal P and then uses the representation $C(XP)$. Here the symbol "$\circ$" denotes the Hadamard product ($A \circ B = (a_{ij}b_{ij})$). The method rotates and reflects the rows of X such that after squaring the elements of each column differ maximally from their mean value, which explains the name varimax.

The nearness problem for our k factor representation is to minimize

$$f(X) = \|A - (I_n + XX^T - \mathrm{diag}(XX^T))\|_F^2 \tag{2.21}$$

over all $X \in \mathbb{R}^{n\times k}$ satisfying the constraints (2.3). As before, $A \in \mathbb{R}^{n\times n}$ is symmetric with unit diagonal and we set $\widehat{A} = A - I_n$. We now obtain the first and second derivatives of f.

Since $\widehat{A}$ has zero diagonal we have $\langle \widehat{A}, \mathrm{diag}(XX^T)\rangle = 0$ and also $\langle \mathrm{diag}(XX^T) - XX^T, \mathrm{diag}(XX^T)\rangle = 0$. The function f can therefore be written

$$f(X) = \langle \widehat{A}, \widehat{A}\rangle - 2\langle \widehat{A}X, X\rangle + \langle XX^T, XX^T\rangle - \langle XX^T, \mathrm{diag}(XX^T)\rangle. \tag{2.22}$$

The next result gives a formula for the gradient, which is now most conveniently expressed as the matrix $\nabla f(X) = (\partial f(X)/\partial x_{ij}) \in \mathbb{R}^{n\times k}$.

Lemma 2.4.1. *For f in (2.21) we have*

$$\nabla f(X) = 4\big(X(X^TX) - \widehat{A}X - \mathrm{diag}(XX^T)X\big). \tag{2.23}$$

Proof. It is straightforward to show that $\nabla\langle \widehat{A}X, X\rangle = 2\widehat{A}X$. Next, consider the term $h_1(x) = \langle XX^T, XX^T\rangle$. Consider the auxiliary function $g_1 : \mathbb{R} \to \mathbb{R}$, given by $g_1(t) = h_1(X + tZ)$, for arbitrary $Z \in \mathbb{R}^{n\times k}$. Clearly, $g_1'(0) = \langle \nabla h_1(X), Z\rangle$. After some algebraic manipulations we find that

$$g_1'(0) = 2\langle X^TX, X^TZ\rangle + 2\langle X^TX, Z^TX\rangle = 4\langle X(X^TX), Z\rangle.$$

Therefore, $\nabla h_1(X) = 4X(X^TX)$. Similarly, we find that the gradient of $h_2(x) = \langle XX^T, \mathrm{diag}(XX^T)\rangle$ is $\nabla h_2(X) = 4\ \mathrm{diag}(XX^T)X$. The result follows. □

Notice that when $k = 1$, (2.23) reduces to (2.17).

The Hessian of f is an $nk \times nk$ matrix that is most conveniently viewed as a matrix representation of the Fréchet derivative $L_{\nabla f}$ of ∇f. Recall that the Fréchet derivative $L_g(X, E)$ of $g : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ at X in the direction E is a linear operator satisfying $g(X + E) = g(X) + L_g(X, E) + o(\|E\|)$ [66, Section 3.1] or Appendix A.2. We can determine the Fréchet derivative of ∇f by finding the linear part of the expansion for $\nabla f(X + E)$. For example, to find the derivative of the first term in (2.23) we set $f_1(X) = X(X^TX)$ and consider

$$f_1(X + E) = f_1(X) + X(X^TE) + X(E^TX) + E(X^TX) + O(\|E\|^2).$$

Hence $L_{f_1}(X, E) = X(X^TE) + X(E^TX) + E(X^TX)$. For the third term, f_3, we have, similarly, $L_{f_3}(X, E) = \mathrm{diag}(XE^T)X + \mathrm{diag}(EX^T)X + \mathrm{diag}(XX^T)E$.

Lemma 2.4.2. *For f in (2.21) we have*

$$\begin{aligned} L_{\nabla f}(X, E) \;=\; & 4\big(X(X^TE) + X(E^TX) + E(X^TX) - \widehat{A}E \\ & - (\mathrm{diag}(XE^T)X + \mathrm{diag}(EX^T)X + \mathrm{diag}(XX^T)E)\big). \quad (2.24) \end{aligned}$$

2.5 Numerical Methods

The problem of interest is

$$\text{minimize } f(X) = \|A - (I_n + XX^T - \mathrm{diag}(XX^T))\|_F^2 \quad (2.25a)$$

$$\text{subject to } X \in \Omega := \left\{ X \in \mathbb{R}^{n\times k} : \textstyle\sum_{j=1}^k x_{ij}^2 \le 1,\ i = 1\colon n \right\}, \quad (2.25b)$$

where $A \in \mathbb{R}^{n\times n}$ is a given symmetric matrix. The set Ω is convex. However, since the objective function f in (2.25a) is nonconvex we can only expect to find a local minimum, though if we achieve $f(X) = 0$ we know that X is a global minimizer.

We consider several different numerical methods for solving the problem. We first consider how to start the iterations. We will take a matrix of a simple, parametrized form, optimize the parameter, and then show that this matrix yields a smaller function value than the zero matrix. Let λ be the largest eigenvalue of A, which is at least 1 if A has unit diagonal, which can be assumed without loss of generality. We take for the starting matrix $X^{(0)}$ a matrix αve^T whose columns are all the same multiple of the eigenvector v corresponding to λ. The scalar α is chosen to minimize $f(\alpha ve^T)$ subject to αve^T staying in the feasible set Ω. Straightforward computations show that the optimal α is

$$\alpha_{\mathrm{opt}} = \min\left\{ \left(\frac{(\lambda - 1)\|v\|_2^2}{k\|v\|_2^4 - k\sum_i v_i^4} \right)^{1/2}, \ \frac{1}{k^{1/2}\max_i |v_i|} \right\}.$$

This $X^{(0)}$ can be inexpensively computed by using the power method or the Lanczos method to obtain λ and v. Moreover, it is guaranteed to yield a smaller value of f than the zero matrix if $\lambda > 1$ since, from (2.22),

$$\begin{aligned} f(\alpha_{\mathrm{opt}} v e^T) &= \langle \widehat{A}, \widehat{A} \rangle - 2\alpha_{\mathrm{opt}}^2 k(\lambda - 1)\|v\|_2^2 + \alpha_{\mathrm{opt}}^4 k^2 \|v\|_2^4 - \alpha_{\mathrm{opt}}^4 k^2 \sum_i v_i^4 \\ &= \langle \widehat{A}, \widehat{A} \rangle - \alpha_{\mathrm{opt}}^2 k \left(2(\lambda - 1)\|v\|_2^2 - \alpha_{\mathrm{opt}}^2 \left(k\|v\|_2^4 - k \sum_i v_i^4 \right) \right) \\ &\leq \langle \widehat{A}, \widehat{A} \rangle - \alpha_{\mathrm{opt}}^2 k \left(2(\lambda - 1)\|v\|_2^2 - (\lambda - 1)\|v\|_2^2 \right) \\ &= \langle \widehat{A}, \widehat{A} \rangle - \alpha_{\mathrm{opt}}^2 k(\lambda - 1)\|v\|_2^2 < f(0). \end{aligned}$$

As noted by Anderson, Sidenius, and Basu [8], and as we will see later for some problem types, minimizing f without the constraint $X \in \Omega$ may yield a solution of the constrained problem (2.25). This motivates us to consider first methods that ignore or only partly incorporate the constraint. The first method is the alternating directions (or coordinate search) method. Regarding f as a function of just x_{ij} we have

$$f(x_{ij}) = \text{const.} + 2 \sum_{q \neq i} \left(a_{iq} - \sum_{s=1}^{k} x_{is} x_{qs} \right)^2,$$

so

$$\begin{aligned} f'(x_{ij}) &= 4 \sum_{q \neq i} (-x_{qj}) \left(a_{iq} - \sum_{s=1}^{k} x_{is} x_{qs} \right) \\ &= 4 \left(- \sum_{q \neq i} x_{qj} a_{iq} + \sum_{q \neq i} x_{qj} x_{ij} x_{qj} + x_{qj} \sum_{s \neq j} x_{is} x_{qs} \right) \\ &= 4 \left(x_{ij} \sum_{q \neq i} x_{qj}^2 + \sum_{q \neq i} x_{qj} \left(\sum_{s \neq j} x_{is} x_{qs} - a_{iq} \right) \right). \end{aligned}$$

Hence $f'(x_{ij}) = 0$ for

$$x_{ij} = \frac{\sum_{q \neq i} x_{qj} \left(a_{iq} - \sum_{s \neq j} x_{is} x_{qs} \right)}{\sum_{q \neq i} x_{qj}^2}. \tag{2.26}$$

We can therefore repeatedly minimize over each x_{ij} in turn using (2.26). If the new x_{ij} is not in the interval $[-1, 1]$ we project it back onto the interval by reducing $|x_{ij}|$ appropriately, since x_{ij} must lie in this interval if it is in Ω. Convergence of this method to a stationary point of f can be proved under suitable conditions [79, Section 8.1], [132]. After the projection step x may nevertheless lie outside Ω if $k > 1$, but we do not project onto Ω because this may cause the method not to converge.

Anderson, Sidenius, and Basu [8] propose another method to solve the k factor problem. For $F(X) = I_n - \mathrm{diag}(XX^T)$ it iteratively generates a sequence $\{X_i\}_{i\geq 0}$ with

$$X_i = \underset{X\in\mathbb{R}^{n\times k}}{\operatorname{argmin}} \|A - F(X_{i-1}) - XX^T\|_F. \tag{2.27}$$

The minimizer of (2.27) is found by principal component analysis. Let $P^T\Lambda P$ be a spectral decomposition of $A - F(X_{i-1})$, with P orthogonal and Λ diagonal with diagonal elements in nonincreasing order. Then the minimizer is (in MATLAB notation) $X_i = P(:, 1\colon k)\widetilde{\Lambda}^{1/2}$, where $\widetilde{\Lambda} = \mathrm{diag}(\max(\lambda_1, 0), \ldots, \max(\lambda_k, 0))$. Thus just the k largest eigenvalues and corresponding eigenvectors of $A - F(X_{i-1})$ are needed, and these can be inexpensively computed by the Lanczos iteration or by orthogonally reducing the matrix to tridiagonal form and applying the bisection method followed by inverse iteration [133, pp. 227 ff.]. This method is also known as the principal factors method [53, Section 10.4].

We note that this method is equivalent to the alternating projections method. Recall from Section 1.4.2 that this method generates a sequence $\{Z_i\}_{i\geq 0}$ with $Z_i = P_S(P_U(Z_{i-1}))$, where P_S and P_U are projection operators onto the sets

$$\begin{aligned} \mathcal{U} &:= \{W \in \mathbb{R}^{n\times n} : w_{ij} = a_{ij} \text{ for } i \neq j\}, \\ \mathcal{S} &:= \{W \in \mathbb{R}^{n\times n} : W = XX^T \text{ for some } X \in \mathbb{R}^{n\times k}\}. \end{aligned}$$

The projection $P_S(Z)$ is formed by the construction described in the previous paragraph. With $Z_0 = X_0X_0^T$, the equivalence between the $\{X_k\}$ and the $\{Z_k\}$ is given by $Z_i \equiv X_iX_i^T$.

Although this method has been successfully used [8], [73] it is not guaranteed to converge. The standard convergence theory [42] for the alternating projections method is not applicable since the set S is not convex for $k < n$ and the sets U and S do not have a point in common unless the objective function f is zero at the global minimum.

Since there is no guarantee that the final iterates of the alternating directions and principal factors methods lie in the feasible set Ω, we project onto this set after the computation. To project an $n \times k$ matrix Y with rows y_i^T onto Ω we simply replace any row y_i^T such that $\|y_i\|_2 > 1$ by $y_i^T/\|y_i\|_2$. We denote this projection by $P(Y)$.

The next method solves the full, constrained problem (2.25) and generates a sequence of matrices that is guaranteed to converge r-linearly to a stationary point of (2.25). Introduced by Birgin, Martínez, and Raydan [16], [17], the spectral projected gradient method aims to minimize a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ on a nonempty closed convex set. The method has the form $x_{k+1} = x_k + \alpha_k d_k$ where d_k is chosen to be $P(x_k - t_k\nabla f(x_k)) - x_k$, with $t_k > 0$ a precomputed scalar. The

direction d_k is guaranteed to be a descent direction [16, Lemma 2.1] and the scalar α_k is selected by a nonmonotone line search strategy. The cost per iteration is low for our problem because the projection P is inexpensive to compute. An R implementation of the method is available [139].

Our analysis in the previous sections suggests applying a Newton method to our problem since the gradient and the Hessian are explicitly known and can be computed in a reasonable time. As the constraints defining Ω in (2.25b) are nonlinear for $k > 1$ we distinguish here between the one factor case and the k factor case.

For $k = 1$ we use the routine `e04lb` of the NAG Toolbox for MATLAB [100], which implements a globally convergent modified Newton method for minimizing a nonlinear function subject to upper and lower bounds on the variables; these bounds allow us to enforce the constraint (2.25b). This method uses the first derivative and the Hessian matrix.

For $k > 1$ we apply the routine `e04wd` of the NAG Toolbox for MATLAB, which implements an SQP method. This routine deals with the nonlinear constraints (2.25b) but does not use the Hessian. In order to have an unconstrained optimization method that we can compare with the principal factors method, we apply the function `fminunc` of the MATLAB Optimization Toolbox [97], which implements a subspace trust region method based on the interior-reflective Newton method. This algorithm uses the first derivative and the Hessian. As for the principal factors method, if necessary we project the final iterate onto the feasible set Ω to satisfy the constraints.

We will use the following abbreviations for the methods:

- AD: alternating directions method.
- PFM: principal factors method.
- SPGM: spectral projected gradient method.
- Newt$_1$: `e04lb`.
- Newt$_2$: `fminunc`.
- SQP: `e04wd`.

We summarize the properties of the methods in Table 2.1.

2.6 Computational Experiments

Our experiments were performed in MATLAB R2007a using the NAG Toolbox for MATLAB Mark 22.0 on an Intel Pentium 4 (3.20 GHz). In order to define the

Table 2.1: Summary of the methods, with final column indicating the available convergence results (see the text for details).

Method	Required derivatives	Constraints satisfied?	Convergence?
AD	none	needs final projection for $k > 1$	yes
PFM	none	needs final projection for all k	no result
SPGM	gradient	yes	r-linear
Newt$_1$ ($k = 1$)	gradient, Hessian	yes	quadratic
Newt$_2$ ($k > 1$)	gradient, Hessian	needs final projection for all k	quadratic
SQP ($k > 1$)	gradient	yes	quadratic

stopping criterion used in all the algorithms we first introduce an easy to compute measurement of stationarity. We define the function $q : \mathbb{R}^{n\times k} \mapsto \mathbb{R}^{n\times k}$ by

$$q(X) = P\big(X - \nabla f(X)\big) - X.$$

It can be shown that a point $X^* \in \Omega$ is a stationary point of our problem (2.25) if and only if $q(X^*) = 0$ [45, (2.6)]. The stopping criterion is

$$\|q(X)\|_F \leq \text{tol}, \tag{2.28}$$

where tol will be specified for the individual tests below. We use the same notation and criterion when no constraints are imposed, in which case P is the identity and $q(x)$ reduces to the gradient $-\nabla f(X)$.

Since the final iterates of these methods may not be in the feasible set Ω, prior to our enforced projection onto it, we introduce a measurement of constraint violation at a point X, given by the function $v : \mathbb{R}^{n\times k} \to \mathbb{R}$ with

$$v(X) = \sum_{i=1}^{n} \max\big(\|y_i\|_2^2 - 1, 0\big), \qquad X^T = [y_1, \ldots, y_n]. \tag{2.29}$$

Our test matrices are chosen from five classes.

- expij: The correlation matrix $(e^{-|i-j|})_{i,j=1}^n$ occurring in annual forward rate correlations associated with LIBOR models [26, Section 6.9] and can be used to calibrate the lognormal forward rate model as described in [6].
- corrand: A random correlation matrix generated by `gallery('randcorr', n)`.
- corkfac: A random correlation matrix generated by $A = \text{diag}(I_n - XX^T) + XX^T$ where $X \in \mathbb{R}^{n\times k}$ is a random matrix with elements from the uniform distribution on $[-1, 1]$ that is then projected onto Ω. Here the objective function f is zero at the global minimum.

- randneig: A symmetric matrix generated by $A = \frac{1}{2}(B + B^T) + \mathrm{diag}(I_n - B)$ where B is the first matrix out of a sequence of random matrices with elements from the uniform distribution on $[-1, 1]$ such that A has a negative eigenvalue.

- cor1399: A symmetric, unit-diagonal matrix constructed from stock data provided by a fund management company. It has dimension $n = 1399$ and is highly rank-deficient but not positive semidefinite. This matrix was also used in [23], [65].

Let us prove that the matrix of the form of expij is always a correlation matrix. The following theorem gives this result for a generalization of the class of matrices described by expij.

Theorem 2.6.1. *Let $\phi(x)$ be a continuous, nonnegative, even, and real function. Let further $\phi(0) = 1$ and let $\phi(x)$ be convex and monotonically decreasing on $[0, \infty]$. Then the matrix*

$$C := [\phi(x_i - x_j)]_{i,j=1}^n$$

is a correlation matrix for any real numbers x_i for $i = 1, \ldots, n$.

Proof. Since $C_{ii} = \phi(0)$ for all $i = 1, \ldots, n$ and $C_{ij} = C_{ji}$ for all $i, j = 1, \ldots, n$ the matrix C is symmetric and has ones on its diagonal. Further, it follows from [14, Section 5.2.15 b] that the function ϕ is positive, i.e. for every positive integer N and every choice of $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$ and $y = (y_1, \ldots, y_N)^T \in \mathbb{C}^N$:

$$\sum_{i,j=1}^{N} \phi(x_i - x_j) y_r \bar{y}_s \geq 0$$

for all $r, s = 1, \ldots, N$. This implies that $y^T C y \geq 0$ for every $y \in \mathbb{R}^n$ and hence, C is a correlation matrix. □

Corollary 2.6.2. *The matrix*

$$C = [e^{-|x_i - x_j|^\alpha}]_{i,j=1}^n$$

is a correlation matrix for $x_1, \ldots, x_n \in \mathbb{R}$ and $\alpha \in (0, 1]$.

Proof. As for all $\alpha \in (0, 1]$ the function $\phi(x) := e^{-|x|^\alpha}$ is one at zero, continuous, nonnegative, even, and convex and monotonically decreasing on $[0, \infty]$ the matrix C is a correlation matrix by Theorem 2.6.1. The latter property is true as the second derivative of $\phi(x)$ is nonnegative on $(0, \infty)$. □

Note that the Corollary 2.6.2 includes the case $C = [e^{-|(i-j)|}]_{i,j=1}^n$ for $x_i = i$ and $i = 1, \ldots, n$. Note also that the result of Corollary 2.6.2 can be extended to $\alpha \in (0, 2]$ as demonstrated in [14, Theorem 5.2.17].

2.6.1 Test Results for $k = 1$

We first consider the one factor case. Each method was started with the rank one matrix defined at the start of Section 2.5.

In Tables 2.2 and 2.3 we report results averaged over 10 instances of each of the three classes of random matrices for $n = 100$ and $n = 2000$ with tolerance $\text{tol} = 10^{-3}$ and $\text{tol} = 10^{-6}$, respectively. Table 2.4 gives the results for the matrix cor1399 with tolerance $\text{tol} = 10^{-3}$. We use the following abbreviations:

- t: mean computational time (seconds).
- it: mean number of iterations.
- it_{sd}: standard deviation of the number of iterations.
- dist_0: mean initial value of $f(X)^{1/2}$.
- dist: mean final value of $f(X)^{1/2}$ after the final projection onto the feasible set.
- nq_0: mean initial value of $\|q(X)\|_F$.
- nq: mean final value of $\|q(X)\|_F$ before the final projection onto the feasible set.
- v: mean final value of $v(X)$ before the final projection onto the feasible set.

For the method AD one iteration is defined to be a sweep over which the objective function f is minimized over each coordinate direction in turn.

Several comments can be made on Tables 2.2–2.4.

- The values of v in (2.29) are all zero except for PFM on the randneig matrices, where the final projection onto Ω causes dist for the accepted X to exceed that for the other methods. Except in these cases the mean function values of the final iterates of the methods do not differ significantly. In particular, for the corkfac matrices the sequences appear to approach the global minimum. Except for the randneig problems all the constraints are inactive at the computed final iterates, so by Theorem 2.3.4 the matrices $C(X)$ have full rank. For the randneig problems about half the constraints are inactive, and this number is slightly bigger for the matrix returned by PFM than for the other methods.
- None of the methods always outperforms the others in computational time. The relative performance of the individual methods depends on the tolerance, the problem size and the problem type. AD performs very well for $\text{tol} = 10^{-3}$

Table 2.2: Results for the random one factor problems with tol = 10^{-3}.

	t	**it**	**it$_{sd}$**	**dist**	**nq**	**v**	**t**	**it**	**it$_{sd}$**	**dist**	**nq**	**v**
	$n = 100$						$n = 2000$					
	corrand, dist$_0$=5.6646, nq$_0$=8e-2						**corrand**, dist$_0$=26.006, nq$_0$=5e-3					
AD	0.22	110	78	5.6642	9e-4	0	3.3	5.2	1.5	26.006	9e-4	0
PFM	0.09	10	5.4	5.6642	8e-4	0	68	1.1	0.2	26.006	2e-4	0
Newt$_1$	0.02	4.7	2.4	5.6643	3e-4	0	23	1.8	0.4	26.006	6e-4	0
SPGM	0.11	57	29	5.6642	6e-4	0	9.8	5.2	0.8	26.006	8e-4	0
	corkfac, dist$_0$=0.3697, nq$_0$=6e0						**corkfac**, dist$_0$=0.3718, nq$_0$=3e1					
AD	0.01	5.0	0.6	2.25e-5	4e-4	0	3.1	5.2	0.6	5.06e-6	4e-4	0
PFM	0.03	3.0	0	4.03e-5	6e-4	0	15	2.2	0.3	1.56e-6	1e-4	0
Newt$_1$	0.01	2.0	0	1.45e-7	3e-6	0	16	2.0	0	1.5e-11	1e-9	0
SPGM	0.02	6.0	1.2	2.67e-5	3e-4	0	11	4.6	0.9	7.72e-6	4e-4	0
	randneig, dist$_0$=43.606, nq$_0$=6e2						**randneig**, dist$_0$=824.13, nq$_0$=2e4					
AD	0.01	5.9	0.3	40.398	3e-4	0	3.8	7.2	1.3	815.79	5e-4	0
PFM	0.03	3	0.2	40.418	6e-4	3	22	3.0	0	815.81	2e-6	15
Newt$_1$	0.16	61.9	5.2	40.398	1e-4	0	4167	1222	22	815.79	2e-6	0
SPGM	0.02	6.0	0.0	40.398	5e-4	0	9.4	7.2	0.4	815.79	2e-4	0

Table 2.3: Results for the random one factor problems with tol = 10^{-6}.

	t	**it**	**it$_{sd}$**	**dist**	**nq**	**v**	**t**	**it**	**it$_{sd}$**	**dist**	**nq**	**v**
	$n = 100$						$n = 2000$					
	corrand, dist$_0$=5.6646, nq$_0$=8e-2						**corrand**, dist$_0$=26.006, nq$_0$=5e-3					
AD	0.72	393	188	5.6642	9e-7	0	3938	7282	1653	26.006	9e-7	0
PFM	0.32	31	13	5.6642	8e-7	0	827	18	5.4	26.006	8e-7	0
Newt$_1$	0.02	7.2	2.5	5.6643	2e-8	0	36	5.0	1.6	26.006	6e-7	0
SPGM	0.22	128	44	5.6642	6e-7	0	638	760	546	26.006	8e-7	0
	corkfac, dist$_0$=0.3632, nq$_0$=6e0						**corkfac**, dist$_0$=0.3718,nq$_0$=3e1					
AD	0.02	9.8	0.5	2.73e-8	4e-7	0	6.1	9.2	2.4	8.73e-9	7e-7	0
PFM	0.06	5.6	0.5	3.19e-8	4e-7	0	21	3.2	0.4	3.91e-9	3e-7	0
Newt$_1$	0.01	3.0	0	1.8e-14	4e-13	0	15	2.0	0	1.5e-11	1e-9	0
SPGM	0.03	9.9	2.0	1.97e-8	2e-7	0	13	8.2	2.4	6.88e-9	3e-7	0
	randneig, dist$_0$=43.606, nq$_0$=6e2						**randneig**, dist$_0$=824.13, nq$_0$=2e4					
AD	0.02	8.6	0.5	40.398	4e-7	0	3.4	10.0	0	815.79	3e-7	0
PFM	0.06	5.0	0	40.418	2e-7	3	19.0	4.0	0	815.81	1e-9	15
Newt$_1$	0.09	61	5.7	40.398	1e-7	0	4171	1222	22	815.79	2e-6	0
SPGM	0.02	9.0	0	40.398	1e-7	0	11	9.6	0.5	815.79	2e-7	0

Table 2.4: Results for the one factor problem for cor1399 with tol = 10^{-3} and tol = 10^{-6}.

	t	**it**	**dist**	**nq**	**v**	**t**	**it**	**dist**	**nq**	**v**
	tol = 10^{-3}					tol = 10^{-6}				
	cor1399, dist$_0$=118.7753, nq$_0$=9e0									
AD	1.08	6.0	118.7752	2e-4	0	1.80	10.0	118.7752	5e-7	0
PFM	0.96	2.0	118.7752	6e-5	0	1.31	3.0	118.7752	2e-7	0
Newt$_1$	8.16	2.0	118.7752	5e-10	0	8.16	2.0	118.7752	5e-10	0
SPGM	4.83	7.0	118.7752	2e-5	0	5.67	10.0	118.7752	9e-7	0

but is the least efficient method for the corrand matrices with tol $= 10^{-6}$. Turning to the problem size, for tol $= 10^{-3}$ an increased n gives a bigger time advantage of AD over the other two methods, which is due to the remarkably low number of approximately $4n^2$ operations taken by AD for each iteration, compared with the Newton method Newt$_1$, which requires $O(n^3)$ operations. Finally, the efficiency of the methods depends on the matrix type, as can be seen for $n = 2000$ in Table 2.3, where in execution time the first three methods rank exactly in the reverse order for the corrand matrices compared with the randneig matrices. For the latter matrices, many steps appear to be required to approach the region of quadratic convergence for the Newton method.

- Interestingly, PFM, for which we do not have a convergence guarantee, shows robust behavior in terms of the required number of iterations and is clearly the best method on the cor1399 matrix. It satisfies the stopping criterion in these tests in a few iterations for every problem instance. However, we found that for small problem sizes PFM can show very poor convergence, as illustrated by the matrix

$$A = \begin{bmatrix} 1.0000 & 1.0669 & -1.0604 & 0.4903 & 0.9747 \\ 1.0669 & 1.0000 & 3.2777 & 0.3914 & 1.0883 \\ -1.0604 & 3.2777 & 1.0000 & 1.1075 & 0.8823 \\ 0.4903 & 0.3914 & 1.1075 & 1.0000 & 1.0431 \\ 0.9747 & 1.0883 & 0.8823 & 1.0431 & 1.0000 \end{bmatrix}. \tag{2.30}$$

  For the corresponding two factor problem PFM requires 11,415,465 iterations to satisfy the stopping criterion (2.28) with tol $= 10^{-3}$. This matrix was found after just 22 function evaluations using the implementation `mdsmax` [63] of the multidirectional search method of Torczon [131] to maximize the number of iterations required by PFM. This is in contrast to maximizing the iterations taken by SPGM for a two factor problem with the same problem size, yielding after 2000 function evaluations in `mdsmax` a matrix requiring only 118 iterations.

2.6.2 Choice of Starting Matrix, and Performance as k Varies

Now we present an experiment that compares different choices of starting matrix and also investigates the effects on algorithm performance of increasing k. Anticipating the results of the next subsection, we concentrate on the SPGM method. We consider four choices of starting matrix.

- Rank1mod: The matrix obtained from one iteration of the AD method starting with the rank one matrix defined at the start of Section 2.5. The reason for using the AD method in this way is that the rank one matrix alone is prone to yielding no descent for $k > 1$.

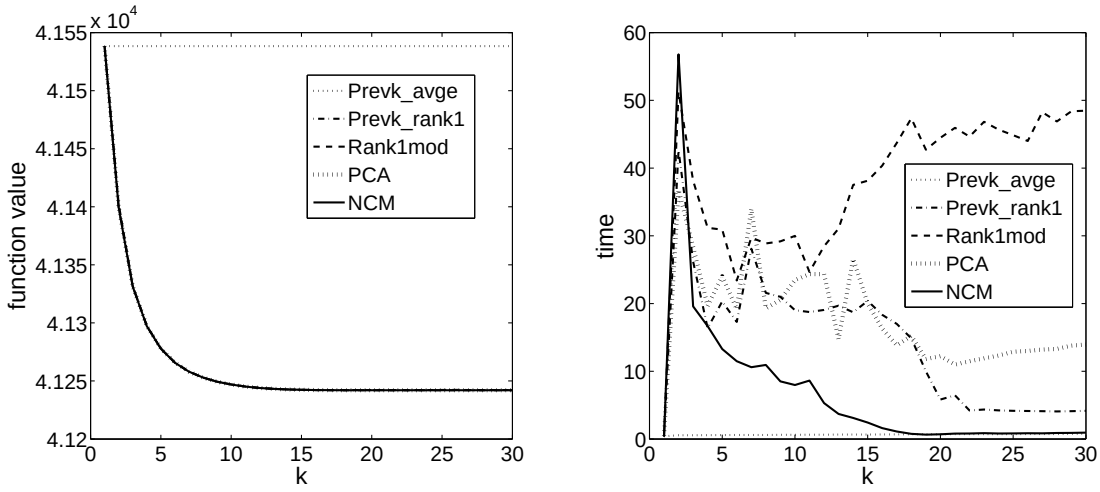

Figure 2.1: Comparison of different starting values for matrices of type randneig: k against final objective function value (left) and time (right).

- PCA: This rank r matrix, where r is a parameter, is obtained by "modified principal component analysis" as described, for example, in [108]. Let $A = Q\Lambda Q^T$ be a spectral decomposition with $\Lambda = \mathrm{diag}(\lambda_i)$ and $\lambda_n \geq \lambda_{n-1} \geq \cdots \geq \lambda_1$. The starting matrix is $X_0 = DQ\Lambda^{1/2}\left[\begin{smallmatrix} I_r \\ 0 \end{smallmatrix}\right] \in \mathbb{R}^{n\times r}$, where the diagonal matrix D is chosen such that every row of X_0 is of unit 2-norm (except that any zero row is replaced by $[1, 0, \ldots, 0]^T$).

- NCM: The nearest correlation matrix, computed using a preconditioned Newton method [23], [110]. This choice of starting matrix is suggested in [87].

- Prevk_rank1 and Prevk_avge: These choices are applicable only when we solve the problem for $k = 1, 2, \ldots$ in turn. We use the solution X_{k-1} of the $k-1$ factor problem as our starting matrix for the k factor problem by appending an extra column. For Prevk_rank1, the extra column is that given by Rank1mod for $k = 1$ applied to the matrix $A \leftarrow A - X_{k-1}X_{k-1} + \mathrm{diag}(X_{k-1}X_{k-1})$; for Prevk_avge, the last column is obtained as the averaged values of each row of X_{k-1}. Where necessary, the resulting matrix is projected onto the feasible set.

With $n = 500$, we took the matrix expij and 10 randomly generated matrices of type randneig and ran SPGM with each of the starting matrices, for a number of factors k ranging from 1 to 280 for expij and from 1 to 30 for randneig. Figures 2.1 and 2.2 show the results for randneig (averaged over the 10 matrices) and expij, respectively. The tolerance is 10^{-3} and the times shown include the time for computing the starting matrix, except in the case of Prevk_rank1 and Prevk_avge.

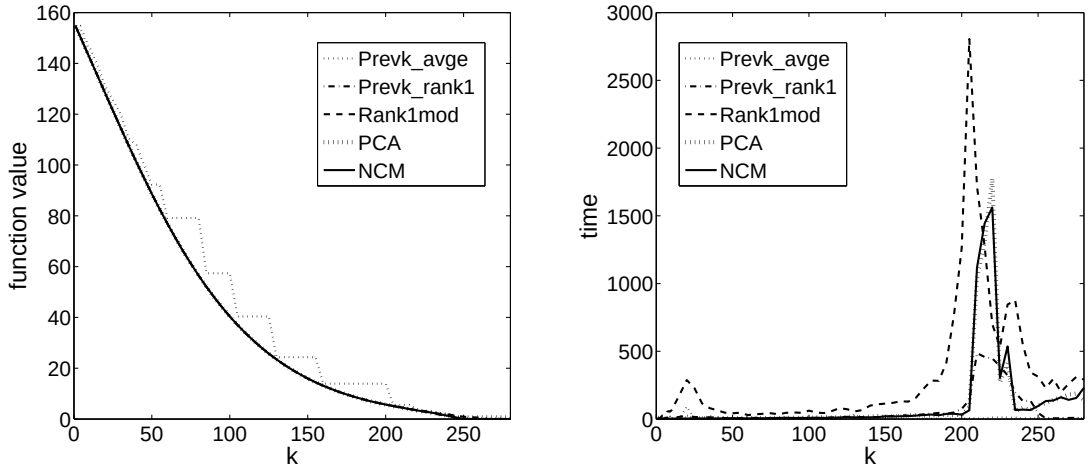

Figure 2.2: Comparison of different starting values for matrices of type expij: k against final objective function value (left) and time (right).

For randneig, Prevk_avge yields a larger final function value than the other starting matrices, and one that does not decay with k. The best of the five starting matrices for $k > 1$ in terms of run time and achieved minimum is clearly NCM; interestingly, the cost of computing it is relatively small. For $k = 1$ the Rank1mod matrix is as good a starting matrix as NCM and is less expensive to compute.

The time to solution as a function of k clearly depends greatly on the type of matrix. These two examples also indicate that the minimum may quickly level off as k increases (randneig) or may steadily decrease with k (expij).

2.6.3 Test Results for $k > 1$

We now repeat the tests from Section 2.6.1 with values of k greater than 1. The starting matrix was NCM in every case. We averaged the results over 10 instances of each of the three classes of random matrices for $n = 1000$ and $k = 2, 6$ and summarize the results in Table 2.5 for tol $= 10^{-3}$ and Table 2.6 for tol $= 10^{-6}$. We make several comments.

- The results for SQP are omitted from the tables because this method was not competitive in cost, although it did correctly solve each problem. In every case it was at least an order of magnitude slower than SPGM, and was about 2000 times slower on the corkfac matrices.

- As for $k = 1$, the values of v in (2.29) are all zero except for the randneig problems, where these values for the methods disregarding the constraints (2.3) (namely, AD, PFM, Newt$_2$) are significantly greater than the convergence tolerance. For AD, therefore, projecting the components of x onto $[-1, 1]$ does not

Table 2.5: Results for the random k factor problems with tol = 10^{-3}.

	t	**it**	**it$_{sd}$**	**dist**	**nq**	**v**	**t**	**it**	**it$_{sd}$**	**dist**	**nq**	**v**
	$k = 2$						$k = 6$					
	corrand, dist$_0$=18.29, nq$_0$=7.93						**corrand**, dist$_0$=18.29, nq$_0$=13.6					
AD	17	75	42	18.24	9e-4	0	95	114	60	18.13	9e-4	0
PFM	13	3.1	2.8	18.24	6e-4	0	8.2	3.2	0.6	18.13	5e-4	0
Newt$_2$	11	9	2	18.24	7e-4	0	19	9	2.3	18.13	3e-4	0
SPGM	4	39	43	18.24	8e-4	0	4.6	45	19	18.13	8e-4	0
	corkfac, dist$_0$=8.54e-1, nq$_0$=41.5						**corkfac**, dist$_0$=1.57, nq$_0$=46.2					
AD	1.6	7	0	1.7e-5	7e-4	0	5.8	7	0	3.3e-5	8e-4	0
PFM	0.9	2	0	1.3e-5	4e-4	0	2.6	3	0	1.0e-6	2e-5	0
Newt$_2$	2.0	2	0.6	4.9e-6	2e-4	0	3.1	3.9	0.3	1.6e-5	4e-4	0
SPGM	1.6	7	0	1.2e-5	3e-4	0	1.6	8	0.7	2.9e-5	5e-4	0
	randneig, dist$_0$=408.4, nq$_0$=4.2e-1						**randneig**, dist$_0$=408.0, nq$_0$=2.8e-1					
AD	101	431	156	408.7	9e-4	21.8	2.4e4	2.9e4	4.8e4	421.0	1e-3	121
PFM	4.2	5.0	0.9	408.7	2e-4	30.9	6.9	7.4	2.3	420.9	6e-4	127
Newt$_2$	28	14	3.8	408.7	4e-4	30.9	121	28	10	420.9	3e-4	127
SPGM	161	1270	638	407.6	8e-4	0	71	783	447	407.3	9e-4	0

Table 2.6: Results for the random k factor problems with tol = 10^{-6}.

	t	**it**	**it$_{sd}$**	**dist**	**nq**	**v**	**t**	**it**	**it$_{sd}$**	**dist**	**nq**	**v**
	$k = 2$						$k = 6$					
	corrand, dist$_0$=18.29, nq$_0$=7.9						**corrand**, dist$_0$=18.29, nq$_0$=13.6					
AD	1072	4540	4465	18.24	1e-6	0	1657	1982	1740	18.13	1e-6	0
PFM	127	24	20	18.24	8e-7	0	33	13	8.9	18.13	7e-7	0
Newt$_2$	61	20	14	18.24	4e-7	0	49	18	9	18.13	7e-7	0
SPGM	52	507	513	18.24	8e-7	0	30	312	230	18.13	8e-7	0
	corkfac, dist$_0$=8.54e-1, nq$_0$=41.5						**corkfac**, dist$_0$=1.57, nq$_0$=46.2					
AD	2.8	12	0	1.1e-8	4e-7	0	10.0	12	0	2.0e-8	4e-7	0
PFM	1.5	4	0	1.3e-9	4e-8	0	3.1	4	0	2.2e-8	4e-7	0
Newt$_2$	3.3	5	0.6	4.1e-9	1e-7	0	5.3	6.6	0.5	1.6e-8	3e-7	0
SPGM	2.0	10	1.3	8.0e-9	2e-7	0	2.1	13	0.7	1.7e-8	4e-7	0
SQP	788	44	12	1.4e-8	4e-7	0	3473	64	11	3.1e-8	5e-7	0
	randneig, dist$_0$=408.4, nq$_0$=4.2e-1						**randneig**, dist$_0$=408.0, nq$_0$=2.8e-1					
AD	195	826	318	408.7	9e-7	21	7e4	8.6e4	1.4e5	421.0	1e-6	121
PFM	7.3	8.6	2.1	408.7	4e-7	31	13	14	4.4	420.9	5e-7	127
Newt$_2$	59	36	9.5	408.7	6e-5	31	165	48	16.4	420.9	1e-4	127
SPGM	454	2882	2514	407.6	8e-7	0	295	3205	1576	407.3	9e-7	0

ensure feasibility. Moreover, the methods AD, PFM, and Newt_2 return a final iterate for $k = 6$ and randneig for which the mean function value is noticeably greater than the mean *initial* function value, caused by the projection onto the feasible set Ω at the end of the computation. This represents a serious failure of the minimization and shows the importance of properly treating the constraints within the method for the randneig problems.

- SPGM is clearly the preferred method in terms of efficiency combined with reliability.

2.7 Conclusions

We have obtained new theoretical understanding of the factor-structured nearest correlation matrix problem, particularly through explicit results for the one parameter and one factor cases. Our original motivation for studying this problem came from the credit basket securities application in [8] and the knowledge that the principal factors method has been used in the finance industry, despite the fact that it ignores the nonlinear problem constraints (2.25b). Our experiments have shown that this method, along with alternating directions and `fminunc`, often performs surprisingly well—partly because the constraints are often not active at the solution. However, all three methods can fail to solve the problem, as the randneig matrices show. Moreover, the principal factors method is not supported by any convergence theory. Our conclusion is that the spectral projected gradient method is the method of choice. It has guaranteed convergence, benefits from the ease with which iterates can be projected onto the convex constraint set, and because of the nonmonotone line search strategy can avoid narrow valleys at the beginning of the convergence process.

Chapter 3

Riemannian Geometry and Optimization

3.1 Introduction

This chapter gives an introduction to Riemannian manifolds and their geometric objects required to optimize objective functions over these sets. This analysis will lead to several minimizing algorithms that generate iterates on the corresponding manifolds. Note that this chapter is introductory and is based on [33], [84] for the general concept of a manifold and for the optimization part on [3], [49], [138] and [46]. We also direct the reader to these references to obtain further insight into this topic.

3.1.1 Motivation for Optimizing over Manifolds

Optimization over manifolds is a subject that has recently become more and more popular in the optimization community. One reason is certainly that constrained sets often satisfy the properties of a manifold—we will give the definition in Section 3.2.1—and therefore by means of geometric tools and algorithms that have now been developed, one can optimize an objective function within the constrained set. Hence, all the iterates that are generated in these algorithms are feasible. Therefore, especially for nonlinear constraints, these algorithms are more convenient to use and make it easier to deal with the constraints. Moreover, these algorithms can even perform better than state-of-the-art algorithms incorporating these constraints conventionally [1], [3], [4], [142].

For instance the set of matrices in $\mathbb{R}^{n\times p}$, $p \leq n$, with orthonormal columns forms a manifold as we will see later in this chapter. To incorporate the property that $Y^TY = I_p$ with $Y \in \mathbb{R}^{n\times p}$ one has to consider $p(p+1)/2$ nonlinear constraints.

Hence for p large the optimization will be expensive. However, by exploiting the fact that this constraint set is a manifold one can avoid increasing the number of free variables, for example by introducing Lagrange multipliers.

Optimization over manifolds also allows one to deal with abstract objects that is certainly another reason why these algorithms have become more attractive. One class of the abstract manifolds is *quotient* manifolds, and these are of significant importance. An example is the Grassmannian manifold that comprises the set of all subspaces with dimension p of a higher dimensional space with dimension n. As we will see in Chapter 5 it can be used to incorporate a rank constraint, which is generally hard to deal with.

3.1.2 Applications

Applications for optimization algorithms over manifolds are wide-ranging and more and more areas where their usage is of importance or unavoidable have become apparent. Examples can be found in image processing where segmentation and registration algorithms often rely on these optimization algorithms [40], [123]. Blind source separation is another application where efficient algorithms were proposed [115]. See also [7]. Low rank nearness problems are also candidates as they can be transformed into optimization problem over manifolds [93]. An extension to tensors was proposed in [71] where their algorithm achieves superlinear convergence. In Chapter 5 we will propose a low rank algorithm for linearly structured matrices where we optimize our objective function over the Grassmannian manifold. Other examples are the nearest weighted low rank correlation matrix algorithm proposed in [61] or the algorithm described in [138] for multilevel optimization of rank constraint matrix problems applied to find the low rank solution of Lyapunov equations. A popular application is also to compute the eigenvalues of a given matrix by minimizing the Rayleigh quotient over the sphere in $\mathbb{R}^n$, which is a manifold [3, Section 2].

3.1.3 Outline

If a manifold is smooth it can intuitively be considered as a structured set that can locally be described as a linear vector space, however it can globally be very different. One often compares it with a smooth surface in a higher dimensional space. In the next Section 3.2 we give the definition of a smooth manifold. Then we consider smooth functions on these manifolds and their properties in Section 3.3, before we investigate manifolds that are embedded in another space in Section 3.4. After we briefly describe the concept of quotient manifolds in Section 3.5 we clarify in the subsequent section when a smooth manifold is a Riemannian manifold. We introduce then the geometric

objects that allow us to optimize an objective function over these sets. In Section 3.8 particular attention is paid to two examples of smooth manifolds: the Stiefel and the Grassmannian manifold. We present numerical methods to optimize over these manifolds in Section 3.9 that we will use in later chapters. Our particular interest lies in the RBFGS-method [109], [49] where we propose a limited memory version of it.

3.2 Smooth Manifolds

3.2.1 Definition

Let $\mathcal{M}$ be a Hausdorff space with a countable basis where *Hausdorff space* refers to a topological space with the property that for any two different points x, y with $x \neq y$ there exists an open neighbourhood U_x of x such that $y \notin U_x$. This definition prevents that a convergent sequence in $\mathcal{M}$ can have several distinct limit points. Next we need the definition of a *chart*, allowing us to describe the set $\mathcal{M}$ locally as a d-dimensional Euclidean space. Note that the term *homeomorphic* function refers to a continuous bijective function with a continuous inverse.

Definition 3.2.1. Let φ be a homeomorphic function mapping from an open subset U of $\mathcal{M}$ onto an open subset in $\mathbb{R}^d$. The pair (U, φ) is called a *chart*.

Let us further assume that every point in $\mathcal{M}$ belongs to at least one chart domain. By introducing these charts we impose a structure on $\mathcal{M}$ that allows us to specify a coordinate system in the neighbourhood of every point in $\mathcal{M}$. However, if one point in $\mathcal{M}$ belongs to two domains of two charts $(U_\alpha, \varphi_\alpha)$ and (U_β, φ_β), i.e. these two domains overlap, the corresponding coordinate systems must be consistent. Therefore we need to introduce the concept of an *atlas* $\mathcal{A}$ of $\mathcal{M}$ into $\mathbb{R}^d$, see [3, p. 19].

Definition 3.2.2. An *atlas* $\mathcal{A}$ is a collection of charts $(U_\alpha, \varphi_\alpha)$ of $\mathcal{M}$ satisfying

- the union of all chart domains cover $\mathcal{M}$, i.e. $\mathcal{M} = \bigcup_\alpha U_\alpha$,
- for any two charts $(U_\alpha, \varphi_\alpha)$ and (U_β, φ_β) with an overlapping domain the function
$$\varphi_\beta \circ \varphi_\alpha^{-1} : \varphi_\alpha(U_\alpha \cap U_\beta) \mapsto \mathbb{R}^d$$
is *smooth*, i.e. it is in C^∞, the set of functions that are continuously differentiable for all degrees of differentiation.

Note that in this chapter we will refer to functions that are in C^∞ as *smooth* functions. Now we are ready to define the term of a *differentiable manifold*.

Definition 3.2.3. Let $\mathcal{A}$ be an atlas of $\mathcal{M}$. Let $\mathcal{A}^+$ be defined as the set of all charts (U, φ) such that $\mathcal{A} \cup \{(U, \varphi)\}$ is an atlas. $\mathcal{A}^+$ is also an atlas of $\mathcal{M}$ and is called the *maximal atlas*. The couple $(\mathcal{M}, \mathcal{A}^+)$ is then called a *d-dimensional smooth* or *differentiable manifold*. For simplicity we write only $\mathcal{M}$ for a smooth manifold in our later notation, although it is assumed that $\mathcal{M}$ comes with a maximal atlas.

3.2.2 Examples of Smooth Manifolds

Finite-Dimensional Vector Spaces

Let $\mathcal{V}$ be an n-dimensional vector space and let $(E_1, \ldots, E_n)$ be a basis for $\mathcal{V}$. Then the map $E : \mathbb{R}^n \mapsto \mathcal{V}$ with

$$E(x) = \sum_{i=1}^{n} x_i E_i$$

is clearly an isomorphism and $(\mathcal{V}, E^{-1})$ a chart. All charts built in this way are compatible and thus they form an atlas on $\mathcal{V}$. Consequently, $\mathcal{V}$ is an n-dimensional smooth manifold. It follows that also $\mathbb{R}^n$ and $\mathbb{R}^{n \times p}$ are smooth manifolds with dimension n and np, respectively.

Open Submanifolds

Let $\mathcal{M}$ be an n-dimensional smooth manifold and U be an open subset of $\mathcal{M}$. For every point $x \in U$ there must exist a chart (W, φ) of $\mathcal{M}$ with $x \in W$. Hence, by setting $V = W \cap U$ we obtain a chart $(V, \varphi\big|_V)$ of U. Let then $\mathcal{A}_U$ be the collection of all these charts for every $x \in U$, which is by construction an atlas of U. By [84, Lemma 1.10] there exists a unique maximal atlas containing $\mathcal{A}_U$ and thus U together with this maximal atlas is an n-dimensional smooth manifold that we call an *open submanifold* of $\mathcal{M}$.

3.3 Smooth Functions and Tangent Spaces

When optimizing a smooth function $f : \mathbb{R}^n \mapsto \mathbb{R}$ over $\mathbb{R}^n$ the usual procedure to find stationary points, i.e. points $x \in \mathbb{R}^n$ with $\nabla f(x) = 0$, is to generate a sequence with

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3.1}$$

starting from a given point x_0. If d_k is a descent direction and α_k is suitably chosen one can show that the sequence converges to a stationary point. However, if we consider to optimize over a smooth manifold $\mathcal{M}$, first we are facing the problem that the iterates x_{k+1} as defined in (3.1) might not be in $\mathcal{M}$, even if $x_0 \in \mathcal{M}$. Therefore

we need to generalize the definition of (3.1) to so called retraction. Similarly, we need a new definition for the gradient of f at a point $x_k \in \mathcal{M}$ and need to generalize the set of the vectors d_k at x_k that can be chosen for a descent direction. This set will be the tangent space, which is a vector space and is one of the geometric objects that needs to be defined in order to be able to optimize over manifolds. Introducing these geometric objects will provide a generalization of the conventional optimization tools to manifolds. We will start with the concept of smooth functions on manifolds and we will then introduce tangent spaces, which are fundamental in Riemannian optimization. Further geometric objects are considered in Section 3.7.

3.3.1 Smooth Functions

Let $\mathcal{M}$ and $\mathcal{N}$ be two smooth manifolds of dimension m and n and let $F : \mathcal{M} \mapsto \mathcal{N}$ a map between them.

Definition 3.3.1. The map F is called *smooth* if for every $x \in \mathcal{M}$ there exist charts (U, φ) with $x \in U$ and (V, ψ) with $F(x) \in V$ such that $F(U) \subset V$ and the map $\widehat{F} := \psi \circ F \circ \varphi^{-1} : \varphi(U) \mapsto \psi(V)$ is smooth.

This allows us to define a smooth objective function on $\mathcal{M}$. Let $\mathcal{F}(\mathcal{M})$ and $\mathcal{F}_x(\mathcal{M})$ denote the set of all smooth real-valued functions defined on $\mathcal{M}$ and only on a neighbourhood of $x \in \mathcal{M}$, respectively. Hence, $\mathcal{F}(\mathcal{M}) \subset \mathcal{F}_x(\mathcal{M})$.

Let the function $\widehat{F}$ be defined as in Definition 3.3.1. As $\widehat{F}$ is a function from $\mathbb{R}^m$ and $\mathbb{R}^n$ the Fréchet derivative $L_{\widehat{F}}(\varphi(x), \cdot) : \mathbb{R}^m \mapsto \mathbb{R}^n$ is well defined in a neighbourhood of U; see Appendix A.2 for a definition of the Fréchet derivative. Thus we can specify the *rank* of F at x as the dimension of the range of $L_{\widehat{F}}$ at $\varphi(x)$. Let $F : \mathcal{M} \mapsto \mathcal{N}$ be a function that has rank equal to n at every point of $\mathcal{M}$ be called a *submersion* and a point $y \in \mathcal{N}$ a *regular point* if F is of full rank at every point x with $x \in F^{-1}(y)$.

Now we can also introduce the concept of a *curve*, which is a smooth mapping $\gamma : \mathbb{R} \mapsto \mathcal{M}$ and can define when a manifold $\mathcal{M}$ is *connected*: for all $x, y \in \mathcal{M}$ there exists a curve γ in $\mathcal{M}$ on the interval $[a, b]$ such that $\gamma(a) = x$ and $\gamma(b) = y$. We are ready to define a tangent vector and tangent space.

3.3.2 Tangent Vectors and Spaces

If $\mathcal{M}$ is a smooth manifold representing a smooth surface in $\mathbb{R}^n$ then one can see the tangent space at a point $x \in \mathcal{M}$ as the tangent plane at x. However, for a general manifold one needs a more abstract definition, see [3, Definition 3.5.1].

Definition 3.3.2. Let γ be a smooth curve with $\gamma(t_0) = x$. Then the mapping $\xi_x : \mathcal{F}_x(\mathcal{M}) \mapsto \mathbb{R}$ with

$$\xi_x f := \left.\frac{d(f(\gamma(t)))}{dt}\right|_{t=t_0} \tag{3.2}$$

is called *tangent vector* to the curve γ at $t = t_0$ and x is the *foot* of ξ_x. We say that such a curve γ *realizes* the tangent vector ξ_x.

Note that the term $\left.\frac{d(f(\gamma(t)))}{dt}\right|_{t=t_0}$ is well defined as $f \circ \gamma$ is a function from $\mathbb{R}$ into $\mathbb{R}$ and the classical derivative can be applied. The concept of a tangent vector can also be introduced by *derivations* which are generalizations of the directional derivative and equivalent to the elements ξ_x; see [84, Chapter 3].

Now let the set of all *tangent vectors* at the point $x \in \mathcal{M}$ be denoted by the *tangent space* $T_x\mathcal{M}$. This set admits the structure of a vector space [3, Section 3.5.1] under the vector operation and scalar multiplication defined by

$$(a\xi_x + b\mu_x)f := a\xi_x f + b\mu_x f$$

for $f \in \mathcal{F}(\mathcal{M})$, $a, b \in \mathbb{R}$ and ξ_x, μ_x two tangent vectors at x. This property is important as we can now locally approximate the manifold by a vector space, making it possible to apply locally our conventional optimization tools.

Let $F : \mathcal{M} \mapsto \mathcal{N}$ be a smooth function between two manifolds $\mathcal{M}$, $\mathcal{N}$. Then we can also consider to map a tangent vector ξ_x of a tangent space at a point x in $\mathcal{M}$ into the tangent space $T_{F(x)}\mathcal{N}$. This is realized by a *differential*.

Definition 3.3.3. The mapping $DF(x) : T_x\mathcal{M} \mapsto T_{F(x)}\mathcal{N}$ with $\xi \mapsto DF(x)[\xi]$ where $DF(x)[\xi]$ is a map from $\mathcal{F}_{F(x)}(\mathcal{N})$ into $\mathbb{R}$ with

$$(DF(x)[\xi_x])f := \xi_x(f \circ F)$$

and $f \in \mathcal{F}_{F(x)}(\mathcal{N})$ is called the *differential* of F at x.

The set $T\mathcal{M} := \bigcup_{x\in\mathcal{M}} T_x\mathcal{M}$ is called a *tangent bundle* and for later use we also define a *vector field* as a smooth mapping $\xi : \mathcal{M} \mapsto T\mathcal{M}$ with $x \mapsto \xi_x \in T_x\mathcal{M}$. Vector fields can also be defined on curves γ by assigning to each t in the domain of γ a tangent vector in $T_{\gamma(t)}\mathcal{M}$. If this tangent vector is realized by the curve γ at $\gamma(t)$ we call the corresponding mapping $\dot{\gamma} : t \mapsto \dot{\gamma}(t)$ the *velocity vector field*.

3.4 Embedded Submanifolds

3.4.1 Recognizing Embedded Submanifolds

To show that a set is a smooth manifold one needs to find a maximal atlas associated with this set, which can often be cumbersome. Fortunately, there is tool that

identifies the preimage of a regular value associated with a smooth function as a smooth manifold with the additional property that it is embedded in the domain of this function. Such *embedded manifolds* are defined as follows [3, Proposition 3.3.2].

Definition 3.4.1. Let $\mathcal{N}$ be a subset of an n-dimensional smooth manifold $\mathcal{M}$. If for every point $x \in \mathcal{N}$ there exists a chart (U, φ) of $\mathcal{M}$ such that

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

for $d < n$ then $\mathcal{N}$ is called a d-dimensional *embedded submanifold* of $\mathcal{M}$.

Now we can state the theorem that gives us a tool to identify embedded submanifolds.

Theorem 3.4.2. *[3, Proposition 3.3.3] Let $F : \mathcal{M} \mapsto \mathcal{N}$ be a smooth mapping between two manifolds with dimension n and m respectively. If x is a regular value of F then the set $F^{-1}(x)$ is a closed embedded submanifold of $\mathcal{M}$ with the dimension $n - m$.*

3.4.2 Manifolds Embedded in Euclidean Space

Manifolds embedded in a Euclidean space (a vector space with an inner product) play an important role in the optimization over manifolds. One reason is that the tangent vectors at a point on the manifold reduce then to the classical directional derivatives and thus the tangent space can be identified with a subspace of the Euclidean space. Let $\mathcal{M}$ be an embedded submanifold of a Euclidean space $\mathcal{E}$. Note that we will use $\mathbb{R}^{n\times p}$ as our Euclidean space later. Further, let γ be a curve in $\mathcal{M}$ and $\gamma(t_0) = x \in \mathcal{M}$ with $t_0 \in \mathbb{R}$ and ξ_x a tangent vector at x realized by the curve γ. As $\mathcal{M}$ is an embedded submanifold of a Euclidean space,

$$\gamma'(t_0) = \lim_{t\to t_0} \frac{\gamma(t) - \gamma(t_0)}{t - t_0} \in \mathcal{E}$$

is well defined. Similarly, for a function $f \in \mathcal{F}_x(\mathcal{M})$ defined on a neighbourhood U of x in $\mathcal{E}$, the classical directional derivative of f

$$Df(x)[z] = \lim_{t\to 0} \frac{f(x + tz) - f(x)}{t}$$

for $z \in \mathcal{E}$ is also well defined.

Let $\widehat{f}$ be the restriction of f to $U \cap \mathcal{M}$. Then the tangent vector ξ_x is related to $\gamma'(t_0)$ as

$$\xi_x \widehat{f} = \frac{d}{dt}\widehat{f}(\gamma(t))\bigg|_{t=t_0} = \frac{d}{dt} f(\gamma(t))\bigg|_{t=t_0} = Df(x)[\gamma'(t_0)],$$

being a one-to-one correspondence. Hence, we have a natural identification of $T_x\mathcal{M}$ with the set

$$\{\gamma'(t_0) : \gamma \text{ curve in } \mathcal{M}, \gamma(t_0) = x\},$$

which is a linear subspace of $\mathcal{E}$ and gives an alternative representation of $T_x\mathcal{M}$.

Note if $\mathcal{M}$ is a Euclidean space then the above derivations imply $T_x\mathcal{M} \simeq \mathcal{M}$. Hence, for a function $F : \mathcal{M} \mapsto \mathcal{N}$ with $\mathcal{M}$ and $\mathcal{N}$ smooth manifolds that are Euclidean spaces the differential DF in Definition 3.3.3 reduces to the Fréchet derivative.

3.5 Quotient Manifolds

3.5.1 Definition

Let $\mathcal{M}$ be a smooth manifold and $\sim$ be an equivalence relation on $\mathcal{M}$.

Definition 3.5.1. [3, Section 3.4] For $x \in \mathcal{M}$ the set

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

is called the *equivalence class*, which obviously contains x. Then the set of all equivalent classes

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

is the *quotient* of $\mathcal{M}$ by $\sim$ and $\mathcal{M}$ is called the *total space* of $\mathcal{M}/\sim$. The corresponding mapping $\pi : \mathcal{M} \mapsto \mathcal{M}/\sim$ with $\pi(x) = [x]$ is denoted by the *canonical projection*.

Under some suitable conditions [3, Proposition 3.4.1] and [3, Proposition 3.4.2] the quotient $\mathcal{M}/\sim$ admits a unique maximal atlas $\mathcal{A}^+$ such that $(\mathcal{M}/\sim, \mathcal{A}^+)$ is a smooth manifold called the *quotient manifold*.

Lemma 3.5.2. *Let π be the canonical projection of a quotient manifold $\mathcal{M}/\sim$ and* $\dim(\mathcal{M}/\sim) < \dim(\mathcal{M})$. *Then the set $\pi^{-1}(\pi(x))$ is an embedded submanifold of $\mathcal{M}$ with the dimension* $\dim(\mathcal{M}) - \dim(\mathcal{M}/\sim)$.

Proof. From the definition of π it follows that the canonical projection π is a submersion. Then Theorem 3.4.2 implies the result. □

3.5.2 Smooth Functions

Optimizing over a quotient manifold $\mathcal{Q} := \mathcal{M}/\sim$ is of interest if the corresponding function $f \in \mathcal{F}(\mathcal{M})$ is invariant under the equivalence relation $\sim$ and in this case one would like to exploit this property, which is that $f(x_1) = f(x_2)$ for $x_1 \sim x_2$ with $x_1, x_2 \in \mathcal{M}$. If f has this property it induces a unique function $\widetilde{f}$ on $\mathcal{Q}$ with $f = \widetilde{f} \circ \pi$. By [3, Proposition 3.4.5] $\widetilde{f}$ is smooth on $\mathcal{Q}$ iff $\widetilde{f} \circ \pi$ is a smooth function on $\mathcal{M}$.

3.5.3 Tangent Space

Let $\mathcal{M}$ be a smooth manifold and $\mathcal{Q} := \mathcal{M}/\sim$ be a quotient manifold of $\mathcal{M}$. Further let x be an element of $\mathcal{M}$ and $y = \pi(x)$ the corresponding equivalence class. Then by Lemma 3.5.2 the set $\pi^{-1}(y)$ is an embedded submanifold of $\mathcal{M}$ whose tangent space $V_x = T_x\pi^{-1}(y)$ at x is called the *vertical space* at x. The subspace H_x of $T_x\mathcal{M}$ complementary to V_x is called the *horizontal space* and thus, it holds that $T_x\mathcal{M} = V_x \oplus H_x$. Moreover, let μ_y be an element of $T_y\mathcal{Q}$ then by [3, Section 3.5.8] there exists a unique element $\mu_x^h \in H_x$ that satisfies

$$D\pi(x)[\mu_x^h] = \mu_y.$$

This element is called the *horizontal lift*. As $\dim(\pi^{-1}(y)) = \dim(\mathcal{M}) - \dim(\mathcal{M}/\sim)$ we have $\dim(H_x) = \dim(\mathcal{M}) - \dim(\pi^{-1}(y)) = \dim(\mathcal{M}/\sim)$ where $\dim(\mathcal{A})$ is the dimension of the space $\mathcal{A}$. Hence, the horizontal space H_x is equivalent to the tangent space of $\mathcal{M}/\sim$ at y. Therefore we can define a bijective function that maps an element in $T_y\mathcal{Q}$ to the corresponding element in H_x and we denote this map by $\tau_x : T_{\pi(x)}\mathcal{Q} \mapsto H_x$.

3.5.4 Quotient Manifolds Embedded in Euclidean Space

Similarly to embedded submanifolds in a Euclidean space, if $\mathcal{Q}$ is a quotient manifold of an embedded submanifold in a Euclidean space $\mathcal{E}$, there is an alternative representation of the tangent vectors $\mu_y \in T_x\mathcal{Q}$. That is the horizontal lift, which, by the derivation in Section 3.4.2, is being represented by an element in $\mathcal{E}$.

3.6 Riemannian Manifolds

3.6.1 Riemannian Metric and Distance

Essential for the optimization is to measure distances on manifolds e.g. in order to define the steepest descent of a function on manifolds. Therefore we introduce the notion of length that applies to tangent vectors and endow every tangent space $T_x\mathcal{M}$ for $\mathcal{M}$ a smooth manifold with an inner product $\langle\cdot,\cdot\rangle_x$, inducing a norm on $T_x\mathcal{M}$ by

$$\|\xi_x\|_x = \sqrt{\langle\xi_x, \xi_x\rangle_x}.$$

If the inner product $\langle\cdot,\cdot\rangle_x$ is smoothly varying with $x \in \mathcal{M}$ it is called a *Riemannian metric*.

Definition 3.6.1. If a smooth manifold $\mathcal{M}$ is endowed with a Riemannian metric with such an inner product then we call $\mathcal{M}$ a *Riemannian manifold*.

As we will see later these manifolds allow the generalization of the conventional optimization tools. Moreover, all manifolds that we will deal with later have the properties of a Riemannian manifold.

Let $\gamma : [a, b] \mapsto \mathcal{M}$ be a curve. Then the length of γ is defined by

$$L(\gamma) = \int_a^b \sqrt{\left\langle \xi_{\gamma(t)}, \xi_{\gamma(t)} \right\rangle_{\gamma(t)}} dt, \tag{3.3}$$

where $\xi_{\gamma(t)}$ is the tangent vector that is realized by the curve γ at $\gamma(t)$.

3.6.2 Riemannian Submanifold

Let $\mathcal{N}$ be a submanifold of a Riemannian manifold $\mathcal{M}$. By definition the tangent space $T_x\mathcal{N}$ is a subspace of $T_x\mathcal{M}$ for $x \in \mathcal{N}$. Therefore by restricting the metric of $\mathcal{M}$ to the tangent space $T_x\mathcal{N}$ we obtain a new metric $\langle \cdot, \cdot \rangle_x^n$ on $T_x\mathcal{N}$, i.e. we set

$$\langle \xi_x, \mu_x \rangle_x^n := \langle \xi_x, \mu_x \rangle_x .$$

We call the smooth submanifold $\mathcal{N}$ equipped with this metric *Riemannian submanifold*. Now we can also define the *normal space* $N_x\mathcal{N}$ at x to $T_x\mathcal{N}$ by

$$N_x\mathcal{N} := \{\xi_x \in T_x\mathcal{M} : \langle \xi_x, \mu_x \rangle_x = 0 \text{ for all } \mu_x \in T_x\mathcal{N}\} .$$

Thus, we have $T_x\mathcal{M} = N_x\mathcal{N} \oplus T_x\mathcal{N}$, allowing the definition of orthogonal projections of an element in $T_x\mathcal{M}$ onto the tangent space $T_x\mathcal{N}$ and $N_x\mathcal{N}$, respectively with

$$\Pi_x^t : T_x\mathcal{M} \mapsto T_x\mathcal{N} \quad \text{and} \quad \Pi_x^n : T_x\mathcal{M} \mapsto N_x\mathcal{N}.$$

3.6.3 Riemannian Quotient Manifold

The situation for quotient manifolds of a Riemannian manifold is similar. Therefore let $\mathcal{Q}$ be a quotient manifold of a Riemannian manifold $\mathcal{M}$ and π the canonical projection. Further let x be a point in $\mathcal{M}$, $y := \pi(x)$ and H_x the horizontal space at x. Recall from Section 3.5 that every $\mu_y \in T_y\mathcal{Q}$ can be represented by the horizontal lift $\xi_x \in H_x$. Hence, we obtain a new metric by restricting $\langle \cdot, \cdot \rangle_x$ to the horizontal space. That is

$$\langle \widehat{\mu}_y, \mu_y \rangle_y^n := \left\langle \widehat{\mu}_x^h, \mu_x^h \right\rangle_x \text{ for all } \widehat{\mu}_y, \mu_y \in T_y\mathcal{Q}$$

and $\widehat{\mu}_x^h = \tau_x(\widehat{\mu}_y)$ and $\mu_x^h = \tau_x(\mu_y)$ the corresponding horizontal lifts in H_x. If this metric is independent of the particular choice of x, i.e., the metric is constant for all elements at $\pi^{-1}(y)$, then we call $\mathcal{Q}$ endowed with this metric a *Riemannian quotient*

manifold. As $T_x\mathcal{M} = V_x \oplus H_x$ with V_x and H_x the vertical and horizontal space, respectively we can also define the orthogonal projectors onto these spaces by

$$\Pi_x^h : T_x\mathcal{M} \mapsto H_x \quad \text{and} \quad \Pi_x^v : T_x\mathcal{M} \mapsto V_x.$$

We have seen that for both embedded submanifolds $\mathcal{N}$ and quotient manifolds $\mathcal{Q}$ of a Riemannian manifold $\mathcal{M}$ we can inherit the Riemannian metric from $\mathcal{M}$ in a natural way. Therefore Riemannian submanifolds and Riemannian quotient manifolds are Riemannian manifolds.

3.7 Geometric Objects

3.7.1 The Gradient

Let $\mathcal{M}$ be a Riemannian manifold. As this manifold is equipped with an inner product we can define the gradient of a real valued function on $\mathcal{M}$.

Definition 3.7.1. The *gradient* $\operatorname{grad} f(x)$ of $f \in \mathcal{F}(\mathcal{M})$ is defined as the unique vector field with $\operatorname{grad} f(x)$ the unique tangent vector in $T_x\mathcal{M}$ at $x \in \mathcal{M}$ satisfying

$$\langle \operatorname{grad} f(x), \xi_x \rangle_x = Df(x)[\xi_x], \quad \forall \xi_x \in T_x\mathcal{M}. \tag{3.4}$$

This definition is well defined by the Riesz representation theorem. Further if we define the *direction of steepest descent* at $x \in \mathcal{M}$ of a function $f \in \mathcal{F}(\mathcal{M})$ as the element in $T_x\mathcal{M}$ that satisfies

$$\operatorname*{argmin}_{\xi_x \in T_x\mathcal{M}, \|\xi_x\|_x = 1} Df(x)[\xi_x],$$

then it is clear from (3.4) that this element is $-\operatorname{grad} f(x)/\|\operatorname{grad} f(x)\|_x$.

3.7.2 Levi-Civita Connection

With the gradient of a real valued function $f \in \mathcal{F}(\mathcal{M})$ on a manifold $\mathcal{M}$ one obtains first-order information about the function. However, often, especially to develop optimization algorithms with higher order of convergence, second-order information about the function is required. In Euclidean space $\mathcal{E}$ one uses the directional derivative of a vector field ξ, which could be e.g. the gradient of f, defined as

$$D\xi(x)[\mu(x)] = \lim_{t \to 0} \frac{\xi(x + t\mu(x)) - \xi(x)}{t} \tag{3.5}$$

and obtains again a vector field in $\mathcal{E}$. However, this is generally not true for manifolds. Therefore, we introduce a generalization of the directional derivative of a vector field on $\mathcal{M}$ which is called the *affine connection*.

Definition 3.7.2. Let $\mathcal{X}(\mathcal{M})$ be the set of all vector fields on $\mathcal{M}$. Then the mapping

$$\nabla : \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \mapsto \mathcal{X}(\mathcal{M})$$

that satisfies for $f, g \in \mathcal{F}(\mathcal{M})$, $\xi, \mu, \nu \in \mathcal{X}(\mathcal{M})$ and $a, b \in \mathbb{R}$

- $\nabla_{f\mu+g\nu}\xi = f\nabla_\mu\xi + g\nabla_\nu\xi$,
- $\nabla_\mu(a\xi + b\nu) = a\nabla_\mu\xi + b\nabla_\mu\nu$ and
- $\nabla_\mu(f\xi) = (\mu f)\xi + f\nabla_\mu\xi$

is called an *affine connection*.

There are infinitely many choices for an affine connection on smooth manifolds. However, it is desirable to have a unique connection that also reduces to the conventional directional derivative as defined in (3.5) in the Euclidean space. The *Levi-Civita* connection satisfies these properties and is therefore a popular choice for an affine connection.

Theorem 3.7.3. *[3, Theorem 5.3.1] Let $\mathcal{M}$ be a Riemannian manifold and $\langle\cdot,\cdot\rangle$ the corresponding metric on $\mathcal{M}$. There exists a unique affine connection ∇ that satisfies for $\xi, \nu, \mu \in \mathcal{X}(\mathcal{M})$*

- $\nabla_\mu\xi - \nabla_\xi\mu = \mu\xi - \xi\mu$ *and*
- $\xi\langle\mu,\nu\rangle = \langle\nabla_\xi\mu,\nu\rangle + \langle\mu,\nabla_\xi\nu\rangle$.

The affine connection that is uniquely defined by this theorem is called *Levi-Civita* connection. Note that $\mu\xi - \xi\mu$ is well defined as it is by [84, Lemma 4.12] a vector field.

Levi-Civita Connection on Embedded Riemannian Submanifolds

The next theorem relates the Levi-Civita connection on a Riemannian manifold $\mathcal{M}$ to the one on an embedded Riemannian submanifold. This result will be particularly useful if $\mathcal{M}$ is a Euclidean space as in this case the directional derivative coincides with the *Levi-Civita* connection.

Theorem 3.7.4. *[3, Proposition 5.3.2] Let $\mathcal{M}$ be a Riemannian manifold and $\mathcal{N}$ an embedded Riemannian submanifold of $\mathcal{M}$. Further let $\nabla^{\mathcal{M}}$ and $\nabla^{\mathcal{N}}$ be the Levi-Civita connections on $\mathcal{M}$ and $\mathcal{N}$, respectively. Then*

$$\nabla^{\mathcal{N}}_{\mu(x)}\xi(x) = \Pi^t_x(\nabla^{\mathcal{M}}_{\mu(x)}\xi(x)) \tag{3.6}$$

for all $\mu(x), \xi(x) \in T_x\mathcal{N}$ and $x \in \mathcal{N}$.

Levi-Civita Connection on Riemannian Quotient Manifolds

A similar result can be obtained for quotient manifolds.

Theorem 3.7.5. *[3, Proposition 5.3.3] Let $\mathcal{M}$ be a Riemannian manifold and $\mathcal{Q} := \mathcal{M}/\sim$ a Riemannian quotient manifold. Further let $\nabla^{\mathcal{M}}$ and $\nabla^{\mathcal{Q}}$ be the Levi-Civita connections on $\mathcal{M}$ and $\mathcal{Q}$, respectively. Then*

$$\tau_x \left(\nabla^{\mathcal{Q}}_{\mu(y)} \xi(y) \right) = \Pi^h_x \left(\nabla^{\mathcal{M}}_{\tau_x(\mu(y))} \tau_x(\xi(y)) \right) \tag{3.7}$$

where x is any element of $\pi^{-1}(y)$, $y \in \mathcal{Q}$ and $\mu(y), \xi(y) \in T_y\mathcal{Q}$.

3.7.3 Geodesics and Retractions

Geodesics

In Euclidean space $\mathcal{E}$ the optimization is usually performed along straight lines using a line search technique, i.e. we start from an iterate x_k and try to minimize a real function f over a scalar α_k along a descent direction d_k to find a new iterate

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3.8}$$

where α_k is chosen such that the descent in f is sufficiently large. Hence the optimization is carried out along straight lines. As (3.8) is not defined on a manifold the notion of straight lines needs to be generalized by using the property that characterizes straight lines, that is

$$\frac{d^2}{dt^2}\gamma(t) = 0 \text{ for all } t$$

and γ a curve in $\mathcal{E}$.

Definition 3.7.6. Let $\mathcal{M}$ be a Riemannian manifold and $\gamma : (a, b) \mapsto \mathcal{M}$ be a curve and let ∇ denote the Levi-Civita connection. If this curve satisfies

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

for all $t \in (a, b)$ then the curve γ is called a *geodesic*.

The next lemma ensures its existence and uniqueness, see [3, Section 5.4].

Lemma 3.7.7. *For any tangent vector $\xi_x \in T_x\mathcal{M}$ for $x \in \mathcal{M}$ there exists an interval I about zero and a geodesic $\gamma(t)$ with $\gamma(0) = x$ and $\dot\gamma(0) = \xi_x$ that is unique.*

Note that the geodesic can also be described by the curve of shortest length connecting two points on the manifold [46] where the length of the curve between two points is defined by (3.3).

Retractions

As computing the geodesics is rather costly for common manifolds like the Stiefel or Grassmannian manifold the usage of approximations of these geodesics are inevitable to develop efficient algorithms. These approximations are called *retractions*. We only look at first-order retractions that approximate the geodesic up to first order.

Definition 3.7.8. [3, Definition 4.1.1], [138, Definition 2.31] Let $\mathcal{M}$ be a Riemannian manifold. Let R be a smooth mapping with $R : T\mathcal{M} \mapsto \mathcal{M}$ and R_x the restriction to $T_x\mathcal{M}$ for $x \in \mathcal{M}$. If R satisfies

- $R_x(0_x) = x$ where 0_x is the zero element of $T_x\mathcal{M}$ and
- for every tangent vector $\xi_x \in T_x\mathcal{M}$ the curve $\gamma_{\xi_x} : t \mapsto R_x(t\xi_x)$ realizes ξ_x at 0

then R is called a *retraction*.

3.7.4 The Riemannian Hessian

The above definition of the Levi-Civita connection ∇ also allows us to define the *Riemannian Hessian* of a real-valued function $f \in \mathcal{F}(\mathcal{M})$.

Definition 3.7.9. [3, Definition 5.5.1] Let $\mathcal{M}$ be a Riemannian manifold and ∇ the Levi-Civita connection. Then the *Riemannian Hessian* of $f \in \mathcal{F}(\mathcal{M})$ at $x \in \mathcal{M}$ is defined as the linear mapping $\operatorname{Hess} f(x) : T_x\mathcal{M} \mapsto T_x\mathcal{M}$ with

$$\operatorname{Hess} f(x)[\xi_x] = \nabla_{\xi_x} \operatorname{grad} f(x) \tag{3.9}$$

for $\xi_x \in T_x\mathcal{M}$.

Note that it holds that

$$\langle \operatorname{Hess} f(x)[\xi_x], \xi_x \rangle = \frac{d^2}{dt^2} f(\gamma(t))\bigg|_{t=0}$$

for $\gamma(t)$ the geodesic with $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi_x$, see [3, Proposition 5.5.4]. Hence, the Hessian operator $\operatorname{Hess} f(x)$ captures second-order information of f.

3.7.5 Vector Transport

For quasi-Newton methods like the BFGS-method where a closed form of the Hessian is not available, one needs to compare first-order information at different points. This first-order information correspond to tangent vectors on manifolds. Therefore we need to introduce the concept of vector transport, mapping a tangent vector ξ_x along a direction μ_x into the tangent space at $R_x(\mu_x)$, making it possible to compare ξ_x with tangent vectors in $T_{R_x(\mu_x)}\mathcal{M}$.

Vector Transport on Smooth Manifolds

Definition 3.7.10. [3, Definition 8.1.1] A smooth mapping $\mathcal{T} : T\mathcal{M} \times T\mathcal{M} \mapsto T\mathcal{M}$ written as $\mathcal{T}_{\mu_x}(\xi_x) := \mathcal{T}(\mu_x, \xi_x)$ is called a *vector transport* if is satisfies

- there exists a retraction R associated with $\mathcal{T}$ such that the following diagram commutes

$$\begin{array}{ccc} (\mu_x, \xi_x) & \xrightarrow{\mathcal{T}} & \mathcal{T}_{\mu_x}(\xi_x) \\ \downarrow & & \downarrow \varsigma \\ \mu_x & \xrightarrow{R} & \varsigma\left(\mathcal{T}_{\mu_x}(\xi_x)\right) \end{array}$$

  where $\varsigma(\xi_x)$ returns the foot for a tangent vector $\xi_x \in T\mathcal{M}$,

- $\mathcal{T}_{0_x}\xi_x = \xi_x$ for all $\xi_x \in T_x\mathcal{M}$ and
- $\mathcal{T}_{\mu_x}(a\xi_x + b\nu_x) = a\mathcal{T}_{\mu_x}(\xi_x) + b\mathcal{T}_{\mu_x}(\nu_x)$ for $a, b \in \mathbb{R}$ and $\mu_x, \xi_x, \nu_x \in T_x\mathcal{M}$.

Note that

$$\mathcal{T}_{\mu_x}\xi_x := \frac{d}{dt} R_x(\mu_x + t\xi_x)\Big|_{t=0}$$

defines a vector transport on a Riemannian manifold $\mathcal{M}$ endowed with a retraction R [3, Section 8.1.2].

Parallel Transport - An Isometric Vector Transport

Let ∇ be the Levi-Cevita connection on a Riemannian manifold $\mathcal{M}$ and ξ a vector field on a curve γ in $\mathcal{M}$ and let ν be a vector field realized by the curve γ. If $\nabla_\nu \xi = 0$ then ξ is called a *parallel vector field* on γ. For a tangent vector $\xi_{\gamma(a)} \in T_{\gamma(a)}\mathcal{M}$ for a in the domain of γ there exists a unique parallel vector field ξ with $\xi(\gamma(a)) = \xi_{\gamma(a)}$. Hence, we can define an operator $P_\gamma^{b \leftarrow a}$ that maps $\xi_{\gamma(a)}$ to $\xi(\gamma(b))$ for b in the domain of γ, see [3, p. 104]. This operator is called *parallel translation* along the curve γ. Note that by [3, Proposition 8.1.2] the parallel translation is a vector transport. Moreover, this vector transport preserves the metric [104, Lemma 3.20], which is an important feature when it comes to generalizing optimization routines to manifolds.

Vector Transport on Riemannian Submanifold

For $\mathcal{M}$ an embedded submanifold of a Euclidean space $\mathcal{E}$ endowed with a retraction R we can define the vector transport by

$$\mathcal{T}_{\mu_x}\xi_x := \Pi^t_{R_x(\mu_x)}\xi_x. \tag{3.10}$$

3.8 Examples of Riemannian Manifolds

In this section we introduce the geometric objects for two particular manifolds, the Stiefel manifold and Grassmannian manifold, providing us with the necessary tools to be able to optimize over these two manifolds.

3.8.1 The Stiefel Manifold

Definition of the Manifold

The *Stiefel manifold* is the set of matrices in $\mathbb{R}^{n\times p}$ with orthonormal columns and $p \leq n$, that is

$$\mathsf{St}(n,p) := \left\{Y \in \mathbb{R}^{n\times p} : Y^TY = I_p\right\}.$$

Often this set is also called *compact Stiefel manifold*.

Lemma 3.8.1. *$\mathsf{St}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n\times p}$ with dimension $np - p(p+1)/2$. See Definition 3.4.1 when a manifold is called embedded in $\mathbb{R}^{n\times p}$.*

Proof. Consider the function $F : \mathbb{R}^{n\times p} \mapsto \mathcal{S}^p$ with $F(Y) = Y^TY - I_p$. Then it is clear that $F^{-1}(0) = \mathsf{St}(n,p)$. Hence, by Theorem 3.4.2 it is enough to show that 0 is a regular point. We have that

$$DF(Y)[Z] = Z^TY + Y^TZ.$$

For $S \in \mathcal{S}^p$ arbitrary by choosing $Z = \frac{1}{2}YS$ we obtain $DF(Y)[Z] = S$. Thus F has full rank for all $Y \in \mathsf{St}(n,p)$ and consequently 0 is a regular point. ∎

Tangent Space

Let us now derive the tangent space of this manifold at a point $Y \in \mathsf{St}(n,p)$ and define a metric in these spaces. Let $Y(t)$ be a curve in $\mathsf{St}(n,p)$ with $Y(0) = Y$ then by differentiating the condition $Y^TY = I_p$ with respect to t at $t = 0$ we obtain

$$Y^TY'(0) + Y'(0)^TY = 0, \tag{3.11}$$

meaning that $Y^TY'(0)$ is skew-symmetric for every matrix $Y'(0)$ in the tangent space $T_Y\mathsf{St}(n,p)$ at Y. Moreover, as (3.11) imposes $p(p+1)/2$ constraints on $Y'(0) \in \mathbb{R}^{n\times p}$, the set of all matrices satisfying property (3.11) has the same dimension as $\mathsf{St}(n,p)$, which is $\dim \mathsf{St}(n,p) = np - p(p+1)/2$. Thus, (3.11) gives a defining property for the tangent space $T_Y\mathsf{St}(n,p)$.

Let $\mathcal{K}_p$ be the set of all skew-symmetric matrices in $\mathbb{R}^{p\times p}$. Then the set

$$N := \{YA + Y_\perp B : A \ \in \mathcal{K}_p, B \in \mathbb{R}^{(n-p)\times p}\} \tag{3.12}$$

describes fully the tangent space $T_Y\mathsf{St}(n,p)$ as it has the same dimension and every element of N satisfies (3.11). In (3.12) $Y_\perp$ denotes a matrix in $\mathbb{R}^{n\times(n-p)}$ that has orthonormal columns and are complementary to the columns of Y.

Now let this manifold be endowed with the metric $\langle A,B\rangle_Y := \text{trace}(B^TA)$ for A,B in the tangent space of $\mathsf{St}(n,p)$ at Y.

As the Stiefel manifold is an embedded submanifold of the Euclidean space $\mathbb{R}^{n\times p}$ endowed with the inner product $\langle A,B\rangle_Y := \text{trace}(B^TA)$ we can also define the projection of a vector in $\mathbb{R}^{n\times p}$ onto $T_Y\mathsf{St}(n,p)$ at a point Y. From (3.12) it is easy to check that the projection is defined by

$$\Pi_Y^t(Z) = Y\text{skew}(Y^TZ) + (I_n - YY^T)Z$$

where $\text{skew}(A)$ is the skew-symmetric part $(A-A^T)/2$ of $A\in\mathbb{R}^{p\times p}$.

From (3.12) we also obtain the normal space

$$N_Y\mathsf{St}(n,p) = \left\{Z\in\mathbb{R}^{n\times p} : Z = YS \text{ with } S\in\mathcal{S}^p\right\}$$

and the projection onto it

$$\Pi_Y^n(Z) = Y\text{sym}(Y^TZ)$$

with $\text{sym}(A)$ the symmetric part $(A+A^T)/2$ of $A\in\mathbb{R}^{p\times p}$.

The Gradient

Let $f\in\mathcal{F}(\mathsf{St}(n,p))$ and let the inner product be the Euclidean inner product as in the previous section. Then from Section 3.7.1 the gradient at Y is the element $\text{grad}\, f(Y)$ in $T_Y\mathsf{St}(n,p)$ satisfying

$$\langle \text{grad}\, f(Y), \xi_Y\rangle_Y = Df(Y)[\xi_Y] \tag{3.13}$$

for all $\xi_Y\in T_Y\mathsf{St}(n,p)$. Then it is easily verified that

$$\zeta_Y := \nabla f(Y) - Y\text{sym}\left(Y^T\nabla f(Y)\right) \tag{3.14}$$

satisfies (3.13) and lies in $T_Y\mathsf{St}(n,p)$ for all $Y\in\mathsf{St}(n,p)$. Therefore ζ_Y is the gradient at Y.

Geodesics

The formula for the geodesics is given in [46]. Let $Y\in\mathsf{St}(n,p)$ and $\xi_Y\in T_Y\mathsf{St}(n,p)$ a direction in the tangent space then

$$Y(t) = \begin{bmatrix} Y & \xi_Y\end{bmatrix} \exp\left(t\begin{bmatrix} Y^T\xi_Y & -\xi_Y^T\xi_Y \\ I_p & Y^T\xi_Y\end{bmatrix}\right)\begin{bmatrix} I_p \\ 0\end{bmatrix}\exp(-Y^T\xi_Y t)$$

is the geodesic along ξ_Y.

Retractions

Let us present the two most popular retractions on the Stiefel manifold [3, Example 4.1.3]. The first is based on the unitary polar factor of $Y + \xi_Y$, that is

$$R_Y(\xi_Y) = (Y + \xi_Y)(I_p + \xi_Y^T \xi_Y)^{-1/2}.$$

See [66, Chapter 8] for details of the polar decomposition. The other one is based on the Q-factor of the QR-decomposition, that is

$$R_Y(\xi_Y) = \mathrm{qf}(Y + \xi_Y)$$

where $\mathrm{qf}(A)$ is the Q-factor of $A \in \mathbb{R}^{n \times p}$ with A of full rank.

Vector Transport

To obtain a vector transport on the Stiefel manifold one can apply (3.10) as the projection onto the tangent space is known.

3.8.2 The Grassmannian Manifold

Definition

The next manifold that we are looking at more in detail is the Grassmannian manifold.

Definition 3.8.2. Let $\sim$ denote an equivalence relation defined on the Stiefel manifold with

$$X \sim Y \iff \mathrm{span}(X) = \mathrm{span}(Y).$$

for $X, Y \in \mathsf{St}(n,p)$. Then the quotient space $\mathsf{Gr}(n,p) := \mathsf{St}(n,p)/\sim$ is called the *Grassmannian manifold*. Note that the mapping $X \mapsto XQ$ for $X \in \mathsf{St}(n,p)$ and $Q \in \mathsf{O}(p)$ corresponds to all possible changes of the basis of $\mathrm{span}(X)$ where $\mathsf{O}(p)$ is the set of all orthogonal matrices in $\mathbb{R}^{p \times p}$. Therefore we can also write $\mathsf{Gr}(n,p) = \mathsf{St}(n,p)/\mathsf{O}(p)$ and $\mathsf{Gr}(n,p)$ is the collection of all equivalent classes

$$[X] := \{XQ : Q \in \mathsf{O}(p)\} \tag{3.15}$$

for $X \in \mathsf{St}(n,p)$.

The Grassmannian manifold $\mathsf{Gr}(n,p)$ corresponds to the set of all p-dimensional subspaces of $\mathbb{R}^n$. One element in $\mathsf{Gr}(n,p)$ is the collection of all possibles matrices with orthonormal columns that can be identified as orthonormal basis vectors, spanning the same subspace.

Lemma 3.8.3. *The quotient set $\mathsf{Gr}(n,p)$ admits a unique structure of quotient manifold and has dimension $p(n-p)$.*

Proof. As $\mathsf{Gr}(n,p) \simeq \mathbb{R}_*^{n\times p}/\mathbb{R}^{p\times p}$ we obtain the result by [3, Proposition 3.4.6] where $\mathbb{R}_*^{n\times p}$ is the set of matrices in $\mathbb{R}^{n\times p}$ with full rank. The dimension follows from [3, p. 32]. ∎

Tangent Space

As from the previous section the tangent space $T_X\mathsf{St}(n,p)$ of the total space $\mathsf{St}(n,p)$ of $\mathsf{Gr}(n,p)$ is already known we need to determine the horizontal and vertical space. From (3.12) and (3.15) it is clear that the vertical space is

$$V_X := \{XA : A \in \mathcal{K}_p\}.$$

Hence, the horizontal space at $X \in \mathsf{St}(n,p)$ is then

$$H_X := \left\{X_\perp B : B \in \mathbb{R}^{(n-p)\times p}\right\},$$

which is equivalent to the tangent space $T_{\pi(X)}\mathsf{Gr}(n,p)$ for π the canonical projection. The projection onto H_X is then clearly given by

$$\Pi_X^h(Z) = (I_n - XX^T)Z. \tag{3.16}$$

Let $T_{\pi(X)}\mathsf{Gr}(n,p)$ be endowed with the inner product $\left\langle \xi_{\pi(X)}, \nu_{\pi(X)}\right\rangle_{\pi(X)} := \operatorname{trace}\left(\nu_X^T\xi_X\right)$ for $\xi_{\pi(X)}, \nu_{\pi(X)} \in T_{\pi(X)}\mathsf{Gr}(n,p)$ and ξ_X, ν_X the corresponding elements in H_X. This is well defined as the corresponding elements do not depend on the specific choice of X in $\pi^{-1}(\pi(X))$. As the inner product does not depend on $\pi(X)$ we can just write $\left\langle \xi_{\pi(X)}, \nu_{\pi(X)}\right\rangle$. Therefore, together with the inner product $\left\langle \xi_{\pi(X)}, \nu_{\pi(X)}\right\rangle$, $\mathsf{Gr}(n,p)$ is a Riemannian quotient manifold.

The Gradient

Let $\widetilde{f} \in \mathcal{F}(\mathsf{Gr}(n,p))$ and the inner product be as in the previous section. Then from Section 3.7.1 the gradient at $\mathcal{Y} \in \mathsf{Gr}(n,p)$ is the element $\operatorname{grad}\widetilde{f}$ in $T_{\mathcal{Y}}\mathsf{Gr}(n,p)$ satisfying

$$\left\langle \operatorname{grad}\widetilde{f}(\mathcal{Y}), \xi_{\mathcal{Y}}\right\rangle = D\widetilde{f}(Y)[\xi_{\mathcal{Y}}] \tag{3.17}$$

for all $\xi_{\mathcal{Y}} \in T_{\mathcal{Y}}\mathsf{Gr}(n,p)$. With $f(X) := \widetilde{f}(\pi(X))$ it is easily verified that

$$\zeta_X := (I_n - XX^T)\nabla f(X) \in H_X \tag{3.18}$$

for $X \in \pi^{-1}(\mathcal{Y})$ satisfies $\langle \zeta_X, \mu_X\rangle = Df(X)[\mu_X]$ for $\mu_X \in H_X$. See [46, Section 2.5.3].

Let $\nu_{\mathcal{Y}} = \tau_X^{-1}(\zeta_X)$ where ζ_X is by definition independent of the choice of X. Then

$$\begin{aligned}\langle \nu_{\mathcal{Y}}, \xi_{\mathcal{Y}}\rangle &= \langle \zeta_X, \mu_X\rangle \\ &= Df(X)[\mu_X] \\ &= D\widetilde{f}(\pi(X))[D\pi(X)[\mu_X]] \\ &= D\widetilde{f}(\mathcal{Y})[\xi_{\mathcal{Y}}]\end{aligned}$$

for $\mu_X = \tau_X(\xi_{\mathcal{Y}})$. Hence, the element $\nu_{\mathcal{Y}}$ satisfies (3.17) and is therefore the gradient of $\widetilde{f}$ at $\mathcal{Y}$.

Note that later we will deal with the element ζ_X, written as $\operatorname{grad} f(X)$, in the horizontal space H_X instead of $\operatorname{grad} \widetilde{f}(\mathcal{Y})$ as we have a matrix representation of ζ_X. Moreover, in order to minimize a function over $\mathsf{Gr}(n,p)$ we will see later that we can entirely operate on the matrices X in the total space $\mathsf{St}(n,p)$ and the elements in H_X in our optimization algorithms as we can always map bijectively the elements of $\mathsf{Gr}(n,p)$ and $T_{\mathcal{Y}}\mathsf{Gr}(n,p)$ to elements in $\mathsf{St}(n,p)$ and H_X, respectively.

Levi-Civita Connection

From (3.7) the Levi-Civita connection $\nabla^{\mathsf{Gr}(n,p)}$ on $\mathsf{Gr}(n,p)$ is given by

$$\tau_X\big(\nabla^{\mathsf{Gr}(n,p)}_{\mu(\mathcal{Y})}\xi(\mathcal{Y})\big) = \Pi^h_X\big(\nabla^{\mathsf{St}(n,p)}_{\tau_X(\mu(\mathcal{Y}))}\tau_X(\xi(\mathcal{Y}))\big)$$

where $X \in \pi^{-1}(\mathcal{Y})$, $\mathcal{Y} \in \mathsf{Gr}(n,p)$, $\xi(\mathcal{Y}), \mu(\mathcal{Y}) \in T_{\mathcal{Y}}\mathsf{Gr}(n,p)$ and $\nabla^{\mathsf{St}(n,p)}$ the Levi-Civita connection on the Stiefel manifold. By [3, Example 5.3.3] $\tau_X\big(\nabla^{\mathsf{Gr}(n,p)}_{\mu(\mathcal{Y})}\xi(\mathcal{Y})\big)$ reduces to

$$\tau_X\big(\nabla^{\mathsf{Gr}(n,p)}_{\mu(\mathcal{Y})}\xi(\mathcal{Y})\big) = \Pi^h_X\left(D(\tau_X(\mu(\mathcal{Y})))[\tau_X(\xi(\mathcal{Y}))]\right).$$

Geodesics

Let $\mathcal{Y} \in \mathsf{Gr}(n,p)$, $\xi_{\mathcal{Y}} \in T_{\mathcal{Y}}\mathsf{Gr}(n,p)$ and X be an element of $\pi^{-1}(\mathcal{Y})$ with $\mu_X = \tau_X(\xi_{\mathcal{Y}})$ in H_X. By [46, Theorem 2.3] the geodesic of $\mathsf{Gr}(n,p)$ at a point $\mathcal{Y} \in \mathsf{Gr}(n,p)$ in direction $\xi_{\mathcal{Y}} \in T_{\mathcal{Y}}\mathsf{Gr}(n,p)$ is then given by $\mathcal{Y}(t) = \pi(X(t))$ with

$$X(t) = \begin{bmatrix} XV & U \end{bmatrix} \begin{bmatrix} \cos(t\Sigma) \\ \sin(t\Sigma) \end{bmatrix} V^T, \tag{3.19}$$

where $\mu_X = U\Sigma V^T$ is the compact singular value decomposition of μ_X and $\cos(D)$ and $\sin(D)$ are the operators applying the cos and the sin function, respectively on the diagonal elements of the diagonal matrix D. A constructive proof of (3.19) can be found in [2].

Retractions

By [3, Example 4.1.3] and [3, Proposition 4.1.3] a retraction of $\mathsf{Gr}(n,p)$ at a point $\mathcal{Y}$ is given by

$$R_{\mathcal{Y}}(\xi_{\mathcal{Y}}) = \pi\left((X + \mu_X)(I_p + \mu_X^T\mu_X)^{-1/2}\right) \tag{3.20}$$

where $(X + \mu_X)(I_p + \mu_X^T\mu_X)^{-1/2}$ is the unitary polar factor of $(X + \mu_X)$ with X and μ_X as defined above.

Vector Transport

By using the projection $\Pi^h_X(Z)$ onto H_X as defined in (3.16) we obtain a vector transport by

$$\tau_{\widehat{X}}(\mathcal{T}_{\mu_\mathcal{Y}}(\xi_\mathcal{Y})) = \Pi^h_{\widehat{X}}(\nu_X) \tag{3.21}$$

for $X \in \pi^{-1}(\mathcal{Y})$ and $\widehat{X} \in \pi^{-1}(R_\mathcal{Y}(\mu_\mathcal{Y}))$, and $\nu_X = \tau_X(\xi_\mathcal{Y})$.

Another vector transport is obtained by the parallel translation along the geodesics. Let $\xi_\mathcal{Y}$ and $\mu_\mathcal{Y}$ be tangent vectors in $T_\mathcal{Y}\mathsf{Gr}(n,p)$ and let $\tau_X(\mu_\mathcal{Y}) = U\Sigma V^T$ the compact singular value decomposition of $\tau_X(\mu_\mathcal{Y})$ for $X \in \pi^{-1}(\mathcal{Y})$. Then the parallel translation along the geodesic $\mathcal{Y}(t) = \pi(X(t))$ in direction $\mu_\mathcal{Y}$ at $\mathcal{Y} = \mathcal{Y}(0)$ given by [46, Theorem 2.4] is

$$\tau_{X(t)}(P^{t\leftarrow 0}_{\mathcal{Y}(t)}(\xi_\mathcal{Y})) = \left(\begin{bmatrix} XV & U \end{bmatrix} \begin{bmatrix} -\sin(t\Sigma) \\ \cos(t\Sigma) \end{bmatrix} U^T + (I - UU^T) \right) \tau_X(\xi_\mathcal{Y}). \tag{3.22}$$

We are ready now to introduce optimization algorithms minimizing an objective function f over a Riemannian manifold.

3.9 Optimization Algorithms

In this section we will introduce popular optimization algorithms that have already successfully been applied to many problem in science [46], [138], [3]. One of them is the BFGS-algorithm over Riemannian manifolds called Riemannian BFGS (RBFGS), which was proposed in [49]. This algorithm has recently been extended for the use of retractions [109]. However, the approximation to the Hessian obtained by rank two updates is stored fully in memory. As this approximated Hessian stored as a matrix can be large in dimension for e.g. the Stiefel or Grassmannian manifold we propose a limited memory version of the RBFGS algorithm. Let us now start with the nonlinear conjugate gradient algorithm.

3.9.1 Nonlinear Conjugate Gradient Algorithm

In $\mathbb{R}^n$ the linear Conjugate Gradient (CG) method proposed by Hestenes and Stiefel [62] minimizes a quadratic function $q(x) = x^TAx + bx + c \in \mathbb{R}$ with $A \in \mathbb{R}^{n\times n}$ symmetric positive definite and $b \in \mathbb{R}^n$, $c \in \mathbb{R}$. It generates a sequence of iterates x_k minimizing $q(x)$ over the set $x_0 + \operatorname{span}\{p_0, \ldots, p_{k-1}\}$ where the basis vectors p_k are successive search direction chosen to be A-conjugate to all previous search directions $p_0, \ldots, p_{k-1}$, i.e. $\langle Ap_i, p_j\rangle = 0$ for $i \neq j$. The first search direction is the steepest descent direction $p_0 = \nabla q(x_0)$. As in iteration k, the previous directions $p_0, \ldots, p_{k-1}$

have been chosen to be A-conjugate, one needs only to determine the new search direction p_k and to perform an exact line-search to find the coefficient α_k such that

$$x_{k+1} = x_k + \alpha_k p_k \tag{3.23}$$

minimizes $q(x)$. Since α_k is determined by means of exact line-search the gradient $\nabla q(x)$ at x_{k+1} is orthogonal to p_k. The idea is now to determine the new search direction p_{k+1} by a composition of p_k and $\nabla q(x_{k+1})$ that is

$$p_{k+1} = -\nabla q(x_{k+1}) + \beta_{k+1} p_k \tag{3.24}$$

where β_{k+1} is chosen such that p_{k+1} is A-conjugate to p_k. The linear CG method converges in n iterations. For the nonlinear CG method for nonlinear function f two changes need to be made. As determining the exact line-search for α_k becomes more complicated this is replaced by a line-search strategy. Now in order to compensate this modification several alternatives to β_k are possible such that the nonlinear CG reduces to the linear CG if $q(x)$ is strictly convex and α_k is chosen by means of exact line-search procedure. The most popular choices are the one by Fletcher-Reeves

$$\beta_k = \frac{\|\nabla f(x_k)\|_2^2}{\|\nabla f(x_{k-1})\|_2^2} \tag{3.25}$$

and by Polak-Riberière

$$\beta_k = \frac{\|\nabla f(x_k)\|_2^2 - \langle \nabla f(x_k), \nabla f(x_{k-1}) \rangle}{\|\nabla f(x_{k-1})\|_2^2}. \tag{3.26}$$

For more on linear and nonlinear CG methods see [102, Chapter 5].

Now we are going to generalize this to manifolds that requires some modifications to (3.23), (3.24), in the line-search strategy, and in (3.25) or (3.26). Remediation is provided by the concept of retractions and vector transport. Let the iterates x_k now be generated on the manifold $\mathcal{M}$ and let the new search direction p_k be chosen in $T_{x_k}\mathcal{M}$. Therefore we replace (3.23) by $x_{k+1} = R_{x_k}(\alpha_k \xi_{x_k})$ for $\xi_{x_k} \in T_{x_k}\mathcal{M}$. For (3.24) we need to transport the vector ξ_{x_k} that is the search direction in iteration k into $T_{x_{k+1}}\mathcal{M}$ in order to form the sum with $\operatorname{grad} f(x_{k+1})$. We proceed similarly with the line-search procedure and the choices for β_{k+1}, yielding Algorithm 3.9.1, see [3, Algorithm 13].

Under suitable conditions this algorithm converges globally [102, Section 5.2], [3, Theorem 4.3.1] and has superlinear convergence [124]. For the global convergence one requires that the direction in line 6 in Algorithm 3.9.1 is a descent direction. In $\mathbb{R}^n$ this condition can be ensured for the choice of Fletcher-Reeves by imposing strong Wolfe conditions [102, p. 34] in the backtracking strategy. If the vector transport is

Algorithm 3.9.1 Nonlinear Conjugate Gradient Algorithm on Riemannian manifold $\mathcal{M}$.

This algorithm implements the nonlinear conjugate gradient algorithm for Riemannian manifolds by generating a sequence $x_k \in \mathcal{M}$ of iterates to find a local minimum of a real valued function $f \in \mathcal{F}(\mathcal{M})$.

Require: $x_0 \in \mathcal{M}$, $\varepsilon > 0$ the tolerance for the norm of gradient at the last iterate.

1 Compute $\xi_{x_0} = -\operatorname{grad} f(x_0)$. Set $k = 0$.

2 **while** $\|\operatorname{grad} f(x_k)\|_{x_k} > \varepsilon$ **do**

3 Compute step size α_k by means of a backtracking strategy, e.g. Armijo-backtracking: find the smallest natural number $m_k \in \mathbb{N} \cup \{0\}$ such that

$$f(R_{x_k}(\alpha\rho^{m_k}\xi_{x_k})) < f(x_k) + \gamma\alpha\rho^{m_k} \langle \operatorname{grad} f(x_k), \xi_{x_k} \rangle_{x_k} . \tag{3.27}$$

for $\alpha > 0$, $\gamma, \rho \in (0, 1)$. Then $\alpha_k = \alpha\rho^{m_k}$.

4 Set $x_{k+1} := R_{x_k}(\alpha_k \xi_{x_k})$.

5 Compute β_{k+1} by Fletcher-Reeves, that is,

$$\beta_{k+1} = \frac{\|\operatorname{grad} f(x_{k+1})\|^2_{x_{k+1}}}{\|\operatorname{grad} f(x_k)\|^2_{x_k}}$$

or β_{k+1} by Polak-Ribière, that is,

$$\beta_{k+1} = \frac{\|\operatorname{grad} f(x_{k+1})\|^2_{x_{k+1}} - \langle \operatorname{grad} f(x_{k+1}), \mathcal{T}_{\alpha_k \xi_k}(\operatorname{grad} f(x_k)) \rangle_{x_{k+1}}}{\|\operatorname{grad} f(x_k)\|^2_{x_k}}.$$

6 Set $\xi_{x_{k+1}} = -\operatorname{grad} f(x_{k+1}) + \beta_{k+1} \mathcal{T}_{\alpha\xi_{x_k}}(\xi_{x_k})$.

7 Set $k = k + 1$.

8 **end while**

9 **return** x_k.

the parallel translation, see Section 3.7.5 then due to the preservation of the metric these conditions can be generalized to Riemannian manifolds.

Note that the nonlinear CG method is often the method of choice for large scale problems where no second-order information is available. In particular it only requires to compute the gradient at the iterates and at most two vector transports per iteration.

3.9.2 Limited Memory RBFGS

Let us now consider the BFGS-method on Riemannian manifolds. The BFGS-method is a quasi-Newton method named after its discoverers Broyden, Fletcher, Goldfarb, and Shanno. Quasi-Newton methods are iterative methods that construct a model of the objective function by only requiring the gradient of the function at every iteration. A descent direction of the objective function is then found by minimizing this constructed model. To describe the idea of the BFGS-method we first consider the method in $\mathbb{R}^n$ equipped with the Euclidean inner product and move then to the generalisation for Riemannian manifolds. Let f be a function in $\mathbb{R}$ to be minimized. The idea of this method is to generate a sequence of iterates $x_{k+1} = x_k + \alpha_k p_k$ where the descent direction p_k is chosen to minimize the quadratic model

$$m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p$$

of f at x_k where B_k is a symmetric positive definite matrix, determined in the following way. Since in the optimization routine one needs to deal with $H_k = B_k^{-1}$ instead of B_k one rather considers how to choose H_k. The first requirement on H_k is that the gradient of the model $m_k(p)$ should match $\nabla f(x)$ at the iterates x_k and x_{k-1}. This condition implies

$$H_k y_k = s_k \text{ with } y_k = \nabla f(x_k) - \nabla f(x_{k-1}) \text{ and } s_k = x_k - x_{k-1}, \tag{3.28}$$

imposing n constraints on H_k. As H_k is also symmetric positive definite the condition (3.28) is only well defined if $\rho_k := \langle s_k, y_k \rangle > 0$. By enforcing the Wolfe conditions [41, Section 6.3] in the line-search procedure one can guarantee that ρ_k is positive. The second requirement, determining the matrix H_k uniquely, is that the new matrix H_k is closest to the previously chosen symmetric positive definite matrix H_{k-1}. The condition of closeness is thereby expressed in the weighted Frobenius norm $\|A\|_W = \|W^{1/2} A W^{1/2}\|_F$ with $W = \bar{G}_{k-1}$ the inverse of the averaged Hessian defined by

$$\bar{G}_{k-1} = \int_0^1 \nabla^2 f(x_{k-1} + t\alpha_{k-1} p_{k-1}) dt.$$

This choice of the weighting matrix W ensures that the updating formula for H_k is invariant to scaling transformations. To determine H_k we need to find the solution of

$$\begin{aligned} \min_{H \in \mathcal{S}_n^+} \quad & \|H - H_{k-1}\|_W \\ \text{s.t.} \quad & Hy_k = s_k. \end{aligned}$$

This optimization problem has the unique solution H_k given by [102, Section 6.1]

$$H_k = \left(I - \frac{1}{\rho_k} s_k y_k^T\right) H_{k-1} \left(I - \frac{1}{\rho_k} y_k s_k^T\right) + \frac{1}{\rho_k} s_k s_k^T. \tag{3.29}$$

Note that this corresponds to a rank two update of the matrix H_{k-1}. Certainly other choices for H_k, respectively B_k can be considered. However, the BFGS-method is known to be highly efficient [102, Section 6.1]. As H_k is chosen to be positive definite p_k is a descent direction. Under suitable conditions this algorithm converges globally to a minimum of the function f [102, Theorem 6.5].

The matrix H_k is usually dense and therefore for large values of n it is impractical to store it. In this case one would like to efficiently compute $H_k d$ for an arbitrary vector d without storing H_k explicitly. The solution is the limited memory BFGS-method described in [101]. The idea is to store only the most recent $M \ll n$ pairs (y_i, s_i) and approximate $H_k d$ by starting from $H_{\max\{0,k-M\}} := H_0$ and applying iteratively (3.29). This corresponds to performing a sequence of inner products and summation of vectors by using only the most M recent pairs (y_i, s_i).

Now we are going to generalize this method to minimize a function $f \in \mathcal{F}(\mathcal{M})$ over the Riemannian manifold $\mathcal{M}$. Let $x_k = R_{x_{k-1}}(\alpha_{k-1}\mu_{x_{k-1}}) \in \mathcal{M}$ be the current iterate where $\mu_{x_{k-1}} \in T_{x_{k-1}}\mathcal{M}$ is the tangent vector that was chosen as a descent direction in the previous iteration and α_{k-1} the step size. Then the pair (y_k, s_k) becomes

$$s_k = \mathcal{T}_{\alpha_{k-1}\mu_{x_{k-1}}}(\alpha_{k-1}\mu_{x_{k-1}}), \quad y_k = \operatorname{grad} f(x_k) - \mathcal{T}_{\alpha_{k-1}\mu_{x_{k-1}}}(\operatorname{grad} f(x_{k-1})). \tag{3.30}$$

We equally define ρ_k as

$$\rho_k := \langle s_k, y_k \rangle_{x_k}. \tag{3.31}$$

Note that if the Wolfe condition on α_k are imposed and the vector transport is the parallel translation along geodesics then ρ will still be positive.

Lemma 3.9.1. *Let $x \in \mathcal{M}$ and $\mu_x \in T_x\mathcal{M}$ a tangent vector with $\langle \operatorname{grad} f, \mu_x \rangle_x < 0$. Further let $\gamma(\alpha)$ be the geodesic with $\gamma(0) = x$ and $\dot{\gamma}(0) = \mu_x$ and $\mathcal{T}_\cdot(\cdot)$ be the parallel translation at x associated with γ. Assume that $\{f(\gamma(\alpha)) : \alpha > 0\}$ is bounded from below. Then there exists an α for $0 < c_1 < c_2 < 1$ such that the generalized Wolfe conditions*

$$f(\gamma(\alpha)) \leq f(x) + c_1 \alpha \langle \operatorname{grad} f, \mu_x \rangle_x \tag{3.32}$$

and

$$\langle \operatorname{grad} f(\gamma(\alpha)), \mathcal{T}_{\alpha\mu_x}\mu_x\rangle_{\gamma(\alpha)} \geq c_2 \langle \operatorname{grad} f(x), \mu_x\rangle_x \tag{3.33}$$

are satisfied.

Proof. The proof is similar to the one in $\mathbb{R}^n$ [41, Theorem 6.3.2]. As $c_1 < 1$ condition (3.32) is satisfied for $\alpha > 0$ small enough. As $f(\gamma(\alpha))$ is bounded below there exists a smallest positive $\bar{\alpha}$ such that

$$f(\gamma(\bar{\alpha})) = f(x) + c_1\bar{\alpha}\langle \operatorname{grad} f(x), \mu_x\rangle_x \tag{3.34}$$

and thus for any $\alpha \in (0, \bar{\alpha}]$ condition (3.32) is satisfied. As $f(\gamma(\alpha))$ is a function from $\mathbb{R} \mapsto \mathbb{R}$ we can apply the mean value theorem and obtain for an $\widehat{\alpha} \in (0, \bar{\alpha})$

$$f(\gamma(\bar{\alpha})) - f(\gamma(0)) = \bar{\alpha}\langle \operatorname{grad} f(\gamma(\widehat{\alpha})), \dot{\gamma}(\widehat{\alpha})\rangle_{\gamma(\widehat{\alpha})}. \tag{3.35}$$

As the geodesic parallel transports its own tangent vector [33, p. 228] $\dot{\gamma}(\widehat{\alpha}) = \mathcal{T}_{\widehat{\alpha}\mu_x}(\mu_x)$. Combining this with (3.35) and (3.34) we have

$$\langle \operatorname{grad} f(\gamma(\widehat{\alpha})), \mathcal{T}_{\widehat{\alpha}\mu_x}(\mu_x)\rangle_{\gamma(\widehat{\alpha})} = c_1\langle \operatorname{grad} f(x), \mu_x\rangle_x > c_2\langle \operatorname{grad} f(x), \mu_x\rangle_x.$$

Hence, $\widehat{\alpha}$ satisfies the both conditions. ∎

If the Wolfe conditions are satisfied in every iteration we obtain that $\rho_k > 0$ when using the parallel translation along the geodesics and μ_k defined as in Lemma 3.9.1 as then

$$\begin{aligned}
\rho_k &= \langle s_k, y_k\rangle_{x_k} \\
&= \left\langle \mathcal{T}_{\alpha_{k-1}\mu_{x_{k-1}}}(\alpha_{k-1}\mu_{x_{k-1}}), \operatorname{grad} f(x_k)\right\rangle_{x_k} \\
&\quad - \left\langle \mathcal{T}_{\alpha_{k-1}\mu_{x_{k-1}}}(\alpha_{k-1}\mu_{x_{k-1}}), \mathcal{T}_{\alpha_{k-1}\mu_{x_{k-1}}}(\operatorname{grad} f(x_{k-1}))\right\rangle_{x_k} \\
&\geq c_2\left\langle \alpha_{k-1}\mu_{x_{k-1}}, \operatorname{grad} f(x_{k-1})\right\rangle_{x_{k-1}} - \left\langle \alpha_{k-1}\mu_{x_{k-1}}, \operatorname{grad} f(x_{k-1})\right\rangle_{x_{k-1}} > 0,
\end{aligned}$$

where we used that the vector transport is isometric. Hence, we can ensure the positiveness of ρ_k.

By using the concept of vector transport we approximate the inverse of the Hessian operator $\operatorname{Hess} f$ applied to a tangent vector $\xi_{x_k} \in T_{x_k}\mathcal{M}$ at x_k by generalizing (3.29) to

$$\begin{aligned}
H_k(\xi_{x_k}) =& \widetilde{H}_{k-1}(\xi_{x_k}) - \frac{\left\langle y_k, \widetilde{H}_{k-1}(\xi_{x_k})\right\rangle_{x_k}}{\rho_k} s_k - \frac{\langle s_k, \xi_{x_k}\rangle_{x_k}}{\rho_k}\widetilde{H}_{k-1}(y_k) \\
&+ \frac{\langle s_k, \xi_{x_k}\rangle_{x_k}\left\langle y_k, \widetilde{H}_{k-1}(y_k)\right\rangle_{x_k}}{\rho_k^2} s_k + \frac{\langle s_k, \xi_{x_k}\rangle_{x_k}}{\rho_k} s_k.
\end{aligned} \tag{3.36}$$

where $\widetilde{H}_{k-1}(\cdot) = \mathcal{T}_{\alpha_{k-1}\mu_{x_{k-1}}} \circ H_{k-1} \circ \mathcal{T}^{-1}_{\alpha_{k-1}\mu_{x_{k-1}}}(\cdot)$.

From (3.29) and (3.36) we see that the operator is positive definite if and only if $\widetilde{H}_{k-1}$ is positive definite. This is clearly the case if the vector transport is isometric. Note that $H_k(\cdot)$ is now considered to be positive definite if $\langle \xi_{x_k}, H_k(\xi_{x_k})\rangle_{x_k} > 0$ for all $\xi_{x_k} \in T_{x_k}\mathcal{M}$ and ξ_{x_k} nonzero.

For the limited memory BFGS Nocedal describes in [101] an algorithm to efficiently compute the product $H_k(p)$ from the at most last M pairs (y_i, s_i), achieving R-linear convergence under suitable conditions [89, Theorem 6.1]. In comparison the BFGS algorithm has superlinear convergence for similar conditions [102, Theorem 6.6]. The algorithm in [101] can easily be generalized to Riemannian manifolds by using (3.36) resulting in Algorithm 3.9.2. The problem that remains is how to choose the initial operator $\widetilde{H}_0$ as H_0 is an operator from $T_{x_0}\mathcal{M}$ into the same space. In order to apply the vector ν_0 to H_0 we need to transport ν_0 into $T_{x_0}\mathcal{M}$, possibly requiring to store all previous descent directions. In order to avoid this we choose $\widetilde{H}_0(\nu_0)$ to be a multiple of ν_0, which is also a popular choice in $\mathbb{R}^n$ [101, p. 142-143]. The multiple that has proven to be efficient in $\mathbb{R}^n$ is $\frac{\langle y_k, s_k\rangle}{\langle y_k, y_k\rangle}$. Therefore we adjust this choice where use the inner product in $T_{x_k}\mathcal{M}$. We are ready now to state the limited memory BFGS algorithm for Riemannian manifolds in Algorithm 3.9.3, which is similar to the RBGFS algorithm in [109].

Gabay obtains a global convergence result of this algorithm in [49, Theorem 4.6] for functions f having the additional property that the Hessian of f is non-degenerate at all stationary points of f. Note that for this result he does not limit the memory, i.e. $M = k$ in Algorithm 3.9.2 and he uses the geodesic for the retraction and the parallel translation for the vector transport. The Hessian is called non-degenerated at x if $\langle \xi_x, \operatorname{Hess} f(x)[\xi_x]\rangle = 0$ for $\xi_x \in T_x\mathcal{M}$ implies $\xi_x = 0$. Under suitable conditions the author even obtains superlinear convergence. See [49, Theorem 4.7].

3.10 Conclusions

In this chapter we gave a brief introduction to Riemannian manifolds by defining necessary geometric objects to optimize real valued function over these manifolds. We continued by considering two Riemannian manifolds in detail that will play an important role in the later chapters, that is the Stiefel and the Grassmannian manifold. Two different optimization methods were introduced to minimize nonlinear function on Riemannian manifolds, namely the nonlinear CG and the RBFGS method, and particular attention was paid to generalize the limited memory BFGS method to Riemannian manifolds.

Algorithm 3.9.2 Algorithm to compute $H_k(\xi_{x_k})$ for the limited memory BFGS.

This algorithm computes an approximation of the inverse Hessian of a function $f \in \mathcal{F}(\mathcal{M})$ applied to a tangent vector at x_k by using the most M recent pairs (y_i, s_i). This algorithm is our modification of the algorithm in [101] generalized to Riemannian manifolds.

Require: $x_0 \in \mathcal{M}$, M the maximal number of pairs (y_i, s_i) stored. The tangent vector ξ_{x_k} that is required be to applied to H_k, the at most M recent pairs $(y_{\max\{1,k+1-M\}}, s_{\max\{1,k+1-M\}}), \ldots, (y_k, s_k)$ and at most M recent directions $\mu_{\max\{0,k-M\}}, \ldots, \mu_{k-1}$.

if $k \leq M$ **then**
  $B = k$.
  $c = 0$.
else
  $B = M$.
  $c = k - M$.
end if
$\nu_B = \xi_{x_k}$
for $i = B : -1 : 1$ **do**
  $j = i + c$
  $\delta_i = \frac{1}{\rho_j} \langle s_j, \nu_i \rangle_{x_j}$
  $\nu_{i-1} = \mathcal{T}^{-1}_{\alpha_{j-1}\mu_{j-1}}(\nu_i - \delta_i y_j)$
end for
$\eta_0 = \widetilde{H}_0(\nu_0)$.
for $i = 1 : B$ **do**
  $j = i + c$
  $\eta_i = \mathcal{T}_{\alpha_{j-1}\eta_{j-1}}(\eta_{i-1})$
  $\beta_i = \frac{1}{\rho_j} \langle y_j, \eta_i \rangle_{x_j}$
  $\eta_i = \eta_i - (\delta_i - \beta_i) s_j$
end for
return ν_B.

Algorithm 3.9.3 Limited memory BFGS algorithm for Riemannian manifolds.

This algorithm implements the limited memory BFGS algorithm for Riemannian manifolds to find a local minimum of a function $f \in \mathcal{F}(\mathcal{M})$.

Require: $x_0 \in \mathcal{M}$, $\varepsilon > 0$ the tolerance for the norm of gradient at the last iterate.

Compute $\mu_{x_0} = -\operatorname{grad} f(x_0)$. Set $k = 0$.
while $\|\operatorname{grad} f(x_k)\|_{x_k} > \varepsilon$ **do**
  Compute step size α_k by means of a line-search strategy such that $x_{k+1} := R_{x_k}(\alpha_k \mu_{x_k})$ satisfies the Wolfe conditions (3.32) and (3.33).
  Set $k = k + 1$ and s_k, y_k as in (3.30).
  Compute $\mu_{x_k} = -H_k(\operatorname{grad} f(x_k))$ by using Algorithm 3.9.2.
end while
return x_k.

Chapter 4

Two-Sided Optimization Problems with Orthogonal Constraints Arising in Chemistry

4.1 Introduction

In this chapter we consider two problems that arise in atomic chemistry and involve minimization over the Stiefel manifold $\mathsf{St}(n, p)$. The minimum value of the first problem can be derived [28] by exploiting the structure of the stationary points. We briefly repeat the analysis in [28] and extend it by addressing the question of how to find the points at which the minimum value is attained. From the derivations arising in the first problem we show that the second problem is equivalent to a convex quadratic programming problem. We propose to use the active-set method to solve this problem and we show that it converges in at most $2p$ iterations to an optimal solution despite the lack of strict convexity of the objective function. We also examine the set of optimal solutions of the first problem and show that a slight modification of this set is a Riemannian manifold for which we can evolve all necessary geometric objects discussed in Chapter 3 to make an optimization over this manifold possible. The development of these objects leads to a new algorithm to optimize an arbitrary smooth function over the set of optimal solutions of the first problem. This new algorithm is an augmented Lagrangian method where the inner problem is to minimize the augmented Lagrangian function over this new Riemannian manifold. We show that this algorithm outperforms the classical approach, that is, to use again the augmented Lagrangian method but to formulate a different augmented Lagrangian function with $(p-1)p/2$ more Lagrange multipliers where the inner problem is to minimize this function over the Stiefel manifold.

The outline of this chapter is as follows. In the next section we introduce the

problems and describe briefly their applications in chemistry. We derive the optimal conditions for the first problem in Section 4.3 and discuss how to obtain the stationary points with the minimal value in Section 4.4. This leads to Algorithm 4.4.1 that computes the minimum of the first problem. In Section 4.5 we show how we can then determine the optimal solution of the second problem, leading to Algorithm 4.5.1. Subsequently, we evolve the necessary geometric objects mentioned above in Section 4.6 and investigate the performance of the resulting algorithm in our numerical tests in Section 4.7.

4.2 The Problems

Let $N \in \mathbb{R}^{n\times n}$ be a given symmetric matrix and $D = \operatorname{diag}(d_1, d_2, \ldots, d_p) \in \mathbb{R}^{p\times p}$ be diagonal with $(D)_{ii} = d_i$ and $p \leq n$. We define

$$\langle A, B\rangle := \operatorname{trace}(B^T A)$$

as our inner product in $\mathbb{R}^{n\times p}$ and the corresponding norm is the Frobenius norm $\|A\|_F^2 := \langle A, A\rangle$. Let us now state the two problems that arise from atomic chemistry.

4.2.1 Problem 1

The first problem concerns the minimization in the Frobenius norm of the difference between a symmetric matrix and a diagonal matrix D, that is,

$$\min_{Y^TY=I_p,\, Y\in\mathbb{R}^{n\times p}} \|Y^TNY - D\|_F^2. \tag{4.1}$$

For simplicity we consider N being symmetric and Y having orthonormal columns, although the analysis in Section 4.3 can also be applied for N Hermitian and Y having unitary columns.

We became aware of (4.1) from Prof. Alexander Sax, University of Graz, who came across this problem in atomic chemistry [72], [90]. In his particular problem N is a block of a density operator of a molecular system defined on a large space of atomic orbitals and D is the occupation numbers of p atomic orbitals. Then the application of (4.1) is to determine a minimal set of localised atomic orbitals having occupation numbers closest to the occupation numbers prescribed in D.

4.2.2 Problem 2

For the second problem we are interested in a different distance measure, that is

$$\min_{Y^TY=I_p,\, Y\in\mathbb{R}^{n\times p}} (\operatorname{trace}(Y^TNY) - c)^2, \tag{4.2}$$

where $c \in \mathbb{N}$ is given and could be considered as $c = \text{trace}(D)$.

This problem was also provided by Prof. Sax [120] and its applications are similar to those of (4.1). The difference is that in (4.2) one is only interested in reproducing the number of electrons of the atoms, c, and not in prescribing the occupation numbers of the atomic orbitals.

In the next section we will look more closely at (4.1) and will detect the stationary points on the Stiefel manifold by exploiting the structure of Y at those points. From this analysis we address the question of how to find a solution of (4.1) and develop an algorithm returning this solution.

4.3 Optimality Conditions for Problem 1

The analysis of this section is mainly from [28] where the stationary points of (4.1) are investigated and an optimal function value is found. Note that the authors consider a more general version of (4.1) in [28], where the objective function is $\|Y^*NY - U^*BU\|_F^2$ for given Hermitian matrices N and B of possible different dimensions. The optimization is then carried out for (Y, U) in the product manifold of two sets of rectangular matrices with unitary columns.

4.3.1 Conditions for Stationary Points

Since our constraint set of (4.1) is the compact Stiefel manifold introduced in Section 3.8.1 we consider how to determine the stationary points of the problem in (4.1) on this manifold. Recall from Section 3.8.1 that the gradient of the Stiefel manifold in the Euclidean inner product is

$$\text{grad}\, f(Y) = \nabla f(Y) - Y\text{sym}\left(Y^T \nabla f(Y)\right)$$

where $f(Y) := \|Y^T NY - D\|_F^2$ is the objective function of (4.1). Therefore we need to find the points $Y \in \mathsf{St}(n, p)$ that satisfy

$$\nabla f(Y) - Y\text{sym}\left(Y^T \nabla f(Y)\right) = 0. \tag{4.3}$$

By substituting the matrix of partial derivatives

$$\nabla f(Y) = 4NY(Y^T NY - D)$$

into (4.3) we obtain

$$NY(Y^T NY - D) = Y\text{sym}\left(Y^T NY(Y^T NY - D)\right). \tag{4.4}$$

Now when multiplying Y^T from the left in (4.4) we notice that $Y^T NYD$ must be symmetric, which implies that we can assume that $Y^T NY$ and D have the same

eigenvectors [69, Theorem 2.3.3]. As the Frobenius norm is invariant to orthogonal transformations it follows from (4.1) that we can also assume without loss of generality that Y^TNY is diagonal. We denote this diagonal matrix by

$$\Delta := Y^TNY \text{ with } (\Delta)_{ii} = \delta_i \tag{4.5}$$

for $i = 1, \ldots, p$.

It follows that the condition (4.4) for $Y \in \mathsf{St}(n,p)$ to be a stationary point simplifies to

$$(NY - Y\Delta)(\Delta - D) = 0.$$

This means either the ith diagonal entry of Δ coincides with the ith diagonal entry of D or (Y_i, δ_i) is an eigenpair of N where Y_i denotes the ith column of Y.

4.3.2 Attaining Optimal Function Value

Let us now have a look at the objective function at the stationary points. As Y^TNY is diagonal the function value $f(Y)$ simplifies to

$$f(Y) = \|Y^TNY - D\|_F^2 = \|\Delta - D\|_F = \sum_{i=1}^{p} |\delta_i - d_i|^2 . \tag{4.6}$$

As a consequence, to find the stationary point with the smallest function value, we need to choose the columns of Y such that the diagonal elements of Δ are closest to the corresponding elements of D. To achieve this goal, we will apply a theorem that specifies when a matrix *imbeddable*. The latter property is defined as follows.

Definition 4.3.1. Let $A \in \mathbb{R}^{n\times n}$ and $A_Y \in \mathbb{R}^{p\times p}$ be two square matrices with $n \geq p$. Then A_Y is called *imbeddable* in A if there exists a matrix $Y \in \mathsf{St}(n,p)$ such that $Y^TAY = A_Y$.

The next theorem by Fan and Paul [47] gives a relation of the eigenvalues of a matrix A_Y that is imbeddable in a larger matrix A to the eigenvalues of A. This will help us to determine the diagonal elements of Δ at which the optimum is attained.

Theorem 4.3.2. *[47, Theorem 1] Let $A \in \mathbb{R}^{n\times n}$ and $A_Y \in \mathbb{R}^{p\times p}$ be two symmetric matrices and $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_p$ their corresponding eigenvalues. Then A_Y is imbeddable in A iff*

$$\theta_i \in [\lambda_i, \lambda_{i-p+n}] \quad i = 1, \ldots, p.$$

Let $N = P\Lambda P^T$ be the spectral decomposition of N with Λ the diagonal matrix of the eigenvalues $\lambda_1, \ldots, \lambda_n$ of N with $\lambda_1 \leq \cdots \leq \lambda_n$. Let the diagonal elements

of $D = \operatorname{diag}\big((d_1, d_2, \ldots, d_p)^T\big)$ be in increasing order $d_1 \leq d_2 \leq \cdots \leq d_p$. We can assume this, as the Frobenius norm is invariant under orthogonal transformations so that (4.1) does not change.

Then from (4.6) one can see that a minimum value for $f(Y)$ at the stationary points is achieved if the diagonal elements of Δ are also increasing. Hence, let $\Delta = \operatorname{diag}((\delta_1, \delta_2, \ldots, \delta_p)^T)$ with $\delta_1 \leq \delta_2 \leq \cdots \leq \delta_p$.

Since Δ is imbeddable by definition in (4.5) it holds that $\delta_i \in [\lambda_i, \lambda_{i-p+n}]$ for all $i = 1, \ldots, p$. Hence by Theorem 4.3.2 the smallest value at a stationary point is attained if δ_i is chosen such that the distance between δ_i and d_i is smallest. Thus we obtain

$$\delta_i^* = \begin{cases} d_i & \text{if } d_i \in (\lambda_i, \lambda_{i-p+n}) \\ \lambda_i & \text{if } d_i \leq \lambda_i \\ \lambda_{i-p+n} & \text{otherwise} \end{cases} \tag{4.7}$$

and the function value at this point is

$$\sum_{i=1}^{p} \big(\max\{0, \lambda_i - d_i, d_i - \lambda_{i-p+n}\}\big)^2. \tag{4.8}$$

This is the minimal function value of (4.1). Let Y_* be an optimal solution. Then $Y_*^T N Y_* = \operatorname{diag}(\delta_1^*, \ldots, \delta_p^*)$ and we denote this matrix as Δ_*.

It remains to compute Y_*. In the subsequent section we address how a solution Y_* is obtained and introduce an algorithm by using the analysis of this section.

4.4 Steps to Optimal Solution of Problem 1

From now on we present our idea how to obtain an optimal solution Y_* of (4.1) and develop an algorithm that computes a $Y_* \in \mathsf{St}(n, p)$ at which the function value of (4.8) is attained. If $p = n - 1$ by Theorem 4.3.2 the eigenvalues $\lambda_1, \ldots, \lambda_n$ of N have the property of interlacing $\delta_1, \ldots, \delta_p$, i.e.

$$\lambda_1 \leq \delta_1 \leq \lambda_2 \leq \cdots \leq \delta_p \leq \lambda_n.$$

This property will allow us to apply a theorem in [107] for $p = n - 1$, which shows that we can construct an arrowhead matrix A such that the $(n-1)$st principal minor of A is Δ_* and the eigenvalues of A coincide with the eigenvalues of N. By using this theorem we can easily obtain a solution of (4.1).

To generalize this procedure to any p our idea is now after diagonalizing N to apply permutation matrices to Λ and Δ_*, respectively such that we obtain smaller diagonal matrices that allow us to apply the theorem in [107]. This method has the advantage that we can obtain a solution of (4.1) by only performing a few eigenvalue

decompositions of dimension less than n provided the spectral decomposition of N is already determined. We show that it is always possible to find permutation matrices such that these smaller diagonal matrices are obtained. The result is an algorithm that returns a solution of (4.1) for any $p \leq n$.

Note that for simplicity reasons we say that a diagonal matrix $A \in \mathbb{R}^{m\times m}$ *interlaces* a diagonal matrix $B \in \mathbb{R}^{(m-1)\times(m-1)}$ if their diagonal elements interlace. Let us now state the theorem in [107] and show how this yields a solution of (4.1) if Λ interlaces Δ_*.

4.4.1 Construction of Arrowhead Matrix with Prescribed Eigenspectrum

In [107, Theorem 1] Parlett and Strang distinguished three cases how the elements can interlace depending on the occurrence of strict inequalities between these elements. We have rewritten this theorem in a more compressed form without distinguishing these three cases, which makes the later algorithm clearer. As the statement of this theorem is clearly the same we omit the proof.

Theorem 4.4.1. *[107, Theorem 1] Let $\lambda_1, \ldots, \lambda_m$ interlace $\delta_1, \ldots, \delta_{m-1}$, i.e.*

$$\lambda_1 \leq \delta_1 \leq \lambda_2 \leq \cdots \leq \lambda_{m-1} \leq \delta_{m-1} \leq \lambda_m$$

for $m > 1$. Further for $k \in \mathbb{N}$ let the vectors

$$v_1 = \begin{bmatrix} 1 \\ v_1(2) \end{bmatrix}, \; v_2 = \begin{bmatrix} v_1(2)+1 \\ v_2(2) \end{bmatrix}, \ldots, \; v_k = \begin{bmatrix} v_{k-1}(2)+1 \\ m-1 \end{bmatrix} \in \mathbb{N}^2$$

be chosen such that for all $i = 1, \ldots, k$

$$\lambda_{v_i(1)} \leq \delta_{v_i(1)} = \lambda_{v_i(1)+1} = \cdots = \lambda_{v_i(2)} = \delta_{v_i(2)} \leq \lambda_{v_i(2)+1}$$

with $v_i(2) - v_i(1)$ maximal. Then there exist $c_1, \ldots, c_{m-1} \in \mathbb{R}$ such that the symmetric arrowhead matrix

$$A = \begin{bmatrix} \delta_1 & & & c_1 \\ & \ddots & & \vdots \\ & & \delta_{m-1} & c_{m-1} \\ c_1 & \ldots & c_{m-1} & \sum_{i=1}^{m-1}(\lambda_i - \delta_i) + \lambda_m \end{bmatrix}$$

has the eigenvalues $\lambda_1, \ldots, \lambda_m$. For the values of $c_1, \ldots, c_{m-1}$ it holds that for $i = 1, \ldots, k$

$$c_{v_i(1)}^2 + \cdots + c_{v_i(2)}^2 = -\frac{\Pi_{j<=v_i(1)}(\delta_{v_i(1)} - \lambda_j)\Pi_{j>v_i(2)}(\delta_{v_i(1)} - \lambda_j)}{\Pi_{j<v_i(1)}(\delta_{v_i(1)} - \delta_j)\Pi_{j>v_i(2)}(\delta_{v_i(1)} - \delta_j)} \geq 0.$$

Note that the condition of $v_i(2) - v_i(1)$ to be maximal ensures that the vectors $v_1, \ldots, v_k$ are uniquely determined. Also if $v_i(1) = v_i(2)$ then $c^2_{v_i(1)}$ is uniquely determined. Further if $\lambda_{v_i(1)} = \delta_{v_i(1)}$ or $\delta_{v_i(2)} = \lambda_{v_i(2)+1}$ then all the $c_{v_i(1)}, \ldots, c_{v_i(2)}$ are zero.

In case $p = n - 1$ we now obtain Y_* that imbeds Δ_* chosen in (4.8) into N by first forming the corresponding arrowhead matrix A accordingly to Theorem 4.4.1. Then as A is symmetric there is a spectral decomposition $A = V\Lambda V^T$ and we can set $Y := PV^T I_{n,n-1}$ with $I_{n,n-1} = [I_{n-1}, 0]^T$. Since $Y^T NY = I^T_{n,n-1} A I_{n,n-1} = \Delta_*$ and Y satisfies $Y^T Y = I_p$, Y is a solution of (4.1) with (4.8) as solution value.

4.4.2 Obtaining an Optimal Solution

Now we show how Theorem 4.4.1 can be used to obtain a solution for any p with $p \leq n$. Recall we are trying to find a matrix $Y \in \mathsf{St}(n, p)$ satisfying

$$Y^T NY = \Delta_* \tag{4.9}$$

for Δ_* having the prescribed diagonal elements of (4.7). As the matrix Δ_* is chosen to be imbeddable in N (4.9) is well defined.

This becomes trivial for $n = p$ and as we assumed in Section 4.3.1 that the diagonal elements of Λ are in increasing order the solution of (4.9) is just the matrix of eigenvectors P of N. Let us now assume that $p < n$. As mentioned above our idea is to permute the diagonal elements of Λ and Δ such that we obtain smaller diagonal matrices $\Lambda_1, \ldots, \Lambda_{q+1}$ and $\Delta_1, \ldots, \Delta_q$ that allow us to apply Theorem 4.4.1 to these smaller matrices. The next lemma shows that the corresponding permutation matrices always exist and as the proof of this lemma is constructive it also tells us how to choose these permutation matrices.

Lemma 4.4.2. *Let $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$, $\Delta = \operatorname{diag}(\delta_1, \ldots, \delta_p) \in \mathbb{R}^{p\times p}$ be diagonal matrices with $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $\delta_1 \leq \delta_2 \leq \cdots \leq \delta_p$. Further let $p < n$, $q = \min\{p, n - p\}$ and*

$$\delta_i \in [\lambda_i, \lambda_{i-p+n}] \quad \forall i = 1, \ldots, p. \tag{4.10}$$

Then there exist permutation matrices $U \in \mathbb{R}^{n\times n}, Q \in \mathbb{R}^{p\times p}$ and diagonal matrices $\Lambda_i \in \mathbb{R}^{s_i \times s_i}$ for $i = 1, \ldots, q + 1$ and $\Delta_i \in \mathbb{R}^{(s_i-1)\times(s_i-1)}$ for $i = 1, \ldots, q$ such that

$$U^T \Lambda U = \operatorname{diag}(\Lambda_1, \Lambda_2, \ldots, \Lambda_{q+1}), \; Q^T \Delta Q = \operatorname{diag}(\Delta_1, \Delta_2, \ldots, \Delta_q) \tag{4.11}$$

and Λ_i interlaces Δ_i for $i = 1, \ldots, q$.

Note that Λ_{q+1} contains the diagonal elements of Λ that are not used to interlace the diagonal elements of Δ and are thus, remaining. Before we prove this lemma we show how by means of this lemma an optimal solution of (4.1) is easily found.

For now let us assume Lemma 4.4.2 is true then with $q = \min\{p, n-p\}$ there exist two permutation matrices U and Q such that $U^T \Lambda U = \operatorname{diag}(\Lambda_1, \Lambda_2, \ldots, \Lambda_{q+1})$ and $Q^T \Delta_* Q = \operatorname{diag}(\Delta_1, \Delta_2, \ldots, \Delta_q)$ where the smaller diagonal matrices $\Lambda_1, \Lambda_2, \ldots, \Lambda_{q+1}$ and $\Delta_1, \Delta_2, \ldots, \Delta_q$ satisfy the conditions in Lemma 4.4.2. Since Λ_i interlaces Δ_i we can apply Theorem 4.4.1 and obtain $Y_i \in \mathsf{St}(s_i, s_i - 1)$ for all $i = 1, \ldots, q$ such that $Y_i^T \Lambda_i Y_i = \Delta_i$. Hence the matrix

$$\widehat{Y} := \operatorname{diag}(Y_1, Y_2, \ldots, Y_q) \tag{4.12}$$

solves $\widehat{Y}^T U^T \Lambda U \widehat{Y} = Q^T \Delta_* Q$ and consequently, the solution of $Y^T N Y = \Delta_*$ can be obtained by setting $Y := PU\widehat{Y}Q^T$. It remains to prove Lemma 4.4.2.

Proof. The proof of Lemma 4.4.2 is constructive, i.e. we will obtain diagonal matrices $\Lambda_1, \ldots, \Lambda_{q+1}, \Delta_1, \ldots, \Delta_q$ that satisfy the required properties. Using the floor operator $\lfloor\ \rfloor : \mathbb{R} \mapsto \mathbb{Z}$ with

$$\lfloor x \rfloor = \max_{y \in \mathbb{Z}, y \leq x} y$$

we define the numbers s_i as

$$s_i := \left\lfloor \frac{n-i}{n-p} \right\rfloor + 1, \quad i = 1, \ldots, q$$

and $s_{q+1} := n - \sum_{i=1}^{q} s_i$. Note that $s_i \geq 2$ for $i = 1, \ldots, q$ due to the choice of q.

Let $\Lambda_1, \ldots, \Lambda_{q+1}$ and $\Delta_1, \ldots, \Delta_q$ be chosen as

$$\begin{aligned} \Lambda_i &:= \operatorname{diag}(\lambda_i, \lambda_{i+n-p}, \ldots, \lambda_{i+(s_i-1)(n-p)}), \\ \Delta_i &:= \operatorname{diag}(\delta_i, \delta_{i+n-p}, \ldots, \delta_{i+(s_i-2)(n-p)}) \end{aligned} \tag{4.13}$$

for $i = 1, \ldots, q$ and Λ_{q+1} having on its diagonal all the elements $\lambda_1, \ldots, \lambda_n$ that are remaining. Let further $\widehat{\Lambda} := \operatorname{diag}(\Lambda_1, \ldots, \Lambda_{q+1})$, $\widehat{\Delta} := \operatorname{diag}(\Delta_1, \ldots, \Delta_q)$ and let $\mathcal{L}_i = \{(\Lambda_i)_{11}, \ldots, (\Lambda_i)_{s_i,s_i}\}$ be the set of all diagonal elements of Λ_i for $i = 1, \ldots, q+1$ and let $\mathcal{D}_i = \{(\Delta_i)_{11}, \ldots, (\Delta_i)_{s_i-1,s_i-1}\}$ be the corresponding set for Δ_i for $i = 1, \ldots, q$.

It remains to show that the choice (4.13) is well defined, i.e. the following properties are satisfied.

(i) There exists a bijective function, which represent an one to one correspondence between the diagonal elements of Λ and $\widehat{\Lambda}$. This is equivalent to $\bigcup_{i=1}^{q+1} \mathcal{L}_i = \{\lambda_1, \ldots, \lambda_n\}$ as by construction of $\mathcal{L}_1, \ldots, \mathcal{L}_{q+1}$ $\mathcal{L}_i \cap \mathcal{L}_j = \emptyset$ for $i \neq j$.

(ii) Analogously to (i) $\bigcup_{i=1}^{q} \mathcal{D}_i = \{\delta_1, \ldots, \delta_n\}$.

(iii) The elements of Λ_i interlace the diagonal elements of Δ_i for $i = 1, \ldots, q$.

Note that condition (i) and (ii) imply the existence of U and Q, respectively. Let us first prove these two conditions. As for all $i = 1, \ldots, q$

$$1 \leq i + (s_i - 1)(n - p) = i + \left\lfloor \frac{n-i}{n-p} \right\rfloor (n - p) \leq i + n - i \leq n$$

and

$$1 \leq i + (s_i - 2)(n - p) = i + \left(\left\lfloor \frac{n-i}{n-p} \right\rfloor - 1 \right) (n - p) \leq i + n - i + p - n = p$$

we have $\bigcup_{i=1}^{q+1} \mathcal{L}_i \subset \{\lambda_1, \ldots, \lambda_n\}$ and $\bigcup_{i=1}^{q} \mathcal{D}_i \subset \{\delta_1, \ldots, \delta_n\}$. Conversely, we will show that $\sum_{i=1}^{q+1} s_i = n$ and $\sum_{i=1}^{q} (s_i - 1) = p$ implying that $\bigcup_{i=1}^{q+1} \mathcal{L}_i \supset \{\lambda_1, \ldots, \lambda_n\}$ and $\bigcup_{i=1}^{q} \mathcal{D}_i \supset \{\delta_1, \ldots, \delta_n\}$, respectively.

We consider the two cases for q separately. Let us first assume that $q = p$. Then

$$\sum_{i=1}^{q} (s_i - 1) = \sum_{i=1}^{p} \left\lfloor \frac{n-i}{n-p} \right\rfloor \geq p$$

as $n - i \geq n - p$ and therefore, every term in the sum is greater than or equal to 1. Conversely,

$$\sum_{i=1}^{q} \left\lfloor \frac{n-i}{n-p} \right\rfloor \leq \sum_{i=1}^{q} \left\lfloor \frac{2(n-i)}{n} \right\rfloor \leq p$$

as $n - p \geq p \iff n/2 \geq p$ and $n - i < n$. It follows $\sum_{i=1}^{q} (s_i - 1) = p$ for $q = p$. Let us assume $q = n - p$ now. Then from $p \geq n - p$ we have

$$\sum_{i=1}^{q} \left\lfloor \frac{n-i}{n-p} \right\rfloor \geq \sum_{i=1}^{q} \left\lfloor \frac{n-i}{p} \right\rfloor \geq p.$$

On the other hand it holds that

$$\sum_{i=1}^{q} \left\lfloor \frac{n-i}{n-p} \right\rfloor \leq \sum_{i=1}^{q} \left\lfloor \frac{n-p+p-i}{n-p} \right\rfloor \leq n - p + \sum_{i=1}^{n-p} \left\lfloor \frac{p-i}{n-p} \right\rfloor.$$

The term $\frac{p-i}{n-p}$ is an integer for exactly one $i \in \{1, \ldots, n-p\}$. Let this index be denoted by i^* and the corresponding integer number by z. Then

$$\sum_{i=1}^{q} \left\lfloor \frac{n-i}{n-p} \right\rfloor \leq n - p + (z-1)(p - i^* - (2p - n)) + z(p - (p - i^*)) = p.$$

Hence, $\sum_{i=1}^{q} (s_i - 1) = p$ is also satisfied for $q = n - p$.

The condition $\sum_{i=1}^{q} s_i = n$ is easily shown. As the dimensions of Λ_i differ from the dimensions of Δ_i by exactly one for $i = 1, \ldots, q$ we have $\sum_{i=1}^{q} s_i = p + q \leq n$ and hence, by definition of Λ_{q+1} $\sum_{i=1}^{q} s_i = n$.

It remains to show condition (iii), which follows directly from (4.10) and the construction of (4.13). We obtain the assertion. □

Now we are ready to state the algorithm that computes the solution of (4.1). All the steps that need to be taken are listed in Algorithm 4.4.1. Let us now consider the major cost of this algorithm. First we need to count the computation of the spectral decompositions of N in line 1 and of A_i in line 13 for $i = 1, \ldots, q$, which requires approximately $25n^3 + 25\sum_{i=1}^{q} s_i^3$ flops [66, p. 337] with s_i as defined in proof of Lemma 4.4.2. As Q, W, and U are permutation matrices we additionally need only to consider the cost of one sparse matrix-matrix multiplication in line 16, which requires at most $2n^2p$ flops. In total, the major cost of Algorithm 4.4.1 is approximately $25(n^3 + \sum_{i=1}^{q} s_i^3) + 2n^2p$ flops.

Algorithm 4.4.1 Algorithm for computing the solution of (4.1).

Require: $N \in \mathbb{R}^{n\times n}$ symmetric and $D \in \mathbb{R}^{p\times p}$ diagonal, with $p \leq n$.

1 Compute the spectral decomposition $N = P\Lambda P^T$ with P orthogonal and such that the eigenvalues are sorted in increasing order, i.e., $\Lambda = \operatorname{diag}((\lambda_1, \ldots, \lambda_n)^T)$ with $\lambda_1 \leq \cdots \leq \lambda_n$.
2 Compute the permutation matrix W such that $\widetilde{D} = W^T D W$ has diagonal elements that are in increasing order.
3 Determine the diagonal elements of Δ_* by means of (4.7).
4 Compute the minimal function value $f^* := \sum_{i=1}^{p} \left|\delta_i^* - \widetilde{D}_{ii}\right|^2$ for (4.1).
5 **if** $n = p$ **then**
6 Compute the solution $Y := PW^T$.
7 **return** Y, f^*.
8 **end if**
9 Set $q := \min\{p, n-p\}$.
10 Determine the diagonal matrices $\Lambda_1, \ldots, \Lambda_{q+1}, (\Delta_*)_1, \ldots, (\Delta_*)_q$ as constructed in the proof of Lemma 4.4.2 and the corresponding permutation matrices U and Q.
11 **for** $i = 1 : q$ **do**
12 Construct the arrowhead matrices A_i for the diagonal elements of Λ_i and $(\Delta_*)_i$ according to Theorem 4.4.1.
13 Determine the spectral decomposition of $A_i = V_i \Lambda_i V_i^T$.
14 Set $Y_i := V_i^T I_{s_i, s_i - 1}$.
15 **end for**
16 Compute $\widehat{Y}$ as in (4.12) and determine the solution $Y := PU\widehat{Y}Q^T W^T$.
17 **return** Y, f^*.

4.5 Steps to Optimal Solution of Problem 2

In this section we consider how to determine the optimal solution of (4.2). By applying Theorem 4.3.2 we show that (4.2) is equivalent to a convex quadratic programming problem with box constraints. For solving this problem we consider the active-set method described in [102, Algorithm 16.3] which turns out to be efficient, as for

the inner optimization problem with equality constraints no linear system needs to be solved. Moreover, we show that the active-set method terminates in most $2p$ iterations.

4.5.1 Reformulation into a Convex Quadratic Programming

Let $\lambda_1 \leq \cdots \leq \lambda_n$ be the eigenvalues of N in (4.2). By Theorem 4.3.2 it holds for the eigenvalues $\theta_1, \ldots, \theta_p$ of $Y^T NY$ with $\theta_1 \leq \cdots \leq \theta_p$ that $\theta_i \in [\lambda_i, \lambda_{i+n-p}]$. Therefore, as the trace of a square matrix A is equal to the sum of the eigenvalues of A, we have that $\operatorname{trace}(Y^T NY) = \sum_{i=1}^p \theta_i$ with $\theta_i \in [\lambda_i, \lambda_{i+n-p}]$. Hence with $c_p := c/p$

$$\begin{array}{lll} \min & (\operatorname{trace}(Y^T \Lambda Y) - c)^2 & \\ \text{s.t.} & Y \in \mathsf{St}(n,p) & \end{array} \quad \Longleftrightarrow \quad \begin{array}{ll} \min & (\sum_{i=1}^p (\theta_i - c_p))^2 \\ \text{s.t.} & \theta_i \in [\lambda_i, \lambda_{i+n-p}] \end{array}$$

$$\Longleftrightarrow \quad \begin{array}{ll} \min & \left(\sum_{i=1}^p \mu_i\right)^2 \\ \text{s.t.} & \mu_i \in [\lambda_i - c_p, \lambda_{i+n-p} - c_p]. \end{array}$$

The objective function of the latter problem can be rewritten as $(\sum_{i=1}^p \mu_i)^2 = (e^T \mu)^2 = \mu^T e e^T \mu$ with $\mu = (\mu_1, \ldots, \mu_p)^T$ and $e \in \mathbb{R}^p$ the vector of ones. Then this problem is of the form of a convex quadratic programming problem

$$\begin{array}{ll} \min & \mu^T e e^T \mu \\ \text{s.t.} & \mu_i \in [\lambda_i - c_p, \lambda_{i+n-p} - c_p], \quad i = 1, \ldots, p. \end{array} \tag{4.14}$$

As the feasible set of this problem is closed, convex and not empty and the objective function is convex, however not strictly convex, a solution of (4.14) always exists but may not be unique.

If μ_* is the solution of (4.14) then from the above derivation it is clear that the function value of (4.2) at the solution is just $(\mu_*^T e)^2$. To obtain the solution of (4.2) it remains to determine $Y \in \mathsf{St}(n,p)$ such that

$$Y^T NY = \operatorname{diag}(\mu_* + c_p e),$$

which is found by the solution of (4.1) with $D = \operatorname{diag}(\mu_* + c_p e)$ and can be computed by means of Algorithm 4.4.1.

4.5.2 Active-Set Method for Convex Quadratic Problems

Before we consider how to solve (4.14) we will briefly introduce the primal active-set method for convex quadratic problems described in [102, Section 16.5] and we will apply it to (4.14) in the next section. In general active-set methods deal with convex

programming problems of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & q(x) := x^T G x + x^T c \\ \text{s.t.} \quad & a_i^T x = b_i, \quad i \in \mathcal{E}, \\ & a_i^T x \geq b_i, \quad i \in \mathcal{I}, \end{aligned} \tag{4.15}$$

where $G \in \mathcal{S}_n^+$, $c \in \mathbb{R}^n$ and $\mathcal{E}$ and $\mathcal{I}$ are finite sets of indices indicating the equality and inequality constraints, respectively. Furthermore, for all $i \in \mathcal{I} \cup \mathcal{E}$ $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. Let $\mathcal{A}(x) \subset \mathcal{E} \cup \mathcal{I}$ be the active set, the set of indices of the equality and inequality constraints that are active at x, i.e. $a_i^T x = b_i$ for all $i \in \mathcal{A}(x)$.

As we do not know the active set at the solution x_* in general the idea is to solve iteratively convex quadratic programming subproblems with only those constraints that are in $\mathcal{E}$ or a subset of the active set at the current iterate x_k, transformed into equality constraints. We call this subset the working set and denote it by $\mathcal{W}_k$. Assuming the convex quadratic programming subproblems have always a solution, which can easily be obtained by the Lagrangian method, solving these subproblems yields either zero or a direction d_k along which the constraints in the current working set are not violated and for which $q(x_k) \geq q(x_k + \alpha d_k)$ for small $\alpha > 0$.

To demonstrate this, let $d := x - x_k$. The objective of (4.15) becomes then $d^T G d + g_k^T d + z_k$ where $g_k = 2Gx_k + c$ and $z_k = x_k^T G x_k + c^T x_k$ and for all $i \in \mathcal{W}_k$ the constraints are $a_i^T d = 0$. Thus to find the direction that gives the largest reduction in the objective function we have to solve the convex quadratic problem

$$\begin{aligned} \min_d \quad & d^T G d + g_k^T d \\ \text{s.t.} \quad & a_i^T d = 0, \quad i \in \mathcal{W}_k, \end{aligned} \tag{4.16}$$

in every iteration. Let d_k be the solution of (4.16). All constraints in $\mathcal{W}_k$ are satisfied for the new iterate $x_{k+1} = x_k + \alpha_k d_k$ with $\alpha_k \in [0, 1]$ as $a_i^T x_{k+1} = b_i + \alpha_k a_i^T d_k = b_i$.

If d_k is nonzero a maximal step length $\alpha_k \in [0, 1]$ is chosen such that all other constraints are not violated. If there is a blocking constraint it will be added to the working set of the next iterate. If α_k is one and no blocking constraint exists the solution of the convex quadratic programming in the next iteration will be zero, which is the first case. As in this situation the first-order optimality condition for (4.16) is satisfied it holds that for some Lagrange multipliers θ_i, $i \in \mathcal{W}_k$

$$\sum_{i\in\mathcal{W}_k} a_i \theta_i = g_k = 2Gx_k + c. \tag{4.17}$$

If all Lagrange multipliers θ_i for $i \in \mathcal{W}_k \cap \mathcal{I}$ are nonnegative from [102, Section 16.5] follows x_k is a KKT point of (4.15) and thus a global solution.

If there are one or more Lagrange multipliers θ_i for $i \in \mathcal{W}_k \cap \mathcal{I}$ that are negative then one constraint with a negative Lagrange multiplier is removed from the working

set. The next theorem states that under certain conditions the solution of the next subproblem will yield a direction for $q(\cdot)$ along which the dropped constraint will be satisfied.

Theorem 4.5.1. *Let x_k satisfy the first-order condition for the equality-constrained subproblem with working set $\mathcal{W}_k$, that is, (4.17) holds for x_k and $a_i^T x_k = b_i$ for all $i \in \mathcal{W}_k$. Let further the constraint gradients a_i for all $i \in \mathcal{W}_k$ be linearly independent and assume there is an index $j \in \mathcal{W}_k \cap \mathcal{I}$ with $\theta_j < 0$. Let d be the solution of (4.16) for $\mathcal{W}_{k+1} \setminus \{j\}$. Then $a_j^T d \geq 0$.*

Proof. This is proven by the first part of [102, Theorem 16.5]. □

The next theorem shows that the active-set method will converge to a global solution in a finite number iterations under certain assumptions.

Theorem 4.5.2. *Suppose that whenever the solution d_k of the subproblem (4.16) is nonzero d_k is a descent direction for $q(\cdot)$ and the method takes a nonzero step length $\alpha_k > 0$. Suppose further that if $d_k = 0$ and there exists a Lagrange multiplier $\theta_j < 0$ with $j \in \mathcal{W}_k \cap \mathcal{I}$ then d_{k+1} will be nonzero. Then the active-set method converges to a global solution in a finite number of iterations.*

Proof. The proof of this theorem follows from the discussion on [102, page 477]. □

If the method cannot always take a nonzero step length α_k whenever d_k computed from (4.16) is nonzero the algorithm may undergo cycling. This refers to the situation when after a certain number $l > 0$ of iterates there is no movement in $x_k = x_{k+l}$ and $\mathcal{W}_k = \mathcal{W}_{k+l}$. However, there are techniques that prevent the algorithm from cycling. We will not go into detail and direct the reader to [102, Chapter 13]. In the case of G positive definite [102, Theorem 12.5] together with [102, Theorem 12.6] show that if $d_k \neq 0$ it will be a descent direction for $q(\cdot)$. Moreover, if $d_k = 0$ and $\theta_j < 0$ for a $j \in \mathcal{W}_k \cap \mathcal{I}$ the computed direction d_{k+1} in the next iteration will be a descent direction.

4.5.3 Applying Active-Set Method to Problem 2

In this section we apply the active-set method to (4.14) and show that no linear system needs to be solved to find a solution of the convex quadratic programming subproblems. Further, it will turn out that the active-set method always converges to an optimal solution of (4.14) in at most $2p$ iterations.

Let us first transform the box constraints of (4.14) into linear inequality constraints so that this problem is of the form of (4.15). We obtain

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & x^T e e^T x \\ \text{s.t.} \quad & e_i^T x \geq \lambda_i - \delta_p, \\ & -e_i^T x \geq -\lambda_{i+n-p} + \delta_p, \quad i = 1, \ldots, p. \end{aligned} \tag{4.18}$$

We assume that $\lambda_i \neq \lambda_{i+n-p}$ for all $i = 1, \ldots, p$. This is no restriction as all elements of x with $\lambda_i = \lambda_{i+n-p}$ are fixed so that the programming problem can be reduced to an equivalent programming problem satisfying the assumption. As a consequence the constraint gradients of all inequalities in $\mathcal{A}(x)$ are either of the form $-e_i$ or e_i and are linearly independent at all x in the feasible set.

Let us now consider how to solve the convex quadratic subproblem without solving a linear system, which is the result of the next lemma.

Lemma 4.5.3. *Let $\mathcal{W}_k$ be the current working set of iteration k and $r := |\mathcal{W}_k|$. Let further $A_k \in \mathbb{R}^{p\times r}$ denote the matrix whose columns are the constraint gradients of the subproblem corresponding to $\mathcal{W}_k$. As all the constraints of the subproblem are equality constraints the constraint gradients a_i of the subproblem are assumed to be of the form of e_i for $i \in \{1, \ldots, p\}$. Therefore there exists a permutation matrix $P_k \in \mathbb{R}^{p\times p}$ such that $P_k^T A_k = [\ I_r \quad 0\ ]^T$. Then*

$$d_k := \begin{cases} P_k \begin{bmatrix} 0_{r\times 1} \\ -\frac{e^T x_k}{p-r} e \end{bmatrix} & \text{for } r < p \\ 0 & \text{otherwise} \end{cases} \tag{4.19}$$

is an optimal solution of the convex quadratic subproblem (4.16) *for* (4.18) *where $0_{r\times 1} = [0, \ldots, 0]^T \in \mathbb{R}^r$. If $r < p$ all Lagrange multipliers corresponding to the inequalities in $\mathcal{W}_k$ will be zero.*

Proof. Let $\iota_k : \mathcal{W}_k \mapsto \{1, \ldots, p\}$ be the function that maps the index of the constraint to the corresponding index of the unit basis vector, that is $e_{\iota_k(j)} = a_j$ for $j \in \mathcal{W}_k$. We now consider the Lagrangian function of (4.16) for our particular problem

$$L(d, \theta) = d^T e e^T d + (2 e e^T x_k)^T d + \sum_{j\in\mathcal{W}_k} \theta_{\iota_k(j)} e_{\iota_k(j)}^T d$$

where $\theta = (\theta_{\iota_k(j)})_{j\in\mathcal{W}_k}$ are the Lagrange multipliers. As this function is convex the condition for $d \in \mathbb{R}^p$ to be a global solution is thus

$$\begin{bmatrix} 2ee^T & A_k \\ A_k^T & 0 \end{bmatrix} \begin{bmatrix} d \\ \theta \end{bmatrix} = \begin{bmatrix} -2ee^T x_k \\ 0 \end{bmatrix} \tag{4.20}$$

If $r = p$ the columns of A_k form a basis of $\mathbb{R}^p$. Hence, $A_k^T d = 0$ has a unique solution, that is $d_k = 0$.

Let us now assume $r < p$. Substituting d_k of (4.19) in (4.20) we obtain for the left-hand side

$$\begin{bmatrix} 2ee^T & A_k \\ A_k^T & 0 \end{bmatrix} \begin{bmatrix} P_k \begin{bmatrix} 0_{r\times 1} \\ -\frac{e^T x_k}{p-r} e \end{bmatrix} \\ \theta \end{bmatrix} = \left[\begin{array}{cc|c} 2ee^T & & A_k \\ \hline I_r & 0 & 0 \end{array}\right] \left[\begin{array}{c} 0_{r\times 1} \\ -\frac{e^T x_k}{p-r} e \\ \hline \theta \end{array}\right] \tag{4.21}$$

$$= \begin{bmatrix} -2e^T x_k e + A_k \theta \\ 0 \end{bmatrix}$$

For $\theta = 0$, (4.21) is equal to the right-hand side of (4.20) and thus d_k is an optimal solution of (4.16) for our particular problem. □

Note that the direction d_k chosen in Lemma 4.5.3 is a descent direction for $q(\cdot)$ whenever it is nonzero. The reason is that for $d_k \neq 0$ $d_k^T G d_k = (e^T x_k)^2 > 0$ as $e^T x_k$ is only zero for $d_k = 0$. Since d_k is a solution of (4.16) we have

$$q(x_k + d_k) = q(x_k) + g_k^T d_k + d_k^T G d_k \leq q(x_k).$$

Hence $g_k^T d_k < 0$ and thus d_k is a descent direction.

The statement of the next lemma is needed to show subsequently that the active-set method takes at most $2p$ iterations for (4.18).

Lemma 4.5.4. *Let j be a blocking constraint that is added to the working set in iteration k in the active-set method for* (4.18)*. Then this constraint will not be removed from the working set in the algorithm.*

Proof. Let us assume that $l > k$ is the iteration number when the constraint j is removed from the working set $\mathcal{W}_l$. By Lemma 4.5.3 a constraint is only removed if $r = p$. Therefore by Theorem 4.5.1 we have that

$$a_j^T d_{l+1} > 0. \tag{4.22}$$

Further, as j was a blocking constraint at iteration k it holds that

$$a_j^T d_k = -a_j^T P_k \begin{bmatrix} 0_{r\times 1} \\ \frac{e^T x_k}{p-r} e \end{bmatrix} \leq 0. \tag{4.23}$$

As we minimize the function $(e^T x)^2$, $e^T x_k$ and $e^T x_{l+1}$ must have the same sign. Thus from (4.23)

$$a_j^T d_{l+1} = -a_j^T P_{l+1} \begin{bmatrix} 0_{r\times 1} \\ e^T x_{l+1} e \end{bmatrix} \leq 0,$$

which is a contradiction to (4.22). □

Now we are ready to prove the active-set method converges to the global solution in at most $2p$ iterations.

Theorem 4.5.5. *Let the directions d_k be chosen as in* (4.19) *then the active-set method converges to a global solution of* (4.18) *and terminates in at most* $2p$ *iterations.*

Proof. We need to show that the active-set method converges in at most $2p$ iterations to a point x_* and working set $\mathcal{W}_*$ where the convex quadratic subproblem (4.16) has the solution $d_* = 0$ and all corresponding Lagrange multipliers θ_i with $i \in \mathcal{W}_k$ are nonnegative [102, Section 16.5]. Let again $r := |\mathcal{W}_k|$ and let us first assume $r < p$.

At iteration k, either d_k is nonzero or d_k is zero, which implies according to Lemma 4.5.3 that all Lagrange multipliers are zero so that x_k is an optimal solution of (4.18). In the former case a constraint will be added to the working set $\mathcal{W}_k$ or in the subsequent iteration the new direction d_{k+1} is equal to zero, implying, unless $r = p$, that an optimal solution is found. Therefore, for $r < p$ in every iteration one constraint is added to the working set and no constraint is removed until $r = p$ or an optimal solution has been found. Let m be the number of iterations until $r = p$. Note that m is bounded from above by p.

Let us now consider $r = p$ and assume that there exists a Lagrange multiplier with $\theta_j < 0$ for $j \in \mathcal{W}_k$. As $d_k = 0$ one of these Lagrange multipliers with $\theta_j < 0$ will be removed from the working set $\mathcal{W}_k$. Since in the next iteration we will have $r = p - 1 < p$ we obtain for d_{k+1} either zero and an optimal solution is found or according to Theorem 4.5.1 a descent direction for $q(\cdot)$ along which the inequality j is satisfied. If a blocking constraint exists then this constraint will be added to the new working set and $r = p$, otherwise an optimal solution is found. By Lemma 4.5.4 this procedure can happen at most $p - m$ times, requiring at most $2(p - m)$ additional iterations until an optimal solution is found. Thus, in total we have at most $2(p - m) + m$ iterations and as m can be zero, the algorithm takes at most $2p$ iterations. Note that the factor 2 results from the iteration where $r = p$ and one constraint is removed from the working set, and the subsequent iteration where a reduction of the objective function is achieved. □

As a consequence of Theorem 4.5.5 we obtain an algorithm that terminates in at most $2p$ iterations and returns an optimal solution of (4.14). We state the active-set method [102, Algorithm 16.3] for our particular problem in Algorithm 4.5.1.

Algorithm 4.5.1 Active-set method for computing the solution of (4.14).

Require: $p, n \in \mathbb{N}$ and $p \leq n$, c_p, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with $\lambda_1 \leq \cdots \leq \lambda_n$.

Determine the range for $x \in [l, u]$. $l := (\lambda_1 - c_p, \ldots, \lambda_p - c_p)^T$, $u := (\lambda_{1+n-p} - c_p, \ldots, \lambda_n - c_p)^T$.
Reduce the corresponding convex quadratic programming by removing all elements of x with $l_i = u_i$.
Set $k := 0$, determine a feasible starting value x_0, e.g. $x_0 = u$ and set $\mathcal{W}_0$ to be a subset of the active set at x_0.
loop
  Compute solution d_k of the current subproblem (4.16) by means of (4.19).
  if $d_k = 0$ **then**
    if $|\mathcal{W}_k| \neq p$ **then**
      break
    else
      Compute Lagrange multipliers θ_i corresponding to the inequalities of $\mathcal{W}_k$.
      if all Lagrange multipliers are nonnegative **then**
        break
      else
        $j := \operatorname{argmin}_{j \in \mathcal{W}_k \cap \mathcal{I}} \theta_j$
        Set $\mathcal{W}_{k+1} := \mathcal{W}_k \setminus \{j\}$.
      end if
    end if
  else
    Compute $\alpha_k := \min\left\{1, \min_{i \notin \mathcal{W}_k, a_i^T d_k < 0} \frac{b_i - a_i^T x_k}{a_i^T d_k}\right\}$.
    Set $x_{k+1} = x_k + \alpha_k d_k$.
    if there is a blocking constraint j **then**
      $\mathcal{W}_{k+1} := \mathcal{W}_k \cup \{j\}$
    else
      $\mathcal{W}_{k+1} := \mathcal{W}_k$
    end if
    $k := k + 1$
  end if
end loop
return x_k.

4.6 Optimizing Arbitrary Smooth Functions over Set of Optimal Solutions

4.6.1 Introduction

In Section 4.4 and 4.5 we solved the problems that were introduced in Section 4.2. However, the solutions obtained are generally not unique. For (4.1) we have shown that the set of optimal solutions is equivalent to

$$\mathcal{C} := \left\{ Y \in \mathsf{St}(n,p) : Y^T \Lambda Y = D \right\} \tag{4.24}$$

with $D = \Delta_*$ defined in (4.7) and Λ the diagonal matrix with the eigenvalues of N on its diagonal. Moreover, we have seen in Section 4.5 that this set plays an important role in solving (4.2), too.

To select a particular solution out of the set in (4.24) the idea is to pose a new optimization problem. We therefore establish a new framework in this section that allows the optimization of an arbitrary smooth function f over the set (4.24). Depending on the application, this function should then be chosen such that the minimum value of f is attained at the points of interest in (4.24). Our approach assumes that the diagonal elements of D are distinct and in increasing order.

We will first consider a set that imposes p fewer constraints on $Y \in \mathsf{St}(n,p)$ than $\mathcal{C}$ but can easily been proven to be a Riemannian manifold. We will then show that all geometric objects can be derived to make an optimization over this manifold possible by using optimizing algorithms that are applicable. See Chapter 3 for an introduction to optimization over Riemannian manifolds. To optimize eventually over $\mathcal{C}$ it remains to impose the p constraints that we have disregarded. We tackle this problem by applying the augmented Lagrangian method [13, Section 4.2], [102, Chapter 17].

4.6.2 Modified Constraint Set Forming Riemannian Manifold

Let us now define the new constraint set as

$$\mathcal{B}(n,p) = \left\{ Y \in \mathsf{St}(n,p) : \operatorname{offdiag}(Y^T \Lambda Y) = 0 \text{ and } (Y^T \Lambda Y)_{11} < \cdots < (Y^T \Lambda Y)_{pp} \right\}$$

where $\operatorname{offdiag} : \mathbb{R}^{p \times p} \mapsto \mathbb{R}^{p(p-1)}$ is the operator that stacks the off-diagonals into a long vector starting from the most upper right. Note that $\mathcal{B}(n,p)$ does not impose the constraints that the diagonal elements of $Y^T \Lambda Y$ coincide with the diagonal elements of D. The idea is to impose these constraints separately in our optimization routine.

Constraint Set As Embedded Submanifold of $\mathbb{R}^{n\times p}$

Let us now show that $\mathcal{B}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n\times p}$.

Lemma 4.6.1. *The set $\mathcal{B}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n\times p}$ with dimension $np-p^2$.*

Proof. Let Y be an element in $\mathcal{B}(n,p)$. Then there exists an open neighbourhood U_Y of Y in $\mathbb{R}^{n\times p}$ such that the diagonal elements of $X^T\Lambda X$ are distinct for all $X\in U_Y$. Let $U=\bigcup_{Y\in\mathcal{B}(n,p)}U_Y$. As U_Y is an open subset of $\mathbb{R}^{n\times p}$ for all $Y\in\mathcal{B}(n,p)$ the set U is also an open subset of $\mathbb{R}^{n\times p}$. Therefore from the discussions in Section 3.2.2, U is an open submanifold of $\mathbb{R}^{n\times p}$ of dimension np.

Consider the function $F:U\mapsto\mathcal{S}^p\times\mathcal{S}_0^p$ with

$$F(X)=\begin{bmatrix}X^TX-I_p\\ X^T\Lambda X-\operatorname{diag}\left(X^T\Lambda X\right)\end{bmatrix}, \tag{4.25}$$

where $\mathcal{S}_0^p:=\{Z\in\mathcal{S}^p:\operatorname{diag}(Z)=0\}$. Then by construction it holds that $F^{-1}(0)=\mathcal{B}(n,p)$. Now in order to apply Theorem 3.4.2 we need to show that F is a submersion at all $X\in\mathcal{B}(n,p)$.

Let $\Theta=X^T\Lambda X$ with $X\in\mathcal{B}(n,p)$. Let further $S=\begin{bmatrix}S_1 & S_2\end{bmatrix}^T\in\mathcal{S}^p\times\mathcal{S}_0^p$ be arbitrary and $\widehat{Z}=\frac{1}{2}X(S_1+K)$ with $K\in\mathcal{K}_p$ and

$$K_{ij}=\begin{cases}\frac{(\Theta S_1+S_1\Theta-2S_2)_{ij}}{\Theta_{jj}-\Theta_{ii}} & \text{for } i\neq j\\ 0 & \text{otherwise.}\end{cases}$$

Then from

$$DF(X)[Z]=\begin{bmatrix}X^TZ+Z^TX\\ 2\operatorname{sym}(X^T\Lambda Z)-2\operatorname{diag}\left(\operatorname{sym}\left(X^T\Lambda Z\right)\right)\end{bmatrix}$$

we have that $DF(X)[\widehat{Z}]=\begin{bmatrix}S_1 & S_2\end{bmatrix}^T$. As the matrix S was chosen arbitrarily F is of full rank at all $X\in\mathcal{B}(n,p)$, which implies that 0 is a regular value of F. Hence, by Theorem 3.4.2 $\mathcal{B}(n,p)$ is an embedded submanifold of U with dimension $\dim(U)-\dim(\mathcal{S}^p\times\mathcal{S}_0^p)=np-p^2$. As U covers $\mathcal{B}(n,p)$ by Definition 3.4.1 $\mathcal{B}(n,p)$ is an embedded submanifold of $\mathbb{R}^{n\times p}$. ∎

Note that $\mathcal{B}(n,p)$ can likewise be considered as embedded submanifold of $\mathsf{St}(n,p)$. The Riemannian manifold $\mathcal{B}(n,p)$ is bounded as each column of $Y\in\mathcal{B}(n,p)$ has 2-norm one, implying that $\|Y\|_F=\sqrt{p}$. However, it is not closed in $\mathbb{R}^{n\times p}$ as demonstrated by the following example. Let $\{\varepsilon_k\}_{k\geq 0}$ be a sequence with $\varepsilon_k\searrow 0$ as $k\to\infty$.

Let $\Lambda = \operatorname{diag}(1, 1, 2)$ and $\{Y_k\}_{k\geq 0}$ be a sequence with

$$Y_k = \begin{bmatrix} 1 & 0 \\ 0 & (1-\varepsilon_k)/s_k \\ 0 & \varepsilon_k/s_k \end{bmatrix} \text{ where } s_k = \sqrt{\varepsilon_k^2 + (1-\varepsilon_k)^2}.$$

Then Y_k is in $\mathcal{B}(3,2)$ for all k as $Y_k^T \Lambda Y_k = \operatorname{diag}\left(1, \frac{2\varepsilon_k^2+(1-\varepsilon_k)^2}{\varepsilon_k^2+(1-\varepsilon_k)^2}\right)$. However $Y_* := \lim_{k\to\infty} Y_k \notin \mathcal{B}(3,2)$ as $Y_*^T \Lambda Y_* = \operatorname{diag}(1,1)$.

Set of Constructed Solutions Connected on Manifold

By means of the next lemma we prove that the optimal solutions of (4.1) for different values of D that are obtained by Algorithm 4.4.1 are connected on the manifold.

Lemma 4.6.2. *Let Λ be the diagonal matrix with the eigenvalues of N in (4.1) on its diagonal. Suppose the elements of Λ are distinct. Let $\widehat{D}, \widetilde{D} \in \mathbb{R}^{p\times p}$ be two diagonal matrices that are imbeddable in Λ and that have strictly increasing diagonal elements. Then the optimal solutions of (4.1) for $D = \widehat{D}$ and $D = \widetilde{D}$, respectively, obtained by Algorithm 4.4.1 are connected on $\mathcal{B}(n,p)$.*

Proof. We need to show that there exists a curve $Y : [a,b] \mapsto \mathcal{B}(n,p)$ such that $Y(a) = Y_1$ and $Y(b) = Y_2$ where Y_1, Y_2 are computed by Algorithm 4.4.1 with $Y_1^T \Lambda Y_1 = \widehat{D}$ and $Y_2^T \Lambda Y_2 = \widetilde{D}$. Without loss of generality we can assume that the diagonal elements of Λ are in increasing order, that is $\lambda_1 < \lambda_2 < \cdots < \lambda_n$. As the diagonal elements of $\widehat{D}$ and $\widetilde{D}$ are in strictly increasing there exists a curve $D(t)$ with $D(t)$ diagonal, $D(a) = \widehat{D}$, $D(b) = \widetilde{D}$, and $D_{ii}(t) \neq D_{jj}(t)$ for $i \neq j$ and all $t \in [a,b]$. For example $D(t)$ defined as $D_{ii}(t) := \widehat{D}_{ii}\frac{b-t}{b-a} + \widetilde{D}_{ii}\frac{t-a}{b-a}$ for all i is such a curve. Then from the proof of Lemma 4.4.2 there exist two permutation matrices P and Q independent of t such that $U^T \Lambda U = \operatorname{diag}(\Lambda_1, \ldots, \Lambda_{q+1})$ and $Q^T D(t) Q = \operatorname{diag}(D_1(t), \ldots, D_q(t))$ with $q = \min\{n-p, p\}$ and Λ_i interlacing $D_i(t)$ for all $i = 1, \ldots, q$ and all $t \in [a,b]$. As $D_i(t)$ is smooth the curve of arrowhead matrices $A_i(t)$ whose upper left part coincides with $D_i(t)$ and whose eigenvalues are the diagonal element of Λ_i is also smooth for all $i = 1, \ldots, q$. Since all diagonal elements of Λ_i are distinct we can apply [43, Proposition 2.4] and obtain a smooth spectral decomposition of $A_i(t) = V_i(t)\Lambda_i V_i(t)^T$ for all i. Hence, $Y(t) := U\widehat{Y}(t)Q^T$ with

$$\widehat{Y}(t) = \operatorname{diag}\left(V_1(t)^T [I_{s_1-1}\ 0]^T, \ldots, V_q(t)^T [I_{s_q-1}\ 0]^T\right)$$

is the curve that we are looking for as it is smooth for all $t \in [a,b]$ and $Y(a)^T \Lambda Y(a) = \widehat{D}$ and $Y(b)^T \Lambda Y(b) = \widetilde{D}$ where $Y(a)$ and $Y(b)$ is computed by Algorithm 4.4.1. ∎

Note that the assumption of Lemma 4.6.2 that the diagonal elements of Λ are distinct can be relaxed. If Λ_i as determined in the proof of Lemma 4.6.2 has diagonal elements that are not distinct then the problem is that we cannot apply [43, Proposition 2.4] to find a smooth spectral decomposition of $A_i(t) = V_i(t)\Lambda_i V_i(t)^T$. However, we can overcome this problem. For simplicity reasons we assume that Λ_i has only two diagonal elements μ_1 and μ_2 with the same value. If one or more elements are of multiple occurrence then the same procedure can be applied. As Λ_i interlaces $D_i(t)$ there exists a diagonal element $d(t) = (D_i(t))_{jj}$ with $\mu_1 = d(t) = \mu_2$. Hence, $d(t)$ is constant for all $t \in [a, b]$. If we now diagonalize the arrowhead matrix $A_i(t)$ in our Algorithm 4.4.1 as follows we still obtain a smooth spectral decomposition of $A_i(t) = V_i(t)\Lambda_i V_i(t)^T$. First we apply [43, Proposition 2.4] to the minor $A_i(j, j)$ yielding a smooth decomposition $W(t)\Lambda_i(j, j)W(t)$ of $A_i(j, j)$. Then we set $V_i(t)(j, j) := W(t)$ and the jth row and column of $V_i(t)$ to e_j and e_j^T, respectively. As $d(t)$ is constant $V_i(t)$ diagonalizes $A_i(t)$ and we obtain our smooth spectral decomposition.

4.6.3 Geometric Objects of this Manifold

The Tangent Space

Let us now consider the tangent space of $\mathcal{B}(n, p)$ for whose definition we need the operator $A : \mathbb{R}^{(n-p)\times p} \mapsto \mathcal{K}_p$ at $Y \in \mathcal{B}(n, p)$ with

$$A_{ij}(Z) = \frac{2\mathrm{sym}(Y^T \Lambda Y_\perp Z)_{ij}}{(Y^T \Lambda Y)_{jj} - (Y^T \Lambda Y)_{ii}} \tag{4.26}$$

for $i \neq j$ and $A_{ii} = 0$ for all $i = 1, \ldots, p$.

Lemma 4.6.3. *The tangent space $T_Y\mathcal{B}(n, p)$ of $\mathcal{B}(n, p)$ at $Y \in \mathcal{B}(n, p)$ is*

$$\mathcal{N} = \left\{ Z = YA(B) + Y_\perp B : B \in \mathbb{R}^{(n-p)\times p} \text{ free.} \right\}. \tag{4.27}$$

Proof. Let $Y(t)$ be a curve in $\mathcal{B}(n, p)$ with $Y(0) = Y$. From the argument in Section 3.8.1 it is clear that the condition $Y^T Y = I_p$ imposes $p(p + 1)/2$ constraints on $Y'(0)$. Let us now differentiate $\mathrm{offdiag}(Y^T \Lambda Y) = 0$ with respect to t. We obtain that

$$\mathrm{offdiag}\left(\mathrm{sym}\left(Y^T \Lambda Y'(0)\right)\right) = 0, \tag{4.28}$$

imposing another $p(p - 1)/2$ constraints on $Y'(0)$. Hence, the vector space $\mathbb{R}^{n\times p}$ imposing these constraints has dimension $np - p(p + 1)/2 - p(p - 1)/2 = np - p^2$. This is obviously the same as the dimension of $\dim(T_Y\mathcal{B}(n, p))$ and the dimension of (4.27). Therefore it is enough to show that all elements $Z \in \mathcal{N}$ satisfy $Y^T Z$ skew-symmetric and $\mathrm{offdiag}\left(\mathrm{sym}\left(Y^T \Lambda Z\right)\right) = 0$. Substituting $Z = YA + Y_\perp B$ with A as chosen in (4.27) verifies the claim. □

Now we endow all tangent spaces of $\mathcal{B}(n,p)$ with the Euclidean inner product $\langle A, B\rangle := \operatorname{trace}(B^T A)$ and obtain a Riemannian submanifold of $\mathbb{R}^{n\times p}$.

The Normal Space

Lemma 4.6.4. *The normal space of $\mathcal{B}(n,p)$ at Y is given by*

$$\mathcal{N} := \Big\{ Z \in \mathbb{R}^{n\times p} : Z = \Lambda Y(C + C^T) - Y(CD + DC^T - T) \\ \text{for } C, T \in \mathbb{R}^{p\times p} \text{ with } C_{ii} = 0 \text{ and } T \text{ diagonal} \Big\}.$$

Proof. First, $\mathcal{N}$ has dimension p^2 as $Y^T Z(C,T) = DC - CD + T$ is of dimension p^2 for $Z(C,T) := \Lambda Y(C + C^T) - Y(CD + DC^T - T) \in \mathcal{N}$ and as the variables of freedom in $\mathcal{N}$ is also $\dim(C) + \dim(T) = p^2$. Now let $Z \in \mathcal{N}$ be arbitrary. Since for all $V = YA(B) + Y_\perp B \in T_Y\mathcal{B}(n,p)$ with $B \in \mathbb{R}^{(n-p)\times p}$

$$\begin{aligned} \operatorname{trace}(V^T Z) &= \operatorname{trace}\big(\big(A(B)^T Y^T + B^T Y_\perp^T\big)\big(\Lambda Y(C + C^T) \\ &\quad - Y(CD + DC^T - T)\big)\big) \\ &= \operatorname{trace}\big(A(B)^T D(C + C^T) - A(B)^T(CD + DC^T - T) \\ &\quad + B^T Y_\perp^T \Lambda Y(C + C^T)\big) \\ &= \operatorname{trace}\big(\big(A(B)^T D + DA(B) + B^T Y_\perp^T \Lambda Y + Y^T \Lambda Y_\perp B\big)C\big) \\ &= 0 \end{aligned} \tag{4.29}$$

we have that Z is an element of the normal space at Y. The latter equality in (4.29) holds as $A(B)^T D + DA(B) + B^T Y_\perp^T \Lambda Y + Y^T \Lambda Y_\perp B = 0$ by the definition of $A(\cdot)$ in (4.26). ∎

Projection onto Tangent and Normal Space

To compute the projection of an element $Z \in \mathbb{R}^{n\times p}$ onto the tangent space $T_Y\mathcal{B}(n,p)$ it is required to determine the element $Z_p \in T_Y\mathcal{B}(n,p)$ satisfying

$$\big\langle Z - Z_p, YA(e_i e_j^T) + Y_\perp e_i e_j^T \big\rangle = 0 \qquad \text{for all } i = 1,\dots,n-p \text{ and } j = 1,\dots,p.$$

Hence, we need to solve a linear system of dimension $p \times (n-p)$. If $p \ll n$ then it is clearly less expensive to compute the projection onto the normal space at Y instead and subtract it from Z as this involves solving only a linear system of dimension p^2. Since we assume that p is small in comparison to n we devote yourselves to look at this projection more in detail. It will turn out that it is sufficient to solve a linear system of dimension $p(p-1)/2$ only.

Let $Q : \{X \in \mathbb{R}^{p\times p} : \operatorname{diag}(X) = 0\} \mapsto \mathbb{R}^{n\times p}$ be an operator with

$$Q(C) = \Lambda Y(C^T + C) - Y(CD + DC^T). \tag{4.30}$$

Then to find the projection of Z onto the normal space at Y the aim is to determine the element $Z_n \in N_Y\mathcal{B}(n,p)$ satisfying

$$\begin{aligned} \left\langle Z - Z_n, Q(e_ie_j^T)\right\rangle &= 0 \quad \text{for all } i \neq j \text{ and} \\ \left\langle Z - Z_n, Ye_ie_i^T\right\rangle &= 0 \quad \text{for all } i = 1,\dots,p. \end{aligned}$$

Since we compute the projection onto a Hilbert space this element exists and is unique. Let $\mathcal{H} \in \mathbb{R}^{p\times p\times p\times p}$ be a tensor with

$$\mathcal{H}_{i,j,k,l} = \begin{cases} \left\langle Q(e_ie_j^T), Q(e_ke_l^T)\right\rangle & \text{for } i \neq j, k \neq l \\ \left\langle Ye_ie_j^T, Q(e_ke_l^T)\right\rangle & \text{for } i = j, k \neq l \\ \left\langle Q(e_ie_j^T), Ye_ke_l^T\right\rangle & \text{for } i \neq j, k = l \\ \left\langle Ye_ie_j^T, Ye_ke_l^T\right\rangle & \text{for } i = j, k = l \end{cases} \tag{4.31}$$

and let $B \in \mathbb{R}^{p\times p}$ be defined as

$$B_{ij} = \begin{cases} \left\langle Z, Q(e_ie_j^T)\right\rangle & \text{for } i \neq j \\ \left\langle Z, Ye_je_i^T\right\rangle & \text{for } i = j. \end{cases}$$

Then with $H \in \mathbb{R}^{p^2\times p^2}$ being the unfolding of the tensor $\mathcal{H}$ in mode 1 and 2 along the rows and mode 3 and 4 along the columns and $b \in \mathbb{R}^{p^2}$ the mode 1 unfolding of B the linear system that needs to be solved to compute the projection Z_n is $Hz = b$. The vector $z \in \mathbb{R}^{p^2}$ is related to Z_n as follows. Let $C \in \mathbb{R}^{p\times p}, T \in \mathbb{R}^{p\times p}$ be defined as

$$C := \sum_{i\neq j} z_{((j-1)p+i)}e_ie_j^T \text{ and } T := \sum_{i=1}^{p} z_{((i-1)p+i)}e_ie_i^T \tag{4.32}$$

then $Z_n = Q(C) + YT$.

First we note that only the right-hand side b depends on Z. Hence, a multiple projection onto the same normal space is not of much higher cost than a single projection. Second, we show by the next lemma that it is sufficient to solve $Hz = b$ by determining the solution of a smaller linear system of dimension $q := (p-1)p/2$. Let us first define the map $\iota : \{2,\dots,p\} \times \{1,\dots,p-1\} \mapsto \{1,\dots,q\}$ with $\iota(i,j) = (j-1)(p-1-j/2) + i - 1$.

Lemma 4.6.5. *The linear system $Hz = b$ is equivalent to a linear system $\widetilde{H}\widetilde{z} = \widetilde{b}$ with $\widetilde{z}, \widetilde{b} \in \mathbb{R}^{p^2}$ and $\widetilde{H} \in \mathbb{R}^{p^2\times p^2}$ being of the form*

$$\widetilde{H} = \left[\begin{array}{cc|c} S & 0_{q\times p} & \widetilde{H}_1 \\ 0_{p\times q} & I_p & \\ \hline & 0_{q\times(q+p)} & \widetilde{H}_2 \\ & & \end{array}\right] \tag{4.33}$$

where $S \in \mathbb{R}^{q\times q}$ *is a diagonal matrix with* $S_{\iota(i,j),\iota(i,j)} = (D_{ii} - D_{jj})^2$ *for* $i > j$ *and* $i, j = 1, \dots, p$ *and* $\widetilde{H}_1 \in \mathbb{R}^{(q+p)\times q}, \widetilde{H}_2 \in \mathbb{R}^{q\times q}$ *are full matrices.*

Proof. Let us first look at the $((l-1)p+k)$th column of H for $k \neq l$, which corresponds in MATLAB notation to $\mathcal{H}(:,:,k,l)$. From (4.31) we have for $i \neq j$

$$\begin{aligned}\mathcal{H}_{i,j,k,l} = &\operatorname{trace}\left((e_k e_l^T + e_l e_k^T)(Y^T \Lambda \Lambda Y + D_{ll} D_{jj} I_p)(e_j e_i^T + e_i e_j^T)\right)\\ &- \operatorname{trace}\left((e_k e_l^T + e_l e_k^T)(D_{jj} - D_{ll})(D_{ii} e_i e_j^T + D_{jj} e_j e_i^T)\right)\end{aligned}$$

and for $i = j$

$$\mathcal{H}_{i,j,k,l} = 0.$$

Therefore for $k \neq l$

$$\begin{aligned}\mathcal{H}_{i,j,k,l} - \mathcal{H}_{i,j,l,k} &= \operatorname{trace}\left((e_k e_l^T + e_l e_k^T)(D_{ll} - D_{kk})(D_{jj} - D_{ii}) e_i e_j^T\right)\\ &= \begin{cases}(D_{ll} - D_{kk})^2 & \text{for } j = l, i = k\\ -(D_{ll} - D_{kk})^2 & \text{for } j = k, i = l\\ 0 & \text{otherwise,}\end{cases}\end{aligned}$$

which implies that there exists an invertible lower triangular matrix $L \in \mathbb{R}^{p^2\times p^2}$ and permutation matrices P_c, P_r such that

$$P_r L^{-1} H L P_c = \left[\begin{array}{c|c} S & * \\ \hline 0_{(q+p)\times q} & * \end{array}\right]. \tag{4.34}$$

Note that the lower triangular matrix L corresponds to subtracting the $((k-1)p+l)$th column of H from the $((l-1)p+k)$th column for all $k > l$. Similarly, L^{-1} corresponds to adding the $((i-1)p+j)$th row of H to the $((j-1)p+i))$th row for $j > i$. These row and columns operations yield together with the permutation matrices P_c, P_r the diagonal matrix S in the upper left corner in (4.34). By noticing that for $k \neq l$

$$\mathcal{H}_{i,j,k,l} = \begin{cases}1 & \text{for } i = j\\ 0 & \text{for } i \neq j\end{cases}$$

we see that there exists a further permutation matrix P such that $P^T P_r L^{-1} H L P_c P$ has the form of $\widetilde{H}$ in (4.33). Therefore by setting $\widetilde{b} := P^T P_r L^{-1} b$ and $\widetilde{z} := P_c^T P^T L^{-1} z$ we obtain our equivalent linear system $\widetilde{H}\widetilde{z} = \widetilde{b}$. ∎

Let $\widetilde{H}\widetilde{z} = \widetilde{b}$ be defined as in Lemma 4.6.5. Let further $\widetilde{z}_1, \widetilde{b}_1 \in \mathbb{R}^{q+p}, \widetilde{z}_2, \widetilde{b}_2 \in \mathbb{R}^q$ be defined as

$$\widetilde{z} = \begin{bmatrix}\widetilde{z}_1\\ \widetilde{z}_2\end{bmatrix} \quad \text{and} \quad \widetilde{b} = \begin{bmatrix}\widetilde{b}_1\\ \widetilde{b}_2\end{bmatrix}.$$

Then from Lemma 4.6.5 it follows that we can determine the vector z in $Hz = b$ by first solving

$$\widetilde{H}_2\widetilde{z}_2 = \widetilde{b}_2 \tag{4.35}$$

and

$$\begin{bmatrix} S & \\ & I_p \end{bmatrix} \widetilde{z}_1 = \widetilde{b}_1 - \widetilde{H}_1\widetilde{z}_2, \tag{4.36}$$

and computing z via $z = LPP_c\widetilde{z}$. The major cost is surely to solve the linear system $\widetilde{H}_2\widetilde{z}_2 = \widetilde{b}_2$, which is of order q. Fortunately, $\widetilde{H}_2$ has additional structure shown by the next lemma that can be exploited to solve this linear system.

Lemma 4.6.6. *The matrix $\widetilde{H}_2$ in Lemma 4.6.5 is symmetric and sparse for p large whereas the ratio R of the number of zeros to the total number of elements in $\widetilde{H}_2$ is*

$$R \geq 1 - \frac{4}{p-1} + \frac{6}{p(p-1)}, \quad p \neq 1. \tag{4.37}$$

Proof. From above it follows that

$$\begin{aligned} (\widetilde{H}_2)_{st} = \mathcal{H}_{i,j,k,l} + \mathcal{H}_{j,i,k,l} \quad & \text{for } 1 \leq i < j \leq p,\ 1 \leq k < l \leq p \\ & \text{and } s = \iota(j,i),\ t = \iota(l,k). \end{aligned} \tag{4.38}$$

Let $M := Y^T\Lambda\Lambda Y$. Then if we substitute $\mathcal{H}_{i,j,k,l}$ in (4.31) into (4.38) and simplify the equations we obtain that

$$\begin{aligned} (\widetilde{H}_2)_{st} &= \left\langle Q(e_ie_j^T), Q(e_ke_l^T + e_le_k^T) \right\rangle \\ &= \begin{cases} 0 & \text{for } i \neq k, i \neq l, j \neq k, j \neq l \\ 2M_{ki} & \text{for } i \neq k, i \neq l, j = l \\ 2M_{li} & \text{for } i \neq k, i \neq l, j = k \\ 2M_{kj} & \text{for } i = l, j \neq k, j \neq l \\ 2M_{lj} & \text{for } i = k, j \neq k, j \neq l \\ 2M_{ii} + 2M_{jj} - (D_{ii} + D_{jj})^2 & \text{for } i = k, j = l, \end{cases} \end{aligned} \tag{4.39}$$

implying that $(\widetilde{H}_2)_{st} = (\widetilde{H}_2)_{ts}$ as M is symmetric. It remains to show (4.37). From (4.39) it follows that we need to count the elements of $\widetilde{H}_2$ for which $i \neq k, i \neq l, j \neq k, j \neq l$. Let K be this number. In order to determine K, we are going to subtract from the number of all elements, the number of elements where $i = k$ as well as where $j = l \wedge k \neq i$ and where $i < j = k < l$ and $k < l = i < j$. We have

$$\begin{aligned} K &= q^2 - \left(\sum_{i=1}^{p-1} (p-i)^2 + \sum_{i=1}^{p-2} (p-i)(p-i-1) + 2\sum_{i=1}^{p-2} i(p-i-1) \right) \\ &= \frac{p(p-1)}{4}(p^2 - 5p + 6). \end{aligned}$$

Hence,

$$R = \frac{K}{q^2} = 1 - \frac{4}{p-1} + \frac{6}{p(p-1)}.$$

□

In addition to the sparsity of $\widetilde{H}_2$ we observed in our numerical tests that $\widetilde{H}_2$ was diagonally dominant. However unfortunately, we could not prove this observation. Nevertheless this gives support to the idea of using an iterative solver to determine $\widetilde{z}_2$ in (4.35) with the diagonal of $\widetilde{H}_2$ as preconditioner.

Note also that to obtain $\widetilde{z}_1$ in (4.36) we need to divide the first q elements of $\widetilde{b}_3 := \widetilde{b}_1 - \widetilde{H}_1\widetilde{z}_2$ by the diagonal elements of S. If the gaps between the diagonal elements in S are small then we need to divide by these small numbers squared that can cause numerical difficulties. The key observation to remedy this problem is that the first q elements of $\widetilde{z}_1$ describe the lower triangular part of C as defined in (4.32) and that the upper triangular part of C is described by again the first q elements of $\widetilde{z}_1$ and the elements of $\widetilde{z}_2$ as follows. Let $\widetilde{z}_{11}$ and $\widetilde{z}_{12}$ be the vectors with the first q and last p elements of $\widetilde{z}_1$, respectively. Let further $C_L, C_U \in \mathbb{R}^{p\times p}$ be defined as $C_L := \sum_{i>j} \widetilde{z}_{11}(\iota(i,j))e_i e_j^T$ and $C_U := \sum_{j>i} \widetilde{z}_2((j-2)(j-1)/2+i)e_i e_j^T$. Then $C = C_L + C_U - C_L^T$ and $T = \sum_{i=1}^{p} \widetilde{z}_{12}(i)$. Now, from the definition of Q in (4.30) we do not need to compute C in order to determine the projection onto the normal space it is enough to compute $C + C^T$ and $CD + DC^T$. We see that

$$C + C^T = C_L - C_L + C_L^T - C_L^T + C_U + C_U^T = C_U + C_U^T.$$

Hence, in order to compute $C + C^T$ we do not need to determine C_L and avoid dividing by the diagonal elements of S. Let us now look at

$$\begin{aligned} CD + DC^T &= C_L D - C_L^T D + DC_L^T - DC_L + C_U D + DC_U^T \\ &= C_L D - DC_L + (C_L D - DC_L)^T + C_U D + DC_U^T. \end{aligned}$$

As for $i > j$ with $\widetilde{b}_{31}$ the first q elements of $\widetilde{b}_3$

$$\begin{aligned} (C_L D - DC_L)_{ij} &= (C_L)_{ij} D_{jj} - D_{ii}(C_L)_{ij} \\ &= (C_L)_{ij}\sqrt{S_{\iota(i,j),\iota(i,j)}} \\ &= \widetilde{z}_{11}(\iota(i,j))\sqrt{S_{\iota(i,j),\iota(i,j)}} \\ &= -\widetilde{b}_{31}(\iota(i,j))/\sqrt{S_{\iota(i,j),\iota(i,j)}} \end{aligned}$$

we do not need to solve (4.36) for $\widetilde{z}_1$ it is sufficient to divide the elements of $\widetilde{b}_{31}$ by only the differences between the diagonal elements of D, corresponding to the diagonal elements of S, and not these elements squared.

A Retraction

With these tools that we have developed in the previous sections we can already obtain a direction along which we can optimize an objective function over this manifold. However, we still need a curve on the manifold, going in this direction at a point $Y \in \mathcal{B}(n,p)$.

Lemma 4.6.7. *Let $Y \in \mathcal{B}(n,p)$ and $H \in T_Y\mathcal{B}(n,p)$. Let $\widehat{R}_Y(H)$ be a retraction on the Stiefel manifold $\mathsf{St}(n,p)$ at Y. This is well defined as $Y \in \mathsf{St}(n,p)$ and $H \in T_Y\mathsf{St}(n,p)$. Then*

$$R_Y(H) = \widehat{R}_Y(H)P$$

is a retraction on $\mathcal{B}(n,p)$ at Y with $P\operatorname{diag}(\theta_1,\dots,\theta_p)P^T$ the spectral decomposition of $F(H) := \widehat{R}_Y(H)^T\Lambda\widehat{R}_Y(H)$ and $\theta_1 \le \theta_2 \le \cdots \le \theta_p$.

Proof. Let us check the conditions of Definition 3.7.8. For H in the neighbourhood of $0_Y \in T_Y\mathcal{B}(n,p)$ $R_Y(H)$ is clearly smooth as the diagonal elements of $F(0_Y)$ are distinct. Furthermore as this matrix is diagonal at 0_Y we have that $P = I_p$ for $H = 0_Y$ and

$$R_Y(0_Y) = Y.$$

Let us now consider the curve $R_Y(tH) = \widehat{R}(tH)P(t)$, which exists for all t sufficiently small [43, Section 2.2], where $P(t)$ is the orthogonal matrix that diagonalizes $F(tH)$. Then

$$\frac{d}{dt}R_Y(tH)\Big|_{t=0} = HI_p + Y\frac{d}{dt}P(t)\Big|_{t=0}. \tag{4.40}$$

From [43, Section 2.2] we obtain that

$$\frac{d}{dt}P(t)\Big|_{t=0} = P(0)T$$

with T skew-symmetric and

$$T_{ij} = \frac{\left(P(0)^T\frac{d}{dt}\left(F(tH)\right)\Big|_{t=0}P(0)\right)_{ij}}{d_j - d_i}$$

for $i \neq j$ and d_i the diagonal elements of $Y^T\Lambda Y$ for $i = 1,\dots,p$. As by (4.28) $\operatorname{offdiag}\left(\frac{d}{dt}\left(F(tH)\right)\Big|_{t=0}\right) = \operatorname{offdiag}\left(H^T\Lambda Y + Y^T\Lambda H\right) = 0$ and T skew-symmetric we have that $T = 0$. This implies that the left-hand side of (4.40) is H. Therefore all conditions for $R_Y(H)$ to be a retraction are satisfied. ∎

4.6.4 Optimization over Whole Constraint Set

Now we have developed all necessary tools to apply one of the algorithms discussed in Section 3.9 to optimize a smooth function f over $\mathcal{B}(n,p)$.

Reformulation of Problem

However, the aim is to optimize a smooth function f over $\mathcal{C}$ as defined in (4.24). Therefore, in order to incorporate the p constraints of $\mathcal{C}$ that are disregarded in $\mathcal{B}(n,p)$ we are interested in solving

$$\begin{array}{ll} \min_{Y\in\mathcal{B}(n,p)} & f(Y) \\ \text{s.t.} & c_i(Y)=0 \quad \text{for all } i=1,\dots,p, \end{array} \tag{4.41}$$

where $c_i(Y)\in\mathcal{F}(\mathcal{B})$ are the p equality constraints with

$$c_i(Y) = Y_i^T \Lambda Y_i - D_{ii} \quad \text{for all } i=1,\dots,p.$$

The index i in Y_i denotes the ith columns of Y.

Our Algorithm

Now, we are ready to propose our algorithm to solve (4.41). That is to apply the augmented Lagrangian method [102, Algorithm 17.4] to (4.41) and to use the nonlinear CG method described in Section 3.9 to solve the inner problem (4.43).

The augmented Lagrangian method for solving (4.41) can be stated as follows. Let us first define the augmented Lagrangian function of (4.41), that is

$$G_{\mu,\theta}(Y) = f(Y) - \sum_{i=1}^{p} \theta_i c_i(Y) + \frac{\mu}{2}\sum_{i=1}^{p} c_i(Y)^2 \tag{4.42}$$

where $Y\in\mathcal{B}(n,p)$, $\theta\in\mathbb{R}^p$ are the Lagrange multipliers and $\mu>0$ is the penalty parameter. Let $\mu_0>0$ and $\theta^0\in\mathbb{R}^p$ be the initial estimate of the Lagrange multipliers. Then the augmented Lagrangian method is to determine at the kth iteration

$$Y_{k+1} \in \operatorname*{argmin}_{Y\in\mathcal{B}(n,p)} G_{\mu_k,\theta^k}(Y) \tag{4.43}$$

and, according to some rules [102, Algorithm 17.4], to update the Lagrange multipliers by

$$\theta_i^{k+1} := \theta_i^k - \mu_k c_i(Y_{k+1}) \;\text{ or }\; \theta_i^{k+1} := \theta_i^k$$

and to update the penalty parameter by

$$\mu_{k+1} := \mu_k \;\text{ or }\; \mu_{k+1} > \mu_k.$$

Note that (4.43) is only well defined if the minimizer is attained in $\mathcal{B}(n,p)$, which is generally true for μ_k large.

The minimization of the augmented Lagrangian function over $\mathcal{B}(n,p)$ in (4.43) can be carried out by applying one of the algorithms discussed in Section 3.9. We

use the nonlinear CG method in our numerical tests in the next section. Further, for the algorithm to work we need to slightly modify the Armijo-backtracking procedure in the nonlinear CG method by additionally checking that the new point Y is on the manifold $\mathcal{B}(n,p)$. The reason for this modification is that for large step sizes the diagonal elements of $Y^T\Lambda Y$ may not satisfy the conditions of $\mathcal{B}(n,p)$. In our code we check whether the diagonal elements of $Y^T\Lambda Y$ are in increasing order and for numerical stability whether the minimal gap between two diagonal elements are greater than a certain tolerance d_m = 1e-5. If the α-condition (3.27) of the Armijo-backtracking procedure is satisfied and either of the other checks fail we continue our backtracking to find a new feasible point but exit the nonlinear CG method in the next instance and restart with an increased penalty parameter μ and updated Lagrange multipliers. This procedure guarantees that the new point will always be on the manifold $\mathcal{B}(n,p)$. We briefly summarized the steps of this algorithm in Algorithm 4.6.1 that we will refer to as ALB.

Algorithm 4.6.1 (ALB) This algorithm minimizes an arbitrary smooth function f over the set $\mathcal{C}$ in (4.24).

Require: $p, n \in \mathbb{N}$, $p \leq n$, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $D = \operatorname{diag}(d_1, \ldots, d_p)$ with $d_1 < d_2 < \cdots < d_p$, and a smooth function $f : \mathbb{R}^{n\times p} \mapsto \mathbb{R}$ that is to minimize and its derivative ∇f.

1 Apply the augmented Lagrangian method [102, Algorithm 17.4] to (4.41) where the inner problem (4.43) is to minimize the augmented Lagrangian function (4.42) over $\mathcal{B}(n,p)$. Solve (4.43) by using the nonlinear CG method in Section 3.9 with an Armijo-backtracking procedure that is modified according to the discussions in Section 4.6.4.

2 **return** Minimizer Y_*.

Optimality Conditions for Nonlinear Programming

In $\mathbb{R}^n$ we can derive a necessary condition in connection with the corresponding augmented Lagrangian function for a point $x_* \in \mathbb{R}^n$ to be a local minimizer of a nonlinear programming

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0 \quad \text{for all } i = 1, \ldots, p, \end{aligned} \tag{4.44}$$

where $f(x), c_1(x), \ldots, c_p(x)$ are smooth functions in $\mathbb{R}^n$.

Theorem 4.6.8. *[102, Theorem 17.5] Let x_* be a local minimizer of* (4.44) *at which the second-order sufficient conditions [102, Theorem 12.6] are satisfied for the Lagrange multipliers θ_* and let further the linear independent constraint qualification (LICQ) be satisfied at x_*, that is that $\nabla c_1(x), \ldots, \nabla c_p(x)$ are linearly independent.*

Then x_* is a strict local minimum of the corresponding augmented Lagrangian function $G_{\mu,\theta_*}(x)$ for all μ large enough.

We will show that this can be generalized to Riemannian manifolds that are submanifolds of $\mathbb{R}^n$.

Recently Yang and Zhang [142] have shown that the concept of the Lagrange multipliers for nonlinear programming problems can be generalized to a Riemannian manifold $\mathcal{M}$. Furthermore, they derived similar necessary and sufficient conditions as in $\mathbb{R}^n$ that a Karush-Kuhn-Tucker (KKT) point $Y \in \mathcal{M}$ is a local optimum of the corresponding nonlinear programming problem. Let now $c_i \in \mathcal{F}(\mathcal{M})$ for $i = 1, \ldots, p$ be p equality constraints of an optimization problem of the form

$$\begin{array}{ll} \min_{Y \in \mathcal{M}} & f(Y) \\ \text{s.t.} & c_i(Y) = 0 \quad \text{for all } i = 1, \ldots, p, \end{array} \tag{4.45}$$

where $f \in \mathcal{F}(\mathcal{M})$. Then we first need to generalize LICQ as follows [142, Equation (15)].

Definition 4.6.9. The *LICQ* on $\mathcal{M}$ at $Y \in \mathcal{M}$ holds if $\operatorname{grad} c_1(Y), \ldots, \operatorname{grad} c_p(Y) \in T_Y\mathcal{M}$ are linearly independent where $\operatorname{grad} c_i(Y)$ is the gradient in the tangent space $T_Y\mathcal{M}$ defined in (3.4) for all $i = 1, \ldots, p$.

Let

$$L_\theta(Y) := f(Y) - \sum_{i=1}^{p} \theta_i c_i(Y) \tag{4.46}$$

be the Lagrangian function of (4.45) where θ_i are the Lagrange multipliers. The next theorem gives a necessary condition of $Y \in \mathcal{M}$ to be a solution of (4.45) [142, Theorem 3.7].

Theorem 4.6.10. *Let Y_* be a local solution of* (4.45) *and let the LICQ be satisfied at Y_*. Then there exist Lagrange multipliers θ_* such that* $\operatorname{grad} L_{\theta_*}(Y_*) = 0$ *and* $c_i(Y_*) = 0$ *for all $i = 1, \ldots, p$.*

We can also derive a sufficient condition.

Theorem 4.6.11. *[142, Theorem 3.11] Suppose $Y_* \in \mathcal{M}$ and $\theta_* \in \mathbb{R}^p$ satisfy* $\operatorname{grad} L_{\theta_*}(Y_*) = 0$ *and $c_i(Y_*) = 0$ for all $i = 1, \ldots, p$. Suppose also that*

$$\langle Z, \operatorname{Hess} L_{\theta_*}(Y_*)[Z] \rangle_{Y_*} > 0$$

for all $Z \in T_{Y_*}\mathcal{M}$ with $\langle \operatorname{grad} c_i(Y_*), Z \rangle_{Y_*} = 0$ for all $i = 1, \ldots, p$, and $Z \neq 0$. Then Y_* is a strict local solution of (4.45). *Recall that* $\operatorname{Hess} L_{\theta_*}(Y_*)[Z]$ *denotes the Hessian operator of $L_{\theta_*}(Y_*)$ that we defined in* (3.9).

Now we are ready to generalize the Theorem 4.6.8 to Riemannian manifolds that are embedded in $\mathbb{R}^n$.

Theorem 4.6.12. *Let $\mathcal{M}$ be a Riemannian manifold embedded in $\mathbb{R}^n$ with $\langle\cdot,\cdot\rangle$ the Riemannian metric and $\|\cdot\|$ the induced norm. Let Y_* satisfy the condition of Theorem 4.6.10 and Theorem 4.6.11 with Lagrange multipliers θ_*. Then Y_* is a strict local minimum of the augmented Lagrangian function*

$$G_{\mu,\theta_*} = f(Y) - \sum_{i=1}^{p} \theta_i c_i(Y) + \frac{\mu}{2} \sum_{i=1}^{p} c_i(Y)^2$$

for all $\mu > 0$ sufficiently large.

Proof. The proof is analogous to the proof of [102, Theorem 17.5]. First as Y_* is a local solution of (4.45) we have

$$\begin{aligned}\operatorname{grad} G_{\mu,\theta_*}(Y_*) &= \operatorname{grad} f(Y_*) - \sum_{i=1}^{p} \big((\theta_*)_i - \mu c_i(Y_*)\big) \operatorname{grad} c_i(Y_*) \\ &= \operatorname{grad} f(Y_*) - \sum_{i=1}^{p} (\theta_*)_i \operatorname{grad} c_i(Y_*) = \operatorname{grad} L_{\theta_*}(Y_*) = 0.\end{aligned}$$

Hence, Y_* is a stationary point of $G_{\mu,\theta_*}(Y_*)$. Let us now assume that $\operatorname{Hess} G_{\mu,\theta_*}(Y_*)$ is not positive definite for all μ sufficiently large. Therefore we can choose a vector $Z_k \in T_{Y_*}\mathcal{M}$ with $\|Z_k\| = 1$ for each integer $k \geq 1$ sufficiently large such that

$$\langle \operatorname{Hess} G_{k,\theta_*}(Y_*)[Z_k], Z_k\rangle \leq 0. \tag{4.47}$$

Now let $g(Y) := k\sum_{i=1}^{p} c_i(Y)^2$. Then $\operatorname{grad} g(Y) = k\sum_{i=1}^{p} c_i(Y) \operatorname{grad} c_i(Y)$. From (3.6) and (3.9) we have

$$\operatorname{Hess} g(Y)[Z] = \nabla_Z \operatorname{grad} g(Y) = \Pi_{T_Y\mathcal{M}}(D_{\operatorname{grad} g(Y)}(Y, Z)),$$

where $\nabla_Z \operatorname{grad} g(Y)$ is the Levi-Civita connection from Theorem 3.7.3, $\Pi_{T_Y\mathcal{M}}(\cdot)$ is the projection onto the tangent space $T_Y\mathcal{M}$, and $D_{\operatorname{grad} g(Y)}(Y, Z)$ is the Fréchet derivative, see Appendix A.2. By using Taylor's formula we obtain

$$D_{\operatorname{grad} g(Y)}(Y_*, Z) = k\sum_{i=1}^{p} \langle \nabla c_i(Y_*), Z\rangle \operatorname{grad} c_i(Y_*).$$

Hence, from (4.47)

$$\begin{aligned}\langle \operatorname{Hess} G_{k,\theta_*}(Y_*)[Z_k], Z_k\rangle &= \langle \operatorname{Hess} L_{\theta_*}(Y_*)[Z_k], Z_k\rangle \\ &\quad + \left\langle k\sum_{i=1}^{p} \langle \nabla c_i(Y_*), Z_k\rangle \operatorname{grad} c_i(Y_*), Z_k \right\rangle \leq 0.\end{aligned}$$

where $L_{\theta_*}(Y)$ is the Lagrangian function defined in (4.46). It follows that

$$\sum_{i=1}^{p} \langle \operatorname{grad} c_i(Y_*), Z_k \rangle^2 \leq -\frac{1}{k} \langle \operatorname{Hess} L_{\theta_*}(Y_*)[Z_k], Z_k \rangle \to 0$$

as $k \to \infty$. As the vectors Z_k lie in a compact set, there exists an accumulation point Z. Hence, it follows that $\langle \operatorname{grad} c_i(Y_*), Z \rangle = 0$ for all $i = 1, \ldots, p$ and $\langle \operatorname{Hess} L_{\theta_*}[Z], Z \rangle \leq 0$, which contradicts our assumption. Hence, we have that $\operatorname{Hess} G_{\mu,\theta_*}(Y_*)$ is positive definite for μ sufficiently large. □

Let $x_* \in \mathbb{R}^n$ and $\theta_* \in \mathbb{R}^p$ be defined as in Theorem 4.6.8 and let the assumption of this theorem be satisfied at x_* and θ_*. Then in $\mathbb{R}^n$ Bertsekas [13, Proposition 4.2.3] obtained a convergence result for the sequence $\{(x_k, \theta^k, \mu_k)\}_{k \geq 0}$ that is generated by the augmented Lagrangian method. There exists constants $\delta > 0$ and $\varepsilon > 0$ so that if

$$\|\theta^k - \theta_*\|_2 \leq \mu_k \delta \quad \text{for all } \mu_k \text{ sufficiently large} \quad \text{and} \quad \|x_k - x_*\|_2 \leq \varepsilon$$

we have that

$$\|x_k - x_*\|_2 \leq M \|\theta^k - \theta_*\|_2 / \mu_k \quad \text{and} \quad \|\theta^k - \mu_k c(x_k)\|_2 \leq M \|\theta^k - \theta_*\|_2 / \mu_k$$

for a constant $M > 0$ and $c(x) = [c_1(x), \ldots, c_p(x)]^T$.

Since we have not been able to prove that the LICQ condition for the constraints in (4.41) is always satisfied at the optimal points Y_* of (4.41) and since the result by Bertsekas relies again on the LICQ we did not pursue to generalize this result on Riemannian manifolds. The same holds for the condition that $\langle Z, \operatorname{Hess} L_{\theta_*}(Y_*)[Z] \rangle > 0$ for all $Z \in T_{Y_*}\mathcal{M}$ with $\langle \operatorname{grad} c_i(Y_*), Z \rangle = 0$ for all $i = 1, \ldots, p$ and $Z \neq 0$. Therefore we cannot guarantee convergence of ALB in general. Note further that the minimization in (4.43) can fail as our retraction proposed in Lemma 4.6.7 is only defined in a small neighbourhood of our starting point, thus we might only be able to optimize locally.

We will check the LICQ in our numerical tests in the next section, in which we will investigate the performance of ALB by applying this algorithm to a test problem.

4.7 Computational Experiments

As the initial eigenvalue decomposition of N is the major cost for Algorithm 4.4.1 and 4.5.1, and the iteration number of Algorithm 4.5.1 is at most $2p$ we do not expect to gain further insight into the performance of these algorithms by applying them to test examples. Therefore we focus on investigating the performance of Algorithm ALB in this section.

4.7.1 Test Problem

Let us first introduce a test problem to investigate the performance of ALB, which also arises from the application in atomic chemistry that leaded us to our problems (4.1) and (4.2). In this application it is desired to find a point $Y \in \mathsf{St}(n,p)$ that is optimal with respect to (4.1) but also has columns that ideally preserve the sign characteristics of the eigenvectors of N. Surely this preservation is a constraint that is usually hard to deal with. We therefore pose a new optimization problem over the set of optimal solutions of (4.1) and are looking for an appropriate smooth objective function $f(Y)$ whose minimization drives us to a solution in the set of optimal solutions of (4.1) that preserves the sign characteristics. This optimization problem will be our test problem for ALB.

Our idea to approach the preservation of the sign characteristics is to minimize the angle in modulus between every column of Y and the eigenvectors of P by solving

$$\begin{array}{cc} \min_{Y\in\mathsf{St}(n,p)} & -\sum_{i=1}^{p} \|P^T Y_i\|_\infty \\ \text{s.t.} & Y^T N Y = D, \end{array} \tag{4.48}$$

where Y_i is the ith column of Y. As the infinity norm $\|\cdot\|_\infty$ is not smooth we will use the q-norm $\|x\|_q := (\sum_{i=1}^{n} |x_i|^q)^{\frac{1}{q}}$ for $q = 4$ instead. This yields the problem

$$\begin{array}{cc} \min_{Y\in\mathsf{St}(n,p)} & -\sum_{i=1}^{p} \|P^T Y_i\|_q^q, \\ \text{s.t.} & Y^T N Y = D, \end{array}$$

which can equivalently be written as

$$\min_{Y\in\mathcal{C}} f(Y) := -\sum_{i=1}^{p} \|Y_i\|_q^q. \tag{4.49}$$

The latter problem (4.49) will be our test problem for investigating the performance of Algorithm ALB.

Note that another alternative for the objective function in (4.48) is $\widehat{f}(Y) = \sum_{i=1}^{p}(\|P^T Y_i\|_1 - 1)^2$. The idea is here to reduce for all $i = 1,\dots,p$ the angle in modulus between Y_i and one column of P and to simultaneously enlarge the angle in modulus between Y_i and the other columns of P. This function is surely not smooth but could, for example, be approximated by $\widetilde{f}(Y) = \sum_{i=1}^{p}\big(\sum_{j=1}^{n}((P^T Y_i)_j^2)^{\frac{q}{2q-1}} - 1\big)^2$ for q large. We will not pursue these ideas any further.

In order to apply ALB to (4.49) it remains to determine the matrix of partial derivatives of $f(Y)$, that is

$$\nabla f(Y) = -q\big(Y_{ij}^{q-1}\big)_{i=1,j=1}^{n,p}.$$

4.7.2 Numerical Methods

In addition to ALB we use the following algorithm for comparing purposes in our tests. We will refer to this algorithm as ALS. Let $c_{ij} : \mathbb{R}^{n\times p} \mapsto \mathbb{R}$ be defined as $c_{ij}(Y) = (Y^T \Lambda Y)_{ij}$ for $i > j$ and $c_{ii}(Y) = (Y^T \Lambda Y)_{ii} - D_{ii}$ for $i = j$ and $i, j = 1, \ldots, p$. We reformulate (4.49) as

$$\begin{array}{ll} \min_{Y\in \mathsf{St}(n,p)} & f(Y) \\ \text{s.t.} & c_{ij}(Y) = 0 \qquad i, j = 1, \ldots, p \text{ and } i \geq j \end{array} \tag{4.50}$$

and apply, similarly to ALB, the augmented Lagrangian method [102, Algorithm 17.4] to (4.50). The inner problem is then to minimize the following augmented Lagrangian function

$$G_{\mu,\theta}(Y) = f(Y) - \sum_{i\geq j} \theta_{ij} c_{ij}(Y) + \frac{\mu}{2} \sum_{i\geq j} c_{ij}(Y)^2 \tag{4.51}$$

over the Stiefel manifold. In (4.51) θ_{ij} are the Lagrange multipliers for $i \geq j$. To minimize $G_{\mu,\theta}(Y)$ we use again the nonlinear CG method discussed in Section 3.9.

4.7.3 Test Matrices and Starting Values

In order to investigate the performance of these two algorithms we need to provide test matrices for the diagonal matrices Λ and D in (4.49). We look at two different classes whereas the first is more for demonstrating purposes.

- **ldchem**: Prof. Alexander Sax provided us with a small example for $N \in \mathbb{R}^{11\times 11}$ in (4.1) that is

$$N = \begin{bmatrix} 0.0001 & 0.0002 & 0.0005 & 0.0013 & 0.0028 & 0.0054 & 0.0080 & 0.0078 & 0.0041 & 0.0006 & -0.0004 \\ 0.0002 & 0.0005 & 0.0013 & 0.0031 & 0.0068 & 0.0131 & 0.0194 & 0.0188 & 0.0098 & 0.0015 & -0.0011 \\ 0.0005 & 0.0013 & 0.0032 & 0.0075 & 0.0166 & 0.0319 & 0.0472 & 0.0459 & 0.0239 & 0.0036 & -0.0027 \\ 0.0013 & 0.0031 & 0.0075 & 0.0178 & 0.0393 & 0.0755 & 0.1118 & 0.1086 & 0.0565 & 0.0082 & -0.0166 \\ 0.0028 & 0.0068 & 0.0166 & 0.0393 & 0.0869 & 0.1667 & 0.2470 & 0.2398 & 0.1244 & 0.0174 & -0.0153 \\ 0.0054 & 0.0131 & 0.0319 & 0.0755 & 0.1667 & 0.3198 & 0.4739 & 0.4599 & 0.2375 & 0.0307 & -0.0317 \\ 0.0080 & 0.0194 & 0.0472 & 0.1118 & 0.2470 & 0.4739 & 0.7023 & 0.6814 & 0.3512 & 0.0438 & -0.0486 \\ 0.0078 & 0.0188 & 0.0459 & 0.1086 & 0.2398 & 0.4599 & 0.6814 & 0.6633 & 0.3547 & 0.0737 & -0.0194 \\ 0.0041 & 0.0098 & 0.0239 & 0.0565 & 0.1244 & 0.2375 & 0.3512 & 0.3547 & 0.2639 & 0.2189 & 0.1526 \\ 0.0006 & 0.0015 & 0.0036 & 0.0082 & 0.0174 & 0.0307 & 0.0438 & 0.0737 & 0.2189 & 0.4434 & 0.3980 \\ -0.0004 & -0.0011 & -0.0027 & -0.0066 & -0.0153 & -0.0317 & -0.0486 & -0.0194 & 0.1526 & 0.3980 & 0.3919 \end{bmatrix},$$

  which corresponds to an s-block of C-atom. The three largest eigenvalues of N are $0.0144, 0.881$, and 1.99 and all other eigenvalues have a modulus of less than 0.001. If we prescribe two orbitals with occupation numbers 1.5 and 0.1, by the analysis of Section 4.4 it follows that the set of optimal solutions of (4.1) is equivalent to

$$\left\{ Y \in \mathsf{St}(n,p) : Y^T \Lambda Y = \operatorname{diag}(0.1, 1.5) \right\}$$

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues of N in increasing order. Therefore we use Λ and $D = \text{diag}(0.1, 1.5)$ as our test matrices.

- **ldrand**: The second class is drawn from purely randomly generated matrices. We generate a symmetric matrix $N \in \mathbb{R}^{n\times n}$ by means of the MATLAB commands

  ```
  N=rand(n,n);N=N+N';
  ```

  and compute the matrix of eigenvalues of N, which is our test matrix Λ. For the diagonal matrix D we first generate randomly a diagonal matrix in MATLAB by using

  ```
  diag(sort(rand(p,1)*p));
  ```

  and set then, accordingly to (4.7), D to the closest diagonal matrix that is imbeddable in N. If one diagonal element of D is within the range of 0.01 of another we repeat the process of generating D.

For our starting matrix Y_0 we can use the matrix computed by Algorithm 4.4.1; however our numerical tests indicate that these points are local minima or are close to local minima of (4.49) as we will demonstrate in the first numerical test in Section 4.7.4. Hence, they are not suitable to investigate the performance of our algorithms ALB and ALS.

Therefore we randomly generate a matrix $Y \in \mathsf{St}(n, p)$ by applying the MATLAB function `rand` and computing the Q-factor of the randomly generated matrix by means of `qr`. Thereafter we set $Y_0 = YP$ where $P \in \mathsf{O}(p)$ computed by `eig` diagonalizes $Y^T \Lambda Y$ with the diagonal elements increasing. If the distance between two diagonal elements is less than 0.01 we repeat the procedure, making sure that $Y_0 \in \mathcal{B}(n, p)$.

4.7.4 Numerical Tests

Chosen Parameters

We implement and test both algorithms on an Intel(R) Core(TM)2 Quad CPU (2.83GHz each processor) with 4GB RAM, Ubuntu Linux 10.04.1 64bit in MATLAB R2010a. We use for both algorithms the augmented Lagrangian method proposed in [102, Algorithm 17.4] with the following modifications.

- If the penalty parameter needs to be enlarged we only increase it by a factor of 2 instead of 100 as we have empirically experienced better performance. Furthermore, we use $\mu_0 = 10$ for our initial choice of the penalty parameter μ.

- We change the stopping criterion for the inner problem to

$$\|\operatorname{grad} G_{\mu,\theta_k}(Y_k)\|_F < np^2\omega_k \tag{4.52}$$

  where $\operatorname{grad} G_{\mu,\theta_k}(Y_k)$ is the gradient at Y_k of the augmented Lagrangian function (4.42) for ALB and (4.51) for ALS.

- We choose η_*, which is tolerance for the constraint violation at the final iterate Y_*

$$\sqrt{\sum_{i=1}^{p} c_i(Y_*)^2} \leq p^2\eta_* \text{ for ALB and } \sqrt{2\sum_{i>j}^{p} c_{ij}(Y_*)^2 + \sum_{i=1}^{p} c_{ii}(Y_*)} \leq p^2\eta_* \text{ for ALS,}$$

  to be 10^{-6} and ω_*, the tolerance of (4.52) at Y_*, to be 10^{-6}.

- We limit the number of iterations in the augmented Lagrangian method to 100.

For the nonlinear CG method (see Algorithm 3.9.1) we use the scalar β_k that was proposed by Polak-Ribière in both algorithms ALB and ALS as this choice outperforms the version of Fletcher-Reeves in our numerical tests. If $\xi_{x_{k+1}}$ in Algorithm 3.9.1 is not a descent direction we use the steepest descent direction $-\operatorname{grad} f(x_{k+1})$ instead. Further, we limit the number of iterations in the nonlinear CG method to $50,000$ and use for the backtracking strategy the Armijo-backtracking procedure as stated in Algorithm 3.9.1 with $\rho = 0.5$ and $\gamma = 10^{-4}$, where these parameters are proposed in [41, Algorithm A6.3.1]. To avoid many backtracking steps we compute a guess for the initial step length α in Algorithm 3.9.1 as follows. Let Y_i be the current iterate and ξ_{Y_i} be the current direction in the tangent space at Y_i in the nonlinear CG method at iteration i. Further let θ_k and μ_k be the current value for the Lagrange multipliers and the penalty parameter, respectively. Then our approach to find a good guess for the initial step length in both algorithms is to solve

$$\min_{t\in(0,1]} G_{\mu_k,\theta_k}(Y_i + t\xi_{Y_i}), \tag{4.53}$$

where the optimal solution is our initial step length α. Solving (4.53) corresponds to finding the roots of a polynomial of degree $q-1$. If no optimal solution can be found we set α to 1. If more than one optimal solutions exist we take the largest of the optimal solutions that are smaller than or equal to 1. Our numerical tests show that this choice for the initial steps length yields the desired results as most often only one or two backtracking steps are necessary to find a step length that is accepted.

For the retraction on the Stiefel manifold in ALS we use the unitary polar factor as described in Section 3.8.1 since, according to our tests, this choice gives us the best performance. We compared our results with the geodesic and the Q-factor. For ALB we use the retraction that we introduced in Lemma 4.6.7 and choose for $\widehat{R}_Y(H)$ again the unitary polar factor. For the vector transport in the nonlinear CG method we apply for both algorithms (3.10). As discussed in Section 4.6.3 for ALB we need to solve a linear system to compute the projection onto the normal space $N_Y\mathcal{B}(n,p)$. To solve this system we apply the algorithm provided by MATLAB that computes the LU-decomposition for dense matrices. This algorithm actually runs the LAPACK routine DGETRF [9]. In the third test of this section we will demonstrate by means of numerical tests that we cannot make use of the sparsity of the coefficient matrix $\widetilde{H}_2$ in Lemma 4.6.5 for the sizes of our test matrices that are used in our tests.

Test 1

Our first test is for demonstrating purposes of ALB. Our aim is to find a better solution of (4.49) than obtained by Algorithm 4.4.1 with respect to our test problem (4.49) for the test matrices **ldchem**.

To achieve this goal we generate 100 instances of the starting matrix Y_0 as described in Section 4.7.3 and apply ALB for every single starting matrix. To ensure that our output matrix is comparable with the matrix returned by Algorithm 4.4.1 we tighten our tolerance for the constraint violation and set η_* to 10^{-15}. The matrix returned by Algorithm 4.4.1 is

$$Y_{A1}=\begin{bmatrix} -0.941521479950964 & 0 \\ 0 & 0.499232223306862 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.336952968218038 & 0 \\ 0 & -0.866468226314207 \end{bmatrix} \tag{4.54}$$

and the objective value of (4.49) at Y_{A1} is -1.424475197032718.

Let us now apply ALB starting from these generated matrices and from the matrix Y_{A1}. We present our results in Table 4.1 where we only show the output of ALB for Y_{A1} and for the generated starting matrix that gives us the best objective function value among all generated matrices. Let this matrix be called Y_G. We denote the final iterate of Y, μ, and θ by Y_*, μ_*, and θ_*, respectively. The returned matrix Y_* is

Table 4.1: Output for ALB for test matrices **ldchem**.

	Output for Y_G	**Output for Y_{A1}**
Outer iterations	68	64
Total number of iterations in CG	179	121
Total number of backtracking steps in CG	60	18
$\operatorname{grad} G_{\mu_*,\theta_*}(Y_*)$	2.8e-8	1.5e-9
μ_*	1310720	655360
Computational time in seconds	1.1	0.5
Objective function at Y_0	-0.305477011857360	-1.424475197032718
Objective function at final iterate	-1.424475179359067	-1.424475197032720
Constraint violation $\|[c_1(Y_*),\ldots,c_p(Y_*)]\|_2$	3.2e-16	3.3e-16
rank of $[\operatorname{grad} c_1(Y_*), \operatorname{grad} c_2(Y_*)]$	2	2

the same as (4.54) for the starting matrix Y_{A1} and is similar for Y_G, which is

$$Y_*= \begin{bmatrix} 0.941521479950964 & -0.000000000143277 \\ 0.000000000007014 & 0.000000000941466 \\ 0.000000000011416 & 0.000000001576076 \\ -0.000000000003769 & 0.499232240953539 \\ 0.000000000001399 & -0.000000002465845 \\ -0.000000000001018 & -0.000000000203445 \\ 0.000000000014893 & 0.000000001699105 \\ -0.000000000015848 & 0.000000000165078 \\ -0.000000000015340 & 0.000000027803602 \\ -0.336952968218039 & -0.000000000725956 \\ -0.000000000124452 & 0.866468216146736 \end{bmatrix}. \tag{4.55}$$

We see that the matrix computed by Algorithm 4.4.1 is obviously close to a stationary point of the augmented Lagrangian function $G_{\mu_*,\theta_*}(Y)$ and we also observe that we cannot improve the objective function of (4.49) by starting from different randomly generated matrices. However, as we see by the example (4.55) we can converge to a different matrix. For Y_{A1} as a starting matrix we move first away from Y_{A1} as the penalty parameter is small but return back to our starting point during the iterations of the augmented Lagrangian method. Our penalty parameter is relatively large at the final iterate, which can be explained by the small tolerance for η_*. If we increase this tolerance we obtain moderate sizes for the penalty parameter.

As the rank of $[\operatorname{grad} c_1(Y_*), \operatorname{grad} c_2(Y_*)]$ is two for both returned matrices the LICQ defined in Definition 4.6.9 is satisfied at Y_*, giving us theoretical support that both Y_* are local solutions of (4.49). Note that for both matrices the smallest singular value of $[\operatorname{grad} c_1(Y_*), \operatorname{grad} c_2(Y_*)]$ is approximately 0.279.

Note further, we also checked for P the matrix of eigenvectors of N whether the solutions $\widetilde{Y}_1 := PY_{A1}$ and $\widetilde{Y}_2 := PY_*$ with Y_* defined in (4.55) preserve the sign characteristics of the eigenvectors of N. The multiplication by P from the left is needed to obtain the corresponding solution of the original problem (4.1). The result is that for both $\widetilde{Y}_1$ and $\widetilde{Y}_2$ the first column does have the same sign characteristics as the first column of P but unfortunately the second column does not have the same characteristics as any column of P. Therefore let us consider (4.49) exclusively as a

Table 4.2: Results for the randomly generated matrices Λ and D.

	ALB				ALS			
	t	**it**	**fv**	**gradn**	**t**	**it**	**fv**	**gradn**
$n = 100$								
$p = 10$	86	2420	-8.922100	0.009	322	1.5e4	-8.465806	0.010
$p = 20$	392	3597	-19.636700	0.039	1699	5.7e4	-19.410440	0.040
$p = 30$	1483	3304	-29.424377	0.083	6027	1.3e5	-29.095622	0.090
$p = 40$	682	559	-39.188760	0.159	2711	5.2e4	-39.508334	0.140
$p = 50$	1249	539	-49.283334	0.213	1.9e4	3.2e5	-49.472668	0.027
$n = 150$								
$p = 10$	163	5848	-8.234490	0.014	126	6082	-8.641289	0.015
$p = 20$	675	1.2e4	-18.159201	0.054	3417	1.3e5	-17.976303	0.059
$p = 30$	1899	9834	-28.033409	0.134	3166	9.1e4	-27.970304	0.134
$p = 40$	4739	8401	-37.627541	0.193	6808	1.7e5	-37.624987	0.240
$p = 50$	6770	5866	-48.566146	0.364	2.9e4	5.9e5	-48.951799	0.098
$n = 250$								
$p = 10$	51	1564	-7.646136	0.024	151	5636	-7.416588	0.025
$p = 20$	1190	1.5e4	-17.057930	0.099	2797	7.2e4	-17.227175	0.099
$p = 30$	5240	2.6e4	-26.942655	0.212	1.9e4	4.1e5	-27.322446	0.225
$p = 40$	1.0e4	1.9e4	-37.807461	0.355	3.2e4	5.8e5	-39.011732	0.164
$p = 50$	2.6e4	2.3e4	-48.070583	0.507	2.7e4	4.2e5	-46.994304	0.244

test problem.

In the second test we compare the performance of ALB with ALS.

Test 2

In the second test we randomly generate matrices for Λ and D of type **ldrand** for $n = 50, 100, \ldots, 250$ and $p = 5, \ldots, 50$ and apply ALB and ALS to our test problem (4.49) with these matrices. We show a selection of our results in Table 4.2 where we use the following abbreviations.

- **t**: time in seconds to compute the final iterate Y_*,
- **it**: total number of iterations in the nonlinear CG method,
- **fv**: function value at Y_*,
- **gradn**: grad $f(Y_*)$.

We see in Table 4.2 that in most tests ALB outperforms ALS in terms of the computational time. The main reason is that the ALB needs fewer iterations in the nonlinear CG method to satisfy the stopping criterion. This total number of iterations in the nonlinear CG method can differ by a factor of 100 but interestingly, it does not depend much on p. We do not report the number of the outer iterations

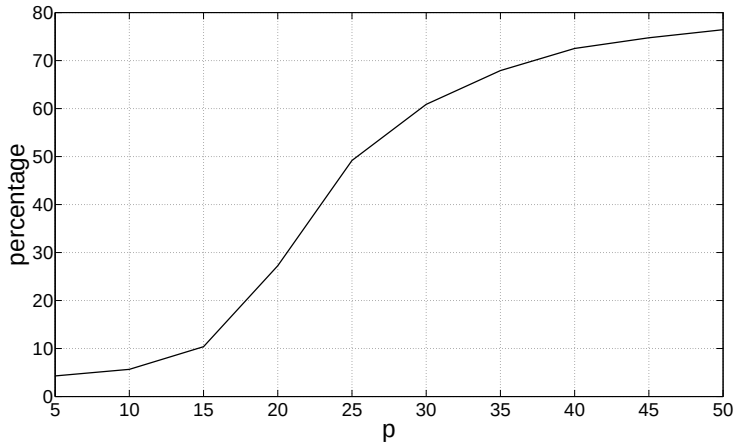

Figure 4.1: Ratio of time spent on computing the projection to total time

as this number does not vary much with n or p and is in the range of 10 to 20 whereas ALS takes most often a few outer iterations more than ALB.

We also observe that the cost per iteration is more expensive in ALB than in ALS and the relative difference is increasing with p. An explanation is clearly that the cost to compute the projection onto the normal space of $\mathcal{B}(n,p)$ is of order $\mathcal{O}(p^6)$. To demonstrate this more illustratively we plot the fraction of the time taken to compute the projection to the total time in Figure 4.1 for $n = 200$ and $p = 5, 10, \ldots, 50$. For $p = 50$ approximately 75% of the runtime of the code is spent to compute the linear system (4.35).

The function value $f(Y_*)$ at the final iterate Y_* does not differ largely between ALB and ALS and is most often slightly smaller than the function value at the point Y_{A1} that is returned by Algorithm 4.4.1 as can be seen in Figure 4.2. This affirms the observation of the previous test that the computed point Y_{A1} is close to a local minimum of your test problem (4.49). One also needs to consider that we allow for Y_* to violate the constraints slightly whereas a constraint violation at Y_{A1} can only be caused due to limitations in the numerical computation.

In our tests it happened several times in ALB that backtracking steps were required to remain on the manifold; see the derivation of ALB at the end of Section 4.6. It has occurred twice that due to this backtracking the step size was smaller than our minimal allowed step size so that the algorithm failed to return a point that satisfies our convergence criteria. In Figure 4.2 we see where this occurred as at $n = 200$, $p = 50$ and $n = 250$, $p = 35$ the function value of the point returned by ALB is clearly larger than at the point returned by ALS. In Figure 4.3 we show the rank of $\operatorname{grad} c(Y_*) = [\operatorname{grad} c_1(Y_*), \ldots, \operatorname{grad} c_p(Y_*)]$ where Y_* is the point that is returned by ALB. If $\operatorname{grad} c(Y_*)$ is of full rank then the LICQ is satisfied Y_*, giving us theoretical support that Y_* is a local minimum of (4.49). We see in this figure the LICQ is only

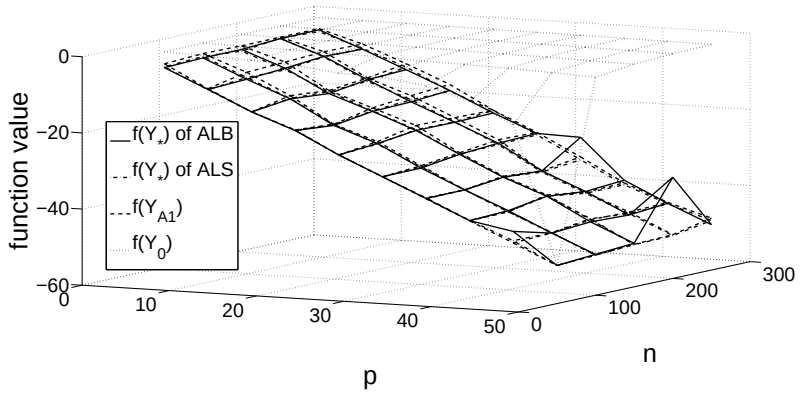

Figure 4.2: Comparison of function values

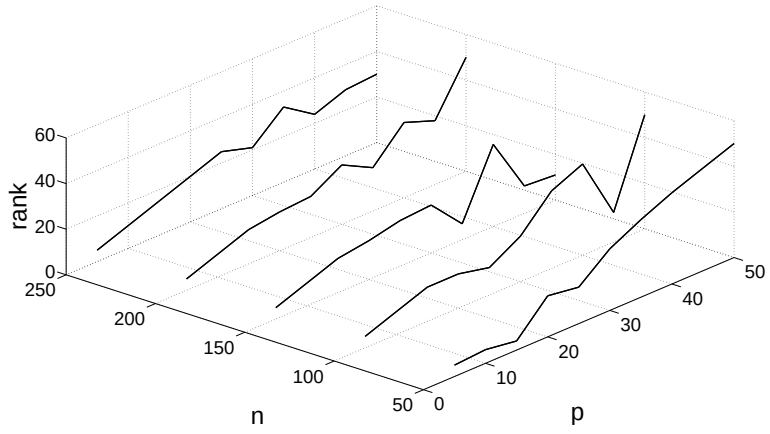

Figure 4.3: Rank of grad $c(Y_*)$

satisfied for p and n small. For larger values the rank is often even only of half of the size of p so that the existing theory discussed in Section 4.6.4 is not applicable.

Test 3

We have seen in Test 2 that solving the linear system (4.35) to compute the projection onto the normal space $N_Y\mathcal{B}(n,p)$ is the major cost of ALB. In addition by Lemma 4.6.6 the coefficient matrix $\widetilde{H}_2$ of linear system is sparse for p large. Therefore the question arises whether we can improve the performance of ALB by using a solver for sparse matrices to solve the linear system. We address this question in this test by comparing the following different algorithms to solve $\widetilde{H}_2\widetilde{z}_2 = \widetilde{b}_2$ in (4.35):

- lu: algorithm for the LU-decomposition for dense matrices provided by MATLAB and used in the previous tests (LAPACK routine DGETRF [9]),
- lus: algorithm for the LU-decomposition for sparse matrices provided by MATLAB (UMFPACK routine [37]).

Table 4.3: Results for different methods to solve the linear system $\widetilde{H}_2\widetilde{z}_2 = \widetilde{b}_2$ in (4.35).

p	**time in seconds**				
	lu	lus	minres	chol	chols
25	0.2	1.1	1.6	0.5	0.6
50	7.6	17	8.4	3.7	8.7
75	69	99	34	27	58
100	329	343	156	147	223

- minres: MINRES [105] provided by MATLAB to solve $\widetilde{H}_2\widetilde{z}_2 = \widetilde{b}_2$ with default tolerances, that is
$$\|\widetilde{H}_2\widetilde{z}_2 - \widetilde{b}_2\|_2 \leq 1e\text{-}6\|\widetilde{b}_2\|_2$$
with the diagonal of $\widetilde{H}_2$ as preconditioner.

- chol: algorithm for the Cholesky-decomposition for dense matrices provided by MATLAB (LAPACK routine DPOTRF [9]).

- chols: algorithm for the Cholesky-decomposition for sparse matrices provided by MATLAB (CHOLMOD [38]).

We proceed as follows: we generate 30 matrices N, D of type **ldrand** for $n = 150$, $p = 25, 50, 75, 100$, and a matrix A that is to project by using the MATLAB function `rand`. We then project A onto the tangent space $T_{Y_0}\mathcal{B}(n,p)$ at our starting matrix Y_0 for all 30 matrices and report the total time in seconds that is spent to solve the corresponding linear systems for every algorithm. Note that for the factorization methods only the time to compute the factorization of $\widetilde{H}_2$ is measured.

Our results are listed in Table 4.3. We observe that the direct solvers that make use of the sparsity of $\widetilde{H}_2$ are slower than those for dense matrices. Explanations could be that p is not large enough so that the ratio in Lemma 4.6.6 between the number of zeros and the total number of elements in $\widetilde{H}_2$ is too large to exploit the sparsity. Certainly also the pattern of the nonzeros in $\widetilde{H}_2$ could slow down the algorithms for sparse matrices. We also need to take into account that the routines come from different sources. Hence, they may differ in their level of tuning. Interestingly, the iterative solver MINRES for only one right-hand side takes more time than computing the Cholesky decomposition. Hence, the sizes of the matrices $\widetilde{H}_2$ that we tested are still in the range where direct solvers yield better results than iterative solvers.

Obviously, the algorithm for computing the Cholesky decomposition for dense matrices gives us the best time. However, this assumes that the matrix $\widetilde{H}_2$ is positive definite, which may not be the case for certain matrices input matrices N, D and

could cause a break down of ALB. To overcome this problem one could compute the LU-decomposition as in our previous tests or apply the algorithm for the Cholesky decomposition and in case of failure to revert to the LU-decomposition. The latter procedure could improve the performance of ALB in our second test.

To conclude, we have seen that ALB outperforms ALS in terms of time and number of iterations taken in the nonlinear CG method. For p large the projection onto the tangent space is the bottleneck of ALB as the corresponding computation is then the major cost of the algorithm. From the last test we conclude that the time spent to compute the projection can be reduced by using the Cholesky decomposition for solving the linear system if the coefficient matrix is positive definite.

In terms of robustness and stability we have observed that ALB can fail to return a stationary point of the augmented Lagrangian function. These failure were caused by the problem that starting from a point $Y_0 \in \mathcal{B}(n,p)$ we can only guarantee for small movements realized by our retraction proposed in Section 4.6.3 that we remain on the manifold. Therefore if sufficient decrease in the augmented Lagrangian function is only achieved by leaving the neighbourhood that is characterized by these small movements ALB might fail. One positive effect of ALB is that the total number of iterations in the nonlinear CG method do not vary as much as for ALS. Hence ALB shows a more stable behaviour.

4.8 Conclusions

In this chapter we have looked at two two-sided optimization problems with orthogonal constraints arising in atomic chemistry. We investigated these two problems and showed that they do generally not have unique optimal solutions. We proposed two algorithms to find optimal solutions of either problems whose major cost are a few eigenvalue decompositions. To establish the opportunity to optimize over the set of the optimal solutions of the first problem we investigated this set further. We showed that a slight modification of this set is a Riemannian manifold that allows us to evolve all geometric objects that are needed for an optimization. To deal with the constraints that we had dropped we proposed an augmented Lagrangian-based algorithm whose inner problem is to minimize the augmented Lagrangian function over this new manifold. To solve this problem we used the nonlinear CG method. We investigated the numerical performance of this algorithm that we called ALB by applying it to a test problem and compared it with the performance of algorithm ALS that we introduced in Section 4.7. This algorithm is again an augmented Lagrangian-based method whose inner problem is to minimize the augmented Lagrangian function over the Stiefel manifold. To incorporate all the constraints in this method we need to

use $p(p-1)/2$ more Lagrange multipliers than in ALB. Our numerical tests showed that ALB outperforms ALS as it took less time and fewer iterations in the nonlinear CG method in most tests. However, as the computation of the projection onto the tangent space of the new manifold costs of the order $\mathcal{O}(p^6)$ operations the cost per iteration of ALB is significantly more expensive than ALS. Therefore the saving in using ALB is rather moderate for p large.

Chapter 5

Low Rank Linearly Structured Matrix Nearness Problems

5.1 Introduction

In this chapter we consider the problem of finding a low rank matrix that is of linear structure and closest to a given matrix in the Q-norm. Recall from Section 1.5 that for $U_1, \ldots, U_s \in \{0,1\}^{n\times p}$ the set of linearly structured matrices is defined as

$$\mathcal{L}(U_1, \ldots, U_s) := \left\{ X : X = \sum_{i=1}^{s} x_i U_i, \text{ and } x_i \in \mathbb{R} \text{ for all } i = 1, \ldots, s \right\}, \tag{5.1}$$

where every element $X \in \mathcal{L}$ can be written as $\operatorname{vec}(X) = Ux$ with U defined in (1.7) and $x = (x_1, \ldots, x_s)^T$. Throughout this section we assume that U is of full rank.

Our main interest in this chapter lies in algorithms that solve the low rank matrix nearness problem in the Q-norm and for any linear matrix structure U although for special matrices U more efficient algorithms can be developed that exploit the structure and can thus, perform better. However, we are more concerned about algorithms that are more flexible towards the matrix structure. Let us now state the low rank problem.

5.1.1 The Problem

Let $A \in \mathbb{R}^{n\times p}$ be given, $s \leq np$, and $r < \operatorname{rank}(A)$. Then we are interested in solving

$$\min_{X\in\mathcal{L}\cap\mathcal{R}_r} \frac{1}{2}\|A - X\|_Q^2, \tag{5.2}$$

where

$$\mathcal{R}_r := \left\{ X \in \mathbb{R}^{n\times p} : \operatorname{rank}(X) \leq r \right\}. \tag{5.3}$$

and $\mathcal{L}$ is described by $U_1 \ldots, U_s$ as defined in (5.1). Note that the $\mathcal{R}_r$ is generally nonconvex so that we cannot apply the alternating projection method that we introduced in Section 1.4.2. Therefore we need to consider different algorithms that solve this problem.

We have found different approaches in the literature to deal with this problem, but the set of matrices is often restricted to a certain class of structured matrices like Hankel, Toeplitz, or Sylvester structure. Certainly one reason is the computational complexity for general linear structures, especially in the weighted case where often one can only apply general nonlinear optimization techniques to find local minima. When considering particular structures often more efficient algorithms can be developed by exploiting the structure. However, the drawback is that for every class of linear matrix structure one needs to develop a different algorithm. This is certainly not practical for any class of different linear matrix structure, and therefore there is need for a robust algorithm providing low rank approximations for any linear matrix structure for a moderate problem size.

Another issue is the existence of the solution. If the rank of the matrix X in (5.2) needs to be predetermined then the existence question is hard to address for general linear structures. However, there are results for classes of structured matrices. Examples are symmetric Toeplitz matrices and squared Hankel matrices, for which solutions exist for any rank according to [30, Theorem 3.2. and Theorem 3.3]. We avoid this problem by defining $\mathcal{R}_r$ as in (5.3) so that the matrix $X = 0$ is always a solution and thus, the set of solutions is not empty.

5.1.2 Applications

The low rank constraint is of high interest in many applications and is often associated with model reduction, in particular for data analysis. As already seen in Section 1.7.2 this low rank problem occurs in areas of engineering such as speech encoding or filter design [121] where one is dealing with Hankel matrices of low rank in order to reduce the dimension of the problem or to remove noise of the incoming signal [106]. Similar applications arise in system identification, system response prediction [118], [135], and frequency estimation [116].

In latent semantic indexing, which is a method used for automatic indexing and retrieval of information, one is seeking a sparse low rank approximation of a sparse matrix [39] where often the singular value decomposition is used to obtain this low rank approximation. However, this decomposition does not need to be sparse. Therefore other methods are required. On example that takes the preservation of the sparsity in the decomposition into account is the truncated pivoted QR approximations to a sparse matrix by Stewart [128], [12]. Another way of finding a sparse low rank

approximation, in particular if certain patterns of the sparsity structure should be preserved, is to transform this problem into a problem of the form in (5.2).

Recall from Section 1.7.4 that an application also arises in computer algebra systems where one is interested in computing approximations to two polynomials that have a greatest common divisor with a degree greater or equal to a predefined number.

5.1.3 Outline

In the next section we will introduce three different algorithms in the existing literature dealing with any linear matrix structure and discuss their strengths and weaknesses. We look then more closely at a geometric optimization approach in Section 5.3 where the problem is reformulated to apply the augmented Lagrangian method. The latter requires to optimize over the Grassmannian manifold. We consider different optimization techniques to make this approach more efficient and propose at the end of this section an algorithm for solving these problems. Unfortunately we cannot prove convergence for this algorithm in general and will therefore carry out extensive tests in Section 5.4, demonstrating numerically the algorithm's superior performance in comparison to existing algorithms.

5.2 Algorithms Dealing with Any Linear Structure

5.2.1 The Lift and Projection Algorithm

The first algorithm is the lift and projection algorithm proposed and discussed in [27], [30], [29] for a particular application. It tackles the problem by alternating between unstructured low rank minimization and structure enforcement procedure. Therefore this algorithm is only valid if both optimization problems can be solved. Furthermore, since the set of low rank matrices is not convex the theory of the alternating projections methods as outlined in Section 1.4.2 does not apply and thus, the algorithm proposed may not converge. Let Y_k be the solution of the unstructured low rank minimization and Z_k the solution of the subsequent enforcement procedure in the Frobenius norm at iteration k. Then

$$\|Y_{k+1} - Z_{k+1}\|_F \leq \|Y_k - Z_k\|_F$$

as shown by the authors in [29]. However, as mentioned above and pointed out in [29] this is no guarantee that the algorithm does converge or even converges to the

nearest low rank linearly structured matrix if we start the iteration with our given matrix A in (5.2). Therefore the authors in [29] suggest to use this algorithm as a tool to find a point in the intersection of both constraining sets. The idea is then by means of direct search optimization methods [79] to select a good starting point for the lift and projection algorithm.

A practical tool to solve the low rank minimization in the Frobenius norm or W-norm is the truncated SVD [127], [57, Section 2.5]. However, in the H-norm for general matrices $H \in \mathbb{R}^{n\times p}$ with $h_{ij} \neq 0$ only general nonlinear optimization techniques [126], [50] can be applied, which makes this approach impractical even for small dimensions. The minimum in the structure enforcement procedure can be found by e.g. the derivation in Section 1.5.2.

5.2.2 Transformation into a Structured Total Least Norm Problem

Let now $A \in \mathcal{L}$ in (5.2). Then another approach to tackle the problem (5.2) is to reformulate (5.2) into a structured total least norm problem as in [106]. Let us therefore first introduce what a structured total least norm problem is.

Definition 5.2.1. Let $U_1, \ldots, U_s \in \{0,1\}^{n\times(k+l)}$ and $[B\ C] \in \mathcal{L}(U_1, \ldots, U_s)$ with $B \in \mathbb{R}^{n\times k}$ and $C \in \mathbb{R}^{n\times l}$. Let further $\widehat{Q} \in \mathcal{S}^+_{n(k+l)}$ be the weighting matrix in the Q-norm in (5.4). Then we define the weighted *structured total least norm problem* as

$$\begin{aligned} \min_{E,R} \quad & \tfrac{1}{2}\|[E\ R]\|^2_{\widehat{Q}} \\ \text{s.t.} \quad & (B+E)Z = C + R \text{ for a } Z \in \mathbb{R}^{k\times l}, \\ & [E\ R] \in \mathcal{L}(U_1, \ldots, U_s). \end{aligned} \tag{5.4}$$

Note that if $U_1, \ldots, U_s$ in Definition 5.2.1 is a basis in $\mathbb{R}^{n\times(k+l)}$ and $\widehat{Q} = \widehat{Q}_1 \otimes \widehat{Q}_2$ for $\widehat{Q}_1 \in \mathcal{S}^+_{k+l}$, and $\widehat{Q}_2 \in \mathcal{S}^+_n$ diagonal then the definition coincides with the definition of the total least squares problem in [57, Chapter 12.3], [56]. If the optimal solution (E_*, R_*) of the problem in (5.4) exists then it gives the minimal perturbation in B and C in the Q-norm such that $\operatorname{range}(C+R_*) \subset \operatorname{range}(B+E_*)$. We use this property to reformulate our rank constraint in (5.2) as follows.

Let $\Upsilon_{a,b} : \mathbb{R}^{n\times p} \mapsto \mathbb{R}^{n\times(b-a+1)}$ be an operator with $\Upsilon_{a,b}(A) = [A_a\ A_{a+1}\ \cdots\ A_b]$, where A_i denotes the ith column of A. Further, let X_* be an optimal solution of (5.2) and $P \in \mathsf{O}(p)$ be a permutation matrix such that the range of the last $p-r$ columns of X_*P is a subset of the range of the first r columns of X_*P. Hence, $\operatorname{range}(\Upsilon_{r+1,p}(X_*P)) \subset \operatorname{range}(\Upsilon_{1,r}(X_*P))$. Note that this permutation matrix exists as $\operatorname{rank}(X_*) \leq r$. Then we can reformulate (5.2) with $X_1 := \Upsilon_{1,r}(XP)$,

$X_2 = \Upsilon_{r+1,p}(XP)$, $A_1 := \Upsilon_{1,r}(AP)$, and $A_2 := \Upsilon_{r+1,p}(AP)$ as

$$\begin{array}{ll} \min_{X_1,X_2} & \frac{1}{2}\|[A_1 - X_1 \; A_2 - X_2]P^T\|_Q^2 \\ \text{s.t.} & X_1 Z = X_2, \text{ for a } Z \in \mathbb{R}^{r\times(p-r)}, \\ & [X_1 \; X_2] \in \mathcal{L}(U_1P, \ldots, U_sP), \end{array} \tag{5.5}$$

where the constraint $X_1Z = X_2$ ensures that the rank of $X = [X_1 \; X_2]P^T$ is less than or equal to r. Note that for $B = A_1$, $C = A_2$, $E = -(A_1 - X_1)$, $R = -(A_2 - X_2)$, and $\widehat{Q} = (P^T \otimes I_n)Q(P \otimes I_n)$ (5.5) is of the form of (5.4) and is thus a structured total least norm problem.

The problem (5.5) is now 'easier' to solve than (5.2) since the constraint $X_1Z = X_2$ is linear and it allows to apply well known optimization techniques for nonlinear constrained optimization [102, Chapter 15] as we will see below. However, how to choose P in advance, and therefore the partition of A into A_1 and A_2 such that the problem (5.2) is equivalent to (5.5), and such that $X_1Z = X_2$ has a solution, is in general not clear. It is generally not known which columns of the optimal solution X_* of (5.2) are independent. However, if X is for example a Sylvester matrix, see Section 1.7.4, then the independent and dependent columns of X are known, allowing the reformulation of (5.2) into an equivalent structured total least norm problem [78].

The solution of the structured total least norm problem for unstructured matrices, i.e. $X \in \mathbb{R}^{n\times p}$, can be found in the Frobenius- and in the W-norm by applying the SVD [56]. The latter case can be reduced to the former by setting $\widehat{A} := W^{1/2}AW^{1/2}$ and $\widehat{X} = W^{1/2}XW^{1/2}$. The structured total least norm problem is then a total least squares problem. Note however, that the solution of such problems may not exist whereas fortunately, a sufficient condition for the existence can be derived. If we reformulate problem (5.5) into a total least squares problem with a single column as right-hand side as in [134, Section IV] and then apply [56, Theorem 4.1] we obtain the following sufficient condition for the existence and uniqueness of an optimal solution of (5.5). See an overview of total least squares methods in [95].

Lemma 5.2.2. *Let σ and $\widehat{\sigma}$ be the smallest singular values of $[I_{p-r} \otimes A_1 \; \mathrm{vec}(A_2)]$ and $I_{p-r} \otimes A_1$, respectively. If $\sigma < \widehat{\sigma}$ then there exists a unique solution of (5.5) in the Frobenius norm.*

Our aim however, is to compute the nearest low rank linearly structured matrix $X \in \mathcal{L}$. Since the solution obtained by applying the SVD to the unstructured total least squares does not generally preserve the structure of X a different approach is required to solve (5.5). Let us assume the permutation matrix P is known in (5.5). In this case the authors in [118] propose for $p - r = 1$ first to reformulate (5.5) as

$$\min_{x\in\mathbb{R}^s, z\in\mathbb{R}^{p-1}} \frac{1}{2}\|[A_1 - X_1(x) \; A_2 - X_2(x)]P^T\|_Q^2 + \mu\|X_1(x)z - X_2(x)\|_2, \tag{5.6}$$

where $X_1(x) = \sum_{i=1}^{s} x_i \Upsilon_{1,r}(U_i P)$ and $X_2(x) = \sum_{i=1}^{s} x_i \Upsilon_{r+1,p}(U_i P)$ and μ a penalty parameter, chosen large enough such that the *structural residual* $X_1(x)z - X_2(x)$ at the optimal solution of (5.6) is small. Secondly the authors suggest to solve the problem by applying an iterative algorithm where they linearize the objective function of (5.6) at the current iterate and solve the resulting problem to find a descent direction, yielding a new iterate. Similar ideas are pursued in [10], [134], where in the latter $p - r > 1$ is also considered. In this case the authors in [134] rewrite the structured total least norm problem for $p - r > 1$ as an equivalent structured total least norm problem with only one right-hand side, that is $p - r = 1$, allowing to apply the algorithms for (5.6). However, the linear system that needs to be solved in the optimization routine becomes significantly larger in dimension, resulting in algorithms that are computationally expensive and therefore impractical for $p - r$ large. Note that in [118] only a special case of (5.6) is considered, which does not include weights. Note further, as $A \in \mathcal{L}$ the dimension of $[A_1 - X_1(x)\ A_2 - X_2(x)]$ can be significantly reduced by accounting for the repetition of the elements of x in (5.6).

The approach described above is often analysed and further improved in terms of performance with regard to Toeplitz and Hankel matrices [134], [118], [106], [96], resulting in more efficient algorithms. For instance, in [106] Park et al. show by means of their numerical results that their algorithm for the nearest low rank Hankel matrix, based on structured total least norm, outperforms the alternating projection method of [27].

The drawback of reformulating the problem (5.2) and solving the resulting structured total least norm problem is surely that it is unclear how to choose the permutation matrix P in (5.5) such that the rank constraint is equivalently replaced. Moreover, in addition to the problem that a large penalty parameter can cause numerical difficulties, for $p - r > 1$ and for a general Q-norm the approach is computationally challenging and rather not applicable for practical applications.

5.2.3 Reformulating and Applying Geometric Optimization

Now we consider an approach that reformulates (5.2) into an optimization problem that requires to optimize over the Grassmannian manifold. See Section 3.8.2 for an introduction to this manifold. This approach is applicable for any Q-norm and the only requirement is that the matrix A is of the same linear structure as X in (5.2). The idea of this approach goes back to [93] that proposed a geometric optimization algorithm to find a nearest low rank approximation of unstructured rectangular matrices in the Q-norm. Based on this work Schuermans et al. published in [121] a similar approach for structured matrices that we now briefly introduce and improve

in the following sections.

First, let us define the function $f : \mathbb{R}^{p\times(p-r)} \mapsto \mathbb{R}$ with

$$f(N) := \min\left\{\frac{1}{2}\|A - X\|_Q^2 : X \in \mathcal{L}, XN = 0\right\}. \tag{5.7}$$

Note that the function f only depends on the subspace that is spanned by the columns of N. Therefore as observed by Manton et al. in [93] optimizing $f(N)$ over all matrices N that span different subspaces of dimension $p-r$ in $\mathbb{R}^{p\times p}$ is equivalent to (5.2). The set of all $p-r$ dimensional subspaces in $\mathbb{R}^{p\times p}$ is the Grassmannian manifold $\mathsf{Gr}(p, p-r)$ that we introduced in Section 3.8.2. Now in order to deal with these subspaces in a more convenient way we rather consider matrix representations of these subspaces whose columns are orthonormal and span the corresponding subspace. This is possible since we can always find a matrix representation to every element in $\mathsf{Gr}(p, p-r)$ and we are able to apply the geometric optimization tools to these matrix representations as we have seen in Section 3.8.2. Let us now simplify the notation for these matrix representations.

Let $\mathcal{Y} \in \mathsf{Gr}(p, p-r)$ and $\pi : \mathsf{St}(p, p-r) \mapsto \mathsf{Gr}(p, p-r)$ be the canonical projection defined in Definition 3.5.1. Then any $N \in \pi^{-1}(\mathcal{Y})$ is a matrix representation of $\mathcal{Y}$. For simplicity we now write $N \in \mathsf{Gr}(p, p-r)$ where N is a matrix representation of the corresponding equivalence class $\pi(N)$. Similarly, we deal with the tangent space. Let $T_{\mathcal{Y}}\mathsf{Gr}(p, p-r)$ be the tangent space of $\mathsf{Gr}(p, p-r)$ at $\mathcal{Y}$. Then from Section 3.5.3 we can define a bijective map $\tau_N : T_{\mathcal{Y}}\mathsf{Gr}(p, p-r) \mapsto H_N$ where H_N is the horizontal space at N whose elements are in $\mathbb{R}^{p\times(p-r)}$. Therefore from now on we write only $T_N\mathsf{Gr}(n, p)$ to mean the horizontal space at N. This allows us now to deal only with matrices.

By using these notations we can reformulate (5.2) as

$$\min_{N\in\mathsf{Gr}(p,p-r)} \quad f(N). \tag{5.8}$$

As we assume that A has the same structure as X we write $\mathrm{vec}(A) = Ua$ with U defined in (1.7) and $a \in \mathbb{R}^s$ and define a weighting matrix $\widehat{Q} := U^T Q U$, which is by definition symmetric and positive definite. Then by using the properties of the Kronecker product (Appendix A.1.2) we can rewrite the constraint $XN = 0$ in (5.7) as

$$\begin{aligned}\mathrm{vec}(XN) &= (N^T \otimes I_n)\mathrm{vec}(X)\\ &= (N^T \otimes I_n)Ux = 0\end{aligned}$$

and $f(N)$ in (5.7) simplifies to

$$f(N) = \min\left\{\frac{1}{2}(a-x)^T\widehat{Q}(a-x) : x \in \mathbb{R}^s, (N^T \otimes I_n)Ux = 0\right\}. \tag{5.9}$$

To obtain a solution of (5.9), Schuermans et al. [121] find stationary points of the corresponding Lagrangian function

$$L_N(x,\lambda) = \frac{1}{2}(a-x)^T\widehat{Q}(a-x) - \lambda^T(N^T \otimes I_n)Ux \tag{5.10}$$

where $\lambda = (\lambda_1, \ldots, \lambda_{n(p-r)})^T$ are the Lagrange multipliers. Differentiating $L_N(x,\lambda)$ and setting it to zero yields for $\widehat{U} := (N^T \otimes I_n)U$ and $\widehat{U} \in \mathbb{R}^{n(p-r)\times s}$ the linear system

$$\begin{bmatrix} \widehat{Q} & -\widehat{U}^T \\ \widehat{U} & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} \widehat{Q}a \\ 0 \end{bmatrix} \tag{5.11}$$

that needs to be solved.

By assuming that $\widehat{U}\widehat{Q}^{-1}\widehat{U}^T$ is of full rank and using the Schur complement of the coefficient matrix of (5.11) Schuermans et al. obtained the solution

$$x_* = (I_s - \widehat{Q}^{-1}\widehat{U}^T(\widehat{U}\widehat{Q}^{-1}\widehat{U}^T)^{-1}\widehat{U})a \tag{5.12}$$

of the linear system (5.11). Hence, x_* is the projection of a onto the orthogonal complement of the set spanned by $\widehat{U}$ with respect to the weighted Euclidean metric $\left\langle \widehat{Q}^{-1}x, y \right\rangle = y^T\widehat{Q}^{-1}x$. By substituting (5.12) into (5.9) $f(N)$ simplifies to

$$f(N) = \frac{1}{2}a^T\widehat{U}^T\left(\widehat{U}\widehat{Q}^{-1}\widehat{U}^T\right)^{-1}\widehat{U}a,$$

which needs to be minimized over the Grassmannian manifold.

The assumption of $\widehat{U}\widehat{Q}^{-1}\widehat{U}^T$ being of full rank can clearly be made if $\widehat{U}$ has full row rank. However, if $n(p-r) \geq s$, which holds in most applications of interest, we obtain either the trivial solution $x = 0$ or $\widehat{U}\widehat{Q}^{-1}\widehat{U}^T$ is singular. This observation was also made by the authors in [121] when they considered $\mathcal{L}$ to be the set of Hankel matrices. If in this case $p - r > 1$ corresponding to $n(p-r) \geq s$ their algorithm breaks down returning the trivial solution $x = 0$.

They remedied the problem but only for $\mathcal{L}$ being the set of Hankel matrices by showing that in this case (5.8) simplifies to

$$\min_{y \in \mathsf{St}(r+1,1)} \quad f(y) \tag{5.13}$$

with

$$f(y) = \frac{1}{2}a^T\widetilde{U}(y)^T\left(\widetilde{U}(y)\widehat{Q}\widetilde{U}(y)^T\right)^{-1}\widetilde{U}(y)a \tag{5.14}$$

and

$$\widetilde{U}(y) = \begin{bmatrix} y_1 & \cdots & y_{r+1} & 0 & 0 & \cdots & 0 \\ 0 & y_1 & \cdots & y_{r+1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \ddots & & \\ 0 & \cdots & 0 & y_1 & \cdots & & y_{r+1} \end{bmatrix} \in \mathbb{R}^{(n+p-1-r)\times(n+p-1)}$$

with $y = (y_1, \ldots, y_{r+1})^T$. The matrix $\widetilde{U}(y)$ is obviously of full row rank so that minimizing the objective function $f(y)$ in (5.14) will not break down.

Let y_* be the optimal solution of (5.13). Then the optimal solution of (5.2) is $X_* = \sum_{i=1}^{s}(x_*)_i U_i$ where similar to (5.12) x_* is the projection of a onto the orthogonal complement of the set spanned by the columns of $\widetilde{U}(y_*)$. Schuermans et al. used MATLAB's nonlinear least squares method `lsqnonlin`, to solve (5.13). However, they did not consider to take derivatives into account to possibly improve the performance, although this can be done by using e.g. the Fréchet derivative for orthogonal projections [55]. We will not pursue this idea any further.

Since the reformulation of (5.2) as the geometric optimization problem (5.8) is applicable for any Q-norm and any linear structure without any adjustments by the user it is most promising for our interest in this chapter. Therefore we investigate how we can solve (5.8) more closely and discuss improvements of the approach by Schuermans et. al in the next section. We will see that we can develop an algorithm returning a low rank approximation for any linear structure that does not break down and shows good performance. However, unfortunately we cannot guarantee convergence in general and have only numerical results as evidence that this algorithm works.

5.3 Steps to Our Method

Now, we will further investigate the last approach introduced in the previous Section 5.2.3. As we have seen when applying the Lagrangian method to (5.9) we obtain an optimal solution but only if $\widehat{U}\widehat{Q}^{-1}\widehat{U}^T$ is nonsingular. To remedy the latter problem one could compute the pseudo-inverse of $\widehat{U}\widehat{Q}^{-1}\widehat{U}^T \in \mathbb{R}^{n(p-r)\times n(p-r)}$ but for $n(p-r) \gg s$ there is an approach, which requires fewer operations and gives us further insight into the problem.

If we multiply the system (5.11) by the matrix $\operatorname{diag}\big(-I_s, \widehat{U}^T\big)$ from the left we obtain an equivalent linear system

$$\begin{bmatrix} -\widehat{Q} & \widehat{U}^T \\ \widehat{U}^T\widehat{U} & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -\widehat{Q}a \\ 0 \end{bmatrix}.$$

Then by substituting λ with $\lambda := \widehat{U}\widehat{\lambda}$ the latter linear system becomes

$$\begin{bmatrix} -\widehat{Q} & \widehat{U}^T\widehat{U} \\ \widehat{U}^T\widehat{U} & 0 \end{bmatrix} \begin{bmatrix} x \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} -\widehat{Q}a \\ 0 \end{bmatrix}, \tag{5.15}$$

which is of significantly smaller dimension than (5.11) if $n(p-r) \gg s$ and we see that the vector x that solves (5.11) also solves (5.15) and vice versa. Now let $F(N) :=$

$\widehat{U}^T\widehat{U}$. As this matrix is symmetric there exists a spectral decomposition of $F(N) = PDP^T$ where D is a diagonal matrix with the eigenvalues in decreasing order on the diagonal and $P \in \mathsf{O}(s)$. This allows us to rewrite (5.15) as

$$\begin{bmatrix} -P^T\widehat{Q}P & D \\ D & 0 \end{bmatrix} \begin{bmatrix} \widehat{x} \\ \widetilde{\lambda} \end{bmatrix} = \begin{bmatrix} -P^T\widehat{Q}a \\ 0 \end{bmatrix} \tag{5.16}$$

with $\widehat{x} = P^T x$ and $\widetilde{\lambda} = P^T\widehat{\lambda}$. Let t be the number of nonzero diagonal elements of D with $0 < t < s$ and $P = [P_1\ P_2]$ with $P_1 \in \mathsf{St}(s,t)$ and $P_2 \in \mathsf{St}(s, s-t)$. Note if $t = s$ we obtain $x = 0$ for the solution and for $t = 0$ we get $x = a$. Then from (5.16) follows that the first t entries of $\widehat{x}$ are zero and thus in MATLAB notation

$$\begin{bmatrix} -P_1^T\widehat{Q}P_2 & D(1:t,1:t) \\ -P_2^T\widehat{Q}P_2 & 0_{(s-t)\times t} \end{bmatrix} \begin{bmatrix} \widehat{x}(t+1:s) \\ \widetilde{\lambda}(1:t) \end{bmatrix} = \begin{bmatrix} -P_1^T\widehat{Q}a \\ -P_2^T\widehat{Q}a \end{bmatrix},$$

which always has a solution. We obtain $(\widehat{x}_{t+1}, \ldots, \widehat{x}_s)^T = (P_2^T\widehat{Q}P_2)^{-1}P_2^T\widehat{Q}a$ and thus $x = P_2(P_2^T\widehat{Q}P_2)^{-1}P_2^T\widehat{Q}a$. Hence, x depends only on the eigenspace spanned by the zero eigenvalues of $F(N)$ and is independent of the particular choice of P_2. If we substitute the solution x into (5.9) $f(N)$ simplifies to

$$f(N) = \frac{1}{2}a^T(\widehat{Q} - \widehat{Q}P_2(P_2^T\widehat{Q}P_2)^{-1}P_2^T\widehat{Q})a. \tag{5.17}$$

The problem with this approach is that the function $f(N)$ is highly dependent on the eigenspace corresponding to the zero eigenvalues of $F(N)$, which does not generally change continuously with N. If N changes either the eigenspace remains the same, meaning that the function value of $f(N)$ stays constant, or it does change but then the function value of $f(N)$ may jump. Hence, the function $f(N)$ is discontinuous. These circumstances make the minimization of $f(N)$ infeasible.

The idea in this section in order to overcome the problem discussed above is to apply the augmented Lagrangian method to

$$\begin{array}{ll} \min_{(N,x)\in\mathsf{Gr}(p,p-r)\times\mathbb{R}^s} & \frac{1}{2}(a-x)^T\widehat{Q}(a-x) \\ \text{s.t.} & (N^T \otimes I_n)Ux = 0, \end{array} \tag{5.18}$$

which is equivalent to (5.8). As we will see this has the advantage of only dealing with smooth functions.

We could also apply the quadratic penalty method. However, we experienced numerical difficulties for large penalty parameters in our numerical tests. Therefore we concentrate on the augmented Lagrangian method since this method diminishes the possibility that these numerical problems occur by introducing explicit Lagrange multipliers into the function to be minimized, see [102, Section 17.3]. Note that the set $\mathsf{Gr}(p, p-r) \times \mathbb{R}^s$ is a product manifold [3, Section 3.1.6].

5.3.1 Applying the Augmented Lagrangian Method

First, let us form the augmented Lagrangian function $G_{\mu,\lambda}(N,x)$ of (5.18), which is a combination of the standard Lagrangian function and the quadratic penalty function, penalizing the constraint $(N^T \otimes I_n)Ux = 0$. That is

$$G_{\mu,\lambda}(N,x) := \frac{1}{2}(a-x)^T\widehat{Q}(a-x) + \frac{\mu}{2}\|(N^T \otimes I_n)Ux\|_2^2 - \lambda^T(N^T \otimes I_n)Ux, \quad (5.19)$$

where $\lambda \in \mathbb{R}^{n(p-r)\times 1}$ are the Lagrange multipliers and $\mu > 0$ is the penalty parameter. As already mentioned in Section 4.6.4 the idea of the augmented Lagrangian method is at iteration k to first minimize $G_{\mu_k,\lambda^k}(N,x)$ over $(N,x) \in \mathsf{Gr}(p,p-r) \times \mathbb{R}^s$ for a fixed value of λ^k and μ_k and then increase μ_k as k increases and update, according to some rules, the Lagrange multipliers

$$\lambda^{k+1} = \lambda^k - \mu_k(N_k^T \otimes I_n)Ux_k \quad (5.20)$$

where (N_k, x_k) is the minimizer of $G_{\mu_k,\lambda^k}(N,x)$. See [102, Section 17.3] for more details on augmented Lagrangian methods.

Let us now consider how to solve

$$\min_{(N,x)\in\mathsf{Gr}(p,p-r)\times\mathbb{R}^s} G_{\mu,\lambda}(N,x) \quad (5.21)$$

for a fixed λ and μ. Let us first minimize $G_{\mu,\lambda}$ with respect to x. Differentiating $G_{\mu,\lambda}$ yields

$$\nabla_x G_{\mu,\lambda} = \left(\widehat{Q} + \mu F(N)\right)x - \widehat{U}^T\lambda - \widehat{Q}a.$$

Recall that $F(N) = U^T(N \otimes I_n)(N^T \otimes I_n)U$. The first order necessary optimality condition for x implies that

$$x(N) = \left(\widehat{Q} + \mu F(N)\right)^{-1}\left(\widehat{Q}a + \widehat{U}^T\lambda\right). \quad (5.22)$$

Note that the inverse of $\widehat{Q} + \mu F(N)$ exists for any μ since this matrix is square and of full rank. Moreover, since $\widehat{Q} + \mu F(N)$ is symmetric positive definite $x(N)$ in (5.22) is the unique global minimizer of $G_{\mu,\lambda}(\cdot, N)$. Substituting (5.22) into (5.21) and using the properties of the Kronecker product, see Appendix A.1.2, yields an equivalent problem to (5.21)

$$\min_{N\in\mathsf{Gr}(p,p-r)} f_{\mu,\lambda}(N) := -\frac{1}{2}y(N,\lambda)^T\left(\widehat{Q} + \mu F(N)\right)^{-1}y(N,\lambda), \quad (5.23)$$

where $y(N,\lambda) = \widehat{Q}a + U^T\mathrm{vec}(\Lambda N^T)$ with $\Lambda = \mathrm{vec}^{-1}(\lambda) \in \mathbb{R}^{n\times(p-r)}$. The advantage of this method in comparison with the method in [121] is that the term $\widehat{Q} + \mu F(N)$ is now invertible for any μ.

5.3.2 Steps to Compute $f_{\mu,\lambda}$

Before we can state our algorithm we need to discuss how to efficiently compute $f_{\mu,\lambda}(N)$ and $F(N)$, and as outlined in Section 3.9 we require to derive the derivative for $f_{\mu,\lambda}(N)$ to optimize over the Grassmannian manifold $\mathsf{Gr}(p, p-r)$.

First observe that the matrix $F(N)$ is symmetric so that is enough to consider how to compute the (i,j)th entry of this matrix for $i \leq j$ and $i, j = 1, \ldots, s$. Let us look at

$$\begin{aligned} F_{ij}(N) &= \operatorname{vec}(U_i)^T (NN^T \otimes I_n) \operatorname{vec}(U_j) \\ &= \operatorname{vec}(U_i)^T \operatorname{vec}(U_j NN^T) \\ &= \operatorname{trace}(U_i^T U_j NN^T). \end{aligned} \tag{5.24}$$

From (5.24) we note that the cost for computing the function value for every N can be significantly reduced if the matrices $U_i^T U_j$ for $i \leq j$ are precomputed. As the matrices U_j for $j = 1, \ldots, s$ are usually sparse the precomputation requires $s(s+1)/2$ sparse matrix-matrix multiplications. Note that this computation only depends on the structure of the matrix X and on the weighting Q so that the same matrices $U_i^T U_j$ can be used for different input matrices A and are required to be computed only once. If these matrices $U_i^T U_j$ are assumed to be given for $i \leq j$ and $i, j = 1, \ldots, s$ computing $F(N)$ requires $2p^2(p-r)$ operations to form NN^T and $2p^2$ operations to compute $\operatorname{trace}(U_i^T U_j NN^T)$ for all $i \leq j$. To determine $f_{\mu,\lambda}(N)$ we require another $s^3/3$ operations to compute the Cholesky factorization of $\widehat{Q} + \mu F(N)$ and $2n(p-r)p(s+1)$ operations to determine $U^T \operatorname{vec}(\Lambda N^T)$ in $y(N, \lambda)$. All remaining operations are of order $\mathcal{O}(s^2)$. In total this adds up to approximately $s^3/3 + 2p^2(p-r) + p^2 s^2 + 2n(p-r)p(s+1) + \mathcal{O}(s^2)$ operations to evaluate $f_{\mu,\lambda}(N)$ at N.

5.3.3 Forming the Derivative of the Objective Function

It remains to determine the derivative of $f_{\mu,\lambda}(N)$ in (5.23) for the geometric optimization. As the term $\widehat{Q} + \mu F(N)$ is invertible for all N this derivative exists. Let us first compute the Fréchet derivative, see Appendix A.2 for a definition. Therefore we define $K(N) := (\widehat{Q} + \mu F(N))^{-1}$ and $g(N) := K(N) y(N, \lambda)$. Then by using the product rule, see e.g. [66, Theorem 3.3] the Fréchet derivative of $f_{\mu,\lambda}(N)$ at N in direction $E \in \mathbb{R}^{p \times (p-r)}$ is

$$\begin{aligned} L_{f_{\mu,\lambda}}(N, E) =& -\frac{1}{2}\left(y(N,\lambda) L_g(N, E) + L_y(N, E)^T g(N)\right) \\ =& -\frac{1}{2} y(N,\lambda)^T \left(K(N) L_y(N, E) + L_K(N, E) y(N, \lambda)\right) \\ & -\frac{1}{2} L_y(N, E)^T g(N) \\ =& -L_y(N, E)^T g(N) - \frac{1}{2} y(N,\lambda)^T L_K(N, E) y(N, \lambda), \end{aligned} \tag{5.25}$$

where $L_K(N,E)$, $L_g(N,E)$, and $L_y(N,E)$ are the Fréchet derivative of $K(N)$, $g(N)$, and $y(N,\lambda)$, respectively. In order to obtain the derivative $\nabla f_{\mu,\lambda}$ we will use that $\nabla f_{\mu,\lambda} = (L_{f_{\mu,\lambda}}(N,E_{ij}))_{i,j=1}^{p,(p-r)}$ with $E_{ij} = e_i e_j^T \in \mathbb{R}^{p\times(p-r)}$. Hence, we need to compute $L_y(N,E_{ij})^T$ multiplied by $g(N)$ and $y(N)^T L_K(N,E_{ij})y(N)$. Let us first look at $L_y(N,E_{ij})^T g(N)$, which is

$$\begin{aligned} L_y(N,E_{ij})^T g(N) &= \mathrm{vec}(\Lambda E_{ij}^T)^T U g(N) \\ &= \mathrm{vec}(\Lambda e_j e_i^T)^T U g(N) \\ &= \mathrm{vec}(e_j e_i^T)^T (I_p \otimes \Lambda^T) U g(N) \\ &= \mathrm{vec}(e_j e_i^T)^T \mathrm{vec}\left(\Lambda^T \mathrm{vec}^{-1}(U g(N))\right). \end{aligned}$$

Hence

$$\left(L_y(N,E_{ij})^T g(N)\right)_{i,j}^{p,(p-r)} = \left(\mathrm{vec}^{-1}(U g(N))\right)^T \Lambda. \tag{5.26}$$

Let us now look at the functions $t(M) := (I_s+M)^{-1}$ and $F(N)$ and let us compute their Fréchet derivatives. We have for $\widehat{E} \in \mathbb{R}^{s\times s}$ and $\widehat{M} = I_s + M$

$$\begin{aligned} t(M+\widehat{E}) &= (\widehat{M}+\widehat{E})^{-1} \\ &= \widehat{M}^{-1} - \widehat{M}^{-1}\widehat{E}\widehat{M}^{-1} + \mathcal{O}(\|\widehat{E}\|^2), \end{aligned}$$

implying that $L_t(M,\widehat{E}) = -\widehat{M}^{-1}\widehat{E}\widehat{M}^{-1}$. Furthermore, it is easily verified that $L_F(N,E) = U^T((NE^T+EN^T)\otimes I_n)U$. Observe then by using the chain rule for the Fréchet derivative [66, Theorem 3.4] that

$$\begin{aligned} & y(N,\lambda)^T L_K(N,E) y(N,\lambda) \\ = \;& y(N,\lambda)^T \widehat{Q}^{-1/2} L_t\Big(\mu\widehat{Q}^{-1/2}F(N)\widehat{Q}^{-1/2}, \mu\widehat{Q}^{-1/2}L_F(N,E)\widehat{Q}^{-1/2}\Big)\widehat{Q}^{-1/2} y(N,\lambda) \\ = \;& -\mu y(N,\lambda)^T K(N) U^T((NE^T+EN^T)\otimes I_n) U K(N) y(N,\lambda) \\ = \;& -2\mu g(N)^T U^T (EN^T \otimes I_n) U g(N). \end{aligned}$$

Therefore with

$$X(N) := \mathrm{vec}^{-1}\left(U g(N)\right) \in \mathbb{R}^{n\times p} \tag{5.27}$$

we have

$$\left(y(N,\lambda)^T L_K(N,E_{ij}) y(N,\lambda)\right)_{i,j=1}^{p,(p-r)} = -2\mu X(N)^T X(N) N$$

and it follows together with (5.25) and (5.26) that

$$\nabla f_{\mu,\lambda} = -X(N)^T(\Lambda - \mu X(N)N). \tag{5.28}$$

The matrix $X(N)$ is equal to $\sum_{i=1}^s x_i(N)U_i$ and is therefore our matrix of interest. Furthermore, $X(N)N$ is our constraint of (5.18) so that the term $(\Lambda - \mu X(N)N)$ in (5.28) is exactly the term that we use to update Λ in (5.20) in the augmented Lagrangian method. Note that $K(N)$ also occurs in the objective function. Therefore

the expensive operations to determine e.g. the Cholesky decomposition of $\widehat{Q}+\mu F(N)$ have already been carried out when the objective function was computed. Hence, the additional cost for determining $g(N)$ is only of $\mathcal{O}(s^2)$ operations. It remains to compute $Ug(N)$, $\Lambda - \mu X(N)N$, and $X(N)^T(\Lambda - \mu X(N)N)$. In order, this requires $2nps$, $(2p+1)n(p-r)$, and $2np(p-r)$ operations, which is in total $2np(s+2(p-r))+n(p-r)+\mathcal{O}(s^2)$ operations. Note if e.g. $\mathcal{L}$ is the set of Hankel matrices then $s = p+n-1$ and the derivative is relatively cheap to obtain in comparison to computing the objective function in Section 5.3.2.

5.3.4 Convergence

Recall from Section 4.6.4 that Yang and Zhang showed in [142] that similarly to $\mathbb{R}^n$, necessary and sufficient optimality conditions for nonlinear programming problems over Riemannian manifolds can be derived and that the concept of Lagrange multipliers can analogously be applied. In particular, we showed in Theorem 4.6.12 that if a point Y_* in a Riemannian manifold $\mathcal{M}$ embedded in $\mathbb{R}^n$ is a local solution of the programming problem then under some conditions Y_* is a local solution of the corresponding augmented Lagrangian function. In $\mathbb{R}^n$ under the assumptions of this theorem Bertsekas obtained a convergence result for the sequence generated by the augmented Lagrangian method [13, Proposition 4.2.3]. This is our motivation to check the assumptions of Theorem 4.6.12 for problem (5.18).

Let $c(N,x) := (N^T \otimes I_n)Ux \in \mathbb{R}^{n(p-r)\times 1}$. As the objective function and the constraints $c(N,x)$ of (5.18) are smooth we fulfil the first requirement. However, we also need two further requirements: the first one is that at a local solution of (5.18) (N_*, x_*) the LICQ defined in Definition 4.6.9 is satisfied, giving us the existence of the Lagrange multipliers λ_* at (N_*, x_*), and the second is that

$$\langle Z, \operatorname{Hess} G_{0,\lambda_*}(N_*, x_*)[Z]\rangle > 0$$

for all $Z \in T_{N_*}\mathsf{Gr}(p, p-r)\times\mathbb{R}^s$ with $\langle \operatorname{grad} c_i(N_*, x_*), Z\rangle = 0$ for all $i = 1, \ldots, n(p-r)$, and $Z \neq 0$. Recall that $\operatorname{Hess} G_{0,\lambda_*}(N_*, x_*)[Z]$ denotes the Hessian operator of $G_{0,\lambda_*}(N,x)$ that we defined in (3.9) and $G_{0,\lambda_*}(N_*, x_*)$ is the function in (5.19).

We will not look at the Hessian operator, we will rather investigate the LICQ and we will show by means of two examples that this condition may or may not be satisfied, depending on the problem. Therefore we cannot guarantee convergence of this algorithm in general. To check the LICQ we need to compute the gradient of $c_i(N,x)$ on the product manifold $\mathsf{Gr}(p,p-r)\times\mathbb{R}^s$ at (N,x). Recall from Section 3.8.2 that the gradient on the Grassmannian manifold is represented by the element $(I_n - NN^T)\nabla f(N)$. Therefore the gradient on the product manifold $\operatorname{grad} c_i(N,x)$ for all $i =$

$1, \ldots, n(p-r)$ at (N_*, x_*) of (5.18) is $\operatorname{grad} c_i(N,x) = (\operatorname{grad}_N c_i(N,x), \operatorname{grad}_x c_i(N,x)) \in T_N\mathsf{Gr}(p-r,p) \times \mathbb{R}^s$. Let us first determine $\operatorname{grad}_x c_i(N,x)$ that is

$$\operatorname{grad}_x c_i(N,x) = \nabla_x c_i(N,x) = [U^T(N \otimes I_n)]_i \in \mathbb{R}^s,$$

where $[A]_i$ denotes the ith column of A. By using similar techniques as in the previous section we obtain that the gradient $\operatorname{grad}_N c_i(N,x)$ reshaped in a long vector is

$$\operatorname{vec}(\operatorname{grad}_N c_i(N,x)) = [I_{p-r} \otimes ((I_p - NN^T)X^T)]_i = [I_{p-r} \otimes (X^T)]_i \tag{5.29}$$

for all $i = 1, \ldots, n(p-r)$ where $X = \sum_{i=1}^s x_i U_i$. The latter equality in (5.29) holds since we have for a solution of (5.18) that $XN = 0$. Hence, the LICQ is equivalent to the condition that the matrix

$$\begin{bmatrix} U^T(N \otimes I_n) \\ I_{p-r} \otimes (X^T) \end{bmatrix} \in \mathbb{R}^{((p-r)p+s) \times n(p-r)} \tag{5.30}$$

is of full column rank at (N_*, x_*). At this point the matrix $I_{p-r} \otimes (X^T)$ is at most of row rank $r(p-r)$ and as we have that $X_* N_* = 0$ with $X_* = \sum_{i=1}^s (x_*)_i U_i$ the matrix

$$U^T(N \otimes I_n) = [\operatorname{vec}(U_1)N, \ldots, \operatorname{vec}(U_s)N]^T$$

is at most of row rank $s-1$ if $x_* \neq 0$. Therefore the necessary condition that the matrix in (5.30) has full column rank in case of x_* nonzero is that

$$s \geq (n-r)(p-r) + 1. \tag{5.31}$$

For $x_* = 0$ we require at least $s \geq n(p-r)$ as $I_{p-r} \otimes (X^T)$ is of rank zero. The next example shows that even if $s \geq (n-r)(p-r)+1$ and x_* nonzero the LICQ does not need to be satisfied. Let

$$Q = \frac{1}{2} I_4,\ a = \begin{bmatrix} 1.2 \\ 1 \end{bmatrix},\ U_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},\ U_2 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},\ \text{and } r = 1$$

in (5.18) so that U is of full rank and $\widehat{Q} = I_2$. We attain the optimal solution value at $N_* = \frac{1}{\sqrt{2}}(-1,1)^T$ and $x_* = 1.1(1,1)^T$. At this point the matrix

$$U^T(N \otimes I_n) = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{ and } I_{p-r} \otimes (X^T) = 1.1 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Therefore the matrix in (5.30) is only of rank one and the LICQ is not satisfied despite that s is equal to $s = (n-r)(p-r)+1 = 2$.

On the other hand the LICQ can also be satisfied at the optimal point as we see in the next example. Consider

$$Q = I_4,\ a = \begin{bmatrix} 2 \\ 1 \end{bmatrix},\ U_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\ U_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},\ \text{and } r = 1 \tag{5.32}$$

in (5.18). Then U has full rank and $\widehat{Q} = I_2$. The optimal solution (N_*, x_*) of (5.18) is clearly obtained with $N_* = (0,1)^T$ and $x_* = (2,0)^T$. Since at this point $U^T(N \otimes I_n) = U_2$ and $I_{p-r} \otimes (X^T) = 2U_1$ the matrix in (5.30) is of full column rank and the LICQ is satisfied.

Since our constraint $X(N)N$ appears in the derivative of $f_{\mu,\lambda}$ in (5.28) we can derive some results that relate the geometric gradient $\operatorname{grad} f_{\mu,\lambda}$ with classical derivative $\nabla f_{\mu,\lambda}$.

Lemma 5.3.1. *Let $\mu\|X(N)N\|_F \leq \varepsilon$, $\|\operatorname{grad} f_{\mu,\lambda}(N)\|_F \leq \varepsilon_2$ and Λ be bounded then with $c = \big((\sqrt{p}+p-r+1)\|X(N)\|_F + \sqrt{p}\|\Lambda\|_F\big)$ and $\mu \geq 1$*

$$\|\nabla f_{\mu,\lambda}\|_F \leq \varepsilon c + \varepsilon_2,$$

where $\operatorname{grad} f_{\mu,\lambda}(N)$ is the gradient of $f_{\mu,\lambda}(N)$ in the tangent space $T_N\mathsf{Gr}(p,p-r)$ at N; see Section 3.8.2.

Proof. From (3.18), (5.28) we have that $\operatorname{grad} f_{\mu,\lambda}(N) = (I_p - NN^T)(\mu X(N)^T X(N) - X(N)^T\Lambda)$. By using that $\|B-A\|_F \geq \|B\|_F - \|A\|_F$ for $A, B \in \mathbb{R}^{n\times p}$

$$\begin{aligned}
\varepsilon_2 &\geq \|(I_p - NN^T)X(N)^T(\Lambda - \mu X(N)N)\|_F \\
&\geq \|(I_p - NN^T)X(N)^T\Lambda\|_F - \mu\|(I_p - NN^T)X(N)^TX(N)N\|_F \\
&\geq \|(I_p - NN^T)X(N)^T\Lambda\|_F - \mu\|I_p - NN^T\|_F\|X(N)\|_F\|X(N)N\|_F.
\end{aligned}$$

Hence,

$$\|(I_p - NN^T)X(N)^T\Lambda\|_F \leq \varepsilon_2 + \varepsilon(\sqrt{p}+p-r)\|X(N)\|_F.$$

Then from

$$\|(I_p - NN^T)X(N)^T\Lambda\|_F \geq \|X(N)^T\Lambda\|_F - \|N\|_F\|X(N)N\|_F\|\Lambda\|_F,$$

we have

$$\begin{aligned}
\|X(N)^T\Lambda\|_F &\leq \|(I_p - NN^T)X(N)^T\Lambda\|_F + \|N\|_F\|X(N)N\|_F\|\Lambda\|_F \\
&\leq \varepsilon_2 + \varepsilon((\sqrt{p}+p-r)\|X(N)\|_F + \sqrt{p}\|\Lambda\|_F)
\end{aligned}$$

and we obtain that

$$\begin{aligned}
\|\mu X(N)^TX(N)N - X(N)^T\Lambda\|_F &\leq \varepsilon\|X(N)\|_F + \|X(N)^T\Lambda\|_F \\
&\leq \varepsilon_2 + \varepsilon\big((\sqrt{p}+p-r+1)\|X(N)\|_F + \sqrt{p}\|\Lambda\|_F\big).
\end{aligned}$$

□

As $\|X(N)\|_F$ is bounded the previous Lemma 5.3.1 implies that if $\mu_k X(N_k)N_k$ and $\operatorname{grad} f_{\mu_k,\lambda^k}(N_k)$ is going to zero then also the matrix of partial derivatives $\nabla f_{\mu_k,\lambda^k}(N_k)$ is going to zero. Note also in the penalty method where $\lambda = 0$ all the points N with $X(N)N = 0$ and $X(N)$ determined as in (5.27) are stationary points of $f_{\mu,0}$ as

$$\begin{aligned}
\|\operatorname{grad} f_{\mu,0}(N)\|_F &\leq \mu\|(I_p - NN^T)X(N)^TX(N)N\|_F \\
&\leq \mu(\sqrt{p}+(p-r))\|X(N)\|_F\|X(N)N\|_F.
\end{aligned}$$

5.3.5 Our Algorithm

We state our proposed method in Algorithm 5.3.1 where, unlike in Section 4.7, we do not use the practical augmented Lagrangian algorithm proposed in [102, Algorithm 17.4]. We rather base our augmented Lagrangian algorithm on the [102, Framework 17.3] and initiate and update our parameters as shown in Algorithm 5.3.1. The reason for not using [102, Algorithm 17.4] is that the configurations for the parameters in Algorithm 5.3.1 yield empirically significantly better numerical results. The main difference is first that we let the initial penalty parameter and the magnitude of increase on line 10 depend on the function value (5.17). The second is that we use quite a large initial tolerance for the stopping criterion on line 4 and tighten it only moderately in every iteration. To determine the objective function value of (5.23) we need to compute $z = (\widehat{Q} + \mu F(N))^{-1} y(N, \lambda)$. For large value of μ we expect the matrix $\widehat{Q} + \mu F(N)$ to be ill-conditioned. Nocedal and Wright discuss ideas in [102, Section 17.1] to reduce the condition number of such a system by solving an equivalent larger linear system

$$\begin{bmatrix} \widehat{Q} & L^T \\ L & -\frac{1}{\mu} I_s \end{bmatrix} \begin{bmatrix} z \\ \eta \end{bmatrix} = \begin{bmatrix} y(N, \lambda) \\ 0_{s \times 1} \end{bmatrix}$$

with $F(N) = L^T L$. However, numerical tests have shown that the actual condition number of the coefficient matrix is increased if we use this approach. Therefore we use the Jacobi preconditioner, that is to scale the diagonal of $\widehat{Q}+\mu F(N)$ to one. Let D be the diagonal of $\widehat{Q}+\mu F(N)$ then we solve $Ax = b$ by using the Cholesky decomposition of A with $A = D^{-1/2}(\widehat{Q} + \mu F(N))D^{-1/2}$, $x = D^{1/2} z$, and $b = D^{-1/2} y(N, \lambda)$. In contrast to the approach above, we do not need to solve a larger system and our numerical tests have shown that this approach yields a significant reduction of the condition number of the system. Note that the Jacobi preconditioner has also a theoretical motivation. By [64, Corollary 7.3]

$$\kappa_2(A) \leq s \min_{D \in \mathcal{D}_s} \kappa_2(D(\widehat{Q} + \mu F(N))D)$$

where $\mathcal{D}_s$ is the set of all diagonal matrices in $\mathbb{R}^{s \times s}$ and $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$.

In the next section we will investigate the performance of Algorithm 5.3.1 and we will also look at which of the algorithms introduced in Section 3.9 is most suitable to solve the inner problem on line 4 of Algorithm 5.3.1.

5.4 Computational Experiments

In this section we concentrate on investigating how well our Algorithm 5.3.1 proposed in the previous section works by performing several numerical tests. Our purpose is

Algorithm 5.3.1 This algorithm finds the nearest low rank linearly structured matrix to a given linearly structured matrix by minimizing (5.18).

Require: $U_1, \ldots, U_s$, $a \in \mathbb{R}^s$, $r \in \mathbb{N}$ with $r \leq \operatorname{rank}(\sum_{i=1}^{s} a_i U_i)$, $\widehat{Q} \in \mathbb{R}^{s \times s}$ symmetric positive definite, and $N_0 \in \mathsf{Gr}(p, p-r)$.

1 Apply the augmented Lagrangian method to (5.18) based on [102, Framework 17.3]:
2 Set $k_{\max} = 100$, $\lambda^0 = 0$, $\mu_0 = \frac{p-r}{4p} f(N_0)$ with $f(N_0)$ defined in (5.17), the tolerance $\omega_0 = p(p-r)10^{-3}$, and the violation tolerance $\eta_* = n(p-r)^2 10^{-11}$.
3 **for** $k = 0 : k_{\max} - 1$ **do**
4 Find an approximate minimizer N_{k+1} of (5.23) starting from N_k such that

$$\|\operatorname{grad} f_{\mu_k, \lambda^k}(N_{k+1})\|_F^2 \leq \omega_k$$

by using either the nonlinear CG method, Algorithm 3.9.1, or the limited memory RBFGS-method, Algorithm 3.9.3.
5 **if** $\|X(N_{k+1})N_{k+1}\|_F^2 \leq \eta_*$ **then**
6 **break**
7 **end if**
8 Update Lagrange multipliers $\lambda^{k+1} = \lambda^k - \mu_k \operatorname{vec}(X(N_{k+1})N_{k+1})$ with $X(N_{k+1})$ defined in (5.27).
9 Update the tolerance $\omega_{k+1} = \omega_k / 1.01$.
10 Update penalty parameter $\mu_{k+1} = \mu_k + \frac{(k+1)^2(p-r)}{8p}(f(N_{k+1}) + 1)$.
11 **end for**
12 **return** $X(N_{k+1})$ in (5.27).

to find a nearest low rank linearly structured matrix to a given linearly structured matrix by solving (5.18) with different values for a, r and $U_1, \ldots, U_s$. Let us first introduce our test matrices.

5.4.1 Test Matrices

In all our tests we use $\widehat{Q} = U^T U$ where U is defined in (1.7). We choose the matrices $U_1, \ldots, U_s$ out of three different classes.

- **uexample**: The first one is the example in (5.32) where we know that the LICQ is satisfied at the optimal solution.

- **uhankel**: In the second class the matrix U is chosen such that $\mathcal{L}$ is identical to the set of Hankel matrices of given dimension n-by-p, see Section 1.7.2. As by [106, Theorem 2.1], [30, Theorem 3.3] a nontrivial low rank solution always exists for square Hankel matrices this class seems suitable for our tests in particular for $n = p$, although better algorithms as mentioned in Section 5.2 are available to find a low rank Hankel matrix. We choose

$$U_1 = \begin{bmatrix} 1 & 0_{1\times(p-1)} \\ 0_{1\times(n-1)} & 0_{(n-1)\times(p-1)} \end{bmatrix}, \; U_2 = \begin{bmatrix} 0 & 1 & \\ 1 & 0 & 0_{2\times(p-2)} \\ 0_{(n-2)\times 2} & & 0_{(n-2)\times(p-2)} \end{bmatrix}, \; \ldots,$$

$$U_s = \begin{bmatrix} 0_{(n-1)\times(p-1)} & 0_{1\times(n-1)} \\ 0_{1\times(p-1)} & 1 \end{bmatrix} \text{ with } s = n + p - 1.$$

  The vector a in (5.18) is determined by the MATLAB `rand` function.

- **urand**: The third class is drawn from randomly generated matrices $U_1, \ldots, U_s$ that are computed as follows. Let n, p, $s \in \mathbb{N}$ with $n \geq p$ be given. For $k = 1 : (s-1)$ we choose randomly by means of the MATLAB function `rand` a number $n_k \in \left\{1, \ldots, np - (s-k) - \sum_{j=1}^{k-1} n_j\right\}$ and determine randomly n_k indices (i, j) for $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$ that have not been chosen before. Let $\mathcal{I}_k$ be the set of chosen indices at iteration k. Then we set for all $l = 1, \ldots, n$ and $q = 1, \ldots, p$

$$(U_k)_{lq} = \begin{cases} 1 & \text{for } (l, q) \in \mathcal{I}_k \\ 0 & \text{otherwise.} \end{cases} \tag{5.33}$$

  Let $\mathcal{I}_s$ be the set of indices that have not been chosen after $(s-1)$ iterations. Then we set U_s as in (5.33). This procedure ensures that the matrix U as defined in (1.7) is of full rank. We choose the vector a as in the second class. Note that in general we cannot guarantee for this class that there is a nontrivial solution of (5.18).

5.4.2 Numerical Methods

Let us now look at the numerical methods. Since all the methods that we will look at solve a different problem to find the nearest low rank linearly structured matrix it is difficult to find a common stopping criterion. Therefore we use for every method a stopping criterion that is suitable for the method. However, we make sure that the constraint violations of all returned points are at least as small as at the points returned by Algorithm 5.3.1 as we are mainly interested in how large the constraint violation of the returned points and the distance to the given matrix A is. In addition to Algorithm 5.3.1 we look at the following methods to find the nearest low rank linearly matrix.

- LIFTPROJ: The first is the lift and projection algorithm described in Section 5.2.1, [27], [30]. This algorithm has no convergence guarantee and is only applicable for the W-norm if the truncated SVD is used to compute the nearest rank k matrix. As we have chosen Q to be the identity we can use the truncated SVD. For this method we use our own implementation in MATLAB. Let Y_k be the optimal solution of the unstructured low rank minimization and Z_k the optimal solution of the subsequent enforcement procedure at iteration k; see Section 5.2.1. We stop the iteration of this algorithm if after the structure enforcement procedure at iteration k

$$\sum_{i=1}^{p-r} \sigma_i \leq \varphi, \tag{5.34}$$

  where $\sigma_1, \ldots, \sigma_{p-r}$ are the smallest $(p-r)$ singular values of the current iterate Z_k and φ is a tolerance that we specify later. As the method may not converge to an intersection point we also stop the iteration if for $k > 100$

$$\frac{\left| \|Y_k - Z_k\|_F^2 - \|Y_{k-100} - Z_{k-100}\|_F^2 \right|}{1 + \|Y_k - Z_k\|_F^2} \leq npu, \tag{5.35}$$

  where u is the unit roundoff, which is in double precision arithmetic $u = 2^{-53}$. We set our maximal number of iterations to $100,000$. As proposed by [29] we use different starting values for this method. However, we do not use direct search solvers as the matrices are too large. We therefore use $A = \sum_{i=1}^{s} a_i U_i$ and 10 different randomly generated starting matrices with the same linear structure as A and return the matrix with the smallest function value that satisfies (5.34).

- FMINCON: The next method is only applicable if the matrix A in (5.2) is square and of full rank r_A and we are seeking a matrix of this structure with

rank less than or equal to $r_A - 1$. In this case we can reformulate the problem as

$$\begin{array}{ll} \min_{X \in \mathcal{L}} & \frac{1}{2}\|A - X\|_Q^2 \\ \text{s.t.} & \det(X) = 0 \end{array} \tag{5.36}$$

with $\det(X)$ the determinant of $X \in \mathbb{R}^{n \times n}$. We apply the MATLAB function `fmincon`, which is a subspace trust-region method based on the interior-reflective Newton method [31], [32], to this problem, where we provide the first and second derivative of the objective function of (5.36) but we do not provide the derivatives of the constraint. Furthermore, we use the same stopping criterion (5.34) as for LIFTPROJ.

As the matrix of derivatives of $\det(X)$ is $\nabla \det(X) = \det(X)(X^{-1})^T$ [114], [92, p. 179] we expect numerical difficulties for X close to the solution of (5.36). Therefore this method is certainly not practical for general usage and we consider it only for comparison.

5.4.3 Numerical Tests

All our tests are performed in MATLAB R2010a on an Intel(R) Xeon(R) with 3GHz with eight cores and 16GB RAM, Scientific Linux release 6.1 (Carbon).

Test 1

In our first test we investigate the performance of the algorithms introduced in Section 3.9 for solving the inner problem on line 4 of Algorithm 5.3.1. Our specifications for the nonlinear CG and the limited RBFGS method are as follows. Let $A = \sum_{i=1}^{s} a_i U_i$ be the given matrix in (5.2) and $A = P\Sigma V^T$ its SVD decomposition with the diagonal elements of Σ in decreasing order. Then we use in MATLAB notation

$$N_0 = V(:, p - r + 1 : p) \tag{5.37}$$

as our starting matrix in Algorithm 5.3.1. To continue, we also limit the maximum number of iterations to $100,000$ for both the nonlinear CG and the RBFGS algorithm and since we do not have a good initial step length for the problem considered as in Section 4.7 we apply a more sophisticated line-search strategy in these algorithms. For the nonlinear CG method we use the algorithm proposed in [41, Algorithm A6.3.1]. Similar to the Armjio-backtracking procedure stated in Algorithm 3.9.1, this algorithm also uses a backtracking strategy; the difference is that the desired step size is found by using cubic interpolation of the objective function. Since the limited memory RBFGS algorithm, Algorithm 3.9.3 is a secant method we also need to check whether the step length is large enough. Therefore we use [41,

Table 5.1: Performance of Algorithm 5.3.1 for different methods to solve (5.23) for test matrices **uhankel** and $r = n - 5$.

	$\overline{\mathbf{t}}$					$\overline{\mathbf{it}}$				
n	**50**	**80**	**100**	**120**	**150**	**50**	**80**	**100**	**120**	**150**
RB M=1	68	331	922	1710	3192	1753	4523	6631	9435	1.1e4
RB M=5	50	156	241	664	963	1055	1815	1891	3843	3507
RB M=10	43	138	212	645	822	730	1442	1481	3628	3169
RB M=20	32	77	143	440	602	459	696	862	2035	2075
RB M=30	42	83	145	369	409	474	625	842	1627	1435
RB M=40	47	93	171	434	427	523	662	861	1717	1376
RB M=50	49	86	141	312	415	528	615	736	1256	1272
CG-PF	58	316	501	834	959	1221	2718	2525	2864	2162
CG-Geo	41	187	431	844	954	787	1761	2076	2782	2097

Algorithm A6.3.1mod] for the line-search strategy in Algorithm 3.9.3, which is [41, Algorithm A6.3.1] with this additional check on the step length. Having specified the line-search strategy it remains to determine which retraction and vector transport is to use. To guarantee in the RBFGS method that $\rho_k > 0$ in (3.31) we will only consider to use the geodesic (3.19) for the retraction and the parallel translation in (3.22) for the vector transport as in this case the condition $\rho_k > 0$ is always satisfied. Which retraction to use for the nonlinear CG method will be part of our investigation of this test. For the version of Polak-Ribière we will look at the performance when using two different retractions. The first is the geodesic in (3.19) for which we use the parallel translation in (3.22) for the vector transport (CG-Geo) and the second is the unitary polar factor defined in (3.20) combined with the vector transport in (3.21) (CG-PF). As in Section 4.7 if the direction $\xi_{x_{k+1}}$ in Algorithm 3.9.1 is not a descent direction we use the steepest descent direction $-\operatorname{grad} f(x_{k+1})$.

For the limited memory RBFGS method (RB) we will consider the performance for different values of M in Algorithm 3.9.2, that is the maximal number of pairs (y_i, s_i) stored and used to approximate the Hessian.

Note that we also investigated the version of the nonlinear CG method proposed by Fletcher and Reeves but as from our tests we clearly observed that this version is not competitive we omitted the results. In our tests the version of Fletcher-Reeves takes at least a factor of 10 more iterations in the nonlinear CG method and performs clearly worse in terms of time than the version of Polak-Ribière and often fails to converge to a point that satisfies our stopping criterion. The failure is caused in the backtracking procedure due to limitations of the smallest step size allowed.

For the test in this section we generate 5 instances of square Hankel matrices of

type **uhankel** with dimension $n = 50, 80, 100, 120, 150$ and try to reduce the rank of $A = \sum_{i=1}^{s} a_i U_i$ by 5 and 10 with a and $U_1, \ldots, U_s$ as specified for **uhankel**. We report a selection of our results in Table 5.1 and Table 5.2 where we use the following abbreviations:

- $\bar{\mathbf{t}}$: mean computation time (in seconds) taken to run Algorithm 5.3.1.
- $\overline{\mathbf{it}}$: mean total iteration number taken in either the CG algorithm or in the limited memory RBFGS method to solve the inner problem on line 4 of Algorithm 5.3.1.
- $\overline{\mathbf{f}(\mathbf{X}_*)}$: mean objective function value $\frac{1}{2}\|A - X_*\|_Q^2$ of (5.2) at the returned point X_*.
- $\overline{\text{Assv}}$: mean sum of the $p - r$ smallest singular values of A.
- $\overline{\mathbf{Xssv}}$: mean sum of the $p - r$ smallest singular values of X_*.
- $\overline{\mu_*}$: mean value of μ at the last iterate.

Note that we observed no significant variation between the 5 instances during our tests so that the mean values displayed in Table 5.1 and Table 5.2 are good representation of the overall performance of Algorithm 5.3.1 and the methods to solve the inner problem. Note further that for all our tests we do not time the seconds that are spent on computing the $s(s+1)/2$ sparse matrix-matrix-multiplications $U_i^T U_j$ in (5.24), although for n and p large computing these products can make a large contribution to the total time. However, in all our tests n and p are sufficiently small so that the contribution to the total computation time is negligible.

From Table 5.2 we observe that the constraint violation is reduced by applying Algorithm 5.3.1 and that the points X_* returned by Algorithm 5.3.1 for the different methods used to solve the inner problem are comparable. If we look at the performance in Table 5.1 we see that in terms of iterations and time taken the limited memory RBFGS clearly outperforms the nonlinear CG if M is large enough. For which value of M the best performance is achieved depends thereby on the value of n. The larger n the more stored pairs (y_i, s_i) are required to obtain the best performance. One reason for the superiority of the RBFGS method is also observed by looking at the iterations required in the backtracking procedure. For instance, the RBFGS method with $M = 50$ takes only about a third of the number of iterations in the backtracking procedure than taken by the nonlinear CG method. Therefore the computational overhead for computing the descent direction in the RBFGS in comparison to the CG method pays off and yields eventually better performance.

Table 5.2: Results for Algorithm 5.3.1 for different methods to solve (5.23) for test matrices **uhankel**.

	$r = n - 5$			$r = n - 10$		
	$\overline{\mathbf{f}(\mathbf{X}_*)}$	$\overline{\mathbf{Xssv}}$	$\overline{\mu_*}$	$\overline{\mathbf{f}(\mathbf{X}_*)}$	$\overline{\mathbf{Xssv}}$	$\overline{\mu_*}$
n = 50	$\overline{\text{Assv}}$= 1.37			$\overline{\text{Assv}}$= 4.46		
RB M=1	1.163	6.9e-5	1571	3.640	1.3e-4	2188
RB M=30	1.109	3.6e-5	1751	3.634	1.2e-4	1615
RB M=50	1.109	4.8e-5	1731	3.637	8.7e-5	1502
CG-Geo	1.176	6.9e-5	2088	3.638	1.6e-4	1898
n = 100	$\overline{\text{Assv}}$= 1.09			$\overline{\text{Assv}}$= 3.79		
RB M=1	0.924	8.4e-5	2167	3.865	1.6e-4	2767
RB M=30	0.959	5.6e-5	3175	3.681	1.4e-4	2750
RB M=50	0.965	8.7e-5	2284	3.728	1.2e-4	2724
CG-Geo	0.970	9.2e-5	2638	4.291	9.2e-5	3286
n = 150	$\overline{\text{Assv}}$= 0.75			$\overline{\text{Assv}}$= 3.22		
RB M=1	1.088	7.4e-5	5596	4.494	1.5e-4	3426
RB M=30	1.090	9.9e-5	3838	4.402	2.5e-4	2837
RB M=50	1.090	7.6e-5	3549	4.527	2.5e-4	3459
CG-Geo	1.119	1.1e-4	3168	5.067	2.2e-4	4291

In all subsequent tests when we refer to Algorithm 5.3.1 we use the limited memory RBFGS, Algorithm 3.9.3 to solve the inner problem in Algorithm 5.3.1 with the geodesic (3.19) as retraction and the parallel translation in (3.22) as vector transport. We also set the maximal number of stored pairs to $M = 30$ and use the starting matrix N_0 in (5.37).

Test 2

The first test was devoted to finding the specifications and parameters for Algorithm 5.3.1 that yield the best performance. The second and third test is to compare Algorithm 5.3.1 with the methods in Section 5.4.2, namely LIFTPROJ and FMINCON. We start by applying all methods to our problem for the test matrices of type **uexample** as we know the optimal solution for this example and that the LICQ defined in Definition 4.6.9 is satisfied at this solution. The aim is to reduce the rank of the matrix A of **uexample** by one.

In LIFTPROJ we set φ to the smallest singular value of the point returned by Algorithm 5.3.1. Our results are presented in Table 5.3 where the number of iterations refers to the total number of iterations in the nonlinear CG method for Algorithm 5.3.1, for LIFTPROJ to the total number of performed iterations for the

Table 5.3: Results for test matrices of type **uexample**.

	Algorithm 5.3.1 (**RB**, $M = 30$)	**LIFTPROJ**	**FMINCON**
total number of iterations	9	1	5
μ_*	19.1	-	-
starting point	$N_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$X_0 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$	$x_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$
returned point X_*	$\begin{bmatrix} 2 & 0 \\ 0 & 3.1\text{e-}6 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 \\ 0 & -1.1\text{e-}10 \end{bmatrix}$

starting matrix that yields the smallest function value and for FMINCON it refers to the number of iterations in the trust-region algorithm. As we see from Table 5.3 all algorithms return a point that is close to the global solution. However, if we change the starting value of Algorithm 5.3.1 to $N_0 = (1, 0)^T$ this algorithm returns only a local solution that is

$$X_* = \begin{bmatrix} 2.2\text{e-}5 & 0 \\ 0 & 1 \end{bmatrix},$$

which gives support for your chosen starting value N_0 in (5.37) for Algorithm 5.3.1. This shows that we deal with a highly nonconvex objective function in (5.23). Therefore we can expect at most that the Algorithm 5.3.1 returns a point being close to a local minimum.

Test 3

In the third test we look at the performance of all three methods introduced in Section 5.4.2 for different test matrices of type **uhankel** and **urand**. We are particularly interested in how far the distance between the given matrix A and the returned point X_* is.

For type **urand** we generate square test matrices of dimension $n = 100, 150, 200$ and we apply our algorithms to find a solution of (5.2). As we can apply FMINCON only for a reduction of the rank of A by one we first devote ourselves to only $r = p-1$. For **urand** we set $s = 2n \geq (n-r)(p-r)+1$ to satisfy condition (5.31). Our results are summarized in Table 5.4 where we use the following abbreviations:

- t: computation time (in seconds) taken to run the particular algorithm.
- it: depending on the algorithm:
  - Algorithm 5.3.1: total number of iterations in the limited memory RBFGS algorithm.
  - LIFTPROJ: number of iterations in LIFTPROJ.

Table 5.4: Results for $r = p - 1$ and test matrices **urand**.

	Algorithm 5.3.1 (**RB**, $M = 30$)			**LIFTPROJ**			**FMINCON**		
n	**100**	**150**	**200**	**100**	**150**	**200**	**100**	**150**	**200**
t	5.0	12.5	23.6	10.3	2.6	6.5	327	10.2	692
it	38	104	116	102	9	12	298	6	164
$\frac{1}{2}\|A - X_*\|_F^2$	0.0005	0.0018	0.0041	11.4	51.8	49.1	47.1	1.32	665
Assv	4e-3	6e-3	9e-3	4e-3	6e-3	9e-3	4e-3	6e-3	9e-3
Xssv	4e-7	7e-6	2e-7	6e-10	3e-10	5e-13	2e-7	5e-6	8e-3
$\|X_*\|_F^2$	5.2e2	1.1e4	9.6e3	3.6e-4	1.3e-12	2.4e-15	4.3e2	1.1e4	9.2e3

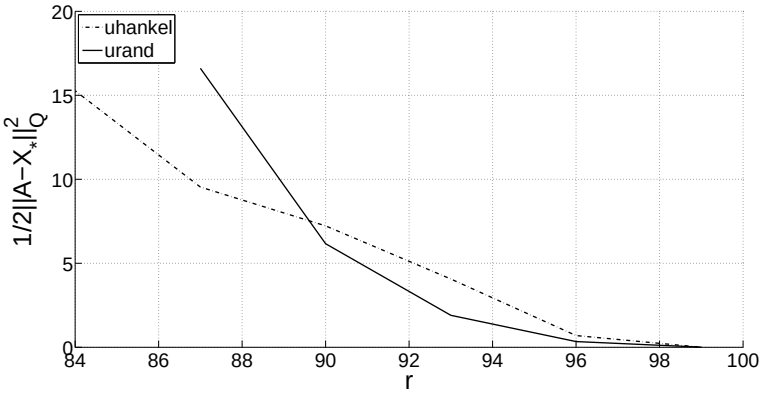

Figure 5.1: $\frac{1}{2}\|A - X_*\|_Q^2$ against the objective rank r.

– FMINCON: number of iterations in the trust-region algorithm.

- Assv: sum of the smallest $p - r$ singular values of A.
- Xssv: sum of the smallest $p - r$ singular values of X_*.

Note that we cannot guarantee in general that a nontrivial solution for the test matrices **urand** exists. To identify whether the returned points are close to zero we also report $\|X_*\|_F^2$ in Table 5.4.

From this table we clearly see that Algorithm 5.3.1 outperforms the other methods and yields the smallest function value. LIFTPROJ returns only the trivial solution $X_* = 0$ and FMINCON returns a nonzero solution but the function value at this solution is far larger than for the solution returned by Algorithm 5.3.1. For $n = 200$ FMINCON even fails to satisfy the stopping criterion (5.34) due to numerical limitations.

To test the performance of Algorithm 5.3.1 for a rank reduction of more than one we also generated test matrices of type **uhankel** and **urand** for $n = 100$, $p = 100$, $r = 1, 4, 7, \ldots, 16$. In Figure 5.1-5.3 we show the results where we plot in this order the distance $1/2\|A - X_*\|_Q^2$, the sum of the $p - r$ smallest singular values of X_*, and

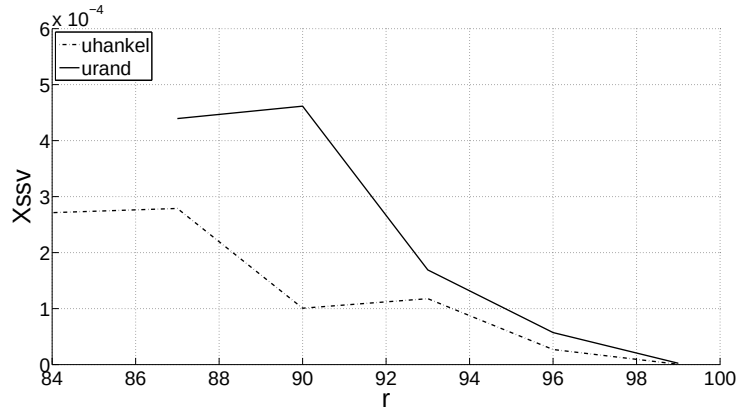

Figure 5.2: Xssv against the objective rank r.

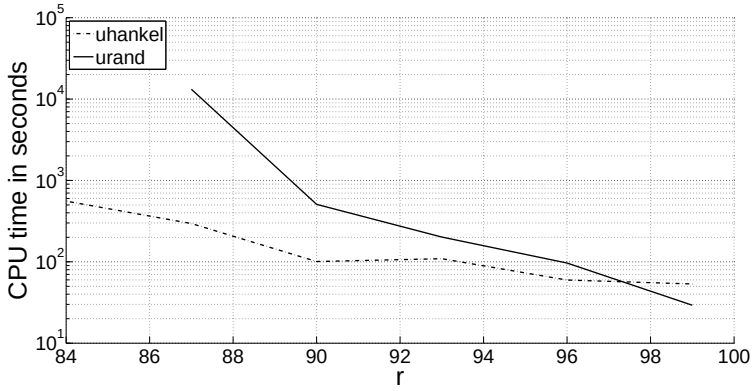

Figure 5.3: Computational time in seconds against the objective rank r.

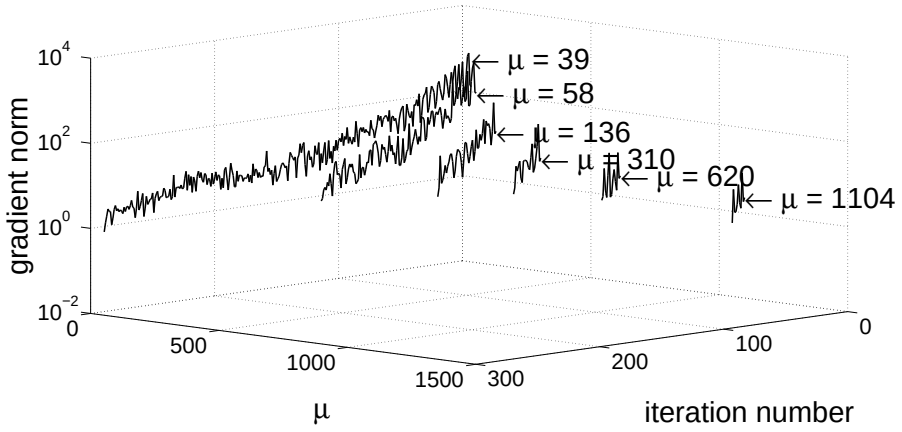

Figure 5.4: Norm of gradient against number of iterations in RBFGS algorithm.

the computational time in seconds against the objective rank r of X_*. We see in Figure 5.1 that the distance between A and X_* is dramatically increasing when r becomes smaller. Therefore for r small the smallest function value may be attained at the trivial solution $X_* = 0$. In Figure 5.3 we see for $r = 87$ that the algorithm has difficulties to find a solution and fails for $r = 84$. This could be an indicator that no nontrivial solution of this problem exists. Note that for this test we also applied LIFTPROJ but it always returned the trivial solution $X_* = 0$.

Test 4

The fourth test is only for illustrative reasons. We depict the behaviour of Algorithm 5.3.1 in Figure 5.4 for $n = 100$, $p = 100$ with test matrices of type **uhankel** and $r = p - 10$ where the $\|\text{grad}\, f_{\mu,\lambda}(N)\|_F$ with $f_{\mu,\lambda}(N)$ defined in (5.23) is plotted against the number of iterations in the limited memory RBFGS algorithm and all μ_k. We see that the gradient decreases but also oscillates heavily in the course of iterations in the RBFGS method pointing out again that the objective function is highly nonconvex. During our tests we observed that this behaviour of Algorithm 5.3.1 is more general for this problem size and by no means only specific to the test problem used to create Figure 5.4. We also see that most iterations in the limited memory RBFGS are taken for the initial penalty parameter μ_0 to satisfy the stopping criterion on line 4 of Algorithm 5.3.1. For the subsequent calls of the limited memory RBFGS far fewer iterations are required. In total the Algorithm 5.3.1 took 98 seconds to compute the final iterate X_* where 30% of the total time was spent on computing $F(N)$ in (5.24) and 25% on performing the vector transports in Algorithm 3.9.2, respectively.

Conclusion

In this numerical section we looked at different specifications for Algorithm 5.3.1, in particular which algorithm of Section 3.9 should be used to solve the inner problem on line 4 of Algorithm 5.3.1. According to our tests it turned out that the limited memory RBFGS method outperforms the nonlinear CG method if the number of stored pairs (y_i, s_i) is large enough. We continued to look at the overall performance of Algorithm 5.3.1 and showed that it yields in most cases better results than those methods that we compared it with. In particular in all tests the $p-r$ smallest singular values of the points returned by Algorithm 5.3.1 had significantly smaller values than those of the given matrix A. In contrast to the points returned by LIFTPROJ they were also notably different from the trivial solution $X_* = 0$ so that they could be of use for possible applications.

Certainly, in most tested examples the $p - r$ smallest singular values of X_* were greater than

```
tol = max(size(A)) * eps(norm(A))
```

that is the tolerance in the MATLAB function `rank`. This function returns the rank of a matrix by counting all singular values that are greater than `tol`. Therefore if we apply the function `rank` to our returned matrix X_* the rank will in most cases still be the rank of our given matrix A.

For a generated matrix A of type **uhankel** with dimension $n = 100$, $p = 100$ we tried by changing the parameters in Algorithm 5.3.1 what smallest possible size of the sum of the $p - r$ smallest singular of X_* we can achieve before we encounter numerical limitations. For $r = p-1$ the smallest singular value was about 2.6×10^{-14} but for $r = p - 5$ the smallest possible sum was only approximately 2.46×10^{-7}. In the latter case we cannot reduce $\|\text{grad } f_{\mu,\lambda}(N_k)\|_F$ any further than 0.0147 as for the next iterate N_{k+1} no feasible step length can be found in the Armjio-backtracking procedure due to numerical rounding errors. Besides having no convergence guarantee this shows another weakness of Algorithm 5.3.1. Nevertheless, as we have seen in our tests this algorithm gives us a tool to approximate the solution of the problem and it actually reduces the size of the smallest singular value despite the fact that we cannot guarantee convergence.

5.5 Conclusions

In this chapter we looked at the low rank linearly structured matrix nearness problem in the Q-norm. That is, given a rectangular matrix A we are trying to find a matrix $X \in \mathcal{L}$ defined in (1.6) that is closest to A in the Q-norm and of lower rank r.

We first looked at existing algorithms and we then investigated further the geometric approach by Schuermans et al. [121]. The idea of reformulating the nearest low rank problem for $X \in \mathbb{R}^{n\times p}$ as an optimization problem over the Grassmannian manifold goes back to an idea of Manton et al. in [93]. Schuermans et al. investigated then in [121] the problem when additional linear structure on the matrix X is imposed. However, the authors disregarded this approach in their tests as they only obtained the trivial solution $X_* = 0$ for $p - r > 1$ and $\mathcal{L}$ the set of Hankel matrices. We analysed this method further and pointed out in Section 5.2.3 why the authors obtained only the trivial solution in [121]. Furthermore, to be able to optimize the problem posed by Schuermans et al. we proposed to use the augmented Lagrangian method and developed all necessary tools that make the application of this method possible. We also discussed certain improvements regarding the efficiency of the algorithm and eventually stated our method in Algorithm 5.3.1. This algorithm is applicable for any linear structure and any Q-norm whereas the additional cost for Q dense is only moderate. Unfortunately, we cannot guarantee convergence of this algorithm. Instead we showed by means of two examples that the LICQ, which is a requirement to apply the existing convergence theory, may or may not be satisfied, depending on the problem.

In all our numericals tests we have seen that during the iterations this algorithm reduces the smallest singular values of the iterates X_k and does generally not return the trivial solution $X_* = 0$. We compared the results with other existing algorithms and observed that most often Algorithm 5.3.1 returned a point that was closest to the given matrix A. Therefore from the numerical tests we conclude that Algorithm 5.3.1 outperforms the other tested methods.

We also pointed out that the reduction of the smallest singular values of the given matrix A is restricted due to numerical rounding errors so that, depending on the problem, the final iterate X_* is often not of lower rank, only the smallest singular values are reduced.

In [122] the authors claim that it is not possible to obtain an algorithm that reduces the rank for any linear structured matrices. With Algorithm 5.3.1 we tried to contradict this statement and at least for a small reduction of the rank of the given matrix A we claim that we made a step forward to achieve this target.

Chapter 6

Conclusions and Future Work

Throughout this thesis we have looked at different structured matrix nearness problems that all come from real applications. In particular, we investigated algorithms that solve these nearness problems efficiently.

In Chapter 2 we investigated correlation matrices with k factor structure that mainly arise in the area of finance and we compared different algorithms to solve the corresponding nearness problem. Throughout this chapter we obtained more theoretical understanding of these factor-structured problems, particularly through explicit results for the one parameter and one factor cases. Furthermore, we arrived at the conclusion that the spectral projected gradient method is the method of choice for these kinds of problems as it guarantees convergence and performed best in most cases among all tested algorithms. We are convinced that this work is of great use for many scientists and financial analysts as the nearness problem can appear whenever a factor model is used, which is a well established tool in financial modelling. In particular, the algorithm proposed to solve the nearness problem provides a reliable tool for financial analysts to validate whether a factor model is appropriate for their analysis. Finally, as a result of our investigation in this chapter NAG, the Numerical Algorithms Group, included the algorithm that arose from our analysis in their library [103], convinced that this algorithm is of interest to their customers. The corresponding routine is available in their latest release and plans to parallelize this algorithm are being made.

The next structured matrix nearness problems that we looked at were the two two-sided optimization problems in Chapter 4 that arise in atomic chemistry. At the beginning we analyzed the first problem and proposed then an analytical optimal solution of it that is computed by Algorithm 4.4.1. We also showed by means of this algorithm and applying the active-set method that we can also find an optimal solution of the second problem. As the optimal solution of the first problem is generally not unique we established thereafter an optimization framework that allows to optimize an arbitrary smooth function subject to the constraints that describe

the structure of the optimal solutions. This mainly involved deriving all geometric objects of a new Riemannian manifold that are required to apply first-order optimization methods. The algorithm proposed is then an augmented Lagrangian-based algorithm whose inner problem is solved by applying the nonlinear CG method for Riemannian manifolds that we discussed in Section 3.9. We compared our algorithm with an augmented Lagrangian-based method whose inner problem is to minimize the augmented Lagrangian function over the Stiefel manifold. To incorporate all the constraints in this method we needed to use $p(p-1)/2$ more Lagrange multipliers than in our proposed algorithm.

From our numerical tests we concluded that our algorithm showed better performance if p is not too large. We also pointed out that the projection onto the tangent space of our Riemannian manifold is the bottleneck of our algorithm as this involved solving a linear system of order $p(p-1)/2$. Another weakness of this algorithm is that it can fail to find a stationary point of our augmented Lagrangian function. This is due to the structure of the manifold. We can only guarantee a reduction of the augmented Lagrangian function in the neighbourhood of our starting point. Certainly this is a crucial point of our algorithm that requires further improvement. We believe that one step forward to tackle this problem is to accept step sizes in the Armijo-backtracking procedure that yield points $Y \in \mathsf{St}(n,p)$ with $Y^T \Lambda Y$ diagonal but whose diagonal elements are not in increasing order. These points are surely not on our manifold $\mathcal{B}(n,p)$ but can be projected onto $\mathcal{B}(n,p)$ by multiplying an appropriate permutation matrix from the right. The latter operation may cause an increase in the objective function that could e.g. be tackled by a nonmonotone line search strategy. However, this needs further analysis. Further improvements of our algorithm could be achieved by looking at different methods to solve the inner problem in the augmented Lagrangian method. Candidates are for example the limited memory RBGFS algorithm introduced in Section 3.9 or the trust region algorithm [3, Algorithm 10].

Further investigation is also required to find an appropriate objective function whose minimization drives us to a point, at which the optimal solution of the first problem preserves the sign characteristics of the eigenvectors of N. We mentioned ideas in Section 4.7.1 but did not pursue them. Overall we conclude that we have provided optimal solutions to the problems given by Prof. Sax, University of Graz, and proposed an efficient algorithm that allows to select a specific solution from the set of optimal solutions by posing a new optimization problem. Therefore this research could provide helpful tools to obtain meaningful results in science, in particular, in atomic chemistry and will hopefully contribute to new finding in this area.

In the last chapter we considered the problem of finding a nearest low rank linearly structured matrix. We looked at different existing algorithms that solve these problems and mentioned pros and cons. Thereafter, we investigated the geometric approach and proposed to apply the augmented Lagrangian method to the resulting optimization problem. We discussed some efficiency aspects of the resulting algorithm and stated it then in Algorithm 5.3.1. By means of two examples we showed that the LICQ may but also may not be satisfied at the optimal points of the augmented Lagrangian function, showing that we cannot guarantee convergence of the algorithm in general by applying the existing theory. However, we observed in all our tests that our algorithm returns points that have small $p - r$ singular values and performed better than existing algorithms in terms of time spent. Therefore we are convinced that is algorithm this of use in many applications, in particular, as it is applicable to any linear structure and any symmetric positive definite weighting matrix Q in the Q-norm. Similar to Chapter 4 we could also investigate other algorithms like the trust region algorithm [3, Algorithm 10] to solve the inner problem in the augmented Lagrangian method to improve further the performance of our algorithm.

Overall we have seen that structured matrix nearness problems come from many applications and lead to interesting optimization problems in numerical analysis. In particular, we observed that different matrix structures can lead to very different optimization problems. We conclude that throughout these chapters we gained a deeper understanding of these problems and were thus able to propose algorithms that exploit the matrix structure and help to solve these nearness problems efficiently. We hope that these algorithms provide a tool to analysts and scientists to solve the structured matrix nearness problems discussed and thus, help to gain more knowledge in their areas of research.

List of Symbols

Sets

$\mathbb{N}$	set of natural numbers.
$\mathbb{Z}$	set of integer numbers.
$\mathbb{R}$	set of real numbers.
$\mathbb{R}^{n\times p}$	set of real matrices with dimension n-by-p.
$C^s(\mathbb{R}^{n\times p})$	set of functions $f : \mathbb{R}^{n\times p} \mapsto \mathbb{R}$ that are s times continuously differentiable.
$0_{n\times p}$	zero matrix in $\mathbb{R}^{n\times p}$.
I_n	identity matrix in $\mathbb{R}^{n\times n}$.
$\mathcal{S}^n$	set of symmetric matrices in $\mathbb{R}^{n\times n}$.
$\mathcal{S}_n^+$	set of symmetric positive semidefinite matrices in $\mathbb{R}^{n\times n}$.
$\mathcal{K}_n$	set of skew-symmetric matrices in $\mathbb{R}^{n\times n}$.
$\mathcal{S}_0^n$	set of symmetric matrix in $\mathbb{R}^{n\times n}$ with zero diagonal.
$\mathcal{M}$	a smooth manifold.
$T_x\mathcal{M}$	tangent space of $\mathcal{M}$ at x.
$N_x\mathcal{M}$	normal space of $\mathcal{M}$ at x.
$T\mathcal{M}$	tangent bundle of $\mathcal{M}$.
$\mathcal{F}(\mathcal{M})$	set all smooth real-valued functions defined on $\mathcal{M}$.
$\mathsf{O}(n)$	set of orthogonal matrices in $\mathbb{R}^{n\times n}$.
$\mathsf{St}(n,p)$	Stiefel manifold in $\mathbb{R}^{n\times p}$. See Section 3.8.1.
$\mathsf{Gr}(n,p)$	Grassmannian manifold $\mathsf{St}(n,p)/\mathsf{O}(p)$. See Section 3.8.2.

Operators

$\circ :$	$\mathbb{R}^{n\times p} \times \mathbb{R}^{n\times p} \mapsto \mathbb{R}^{n\times p}$	$H = A \circ B$ is the Hadamard product ($h_{ij} = a_{ij}b_{ij}$).
$\mathrm{diag} :$	$\mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$	$\mathrm{diag}(a)$ is the diagonal matrix with a on its diagonal.

diag :	$\mathbb{R}^{n\times n} \times \cdots \times \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{np\times np}$	$\operatorname{diag}(A_1, \ldots, A_p)$ is a block diagonal matrix with $A_1, \ldots, A_p$ on the diagonal.
diag :	$\mathbb{R}^{n\times n} \mapsto \mathbb{R}^n$	$\operatorname{diag}(A)$ is a vector with the diagonal of A.
offdiag :	$\mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n(n-1)}$	$\operatorname{offdiag}(A)$ is a vector with the offdiagonals of A stacked on top of each other starting with the most upper right.
dim :	{set of all spaces with a countable basis} $\mapsto$ $\mathbb{N} \cup \{\infty\}$	$\dim(\mathcal{A})$ is the dimension of $\mathcal{A}$.
trace :	$\mathbb{R}^{n\times n} \mapsto \mathbb{R}$	$\operatorname{trace}(A) = \sum_{i=1}^n a_{ii}$ the trace of A.
vec :	$\mathbb{R}^{n\times p} \mapsto \mathbb{R}^{np}$	$\operatorname{vec}(A)$ stacks the columns of A into a long vector
sym :	$\mathbb{R}^{n\times n} \mapsto \mathcal{S}_0^n$	$\operatorname{sym}(A) = \frac{A+A^T}{2}$ is the symmetric part of A.
skew :	$\mathbb{R}^{n\times n} \mapsto \mathcal{K}^n$	$\operatorname{skew}(A) = \frac{A-A^T}{2}$ is the skew-symmetric part of A.
$\otimes$:	$\mathbb{R}^{n\times p} \times \mathbb{R}^{k\times l} \mapsto \mathbb{R}^{nk\times pl}$	$A \otimes B$ is the Kronecker product of A and B. See Appendix A.1.
∇ :	$C^1(\mathbb{R}^{n\times p}) \mapsto \mathbb{R}^{n\times p}$	∇f is the classical derivative of f.
∇^2 :	$C^2(\mathbb{R}^n) \mapsto \mathbb{R}^{n\times n}$	$\nabla^2 f$ is the second classical derivative of f.
grad :	$\mathcal{F}(\mathcal{M}) \mapsto T\mathcal{M}$	$\operatorname{grad} f$ is the geometric gradient of f defined in (3.4).
$\operatorname{Hess} f(x)$:	$T_x\mathcal{M} \mapsto T_x\mathcal{M}$	$\operatorname{Hess} f(x)$ is the Riemannian Hessian of $f \in \mathcal{F}(\mathcal{M})$ at $x \in \mathcal{M}$ defined in (3.9).
$\mathcal{T}$:	$T\mathcal{M} \times T\mathcal{M} \mapsto T\mathcal{M}$	$\mathcal{T}_{\xi_x}(\eta_x)$ denotes the vector transport defined in Definition 3.7.10.
$\perp$:	$\mathbb{R}^{n\times p} \mapsto \mathbb{R}^{n\times(n-p)}$	$Y_\perp$ is a matrix that has orthonormal columns and satisfies $Y^T Y_\perp = 0$ for $Y \in \mathbb{R}^{n\times p}$.
qf :	$\mathbb{R}^{n\times p} \mapsto \mathbb{R}^{n\times p}$	$\operatorname{qf}(A)$ is the Q-factor of the QR decomposition of A for A of full rank.

Appendix A

Some Definitions

A.1 Kronecker Product

As we often make use of the notation of the Kronecker product, mainly in Chapter 5, for completeness we define it in this section and list some of its key properties. Let us start with the definition.

A.1.1 Definition

Definition A.1.1. The *Kronecker product* $\otimes : \mathbb{R}^{n\times m} \times \mathbb{R}^{r\times t} \mapsto \mathbb{R}^{nr\times mt}$ written as $A \otimes B$ for $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{r\times t}$, is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ \vdots & \vdots & & \vdots \\ a_{n1}B & a_{n2}B & \dots & a_{nm}B \end{bmatrix}.$$

A.1.2 Properties

Let $A \in \mathbb{R}^{n\times m}$, $B, C \in \mathbb{R}^{r\times t}$, and $s \in \mathbb{R}$. Then it holds that

- $A \otimes (B + C) = A \otimes B + A \otimes C$,
- $A(A + B) \otimes C = A \otimes C + B \otimes C$,
- $(sA) \otimes B = A \otimes (sB) = s(A \otimes B)$,
- $(A \otimes B) \otimes C = A \otimes (B \otimes C)$,
- $(A \otimes B)P = P(B \otimes A)$

where P is the permutation matrix defined by $\text{vec}(A^T) = P\text{vec}(A)$. For $m = r$, $D \in \mathbb{R}^{l\times k}$, $E \in \mathbb{R}^{l\times p}$, and $X \in \mathbb{R}^{k\times n}$ we also have

- $(A \otimes D)(B \otimes E) = (AB \otimes DE)$
- $(A^T \otimes D)\text{vec}(X) = \text{vec}(DXA)$.

For more details on Kronecker products, see Horn and Johnson [70, Chapter 4] or Lancaster and Tismenetsky [82, Chapter 12].

A.2 Fréchet Derivative

Definition A.2.1. Let $f : \mathbb{R}^{m\times p} \mapsto \mathbb{R}^{n\times q}$ be a matrix function. Then the Fréchet derivative at a point $X \in \mathbb{R}^{m\times p}$ is a linear mapping $L_f(X, \cdot) : \mathbb{R}^{m\times p} \mapsto \mathbb{R}^{n\times q}$ such that for all $E \in \mathbb{R}^{m\times p}$

$$f(X + E) - f(X) - L_f(X, E) = \text{o}(\|E\|)$$

where $\|\cdot\|$ is any matrix norm.

For a reference see [66, Section 3.1] where the Fréchet derivative is defined on functions from $\mathbb{C}^{n\times n} \mapsto \mathbb{C}^{n\times n}$. However this definition is also valid for real and rectangular matrices. Note if the Fréchet derivative exists it is equal to the directional derivative

$$\lim_{t\to 0} \frac{f(X + tE) - f(X)}{t}$$

[66, Section 3.2].

Bibliography

[1] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, 7(3):303–330, 2007.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220, 2004.

[3] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[4] P.-A. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren. A Grassmann-Rayleigh quotient iteration for computing invariant subspaces. *SIAM Rev.*, 44(1):57–73, 2002.

[5] H. Albrecher, S. Ladoucette, and W. Schoutens. A generic one-factor Lévy model for pricing synthetic CDOs. In M. C. Fu, R. A. Jarrow, J.-Y. J. Yen, and R. J. Elliott, editors, *Advances in Mathematical Finance*, Applied and Numerical Harmonic Analysis, pages 259–277. Birkhäuser, Boston, MA, USA, 2007.

[6] C. Alexander. Common correlation and calibrating the lognormal forward rate model. *Wilmott Magazine*, 2:68–78, 2003.

[7] S. I. Amari, T. P. Chen, and A. Cichocki. Nonholonomic orthogonal learning algorithms for blind source separation. *Neural Computation*, 12(6):1463–1484, 2000.

[8] L. Andersen, J. Sidenius, and S. Basu. All your hedges in one basket. *Risk*, 16:67–72, November 2003. |www.risk.net—.

[9] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, Jack J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney, and D. Sorensen. *LAPACK User's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 3rd edition, 1999.

[10] M. Aoki and P. Yue. On a priori error estimates of some identification methods. *IEEE Trans. Automat. Control*, 15(5):541–548, 1970.

[11] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[12] M. W. Berry, S. A. Pulatova, and G. W. Stewart. Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices. *ACM Transactions on Mathematical Software*, 31(2):252–269, 2005.

[13] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, USA, 1999.

[14] R. Bhatia. *Positive Definite Matrices*. Princeton University Press, 2007.

[15] D. A. Bini and P. Boito. A fast algorithm for approximate polynomial gcd based on structured matrix computations. *Numerical Methods for Structured Matrices and Applications*, pages 155–173, 2010.

[16] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10(4):1196–1211, 2000.

[17] E. G. Birgin, J. M. Martínez, and M. Raydan. Algorithm 813: SPG—Software for convex-constrained optimization. *ACM Trans. Math. Software*, 27(3):340–349, 2001.

[18] E. G. Birgin, J. M. Martínez, and M. Raydan. Spectral projected gradient methods. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization*, pages 3652–3659. Springer-Verlag, Berlin, 2nd edition, 2009.

[19] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[20] R. Borsdorf. Two-sided optimization problems with orthogonal constraints arising in chemisty. In preparation.

[21] R. Borsdorf. A Newton algorithm for the nearest correlation matrix. M.Sc. Thesis, The University of Manchester, Manchester, UK, September 2007. MIMS EPrint 2008.49, Manchester Institute for Mathematical Sciences, The University of Manchester, UK.

[22] R. Borsdorf. Matching hochaufgelöster 3D-Scandaten von teil-deformierten Objekten an CAD-Modelle: Verfahrens-, Laufzeit- und Präzisionsoptimierung. Diplomarbeit, Fak. f. Mathematik, Technische Universität Chemnitz-Zwickau, Chemnitz, FRG, 2009. German.

[23] R. Borsdorf and N. J. Higham. A preconditioned Newton algorithm for the nearest correlation matrix. *IMA J. Numer. Anal.*, 30(1):94–107, 2010.

[24] R. Borsdorf, N. J. Higham, and M. Raydan. Computing a nearest correlation matrix with factor structure. *SIAM J. Matrix Anal. Appl.*, 30(5):2603–2622, 2010.

[25] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersections of convex sets in Hilbert spaces. In *Advances in order restricted statistical inference: proceedings of the Symposium on Order Restricted Statistical Inference, Iowa City, Iowa, September 11-13, 1985*, pages 28–47. Springer-Verlag, 1986.

[26] D. Brigo and F. Mercurio. *Interest Rate Models—Theory and Practice. With Smile, Inflation and Credit*. Springer-Verlag, Berlin, 2nd edition, 2006.

[27] J. A. Cadzow. Signal enhancement—A composite property mapping algorithm. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(1):49–62, 1988.

[28] T. P. Cason, P.-A. Absil, and P. Van Dooren. Comparing two matrices by means of isometric projections. In S. P.; Chan R. H.; Olshevsky V.; Routray A. Van Dooren, P.; Bhattacharyya, editor, *Numerical Linear Algebra in Signals, Systems and Control*, volume 80 of *Lecture Notes in Electrical Engineering*, pages 77–93. Springer-Verlag, 2011.

[29] M. T. Chu, R. E. Funderlic, and R. J. Plemmons. Structured lower rank approximation. 1998. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.1806`.

[30] M. T. Chu, R. E. Funderlic, and R. J. Plemmons. Structured low rank approximation. *Linear Algebra Appl.*, 366:157–172, 2003.

[31] T. F. Coleman and Y. Li. On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. *Math. Programming*, 67(2):189–224, 1994.

[32] T. F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *j-SIOPT*, 6:418–445, 1996.

[33] M. Crampin and F. A. E. Pirani. *Applicable Differential Geometry*, volume 59. Cambridge University Press, 1986.

[34] G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation*, volume 15. Research Studies Press, 1988.

[35] M. Crouhy, D. Galai, and R. Mark. A comparative analysis of current credit risk models. *Journal of Banking & Finance*, 24:59–117, 2000.

[36] P. J. Davis. *Circulant matrices*. Chelsea Pub Co, 2nd edition, 1994.

[37] T. A. Davis. *UMFPACK Version 4.6 User Guide*. Dept. of Computer and Information Science and Engineering, Univ. of Florida, Gainesville, FL, 2002. `http://www.cise.ufl.edu/research/sparse/umfpack`.

[38] T. A. Davis. *CHOLMOD Version 1.0 User Guide*. Dept. of Computer and Information Science and Engineering, Univ. of Florida, Gainesville, FL, 2005. `http://www.cise.ufl.edu/research/sparse/cholmod`.

[39] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science and Technology*, 41(6):391–407, 1990.

[40] A. Del Bue, M. Stosic, M. Dodig, and J. Xavier. 2D-3D registration of deformable shapes with manifold projection. In *Proceedings of the 16th IEEE international conference on Image processing*, pages 1057–1060. IEEE Press, 2009.

[41] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, volume 16. Society for Industrial Mathematics, 1996.

[42] F. Deutsch. *Best Approximation in Inner Product Spaces*. Springer-Verlag, New York, 2001.

[43] L. Dieci and T. Eirola. On smooth decompositions of matrices. *SIAM J. Matrix Anal. Appl.*, 20(3):800–819, 1999.

[44] C. R. Dietrich and G. N. Newsam. Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, 1997.

[45] J. C. Dunn. Global and asymptotic convergence rate estimates for a class of projected gradient processes. *SIAM J. Control Optim.*, 19:368–400, 1981.

[46] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20:303–353, 1998.

[47] K. Fan and G. Pall. Imbedding conditions for Hermitian and normal matrices. *Canad. J. Math.*, 9:298–304, 1957.

[48] C. C. Finger. A methodology to stress correlations. *RiskMetrics Monitor*, Fourth Quarter:3–11, 1997.

[49] D. Gabay. Minimizing a differentiable function over a differential manifold. *J. Optimization Theory and Applications*, 37(2):177–219, 1982.

[50] K. R. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21:489–498, 1979.

[51] J. Garcia, S. Goossens, V. Masol, and W. Schoutens. Lévy base correlation. *Wilmott Journal*, 1(2):95–100, 2009.

[52] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for computer algebra*. Kluwer Academic Publishers, 1992.

[53] J. E. Gentle. *Elements of Computational Statistics*. Springer-Verlag, 2002.

[54] P. Glasserman and S. Suchintabandid. Correlation expansions for CDO pricing. *Journal of Banking & Finance*, 31:1375–1398, 2007.

[55] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.*, pages 413–432, 1973.

[56] G. H. Golub and C. F. Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, 1980.

[57] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[58] J. C. Gower and G. B. Dijksterhuis. *Procrustes Problems*. Oxford University Press, 2004.

[59] I. G. Graham, F. Y. Kuo, D. Nuyens, R. Scheichl, and I. H. Sloan. Quasi-Monte Carlo methods for elliptic pdes with random coefficients and applications. *Journal of Computational Physics*, 2011. DOI:10.1016/j.jcp.2011.01.023, appeared online.

[60] J. Gregory and J.-P. Laurent. In the core of correlation. *Risk*, 17(10):87–91, 2004.

[61] I. Grubišić and R. Pietersz. Efficient rank reduction of correlation matrices. *Linear Algebra Appl.*, 422:629–653, 2007.

[62] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[63] N. J. Higham. The Matrix Computation Toolbox. Available online from `http://www.ma.man.ac.uk/~higham/mctoolbox`.

[64] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2nd edition, 2002.

[65] N. J. Higham. Computing the nearest correlation matrix—A problem from finance. *IMA J. Numer. Anal.*, 22(3):329–343, 2002.

[66] N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[67] J. E. Hilliard and S. D. Jordan. Measuring risk in fixed payment securities: An empirical test of the structured full rank covariance matrix. *The Journal of Financial and Quantitative Analysis*, 26(3):345–362, 1991.

[68] J. B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, 1993.

[69] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1985.

[70] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[71] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer. Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme. *SIAM J. Matrix Anal. Appl.*, 32(1):115–135, 2011.

[72] J. Ivanic, G. J. Atchity, and K. Ruedenberg. Intrinsic local constituents of molecular electronic wave functions. I. Exact representation of the density matrix in terms of chemically deformed and oriented atomic minimal basis set orbitals. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, 120(1):281–294, 2008.

[73] P. Jäckel. Splitting the core. Working Paper, ABN AMRO, London, 2005.

[74] C. R. Johnson, editor. *Matrix Theory and Applications*, volume 40 of *Proceedings of Symposia in Applied Mathematics*. American Mathematical Society, 1990.

[75] C. R. Johnson, B. Kroschel, and H. Wolkowicz. An interior-point method for approximate positive semidefinite completions. *Computational optimization and applications*, 9(2):175–190, 1998.

[76] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 2nd edition, 2002.

[77] H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

[78] E. Kaltofen, Z. Yang, and L. Zhi. Structured low rank approximation of a Sylvester matrix. *Symbolic-Numeric Computation*, pages 69–83, 2007.

[79] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45(3):385–482, 2003.

[80] S. Kotz, W. L. Pearn, and D. W. Wichern. Eigenvalue-eigenvector analysis for a class of patterned correlation matrices with an application. *Statistics and Probability Letters*, 2:119–125, 1984.

[81] M. A. Laidacker. Another theorem relating Sylvester's matrix and the greatest common divisor. *Mathematics Magazine*, 42(3):126–128, 1969.

[82] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*. London, 2nd edition, 1985.

[83] D. N. Lawley and A. E. Maxwell. Factor analysis as a statistical method. *Journal of the Royal Statistical Society, Series D (The Statistician)*, 12(3):209–229, 1962.

[84] J. M. Lee. *Introduction to Smooth Manifolds*. Springer-Verlag, 2003.

[85] N. Li, P. Cheng, M. A. Sutton, and S. R. McNeill. Three-dimensional point cloud registration by matching surface features with relaxation labeling method. *Experimental Mechanics*, 45(1):71–82, 2005.

[86] Q. Li and D. Li. A projected semismooth Newton method for problems of calibrating least squares covariance matrix. *Operations Research Letters*, 39:103–108, 2011.

[87] Q. Li, H. Qi, and N. Xiu. Block relaxation and majorization methods for the nearest correlation matrix with factor structure. *Computational Optimization and Applications*, pages 1–23, 2010.

[88] F. Lillo and R. N. Mantegna. Spectral density of the correlation matrix of factor models: A random matrix theory approach. *Phy. Rev. E*, page 016219, 2005.

[89] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45(1):503–528, 1989.

[90] W. C. Lu, C. Z. Wang, M. W. Schmidt, L. Bytautas, K. M. Ho, and K. Ruedenberg. Molecule intrinsic minimal basis sets. I. Exact resolution of ab initio optimized molecular orbitals in terms of deformed atomic minimal-basis orbitals. *The Journal of Chemical Physics*, 120(6):2629–2637, 2004.

[91] D. G. Luenberger. *Optimization by Vector Space Methods*. Wiley, New York, 1969.

[92] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, Chichester, UK, 1988.

[93] J. H. Manton, R. Mahony, and Y. Hua. The geometry of weighted low-rank approximations. *IEEE Trans. Signal Processing*, 51(2):500–514, 2003.

[94] J. Mao. Optimal orthonormalization of the strapdown matrix by using singular value decomposition. *Computers Math. Applic.*, 12(3):353–362, 1986.

[95] I. Markovsky and S. Van Huffel. Overview of total least squares methods. *Signal Processing*, 87(10):2283–2302, 2007.

[96] I. Markovsky, S. Van Huffel, and R. Pintelon. Block-Toeplitz/Hankel structured total least squares. *SIAM J. Matrix Anal. Appl.*, 26(4):1083–1099, 2005.

[97] *Optimization Toolbox 4 User's Guide*. The MathWorks, Inc., Natick, MA, USA, 2009. Online version.

[98] A. E. Maxwell. Factor analysis. In Samuel Kotz, Campbell B. Read, N. Balakrishnan, and Brani Vidakovic, editors, *Encyclopedia of Statistical Sciences*, New York, 2006. Wiley. Electronic.

[99] M. Morini and N. Webber. An EZI method to reduce the rank of a correlation matrix in financial modelling. *Applied Mathematical Finance*, 13(4):309–331, 2006.

[100] *NAG Toolbox for MATLAB*. NAG Ltd., Oxford. Available online from `http://www.nag.co.uk/`.

[101] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comp.*, 35(151):773–782, 1980.

[102] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[103] Numerical Algorithms Group. *NAG Fortran Library Manual, Mark 23*. NAG, Oxford, UK, 2011.

[104] B. O'Neill. *Semi-Riemannian Geometry*, volume 103 of Pure and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1983.

[105] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–629, 1975.

[106] H. Park, L. Zhang, and J. B. Rosen. Low rank approximation of a Hankel matrix by structured total least norm. *BIT Numerical Mathematics*, 39(4):757–779, 1999.

[107] B. N. Parlett and G. Strang. Matrices with prescribed Ritz values. *Linear Algebra and its Applications*, 428(7):1725–1739, 2008.

[108] R. Pietersz and P. J. F. Groenen. Rank reduction of correlation matrices by majorization. *Quantitative Finance*, 4:649–662, 2004.

[109] C. Qi, K. A. Gallivan, and P.-A. Absil. Riemannian BFGS algorithm with applications. In *Recent Advances in Optimization and its Applications in Engineering*, pages 183–192. Springer-Verlag, 2010.

[110] H.-D. Qi and D. Sun. A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM J. Matrix Anal. Appl.*, 28(2):360–385, 2006.

[111] H.-D. Qi and D. Sun. Correlation stress testing for Value-at-Risk: An unconstrained convex optimization approach. *Computational Optimization and Applications*, 45:427–462, 2010.

[112] H.-D. Qi and D. Sun. An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem. *IMA J. Numer. Anal.*, 31(2):491–511, 2011.

[113] H.-D. Qi, Z. Xia, and G. Xing. An application of the nearest correlation matrix on web document classification. *Journal of Industrial and Management Optimization*, 3(4):701–713, 2007.

[114] C. Radhakrishna Rao. *Matrix Derivatives*. John Wiley & Sons, Inc., 2004.

[115] K. Rahbar and J. Reilly. Geometric optimization methods for blind source separation of signals. In *Second International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki, Finland, June 2000.

[116] MD. A. Rahman and K.-B. Yu. Total least squares approach for frequency estimation using linear prediction. *IEEE Trans. Acoust., Speech, Signal Processing*, 35(10):1440–1454, 1987.

[117] R. Reams. Hadamard inverses, square roots and products of almost semidefinite matrices. *Linear Algebra Appl.*, 288:35–43, 1999.

[118] J. B. Rosen, H. Park, and J. Glick. Total least norm formulation and solution for structured problems. *SIAM J. Matrix Anal. Appl.*, 17(1):110–126, 1996.

[119] S. N. Roy, B. G. Greenberg, and A. E. Sarhan. Evaluation of determinants, characteristic equations and their roots for a class of patterned matrices. *J. Roy. Statist. Soc. Ser. B*, 22(2):348–359, 1960.

[120] A. F. Sax. Localization of molecular orbitals on fragments. *Journal of Computational Chemistry*. To appear.

[121] M. Schuermans, P. Lemmerling, and S. Van Huffel. Structured weighted low rank approximation. *Numer. Linear Algebra Appl.*, 11(5-6):609–618, 2003.

[122] M. Schuermans, P. Lemmerling, and S. Van Huffel. Block-row Hankel weighted low rank approximation. *Linear Algebra Appl.*, 13(4):293–302, 2006.

[123] A. Shaji, S. Chandran, and D. Suter. Manifold optimisation for motion factorisation. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2009.

[124] S. T. Smith. Optimization techniques on Riemannian manifolds. In *Hamiltonian and gradient flows, algorithms and control*, volume 3 of *Fields Inst. Commun.*, pages 113–136. Amer. Math. Soc., Providence, RI, 1994.

[125] P. Sonneveld, J. J. I. M. van Kan, X. Huang, and C. W. Oosterlee. Nonnegative matrix factorization of a correlation matrix. *Linear Algebra Appl.*, 431:334–349, 2009.

[126] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *In 20th International Conference on Machine Learning*, pages 720–728. The AAAI Press, Menlo Park, California, 2003.

[127] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973.

[128] G. W. Stewart. Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numerische Mathematik*, 83(2):313–323, 1999.

[129] T. H. Szatrowski. Patterned covariances. In Samuel Kotz, Campbell B. Read, N. Balakrishnan, and Brani Vidakovic, editors, *Encyclopedia of Statistical Sciences*, New York, 2006. Wiley. Electronic. DOI:10.1002/0471667196.ess1927.pub2.

[130] A. Tchernitser and D. H. Rubisov. Robust estimation of historical volatility and correlations in risk management. *Quantitative Finance*, 9:43–54, 2009.

[131] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1:123, 1991.

[132] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optim.*, 7(1):1–25, 1997.

[133] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[134] S. Van Huffel, H. Park, and J. B. Rosen. Formulation and solution of structured total least norm problems for parameter estimation. *IEEE Trans. Signal Processing*, 44(10):2464–2474, 1996.

[135] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. Society for Industrial Mathematics, 1991.

[136] R. Vandebril, M. Van Barel, G. H. Golub, and N. Mastronardi. A bibliography on semiseparable matrices. *Calcolo*, 42:249–70, 2005.

[137] A. Vandendorpe, N.-D. Ho, S. Vanduffel, and P. Van Dooren. On the parameterization of the CreditRisk$^+$ model for estimating credit portfolio risk. *Insurance: Mathematics and Economics*, 42(2):736–745, 2008.

[138] B. Vandereycken. *Riemannian and multilevel optimization for rank-constrained matrix problems*. PhD thesis, Katholieke Universiteit Leuven, 2010.

[139] R. Varadhan and P. D. Gilbert. BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function. *Journal of Statistical Software*, 32(4):1–26, 2009.

[140] Q. J. Wang, D. E. Robertson, and F. H. S. Chiew. A Bayesian joint probability modeling approach for seasonal forecasting of streamflows at multiple sites. *Water Resources Research*, 45:W05407, 2009.

[141] T. Wansbeek. Eigenvalue-eigenvector analysis for a class of patterned correlation matrices with an application: A comment. *Statistics and Probability Letters*, 3:95–96, 1985.

[142] W. H. Yang and L. H. Zhang. Optimality conditions of the nonlinear programming on Riemannian manifolds. 2011. `http://www.optimization-online.org/DB_FILE/2011/08/3124.pdf`.

[143] C. J. Zarowski, X. Ma, and F. W. Fairman. QR-factorization method for computing the greatest common divisor of polynomials with inexact coefficients. *IEEE Trans. Signal Processing*, 48(11):3042–3051, 2000.

[144] Z. Zhang and L. Wu. Optimal low-rank approximation to a correlation matrix. *Linear Algebra Appl.*, 364:161–187, 2003.