Theory and algorithms for matrix problems with positive semidefinite constraints

Strabić, Nataša

2016

MIMS EPrint: **2016.23**

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

THEORY AND ALGORITHMS FOR MATRIX PROBLEMS WITH POSITIVE SEMIDEFINITE CONSTRAINTS

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

2016

Nataša Strabić
School of Mathematics

Contents

List of Tables

List of Figures

The University of Manchester

Nataša Strabić
Doctor of Philosophy
Theory and algorithms for matrix problems with positive semidefinite constraints
April 4, 2016

This thesis presents new theoretical results and algorithms for two matrix problems with positive semidefinite constraints: it adds to the well-established nearest correlation matrix problem, and introduces a class of semidefinite Lagrangian subspaces.

First, we propose shrinking, a method for restoring positive semidefiniteness of an indefinite matrix M_0 that computes the optimal parameter α_* in a convex combination of M_0 and a chosen positive semidefinite target matrix. We describe three algorithms for computing α_*, and then focus on the case of keeping fixed a positive semidefinite leading principal submatrix of an indefinite approximation of a correlation matrix, showing how the structure can be exploited to reduce the cost of two algorithms. We describe how weights can be used to construct a natural choice of the target matrix and that they can be incorporated without any change to computational methods, which is in contrast to the nearest correlation matrix problem. Numerical experiments show that shrinking can be at least an order of magnitude faster than computing the nearest correlation matrix and so is preferable in time-critical applications.

Second, we focus on estimating the distance in the Frobenius norm of a symmetric matrix A to its nearest correlation matrix $\mathrm{ncm}(A)$ without first computing the latter. The goal is to enable a user to identify an invalid correlation matrix relatively cheaply and to decide whether to revisit its construction or to compute a replacement. We present a few currently available lower and upper bounds for $\mathrm{d_{corr}}(A) = \|A - \mathrm{ncm}(A)\|_F$ and derive several new upper bounds, discuss the computational cost of all the bounds, and test their accuracy on a collection of invalid correlation matrices. The experiments show that several of our bounds are well suited to gauging the correct order of magnitude of $\mathrm{d_{corr}}(A)$, which is perfectly satisfactory for practical applications.

Third, we show how Anderson acceleration can be used to speed up the convergence of the alternating projections method for computing the nearest correlation matrix, and that the acceleration remains effective when it is applied to the variants of the nearest correlation matrix problem in which specified elements are fixed or a lower bound is imposed on the smallest eigenvalue. This is particularly significant for the nearest correlation matrix problem with fixed elements because no Newton method with guaranteed convergence is available for it. Moreover, alternating projections is a general method for finding a point in the intersection of several sets and this appears to be the first demonstration that these methods can benefit from Anderson acceleration.

Finally, we introduce semidefinite Lagrangian subspaces, describe their connection to the unique positive semidefinite solution of an algebraic Riccati equation, and show that these subspaces can be represented by a subset $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ and a Hermitian matrix $X \in \mathbb{C}^{n\times n}$ that is a generalization of a quasidefinite matrix. We further obtain a semidefiniteness-preserving version of an optimization algorithm introduced by Mehrmann and Poloni [*SIAM J. Matrix Anal. Appl.*, 33(2012), pp. 780–805] to compute a pair $(\mathcal{I}_{\mathrm{opt}}, X_{\mathrm{opt}})$ with $M = \max_{i,j}|(X_{\mathrm{opt}})_{ij}|$ as small as possible, which improves numerical stability in several contexts.

Declaration

No portion of the work referred to in the 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://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), 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

- The material in Chapter 3 is based on: N. J. Higham, N. Strabić, and V. Šego. Restoring Definiteness via Shrinking, with an Application to Correlation Matrices with a Fixed Block. MIMS EPrint 2014.54, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, November 2014. Revised June 2015. 20 pp. Accepted for publication in *SIAM Review*.

- The material in Chapter 4 is based on: N. J. Higham and N. Strabić. Bounds for the Distance to the Nearest Correlation Matrix. MIMS EPrint 2015.112, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, December 2015. 15 pp. Submitted to *SIAM J. Matrix Anal. Appl.*

- The material in Chapter 5 is based on: N. J. Higham and N. Strabić. Anderson Acceleration of the Alternating Projections Method for Computing the Nearest Correlation Matrix. *Numer. Algorithms*. 22 pp. Advance Access published December 21, 2015. DOI 10.1007/s11075-015-0078-3.

- The material in Chapter 6 is based on: F. Poloni and N. Strabić. Principal Pivot Transforms of Quasidefinite Matrices and Semidefinite Lagrangian Subspaces. MIMS EPrint 2015.92, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, October 2015. 26 pp. Accepted for publication in *Electron. J. Linear Algebra*.

Acknowledgements

In the course of the amazing years spent working on this thesis I have been able to rely on the help and support of many colleagues, friends, and family, and it gives me great pleasure to record my thanks to them.

I am very grateful to my supervisor Nicholas Higham for all his kind guidance, helpful comments and neverending patience, the fact that he was always approachable and available to answer questions, and for his good–natured sense of humour. His attention to detail, the ability to spot an interesting research direction in almost anything, willingness to always share his experience and expertise, and great support of young researchers never failed to inspire. I also must thank him for buying Toothless (the laptop) for the number crunching part of this work, and for supporting my attendance at many conferences. I also gratefully acknowledge the financial support from the School of Mathematics.

It has been a great pleasure to work with Craig Lucas on presenting the methods developed in this thesis to NAG, and with Chris Mower who implemented shrinking for the NAG library. I am grateful to Françoise Tisseur, Sven Hammarling, and Nicholas Gould for their kind interest in my work through the years and for all our discussions. I also thank Daniel Kressner for his many insightful comments and for inviting me to Lausanne.

A huge thank you goes to Federico Poloni, who patiently answered all my questions about symmetric PPTs and made it possible that a more–than–a–hundred–emails–worth of long–distance work became a paper. Moreover, I am very grateful for his kind invitation for a research visit to Pisa, that included a cloak and dagger sightseeing tour of Scuola Normale and was one of the most fun weeks of my PhD.

Carolyn Dean turned my casual comment about students needing a Writing Mathematics course into us organizing a full-fledged workshop; I very much enjoyed working

with her on this project and I am very grateful for her enthusiastic support and kind guidance.

I greatly appreciate the continuous support of Vjeran Hari, whose enthusiasm for matrix analysis was very much contagious and who helped me make my first research steps. Saša Singer was always there for my Numerical Analysis questions and he kindly provided encouragement when I needed it.

I most gratefully acknowledge the many hours Alexandre Borovik invested in talking to me during difficult times, his kind support, and much welcomed appreciation of my teaching work.

It has truly been a privilege to be a part of the NLA group in Manchester and get to know so many great people. All the members of the group through the years have my deepest gratitude for always being there to offer helpful advice, kind support, emergency coffee rations and, last but certainly not least, for being fun company. A special thank you goes to Fran, Jen, and Bahar for adding some very exotic pebbles from their travels to my rock collection. Vanni gets a thank you too, even though he forgot the Greek rock in the Kalamata hotel.

I thank Bon Jovi, Metallica, Nightwish (the original setup; guys, please, can we get back to that sound?), Within Temptation, Elysion, Sirenia, Xandria, Halestorm, Epica, Edenbridge, and others, for drowning out the world when I need them to, and for their assistence with making the MATLAB code work.

I am extremely grateful for the unshakable support of my family, for their unconditional love and acceptance, and for the hundreds of photos shared to keep me in the loop with the things I had missed (including some very tasty looking food). Martina, Suzana, and Nina are both my safety net and the reason why I can keep climbing higher, and Ines set me on my way. Unfailingly, through many many years, Lana has always been there to face all the problems with me, to listen, to encourage, and to share my obsession with characters from “Supernatural”.

Finally, Vedran always made sure that there was coffee waiting for me in the morning, and he had found a thousand different ways to make me laugh. Without his belief in me and his unwavering support none of this would have been possible.

Chapter 1

Introduction

Structures are the weapons of the mathematician.

—Nicolas Bourbaki

One maxim of numerical linear algebra is to exploit the structure of matrices whenever it appears. The key structure considered in this thesis is positive (semi)definiteness: as it leads to remarkable gains in matrix computations it is one of the most desirable structures a matrix can possess. For example, the triangular factor in the Schur decomposition of a positive semidefinite matrix is a nonnegative diagonal matrix and the complete set of eigenvectors is orthonormal; solving a positive definite linear system using the Cholesky factorization is not only numerically stable but also achieved in half the time and half the space of Gaussian elimination applied to a general system [30, Chap. 2.7.1]; and definite generalized eigenvalue problems are equivalent to standard symmetric ones, see, for example, [28]. It is therefore pleasing that positive semidefinite matrices frequently appear in applications.

However, in several very different practical contexts a positive semidefinite matrix is expected but an indefinite one is obtained, and a lot of effort has been invested into resolving this issue. Modified Cholesky methods of Gill, Murray, and Wright [42, Sec. 4.4.2.2], Eskow and Schnabel [102], [103], and Cheng and Higham [24] are used to deal with indefinite Hessians encountered in nonlinear optimization, and a popular way to correct an indefinite approximation to a *correlation matrix* (a real symmetric positive semidefinite matrix with unit diagonal) is to replace it by the nearest correlation matrix in the Frobenius norm. Chapters 3, 4, and 5 of this thesis are concerned with definiteness in this setting and section 1.1 provides a detailed overview of the nearest correlation matrix problem.

The continuous development of methods that both make use of the structure of the input matrix and ensure the structure of the resulting matrix is of great interest

in many applications from statistics, physics, and engineering, as structure–preserving algorithms are usually faster and more accurate, have reduced storage requirements and computational cost, and, perhaps most importantly, they are expected to produce more meaningful solutions. Namely, structure often comes from the physical properties of the problem and it might get destroyed by rounding or truncation errors, leading to a meaningless result. A beautiful illustration can be found in control theory, where solving a stable Lyapunov equation $A^*X + XA = -B^*B$ by the general Bartels–Stewart method [9] does not guarantee that the solution X will be positive semidefinite in finite arithmetic, but Hammarling's method [49], [50], [106] solves the equation directly for the Cholesky factor R of X, thus guaranteeing that X is semidefinite by construction. Moreover, in some applications (cf. the references in [106]) the Cholesky factor R is in fact more useful than X and overall, since for the 2-norm and condition numbers we have $\kappa(X) = \kappa(R)^2$, X may be significantly more ill-conditioned to work with.

Lagrangian subspaces are an essential structure in control theory applications, especially in the context of algebraic Riccati equations. In Chapter 6 we define Lagrangian *semidefinite* subspaces and develop a structure–preserving algorithm that computes their optimal representation (basis) by working directly with the factored forms of certain matrices; the motivation for this work is similar to the above. Relevant preliminary results for this topic are described in section 1.2.

The next two sections give an introduction to the two central problems on which the subsequent chapters of this thesis are built: the nearest correlation matrix problem, and the representation of Lagrangian subspaces. The remaining sections in this chapter present a detailed overview of the thesis and main research contributions, provide links to the developed software, and describe the test matrix collections that were used in the numerical experiments sections throughout the thesis.

1.1 The nearest correlation matrix problem

In many applications involving statistical modelling the first step in the analysis is to compute the sample correlation matrix—a real symmetric positive semidefinite matrix with unit diagonal—from empirical or experimental data [104, p. 25]. Indefinite

approximations of correlation matrices appear in practice for a variety of reasons and we next present a few key examples.

- *Robust estimation.*

  The sensitivity of sample correlation matrices to outliers in the data has led to the development of robust estimators. Devlin, Gnanadesikan, and Kettenring [31] propose several possibilities and note that some methods that compute the estimator in an elementwise manner can produce matrices with negative eigenvalues.

- *Missing data.*

  The pairwise deletion method (see, for example, [76, Sec. 2.2]) is a very common way of calculating the correlation coefficient between a pair of vectors with missing values. It uses only the components available in both vectors and results in an approximate sample correlation matrix that is symmetric with unit diagonal and off-diagonal elements in $[-1, 1]$, but there is no guarantee that it is positive semidefinite.

- *Expert judgement.*

  Some applications require assigning different values to certain elements of a valid correlation matrix. For example, stress testing in finance [39], [93] is used to explore the effect on a portfolio of pushing risk parameters toward extreme levels. This is achieved by replacing specific elements of a valid correlation matrix by new values, which may result in the new matrix becoming indefinite.

- *Aggregation.*

  Aggregation methods used in large-scale resource assessment, for example in geology [16] or finance [4] combine reliable estimates of correlation matrices for each group, say a geographical region or a market portfolio, into a global correlation matrix. The combination is achieved either by embedding small, "within group" correlation matrices as diagonal blocks into a crudely estimated global correlation matrix, or by constructing a block-diagonal matrix from the individual group correlation matrices and filling out the off-diagonal blocks by assigning the "between group" correlation coefficients according to expert judgement.

Again, there is no guarantee that the newly constructed matrix is in fact positive semidefinite.

To ensure the validity of the subsequent analysis the indefinite approximation needs to be replaced by a valid correlation matrix. This restoration of definiteness is needed in a very wide variety of applications, of which some recent examples include modelling public health [29] and dietary intakes [120], determination of insurance premiums for crops [41], simulation of wireless links in vehicular networks [124], analysis of wind farms [40], reservoir modelling [81], reconstructing 20th century sea levels [96], genetic evaluations for thoroughbred horse breeding [108], probabilistic forecasts of streamflows [123], prediction of electricity peak-demand during the winter season in England and Wales [86], analysis of carbon dioxide storage resources in the US [16], and a modelling framework that combines different sources of variability in biological systems [110].

The matrices arising in these applications are generally dense, with the order ranging from the tens to the tens of thousands. A simple approach for repairing an *invalid correlation matrix*, by which we mean a real symmetric indefinite matrix with unit diagonal, is to compute the nearest positive semidefinite matrix in the Frobenius norm, which amounts to shifting all the negative eigenvalues to zero while keeping the eigenvectors fixed (see Lemma 4.2.1), and then to diagonally scale it to restore the unit diagonal. However, this approach may change the matrix more than necessary—we analyze this in Chapter 4—and so a standard way to correct an invalid correlation matrix A is to replace it by the *nearest correlation matrix* in the Frobenius norm, that is, by the solution of the problem

$$\min\{\, \|A - X\|_F : X \text{ is a correlation matrix} \,\}, \tag{1.1}$$

where $\|A\|_F^2 = \sum_{i,j} a_{ij}^2$. Due to the convexity properties there is a unique global minimizer which we denote by $\mathrm{ncm}(A)$.

The first method for solving (1.1) with guaranteed convergence was the alternating projections method proposed by Higham [56], which iteratively projects onto the set of matrices with unit diagonal and the convex cone of symmetric positive semidefinite matrices, see Algorithm 5.3.1. Since each iteration of the alternating projections method requires a full eigenvalue decomposition and the rate of convergence is at best

linear, the method can potentially be very slow. A quadratically convergent Newton method was subsequently developed by Qi and Sun [92], who work on the dual of (1.1) and use the theory of strongly semismooth matrix functions to prove global convergence. Significant speed up to the original Newton method is due to the refinements introduced by Borsdorf and Higham [18]. Still, the alternating projections method remains widely used in applications and in Chapter 5 we show that its convergence can be accelerated significantly using Anderson acceleration.

The following variants of the problem (1.1) are common in practice and will also be addressed in the work presented in this thesis.

- *Fixed elements.*

  Missing data, correlation stress testing, risk aggregation, and large-scale resource assessment were listed as examples of applications where an indefinite approximation of the correlation matrix might occur but they also naturally lead to the fixed elements requirement, as we now explain.

  In case of missing values, the data from k observations of n random variables is collected in a $k \times n$ matrix X. We may assume that the missing entries do not occur in the first n_1 columns because we can permute them if necessary. The pairwise deletion method [76, Sec. 2.2] results in a unit diagonal symmetric matrix C of the form

$$C = \begin{array}{c} \\ n_1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} n_1 & n_2 \end{array} \\ \begin{bmatrix} A & Y \\ Y^T & B \end{bmatrix} \end{array} \in \mathbb{R}^{n \times n}.$$

  The leading block A is positive semidefinite (hence, a correlation matrix) because it is constructed from the columns of X that have no missing values but if C is indefinite it needs to be replaced with a valid correlation matrix. Since the correlations in A are considered exact we wish to replace C by a valid correlation matrix with this block unchanged.

  In a variant of correlation stress testing [39], [93] the assets are split into two groups. Their correlation matrix can then be block-partitioned as

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{12}^T & C_{22} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

where the inter-group correlations correspond to the diagonal blocks C_{11} and C_{22}, and the off-diagonal block C_{12} carries the cross-group correlations. A stress test replaces the block C_{22} with a new correlation matrix $\widehat{C}_{22}$. If this results in an indefinite modified matrix C we can again compute its replacement correlation matrix, but the C_{11} block should remain unchanged since the first group of assets was not affected.

In risk aggregation [4], [64] and large-scale resource assessment [16] we have a generalization of the above constraint, where in a global block-correlation matrix more diagonal blocks get replaced by new correlation matrices. If this results in the indefiniteness of the global matrix we must restore it while keeping the new diagonal blocks unchanged.

Hence, the nearest correlation matrix problem to be solved is

$$\begin{aligned} \min\{\, \|A - X\|_F : X \text{ is a correlation matrix,} \\ x_{ij} = a_{ij} \text{ for } (i,j) \in \mathcal{N}\,\}, \end{aligned} \tag{1.2}$$

where $\mathcal{N}$ denotes the index set of the fixed off-diagonal elements. Clearly, for $(i,j) \in \mathcal{N}$ we have $(j,i) \in \mathcal{N}$. Unlike (1.1), this variant of the nearest correlation matrix problem might not have a solution: $\mathcal{N}$ must be chosen such that there exists a correlation matrix with the prescribed fixed elements. This need not be true for every $\mathcal{N}$, as the following simple example shows. Take

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{1.3}$$

and $\mathcal{N} = \{(2,3),(3,2),(2,4),(4,2),(3,4),(4,3)\}$. We cannot replace A with a valid correlation matrix while keeping the elements prescribed by $\mathcal{N}$ fixed, since they correspond to the trailing 3×3 block of A, which is indefinite.

- *Bound on the smallest eigenvalue.*

  Singularity of a correlation matrix is an issue in applications where the inverse of the matrix is needed, for example in regression [51], [90] or multivariate data analysis [95].

The sample correlation matrix in the so–called "small k, large n" case is singular, and this case is common in practice. In finance applications for example there can be over a 1 000 stocks to choose from, but there is rarely more than 10 years of monthly data available per stock, while in biological studies, the budgetary and time constraints might dictate that there is only a small number of samples available for analysis, but the data set under observation, such as genomic data, is very large.

Moreover, for an invalid correlation matrix A with t nonpositive eigenvalues, from [56, Thm. 2.5] it follows that the nearest correlation matrix to A will have at least t zero eigenvalues, which means that the alternating projections method and the Newton method will compute a singular matrix.

Nonsingularity requirement leads to formulating the nearest correlation matrix problem as

$$\begin{aligned} \min\{\, \|A - X\|_F : X \text{ is a correlation matrix,} \\ \lambda_{\min}(X) \geq \delta \,\}, \end{aligned} \tag{1.4}$$

where δ is a given positive tolerance and $\lambda_{\min}(X)$ denotes the smallest eigenvalue of the symmetric matrix X. Since for a correlation matrix $\operatorname{trace}(X) = \sum_i \lambda_i(X) = n$, it follows that we must take $\delta \leq 1$. As the original nearest correlation matrix problem, (1.4) always has a unique solution [92, p. 372].

- *Weights.*

  In practice, different elements of an invalid correlation matrix may be known to different levels of accuracy or confidence [12]. Moreover, larger or more important lines of the model might need to be given more significance in the analysis. This can be reflected by introducing a *weighted* Frobenius norm to problem (1.1). The first choice is the W-norm, $\|A\|_W = \|W^{1/2} A W^{1/2}\|_F$, where W is a symmetric positive definite matrix and $W^{1/2}$ its unique positive definite square root (cf. [58, Cor. 1.30]), and the second is the H-norm, $\|A\|_H = \|H \circ A\|_F$, where H is a symmetric elementwise nonnegative matrix and $\circ$ denotes the Hadamard (elementwise) matrix product.

  For the H-norm, a large value of h_{ij} should force x_{ij} to remain close to a_{ij}, which corresponds to the notion that a_{ij} is know accurately and hence we do not wish

it to change much, while a small value of h_{ij} is assigned to the values of a_{ij} which are known relatively inaccurately or it is not important that they stay close to the original values. The W-norm does not allow for individual weighting but it is easier to work with. In practice, a diagonal W is the usual choice.

For both weighted Frobenius norms the solution to the nearest correlation matrix problem is unique [56, p. 330].

In terms of the solution methods for the above problem variants, the alternating projections method of Higham [56] was initially derived for the W-norm, of which the Frobenius norm is a special case. It can easily incorporate the fixed elements constraint to solve (1.2), which was analyzed by Lucas [76] and Borsdorf [17, Chap. 7], as well as solve the positive definite nearest correlation matrix problem (1.4); we provide the details in Chapter 5. As discussed in [56, p. 337], alternating projections method cannot be used in the H-norm case since a closed formula for the projection of a matrix to the positive semidefinite cone in this norm is not known and we cannot efficiently compute it.

Qi and Sun show that their Newton method [92] for the original nearest correlation matrix problem can easily use the W-norm [92, Sec. 4.1] and also compute the positive definite nearest correlation matrix [92, Sec. 4.2]. For the H-norm problem variant they derive a new Newton method in [94] and also note that it could be used to fix elements, but no details are provided for the latter. Moreover, the documentation for the NAG [83] code `g02aj/nag_nearest_correlation_h_weight` which solves the H-weighted nearest correlation matrix problem notes that the algorithm might not converge if the weights vary by several orders of magnitude. Hence, for the fixed elements case, only the alternating projections method is guaranteed to compute the solution, if it indeed exists.

1.2 Lagrangian subspaces

An n-dimensional subspace $\mathcal{U}$ of $\mathbb{C}^{2n}$ is called *Lagrangian* if $u^* J_n v = 0$ for every $u, v \in \mathcal{U}$, where

$$J_n = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}.$$

A matrix $U \in \mathbb{C}^{2n\times n}$ of full column rank is a basis for a Lagrangian subspace if and only if $U^* J_n U = 0$. For $U, V \in \mathbb{C}^{2n\times n}$ of full column rank we write $U \sim V$ if $U = VM$ for a square invertible $M \in \mathbb{C}^{n\times n}$. Note that this implies that U and V have the same column space, i.e. $\mathrm{Im}(U) = \mathrm{Im}(V)$.

Lagrangian subspaces are an essential structure in control theory applications (see, for example, [1], [38], [69], [78]). In computational practice, a subspace $\mathcal{U}$ is typically represented through a matrix U whose columns span it. A key quantity is its *condition number* $\kappa(U) = \sigma_{\max}(U)/\sigma_{\min}(U)$, where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular values, respectively. The sensitivity of $\mathcal{U} = \mathrm{Im}(U)$ as a function of U depends on $\kappa(U)$ [107, p. 154], as well as the numerical stability properties of several linear algebra operations associated to it, for instance, QR factorization [55, Chap. 19] and least-squares problems [55, Chap. 20]. Hence, in most applications the natural choice for a basis is a matrix U with orthonormal columns, which ensures $\kappa(U) = 1$. However, if a matrix U is partitioned as

$$U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \in \mathbb{C}^{2n\times n}, \quad U_1, U_2 \in \mathbb{C}^{n\times n},$$

then it spans a Lagrangian subspace if and only if $U_1^* U_2 = U_2^* U_1$, which is a property very difficult to preserve in finite arithmetic. If the matrix U_1 is invertible, we can write

$$U = \begin{bmatrix} I_n \\ X \end{bmatrix} U_1, \quad X = U_2 U_1^{-1}, \tag{1.5}$$

and hence obtain a different matrix $V = \left[\begin{smallmatrix} I_n \\ X \end{smallmatrix}\right]$ whose columns span the same subspace.

Matrices of the form

$$\mathcal{G}(X) = \begin{bmatrix} I_n \\ X \end{bmatrix}, \quad X \in \mathbb{C}^{n\times n} \tag{1.6}$$

are called *graph matrices*, since their form resembles the definition of the graph of a function as the set of pairs $(x, f(x))$, or *Riccati matrices*, since they are related to the algebraic Riccati equations [69]. Additional details are provided in Chapter 6, as they are the motivation for studying the special Lagrangian subspaces introduced there. We use the name Riccati matrix for $\mathcal{G}(X)$, since it is less likely to induce confusion with graphs as mathematical objects with nodes and edges. From (1.5), since U_1 is nonsingular it follows that

$$U \sim \mathcal{G}(U_2 U_1^{-1}), \tag{1.7}$$

and it is easy to see from the definition that $\operatorname{Im}\mathcal{G}(X)$ is Lagrangian if and only if $X = X^*$, a condition which is trivial to ensure in numerical computation. Hence, if the object of interest is the Lagrangian subspace $\operatorname{Im} U$, we can associate it with the Hermitian matrix X and use only this matrix to store and work on the subspace. The potential difficulties with this approach come from computing $X = U_2U_1^{-1}$ because U_1 could be ill-conditioned or even singular.

Mehrmann and Poloni [79] consider a slightly more general form instead. For each subset $\mathcal{I} \subseteq \{1, 2, \dots, n\}$, the *symplectic swap matrix* associated with $\mathcal{I}$ is defined as

$$\Pi_{\mathcal{I}} = \begin{bmatrix} I_n - D & D \\ -D & I_n - D \end{bmatrix} \in \mathbb{R}^{2n\times 2n}, \tag{1.8}$$

where D is the diagonal matrix such that

$$D_{ii} = \begin{cases} 1, & i \in \mathcal{I}, \\ 0, & i \notin \mathcal{I}. \end{cases}$$

The matrices $\Pi_{\mathcal{I}}$ are symplectic ($\Pi_{\mathcal{I}}^T J_n \Pi_{\mathcal{I}} = J_n$) and orthogonal ($\Pi_{\mathcal{I}}^T \Pi_{\mathcal{I}} = I_{2n}$), and the multiplication with $\Pi_{\mathcal{I}}$ permutes (up to a sign change) the elements of a $2n$-length vector, with the limitation that the ith entry can only be exchanged with the $(n+i)$th, for each $i = 1, 2, \dots, n$.

Example 1.2.1. When $n = 2$, the four symplectic swap matrices are

$$\Pi_{\emptyset} = I_4, \qquad \Pi_{\{1\}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \Pi_{\{2\}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \qquad \Pi_{\{1,2\}} = J_2.$$

Given a full column rank matrix $U \in \mathbb{C}^{2n\times n}$ such that $\operatorname{Im} U$ is Lagrangian and a set $\mathcal{I} \subseteq \{1, 2, \dots, n\}$, define the symplectic swap $\Pi_{\mathcal{I}}$ as in (1.8) and partition

$$\Pi_{\mathcal{I}} U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \quad U_1, U_2 \in \mathbb{C}^{n\times n}. \tag{1.9}$$

If the top $n \times n$ block U_1 is invertible then

$$U \sim \mathcal{G}_{\mathcal{I}}(U_2U_1^{-1}), \tag{1.10}$$

where

$$\mathcal{G}_{\mathcal{I}}(X) = \Pi_{\mathcal{I}}^{T} \begin{bmatrix} I_n \\ X \end{bmatrix}, \quad X \in \mathbb{C}^{n\times n}. \tag{1.11}$$

Note that (1.11) generalizes the notion of a Riccati matrix (1.6) by not requiring that the identity matrix is contained in the top block but that it can be pieced together (modulo signs) from a subset of rows of the matrix $\mathcal{G}_{\mathcal{I}}(X)$. Clearly, the pair $(\mathcal{I}, X)$, with $X = U_2 U_1^{-1}$, identifies $\operatorname{Im} U$ uniquely.

The representation (1.10) is called the *permuted Lagrangian graph representation* in [79] and it generalizes the representation (1.7), while keeping the property that $\operatorname{Im} \mathcal{G}_{\mathcal{I}}(X)$ is Lagrangian if and only if X is Hermitian. We use the name *permuted Riccati representation* (or *basis*) here.

Theorem 1.2.2 ([79, Sec. 3])**.** *Let $U \in \mathbb{C}^{2n\times n}$. The following properties are equivalent.*

1. $\operatorname{Im} U$ *is Lagrangian.*
2. *For a particular choice of $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ we have $U \sim \mathcal{G}_{\mathcal{I}}(X)$ and it holds that $X = X^*$.*
3. *For all choices of $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ such that $U \sim \mathcal{G}_{\mathcal{I}}(X)$, it holds that $X = X^*$.*

Moreover, for each U satisfying the above properties there exists at least one $\mathcal{I}_{\text{opt}} \subseteq \{1, 2, \ldots, n\}$ such that $U \sim \mathcal{G}_{\mathcal{I}_{\text{opt}}}(X_{\text{opt}})$ and $X_{\text{opt}} = X_{\text{opt}}^*$ satisfies

$$|(X_{\text{opt}})_{ij}| \leq \begin{cases} 1, & \text{if } i = j, \\ \sqrt{2}, & \text{otherwise.} \end{cases} \tag{1.12}$$

As with the Riccati matrix representation, we can use any of the matrices X such that $U \sim \mathcal{G}_{\mathcal{I}}(X)$ to store the Lagrangian subspace $\operatorname{Im} U$ on a computer and operate on it, since the property that X must be Hermitian can be easily enforced. The choice with $\mathcal{I}_{\text{opt}}$ is particularly convenient from a numerical point of view: using (1.12), we can prove that $\kappa(\mathcal{G}_{\mathcal{I}}(X))$ cannot be too large [79, Thm. 8.2]. Moreover, using the matrix X_{opt} improves numerical stability in several contexts, see [88].

Example 1.2.3. For the matrix

$$U = \begin{bmatrix} 1 & 2 & 5 & 8 \\ 1 & 1 & 3 & 5 \end{bmatrix}^{T},$$

whose column space $\operatorname{Im} U$ is Lagrangian we have

$$U \sim \mathcal{G}_{\emptyset}\left(\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}\right), \qquad U \sim \mathcal{G}_{\{1\}}\left(\begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix}\right),$$
$$U \sim \mathcal{G}_{\{2\}}\left(\begin{bmatrix} -1/3 & 2/3 \\ 2/3 & -1/3 \end{bmatrix}\right), \qquad U \sim \mathcal{G}_{\{1,2\}}\left(\begin{bmatrix} 3 & -2 \\ -2 & 1 \end{bmatrix}\right).$$

All the matrices X in $\mathcal{G}_{\mathcal{I}}(X)$ are Hermitian. For $\mathcal{I}_{\text{opt}} = \{2\}$, the condition (1.12) is satisfied.

Example 1.2.4. For the matrix

$$U = \begin{bmatrix} 1 & 2 & 6 & 6 \\ 1 & 1 & 4 & 4 \end{bmatrix}^T,$$

whose column space $\operatorname{Im} U$ is Lagrangian we have

$$U \sim \mathcal{G}_{\emptyset}\left(\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right),$$

and for both

$$U \sim \mathcal{G}_{\{1\}}\left(\begin{bmatrix} -1/2 & 1 \\ 1 & 0 \end{bmatrix}\right) \quad \text{and} \quad U \sim \mathcal{G}_{\{2\}}\left(\begin{bmatrix} 0 & 1 \\ 1 & -1/2 \end{bmatrix}\right)$$

the condition (1.12) is satisfied. The top 2×2 block of $\Pi_{\{1,2\}}U$ is singular, hence the permuted Riccati representation (1.10) does not exist for $\mathcal{I} = \{1, 2\}$.

Converting between two different permuted Riccati representations is achieved via the *symmetric principal pivot transform* (PPT). The symmetric PPT of a matrix $X \in \mathbb{C}^{n\times n}$ with respect to an index set $\mathcal{K} \subseteq \{1, 2, \ldots, n\}$ is defined as the matrix Y such that

$$\begin{aligned} Y_{\mathcal{K}\mathcal{K}} &= -X_{\mathcal{K}\mathcal{K}}^{-1}, & Y_{\mathcal{K}\mathcal{K}^c} &= X_{\mathcal{K}\mathcal{K}}^{-1} X_{\mathcal{K}\mathcal{K}^c}, \\ Y_{\mathcal{K}^c\mathcal{K}} &= X_{\mathcal{K}^c\mathcal{K}} X_{\mathcal{K}\mathcal{K}}^{-1}, & Y_{\mathcal{K}^c\mathcal{K}^c} &= X_{\mathcal{K}^c\mathcal{K}^c} - X_{\mathcal{K}^c\mathcal{K}} X_{\mathcal{K}\mathcal{K}}^{-1} X_{\mathcal{K}\mathcal{K}^c}, \end{aligned} \tag{1.13}$$

where $X_{\mathcal{I}\mathcal{J}}$ denotes a submatrix of X with rows and columns indexed by the sets $\mathcal{I}$ and $\mathcal{J}$, respectively (the order of the indices does not matter as long as it is chosen consistently), and $\mathcal{K}^c$ is the complement of $\mathcal{K}$ in $\{1, 2, \ldots, n\}$.

For instance, if $\mathcal{K} = \{1, 2, \ldots, k\}$ is the set of indices corresponding to the leading block of X, then

$$X = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \end{array}, \quad Y = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} -X_{11}^{-1} & X_{11}^{-1}X_{12} \\ X_{21}X_{11}^{-1} & X_{22} - X_{21}X_{11}^{-1}X_{12} \end{bmatrix} \end{array}.$$

Note the peculiar structure of this transformation: a principal submatrix $X_{\mathcal{K}\mathcal{K}}$ of X is replaced by the negative of its inverse, and its complement is overwritten by the *Schur complement* $X_{\mathcal{K}^c\mathcal{K}^c} - X_{\mathcal{K}^c\mathcal{K}}X_{\mathcal{K}\mathcal{K}}^{-1}X_{\mathcal{K}\mathcal{K}^c}$ of $X_{\mathcal{K}\mathcal{K}}$ in X.

The map $X \mapsto Y$ defined in (1.13) is a symmetric variant of the *principal pivot transform* (PPT), which appears across various fields under different names. In statistics it is known as *partial inversion* in the context of linear graphical chain models [126], or as the *sweep operator* when it is used to solve least-squares regression problems [45]. Duffin, Hazony, and Morrison analyze network synthesis [32] and call it *gyration*. In numerical linear algebra the PPT is often called the *exchange operator* and it is of interest since it relates computations in one structured class of matrices to another. Stewart and Stewart [109] use the exchange operator to generate J-orthogonal matrices (matrices $Q \in \mathbb{R}^{n\times n}$ such that $Q^T JQ = J$, where $J = \operatorname{diag}(\pm 1)$ is a signature matrix) from hyperbolic Householder transformations. Higham [57] further shows how to obtain a hyperbolic CS decomposition of a J-orthogonal matrix directly from the standard CS decomposition via the exchange operator. Moreover, certain important classes of matrices are invariant under this operation. Tucker [115] shows that the principal pivot transform of a P-matrix (a matrix whose principal minors are all positive) is again a P-matrix when the matrix is real. This result was extended to complex P-matrices by Tsatsomeros in [114], where further details of the history and properties of the PPT can be found. An overview by Higham [57, Sec. 2] provides additional references.

The following result shows how to use the symmetric PPT (1.13) to convert between permuted Riccati representations of a Lagrangian subspace. This is the crucial step in the optimization algorithm [79, Alg. 2] by Mehrmann and Poloni that computes the bounded representation $(\mathcal{I}_{\text{opt}}, X_{\text{opt}})$ from (1.12).

Lemma 1.2.5 ([79, Lem. 5.1]). *Let $\mathcal{I}, \mathcal{J} \subseteq \{1, 2, \ldots, n\}$, and let $U \in \mathbb{C}^{2n\times n}$ be a matrix whose column space is Lagrangian and such that $U \sim \mathcal{G}_{\mathcal{I}}(X)$. Let $\mathcal{K}$ be the*

symmetric difference set

$$\mathcal{K} = \{i \in \{1, 2, \ldots, n\} : i \text{ is contained in exactly one among } \mathcal{I} \text{ and } \mathcal{J}\}.$$

Then, $U \sim \mathcal{G}_{\mathcal{J}}(X')$ *if and only if* $X_{\mathcal{K}\mathcal{K}}$ *is invertible, and in this case* $X' = DYD$*, where* Y *is the symmetric PPT of* X *defined in* (1.13) *for the index set* $\mathcal{K}$*, and* D *is the diagonal matrix such that*

$$D_{ii} = \begin{cases} -1, & i \in \mathcal{I} \setminus \mathcal{J}, \\ 1, & \text{otherwise.} \end{cases}$$

Informally speaking, when we wish to transform the matrix X such that $U \sim \mathcal{G}_{\mathcal{I}}(X)$ into the matrix X' so that $U \sim \mathcal{G}_{\mathcal{J}}(X')$ for a new set $\mathcal{J}$, we have to perform a symmetric PPT (1.13) with respect to the indices that we wish to add to or remove from $\mathcal{I}$, and then flip the signs in the rows and columns with the indices that we remove from $\mathcal{I}$.

Example 1.2.6. Take $\mathcal{I} = \{1\}$ and the matrix U from Example 1.2.3 so that $U \sim \mathcal{G}_{\{1\}}(X)$ with

$$X = \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix}.$$

Applying Lemma 1.2.5 transforms between the remaining three representations as follows. For $\mathcal{J} = \emptyset$ Lemma 1.2.5 defines $\mathcal{K} = \{1\}$ and $D = \operatorname{diag}(-1, 1)$. Applying (1.13) to X gives

$$Y = \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix}, \quad X' = DYD = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}.$$

Therefore, $U \sim \mathcal{G}_{\emptyset}(X')$ holds. For $\mathcal{J} = \{2\}$ we have $\mathcal{K} = \{1, 2\}$ and $D = \operatorname{diag}(-1, 1)$. In this case

$$Y = -\begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} -1/3 & -2/3 \\ -2/3 & -1/3 \end{bmatrix}, \quad X' = DYD = \begin{bmatrix} -1/3 & 2/3 \\ 2/3 & -1/3 \end{bmatrix},$$

leading to the representation $U \sim \mathcal{G}_{\{2\}}(X')$. Finally, for $\mathcal{J} = \{1, 2\}$ we have $\mathcal{K} = \{2\}$ and $D = I_2$. It follows that $U \sim \mathcal{G}_{\{1,2\}}(X')$ for

$$X' = Y = \begin{bmatrix} 3 & -2 \\ -2 & 1 \end{bmatrix}.$$

1.3 Thesis outline and research contributions

Chapter 2 introduces definitions and standard results in numerical linear algebra that are of relevance to the main chapters of the thesis.

In Chapter 3 we develop a new way of restoring positive semidefiniteness of an indefinite matrix called shrinking. For an indefinite matrix M_0 we construct a convex linear combination $S(\alpha) = \alpha M_1 + (1-\alpha)M_0$ of M_0 and a positive semidefinite *target matrix* M_1, and define the optimal shrinking parameter as $\alpha_* = \min\{\alpha \in [0,1] : S(\alpha) \text{ is positive semidefinite}\}$. We describe three algorithms for computing α_*: one algorithm is based on the bisection method, with the use of Cholesky factorization to test definiteness, a second employs Newton's method, and a third finds the smallest eigenvalue of a symmetric definite generalized eigenvalue problem. We also show that weights that reflect confidence in the individual entries of M_0 can be used to construct a natural choice of the target matrix M_1 with no changes needed to the computation methods. We treat in detail a practically important problem variant in which a positive semidefinite leading principal submatrix of an indefinite approximation to a correlation matrix remains fixed, showing how the fixed block can be exploited to reduce the cost of the bisection and generalized eigenvalue methods. Furthermore, we demonstrate that incorporating the lower bound on the smallest eigenvalue presents only a trivial change to the methods. The aim of this work was to develop an alternative to computing the nearest correlation matrix and numerical experiments show that when applied to indefinite approximations of correlation matrices shrinking can be at least an order of magnitude faster. An implementation of the bisection algorithm is included in Mark 25 of the NAG Library (2015) as the routine `g02anf` [83], and a bisection method that uses weights to define the target matrix is currently being implemented.

While methods for computing the nearest correlation matrix to a given symmetric matrix A are well developed, little attention has been given to estimating the distance $\mathrm{d}_{\mathrm{corr}}(A) = \|A - \mathrm{ncm}(A)\|_F$ without computing the nearest correlation matrix $\mathrm{ncm}(A)$. Importantly, the iterates produced by the Newton and alternating projections methods are not themselves correlation matrices, so no upper bound on $\mathrm{d}_{\mathrm{corr}}(A)$ is available during the iterations. Our goal in Chapter 4 is to obtain bounds on $\mathrm{d}_{\mathrm{corr}}(A)$ that are inexpensive to compute and are of the correct order of magnitude. Indeed bounds

correct to within a small constant factor are entirely adequate for practical applications. We first present an overview of known bounds on the nearest correlation matrix distance and then derive a variety of new upper bounds for $\mathrm{d}_{\mathrm{corr}}(A)$. The bounds are of two main classes: those based on the eigensystem and those based on a modified Cholesky factorization. For unit diagonal A with $|a_{ij}| \leq 1$ for all $i \neq j$ the eigensystem bounds are shown to overestimate the distance by a factor at most $1 + n\sqrt{n}$. We show that for a collection of matrices from the literature and from practical applications the eigensystem-based bounds are often good order of magnitude estimates of the actual distance; indeed the best upper bound is never more than a factor 5 larger than a related lower bound. The modified Cholesky bounds are less sharp but also less expensive, and they provide an efficient way to test for definiteness of the putative correlation matrix. Both classes of bounds enable a user to identify an invalid correlation matrix relatively cheaply and to decide whether to revisit its construction or to compute a replacement, such as the nearest correlation matrix.

Although a globally quadratically convergent Newton algorithm has been developed to solve the nearest correlation matrix problem, the alternating projections method still remains very widely used. The rate of convergence of this method is at best linear, and it can require a large number of iterations to converge to within a given tolerance. In Chapter 5 we show that Anderson acceleration [3], a technique for accelerating the convergence of fixed-point iterations, can be applied to the alternating projections method and that in practice it brings a significant reduction in both the number of iterations and the computation time. We also show that Anderson acceleration remains effective, and indeed can provide even greater improvements, when it is applied to the variants of the nearest correlation matrix problem in which specified elements are fixed or a lower bound is imposed on the smallest eigenvalue. This is particularly significant for the nearest correlation matrix problem with fixed elements because no Newton method is available for it. Alternating projections is a general method for finding a point in the intersection of several sets and ours appears to be the first demonstration that this class of methods can benefit from Anderson acceleration.

In Chapter 6 we introduce a class of semidefinite Lagrangian subspaces and show that they can be represented by a subset $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ and a Hermitian matrix $X \in \mathbb{C}^{n\times n}$ with the property that the submatrix $X_{\mathcal{I}\mathcal{I}}$ is negative semidefinite and the

submatrix $X_{\mathcal{I}^c\mathcal{I}^c}$ is positive semidefinite. A matrix X with these definiteness properties is a generalization of a quasidefinite matrix; we call it $\mathcal{I}$-semidefinite. Under mild hypotheses which hold true in most applications, the Lagrangian subspace associated to the stabilizing solution of an algebraic Riccati equation is semidefinite. It is well-known that the solutions to the algebraic Riccati equations can be obtained by computing deflating subspaces of certain Hamiltonian and symplectic pencils. We show that there is a bijection between these pencils and semidefinite Lagrangian subspaces; hence this structure is ubiquitous in control theory. The symmetric PPT converts between different representations of Lagrangian subspaces. For a semidefinite Lagrangian subspace, we prove that the symmetric PPT of an $\mathcal{I}$-semidefinite matrix X is a $\mathcal{J}$-semidefinite matrix X', and we derive an implementation of the transformation $X \mapsto X'$ that both makes use of the definiteness properties of X and guarantees the definiteness of the submatrices of X' in finite arithmetic. We use the resulting formulae to obtain a semidefiniteness-preserving version of an optimization algorithm introduced by Mehrmann and Poloni [79, Alg. 2] to compute a pair $(\mathcal{I}_{\text{opt}}, X_{\text{opt}})$ with $M = \max_{i,j}|(X_{\text{opt}})_{ij}|$ as small as possible, and show that using semidefiniteness allows us to obtain a stronger inequality on M with respect to the general case.

Finally, Chapter 7 summarizes our findings.

1.4 Computing environment, software, and test matrices

Numerical experiments reported in this thesis were carried out in MATLAB R2014a on a Linux machine with an Intel Core i7-4910MQ 2.90GHz processor and 16GB RAM.

The NAG library routines are from the NAG Toolbox for MATLAB Mark 24 [82].

The MATLAB codes developed for the shrinking algorithms from Chapter 3 are available at `https://github.com/higham/shrinking`, and Python implementations, which do not require access to the NAG Library, are available at `https://github.com/vsego/shrinking`.

For the modified Cholesky algorithms used in Chapter 4, the MATLAB implementation of the Cheng and Higham algorithm [24] is available from `https://github.com/higham/modified-cholesky`. We are grateful to Hawren Fang for providing us

with the implementation of the modified Cholesky algorithm of Schnabel and Eskow [103] that we have used in our experiments.

The code for the alternating projections method for the nearest correlation matrix used in Chapter 5 is based on that of Higham [59], and the Anderson acceleration implementation is adapted from Walker [121]. MATLAB implementations of the algorithms can be found at https://github.com/higham/anderson-accel-ncm.

Chapters 3, 4, and 5 use the following indefinite symmetric matrices with unit diagonal as test matrices. The matrices are taken from the literature and from applications, and they can can be downloaded in MATLAB form from https://github.com/higham/matrices-correlation-invalid, with the exception of the RiskMetrics matrices, which we do not have permission to distribute.

high02 A matrix of order 3 from Higham [56, p. 334].

tec03 A matrix of order 4 from Turkay, Epperlein, and Christofides [116, $\widehat{\Omega}$ on p. 86,].

bhwi01 A matrix of order 5 from Bhansali and Wise [12, Sec. 2, second matrix].

mmb13 A matrix of order 6 constructed from foreign exchange trading data supplied by the Royal Bank of Scotland [80, p. 36].

fing97 A matrix of order 7 from Finger [39, Table 4].

tyda99R1–tyda99R3 The matrices R_1, R_2, and R_3 of order 8 from Tyagi and Das [117, Table 1]. Although thought by those authors to be correlation matrices, as pointed out by Xu and Evers [129] they have some negative eigenvalues.

usgs13 A matrix of order 94 corresponding to carbon dioxide storage assessment units for the Rocky Mountains region of the United States that was generated during the national assessment of carbon dioxide storage resources [118], kindly provided by Madalyn Blondes of the U.S. Geological Survey. Due to the aggregation methodology construction, the matrix has a natural block structure. Its twelve diagonal blocks, with respective sizes 12, 5, 1, 14, 12, 1, 10, 4, 5, 9, 13, and 8, correspond to individual basins in the region and are all positive definite. The block structure can be clearly seen in Figure 1.1.

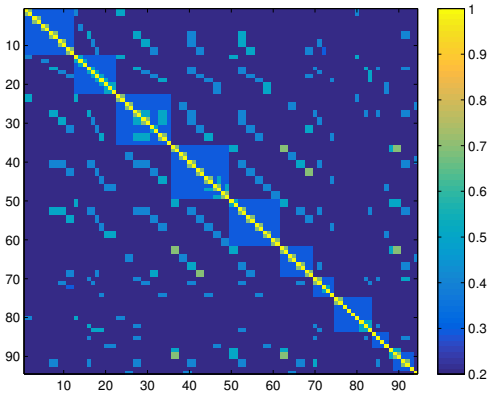

Figure 1.1: The matrix usgs13.

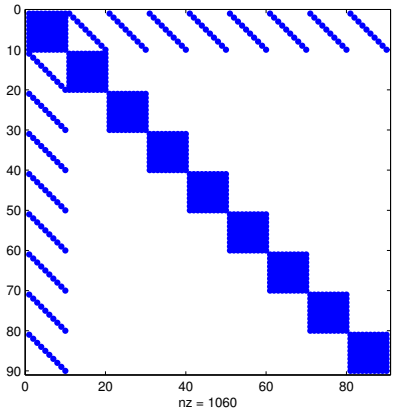

Figure 1.2: The matrix cov90.

RiskMetrics1–RiskMetrics6 Six matrices from the RiskMetrics database. The documentation says that the underlying data sets "contain consistently calculated volatilities and correlation forecasts for use in estimating market risk. The asset classes covered are government bonds, money markets, swaps, foreign exchange and equity indices (where applicable) for 31 currencies, and commodities." Each matrix has dimension 387.

cor1399, cor3210 Two large matrices constructed from stock data: the first of order 1399 is highly rank-deficient and the second of order 3120 is of full rank. The matrices were provided by the investment company Orbis.

In Chapter 5 we also use cov90, an indefinite symmetric block 9×9 matrix with each block of order 10, kindly provided by George Mylnikov of Fischer Francis Trees & Watts, Inc. The diagonal blocks are full, the remaining blocks in the first block-row and block-column are diagonal matrices, and all other elements of the matrix are zero, as can be seen from Figure 1.2.

The MATLAB code used for the experiments in Chapter 6 is based on the package PGDoubling by Poloni, and it can be found at `https://bitbucket.org/fph/pgdoubling-quad`. Test matrices in this chapter are from the benchmark test set [25] that contains 33 problems taken from the standard `carex` test suite [10] for the numerical solution of the continuous-time algebraic Riccati equation, and in addition some examples use different parameters chosen to make the problems more challenging.

Additional details of the experiments are provided in the relevant sections.

Chapter 2

Background Material

Linear algebra is a big part of the small intersection of all general mathematical areas.

—Roger Horn

The material presented here is compiled from the fundamental references for numerical linear algebra: "Matrix Analysis" by Horn and Johnson [62], "Matrix Computations" by Golub and Van Loan [44], and "The Matrix Eigenvalue Problem" by Watkins [125].

2.1 Standard and generalized eigenvalue problems

Let $A \in \mathbb{C}^{n\times n}$. If a scalar $\lambda \in \mathbb{C}$ and a vector $x \in \mathbb{C}^n$, $x \neq 0$ satisfy the equation

$$Ax = \lambda x, \tag{2.1}$$

then λ is called an *eigenvalue* of A and x an *eigenvector* of A associated with λ. Since (2.1) is equivalent to

$$(A - \lambda I)x = 0, \quad x \neq 0,$$

λ is an eigenvalue of A if and only if the matrix $A - \lambda I$ is singular, i.e., rank-deficient. A matrix of order n has n (not necessarily distinct) eigenvalues, and they are the zeros of its characteristic polynomial $\det(A - \lambda I) = 0$.

Two matrices $A, B \in \mathbb{C}^{n\times n}$ are *similar* if there exists a nonsingular matrix $S \in \mathbb{C}^{n\times n}$ such that $B = S^{-1}AS$. Similar matrices have the same eigenvalues. Similarity via a unitary matrix has superior stability properties in numerical computations compared to a general similarity; the canonical form of a matrix under a unitary similarity is given in the following theorem.

Theorem 2.1.1 (Schur decomposition; [125, Thm. 2.2.4]). *Let $A \in \mathbb{C}^{n\times n}$. Then there exist a unitary matrix U and an upper triangular matrix T such that*

$$T = U^{-1}AU = U^*AU.$$

The Schur decomposition $A = UTU^*$ is not unique as the eigenvalues of A can be made to appear in any order on the diagonal of T.

Real matrices may have complex eigenvalues but since they always appear in complex conjugate pairs, if we treat each pair as a unit we can avoid complex arithmetic. A matrix $T \in \mathbb{R}^{n\times n}$ is called *upper quasi-triangular* if it is block upper triangular with the main diagonal blocks all 1×1 or 2×2, and each 2×2 block has complex eigenvalues. This leads to the real version of Schur's theorem, also known as the Wintner–Murnaghan theorem.

Theorem 2.1.2 (Real Schur decomposition; [125, Thm. 2.2.6]). *Let $A \in \mathbb{R}^{n\times n}$. Then there exist an orthogonal matrix U and an upper quasi-triangular matrix T such that*

$$T = U^{-1}AU = U^TAU.$$

Diagonal blocks in T can again appear in any order.

We can generalize the eigenvalue problem for one matrix in the following way. Let $A, B \in \mathbb{C}^{n\times n}$. The set of all matrices of the form $A - \lambda B$ for $\lambda \in \mathbb{C}$ is called a *matrix pencil*. The *eigenvalues of the pencil* are the elements of a set

$$\Lambda(A, B) = \{z \in \mathbb{C}\colon \det(A - zB) = 0\}.$$

If λ is an eigenvalue of the pencil and

$$Ax = \lambda Bx, \quad x \neq 0 \tag{2.2}$$

then x is called an *eigenvector*. When $B = I$ the equation (2.2) reduces to the standard eigenvalue equation (2.1). As such the eigenvalues of a pencil are usually called *generalized eigenvalues* and the problem (2.2) is called the *generalized eigenvalue problem*.

If the matrix B is nonsingular than $Ax = \lambda Bx$ is equivalent to standard eigenvalue problems $AB^{-1}x = \lambda x$ and $B^{-1}Ax = \lambda x$, and there are n finite generalized eigenvalues. If B is rank-deficient then either there are $k < n$ finite generalized eigenvalues and

$n-k$ *infinite* eigenvalues, or every complex number λ is an eigenvalue. In the latter case $\det(A-\lambda B)$ is identically a zero polynomial and we say that the pencil $A-\lambda B$ is *singular*.

The role of a similarity transformation in the generalized eigenvalue problem context is taken by an *equivalence* transformation. Matrix pencils $A_1-\lambda B_1$ and $A_2-\lambda B_2$ are equivalent if there exist nonsingular matrices U and V such that

$$A_1-\lambda B_1 = U(A_2-\lambda B_2)V.$$

Equivalent matrix pencils have the same eigenvalues.

The analog to the Schur decomposition in a generalized eigenvalue problem setting is the following.

Theorem 2.1.3 (Generalized Schur decomposition; [44, Thm. 7.7.1])**.** *If A and B are in $\mathbb{C}^{n\times n}$, then there exist unitary Q and Z such that $Q^*AZ=T$ and $Q^*BZ=S$ are upper triangular. If for some k, t_{kk} and s_{kk} are both zero, then $\Lambda(A,B)=\mathbb{C}$. Otherwise, $\Lambda(A,B)=\{t_{ii}/s_{ii}\colon s_{ii}\neq 0\}$.*

If there is a k such that $t_{kk}=s_{kk}=0$ so that a pencil $A-\lambda B$ is singular we might hope that the ratios of other diagonal elements still have some meaning. Interestingly, this is not true, as shown by Wilkinson in [127], where the following example can be found. Consider the two triangular matrices

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & 0 & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{bmatrix}, \quad B=\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ & 0 & b_{23} & b_{24} \\ & & b_{33} & b_{34} \\ & & & b_{44} \end{bmatrix},$$

where the other elements in the upper triangles are nonzero. Then a_{ii}/b_{ii}, $i=1,3,4$ have no meaningful relationship with the problem $Ax=\lambda Bx$, which can be shown as follows. Let R_{12} be a rotation in the $(1,2)$ plane [44, Sec. 5.1.8]. In the regular case, since the matrix pencil $A-\lambda B$ is equivalent to $AR_{12}-\lambda BR_{12}$, their eigenvalues are the same, and in particular the ratios of the corresponding diagonal elements are the

same. Here, the matrices AR_{12} and BR_{12} are of the form

$$\begin{bmatrix} a'_{11} & a'_{12} & a_{13} & a_{14} \\ & 0 & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} b'_{11} & b'_{12} & b_{13} & b_{14} \\ & 0 & b_{23} & b_{24} \\ & & b_{33} & b_{34} \\ & & & b_{44} \end{bmatrix},$$

where

$$a'_{11} = a_{11}c - a_{12}s, \quad a'_{12} = a_{11}s + a_{12}c,$$

$$b'_{11} = b_{11}c - b_{12}s, \quad b'_{12} = b_{11}s + b_{12}c,$$

in which s and c are the sine and cosine defining the rotation. Then

$$\frac{a'_{11}}{b'_{11}} = \frac{a_{11}c - a_{12}s}{b_{11}c - b_{12}s}, \tag{2.3}$$

so unless $a_{11}/a_{12} = b_{11}/b_{12}$, the right-hand side in (2.3) can take any value by suitable choice of s and c, in particular, it can be made zero or infinity, and certainly, it can be made different from a_{11}/b_{11}.

If A and B are real, then the following decomposition corresponds to the real Schur decomposition.

Theorem 2.1.4 (Generalized real Schur decomposition; [44, Thm. 7.7.2])**.** *If A and B are in $\mathbb{R}^{n\times n}$, then there exist orthogonal Q and Z such that Q^TAZ is upper quasi-triangular and Q^TBZ is upper triangular.*

As in the standard Schur decomposition, the diagonal (blocks) elements in the (quasi)-triangular matrices can be made to appear in any order we specify. The re-ordering of the diagonal elements (blocks) is quite an interesting problem both in the standard and the generalized Schur decompositions [21], [65].

2.2 Definiteness and the Cholesky factorization

A Hermitian matrix $A \in \mathbb{C}^{n\times n}$ is

- *positive semidefinite* if $x^*Ax \geq 0$ for every $x \in \mathbb{C}^n$.
- *positive definite* if $x^*Ax > 0$ for every $0 \neq x \in \mathbb{C}^n$.

- *negative semidefinite* if $x^*Ax \leq 0$ for every $x \in \mathbb{C}^n$.
- *negative definite* if $x^*Ax < 0$ for every $0 \neq x \in \mathbb{C}^n$.
- *indefinite* if there exist $x, y \in \mathbb{C}^n$ such that $(x^*Ax)(y^*Ay) < 0$.

The class of positive semidefinite matrices provides one generalization to matrices of the notion of a nonnegative real number. There are several characterizations of positive (semi)definiteness and we shall make frequent use of the following.

Theorem 2.2.1 ([62, Thm. 7.2.1]). *A Hermitian matrix is positive semidefinite if and only if all of its eigenvalues are nonnegative. It is positive definite if and only if all of its eigenvalues are positive.*

A key property linked to positive definite matrices is the Cholesky factorization.

Theorem 2.2.2 (Cholesky factorization; [125, Thm. 1.4.1]). *Let $A \in \mathbb{C}^{n\times n}$ be Hermitian positive definite matrix. Then there exists a unique upper triangular $R \in \mathbb{C}^{n\times n}$ such that the diagonal entries of R are real and positive and $A = R^*R$. If A is real, then R is real.*

For a positive semidefinite matrix, the situation is more subtle.

Theorem 2.2.3 ([55, Thm. 10.9]). *Let $A \in \mathbb{C}^{n\times n}$ be positive semidefinite of rank r.*

(a) There exists at least one upper triangular R with nonnegative diagonal elements such that $A = R^*R$.

(b) There is a permutation Π such that $\Pi^T A\Pi$ has a unique Cholesky factorization, which takes the form

$$\Pi^T A\Pi = R^*R, \quad R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}, \tag{2.4}$$

where R_{11} is $r \times r$ upper triangular with positive diagonal elements.

The uniqueness in Theorem 2.2.3(b) refers to the (1,1) diagonal block R_{11} of the Cholesky factor R in (2.4), as for a permutation matrix $P = \mathrm{diag}(I_r, \hat{P})$, where $\hat{P}$ is any permutation matrix of order $n - r$, we have that $(\Pi P)^T A(\Pi P) = \hat{R}^*\hat{R}$, with

$$\hat{R} = \begin{bmatrix} R_{11} & \hat{R}_{12} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12}\hat{P} \\ 0 & 0 \end{bmatrix}.$$

The notion of definiteness extends to matrix pencils where it also implies significant theoretical and computational benefits. For $A, B \in \mathbb{C}^{n\times n}$ Hermitian, the pencil $A-\lambda B$ is called *definite* if there exists a value μ for which the matrix $A-\mu B$ is positive definite. The matrices A and B that define a definite pencil are simultaneously diagonalizable: there exists a nonsingular matrix X such that $X^*AX = \mathrm{diag}(a_1, \ldots, a_n)$ and $X^*BX = \mathrm{diag}(b_1, \ldots, b_n)$, and moreover the generalized eigenvalues a_i/b_i are all real [44, Cor. 8.7.2].

Definite pencils can be transformed into pencils in which one of the matrices is positive definite, and there are algorithms for attempting to find such a transformation [26], [47]. The reason for this is that we can then transform the generalized eigenvalue problem stably to a standard Hermitian eigenvalue problem by means of the Cholesky factorization, see, for example, [28], for which there are efficient, structure exploiting algorithms available.

The Cholesky factorization and definite pencils play a crucial role in Chapter 3.

2.2.1 Testing for definiteness in finite arithmetic

In various practical applications a key question is to determine whether a given Hermitian matrix A is positive (semi)definite or not. As argued in [52, Sec. 5], an arbitrarily small perturbation can make a singular positive semidefinite matrix become positive definite and hence in finite precision arithmetic testing for positive semidefiniteness is equivalent to testing for positive definiteness.

Cholesky factorization eliminates the need for an (expensive) computation of any of the eigenvalues: to check if A is positive definite we attempt to compute its Cholesky factorization and declare the matrix positive definite if the process succeeds. Although it might seem numerically dangerous to apply Cholesky factorization to a potentially indefinite matrix, this approach is numerically stable [52, Sec. 5].

The success or failure of the standard Cholesky factorization for a singular positive semidefinite matrix A is unpredictable and instead we need to use the Cholesky factorization with complete pivoting. If the factorization runs to completion with positive pivots we declare the matrix positive semidefinite. If the factorization terminates with a nonpositive pivot, we declare the matrix positive semidefinite if the norm of the remaining block S_k (the Schur complement) satisfies $\|S_k\| \leq c_n u \|A\|$, where c_n is a

constant and u is the unit roundoff, and indefinite otherwise. A weakness of this approach is that c_n could potentially have to be of order 4^{n-1} in order to not misdiagnose a positive semidefinite matrix as indefinite [54], [55, Sec. 10.3.2].

2.2.2 Modified Cholesky factorizations

While invalid correlation matrices are usually encountered in statistics and data analysis applications, a problem of having an indefinite matrix in place of a positive semidefinite one is well known in optimization, where *modified Cholesky algorithms* are used to deal with, for example, indefinite Hessians. Given a symmetric, possibly indefinite matrix A these algorithms construct a factorization

$$P^T(A+E)P = LDL^T,$$

where P is a permutation matrix, L is unit lower triangular, and $A+E$ is positive semidefinite. The algorithms of Gill, Murray, and Wright [42, Sec. 4.4.2.2] and Eskow and Schnabel [102], [103] produce a diagonal D and a diagonal E, while the algorithm of Cheng and Higham [24] produces a block diagonal D with diagonal blocks of order 1 or 2 and an E that is generally full.

In Chapter 4 we shall make use of the above four modified Cholesky algorithms. The review paper by Fang and O'Leary [36] provides an overview and a comparison of the modified Cholesky methods.

2.3 The least-squares problem

For a given matrix $A \in \mathbb{C}^{m\times n}$, $m \geq n$ and a vector $b \in \mathbb{C}^m$ the *linear least-squares* problem is

$$\min_{x\in\mathbb{C}^n} \|Ax-b\|_2. \tag{2.5}$$

Two fundamental matrix decompositions appear in the solution methods for (2.5). They are the QR factorization and the singular value decomposition (SVD).

Theorem 2.3.1 (QR factorization; [62, Thm. 2.1.14]). *Let $A \in \mathbb{C}^{m\times n}$ be given, $m \geq n$. Then $A = QR$, where $Q \in \mathbb{C}^{m\times m}$ is unitary and $R \in \mathbb{C}^{m\times n}$ is upper trapezoidal, that*

is

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

with $R_1 \in \mathbb{C}^{n\times n}$ *upper triangular with nonnegative diagonal elements. Partitioning* $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ *conformably to* R *we have* $A = Q_1 R_1$*, which is called a* reduced QR factorization.

If A *is of full rank then the factors* Q_1 *and* R_1 *are uniquely determined and the diagonal elements of* R_1 *are all positive.*

If A *is real,* Q *and* R *are real.*

Theorem 2.3.2 (SVD; [62, Thm. 2.6.3])**.** *Let* $A \in \mathbb{C}^{m\times n}$*. There exist unitary matrices* $U \in \mathbb{C}^{m\times m}$ *and* $V \in \mathbb{C}^{n\times n}$ *such that*

$$A = U\Sigma V^*, \quad \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}) \in \mathbb{R}^{m\times n}$$

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$*.*

If A *is real, then* U *and* V *are also real.*

The nonnegative real numbers $\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}$ are the *singular values* of A and the columns of U and V are the left and right *singular vectors* of A, respectively. If we take the SVD of A, $A = U\Sigma V^*$ it follows that $A^*A = V\Sigma^*\Sigma V^*$, which is the spectral decomposition of A^*A. Hence, singular values are the nonnegative square roots of the $\min(m, n)$ largest eigenvalues of A^*A.

The SVD allows for the definition of a matrix inverse to be generalized to singular and even rectangular matrices. The *Moore–Penrose generalized inverse* of a matrix $A \in \mathbb{C}^{m\times n}$, whose SVD is $A = U\Sigma V^*$, is a matrix $A^\dagger \in \mathbb{C}^{n\times m}$ defined as

$$A^\dagger = V\Sigma^\dagger U^*,$$

where $\Sigma^\dagger$ is obtained from Σ by replacing each positive singular value with its inverse and then transposing the matrix.

We next summarize the properties of the solution set of the least-squares problem (2.5), using the results in [44, Sec. 5.3 and Sec. 5.5].

Theorem 2.3.3. *Let* $\mathcal{X} = \{x \in \mathbb{C}^n \colon \|Ax - b\|_2 = \min\}$ *be the set of the least-squares solutions to* $Ax = b$*.*

1. $x \in \mathcal{X}$ *if and only if* x *satisfies the* normal equations $A^*Ax = A^*b$.

2. $\mathcal{X}$ *is convex.*

3. *There is a unique* $x_{LS} \in \mathcal{X}$ *having minimum 2-norm.*

4. $\mathcal{X} = \{x_{LS}\}$ *if and only if* A *is of full rank.*

5. $x_{LS} = A^\dagger b$.

Solving least-squares problems is required in Chapter 5. The oldest method to compute x_{LS} when the matrix A is of full rank is to form and solve the normal equations $A^*Ax = A^*b$. Since the matrix A^*A is positive definite in this case, we can use its Cholesky factorization to solve the normal equations by two triangular linear system solves. In finite arithmetic this method might suffer from numerical instability resulting from explicitly computing the Gram matrix A^*A [55, Sec. 20.4]. Development of the structured version of the symmetric PPT in Chapter 6 that avoids forming certain Gram matrices is motivated by this fact.

A backward stable method to solve the full rank least-squares problem uses the QR decomposition [55, Sec. 20.2]. When A is rank-deficient, of the infinitely many solution to the least-squares problem it is of interest to compute x_{LS}, which is now the unique minimal 2-norm solution. This is achieved via the SVD. An excellent reference for the least-squares problem is the book by Björk [15].

Chapter 3

Restoring Definiteness via Shrinking

It is better to solve one problem five ways than to solve five problems the same way.
—George Pólya

3.1 Introduction

Covariance matrices and correlation matrices constructed from discrete sets of empirical data play a key role in many applications. These matrices are symmetric positive semidefinite, with a correlation matrix also having unit diagonal. In this chapter we develop a new method to repair invalid covariance and correlation matrices inspired by an idea from statistics called *shrinking*, which has a long history going back to the work of Stein beginning in the 1950s, and is widely used in statistical estimation; see, for example, [27], [71], [72], [101], [128] and the references therein. A basic idea of shrinking is to form a convex linear combination $\alpha M_1 + (1-\alpha)M_0$ of two correlation or covariance matrices, where $\alpha \in [0,1]$ is chosen based on statistical considerations in order to obtain an estimator that has better properties than the extremes M_0 and M_1.

Our use of shrinking differs from this standard usage in two respects.

1. For us, M_0 is indefinite, not positive semidefinite, so it is not a covariance matrix or a correlation matrix.

2. We make no statistical assumptions about M_0 or M_1 and choose α so that $\alpha M_1 + (1-\alpha)M_0$ is positive semidefinite based solely on information in the matrices M_0 and M_1.

The possibility of using shrinking for restoring definiteness was mentioned by Devlin, Gnanadesikan, and Kettenring [31, Sec. 4.4] and also by Kupiec [66, Sec. 5], who

suggests a "trial and error" way of choosing the shrinking parameter α. Rebonato and Jäckel [97] criticize Kupiec's suggestion on the grounds that it is expensive, since each trial requires a full eigenvalue decomposition, that a target matrix must be chosen, and that "there is no way of determining to what extent the resulting matrix is optimal in any easily quantifiable sense."

Our analysis overcomes the drawbacks pointed out by Rebonato and Jäckel. We define an optimal shrinking parameter that produces a minimal elementwise perturbation to M_0 in the direction of the difference between the target matrix M_1 and the initial approximation M_0, and propose three algorithms for computing the optimal parameter, none of which requires repeated full eigenvalue decompositions.

This rest of this chapter is organized as follows. In the next section we define the shrinking problem, characterize the solution, and discuss the choice of a target matrix. We present three methods to compute the optimal shrinking parameter in section 3.3: one based on the bisection method, a second based on Newton's method, and a third that solves a symmetric definite generalized eigenvalue problem. Since shrinking is proposed as an alternative to computing the nearest correlation matrix we therefore also address the additional requirements that appear in the nearest correlation matrix problem. In section 3.4 we explain how elementwise weighting can be incorporated into the choice of a target matrix. In section 3.5 we focus on restoring definiteness of a correlation matrix while preserving a specified positive semidefinite leading principal submatrix. We show how the bisection and generalized eigenvalue methods can be adapted to exploit the problem structure, explain how the case of a singular fixed block can be reduced to the nonsingular case, show how a lower bound on the smallest eigenvalue can be incorporated, and describe how multiple fixed diagonal blocks can be treated. Numerical experiments are presented in section 3.6, which include a comparison of shrinking with the solution of the nearest correlation matrix problem.

3.2 The shrinking problem

Given a real symmetric indefinite matrix M_0 of order N our task is to modify M_0 to make it positive semidefinite by computing a convex linear combination of M_0 and a

chosen positive semidefinite target matrix M_1. Hence we consider the matrix

$$S(\alpha) = \alpha M_1 + (1-\alpha)M_0, \quad \alpha \in [0,1]. \tag{3.1}$$

Clearly, $S(\alpha)$ is symmetric for every α, $S(0) = M_0$ is indefinite, and $S(1) = M_1$ is positive semidefinite. We define the optimal shrinking parameter as

$$\alpha_* = \min\{\, \alpha \in [0,1] \colon S(\alpha) \text{ is positive semidefinite} \,\}. \tag{3.2}$$

Since $S(\alpha) = M_0 + \alpha(M_1 - M_0)$, it is clear that we are seeking the elementwise minimal change to M_0 in the direction $M_1 - M_0$.

For another interpretation, note that $M_0 - S(\alpha) = \alpha(M_0 - M_1)$, so $\|M_0 - S(\alpha)\| = \alpha\|M_0 - M_1\|$. Since $M_0 - M_1$ is fixed, this means that $S(\alpha_*)$ is the nearest positive semidefinite matrix to M_0 of the form $S(\alpha)$, in any norm.

We now characterize the optimal shrinking parameter α_*. The following results form the basis for the bisection and Newton methods for computing α_* proposed in sections 3.3.1 and 3.3.2.

Since a symmetric matrix is positive semidefinite if and only if its smallest eigenvalue is nonnegative, we focus on the function $f \colon \mathbb{R} \to \mathbb{R}$ defined by

$$f(\alpha) = \lambda_{\min}\left(S(\alpha)\right), \tag{3.3}$$

where $\lambda_{\min}$ denotes the smallest eigenvalue of a symmetric matrix. Note that f is a continuous function, since the eigenvalues of a matrix are continuous functions of its elements [125, Thm. 2.7.1]. Hence α_* is characterized as

$$\alpha_* = \min\{\, \alpha \in [0,1] \colon f(\alpha) \geq 0 \,\}.$$

Recall that a function $g \colon \mathbb{R} \to \mathbb{R}$ is *concave* if for every $\alpha_1, \alpha_2 \in \mathbb{R}$ and $t \in [0,1]$,

$$g\big(t\alpha_1 + (1-t)\alpha_2\big) \geq tg(\alpha_1) + (1-t)g(\alpha_2).$$

In the following lemma we show that the function f defined in (3.3) is concave. In the proof below we will use the characterization [62, Thm. 4.2.2] for symmetric C of order N,

$$\lambda_{\min}(C) = \min\{\, x^T C x \colon x \in \mathbb{R}^N,\ x^T x = 1 \,\}. \tag{3.4}$$

Lemma 3.2.1. *The function f in (3.3) is concave on $\mathbb{R}$.*

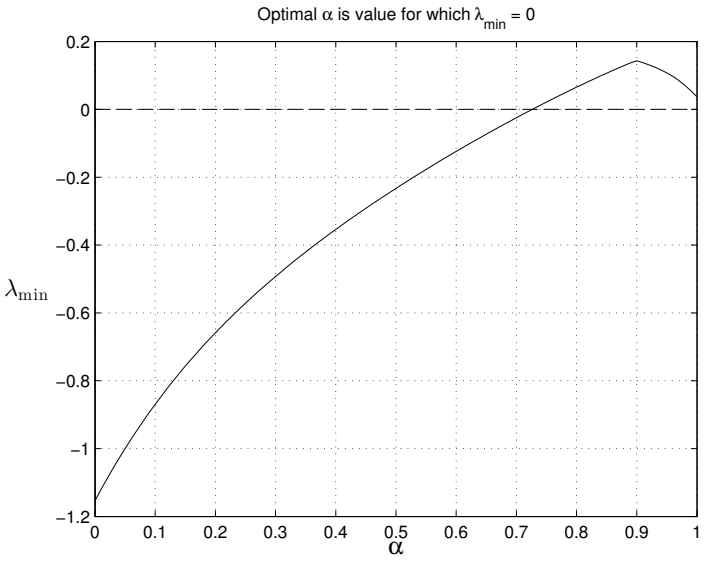

Figure 3.1: Plot of the function $f(\alpha) = \lambda_{\min}(S(\alpha))$ for $S(\alpha)$ in (3.1) for M_1 positive definite.

Proof. Let $\alpha_1, \alpha_2 \in \mathbb{R}$ and $t \in [0,1]$ be arbitrary and note that $S(\alpha)$ is an affine function of α. Then we have

$$\begin{aligned} f(t\alpha_1 + (1-t)\alpha_2) &= \lambda_{\min}\big(S(t\alpha_1 + (1-t)\alpha_2)\big) \\ &= \lambda_{\min}\big(tS(\alpha_1) + (1-t)S(\alpha_2)\big) \\ &\geq \lambda_{\min}\big(tS(\alpha_1)\big) + \lambda_{\min}\big((1-t)S(\alpha_2)\big) \qquad \text{by (3.4)} \\ &= t\lambda_{\min}\big(S(\alpha_1)\big) + (1-t)\lambda_{\min}\big(S(\alpha_2)\big) \\ &= tf(\alpha_1) + (1-t)f(\alpha_2). \qquad \square \end{aligned}$$

Since $f(0) < 0$, $f(1) = \lambda_{\min}(S(1)) = \lambda_{\min}(M_1)$, and f is concave and continuous, it follows that α_* is the unique zero of f in $(0,1)$ if the matrix M_1 is positive definite. In principle we need to allow M_1 to be positive semidefinite and singular, as can happen in our correlation matrix application discussed in section 3.5, but as we show in section 3.5.4 in that case the problem can be reduced to the case in which M_1 is positive definite. A typical f for an indefinite M_0 and a positive definite M_1 is illustrated in Figure 3.1.

When the only goal is to repair the indefiniteness of the matrix M_0, the target matrix M_1 can be chosen as any positive semidefinite matrix. When M_0 is an invalid correlation matrix and we want $S(\alpha_*)$ to be a valid correlation matrix, then from (3.1)

it follows that the target matrix needs to be a correlation matrix, so that the unit diagonal is preserved. Hence the simplest target in this case is the identity matrix.

3.3 Computing the optimal shrinking parameter

We present three methods to compute the optimal shrinking parameter α_* when M_0 is symmetric indefinite and the target matrix M_1 is positive definite. Recall that in this case α_* is the unique zero in $(0, 1)$ of f in (3.3).

3.3.1 Bisection method

The simplest iterative method to find a zero of a function on a given interval is the bisection method, which yields the following algorithm for our problem.

Algorithm 3.3.1 (Bisection method)**.** *Given the indefinite matrix $M_0 \in \mathbb{R}^{N\times N}$, a positive definite target matrix $M_1 \in \mathbb{R}^{N\times N}$, and a convergence tolerance* tol *this algorithm uses the bisection method to compute the optimal shrinking parameter α_* defined by* (3.2).

```
1  α_ℓ = 0, α_r = 1
2  while α_r − α_ℓ > tol
3        α_m = (α_ℓ + α_r)/2
4        if S(α_m) is not positive semidefinite
5           α_ℓ = α_m
6        else
7           α_r = α_m
8        end
9  end
10 α_* = α_r.
```

In the last line of the algorithm we have set α_* to α_r rather than to the generally more accurate value $(\alpha_\ell + \alpha_r)/2$ in order to ensure that $S(\alpha_*)$ is positive semidefinite.

The main computational task in Algorithm 3.3.1 is testing for positive semidefiniteness. As argued in section 2.2.1, this is done by attempting to compute the

Cholesky factorization of $S(\alpha_m)$. Hence, we replace step 4 in Algorithm 3.3.1 with "if the Cholesky factorization of $S(\alpha_m)$ breaks down".

The number of steps needed by Algorithm 3.3.1 is $\lceil |\log_2 \mathrm{tol}| \rceil$, where the ceiling function $\lceil \alpha \rceil$ denotes the smallest integer greater than or equal to α, reflecting the linear convergence of the bisection method. However, in practical applications the data is often accurate only to three or four significant digits, in which case the tolerance tol will be of order 10^{-4} or larger and bisection will need less than 15 iterations. The cost per step of Algorithm 3.3.1 depends on the number of successful elimination stages in the Cholesky factorization of $S(\alpha_m)$ and is at most $N^3/3$ flops.

In case M_1 is positive definite we can narrow the interval [0,1] down to $[0, \alpha_W]$ with $\alpha_W < 1$ by using Weyl's inequality [62, Cor 4.3.15]. The lower bound in Weyl's theorem is

$$\lambda_i(S(\alpha)) = \lambda_i\left(\alpha M_1 + (1-\alpha)M_0\right) \geq \lambda_i((1-\alpha)M_0) + \lambda_{\min}(\alpha M_1).$$

Then for the smallest eigenvalue, because $\alpha \in [0,1]$, it follows that

$$\lambda_{\min}(S(\alpha)) \geq (1-\alpha)\lambda_{\min}(M_0) + \alpha\lambda_{\min}(M_1). \tag{3.5}$$

Therefore, if the right-hand side in (3.5) is nonnegative then $S(\alpha)$ is positive semidefinite. Taking into account that $\lambda_{\min}(M_0) < 0$ and $\lambda_{\min}(M_1) > 0$ it follows that for $\alpha \geq -\lambda_{\min}(M_0)/(\lambda_{\min}(M_1) - \lambda_{\min}(M_0)) = \alpha_W$, the matrix $S(\alpha)$ is positive semidefinite and we can replace the right edge of the interval with α_W.

In our experiments there was no significant difference for the bisection algorithm applied on $[0, \alpha_W]$ instead of $[0,1]$ so the additional expense of computing α_W does not seem justified.

3.3.2 Newton's method

For a method with faster convergence than the bisection method it is natural to turn to Newton's method, defined by $\alpha_{k+1} = \alpha_k - f(\alpha_k)/f'(\alpha_k)$. If $\lambda_{\min}(S(\alpha))$ is a simple eigenvalue then f in (3.3) is differentiable and [44, Sec. 7.2.2]

$$f'(\alpha) = x(\alpha)^T S'(\alpha) x(\alpha), \tag{3.6}$$

where $x(\alpha)$ is a unit norm eigenvector for $\lambda_{\min}(S(\alpha))$ and, from (3.1), $S'(\alpha) = M_1 - M_0$. Note that $S'(\alpha)$ is independent of α.

Lemma 3.3.2. *The Newton iteration for finding the zero of* $f(\alpha) = \lambda_{\min}(S(\alpha))$*, where* $S(\alpha)$ *is defined in* (3.1)*, can be written as*

$$\alpha_{k+1} = \frac{x(\alpha_k)^T M_0 x(\alpha_k)}{x(\alpha_k)^T (M_0 - M_1) x(\alpha_k)}, \tag{3.7}$$

where $x(\alpha_k)$ *is a unit norm eigenvector for* $\lambda_{\min}(S(\alpha_k))$ *and it is assumed that* $\lambda_{\min}(S(\alpha_k))$ *is simple for each* k*.*

Proof. Dropping the index k for simplicity, let us look at the quotient $f(\alpha)/f'(\alpha)$. We have, from (3.1),

$$f(\alpha) = \lambda_{\min}(S(\alpha)) = x(\alpha)^T S(\alpha) x(\alpha) = x(\alpha)^T (M_0 + \alpha(M_1 - M_0)) x(\alpha)$$

and, from (3.6),

$$f'(\alpha) = x(\alpha)^T S'(\alpha) x(\alpha) = x(\alpha)^T (M_1 - M_0) x(\alpha).$$

Hence

$$\alpha_{k+1} = \alpha_k - f(\alpha_k)/f'(\alpha_k) = \alpha_k - \alpha_k - \frac{x(\alpha_k)^T M_0 x(\alpha_k)}{x(\alpha_k)^T (M_1 - M_0) x(\alpha_k)},$$

which yields the result. □

Recall that we are looking for $\alpha_* \in (0,1)$ such that $f(\alpha_*) = 0$, where f is continuous and concave with $f(0) < 0$ and $f(1) > 0$. The function f is monotone increasing on an interval $[0, \beta] \subseteq [0,1]$ that contains α_*, but β is not necessarily equal to 1 (and it is possible that f decreases on $[\beta, 1]$, see Figure 3.1). Moreover, f might not be differentiable for all $\alpha \in (0,1)$. However, from considering the geometrical interpretation of the Newton method it follows that for any $\alpha_0 < \alpha_*$ the Newton iterates converge monotonically to α_* and hence we expect that the Newton method for our problem is globally and quadratically convergent, although precise convergence results require some additional assumptions on the smoothness of the function, see [43, Chap. 5.2]. In practice, we can set $\alpha_0 = 0$. Taking all of this into consideration, we have the following algorithm.

Algorithm 3.3.3 (Newton method)**.** *Given the indefinite matrix* $M_0 \in \mathbb{R}^{N \times N}$*, a positive definite target matrix* $M_1 \in \mathbb{R}^{N \times N}$*, and a convergence tolerance* tol *this algorithm uses Newton's method to compute the optimal shrinking parameter* α_* *defined by* (3.2)*.*

1 $\alpha_0 = 0$, $k = 0$
2 while not converged to within tolerance tol
3 Compute $x(\alpha_k)$, a unit norm eigenvector for $\lambda_{\min}(S(\alpha_k))$
by tridiagonalization followed by bisection and inverse iteration.
4 Compute the new iterate α_{k+1} by (3.7).
5 $k = k + 1$
6 end
7 $\alpha_* = \alpha_k$.

A possible stopping test is $|\alpha_{k+1} - \alpha_k| \leq \text{tol}$, which corresponds to the bisection stopping criterion.

The main computational work in the algorithm is computing a unit norm eigenvector for the smallest eigenvalue at each step, and of the many methods that compute one or a few of the (extremal) eigenvalues and their corresponding eigenvectors we have chosen tridiagonalization followed by the bisection method and inverse iteration [30, Sec. 5.3.4]. Other possibilities include the power method [44, Sec. 8.2.1], orthogonal iteration [44, Sec. 8.2.4], and the Lanczos method [44, Sec. 10.1].

Note that there is no guarantee that the computed α_* from Algorithm 3.3.3 will in fact define a positive semidefinite $S(\alpha_*)$ since the iterates stay to the left of α_*.

We do not consider the secant method. While its convergence rate is lower than for the Newton's method it has the general advantage that it avoids the need for derivatives. However, for Newton's method the cost of computing $\lambda_{\min}(S(\alpha_k))$ and $x(\alpha_k)$ is dominated by the cost of the tridiagonalization, so avoiding the computation of $x(\alpha_k)$ produces no significant saving.

3.3.3 Generalized eigenvalue problem

The third method for computing the optimal shrinking parameter is essentially different from the root-finding methods presented above and it provides the most elegant description of α_*. Recall that we are looking for the smallest $\alpha \in (0, 1)$ for which the matrix

$$S(\alpha) = \alpha M_1 + (1 - \alpha) M_0 = E - \alpha F \tag{3.8}$$

is positive semidefinite. The matrix $S(\alpha)$ is a symmetric matrix for every α, which means that it has real eigenvalues. Then $\alpha \mapsto \lambda_1(S(\alpha)), \ldots, \alpha \mapsto \lambda_N(S(\alpha))$ is a continuous parametrization of the N eigenvalue functions $\lambda_1 \geq \cdots \geq \lambda_N$, and in this notation, $\lambda_N = f$ in (3.3).

If α is such that $\lambda_k(S(\alpha)) = 0$ for some k then the matrix $S(\alpha)$ is singular which means, by definition, that α is a generalized eigenvalue of the pencil $E - \alpha F$. It follows that α_*, the zero of λ_N, is a generalized eigenvalue of the matrix pencil $E - \alpha F$, and among all generalized eigenvalues in $(0, 1)$, α_* is the rightmost one.

The matrices $E = M_0$ and $F = M_0 - M_1$ are symmetric but the QZ algorithm for computing the generalized eigenvalues of $E - \alpha F$ cannot exploit the symmetry. However, in our case, it is trivial to obtain a definite pencil. We write

$$S(\alpha) = (1 - \alpha)\left(\frac{\alpha}{1 - \alpha} M_1 + M_0\right)$$

and, since $\alpha_* < 1$, $S(\alpha)$ is singular precisely at the generalized eigenvalues of the definite pencil $M_0 - \mu M_1$, where $\mu = \alpha/(\alpha - 1)$. We find α_* by computing the smallest generalized eigenvalue of this pencil. To do so we transform it to a standard symmetric eigenvalue problem $C - \mu I$, where $C = R^{-T} M_0 R^{-1}$ and $M_1 = R^T R$ is the Cholesky factorization of M_1; see, for example, [28]. To compute the smallest eigenvalue of the matrix C we use tridiagonalization followed by the bisection method.

The algorithm can be summarized as follows.

Algorithm 3.3.4 (Generalized eigenvalue method)**.** *Given the indefinite matrix $M_0 \in \mathbb{R}^{N \times N}$ and a positive definite target matrix $M_1 \in \mathbb{R}^{N \times N}$, this algorithm uses the generalized eigenvalue interpretation to compute the optimal shrinking parameter α_* defined by (3.2).*

1 Compute the Cholesky factorization $M_1 = R^T R$.
2 Form $C = R^{-T} M_0 R^{-1}$ by multiple right-hand side triangular solves.
3 Find μ_*, the smallest eigenvalue of C, by tridiagonalization followed by bisection.
4 $\alpha_* = \mu_*/(\mu_* - 1)$.

3.3.4 Comparison

Unless explicitly stated otherwise, we use [58, Table C-1] and errata at http://www.maths.manchester.ac.uk/~higham/fm/errors.html to compute the approximate cost in flops for the three algorithms, summarized in Table 3.1.

For Algorithm 3.3.1 the cost per step is $N^3/3$ flops, the cost of the Cholesky factorization of a matrix of size N. For Algorithm 3.3.3 we can break down the cost per iteration as follows.

1. Tridiagonalization of a matrix of size N, with Q stored in factored form and not explicitly formed: $4N^3/3$ flops.

2. Bisection to compute the smallest eigenvalue of a tridiagonal matrix of size N followed by inverse iteration to compute the eigenvector y: $O(N)$ flops [8, p. 50].

3. Computing the required eigenvector of M_0 as $x = Qy$ (applying Q in factored form): $O(N^2)$ flops[1] [44, Sec. 5.1.6].

4. Computing α_{k+1}: $O(N^2)$ flops.

Hence, the dominant cost per step of Algorithm 3.3.3 is $4N^3/3$ flops for the tridiagonalization of the matrix $S(\alpha_k)$. Finally, for Algorithm 3.3.4 we have

1. The Cholesky factorization of a matrix of size N: $N^3/3$ flops.

2. Forming C by 2 triangular multiple-right-hand side system solves of size N: $N^3 + N^3/3$ flops, since we have $X = M_0 R^{-1}$ and $C = R^{-T}X$, with C symmetric.

3. Tridiagonalization of a matrix of size N where only T is needed: $4N^3/3$ flops.

4. Bisection to find the smallest eigenvalue of a tridiagonal matrix of size N: $O(N)$ flops.

This gives $3N^3$ as the dominant cost per step of Algorithm 3.3.4.

Which method is the cheapest depends on the desired accuracy, with relatively large values of tol (corresponding to low precision data) favoring the bisection algorithm.

[1] The errata web page http://www.cs.cornell.edu/cv/GVL4/Errata.htm for the fourth edition of [44] notes that the book incorrectly omits the leading 2 on page 238 from this operation count.

Table 3.1: Approximate costs, in flops, of k_1 iterations of the bisection algorithm (Algorithm 3.3.1 with Cholesky factorization), k_2 iterations of Newton's method (Algorithm 3.3.3), and the generalized eigenvalue-based algorithm (Algorithm 3.3.4), all for M_0 of size N.

Bisection	Newton	Generalized eigenvalue
$\frac{k_1 N^3}{3}$	$\frac{4k_2 N^3}{3}$	$3N^3$

The bisection algorithm also has the advantage of being the easiest to implement and it guarantees a positive semidefinite solution.

Note that when $M_1 = I$, the first two lines of Algorithm 3.3.4 are empty and the cost reduces to $4N^3/3$ flops.

3.4 Introducing weights

In this section we explain how weights can be incorporated into the choice of a target matrix if different elements of M_0 are known to vary in reliability. This can be reflected by introducing a symmetric matrix $W \in \mathbb{R}^{N\times N}$ of nonnegative weights $w_{ij} \in [0,1]$ and defining the target matrix as $M_1 = W \circ M_0$, where $\circ$ is the Hadamard (elementwise) product. Then

$$S(\alpha)_{ij} = \big(1 + \alpha(w_{ij} - 1)\big)(M_0)_{ij}.$$

Therefore a weight $w_{ij} = 1$ signifies that the (i,j) element of M_0 must not be changed, while a weight $w_{ij} = 0$ allows that element to be changed as much as necessary. Intermediate values $w_{ij} \in (0,1)$ put a greater restriction (for larger w_{ij}) or lesser restriction (for smaller w_{ij}) on the relative amount by which the (i,j) elements of M_0 can change. The unit diagonal in correlation problems poses no difficulties as it is simply obtained for W with a unit diagonal.

Weighting provides a natural answer to the question of how to choose the target matrix: it is based on the original information in M_0 and the trust that can be put in each individual entry. However, there is no guarantee that M_1 obtained this way is positive semidefinite, which is a requirement for a target matrix in the shrinking method. If the target matrix turns out to be indefinite then the weights are too restrictive and W should be modified.

Since weighting is reflected entirely in the target matrix M_1, all the methods from section 3.3 apply without change. This is in contrast to the H-weighted nearest correlation matrix problem, as discussed in section 1.1.

3.5 Correlation matrix with fixed block

We now consider an important special case of weighting in which the given matrix M_0 is an invalid correlation matrix and has a positive semidefinite leading principal submatrix that must remain fixed. As explained in section 1.1, this problem arises when a correlation matrix is formed from incomplete data sets or through stress testing, and the alternating projections method can be modified to compute the nearest correlation matrix with these elements fixed but the convergence is at best linear and so it can potentially be slow (this is illustrated by several experiments in Chapter 5). Moreover, since each iteration requires a full eigenvalue decomposition, this approach is very expensive.

We propose an alternative replacement matrix based on shrinking. In this case we have an indefinite M_0 partitioned as

$$M_0 = \begin{array}{c} \\ m \\ n \end{array} \begin{array}{c} \begin{array}{cc} m & n \end{array} \\ \begin{bmatrix} A & Y \\ Y^T & B \end{bmatrix} \end{array} \in \mathbb{R}^{N\times N}, \quad A \text{ a correlation matrix}, \quad b_{ii} = 1,\ i = 1\colon n, \tag{3.9}$$

and we wish for A and the unit diagonal of B to remain unchanged. Hence the $(1,1)$ block of the target matrix M_1 must equal A and the $(2,2)$ block must have a unit diagonal. The target matrix

$$M_1 = \operatorname{diag}(A, I) \tag{3.10}$$

is the simplest matrix that satisfies these conditions. We are looking for

$$\alpha_* = \min\{\, \alpha \in [0,1]\colon f(\alpha) \geq 0 \,\}, \tag{3.11}$$

with $f(\alpha) = \lambda_{\min}(S(\alpha))$ and

$$S(\alpha) = \alpha M_1 + (1-\alpha)M_0 = \begin{bmatrix} A & (1-\alpha)Y \\ (1-\alpha)Y^T & \alpha I + (1-\alpha)B \end{bmatrix}. \tag{3.12}$$

In addition to the interpretation mentioned in section 3.2 that the resulting matrix $S(\alpha_*)$ is the elementwise minimal change of M_0 in the direction $M_1 - M_0$, here we

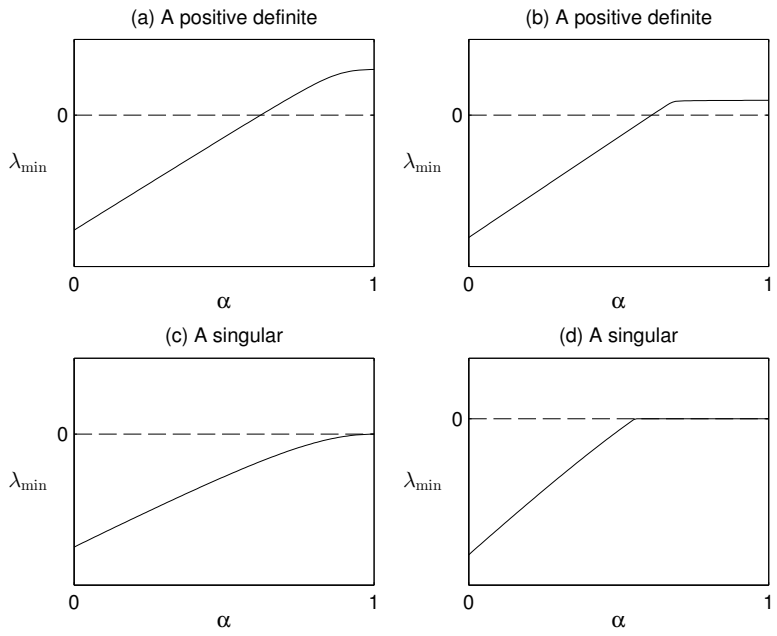

Figure 3.2: Plots of the function $f(\alpha) = \lambda_{\min}(S(\alpha))$ for $S(\alpha)$ in (3.12) for A positive definite in (a) and (b), and positive semidefinite and singular in (c) and (d).

also have that α_* is the minimal relative change applied uniformly to all the unfixed elements of M_0.

A desirable property is that if the rows and columns of A and of B are symmetrically permuted then $S(\alpha_*)$ is permuted in the same way. It is easy to show that this is the case, using the formulae in section 3.5.4.

We first assume that A is positive definite, so that we have a positive definite target matrix. The matrix $S(\alpha)$ has special structure: its leading positive definite block A does not change with α; this can be very efficiently exploited in the bisection method (Algorithm 3.3.1) and the generalized eigenvalue approach (Algorithm 3.3.4) to compute α_*, as we show in the next two sections.

Having the fixed block A singular leads to significant changes to both the problem and the proposed methods for computing the optimal shrinking parameter, as discussed further in section 3.5.4 and illustrated by the differing plots in Figure 3.2.

We now show how to modify the bisection and generalized eigenvalue methods to exploit the fixed block.

3.5.1 Bisection method

Let us look more closely at the Cholesky-based test for definiteness in the case where $S(\alpha)$ is given by (3.12). If $R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$ is the Cholesky factor of $S(\alpha)$ then

1. R_{11} is the Cholesky factor of the fixed block A.

2. R_{12} is the solution of the multiple right-hand side triangular system $R_{11}^T R_{12} = (1-\alpha)Y$.

3. R_{22} is the Cholesky factor of $\alpha I + (1-\alpha)B - R_{12}^T R_{12}$.

Note that R_{11} is independent of α and so needs to be computed only once. Also, since $R_{12} = (1-\alpha)R_{11}^{-T}Y$, once we have computed the solution X of $R_{11}^T X = Y$ then for each α we do not need to solve a linear system for R_{12}, but can instead set $R_{12} = (1-\alpha)X$. Hence, to determine if the matrix $S(\alpha)$ is positive definite or not for a given α we attempt to compute the Cholesky factor of the matrix $\alpha I + (1-\alpha)B - (1-\alpha)^2 X^T X$ (which of course is the Schur complement of A in $S(\alpha)$).

Taking all this into account, our optimized bisection algorithm for the case when A is positive definite is as follows.

Algorithm 3.5.1 (Bisection method). *Given the indefinite matrix M_0 in (3.9) with positive definite $(1,1)$ block A and a convergence tolerance* tol*, this algorithm uses the bisection method with Cholesky factorization to compute the optimal shrinking parameter α_* defined by (3.11) for the target matrix (3.10).*

```
1  α_ℓ = 0,  α_r = 1
2  Compute R_11, the Cholesky factor of A.
3  Compute the solution X of R_11^T X = Y and form Z = X^T X.
4  while α_r − α_ℓ > tol
5        α_m = (α_ℓ + α_r)/2
6        T = α_m I + (1 − α_m)B − (1 − α_m)^2 Z
7        if the Cholesky factorization of T breaks down
8           α_ℓ = α_m
9        else
10          α_r = α_m
```

11 end

12 end

13 $\alpha_* = \alpha_r$.

To estimate the cost of the algorithm, note that for the overhead (steps 2 and 3) we have

1. Cholesky factorization of a matrix of size m: $m^3/3$ flops.

2. Computing X is equivalent to n triangular linear system solves of size m: nm^2 flops.

3. For $Z = X^T X$, since X is $m \times n$, we are computing $(n^2+n)/2$ elements (diagonal and one of the strict triangles), where each is an inner product of vectors of size m, giving in total $(2m-1)(n^2+n)/2$ flops. The dominant term is mn^2.

We add to that the cost of one Cholesky decomposition of a matrix of size n per iteration, which for a given tolerance tol is in total $n^3\lceil|\log_2 \text{tol}|\rceil/3$. Hence, the cost of Algorithm 3.5.1 is at most $m^3/3 + m^2 n + n^2 m + n^3\lceil|\log_2 \text{tol}|\rceil/3$ flops.

3.5.2 Generalized eigenvalue problem

Recall from section 3.3.3 that we are looking for the smallest generalized eigenvalue of the definite pencil $M_0 - \mu M_1$, with M_0 and M_1 now given by (3.9) and (3.10). If $A = R_{11}^T R_{11}$ is the Cholesky factorization then

$$M_0 - \mu M_1 = \begin{bmatrix} A - \mu A & Y \\ Y^T & B - \mu I \end{bmatrix} = \begin{bmatrix} R_{11}^T & 0 \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & R_{11}^{-T} Y \\ Y^T R_{11}^{-1} & B \end{bmatrix} - \mu I \right) \begin{bmatrix} R_{11} & 0 \\ 0 & I \end{bmatrix}.$$

Hence we obtain a standard symmetric eigenvalue problem for the matrix

$$C = \begin{bmatrix} I & R_{11}^{-T} Y \\ Y^T R_{11}^{-1} & B \end{bmatrix}, \tag{3.13}$$

at the cost of one Cholesky factorization and one multiple right-hand side triangular system solve.

In summary, we have the following algorithm.

Algorithm 3.5.2 (Generalized eigenvalue method)**.** *Given the indefinite matrix* M_0 *in* (3.9) *with positive definite* $(1,1)$ *block* A *this algorithm uses the generalized eigenvalue interpretation to compute the optimal shrinking parameter* α_* *defined by* (3.11) *for the target matrix* (3.10)*.*

1 Compute R_{11}, the Cholesky factor of A.
2 Compute the solution X of $R_{11}^T X = Y$ and form C from (3.13).
3 Find μ_*, the smallest eigenvalue of the matrix C, by tridiagonalization (exploiting the identity block) followed by the bisection method.
4 $\alpha_* = \mu_*/(\mu_* - 1)$.

We break down the cost of Algorithm 3.5.2 as follows.

1. Cholesky factorization of a matrix of size m: $m^3/3$ flops.
2. For X, as before, nm^2 flops. No computation is necessary for C.
3. Tridiagonalization of a matrix of size N with only T needed: $4N^3/3$ flops.
4. Bisection adds $O(N)$ flops.

Therefore, the cost of the complete algorithm is at most $m^3/3 + m^2 n + 4(m+n)^3/3$ flops.

The essential difference between Algorithm 3.3.4 and Algorithm 3.5.2 is that we need the Cholesky factorization of M_1 in the former but only that of A in the latter.

There are two main reasons why we have not treated the Newton method in the fixed block case. The first is that in our numerical experiments for the general case presented in section 3.6, Newton's method performed significantly worse than both bisection and generalized eigenvalue method, and we expected the same to happen in the fixed block case. The second reason is that we have not found a way to efficiently exploit the block structure of the matrix $S(\alpha)$ in the Newton method, as we can for the other two. Namely, in each step we need the eigenvector corresponding to the smallest eigenvalue of $S(\alpha_k)$. This matrix changes with each step and it needs to be tridiagonalized but we have not been able to reuse the information from the previous steps or from some preprocessing of $S(\alpha)$.

3.5.3 Enforcing a lower bound on the smallest eigenvalue

In applications it may be required that a replacement correlation matrix is strictly positive definite. When the $(1,1)$ block A is positive definite we therefore generalize the problem to

$$\alpha_*^{\psi} = \min\{\, \alpha \in [0,1] \colon f(\alpha) \geq \psi = \theta\lambda_{\min}(A) \,\}, \tag{3.14}$$

where θ is a parameter. For $\theta = 0$ we have the original problem. To obtain an upper bound on the possible choices of θ we next show that the function f in the fixed block case attains its maximum value at $\alpha = 1$. Note that, since $S(1) = \mathrm{diag}(A, I)$ we have

$$f(1) = \lambda_{\min}\left(S(1)\right) = \min\{\lambda_{\min}(A), 1\} = \lambda_{\min}(A), \tag{3.15}$$

because $\lambda_{\min}(A) \leq 1$ by (3.4) for any symmetric matrix with unit diagonal.

Lemma 3.5.3. *For $S(\alpha)$ in (3.12) with A positive definite the function f defined by (3.3) is nondecreasing on $[0,1]$.*

Proof. Since $f(0) < 0$, $f(1) = \lambda_{\min}(A) > 0$, and f is concave and continuous, it is sufficient to show that for every $\alpha \in [0,1]$ we have $f(\alpha) \leq f(1)$, that is, $\max_{\alpha\in[0,1]} f(\alpha) = f(1)$.

From (3.12), since A is a leading principal submatrix of $S(\alpha)$, for every α we have, using (3.4), $\lambda_{\min}(A) \geq \lambda_{\min}\left(S(\alpha)\right) = f(\alpha)$. Since $f(1) = \lambda_{\min}(A)$ by (3.15), we have $f(\alpha) \leq f(1)$. □

From the proof we have $f(\alpha) \leq \lambda_{\min}(A)$. Therefore θ in (3.14) should be restricted to $[0,1]$ and hence $\psi \in [0, \lambda_{\min}(A)]$. Clearly, finding α_*^{ψ} is equivalent to finding the minimal α such that the matrix

$$S_\psi(\alpha) = S(\alpha) - \psi I \tag{3.16}$$

is positive semidefinite, and since $f_\psi(\alpha) = \lambda_{\min}\left(S_\psi(\alpha)\right) = f(\alpha) - \psi$ it follows that f_ψ has all the same properties as f: it is a concave and nondecreasing function on $[0,1]$.

With $A_\psi = A - \psi I$ and $B_\psi = B - \psi I$, we can write $S_\psi(\alpha)$ from (3.16) as

$$S_\psi(\alpha) = \begin{bmatrix} A_\psi & (1-\alpha)Y \\ (1-\alpha)Y^T & \alpha(1-\psi)I + (1-\alpha)B_\psi \end{bmatrix}.$$

For $\psi < \lambda_{\min}(A)$ the matrix A_ψ is positive definite and the methods from sections 3.5.1 and 3.5.2 for computing α_* in (3.11) can be applied to $S_\psi(\alpha)$ to compute α_*^ψ in (3.14).

The extreme case, when $\theta = 1$, significantly changes the nature of the problem. Here we are asking that α_* is such that $\lambda_{\min}(S(\alpha_*)) = \lambda_{\min}(A)$ and it follows that the matrix A_ψ is singular and positive semidefinite. In this case, $\alpha_* = 1$ might be the only solution or all values on the interval $[\alpha_*, 1]$, with $\alpha_* < 1$, might be solutions. These two cases are illustrated in plots (c) and (d) of Figure 3.2, with A there representing A_ψ.

In the next section we discuss the problems, both theoretical and computational, that arise from the singularity of the leading block in $S(\alpha)$.

3.5.4 Singular A

We now suppose that A in (3.9) is positive semidefinite and singular. For the bisection method, none of the computational savings discussed in section 3.5.1 are now applicable, since they were derived under the assumption that A is positive definite. We still have the basic Algorithm 3.3.1, but f may now be zero on an interval $[\alpha_*, 1]$ (see Figure 3.2(d)). As discussed in section 2.2.1, in this case we need to use the Cholesky factorization with complete pivoting and even then some difficulties remain.

The Newton method can still be performed in the case of an interval of zeros but convergence might no longer be quadratic if α_* has multiplicity greater than one. However, the method might fail when $\alpha = 1$ is the only solution and the function f is slowly increasing near that point, because then the computed iterates might leave the [0,1] bracket. Therefore the Newton method should be safeguarded.

The most severe problems arise in the generalized eigenvalue method. If A is singular then M_1 is singular and we no longer have a definite pencil; moreover, if f has infinitely many zeros then the pencil (3.8) is singular, which means that every α is a generalized eigenvalue and we cannot characterize α_* as before.

Our preferred way to handle the case of singular A is to employ a deflation method that reduces the problem to the nonsingular case. As a bonus, this analysis also allows us to distinguish the case when f has infinitely many zeros in $[0, 1]$ from the case when its only zero is 1.

Since A is singular and positive semidefinite it has the eigenvalue decomposition $A = QDQ^T$, where Q is orthogonal and $D = \mathrm{diag}(0, D_+)$, with D_+ a nonsingular

diagonal matrix of size $r = \operatorname{rank}(A)$, containing all the positive eigenvalues of A. Then

$$\begin{aligned} S(\alpha) &= \alpha \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} + (1-\alpha) \begin{bmatrix} A & Y \\ Y^T & B \end{bmatrix} \\ &= \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix} \left(\alpha \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} + (1-\alpha) \begin{bmatrix} D & Q^T Y \\ Y^T Q & B \end{bmatrix} \right) \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix}^T \\ &= \operatorname{diag}(Q, I) \widetilde{S}(\alpha) \operatorname{diag}(Q, I)^T, \end{aligned}$$

with

$$\begin{aligned} \widetilde{S}(\alpha) &= \alpha \left[\begin{array}{cc|c} 0 & 0 & \\ 0 & D_+ & 0 \\ \hline & 0 & I \end{array}\right] + (1-\alpha) \left[\begin{array}{cc|c} 0 & 0 & \\ 0 & D_+ & Q^T Y \\ \hline & Y^T Q & B \end{array}\right] \\ &= \left[\begin{array}{cc|c} 0 & 0 & \\ 0 & D_+ & (1-\alpha) Q^T Y \\ \hline & (1-\alpha) Y^T Q & \alpha I + (1-\alpha) B \end{array}\right]. \end{aligned}$$

A necessary condition for $\widetilde{S}(\alpha)$ to be positive semidefinite is that the first $m - r$ rows of $(1-\alpha)Q^T Y$ are zero. If the first $m - r$ rows of $Q^T Y$ are not zero then $\alpha_* = 1$. Otherwise, $\alpha_* < 1$ and α_* is the smallest α such that

$$\widetilde{S}_+(\alpha) = \begin{bmatrix} D_+ & (1-\alpha) Z \\ (1-\alpha) Z^T & \alpha I + (1-\alpha) B \end{bmatrix}$$

is positive semidefinite, where Z comprises the last r rows of $Q^T Y$. Since the leading (1,1) block of this matrix is now positive definite we can find α_* by either of the methods from sections 3.5.1 and 3.5.2 with $S(\alpha)$ replaced by $\widetilde{S}_+(\alpha)$.

Note that the condition that the first $m - r$ rows of $Q^T Y$ are zero means that each column of Y is in the column space of A.

3.5.5 Generalization to multiple fixed blocks

The problem of this section generalizes naturally to applications where a large correlation matrix needs to be constructed from blocks, as in the risk aggregation problem described in section 1.1. Shrinking can easily be used to solve this problem by choosing as target the matrix comprising the diagonal blocks of the large matrix:

$M_1 = \mathrm{diag}(A_{11}, A_{22}, \ldots, A_{kk})$. As in the case of keeping just one block fixed, α_* is characterized as the minimal elementwise relative change in the cross-correlations.

When $k = 2$ it is easy to show that the optimized bisection algorithm is a simple modification of Algorithm 3.5.1, where in step 6 the matrix T is now $T = A_{22} - (1 - \alpha_m)^2 Z$. For optimal efficiency the matrix should be reordered, if necessary, so that the larger of the two diagonal blocks is in the $(1,1)$ position. For the generalized eigenvalue method, Algorithm 3.3.4 leads to computing the smallest eigenvalue of

$$C = \begin{bmatrix} I & Z \\ Z^T & I \end{bmatrix},$$

where $Z = R_{11}^{-T} Y R_{22}^{-1}$ is formed by solving linear systems with the Cholesky factors R_{11} of A_{11} and R_{22} of A_{22}. Note that the required smallest eigenvalue of C is equal to $1 - \sigma_*$, where σ_* is the largest singular value of the matrix Z, so instead of computing the smallest eigenvalue of a matrix of order $m+n$ we can compute the largest singular value of an $m \times n$ matrix. For $k > 2$, the general algorithms from section 3.3 should be used.

When some of the diagonal blocks are singular, deflation analogous to that in section 3.5.4 can be done by employing the eigenvalue decomposition of each singular diagonal block.

3.6 Numerical experiments

We first compare the performance of our methods on a correlation matrix problem with a fixed block. We generate M_0 in (3.9) and M_1 in (3.10) by forming $A \in \mathbb{R}^{m \times m}$ using the MATLAB function `gallery('randcorr',m)` and the elements of the blocks $Y \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times n}$ are taken from the uniform distribution on $[-1, 1]$, with B symmetric and forced to have unit diagonal. The size $N = m + n$ of M_0 varies from 300 to 1500 and the test matrices are split into three groups. In the first group the matrices A and B (the diagonal blocks of M_0) are of the same size, in the second A is twice the size of B, and in the third B is twice the size of A. Unless specified otherwise, we use tolerance $\mathrm{tol} = 10^{-6}$, which is small enough for most practical applications.

We use the following algorithms.

1. `bisection`: Algorithm 3.3.1 with Cholesky factorization.

2. `bisection_fb`: Algorithm 3.5.1, the optimized bisection algorithm for the fixed block case.

3. `newton`: Algorithm 3.3.3. On each iteration the required eigenvector is computed by tridiagonalization followed by the bisection method (with the same tolerance as for the Newton iteration itself), using routines from the NAG Toolbox for MATLAB Mark 24 [82].

4. `GEP`: Algorithm 3.3.4, the algorithm based on solving a generalized eigenvalue problem. The tridiagonalization and bisection is again done using the NAG Toolbox.

5. `GEP_fb`: Algorithm 3.5.2, the optimized generalized eigenvalue problem algorithm for the fixed block case.

The computation times for the five methods averaged over 10 matrices of each size are presented in Table 3.2. The average number of steps for `newton` varies from 7 to 10 and the bisection methods always take 20 steps.

The experiments confirm the merit of using the optimized versions of bisection and the generalized eigenvalue method in applications where we keep a block fixed. Newton's method is the slowest of the three methods. `GEP` is a little faster than `bisection`, while `bisection_fb` is faster than `GEP_fb` for $m = n$ and $m = 2n$, and of similar speed for $n = 2m$.

To illustrate the effect of weighting, we consider an example with M_0 and W defined by

$$M_0 = \begin{bmatrix} 1.000 & 0.900 & 0.450 & 0.300 & 0.225 \\ 0.900 & 1.000 & 0.900 & 0.450 & 0.300 \\ 0.450 & 0.900 & 1.000 & 0.900 & 0.450 \\ 0.300 & 0.450 & 0.900 & 1.000 & 0.900 \\ 0.225 & 0.300 & 0.450 & 0.900 & 1.000 \end{bmatrix}, \quad W = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0.5 \\ 0 & 0 & 1 & 0.5 & 1 \end{bmatrix}.$$

The eigenvalues of M_0 are, to the digits shown, $-0.18, 0.05, 0.50, 1.27, 3.36$, therefore

Table 3.2: Computation times in seconds for the three general shrinking algorithms and the two algorithms optimized for the fixed block problem, for invalid correlation matrices of size $m + n$ with fixed leading block of size m.

(m, n)	`bisection`	`bisection_fb`	`GEP`	`GEP_fb`	`newton`
(150,150)	0.0069	0.0028	0.0039	0.0028	0.0194
(300,300)	0.0384	0.0091	0.0224	0.0147	0.1052
(450,450)	0.1029	0.0206	0.0642	0.0399	0.3055
(600,600)	0.2143	0.0435	0.1474	0.0895	0.6242
(750,750)	0.3835	0.0815	0.2913	0.1819	1.4204
(200,100)	0.0075	0.0017	0.0039	0.0031	0.0189
(400,200)	0.0405	0.0058	0.0230	0.0170	0.1215
(600,300)	0.1087	0.0121	0.0679	0.0472	0.3053
(800,400)	0.2381	0.0227	0.1571	0.1093	0.7699
(1000,500)	0.3848	0.0382	0.2744	0.1911	1.5115
(100,200)	0.0067	0.0043	0.0035	0.0025	0.0166
(200,400)	0.0308	0.0138	0.0180	0.0114	0.0827
(300,600)	0.0908	0.0367	0.0528	0.0313	0.2531
(400,800)	0.2068	0.0797	0.1304	0.0727	0.5445
(500,1000)	0.3306	0.1426	0.2667	0.1588	1.1879

M_0 is indefinite. We see that

$$M_1 = W \circ M_0 = \begin{bmatrix} 1.00 & 0.90 & 0.00 & 0.00 & 0.00 \\ 0.90 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.45 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.45 \\ 0.00 & 0.00 & 0.45 & 0.45 & 1.00 \end{bmatrix},$$

with eigenvalues $0.10, 0.36, 1.00, 1.64, 1.90$. Hence, M_1 is a valid target matrix. Using `bisection` we obtain $\alpha_* = 0.24$ and

$$S(\alpha_*) = \begin{bmatrix} \mathbf{1.000} & \mathbf{0.900} & 0.343 & 0.228 & 0.171 \\ \mathbf{0.900} & \mathbf{1.000} & 0.685 & 0.343 & 0.228 \\ 0.343 & 0.685 & \mathbf{1.000} & 0.685 & \mathbf{0.450} \\ 0.228 & 0.343 & 0.685 & \mathbf{1.000} & 0.793 \\ 0.171 & 0.228 & \mathbf{0.450} & 0.793 & \mathbf{1.000} \end{bmatrix},$$

with eigenvalues $0.00, 0.16, 0.52, 1.37, 2.95$. Note that the elements corresponding to weight $w_{ij} = 1$ (typeset in bold) are unchanged, as required by the interpretation of weights.

To explore this example further, denote the elements of M_0 and $S(\alpha_*)$ by s_{ij} and s'_{ij}, respectively. Then $s_{34} = 0.9 = s_{45}$, but since $w_{34} = 0$ and $w_{45} = 0.5$ the elements s'_{34} and s'_{45} are not the same. The relative change in the $(3,4)$ element is

$$\frac{s_{34} - s'_{34}}{s_{34}} = 0.24 = \alpha_* = \alpha_*(1 - w_{34}),$$

but that in the $(4,5)$ element is

$$\frac{s_{45} - s'_{45}}{s_{45}} = 0.12 = \frac{\alpha_*}{2} = \alpha_*(1 - w_{45}),$$

confirming that each element s'_{ij} in $S(\alpha_*)$ was obtained by multiplying the corresponding element s_{ij} in M_0 by $1 + \alpha_*(w_{ij} - 1)$.

In our next example, we use two large invalid correlation matrices cor1399 and cor3120 from our test set described in section 1.4. We also use three further matrices constructed as block 2×2 matrices with diagonal blocks cor1399 and cor1399, cor1399 and cor3120, and cor3120 and cor3120, with remaining off-diagonal elements from the random uniform distribution on $[-1, 1]$. We use the identity matrix as the target in the shrinking method, thus we are not fixing any off-diagonal elements. We compare the execution times of `bisection` and `GEP` with that for computation of the nearest correlation matrix (NCM) by NAG code `g02aa/nag_correg_corrmat_nearest` which implements a Newton method [18], [92]. Convergence tolerances of both 10^{-3} and 10^{-6} are taken for `bisection` and `g02aa`, as well as for the bisection part of `GEP`. The times are shown in Table 3.3, where N denotes the size of the matrix. The shrinking solution is computed one to two orders of magnitude faster than the NCM. It is clear that `g02aa` and `GEP` do not benefit significantly from a relaxed tolerance, whereas the time for `bisection` is proportional to the logarithm of the tolerance. The table also shows that the Frobenius norm distances from the original matrix to the shrinking solution range from being similar to the distance to the NCM to much larger than it.

Our final experiment provides some insight into how the Frobenius norm distance from the original matrix to the shrinking solution compares with the distance to the NCM when the smallest eigenvalue varies in size. We take for M_0 a random symmetric indefinite matrix with unit diagonal of size 500, constructed by a diagonal scaling of a random orthogonal similarity applied to a diagonal matrix D; the diagonal elements of D are generated from the uniform distribution on $[0, 1]$ and half of them are multiplied

Table 3.3: Times in seconds to compute shrinking solution by `bisection` and `GEP`, and nearest correlation matrix using `g02aa`, for tolerances 10^{-3} and 10^{-6}, and Frobenius norm distances to original matrix.

N	Time						Distance	
	`bisection`		`GEP`		`g02aa`		shrinking	NCM
	1e-3	1e-6	1e-3	1e-6	1e-3	1e-6		
1399	0.17	0.28	0.14	0.13	3.66	4.37	321.03	21.03
3120	1.01	2.22	2.44	2.46	28.08	34.31	178.71	5.44
2798	0.70	1.61	1.69	1.69	44.24	50.88	1221.21	1089.51
4519	2.29	4.99	8.10	8.00	220.88	234.68	1761.50	1631.52
6240	7.13	17.50	21.64	21.84	447.32	449.91	2578.10	2446.80

Table 3.4: Comparison of the distances in the Frobenius norm of the NCM and the solution computed by shrinking for matrices M_0 of size 500 with varying order of magnitude for the smallest eigenvalue.

$\lambda_{\min}(M_0)$	Avg. α_*	Distance		
		NCM	shrinking (max)	shrinking (I)
-4.6635e-1	6.0586e-1	5.5041e0	2.3506e1	1.0936e1
-4.2839e-2	9.7077e-2	5.0532e-1	3.5461e0	1.2367e0
-4.2466e-3	1.0144e-2	4.9173e-2	3.5748e-1	1.2255e-1
-4.0378e-4	9.4919e-4	4.6492e-3	3.3126e-2	1.1789e-2
-4.3278e-5	1.0900e-4	5.3498e-4	3.9736e-3	1.2897e-3
-4.0634e-6	1.0014e-5	4.5978e-5	3.4107e-4	1.3969e-4

by -10^{-p} for some p. We generate 10 random target correlation matrices M_1 using MATLAB function `gallery('randcorr',500)` and apply `bisection`. For each M_0, Table 3.4 shows the average shrinking parameter, the NCM distance, the maximum distance for the shrinking solution, and the distance for shrinking with $M_1 = I$. We see that the distance with $M_1 = I$ is smaller than the worst-case for the random targets M_1 and the shrinking distance is one order of magnitude larger than the NCM distance. This experiment gives some feel for the trade-off between the speed of shrinking versus the optimality of the NCM as measured by distance, at least for the case where there are no fixed elements.

It is clear from the last two experiments that in applications where it is not essential to compute the *nearest* correlation matrix, shrinking provides an attractive and much faster alternative for restoring definiteness.

Chapter 4

Bounds for the Distance to the Nearest Correlation Matrix

Be approximately right rather than exactly wrong.
—John W. Tukey

4.1 Introduction

Solving a matrix nearness problem defined by a distance function

$$d(A) = \min\{\, \|A - X\| : X \text{ has property } P \,\}$$

consists of the following tasks [53].

1. Derive an explicit formula or a practical characterization of $d(A)$.
2. Determine the minimizer $X_{\min}$ and whether it is unique.
3. Develop efficient algorithms for computing or estimating $d(A)$ and $X_{\min}$.

Let $A \in \mathbb{R}^{n\times n}$ be symmetric. For the nearest correlation matrix problem, an explicit formula is not known for the optimal distance $\mathrm{d}_{\mathrm{corr}}(A) = \|A - \mathrm{ncm}(A)\|_F$ nor for $\mathrm{ncm}(A)$. The main reason for this seems to be having a problem defined by both a basis independent property of positive semidefiniteness and a basis dependent property of a unit diagonal. A thorough analysis by Higham [56, Sec. 2] gives a characterization of the unique solution and presents the first globally convergent algorithm for computing it—the alternating projections method, presented here as Algorithm 5.3.1. This method is at best linearly convergent and requires an eigenvalue decomposition of a symmetric matrix on each iteration, so it costs at least $10n^3/3$ flops per iteration. The Newton algorithm developed by Qi and Sun [92] and improved by Borsdorf and

Higham [18] also requires an eigendecomposition of a symmetric matrix on each iteration and typically needs about 7 iterations. Hence, the total cost to compute $\mathrm{ncm}(A)$ by the Newton algorithm is at least $70n^3/3$ flops, which is computationally relatively expensive.

In view of efficient methods for computing $\mathrm{ncm}(A)$ being available, estimating the distance $\mathrm{d}_{\mathrm{corr}}(A)$ without computing $\mathrm{ncm}(A)$ has largely been overlooked. We first note that the iterates produced by the alternating projections method are not themselves correlation matrices as (with $\mathcal{P}$ denoting projection) the matrix $\mathcal{P}_{\mathcal{U}_n}(\mathcal{P}_{\mathcal{S}_n}(X))$ might be indefinite and the matrix $\mathcal{P}_{\mathcal{S}_n}(\mathcal{P}_{\mathcal{U}_n}(X))$ might not have an exactly unit diagonal. For the Newton method, the iterates do not satisfy the constraint of having a unit diagonal, as discussed in [18, Sec. 3.4]. Hence for both methods the iterates do not provide upper bounds on $\mathrm{d}_{\mathrm{corr}}(A)$. As the Newton method solves the dual problem of (1.1) [92], on each iteration the value of the dual function provides a lower bound for $\mathrm{d}_{\mathrm{corr}}$ [77]. Second, for practical purposes, determining the correct order of magnitude of $\mathrm{d}_{\mathrm{corr}}(A)$ is sufficient.

In this chapter we summarize the few existing bounds for $\mathrm{d}_{\mathrm{corr}}(A)$ and derive several new upper bounds. While the best bounds have a computational cost of $O(n^3)$ flops they are significantly less expensive to compute than $\mathrm{ncm}(A)$ itself.

Bounds on $\mathrm{d}_{\mathrm{corr}}(A)$ that can be easily evaluated using standard computational tools will certainly be of interest to practitioners, as illustrated by the discovery by Xu and Evers [129] that several matrices thought to be correlation matrices in the work of Tyagi and Das [117] actually had some negative eigenvalues. While attempting to compute the Cholesky factorization is sufficient to determine whether a matrix is positive semidefinite, we propose using a modified Cholesky factorization instead. The standard and modified Cholesky factorizations have the same computational cost, but for an indefinite matrix modified Cholesky factorization provides additional information that can be used to construct an upper bound on $\mathrm{d}_{\mathrm{corr}}(A)$; this bound can help the user to decide whether to revisit the construction of the matrix, perhaps by acquiring more data or by refining the statistical analysis. In our experiments, the best modified Cholesky bound is at most two orders of magnitude larger than $\mathrm{d}_{\mathrm{corr}}(A)$.

Sharper bounds are available based on spectral information. We present several bounds based only on the knowledge of the eigenvalues of A, but the best bound in this

class, which in our experiments is at most one order of magnitude larger than $\mathrm{d}_{\mathrm{corr}}(A)$, uses the nearest positive semidefinite matrix to A and so a knowledge of eigenvectors is also required.

The chapter is organized as follows. In section 4.2 we summarize existing upper and lower bounds on the distance to the nearest correlation matrix. In section 4.3 we derive our new upper bounds and give a result bounding the overestimation by a factor that does not exceed $1+n\sqrt{n}$. We present one result for the weighted Frobenius norm in section 4.4. We analyze the computational cost of the bounds in section 4.5. In section 4.6 we illustrate the quality of the bounds on our invalid correlation matrix test set listed in section 1.4.

4.2 Existing bounds

We first summarize currently available bounds for the distance to the nearest correlation matrix. We will need the following result on the nearest positive semidefinite matrix [52, Thm. 2.1].

Lemma 4.2.1 (Higham). *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with spectral decomposition $A = Q\Lambda Q^T$, where $Q = [q_1, \ldots, q_n]$ is orthogonal and $\Lambda = \mathrm{diag}(\lambda_i)$. Then the unique solution to $\min\{\, \|A - X\|_F : X$ is symmetric positive semidefinite$\,\}$ is*

$$A_+ = Q\,\mathrm{diag}(\max(\lambda_i, 0))Q^T. \tag{4.1}$$

We shall use $A_- = A - A_+ = Q\,\mathrm{diag}(\min(\lambda_i, 0))Q^T$.

The next result is [56, Lem. 1.1].

Lemma 4.2.2 (Higham). *For symmetric $A \in \mathbb{R}^{n\times n}$ with eigenvalues λ_i,*

$$\max\{\alpha_1, \alpha_2\} \le \mathrm{d}_{\mathrm{corr}}(A) \le \min\{\beta_1, \beta_2, \beta_3\},$$

where

$$\alpha_1^2 = \sum_{i=1}^{n}(a_{ii}-1)^2 + \sum_{\substack{|a_{ij}|>1 \\ i\neq j}} (1-|a_{ij}|)^2, \tag{4.2}$$

$$\alpha_2 = \|A - A_+\|_F = \left(\sum_{\lambda_i<0} \lambda_i^2\right)^{1/2}, \tag{4.3}$$

$$\beta_1 = \|A - I\|_F, \tag{4.4}$$

$$\beta_2 = \min\{\, \|A - zz^T\|_F : z_i = \pm 1,\ i = 1\colon n \,\}, \tag{4.5}$$

$$\beta_3 = \min_{-1\le\rho\le 1} \|A - T(\rho)\|_F, \quad \textit{where} \quad (T(\rho))_{ij} = \rho^{|i-j|}. \tag{4.6}$$

The lower bound α_1 follows from the fact that the elements of a correlation matrix are bounded in modulus by 1. The equivalence of the two formulae for α_2 is shown by Lemma 4.2.1. The upper bounds in Lemma 4.2.2 are obtained as the distance to certain classes of correlation matrices. In particular, β_3 arises from the matrices with (i,j) element $\rho^{|i-j|}$, known as Kac-Murdock-Szegő Toeplitz matrices [113], which are positive semidefinite for $-1 \le \rho \le 1$.

Travaglia [112, Prop. 3.1] obtained a further lower bound on $\mathrm{d}_{\mathrm{corr}}(A)$ using the circulant mean A_c, defined as the circulant matrix with first row $(c_0, c_1, \dots, c_{n-1})$, where

$$c_0 = \frac{1}{n}\operatorname{trace}(A),$$
$$c_k = \frac{1}{n}\left(\sum_{i=1}^{n-k} a_{i,i+k} + \sum_{i=1}^{k} a_{i,i+n-k}\right), \quad k = 1, 2, \dots, n-1.$$

This lower bound and a trivial upper bound are combined in the next result.

Lemma 4.2.3 (Travaglia)**.** *For symmetric $A \in \mathbb{R}^{n\times n}$,*

$$\mathrm{d}_{\mathrm{corr}}(A_c) \le \mathrm{d}_{\mathrm{corr}}(A) \le \mathrm{d}_{\mathrm{corr}}(A_c) + \|A - A_c\|_F. \tag{4.7}$$

4.3 New bounds

In this section we derive new upper bounds on the distance to the nearest correlation matrix that do not require the solution to a minimization problem, unlike the bounds β_2 and β_3 from Lemma 4.2.2, and the upper bound in Lemma 4.2.3. Our first bound

is the distance to the correlation matrix obtained by scaling A_+ from (4.1) to have unit diagonal.

Theorem 4.3.1. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with positive diagonal elements. Then*

$$\|A - A_+\|_F \le \mathrm{d_{corr}}(A) \le \|A - \widetilde{A}_+\|_F, \tag{4.8}$$

where $\widetilde{A}_+ = D^{-1/2}A_+D^{-1/2}$, with $D = \mathrm{diag}((A_+)_{ii})$.

Proof. The lower bound is α_2 in (4.3). The upper bound is immediate if we can show that $\widetilde{A}_+$ is a correlation matrix. The only question is whether it is defined, that is, whether the positive semidefinite matrix A_+ has positive diagonal elements, so that D is nonsingular and positive definite. From Lemma 4.2.1 we see that $A_+ - A = -A_-$ is positive semidefinite and it follows that the diagonal elements of A_+ are at least as large as the corresponding diagonal elements of A, and hence they are positive. □

In the next result we obtain an alternative upper bound that, while weaker than that in (4.8), is less expensive to compute, as we explain in section 4.5. Note that the theorem is valid for $t = n$, that is, for a positive semidefinite matrix A.

Theorem 4.3.2. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with positive diagonal elements and eigenvalues $\lambda_1 \ge \cdots \ge \lambda_t \ge 0 > \lambda_{t+1} \ge \cdots \ge \lambda_n$. Then*

$$\left(\sum_{i=t+1}^{n}\lambda_i^2\right)^{1/2} \le \mathrm{d_{corr}}(A) \le \left(\sum_{i=t+1}^{n}\lambda_i^2\right)^{1/2} + \theta\left(\sum_{i=1}^{t}\lambda_i^2\right)^{1/2}, \tag{4.9}$$

where

$$\theta = \max\left\{\left|1 - \frac{1}{\max_i a_{ii} - \min(\lambda_n, 0)}\right|, \left|1 - \frac{1}{\min_i a_{ii}}\right|\right\}.$$

Proof. The lower bound is (4.3).

By Theorem 4.3.1 and the triangle inequality we have

$$\mathrm{d_{corr}}(A) \le \|A - \widetilde{Y}\|_F \le \|A - Y\|_F + \|Y - \widetilde{Y}\|_F, \tag{4.10}$$

where $Y = A_+$ in (4.1) is positive semidefinite and $\widetilde{Y} = D^{-1/2}YD^{-1/2}$ is a correlation matrix, with $D = \mathrm{diag}(d_i)$ and $d_i = y_{ii} > 0$ for all i. We now bound $|y_{ij} - \widetilde{y}_{ij}|$.

Let $m = \min_i a_{ii}$ and $M = \max_i a_{ii}$. With e_i the ith unit vector and $\Lambda_- = \mathrm{diag}(\min(\lambda_i, 0))$, we have

$$d_i = e_i^TYe_i = e_i^T(A - Q\Lambda_-Q^T)e_i = a_{ii} - e_i^TQ\Lambda_-Q^Te_i = a_{ii} - \delta_i,$$

with $\delta_i = e_i^T Q \Lambda_- Q^T e_i$, and so

$$m - \delta_i \le d_i \le M - \delta_i.$$

With $y = Q^T e_i$, we have $\|y\|_2 = 1$, $\delta_i = y^T \Lambda_- y \le 0$, and

$$\min(\lambda_n, 0) = \min_{x \ne 0} \frac{x^T \Lambda_- x}{x^T x} \le \frac{y^T \Lambda_- y}{y^T y} = y^T \Lambda_- y = \delta_i.$$

Note that we must have $\min(\lambda_n, 0)$ for the first equality in the previous line to hold, since λ_n could be positive. It follows that $m \le d_i \le M - \min(\lambda_n, 0)$ for every i and so

$$\left(\frac{1}{M - \min(\lambda_n, 0)} \right)^{1/2} \le d_i^{-1/2} \le m^{-1/2}.$$

Since $\widetilde{y}_{ij} = d_i^{-1/2} d_j^{-1/2} y_{ij}$ we have

$$|y_{ij} - \widetilde{y}_{ij}| = \left|(1 - d_i^{-1/2} d_j^{-1/2}) y_{ij}\right| = \left|1 - d_i^{-1/2} d_j^{-1/2}\right| |y_{ij}|.$$

Finally, from

$$1 - \frac{1}{m} \le 1 - d_i^{-1/2} d_j^{-1/2} \le 1 - \frac{1}{M - \min(\lambda_n, 0)}$$

we have $|y_{ij} - \widetilde{y}_{ij}| \le \theta |y_{ij}|$ and therefore $\|Y - \widetilde{Y}\|_F \le \theta \|Y\|_F$. The upper bound in (4.9) then follows from (4.10). □

For $t = n$, Theorem 4.3.2 yields the following corollary, which quantifies the effect on the distance $\mathrm{d}_{\mathrm{corr}}(A)$ of the departure of the diagonal elements of a positive semidefinite matrix A from 1.

Corollary 4.3.3. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric positive semidefinite with positive diagonal elements. Then*

$$\mathrm{d}_{\mathrm{corr}}(A) \le \max\left\{ \left| 1 - \frac{1}{\max_i a_{ii}} \right|, \left| 1 - \frac{1}{\min_i a_{ii}} \right| \right\} \|A\|_F. \tag{4.11}$$

If all the diagonal elements of a positive semidefinite matrix A are at most 1 then $\mathrm{ncm}(A)$ is easy to compute directly as it is obtained from A by replacing each diagonal element by 1. However, (4.11) applies more generally.

In many applications the invalid approximation to a correlation matrix has unit diagonal and at least one negative eigenvalue. In this case Theorem 4.3.2 simplifies as follows.

Corollary 4.3.4. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with unit diagonal and eigenvalues $\lambda_1 \geq \cdots \geq \lambda_t \geq 0 > \lambda_{t+1} \geq \cdots \geq \lambda_n$, where $\lambda_n < 0$. Then*

$$\left(\sum_{i=t+1}^{n} \lambda_i^2\right)^{1/2} \leq \mathrm{d}_{\mathrm{corr}}(A) \leq \left(\sum_{i=t+1}^{n} \lambda_i^2\right)^{1/2} + \frac{|\lambda_n|}{1+|\lambda_n|}\left(\sum_{i=1}^{t} \lambda_i^2\right)^{1/2}. \tag{4.12}$$

The next result gives a sharper bound than (4.12). The proof uses the idea of shrinking from Chapter 3.

Theorem 4.3.5. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with unit diagonal and smallest eigenvalue $\lambda_n < 0$. Then*

$$\mathrm{d}_{\mathrm{corr}}(A) \leq \frac{|\lambda_n|}{1+|\lambda_n|}\|A - I\|_F, \tag{4.13}$$

and this bound is no larger than the upper bound in (4.12).

Proof. Let $S(\alpha) = \alpha I + (1-\alpha)A$, which has unit diagonal. We have $\mathrm{d}_{\mathrm{corr}}(A) \leq \|A - S(\alpha_*)\|_F$, where $\alpha_* = \min\{\alpha \in [0,1] : S(\alpha) \text{ is positive semidefinite}\}$. It is easy to see that $\alpha_* = -\lambda_n/(1-\lambda_n)$ and $A - S(\alpha_*) = \alpha_*(A - I)$, which gives (4.13).

Now we compare the bound with (4.12). The triangle inequality gives

$$\|A - I\|_F \leq \|A - A_+\|_F + \|A_+ - I\|_F, \tag{4.14}$$

where A_+ from (4.1) is the nearest positive semidefinite matrix to A. For the second term, we have

$$\|A_+ - I\|_F^2 = \operatorname{trace}\big((A_+ - I)^T(A_+ - I)\big) = \|A_+\|_F^2 - 2\operatorname{trace}(A_+) + n. \tag{4.15}$$

We noted in the proof of Theorem 4.3.1 that $A_+ - A$ is positive semidefinite, and since A has unit diagonal it follows that $\operatorname{trace}(A_+) \geq n$. Therefore, $-2\operatorname{trace}(A_+) + n \leq -n < 0$ and so $\|A_+ - I\|_F^2 \leq \|A_+\|_F^2$. Then from (4.14) it follows that $\|A - I\|_F \leq \|A - A_+\|_F + \|A_+\|_F$, so, since $\alpha_* \leq 1$,

$$\alpha_*\|A - I\|_F \leq \alpha_*\|A - A_+\|_F + \alpha_*\|A_+\|_F \leq \|A - A_+\|_F + \alpha_*\|A_+\|_F.$$

This completes the proof, since the right-hand side of the latter inequality is the upper bound in (4.12). ∎

Note that the bound (4.13) is also sharper than β_1 given in (4.4).

We now have several upper bounds and a natural question is "how sharp are they?" For the most practically important case of A with unit diagonal, the next result gives a limit on the overestimation for the upper bound of Theorem 4.3.1.

Theorem 4.3.6. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with unit diagonal, t nonnegative eigenvalues, largest eigenvalue λ_1, and smallest eigenvalue $\lambda_n < 0$. Then, in the notation of Theorem 4.3.1,*

$$\frac{\|A - \widetilde{A}_+\|_F}{\|A - A_+\|_F} \le 1 + \frac{\sqrt{t}\,\lambda_1}{1 + |\lambda_n|}. \tag{4.16}$$

If, in addition, $|a_{ij}| \le 1$ for $i \ne j$, then

$$\frac{\|A - \widetilde{A}_+\|_F}{\|A - A_+\|_F} \le 1 + n\sqrt{t}. \tag{4.17}$$

Proof. Using the triangle inequality and the relation $A = A_+ + A_-$ we have

$$\frac{\|A - \widetilde{A}_+\|_F}{\|A - A_+\|_F} = \frac{\|A_- + A_+ - \widetilde{A}_+\|_F}{\|A_-\|_F} \le 1 + \frac{\|A_+ - \widetilde{A}_+\|_F}{\|A_-\|_F}.$$

As in the proof of Theorem 4.3.2, we have $\|A_+ - \widetilde{A}_+\|_F \le \theta\|A_+\|_F$, where $\theta = |\lambda_n|/(1+|\lambda_n|)$, since A has unit diagonal and $\lambda_n < 0$. Therefore

$$\frac{\|A - \widetilde{A}_+\|_F}{\|A - A_+\|_F} \le 1 + \frac{|\lambda_n|}{1 + |\lambda_n|}\frac{\|A_+\|_F}{\|A_-\|_F}. \tag{4.18}$$

Since $\|A_+\|_F^2 = \sum_{i=1}^t \lambda_i^2 \le t\lambda_1^2$ and $\|A_-\|_F \ge \|A_-\|_2 = |\lambda_n|$, it follows that (noting that the assumptions of the theorem imply $\lambda_1 > 0$)

$$\frac{\|A_+\|_F}{\|A_-\|_F} \le \frac{\sqrt{t}\,\lambda_1}{|\lambda_n|}.$$

Substituting this bound into (4.18) yields (4.16). The bound (4.17) follows because $\lambda_1 \le n$ for any matrix with elements bounded in modulus by 1. □

It is easy to see that the upper bound (4.16) also holds for the ratio of the upper and lower bounds from Corollary 4.3.4, and hence also for the ratio of the shrinking bound (4.13) and (4.3), by Theorem 4.3.5. Moreover, we have $\|A - A_+\|_F \le \mathrm{d}_{\mathrm{corr}}(A) \le \psi\|A - A_+\|_F$, where $\psi = 1 + \sqrt{t}\lambda_1/(1 + |\lambda_n|)$.

Another way to obtain an upper bound on the distance to the nearest correlation matrix is to modify Theorem 4.3.1 by replacing the nearest positive semidefinite matrix A_+ by a more cheaply computable approximation to A_+. To construct such an approximation we will use modified Cholesky factorizations, described in section 2.2.2. Recall that they compute

$$P^T(A + E)P = LDL^T, \tag{4.19}$$

where P is a permutation matrix, L is unit lower triangular, and $A + E$ is positive semidefinite. The cost of these algorithms is the same as the cost of computing the

Cholesky factorization to highest order terms, which is substantially less than the cost of computing A_+. Note that bounds based on modified Cholesky factorizations provide an efficient way to determine whether the matrix is positive semidefinite to start with, as in this case $E = 0$.

Theorem 4.3.7. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with positive diagonal elements. Then*

$$\mathrm{d}_{\mathrm{corr}}(A) \le \|A - \widetilde{A}_{\mathrm{mc}}\|_F, \tag{4.20}$$

where $\widetilde{A}_{\mathrm{mc}} = D^{-1/2} A_{\mathrm{mc}} D^{-1/2}$ with $A_{\mathrm{mc}} = A + E$ from (4.19) and $D = \mathrm{diag}(A_{\mathrm{mc}})$.

As a final new upper bound on $\mathrm{d}_{\mathrm{corr}}(A)$ we make use of one of the rare explicitly known solutions to the nearest correlation matrix problem, for the so-called one parameter model. Here, a matrix $C(w) \in \mathbb{R}^{n\times n}$ is defined for a real parameter w as a unit diagonal matrix with all off-diagonal elements equal to w:

$$C(w) = (1-w)I + wee^T = I + w(ee^T - I),$$

where $e = [1, 1, \dots, 1]^T$. As shown in [19, Lem. 2.1] the matrix $C(w)$ is a correlation matrix if and only if $-1/(n-1) \le w \le 1$.

Theorem 4.3.8. *For $A \in \mathbb{R}^{n\times n}$ symmetric with $n \ge 2$,*

$$\begin{aligned} \mathrm{d}_{\mathrm{corr}}(A) &\le \min\{\, \|A - C(w)\|_F : C(w) \text{ is a correlation matrix}\,\} \\ &= \|A - C(w_{\mathrm{opt}})\|_F, \end{aligned} \tag{4.21}$$

where w_{opt} is the projection of $w = \left(e^T A e - \mathrm{trace}(A)\right)/(n^2 - n)$ onto the interval $[-1/(n-1), 1]$.

Proof. The equality (4.21) is from [19, Thm. 2.2]. □

4.4 Weighted Frobenius norm

In practical applications data is usually known with different levels of confidence which should be reflected in the replacement matrix—correlations in which we have more confidence should change less. This can be achieved to a certain extent by using the W-norm; recall that $\|A\|_W = \|W^{1/2} A W^{1/2}\|_F$, where W is symmetric positive definite and $W^{1/2}$ its unique positive definite square root.

When W is diagonal, and A has a unit diagonal and $\lambda_n < 0$, which is the most common practical case, we can prove the equivalent of Corollary 4.3.4. We shall need the following result.

Theorem 4.4.1 ([56, Thm. 3.2]). *For the W-norm and symmetric matrix A,*

$$\begin{aligned} &\operatorname{argmin}\{\, \|A - X\|_W : X \text{ is symmetric positive semidefinite}\,\} \\ &\qquad = W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_+ W^{-1/2}, \end{aligned} \tag{4.22}$$

where M_+ denotes the nearest positive semidefinite matrix to M in the Frobenius norm and is given by (4.1).

We can now prove the weighted version of Corollary 4.3.4.

Theorem 4.4.2. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with unit diagonal and eigenvalues $\lambda_1 \geq \cdots \geq \lambda_t \geq 0 > \lambda_{t+1} \geq \cdots \geq \lambda_n$, where $\lambda_n < 0$, let W be a diagonal positive definite matrix with the smallest eigenvalue $\lambda_{\min}(W)$, and let $W^{1/2}AW^{1/2}$ have eigenvalues $\mu_1 \geq \cdots \geq \mu_t \geq 0 > \mu_{t+1} \geq \cdots \geq \mu_n$. Then*

$$\begin{aligned} \|A - Y\|_W &\leq \min\{\, \|A - X\|_W : X \text{ is a correlation matrix}\,\} \\ &\leq \|A - Y\|_W + \frac{|\mu_n|}{\lambda_{\min}(W) + |\mu_n|}\, \|Y\|_W, \end{aligned}$$

where Y is the nearest positive semidefinite matrix to A in the W-norm defined in (4.22).

Proof. Since the distance to the nearest correlation matrix is at least as large as the distance to the nearest positive semidefinite matrix, the lower bound holds.

For the upper bound, we first transform Y into a correlation matrix $\widetilde{Y}$ by a diagonal scaling, i.e.

$$\widetilde{Y} = D^{-1/2} Y D^{-1/2},$$

where $D = \operatorname{diag}(d_i)$ and $d_i = y_{ii}$. Since $\min\{\, \|A-X\|_W : X \text{ is a correlation matrix}\,\} \leq \|A - \widetilde{Y}\|_W$ and $\|A - \widetilde{Y}\|_W \leq \|A - Y\|_W + \|Y - \widetilde{Y}\|_W$, our goal is to show that $\|Y - \widetilde{Y}\|_W \leq \theta \|Y\|_W$, for $\theta = |\mu_n|/\big(\lambda_{\min}(W) + |\mu_n|\big)$.

Note that

$$\begin{aligned} A - Y &= W^{-1/2}\left(W^{1/2}AW^{1/2} - (W^{1/2}AW^{1/2})_+\right) W^{-1/2} \\ &= W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2} \end{aligned} \tag{4.23}$$

is negative semidefinite. From (4.23) we have $Y = A - W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2}$ and, since A has unit diagonal, with e_i the ith unit vector,

$$d_i = a_{ii} - e_i^T W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2}e_i = 1 - \delta_i.$$

Since δ_i is a diagonal element of a negative semidefinite matrix (4.23), $\delta_i \leq 0$. We next show that $\mu_n/\lambda_{\min}(W) \leq \delta_i$ by proving that the matrix $W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2} - (\mu_n/\lambda_{\min}(W))I$ is positive semidefinite, and so its diagonal elements are nonnegative. The ith eigenvalue of this matrix is

$$\lambda_i\left(W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2}\right) - \mu_n/\lambda_{\min}(W).$$

By Ostrowski's theorem [62, Thm. 4.5.9],

$$\lambda_i\left(W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2}\right) = t_i\lambda_i\left(\left(W^{1/2}AW^{1/2}\right)_-\right),$$

where $t_i \in [\lambda_{\min}(W^{-1}), \lambda_{\max}(W^{-1})]$.

By the definition of $\left(W^{1/2}AW^{1/2}\right)_-$, the smallest eigenvalue of this matrix is μ_n and so

$$\begin{aligned}
&\lambda_i\left(W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2}\right) - \mu_n/\lambda_{\min}(W) = \\
&= t_i\lambda_i\left(\left(W^{1/2}AW^{1/2}\right)_-\right) - \mu_n/\lambda_{\min}(W) \\
&\geq t_i\mu_n - \mu_n/\lambda_{\min}(W) \\
&= (t_i - 1/\lambda_{\min}(W))\mu_n.
\end{aligned}$$

Since $\lambda_{\max}(W^{-1}) = 1/\lambda_{\min}(W)$ we have $t_i - 1/\lambda_{\min}(W) \leq 0$, which together with $\mu_n < 0$ shows that all eigenvalues of the matrix $W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_- W^{-1/2} - (\mu_n/\lambda_{\min}(W))I$ are nonnegative and hence it is a positive semidefinite matrix. Therefore, $\mu_n/\lambda_{\min}(W) \leq \delta_i \leq 0$.

We now have

$$1 \leq d_i \leq 1 + \frac{|\mu_n|}{\lambda_{\min}(W)} = \frac{\lambda_{\min}(W) + |\mu_n|}{\lambda_{\min}(W)},$$

and so

$$\left(\frac{\lambda_{\min}(W)}{\lambda_{\min}(W) + |\mu_n|}\right)^{1/2} \leq d_i^{-1/2} \leq 1. \tag{4.24}$$

From (4.22) and the definition of $\widetilde{Y}$ it follows that

$$\begin{aligned}\|Y-\widetilde{Y}\|_W &= \|Y-D^{-1/2}YD^{-1/2}\|_W \\ &= \left\|W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_+W^{-1/2}- \right. \\ &\qquad \left. - W^{-1/2}D^{-1/2}\left(W^{1/2}AW^{1/2}\right)_+D^{-1/2}W^{-1/2}\right\|_W \\ &= \left\|(W^{1/2}AW^{1/2})_+ - D^{-1/2}\left(W^{1/2}AW^{1/2}\right)_+D^{-1/2}\right\|_F \\ &= \|Z-D^{-1/2}ZD^{-1/2}\|_F,\end{aligned}$$

were we have used the fact that diagonal matrices commute in the second step.

Now,

$$\begin{aligned}\|Z-D^{-1/2}ZD^{-1/2}\|_F^2 &= \sum_{i,j}(z_{ij}-d_i^{-1/2}d_j^{-1/2}z_{ij})^2 \\ &= \sum_{i,j}(1-d_i^{-1/2}d_j^{-1/2})^2z_{ij}^2.\end{aligned}$$

By (4.24),

$$(1-d_i^{-1/2}d_j^{-1/2})^2 \le \left(1-\frac{\lambda_{\min}(W)}{\lambda_{\min}(W)+|\mu_n|}\right)^2 = \theta^2,$$

and so we have shown that

$$\|Y-\widetilde{Y}\|_W^2 \le \theta^2\sum_{i,j}z_{ij}^2 = \theta^2\|Z\|_F^2.$$

Finally,

$$\|Z\|_F^2 = \|(W^{1/2}AW^{1/2})_+\|_F^2 = \left\|W^{-1/2}\left(W^{1/2}AW^{1/2}\right)_+W^{-1/2}\right\|_W^2 = \|Y\|_W^2,$$

which completes the proof of the theorem. □

4.5 Computing the bounds

The main criteria for judging a bound are its cost and its accuracy. In this section we discuss the cost of the bounds presented above and in the next section we carry out numerical experiments to test their accuracy.

Table 4.1 summarizes the bounds, their applicability, and their cost. We will comment only on the nontrivial entries in the table.

We can evaluate the lower bound α_2 in (4.3) and the upper bounds in (4.9) and (4.12) without computing A_+ explicitly, but rather by computing all the positive eigenvalues or all the negative eigenvalues—whichever are fewer in number—and then using $\sum_{i=1}^{t} \lambda_i^2 + \sum_{i=t+1}^{n} \lambda_i^2 = \|A\|_F^2$. We can assume $t \geq n/2$ without loss of generality and therefore we compute the $n-t$ negative eigenvalues by tridiagonalizing A at the cost of $4n^3/3$ flops [44, p. 459] and then computing the $n-t$ negative eigenvalues of the tridiagonal matrix by the bisection method at a cost of $O(n(n-t))$ flops [8, p. 50], which makes the total cost for the bounds α_2, (4.9), and (4.12) at most $4n^3/3$ flops. The cost of (4.13) is the same.

As noted in [56], computing the upper bound β_2 from Lemma 4.2.2 is equivalent to maximizing $z^T A z$ over all vectors z with elements ± 1, which is an NP-hard problem [98]. For a matrix A of size n there are 2^n positive semidefinite matrices zz^T for such z, which makes an exhaustive search algorithm unfeasible unless n is very small.

A formula for the distance $\mathrm{d}_{\mathrm{corr}}(A_c)$ in Lemma 4.2.3 is given in [112, Thm. 4.1]. However, it requires not only all the eigenvalues but also their multiplicities, which are not reliably computable in floating point arithmetic. We therefore have to regard $\mathrm{d}_{\mathrm{corr}}(A_c)$ as no more easily computable in general than $\mathrm{d}_{\mathrm{corr}}(A)$, and so the bounds of Lemma 4.2.3 are of limited interest.

Next we turn to the bound (4.8), which requires $\widetilde{A}_+$, and hence A_+. Recall that we order the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_t \geq 0 > \lambda_{t+1} \geq \cdots \geq \lambda_n$, and assume without loss of generality that $t \geq n/2$ so that the majority of eigenvalues are nonnegative. We first compute the tridiagonalization $A = QTQ^T$ and do not form Q explicitly but keep it in factored form. By applying bisection and inverse iteration to the tridiagonal matrix T we compute $\lambda_{t+1}, \ldots, \lambda_n$ and the corresponding eigenvectors, which are placed in the columns of $Z = [z_{t+1}, \ldots, z_n]$. We then compute the matrix $W = Z\,\mathrm{diag}(|\lambda_{t+1}|, \ldots, |\lambda_n|)^{1/2} \in \mathbb{R}^{n\times(n-t)}$ and apply Q to get $B = QW$. Finally, $A_+ = A + BB^T$. The total cost is $4n^3/3$ flops for T, $O(n^2)$ flops to compute $\lambda_{t+1}, \ldots, \lambda_n$ and form Z and W, $2n^2(n-t)$ flops[1] to form B [44, Sec. 5.1.6], and $n^2(n-t)$ flops to form A_+, exploiting symmetry throughout. The total cost is therefore $4n^3/3 + 3n^2(n-t) \leq 4n^3/3 + 3n^3/2 = 17n^3/6$ flops.

[1] The errata web page http://www.cs.cornell.edu/cv/GVL4/Errata.htm for the fourth edition of [44] notes that the book incorrectly omits the leading 2 on page 238 from this operation count.

Table 4.1: Approximate cost in flops of the bounds for a symmetric $A \in \mathbb{R}^{n\times n}$. For the bound α_1, k is the number of elements $|a_{ij}| > 1$, $i \neq j$. For the bound (4.11), $m = \min_i a_{ii}$ and $M = \max_i a_{ii}$.

	Definition	Cost (flops)	Restrictions
Lower bounds			
α_1	(4.2)	$3(n+k)$	
$\alpha_2 = \|A - A_+\|_F$	(4.3)	$4n^3/3$	
$\mathrm{d}_{\mathrm{corr}}(A_c)$	(4.7)	As $\mathrm{d}_{\mathrm{corr}}(A)$	
Upper bounds			
$\beta_1 = \|A - I\|_F$	(4.4)	n^2	
$\beta_2 = \min\{\, \|A - zz^T\|_F : z_i = \pm 1 \,\}$	(4.5)	$O(n^2 2^n)$	
$\beta_3 = \min_{-1\le\rho\le 1} \|A - T(\rho)\|_F$	(4.6)	$O(n^2)$	
$\mathrm{d}_{\mathrm{corr}}(A_c) + \|A - A_c\|_F$	(4.7)	As $\mathrm{d}_{\mathrm{corr}}(A)$	
$\|A - \widetilde{A}_+\|_F$	(4.8)	$17n^3/6$	$a_{ii} > 0$
$\|A - A_+\|_F + \theta\|A_+\|_F$	(4.9)	$4n^3/3$	$a_{ii} > 0$
$\max\{\, \|1 - 1/M\|, \|1 - 1/m\| \,\}\|A\|_F$	(4.11)	n^2	$a_{ii} > 0$, $\lambda_n \ge 0$
$\|A - A_+\|_F + \|A_+\|_F\|\lambda_n\|/(1 + \|\lambda_n\|)$	(4.12)	$4n^3/3$	$\lambda_n < 0$, $a_{ii} \equiv 1$
$\|A - I\|_F\|\lambda_n\|/(1 + \|\lambda_n\|)$	(4.13)	$4n^3/3$	$\lambda_n < 0$, $a_{ii} \equiv 1$
$\|A - \widetilde{A}_{\mathrm{mc}}\|_F$	(4.20)	$2n^3/3$	$a_{ii} > 0$
$\|A - C(w_{\mathrm{opt}})\|_F$	(4.21)	$2n^2$	

Three different bounds are obtained from (4.20), corresponding to the three different modified Cholesky algorithms. While E in (4.19) is explicitly produced by the algorithms of Gill, Murray, and Wright, and Eskow and Schnabel, the algorithm of Cheng and Higham does not explicitly produce E, so this algorithm requires an extra matrix multiplication $L \cdot DL^T$. The cost stated in Table 4.1 includes the latter step.

In [56] an approximation for β_3 from Lemma 4.2.2 was computed as the approximate local minimum obtained with the MATLAB `fminbnd` minimizer. We propose an alternative. Note that the function we are minimizing for the bound β_3 is a polynomial in the variable ρ:

$$f(\rho) = \|A - T(\rho)\|_F^2 = 2 \sum_{1\le i<j\le n} (a_{ij} - \rho^{j-i})^2 + \sum_{i=1}^{n} (a_{ii} - 1)^2.$$

We compute the stationary points of f, that is, the zeros of the derivative

$$f'(\rho) = -4 \sum_{1\le i<j\le n} \left[(j-i) a_{ij} \rho^{j-i-1} - (j-i)\rho^{2(j-i)-1} \right],$$

which has degree $2n - 3$. Then we obtain β_3 as the minimum value of f over all stationary points in $[-1, 1]$ along with the endpoints ± 1. The dominant cost for

this bound is computing the stationary points, which are the real eigenvalues of a companion matrix of order $2n-3$ in $[-1,1]$; these can be computed in $O(n^2)$ flops [7].

To summarize, we can separate the new upper bounds into three main categories. The most expensive bound to compute is (4.8) and it uses $\widetilde{A}_+$; the less expensive bounds (4.9), (4.12), and (4.13) are based on the knowledge of eigenvalues only; and the least expensive bound is the modified Cholesky bound (4.20), which has half the cost of the eigenvalue-only based bounds.

4.6 Numerical experiments

In this section we analyze the accuracy of the bounds on our invalid correlation matrix test set from section 1.4. The nearest correlation matrix required to determine the true distance $\mathrm{d}_{\mathrm{corr}}(A)$ is computed by the code `nag_correg_corrmat_nearest (g02aa)` from the NAG Toolbox for MATLAB Mark 24 [82], which implements the preconditioned Newton algorithm of [18], and we chose tolerance $\mathrm{tol} = 10^{-10}$.

In our first test we analyze the performance of the modified Cholesky algorithms used for the bound (4.20) for all our test matrices. In Table 4.2 the matrix $\widetilde{A}_{\mathrm{mc}}$ from (4.20) corresponding to the algorithms of Gill, Murray, and Wright [42, Sec. 4.4.2.2], Schnabel and Eskow [102] and [103], and Cheng and Higham [24] is denoted by GMW, SE90, SE99, and CH, respectively. The results show two main features. First, the four modified Cholesky algorithms provide bounds of similar quality for all but the RiskMetrics matrices, and these bounds are often of the correct order of magnitude. Second, for all the RiskMetrics matrices except RiskMetrics4, $\mathrm{d}_{\mathrm{corr}}(A)$ is relatively small and the revised Schnabel and Eskow [103] algorithm and the Cheng and Higham algorithm provide bounds three or four orders of magnitude smaller than those from the other two algorithms. Since the Cheng and Higham algorithm gives the best bounds overall, we use it in the remaining experiments.

We next compute all our bounds. The results are given in Table 4.3 and Table 4.4. The ordering of the bounds is the same as in Table 4.1, but note that we exclude the bound (4.12) as for our test matrices it is the same as (4.9). The bound α_1 is zero for all examples where $a_{ii} \equiv 1$ and $|a_{ij}| \leq 1$. Moreover, the circulant mean A_c of several of these matrices turns out to be a correlation matrix and so $\mathrm{d}_{\mathrm{corr}}(A_c) = 0$ in (4.7).

Table 4.2: Upper bound (4.20) from the modified Cholesky algorithms.

Ex.	$\|A - \mathrm{GMW}\|_F$	$\|A - \mathrm{SE90}\|_F$	$\|A - \mathrm{SE99}\|_F$	$\|A - \mathrm{CH}\|_F$	$\mathrm{d}_{\mathrm{corr}}(A)$
high02	8.45e-1	6.19e-1	6.19e-1	5.86e-1	5.28e-1
tec03	8.17e-2	9.47e-1	6.33e-2	5.19e-2	3.74e-2
bhwi01	6.31e-1	2.50e-1	2.50e-1	4.30e-1	1.51e-1
mmb13	3.13e1	3.14e1	3.14e1	3.04e1	3.03e1
fing97	1.50e-1	7.66e-2	7.90e-2	9.24e-2	4.91e-2
tyda99R1	2.18	2.35	2.17	2.36	1.40
tyda99R2	1.53	2.19	1.57	1.71	7.75e-1
tyda99R3	1.47	1.46	1.49	1.09	6.72e-1
usgs13	9.69e-1	8.02e-1	5.93e-1	1.92	5.51e-2
RiskMetrics1	1.11e2	1.12e1	1.72e-2	8.81e-3	3.88e-5
RiskMetrics2	1.38e2	6.21	1.90e-2	9.71e-3	4.75e-5
RiskMetrics3	9.33e1	3.14	7.10e-3	4.20e-3	1.81e-5
RiskMetrics4	1.27e2	6.66e1	6.62e1	1.22	8.40e-2
RiskMetrics5	1.32e2	1.44	2.01e-2	9.64e-3	4.46e-5
RiskMetrics6	8.53e1	1.30	5.85e-3	2.85e-3	1.59e-5
cor1399	3.59e2	3.57e2	3.57e2	4.52e1	2.10e1
cor3120	7.83e1	4.18e2	4.16e2	4.40e2	5.44

Table 4.3: Small examples.

	high02	tec03	bhwi01	mmb13	fing97	tyda99R1	tyda99R2	tyda99R3
	Lower bounds							
(4.2)	0.00	0.00	0.00	3.01e1	0.00	0.00	0.00	0.00
(4.3)	4.14e-1	2.78e-2	1.28e-1	2.15e1	3.83e-2	1.15	6.24e-1	5.59e-1
(4.7)	0.00	0.00	0.00	1.17e1	0.00	1.60e-1	0.00	0.00
	True distance							
$\mathrm{d}_{\mathrm{corr}}(A)$	5.28e-1	3.74e-2	1.51e-1	3.03e1	4.91e-2	1.40	7.75e-1	6.72e-1
	Upper bounds							
(4.4)	2.00	2.35	2.43	3.29e1	3.09	4.02	4.02	3.74
(4.6)	9.15e-1	2.03	2.21	3.04e1	2.32	3.98	2.81	3.73
(4.7)	1.15	2.08	2.35	3.98e1	2.50	3.24	2.11	3.28
(4.8)	5.38e-1	3.93e-2	1.61e-1	3.04e1	5.33e-2	1.45	8.41e-1	7.02e-1
(4.9)	1.18	1.11e-1	5.00e-1	4.54e1	1.88e-1	3.55	2.39	2.11
(4.13)	5.86e-1	6.35e-2	2.75e-1	3.14e1	1.14e-1	2.02	1.46	1.25
(4.20)	5.86e-1	5.19e-2	4.30e-1	3.04e1	9.24e-2	2.36	1.71	1.09
(4.21)	1.15	2.08	2.35	3.04e1	2.60	3.71	2.20	3.70

Table 4.4: Real-life examples.

	usgs13	RiskMetrics1	RiskMetrics2	RiskMetrics3	RiskMetrics4	RiskMetrics5	RiskMetrics6	cor1399	cor3120
					Lower bounds				
(4.2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.36e-1	1.06e-1
(4.3)	5.02e-2	3.39e-5	3.97e-5	1.63e-5	8.25e-2	3.76e-5	1.46e-5	1.32e1	2.29
(4.7)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
					True distance				
$\mathrm{d}_{\mathrm{corr}}(A)$	5.51e-2	3.88e-5	4.75e-5	1.81e-5	8.40e-2	4.46e-5	1.59e-5	2.10e1	5.44
					Upper bounds				
(4.4)	2.29e1	1.28e2	1.54e2	1.45e2	1.30e2	1.45e2	1.41e2	3.59e2	4.28e2
(4.6)	2.04e1	1.20e2	1.42e2	1.33e2	1.18e2	1.33e2	1.28e2	3.58e2	4.28e2
(4.7)	6.75	1.10e2	1.33e2	1.23e2	1.04e2	1.22e2	1.15e2	2.05e2	2.92e2
(4.8)	6.55e-2	1.19e-4	1.90e-4	5.88e-5	9.21e-2	1.71e-4	4.83e-5	2.37e1	1.11e1
(4.9)	1.15	1.00e-3	1.27e-3	7.35e-4	1.01e1	1.16e-3	9.98e-4	3.36e2	1.83e2
(4.13)	1.01	9.58e-4	1.22e-3	7.12e-4	9.91	1.11e-3	9.74e-4	3.21e2	1.79e2
(4.20)	1.92	8.81e-3	9.71e-3	4.20e-3	1.22	9.64e-3	2.85e-3	4.52e1	4.40e2
(4.21)	7.64	1.16e2	1.43e2	1.32e2	1.10e2	1.28e2	1.21e2	2.06e2	2.93e2

Several observations can be made about the results.

1. Of the lower bounds, only (4.3) provides useful information. Moreover, in all examples this bound is within a factor 2.4 of $\mathrm{d}_{\mathrm{corr}}$ (the worst case being cor3120).

2. Of the upper bounds, (4.8)—the most expensive bound to compute—is the most accurate and is always within a factor 4 of $\mathrm{d}_{\mathrm{corr}}$ (the worst case being RiskMetrics2).

3. Over all the test matrices, the upper bound (4.8) exceeded the lower bound (4.3) by at most a factor 4.9 (the worst case being cor3120).

4. Of the other eigenvalue-based upper bounds, the bound from shrinking (4.13) is better than the bound (4.9), as we already know from Theorem 4.3.5. The shrinking bound (4.13) is typically an order of magnitude larger than (4.8) on real-life examples.

5. The upper bounds (4.13) and (4.20) based on shrinking and the modified Cholesky factorizations, respectively, are of similar quality and they overestimate $\mathrm{d}_{\mathrm{corr}}$ at most by one or two orders of magnitude. The modified Cholesky bound has the advantage of being computable in half the number of operations as the bound based on shrinking.

6. The upper bounds (4.4), (4.6), and (4.21), which are computable in $O(n^2)$ operations, are poor in these tests, the more so when the distance is small.

Chapter 5

Anderson Acceleration of the Alternating Projections Method for Computing the Nearest Correlation Matrix

Q: How many numerical mathematicians does it take to replace a light bulb?

A: 3.9967 (after 9 iterations).

5.1 Introduction

In this chapter we revisit the perhaps most widely used method for computing the nearest correlation matrix–Higham's alternating projections method [56]. Major reasons for its popularity are its ease of coding and the availability of implementations in MATLAB, Python, R, and SAS [59]. As well as being easy to understand and easy to implement, the alternating projections method has the attractive feature that it is easily modified to incorporate additional constraints on the matrix, in particular to fix certain elements or to compute a strictly positive definite solution with a lower bound on the smallest eigenvalue. Since its rate of convergence is at best linear, the method can potentially be very slow. The aim of this work is to reduce the number of iterations required.

We attempt to accelerate the alternating projections method by employing Anderson acceleration [3], [84, Sec. 1.1.4] also known as Anderson mixing, which is designed for fixed-point problems. While fixed-point iteration uses only the current, kth, iterate to define the next one, Anderson acceleration uses the additional information from the

m_k previous iterations and computes the new iterate as a specific linear combination of these $m_k + 1$ quantities. The selected history length m_k is usually small. A discussion that puts Anderson acceleration in context with other acceleration methods can be found in [122].

In quantum chemistry Anderson acceleration is known as Pulay mixing or direct inversion in the iterative subspace (DIIS) [91] and it has been widely used in electronic structure computations; see [99] and the references therein. Anderson acceleration is related to multisecant methods (extensions of quasi-Newton methods involving multiple secant conditions); in fact, Eyert [35] proves that it is equivalent to the so-called "bad" Broyden's method [23], [68], and a similar analysis is done by Fang and Saad [37] and Rohwedder and Schneider [99]. For linear systems, if $m_k = k$ for each k then Anderson acceleration is essentially equivalent to the generalized minimal residual (GMRES) method [100], as shown by Potra and Engler [89], Rohwedder and Schneider [99], and Walker and Ni [122]. For nonlinear problems Rohwedder and Schneider [99] show that Anderson acceleration is locally linearly convergent under certain conditions. Adding to the above convergence analysis is the recent work by Toth and Kelley [111] concerning Anderson acceleration with $m_k = \min(m, k)$, for a fixed m, applied to contractive mappings.

Even though there are no general guarantees of its convergence, Anderson acceleration has a successful record of use in electronic structure computations. Furthermore, it significantly improved the performance of several domain decomposition methods presented in [122] and has proved to be very efficient on various examples in the above references. Hence Anderson acceleration has great potential for enhancing the convergence of the alternating projections method for the nearest correlation matrix.

Recently, López and Raydan [75] have proposed a geometrically-based acceleration scheme for the alternating projections method that builds a new sequence from the original one by taking linear combinations of successive pairs of iterates. The new sequence is tested for convergence and the original iteration remains unchanged. We compare this method with Anderson acceleration in section 5.4 (Experiment 9).

The rest of this chapter is organized as follows. We present the Anderson acceleration scheme in section 5.2. In section 5.3 we recall the necessary results on the alternating projections method with Dykstra's correction for computing the nearest

correlation matrix and the problem variants in which some elements remain fixed or the smallest eigenvalue of the solution must be above a given threshold, and we explain how to apply Anderson acceleration to these problems. Numerical experiments presented in section 5.4 show that Anderson acceleration at least halves the number of iterations required by the alternating projections method for the nearest correlation matrix problem, which results in a significant reduction in computation time for large problems. The experiments also show that even greater improvements can be achieved for the problem variants, which is especially important for the fixed elements constraint since in this case there is no available Newton method.

5.2 Anderson acceleration for fixed-point iteration

A basic method for the solution of the fixed-point problem $g(x) = x$ for $g\colon \mathbb{R}^n \to \mathbb{R}^n$ is fixed-point iteration, also known as the (nonlinear) Richardson iteration, Picard iteration, or the method of successive substitution. It has the form

$$x_{k+1} = g(x_k), \quad k \geq 1, \qquad x_0 \in \mathbb{R}^n \text{ given.} \tag{5.1}$$

To guarantee convergence of (5.1) assumptions must be made on the function g and the starting vector x_0, and in general convergence is at a linear rate [63, Chap. 4.2]. A method that attempts to encourage or accelerate convergence is Anderson acceleration, which redefines x_{k+1} to make use of the information from the m_k previous steps. We first briefly outline the original method derived by Anderson [3].

Algorithm 5.2.1 (Original Anderson acceleration)**.** *Given $x_0 \in \mathbb{R}^n$ and an integer $m \geq 1$ this algorithm produces a sequence x_k of iterates intended to converge to a fixed point of the function $g\colon \mathbb{R}^n \to \mathbb{R}^n$.*

1 $x_1 = g(x_0)$
2 for $k = 1, 2, \ldots$ until convergence
3 $\quad m_k = \min(m, k)$
4 $\quad$ Determine $\theta^{(k)} = (\theta_1^{(k)}, \ldots, \theta_{m_k}^{(k)})^T \in \mathbb{R}^{m_k}$ that minimizes $\|u_k - v_k\|_2^2$, where
$$u_k = x_k + \sum_{j=1}^{m_k} \theta_j (x_{k-j} - x_k), \quad v_k = g(x_k) + \sum_{j=1}^{m_k} \theta_j \big(g(x_{k-j}) - g(x_k)\big).$$
5 $\quad$ Set $x_{k+1} = v_k$ using the parameters from $\theta^{(k)}$.

6 end

In [3] the final step is $x_{k+1} = u_k + \beta_k(v_k - u_k)$, where u_k and v_k are defined from the computed $\theta^{(k)}$, and $\beta_k > 0$ is empirically determined. The usual choice in the literature is $\beta_k \equiv 1$, which we use here. We have also taken the history length parameter m_k to be fixed, at m, once the first m iterations have been taken.

We can give some insight into Algorithm 5.2.1 by writing

$$u_k = \left(1 - \sum_{j=1}^{m_k} \theta_j^{(k)}\right) x_k + \sum_{j=1}^{m_k} \theta_j^{(k)} x_{k-j} = \sum_{j=0}^{m_k} w_j x_{k-j},$$
$$v_k = \left(1 - \sum_{j=1}^{m_k} \theta_j^{(k)}\right) g(x_k) + \sum_{j=1}^{m_k} \theta_j^{(k)} g(x_{k-j}) = \sum_{j=0}^{m_k} w_j g(x_{k-j}),$$

where $\sum_{j=0}^{m_k} w_j = 1$. Algorithm 5.2.1 minimizes $\|u_k - v_k\|_2^2$ subject to $\sum_{j=0}^{m_k} w_j = 1$. If g is linear then the objective function is $\|u_k - g(u_k)\|_2^2$ and so $v_k = g(u_k)$ is the vector from the affine subspace spanned by the current iterate and the previous m_k iterates that minimizes the residual of the fixed-point equation.

We will use an equivalent form of the method that stores in two matrices the differences of the successive iterates and their function values. These matrices are related by simple update formulae that can be exploited for an efficient implementation. This variant is given by Fang and Saad [37], Plasse [87], Walker [121], and Walker and Ni [122]. Here, Anderson acceleration is applied to the equivalent problem $f(x) = 0$, where $f(x) = g(x) - x$, instead of the fixed-point problem $g(x) = x$.

Algorithm 5.2.2 (Anderson acceleration)**.** *Given $x_0 \in \mathbb{R}^n$ and an integer $m \geq 1$ this algorithm produces a sequence x_k of iterates intended to converge to a zero of the function $f\colon \mathbb{R}^n \to \mathbb{R}^n$. The following notation is used: $m_k = \min(m, k)$, $\Delta x_i = x_{i+1} - x_i$, $\mathcal{X}_k = \begin{bmatrix} \Delta x_{k-m_k} & \dots & \Delta x_{k-1} \end{bmatrix}$, $f_i = f(x_i)$, $\Delta f_i = f_{i+1} - f_i$, and $\mathcal{F}_k = \begin{bmatrix} \Delta f_{k-m_k} & \dots & \Delta f_{k-1} \end{bmatrix}$.*

1 $x_1 = x_0 + f(x_0)$

2 for $k = 1, 2, \dots$ until convergence

3 $\quad m_k = \min(m, k)$

4 $\quad$ Compute $\gamma^{(k)} = (\gamma_{k-m_k}^{(k)}, \dots, \gamma_{k-1}^{(k)})^T \in \mathbb{R}^{m_k}$ that solves $\min_{\gamma \in \mathbb{R}^{m_k}} \|f_k - \mathcal{F}_k \gamma\|_2^2$.

5 $\quad \bar{x}_k = x_k - \sum_{i=k-m_k}^{k-1} \gamma_i^{(k)} \Delta x_i = x_k - \mathcal{X}_k \gamma^{(k)}$

6 $\quad \bar{f}_k = f_k - \sum_{i=k-m_k}^{k-1} \gamma_i^{(k)} \Delta f_i = f_k - \mathcal{F}_k \gamma^{(k)}$

7 $\quad x_{k+1} = \bar{x}_k + \bar{f}_k$

8 end

Line 4 of Algorithm 5.2.2 consists of the following major computations. We assume that $\mathcal{F}_k$ has full rank and that the least-squares problem is solved using a QR factorization of $\mathcal{F}_k$.

1. Compute $f_k = f(x_k)$.
2. Obtain a QR factorization of $\mathcal{F}_k$ from that of $\mathcal{F}_{k-1}$. Since $\mathcal{F}_k$ is just $\mathcal{F}_{k-1}$ with the first column removed (for $k > m$) and a new last column added this is a QR factorization updating problem.
3. Solve the least-squares problem using the QR factorization.

Assume that $k > m$. Since $\mathcal{F}_{k-1}$ is $n \times m$ and its first column is removed in passing to $\mathcal{F}_k$, to update the R factor we need $m^2/2$ flops and to update Q an additional $6mn$ flops [48, p. 28]. Updating the QR factors after the last column has been added to the matrix costs $4mn + 3n$ flops [48, Sec. 2.5.1]. Hence the total cost for step 2 above is at most $m^2/2 + 10mn + 3n$ flops. The cost of step 3 (which forms and solves by back substitution a triangular system involving R) is $2mn + m^2$ flops. Therefore, Anderson acceleration takes an additional $3m^2/2 + 12mn + 3n$ flops per step compared with the unaccelerated iteration.

More details of the updating scheme, as well as a strategy that removes more than one leading column of $\mathcal{F}_k$, if necessary, in order to ensure that the matrix R does not become too ill-conditioned are given in [121], [122, Sec. 4].

5.3 Accelerating the alternating projections method for the nearest correlation matrix

We now summarize the method to which we wish to apply Anderson acceleration: the alternating projections method for computing the nearest correlation matrix in the Frobenius norm. In its basic form the alternating projections method attempts

to find a point in the intersection of two closed subspaces that is nearest to a given point by iteratively projecting onto each subspace. This simple idea is motivated by the fact that it is often easier to compute the individual projections onto the given subspaces than the projection onto their intersection. A detailed exposition of the origins and generalizations of alternating projections methods is given by Escalante and Raydan [34].

Let A be a given symmetric matrix of order n and define the sets

$$\mathcal{S}_n = \{\, X \in \mathbb{R}^{n\times n} : X \text{ is symmetric positive semidefinite} \,\}, \tag{5.2}$$

$$\mathcal{U}_n = \{\, X = X^T \in \mathbb{R}^{n\times n} \colon x_{ii} = 1,\ i = 1\colon n \,\}. \tag{5.3}$$

For the nearest correlation matrix problem, we are looking for the closest matrix to A that lies in the intersection of $\mathcal{S}_n$ and $\mathcal{U}_n$. Since these are convex sets rather than subspaces the alternating projections method has to be used in a modified form proposed by Dykstra [33], in which each projection incorporates a correction; each correction can be interpreted as a normal vector to the corresponding convex set. This correction is not needed for a translate of a subspace [20], so is only required for the projection onto $\mathcal{S}_n$.

Denote the projections of a symmetric matrix X onto $\mathcal{S}_n$ and $\mathcal{U}_n$ by $\mathcal{P}_{\mathcal{S}_n}(X)$ and $\mathcal{P}_{\mathcal{U}_n}(X)$, respectively. We have used the projection $\mathcal{P}_{\mathcal{S}_n}(X)$ in Chapter 4; Lemma 4.2.1 shows that it is computed from an eigenvalue decomposition of X (see also Theorem 5.3.4 below). The projection $\mathcal{P}_{\mathcal{U}_n}(X)$ is obtained by setting the diagonal elements of X to 1.

The use of alternating projections for computing the nearest correlation matrix was proposed by Higham [56, Alg. 3.3] in the following form.

Algorithm 5.3.1. *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ this algorithm computes the nearest correlation matrix Y to A by alternating projections. It requires a convergence tolerance* tol.

$\Delta S_0 = 0$, $Y_0 = A$
for $k = 1, 2, \ldots$
    $R_k = Y_{k-1} - \Delta S_{k-1}$
    $X_k = \mathcal{P}_{\mathcal{S}_n}(R_k)$ % Project onto $\mathcal{S}_n$.

5 $\Delta S_k = X_k - R_k$ % Dykstra's correction.
6 $Y_k = \mathcal{P}_{\mathcal{U}_n}(X_k)$ % Project onto $\mathcal{U}_n$.
7 if $\|Y_k - X_k\|_F \le \mathrm{tol}\,\|Y_k\|_F$, $Y = Y_k$, quit, end
8 end

It is known that X_k and Y_k both converge to the nearest correlation matrix as $k \to \infty$, with a convergence rate that is linear at best [56]. The termination criterion on line 7 is a simplification of the criterion

$$\max\left\{ \frac{\|X_k - X_{k-1}\|_F}{\|X_k\|_F}, \frac{\|Y_k - Y_{k-1}\|_F}{\|Y_k\|_F}, \frac{\|Y_k - X_k\|_F}{\|Y_k\|_F} \right\} \le \mathrm{tol} \tag{5.4}$$

used by Higham [56], who notes that the three terms inside the max are usually of similar size. We use only the last term, since the test on line 7 is equivalent to the robust stopping criterion for Dykstra's algorithm proposed by Birgin and Raydan [14] and this choice works well in all our experiments.

Aitken extrapolation (see, for example, [22]) cannot be used to accelerate Algorithm 5.3.1 because it requires the underlying sequence to be linearly convergent, which is not guaranteed here. We therefore turn to Anderson acceleration. To use it we must recast Algorithm 5.3.1 as a fixed-point method, that is, define the function g for the iteration (5.1). We do this as follows, noting that two matrices are recurred: Y_k and ΔS_k, while X_k is only used for the convergence test.

Algorithm 5.3.2 (Fixed-point form of Algorithm 5.3.1)**.** *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ this algorithm computes the nearest correlation matrix Y to A. It requires a convergence tolerance* tol*.*

1 $\Delta S_0 = 0$, $Y_0 = A$
2 for $k = 1, 2, \ldots$
3 $[X_k, Y_k, \Delta S_k] = g(Y_{k-1}, \Delta S_{k-1})$
4 if $\|Y_k - X_k\|_F \le \mathrm{tol}\,\|Y_k\|_F$, $Y = Y_k$, quit, end
5 end

where the computation of $[X_k, Y_k, \Delta S_k] = g(Y_{k-1}, \Delta S_{k-1})$ is effected by

6 $R_k = Y_{k-1} - \Delta S_{k-1}$

7 $X_k = \mathcal{P}_{\mathcal{S}_n}(R_k)$

8 $\Delta S_k = X_k - R_k$

9 $Y_k = \mathcal{P}_{\mathcal{U}_n}(X_k)$

To apply Anderson acceleration (Algorithm 5.2.2) we write the matrices in terms of vectors via the vec operator, which stacks the columns of a matrix one on top of the other. We denote by unvec the inverse operation to vec. The complete algorithm is then as follows.

Algorithm 5.3.3. *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ this algorithm attempts to compute the nearest correlation matrix Y to A by alternating projections with Anderson acceleration. It requires a convergence tolerance* tol.

1 Run Algorithm 5.2.2 on $f\colon \mathbb{R}^{2n^2} \to \mathbb{R}^{2n^2}$ given by $f(z) = \operatorname{vec}(\widetilde{g}(Z) - Z)$,
where $z_k = \operatorname{vec}(Z_k)$, $Z_k = (Y_k, \Delta S_k) \in \mathbb{R}^{n\times 2n}$, and $[X_k, \widetilde{g}(Z_k)] = g(Z_k)$
for the function g defined by Algorithm 5.3.2.
Terminate the iteration when $\|Y_k - X_k\|_2/\|Y_k\|_2 \le$ tol.
Denote the result by x_*.

2 $Y = \operatorname{unvec}(x_*)$

Note that the convergence criterion in Algorithm 5.3.3 is equivalent to that in Algorithm 5.3.2. Note also that, unlike Algorithms 5.3.1 and 5.3.2, Algorithm 5.3.3 is not guaranteed to converge, since there are no suitable convergence results for Anderson acceleration. Whether convergence can be proved under reasonable assumptions is an open question.

The cost per step of the standard alternating projections method (Algorithm 5.3.1) is dominated by the cost of computing $\mathcal{P}_{\mathcal{S}_n}(R_k)$, which is $10n^3$ flops if we compute a full eigendecomposition, or $17n^3/6$ flops if we use tridiagonalization followed by bisection and inverse iteration (computing just the eigenpairs corresponding to the positive eigenvalues or the negative ones, depending which are fewer in number).

One step of Anderson acceleration applied to the alternating projections method in the fixed-point form (Algorithm 5.3.2) uses $2n^2$-sized vectors, so the method takes at most an additional $3m^2/2 + 24mn^2 + 6n^2$ flops per step. Since we find experimentally (see section 5.4) that taking $m \le 5$ (say) is sufficient, the additional cost of Anderson

acceleration is $O(n^2)$ flops, which is negligible for large n. Anderson acceleration also incurs an increase in storage of $2n^2m$ elements.

We next consider two modifications of the alternating projections method for computing the nearest correlation matrix. The first is the problem variant in which specified elements of A have to remain fixed and the second requires the correlation matrix to have smallest eigenvalue bounded below by a positive tolerance δ.

5.3.1 Fixing elements

The nearest correlation matrix problem with fixed elements was previously investigated by Borsdorf [17, Chap. 7] and Lucas [76]. Here we are looking for the closest matrix in the Frobenius norm to a matrix A that lies in the intersection of the set $\mathcal{S}_n$ from (5.2) and

$$\mathcal{E}_n = \{\, X = X^T \in \mathbb{R}^{n\times n} \colon x_{ii} = 1,\ i = 1, \dots, n \text{ and } x_{ij} = a_{ij} \text{ for } (i,j) \in \mathcal{N} \,\},$$

where $\mathcal{N}$ denotes the symmetric index set of the fixed off-diagonal elements. The intersection $\mathcal{S}_n \cap \mathcal{E}_n$ is nonempty, which is equivalent to the problem having a unique solution, if $\mathcal{N}$ is chosen such that there exists a correlation matrix with the prescribed fixed elements. This need not be true for every $\mathcal{N}$, as we have seen from the matrix (1.3).

The alternating projections method trivially generalizes to incorporate the fixed elements constraint: we simply need to replace the projection $\mathcal{P}_{\mathcal{U}_n}$ by the projection $\mathcal{P}_{\mathcal{E}_n}$ onto the set $\mathcal{E}_n$. For a symmetric matrix X this projection is given by

$$\mathcal{P}_{\mathcal{E}_n}(X)_{ij} = \begin{cases} 1, & i = j, \\ a_{ij}, & (i,j) \in \mathcal{N}, \\ x_{ij} & \text{otherwise.} \end{cases}$$

Since we have assumed that $\mathcal{N}$ does not contain any indices corresponding to diagonal elements, $\mathcal{P}_{\mathcal{E}_n}$ remains well-defined even if A does not have unit diagonal. Algorithm 5.3.1 can now be used to solve this problem with a trivial modification of step 6, where $\mathcal{P}_{\mathcal{U}_n}$ is replaced with $\mathcal{P}_{\mathcal{E}_n}$. The extensive numerical experiments in [17, Sec. 7] show that having the additional constraint can result in a significant increase in the number of iterations compared with solving the original problem, so using an

acceleration method becomes even more appealing. The details of applying Anderson acceleration are the same as in the original problem.

The possible non-existence of a solution of this variant of the nearest correlation matrix problem must be reflected in the convergence test. For the matrix (1.3) it is easy to see that X_k and Y_k are both constant for $k \geq 1$, so the first two terms in (5.4) are zero. The last term of (5.4) is, however, of order 1 for all k. The convergence test on line 7 of Algorithm 5.3.1 is hence suitable both for the original problem and for variants that may not have a solution.

5.3.2 Imposing a lower bound on the smallest eigenvalue

In order to avoid singularity, a common requirement in practice is to compute the nearest correlation matrix Y to A with $\lambda_{\min}(Y) \geq \delta$, where $\lambda_{\min}(Y)$ denotes the smallest eigenvalue of Y and $\delta \leq 1$ is a given positive tolerance.

We discuss this modification of the alternating projections method because it further demonstrates the flexibility of the method, which can easily incorporate both the fixed elements constraint and the eigenvalue constraint, unlike the existing Newton methods.

For a given $0 \leq \delta \leq 1$ we define the set

$$\mathcal{S}_n^\delta = \{\, X = X^T \in \mathbb{R}^{n\times n} : \lambda_{\min}(X) \geq \delta \,\}. \tag{5.5}$$

Clearly, $\mathcal{S}_n^0$ is the original $\mathcal{S}_n$ from (5.2). We are looking for the nearest matrix in the Frobenius norm to A from the intersection of $\mathcal{S}_n^\delta$ and $\mathcal{U}_n$, where $\mathcal{U}_n$ is defined in (5.3). The set $\mathcal{S}_n^\delta$ is closed and convex for each δ and since $I_n \in \mathcal{S}_n^\delta$ for every $0 \leq \delta \leq 1$, the closed convex set $\mathcal{S}_n^\delta \cap \mathcal{U}_n$ is nonempty, which implies that this modification of the nearest correlation matrix problem has a unique solution. A formula for the projection $\mathcal{P}_{\mathcal{S}_n^\delta}$ of a symmetric matrix onto the set $\mathcal{S}_n^\delta$ is given by the following result of Cheng and Higham [24, Thm. 3.1].

Theorem 5.3.4. *Let the symmetric matrix $X \in \mathbb{R}^{n\times n}$ have the spectral decomposition $X = Q\,\mathrm{diag}(\lambda_i)Q^T$ and let $\delta \geq 0$. Then for the Frobenius norm the unique matrix*

nearest to X with the smallest eigenvalue at least δ is given by

$$\mathcal{P}_{\mathcal{S}_n^\delta}(X) = Q\,\mathrm{diag}(\tau_i)Q^T, \quad \tau_i = \begin{cases} \lambda_i, & \lambda_i \geq \delta, \\ \delta, & \lambda_i < \delta. \end{cases}$$

Hence, to solve this version of the nearest correlation matrix problem we simply replace the projection $\mathcal{P}_{\mathcal{S}_n}$ in Algorithm 5.3.1 with $\mathcal{P}_{\mathcal{S}_n^\delta}$. If, in addition, some elements of A must remain fixed, we replace $\mathcal{P}_{\mathcal{U}_n}$ with $\mathcal{P}_{\mathcal{E}_n}$ as well. However, note that the latter problem variant does not have a solution for all possible sets $\mathcal{N}$ of fixed positions.

Finally, we briefly discuss how the use of the $\lambda_{\min}(X) \geq \delta$ constraint can address a subtle issue concerning methods for computing the nearest correlation matrix. The resulting matrix is expected to be a positive semidefinite matrix with unit diagonal closest to A. The Newton algorithm of [18] computes a positive semidefinite solution, but to guarantee a unit diagonal the computed matrix is diagonally scaled, which slightly increases the optimal distance to A. In the alternating projections method (Algorithm 5.3.1) the diagonal elements of the returned matrix are exactly 1 but this computed matrix might be indefinite since it is obtained by modifying the diagonal (as well as any other fixed elements) of the positive semidefinite projection. We could swap the order of the projections so that the result is a positive semidefinite matrix, up to roundoff, but then this matrix will not have an exactly unit diagonal. Probably the best solution to these problems is to impose a lower bound on $\lambda_{\min}$ sufficiently large that changes of order the convergence tolerance, tol, will not affect the definiteness. For example, if tol $\approx 10^{-16}$ then $\delta \approx 10^{-8}$ would be adequate.

5.4 Numerical experiments

Now we present experiments that explore the effectiveness of Anderson acceleration at reducing the number of iterations, and the overall execution time, of the alternating projections method for computing the nearest correlation matrix. Test matrices are listed in section 1.4 and we use the following algorithms.

1. `nearcorr`: the alternating projections method for the nearest correlation matrix, Algorithm 5.3.1, modified to incorporate both the fixed elements constraint and the lower bound δ on the smallest eigenvalue by replacing $\mathcal{P}_{\mathcal{U}_n}$ with $\mathcal{P}_{\mathcal{E}_n}$ and $\mathcal{P}_{\mathcal{S}_n}$

Table 5.1: Iteration counts for four small examples for `nearcorr` and `nearcorr_AA`, for varying m (Experiment 1).

Matrix	`it`	`itAA`					
		$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
tec03	39	15	10	9	9	9	9
bhwi01	27	17	14	12	11	10	10
mmb13	801	305	212	117	126	40	31
fing97	33	15	10	10	10	9	9

with $\mathcal{P}_{\mathcal{S}_n^\delta}$, as described in sections 5.3.1 and 5.3.2. The number of iterations for `nearcorr` is denoted by `it`.

2. `nearcorr_AA`: Algorithm 5.3.3 with QR factorization with updating, as described in section 5.2, applied to `nearcorr`. The number of iterations is denoted by `itAA`.

The convergence tolerance tol is set to nu, where n is the order of the matrix and $u \approx 1.1 \times 10^{-16}$ is the unit roundoff.

Convergence is guaranteed for the alternating projections algorithm assuming there are no fixed off-diagonal elements, but could potentially be destroyed by Anderson acceleration, for which we have no convergence guarantees. However, in every test Anderson acceleration and the corresponding unaccelerated algorithm produced computed matrices Y with values of $\|A - Y\|_F$ agreeing to within a small multiple of the convergence tolerance.

In the first three experiments, we have no fixed elements and set $\delta = 0$, that is, we are solving the standard nearest correlation matrix problem.

Experiment 1. We start by comparing the number of iterations for the algorithms `nearcorr` and `nearcorr_AA` as we vary the parameter m on four small examples of invalid correlation matrices. The results are given in Table 5.1.

Clearly, using Anderson acceleration leads to a significant decrease in the number of iterations, even for $m = 1$, with a 25-fold decrease achieved for the mmb13 matrix with $m = 6$. The number of iterations begins to stagnate as m grows, which is consistent with the reported behaviour of Anderson acceleration in the literature.

Experiment 2. Now we compare the iteration count and the computation time for `nearcorr` and `nearcorr_AA` with $m = 2$ for six RiskMetrics matrices. In Table 5.2 we report the number of iterations along with `t`, the total run time in seconds for

Table 5.2: Iteration counts and computation times in seconds for `nearcorr` and `nearcorr_AA` with $m = 2$ for six RiskMetrics matrices of order 387 (Experiment 2).

Matrix	`nearcorr`		`nearcorr_AA`			
	`it`	`t`	`itAA`	`t`	`t_apm`	`t_AA`
1	26	0.46	15	0.45	0.26	0.12
2	50	0.83	24	0.73	0.41	0.19
3	24	0.43	13	0.38	0.23	0.09
4	47	0.88	22	0.68	0.40	0.17
5	34	0.56	18	0.53	0.30	0.14
6	20	0.33	12	0.35	0.20	0.09

each algorithm, and `t_apm` and `t_AA` for `nearcorr_AA`, which are the total time taken in calls to the function g from Algorithm 5.3.2 and in computing the quantities for the convergence test, and the time taken to solve the least-squares problems, respectively. Anderson acceleration roughly halves the number of iterations and the total computation time for `nearcorr_AA` is a little less than for `nearcorr` in the first 5 examples.

The missing time `t`$-$`t_apm`$-$`t_AA` for `nearcorr_AA` represents MATLAB overheads, such as in the vec and unvec conversions of Algorithm 5.3.3. Computing the eigenvalue decomposition, which is the dominant cost for the alternating projections method, remains the main contributing factor to the computation time of `nearcorr_AA`, with the least-squares update and solve taking less than half as much time.

Experiment 3. In the previous experiments our test matrices were small and the total computation time was not an issue. In order to illustrate the dramatic improvement Anderson acceleration can bring to `nearcorr` we next compare `nearcorr` and `nearcorr_AA` with $m = 2$ on two large matrices cor1399 and cor3120. The results are presented in Table 5.3. We again see a very sharp drop in the number of iterations, with `nearcorr_AA` taking less than a third of the iterations for `nearcorr`. This results in a significant reduction in the computation time, with a speedup of as much as 2.9. Comparing the times for the alternating projections part and the least-squares part of `nearcorr_AA` we see that the former heavily dominates the latter.

We next focus on the nearest correlation matrix problem variant with some fixed off-diagonal elements ($\delta = 0$).

Experiment 4. We compare the performance of the methods on the following three

Table 5.3: Iteration counts and computation times in seconds for `nearcorr` and `nearcorr_AA` with $m = 2$ for cor1399 and cor3120 (Experiment 3).

Matrix	nearcorr		nearcorr_AA				speedup
	it	t	itAA	t	t_apm	t_AA	
cor1399	476	219.0	124	75.0	49.6	16.0	2.9
cor3120	559	2746.4	174	999.5	778.5	137.7	2.7

Table 5.4: Iteration counts for `nearcorr`, `nearcorr` with fixed elements, and Anderson acceleration of the latter with varying m (Experiment 4).

Matrix	it	it_fe	itAA_fe				
			$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
fing97	33	34	14	11	10	9	9
cov90	29	169	93	70	55	45	39
usgs13	18	40	15	14	12	12	12

Table 5.5: Computation times in seconds for `nearcorr` with fixed elements and Anderson acceleration applied to it, with varying m (Experiment 4).

Matrix	time_fe	time_fe_AA				
		$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
fing97	3.40e-3	2.51e-3	2.20e-3	2.11e-3	1.20e-3	1.14e-3
cov90	1.71e-1	1.33e-1	1.14e-1	9.06e-2	7.93e-2	8.02e-2
usgs13	5.21e-2	2.06e-2	1.98e-2	1.87e-2	2.54e-2	2.19e-2

examples. The first is the matrix fing97 that we have used in our first experiment. The original requirement in [39] was to compute the nearest correlation matrix having the same leading principal 3×3 submatrix. The second example is cov90, and we need to compute the nearest positive semidefinite matrix to it while preserving the (positive definite) (1,1) block, the (positive) diagonal, and the diagonals in each of the remaining blocks in the first block-row and block-column (see Figure 1.2). The large matrix does not have a unit diagonal but this makes no difference to the methods since these elements are fixed. In our third example we use usgs13, an invalid correlation matrix of order 94, and we wish to compute nearest correlation matrix to it while keeping all diagonal blocks unchanged (see Figure 1.1).

Table 5.4 reports the number of iterations for `nearcorr` with no fixed elements (`it`), the number of iterations for `nearcorr` with the required elements fixed (`it_fe`), and the number of iterations for Anderson acceleration applied to the latter (`itAA_fe`) with m varying from 1 to 5 for our three examples. Table 5.5 presents the computation time in seconds, `time_fe` and `time_fe_AA`, for the latter two algorithms. Due to small values,

Table 5.6: Computation times in seconds for `nearcorr` and `nearcorr_AA` with varying m for four examples where the leading $n/2 \times n/2$ block of a random matrix of size n remains fixed (Experiment 5).

n	`time_fe`	`time_fe_AA`				
		$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
200	6.41	4.42	2.77	2.67	2.29	2.45
400	18.53	13.44	9.35	8.10	6.91	7.35
600	59.47	47.51	28.15	32.04	26.25	31.50
800	136.12	82.23	53.76	63.77	47.61	51.35

these results are not very accurate. We include `nearcorr` with no fixed elements only to demonstrate the effect on the number of iterations of including this constraint, and as this method does not solve our problem we do not run Anderson acceleration on it. The second and third examples show that the constraint of having fixed elements can significantly increase the number of iterations for the alternating projections method compared with the standard nearest correlation matrix problem. From the number of iterations for `nearcorr` with fixed elements and the accelerated algorithm we see that using Anderson acceleration reduces the number of iterations by a similar factor as in the experiments for accelerating the original `nearcorr`. Hence while the additional constraint makes the problem harder to solve by alternating projections it does not affect the speedup of the Anderson acceleration scheme.

Experiment 5. In our second experiment with fixed elements we generate random invalid correlation matrices of order n, with n equal to 200, 400, 600, and 800 and compare the computation time of `nearcorr` and `nearcorr_AA` for varying m, where for each matrix a leading block of size $n/2$ is kept fixed in computing the nearest correlation matrix. We generate the leading block by the MATLAB function call `gallery('randcorr',n/2)` and embed it into an indefinite unit diagonal matrix of size n where the off-diagonal elements are taken from the uniform distribution on $[-1, 1]$. The results reported in Table 5.6 show that the time decreases for m up to 2, but for $m = 4$ or 5 we have an increase in the computation time, which further confirms the merit of keeping m very small. In each example Anderson acceleration achieves a significant reduction in computation time.

Our third set of experiments concerns the nearest correlation matrix problem with a lower bound on the smallest eigenvalue and no fixed elements.

Table 5.7: Iteration counts for four small examples for `nearcorr` and `nearcorr_AA`, for varying m and two values of δ. (Experiment 6)

$\delta = 10^{-8}$

Matrix	`it`	`itAA`					
		$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
tec03	39	15	10	9	9	9	10
bhwi01	27	17	14	12	11	10	10
mmb13	802	280	177	114	58	39	30
fing97	33	15	10	10	10	9	9

$\delta = 0.1$

Matrix	`it`	`itAA`					
		$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
tec03	66	31	19	16	13	14	13
bhwi01	34	23	15	14	12	12	12
mmb13	895	269	216	127	59	48	41
fing97	54	31	24	15	15	14	14

Experiment 6. We first run `nearcorr` on the four small test matrices already used in Table 5.1 for $\delta = 10^{-8}$ and $\delta = 0.1$. The results, reported in Table 5.7, show that for the smaller value of $\delta = 10^{-8}$ the number of iterations is almost identical to the data in Table 5.1, but here the positive definiteness of the solution is guaranteed. For the larger value $\delta = 0.1$, the number of iterations is increased compared with $\delta = 0$. As with the fixed elements constraint, we see that Anderson acceleration again reduces the iteration number by a similar factor as in the unconstrained case, that is, its performance is not affected by including the bound on the smallest eigenvalue.

Experiment 7. The benefits of Anderson acceleration in the positive definite case are even more evident if we reproduce Experiment 2, now using `nearcorr` with $\delta = 0.1$ and compare the results in Table 5.8 with those in Table 5.2. Computing the positive definite solution takes between 30 and 90 times more iterations than computing the semidefinite nearest correlation matrix but Anderson acceleration now reduces the number of iterations by a factor between 3.6 and 4.6, compared with halving the iterations in the original experiment, which shows that Anderson acceleration can be even more effective for constrained nearest correlation matrix problems than for the original problem. We also see that `nearcorr_AA` requires approximately half the time of `nearcorr`.

We now combine the constraints of keeping elements fixed and of positive definite-

Table 5.8: Iteration counts and computation times in seconds for `nearcorr` with $\delta = 0.1$ and `nearcorr_AA` with $m = 2$ for six RiskMetrics matrices of order 387 (Experiment 7).

	`nearcorr`		`nearcorr_AA`			
Matrix	`it`	`t`	`itAA`	`t`	`t_apm`	`t_AA`
1	1410	20.50	383	10.77	5.70	3.12
2	2100	33.93	513	15.83	8.52	4.56
3	1900	31.14	414	11.58	5.97	3.54
4	1586	29.06	369	12.83	7.09	3.54
5	1812	31.30	400	12.99	7.16	3.62
6	1794	29.08	393	11.63	6.20	3.40

ness.

Experiment 8. We take the three matrices from Experiment 4 with fixed elements and run `nearcorr` and `nearcorr_AA` with $\delta = 0.1$, with varying m. Note that in this case we have no guarantee of the existence of a feasible point and in fact for the cov90 matrix the algorithms do not converge within 100,000 iterations for the default tolerance. Hence we exclude this example and present in Table 5.9 only the results for the test matrices fing97 and usgs13. We note the increase in the number of iterations compared with the data in Table 5.4 where we only fixed the elements. Anderson acceleration (with $m = 5$) reduces the iterations by a factor of 3.6 for the smaller matrix and 6.7 for the larger, while in the original experiment the factors were 3.8 and 3.3.

Table 5.9: Iteration counts and computation times in seconds for `nearcorr` and `nearcorr_AA` with $\delta = 0.1$ and varying m for two examples with fixed elements (Experiment 8).

	`nearcorr`		`nearcorr_AA`									
			$m = 1$		$m = 2$		$m = 3$		$m = 4$		$m = 5$	
Matrix	`t`	`it`	`t`	`it`	`t`	`it`	`t`	`it`	`t`	`it`	`t`	`it`
fing97	2.98e-3	54	4.95e-3	31	4.57e-3	25	2.59e-3	16	2.74e-3	15	2.75e-3	15
usgs13	1.25e-1	128	5.24e-2	36	4.10e-2	25	4.32e-2	24	3.91e-2	20	3.93e-2	19

Table 5.10: Iteration counts for four small examples for `nearcorr`, `nearcorr_AA` with $m = 2$, and the acceleration scheme from [75] (Experiment 9).

Matrix	`it`	`itAA`	`it_2`
tec03	39	10	39
bhwi01	27	14	27
mmb13	801	212	725
fing97	33	10	33

Experiment 9. As a final experiment we use the four matrices from Experiment 1 to compare Anderson acceleration with the acceleration scheme from [75]. Table 5.10 shows the number of iterations, `it_2`, for that scheme, in which we set its safeguard parameter ε to 10^{-14} and use the same convergence tolerance as in all our experiments. The number of iterations for the acceleration scheme is the same as for the unaccelerated method in each case except for the mmb13 matrix, and in that case we see a reduction in the number of iterations by a factor 1.1 versus 3.8 for Anderson acceleration. In all test cases, after a few initial iterations the mixing parameter α_k needed for the scheme [75] could not be computed because the safeguard was triggered. We conclude that the acceleration scheme of [75] is not competitive with Anderson acceleration on this class of problems because it displays the "orthogonality property" discussed in [75, Rem. 1].

To summarize, in these experiments we have found that Anderson acceleration of the alternating projections method for the nearest correlation matrix, with an appropriate choice of $m \in [1, 6]$, results in a reduction in the number of iterations by a factor of at least 2 for the standard algorithm and a factor at least 3 when additional constraints are included. The factors can be much larger than these worst-cases, especially in the experiments with additional constraints, where we saw a reduction in the number of iterations by a factor 21.8 in Table 5.7. Acceleration therefore tends to produce the greatest benefits on the problems that alternating projections finds the hardest. Moreover, the reduction in the number of iterations is generally reflected in the run times, modulo MATLAB overheads.

Chapter 6

Principal Pivot Transforms of Quasidefinite Matrices and Semidefinite Lagrangian Subspaces

Somebody came up to me after a talk I had given, and said, "You make mathematics seem like fun." I was inspired to reply, "If it isn't fun, why do it?"

—Ralph P. Boas

6.1 Introduction and preliminaries

Recall from section 1.2 that in a permuted Riccati representation introduced by Mehrmann and Poloni [79] a Lagrangian subspace is identified with the pair $(\mathcal{I}, X)$, where $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ and $X \in \mathbb{C}^{n\times n}$ is Hermitian. The symmetric PPT (1.13) is used to convert between two different representations in an optimization algorithm [79, Alg. 2] which computes a subset $\mathcal{I}_{\text{opt}}$ and an associated X_{opt} whose elements are bounded by a small constant.

In this chapter we focus on a class of Lagrangian subspaces whose representation $(\mathcal{I}, X)$ has additional structure. Let the symbol $\succ$ denote the Löwner ordering: $A \succ B$ ($A \succeq B$) means that $A - B$ is positive (semi)definite. We say that a Hermitian matrix $X = X^* \in \mathbb{C}^{n\times n}$ is *$\mathcal{I}$-definite*, for $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$, if

$$X_{\mathcal{I}\mathcal{I}} \prec 0 \quad \text{and} \quad X_{\mathcal{I}^c\mathcal{I}^c} \succ 0. \tag{6.1}$$

If the previous definition holds with the symbols $\succ, \prec$ replaced by $\succeq, \preceq$ then X is *$\mathcal{I}$-semidefinite*. For $\mathcal{I} = \emptyset$ an $\mathcal{I}$-definite matrix is simply a positive definite matrix and for $\mathcal{I} = \{1, 2, \ldots, n\}$ an $\mathcal{I}$-definite matrix is negative definite. In all other cases,

an $\mathcal{I}$-definite matrix is a generalization of a *quasidefinite* matrix, which is $\mathcal{I}$-definite for $\mathcal{I} = \{1, 2, \ldots, k\}$ with some $k < n$.

Identifying this class of subspaces and exploiting its properties in applications has several advantages: we can improve a bound on the elements of the matrix X_{opt} and preserve this additional structure, which is, for instance, crucial for the existence of a positive semidefinite solution X of an algebraic Riccati equation.

The rest of the chapter is structured as follows. We introduce a class of Lagrangian *(semi)definite* subspaces in Section 6.2 and prove that for these subspaces the Hermitian matrix X in the pair $(\mathcal{I}, X)$ which represents the Lagrangian semidefinite subspace is $\mathcal{I}$-semidefinite for all possible choices of $\mathcal{I}$. In Section 6.3 we link Lagrangian semidefinite subspaces to Hamiltonian and symplectic pencils appearing in control theory. In Section 6.4 we derive an implementation of the symmetric PPT (1.13) which converts between two different representations $(\mathcal{I}, X)$ and $(\mathcal{J}, X')$ of a Lagrangian semidefinite subspace. Specifically, we show how an $\mathcal{I}$-semidefinite matrix X can be converted to a $\mathcal{J}$-semidefinite matrix X' for a given index set $\mathcal{J}$ by the symmetric PPT that both makes use of the definiteness properties of X and guarantees the definiteness of the blocks of X' in finite arithmetic. The symmetric PPT in one case requires the computation of the inverse of a quasidefinite matrix with factored diagonal blocks and we also present an inversion formula which uses unitary factorizations to directly compute the factors of the diagonal blocks of the quasidefinite inverse. In Section 6.5 we prove that all elements of an $\mathcal{I}_{\text{opt}}$-semidefinite matrix X_{opt} associated with a semidefinite Lagrangian subspace are bounded by 1 in modulus, and present the optimization algorithm which computes an optimal representation. We test the performance of the algorithm on several numerical experiments in Section 6.6.

6.2 Semidefinite Lagrangian subspaces

We start by explaining why the name "Riccati matrix" is fitting for $\mathcal{G}(X) = \left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]$. A fundamental result in the analysis of algebraic Riccati equations is that the matrix $X \in \mathbb{C}^{k\times k}$ is a solution to a continuous-time algebraic Riccati equation $Q + XA +$

$A^*X - XGX = 0$, where $A, G = G^*, Q = Q^* \in \mathbb{C}^{k\times k}$ if and only if

$$H \begin{bmatrix} I_k \\ X \end{bmatrix} = \begin{bmatrix} I_k \\ X \end{bmatrix} (A - GX), \tag{6.2}$$

where the associated matrix H is Hamiltonian ($J_k H = (J_k H)^*$) and given by

$$H = \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix}.$$

The equation (6.2) shows that solving a continuous-time algebraic Riccati equation $Q + XA + A^*X - XGX = 0$ is equivalent to solving an invariant subspace problem for the associated Hamiltonian matrix H, if we impose that the subspace is represented via a Riccati basis $\mathcal{G}(X)$. If the matrices Q and G are positive semidefinite, under standard conditions (see, e.g. [13], [78], [85]) the Riccati equation has a unique positive semidefinite solution, and this is the solution that is usually of interest. A common approach to computing it is to determine a basis for the *stable invariant subspace* of H, i.e., the one corresponding to the eigenvalues of H in the open left half plane (e.g. [2], [70], [78]). This subspace is Lagrangian and if $U = \left[\begin{smallmatrix} U_1 \\ U_2 \end{smallmatrix}\right]$ is its basis then the matrix U_1 is invertible and $X = U_2 U_1^{-1}$ is the positive semidefinite solution to the Riccati equation [85]. Specifically, $U \sim \mathcal{G}(X)$ and the matrix $X = X^* \succeq 0$.

In this section we take a closer look at Lagrangian subspaces which have a Riccati basis with this property. We call a Lagrangian subspace *definite* if it can be written as

$$\mathcal{U} = \operatorname{Im} \mathcal{G}(X), \quad X = X^* \succ 0, \quad X \in \mathbb{C}^{n\times n},$$

where $\mathcal{G}(X)$ is a Riccati matrix defined in (1.6). The following result relates $\mathcal{I}$-definite matrices defined by the property (6.1) and definite Lagrangian subspaces.

Theorem 6.2.1. *Let $U \in \mathbb{C}^{2n\times n}$ have full column rank. The following properties are equivalent.*

1. *$\mathcal{U} = \operatorname{Im} U$ is Lagrangian definite.*
2. *For some $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ we have $U \sim \mathcal{G}_{\mathcal{I}}(X)$, where X is $\mathcal{I}$-definite and $\mathcal{G}_{\mathcal{I}}(X)$ is a permuted Riccati matrix defined in (1.11).*
3. *For all $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ we have $U \sim \mathcal{G}_{\mathcal{I}}(X)$ and X is $\mathcal{I}$-definite.*

Proof. Let $U \sim \mathcal{G}(X)$ and $X \succ 0$. From the definition of a symplectic swap matrix (1.8) it follows that $\Pi_\emptyset = I_{2n}$ and hence $\mathcal{G}_\emptyset(X) = \mathcal{G}(X)$. Therefore, the definition of a Lagrangian definite subspace can be reformulated as stating $U \sim \mathcal{G}_\mathcal{I}(X)$, where X is $\mathcal{I}$-definite for $\mathcal{I} = \emptyset$. If this holds, then for each $\mathcal{J} \subseteq \{1, 2, \ldots, n\}$ Lemma 1.2.5 defines $\mathcal{K} = \mathcal{J}$ and $D = I_n$. Since $X \succ 0$ every principal submatrix $X_{\mathcal{K}\mathcal{K}}$ is also positive definite and therefore $U \sim \mathcal{G}_\mathcal{J}(X')$ for every $\mathcal{J}$, where X' is the symmetric PPT (1.13) of X. It is clear from the formulae (1.13) and the properties of Schur complements [61, Sec. 12.3] that X' is $\mathcal{J}$-definite, as required.

On the other hand, if $U \sim \mathcal{G}_\mathcal{I}(X)$ and X is $\mathcal{I}$-definite for some $\mathcal{I}$, then X is Hermitian by definition and hence U spans a Lagrangian subspace. We prove that the subspace is Lagrangian definite by applying Lemma 1.2.5 with $\mathcal{J} = \emptyset$ to X. It follows that $\mathcal{K} = \mathcal{I}$ and since $X_{\mathcal{K}\mathcal{K}} = X_{\mathcal{I}\mathcal{I}} \prec 0$, we have $U \sim \mathcal{G}_\emptyset(X') = \mathcal{G}(X')$, with $X' = DYD$ as in Lemma 1.2.5. Since X' is defined via congruence it is sufficient to prove that Y, the symmetric PPT of an $\mathcal{I}$-definite matrix with respect to the index set $\mathcal{I}$, is positive definite. This follows from (1.13) due to the definiteness properties of the blocks of X: both $Y_{\mathcal{K}\mathcal{K}} = -X_{\mathcal{I}\mathcal{I}}^{-1}$ and its Schur complement $Y_{\mathcal{K}^c\mathcal{K}^c} - Y_{\mathcal{K}^c\mathcal{K}} Y_{\mathcal{K}\mathcal{K}}^{-1} Y_{\mathcal{K}\mathcal{K}^c} = X_{\mathcal{I}^c\mathcal{I}^c}$ are positive definite, so again by the properties of Schur complements Y is positive definite and the proof is complete. □

More interesting is the corresponding semidefinite case, in which existence of the permuted Riccati representation is not guaranteed for all $\mathcal{I}$, cf. Example 1.2.4.

Theorem 6.2.2. *Let $U \in \mathbb{C}^{2n \times n}$ have full column rank. The following properties are equivalent.*

1. *For some $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ we have $U \sim \mathcal{G}_\mathcal{I}(X)$, where X is $\mathcal{I}$-semidefinite and $\mathcal{G}_\mathcal{I}(X)$ is a permuted Riccati matrix defined in (1.11).*

2. *For all $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ such that the permuted Riccati representation exists, i.e., $U \sim \mathcal{G}_\mathcal{I}(X)$, the matrix X is $\mathcal{I}$-semidefinite.*

When these properties hold, we call the subspace *Lagrangian semidefinite*.

Proof. Let $\mathcal{I}$ be such that $U \sim \mathcal{G}_\mathcal{I}(X)$ and X is $\mathcal{I}$-semidefinite. Consider the matrix Y obtained by perturbing the diagonal entries of X so that the blocks $X_{\mathcal{I}\mathcal{I}}$ and

$X_{\mathcal{I}^c\mathcal{I}^c}$ become strictly definite, that is, for some $\varepsilon > 0$,

$$Y_{\mathcal{I}\mathcal{I}}{=}X_{\mathcal{I}\mathcal{I}} - \varepsilon I \prec 0, \qquad Y_{\mathcal{I}\mathcal{I}^c}{=}X_{\mathcal{I}\mathcal{I}^c},$$
$$Y_{\mathcal{I}^c\mathcal{I}}{=}X_{\mathcal{I}^c\mathcal{I}}, \qquad Y_{\mathcal{I}^c\mathcal{I}^c}{=}X_{\mathcal{I}^c\mathcal{I}^c} + \varepsilon I \succ 0.$$

Then the subspace $\operatorname{Im} U_\varepsilon$, with $U_\varepsilon \sim \mathcal{G}_{\mathcal{I}}(Y)$, is Lagrangian definite, and by Theorem 6.2.1, the permuted Riccati representations $U_\varepsilon \sim \mathcal{G}_{\mathcal{I}}(Z)$ exist for every $\mathcal{I}$ with Z having the required definiteness properties. By passing to the limit $\varepsilon \to 0$, we get the semidefiniteness of the blocks of Z (whenever the representation exists). □

Example 6.2.3. Consider the subspace in Example 1.2.4. We have $U \sim \mathcal{G}_{\emptyset}(X)$ for X positive semidefinite, so U is Lagrangian semidefinite. Other choices of the index set for which the permuted Riccati representations exist are $\mathcal{I} = \{1\}$ and $\mathcal{I} = \{2\}$ and the corresponding matrices X are $\{1\}$-semidefinite and $\{2\}$-semidefinite, respectively.

6.3 Semidefinite Lagrangian subspaces associated with control-theory pencils

Section 6 of [79] introduces a method to map regular matrix pencils with special structures to Lagrangian subspaces. The main reason why this kind of bijection is used is that changing a basis in the subspace is equivalent to premultiplying the pencil by a nonsingular matrix, which preserves eigenvalues and right eigenvectors of regular pencils. This makes it possible to apply several techniques based on PPTs to pencils as well. Specifically, we write

$$M_1 - xN_1 \sim M_2 - xN_2,$$

and say that the two pencils are *left equivalent*, if there exists a nonsingular square matrix S such that $M_1 = SM_2$ and $N_1 = SN_2$. It follows that $M_1 - xN_1 \sim M_2 - xN_2$ if and only if $\begin{bmatrix} M_1 & N_1 \end{bmatrix}^* \sim \begin{bmatrix} M_2 & N_2 \end{bmatrix}^*$. Hence, if we are interested in the eigenvalues and right eigenvectors of a regular pencil we may instead work with any regular pencil left equivalent to it.

We construct here a simple variation of the map from [79] which sends the pencils appearing in most applications in control theory to semidefinite Lagrangian subspaces. The map is defined for pencils $M - xN$ *without a common left kernel*, which means

that there exists no vector $v \neq 0$ such that $v^*M = v^*N = 0$. This is a proper superset of regular pencils, as a common left kernel implies that $\det(M - xN) \equiv 0$ so a pencil is singular but the converse does not hold, with $M - xN = \left[\begin{smallmatrix} 0 & x & 1 \\ x & 0 & 0 \\ 1 & 0 & 0 \end{smallmatrix}\right]$ providing a counterexample.

A *Hamiltonian pencil* is a matrix pencil $M - xN \in \mathbb{C}^{2k\times 2k}[x]$ such that $MJ_kN^* + NJ_kM^* = 0$. In several problems in control theory, e.g. [74], [78], one deals with Hamiltonian pencils in the form

$$\begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} - xI_{2k}, \quad A, G, Q \in \mathbb{C}^{k\times k}, \quad G = G^* \succeq 0, \quad Q = Q^* \succeq 0; \tag{6.3}$$

moreover, factorizations $G = BB^*$ and $Q = C^*C$ (with $B \in \mathbb{C}^{k\times t}$, $C \in \mathbb{C}^{r\times k}$, $r, t \leq k$) are known in advance. In the following theorem, we show that this kind of structure is mapped to a semidefinite Lagrangian subspace by a special bijection between pencils and $4k \times 2k$ matrices.

Theorem 6.3.1. *Let*

$$M - xN = \begin{bmatrix} M_1 & M_2 \end{bmatrix} - x \begin{bmatrix} N_1 & N_2 \end{bmatrix}, \quad M_1, M_2, N_1, N_2 \in \mathbb{C}^{2k\times k}$$

be a matrix pencil without a common left kernel. Construct the matrix

$$U = \begin{bmatrix} M_1 & -N_1 & -N_2 & M_2 \end{bmatrix}^*. \tag{6.4}$$

Then,

1. *$M - xN$ is Hamiltonian if and only if $\operatorname{Im} U$ is Lagrangian.*
2. *If $M - xN$ is in the form (6.3), then $\operatorname{Im} U$ is Lagrangian semidefinite.*

Proof. The first claim is proved by expanding the relation $U^*J_{2k}U = 0$ into blocks. This leads to the expression $-M_1N_2^* - N_1M_2^* + N_2M_1^* + M_2N_1^* = 0$, which we can recombine to get $MJ_kN^* + NJ_kM^* = 0$.

For the second claim, take $M - xN$ as in (6.3), and $\mathcal{I} = \{1, 2, \ldots, k\}$. We have

$$\Pi_{\mathcal{I}}U = \begin{bmatrix} 0 & 0 & I_k & 0 \\ 0 & I_k & 0 & 0 \\ -I_k & 0 & 0 & 0 \\ 0 & 0 & 0 & I_k \end{bmatrix} \begin{bmatrix} A^* & -Q \\ -I_k & 0 \\ 0 & -I_k \\ -G & -A \end{bmatrix} = \begin{bmatrix} 0 & -I_k \\ -I_k & 0 \\ -A^* & Q \\ -G & -A \end{bmatrix} \sim \begin{bmatrix} I_k & 0 \\ 0 & I_k \\ -Q & A^* \\ A & G \end{bmatrix}.$$

Hence, $U \sim \Pi_{\mathcal{I}}^{T}\mathcal{G}(X) = \mathcal{G}_{\mathcal{I}}(X)$, with $X = \left[\begin{smallmatrix} -Q & A^* \\ A & G \end{smallmatrix}\right]$, which is $\mathcal{I}$-semidefinite. Thus, by Theorem 6.2.2, the subspace $\operatorname{Im} U$ is Lagrangian semidefinite. □

Equation (6.4) in [79] gives a matrix U in a form similar to (6.4), which satisfies only the first part of the theorem.

Similarly, a *symplectic pencil* is a matrix pencil $M - xN \in \mathbb{C}^{2k\times 2k}[x]$ such that $MJ_kM^* = NJ_kN^*$. In several problems in discrete-time control theory, e.g. [38], [74], [78], one deals with symplectic pencils in the form

$$\begin{bmatrix} A & 0 \\ -Q & I_k \end{bmatrix} - x \begin{bmatrix} I_k & G \\ 0 & A^* \end{bmatrix}, \quad A, G, Q \in \mathbb{C}^{k\times k}, \quad G = G^* \succeq 0, \quad Q = Q^* \succeq 0; \tag{6.5}$$

again, factorizations $G = BB^*$, $Q = C^*C$ as above are often available. Similarly to the Hamiltonian case, there is a bijection which maps this structure into a semidefinite Lagrangian subspace.

Theorem 6.3.2. *Let*

$$M - xN = \begin{bmatrix} M_1 & M_2 \end{bmatrix} - x \begin{bmatrix} N_1 & N_2 \end{bmatrix}, \quad M_1, M_2, N_1, N_2 \in \mathbb{C}^{2k\times k}$$

be a matrix pencil without a common left kernel. Construct the matrix

$$U = \begin{bmatrix} M_1 & -N_1 & -M_2 & -N_2 \end{bmatrix}^*. \tag{6.6}$$

Then,

1. *$M - xN$ is symplectic if and only if $\operatorname{Im} U$ is Lagrangian.*

2. *If $M - xN$ is in the form (6.5), then $\operatorname{Im} U$ is Lagrangian semidefinite.*

Proof. The proof of both claims is analogous to the proof of Theorem 6.3.1. Specifically, the Lagrangian semidefinite subspace spanned by the columns of U from (6.6) is also associated to the quasidefinite matrix $X = \left[\begin{smallmatrix} -Q & A^* \\ A & G \end{smallmatrix}\right]$. □

Once again, a construction given in Equation (6.2) in [79] provides an analogous bijection that satisfies only the first part of the theorem. The main use for these bijections is producing left-equivalent pencils with better numerical properties. We show it in a simple case.

Example 6.3.3. Consider $k = 1$, $A = 1$, $G = 10^5$, $Q = 0.1$. The Hamiltonian pencil $M - xN$ obtained as in (6.3) has the condition number $\kappa([\begin{smallmatrix} M \\ N \end{smallmatrix}]) \approx 10^5$, that is, a perturbation of relative magnitude 10^{-5} can turn it into a pencil with a common kernel. If we construct the matrix U in (6.4) associated with it and apply Algorithm 6.5.2 described in Section 6.5 to obtain an equivalent permuted Riccati representation of U with smaller entries, we get $U \sim \mathcal{G}_{\mathcal{I}_{\text{opt}}}(X_{\text{opt}})$ with $\mathcal{I}_{\text{opt}} = \{1, 2\}$ and $X_{\text{opt}} = \left[\begin{smallmatrix} -0.1-10^{-5} & 10^{-5} \\ 10^{-5} & -10^{-5} \end{smallmatrix}\right]$. Partitioning the matrix

$$\mathcal{G}_{\mathcal{I}_{\text{opt}}}(X_{\text{opt}}) = \begin{bmatrix} 0.1 + 10^{-5} & -10^{-5} \\ -10^{-5} & 10^{-5} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \hat{M}_1^T \\ -\hat{N}_1^T \\ -\hat{N}_2^T \\ \hat{M}_2^T \end{bmatrix}$$

conformably to (6.4), we obtain a left-equivalent pencil

$$\hat{M} - x\hat{N} = \begin{bmatrix} \hat{M}_1 & \hat{M}_2 \end{bmatrix} - x \begin{bmatrix} \hat{N}_1 & \hat{N}_2 \end{bmatrix} = \begin{bmatrix} 0.1 + 10^{-5} & 0 \\ -10^{-5} & 1 \end{bmatrix} - x \begin{bmatrix} 10^{-5} & -1 \\ -10^{-5} & 0 \end{bmatrix},$$

with $\kappa\left(\left[\begin{smallmatrix} \hat{M} \\ \hat{N} \end{smallmatrix}\right]\right) \approx 14$, a considerably lower value. The two pencils are Hamiltonian and have the same eigenvalues and right eigenvectors, so they are completely equivalent from a numerical perspective.

The optimization algorithm [79, Alg. 2] uses the PPT formulae (1.13) to compute an optimal permuted Riccati representation of a Lagrangian subspace and it can be used to normalize pencils [79, Sec. 6]. If a PPT is applied to a Lagrangian semidefinite subspace $\operatorname{Im} U$, where U is for example given in (6.4) or (6.6), the definiteness properties of the blocks G and Q are not exploited. Furthermore, due to Theorem 6.2.2, for the computed optimal representation $(\mathcal{I}_{\text{opt}}, X_{\text{opt}})$ the matrix X_{opt} must be $\mathcal{I}_{\text{opt}}$-semidefinite but the definiteness properties of its submatrices are not guaranteed due to possible numerical errors. Note the structure of the matrix X appearing in the proof of the second part of Theorem 6.3.1 and Theorem 6.3.2: when the factors B and C are known for representations (6.3) and (6.5), the quasidefinite matrix X is

$$X = \begin{bmatrix} -Q & A^* \\ A & G \end{bmatrix} = \begin{bmatrix} -C^*C & A^* \\ A & BB^* \end{bmatrix}.$$

In the next section we describe the structure preserving implementation of the symmetric PPT (1.13) for $\mathcal{I}$-semidefinite matrices X in factored form, which resolves the

issues described above and leads to the structured version of the optimization algorithm presented in section 6.5.

6.4 Applying a PPT to a factored representation of an $\mathcal{I}$-semidefinite matrix

Let $X \in \mathbb{C}^{n\times n}$ be $\mathcal{I}$-semidefinite and $k = \mathrm{card}(\mathcal{I})$, where $\mathrm{card}(\mathcal{I})$ denotes the number of elements of the set $\mathcal{I}$. Due to the definiteness properties there exist matrices $A \in \mathbb{C}^{(n-k)\times k}$, $B \in \mathbb{C}^{(n-k)\times t}$, and $C \in \mathbb{C}^{r\times k}$ such that

$$\begin{aligned} X_{\mathcal{I}\mathcal{I}} &= -C^*C \in \mathbb{C}^{k\times k}, & X_{\mathcal{I}\mathcal{I}^c} &= A^*, \\ X_{\mathcal{I}^c\mathcal{I}} &= A, & X_{\mathcal{I}^c\mathcal{I}^c} &= BB^* \in \mathbb{C}^{(n-k)\times(n-k)}. \end{aligned} \tag{6.7}$$

Any A, B, and C satisfying (6.7) are called the *factors* of the $\mathcal{I}$-semidefinite matrix X. Specifically, B and C do not have to be of full rank. We also introduce the following compact form of (6.7):

$$X = \mathcal{C}_{\mathcal{I}}\left(\begin{bmatrix} C & 0 \\ A & B \end{bmatrix}\right),$$

and say that the map $\mathcal{C}_{\mathcal{I}}$ converts between any factor representation of the $\mathcal{I}$-semidefinite matrix X and the real matrix. Clearly, the factors B and C are not unique as for any unitary matrices H and U of conformal size we have

$$X = \mathcal{C}_{\mathcal{I}}\left(\begin{bmatrix} C & 0 \\ A & B \end{bmatrix}\right) = \mathcal{C}_{\mathcal{I}}\left(\begin{bmatrix} HC & 0 \\ A & BU \end{bmatrix}\right).$$

Given an $\mathcal{I}$-semidefinite matrix X in a factored form (6.7) and an index set $\mathcal{J}$, our goal in this section is to derive formulae for the symmetric PPT (1.13) needed in Lemma 1.2.5 to compute a $\mathcal{J}$-semidefinite matrix X' so $\mathcal{G}_{\mathcal{I}}(X) \sim \mathcal{G}_{\mathcal{J}}(X')$ where

$$X' = \mathcal{C}_{\mathcal{J}}\left(\begin{bmatrix} C' & 0 \\ A' & B' \end{bmatrix}\right),$$

and the factors A', B', and C' are computed *directly* from A, B, and C.

We distinguish three cases for the index set $\mathcal{J}$ we are converting to:

Case 1: $\mathcal{J} \subseteq \mathcal{I}$ (the negative semidefinite block shrinks, the positive semidefinite block expands), in which case $\mathcal{K} = \mathcal{I} \setminus \mathcal{J}$,

Case 2: $\mathcal{J} \supseteq \mathcal{I}$ (the negative semidefinite block expands, the positive semidefinite block shrinks), where $\mathcal{K} = \mathcal{J} \setminus \mathcal{I}$, and

Case 3: $\mathcal{I} \setminus \mathcal{J} \neq \emptyset$ and $\mathcal{J} \setminus \mathcal{I} \neq \emptyset$, in which case $\mathcal{K} = (\mathcal{I} \setminus \mathcal{J}) \cup (\mathcal{J} \setminus \mathcal{I})$.

We now derive the formulae for A', B', and C' in each case. For simplicity, so that we may use a simpler matrix form instead of working with a generic block partition (6.7), take $\mathcal{I} = \{1, 2 \ldots, k\}$ so that X is

$$X = \mathcal{C}_{\mathcal{I}}\left(\begin{bmatrix} C & 0 \\ A & B \end{bmatrix}\right) = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} -C^*C & A^* \\ A & BB^* \end{bmatrix} \end{array}. \tag{6.8}$$

6.4.1 Case 1.

Recall that we have $A \in \mathbb{C}^{(n-k)\times k}$, $B \in \mathbb{C}^{(n-k)\times t}$, $C \in \mathbb{C}^{r\times k}$ as factors of an $\mathcal{I}$-semidefinite matrix X. Again for simplicity, we take $\mathcal{J} = \{1, 2, \ldots, k-l\}$ for some l with $1 \leq l \leq k$. Let H be a unitary matrix such that

$$A = \begin{array}{c} \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k-l & l \end{array} \\ \begin{bmatrix} A_1 & A_2 \end{bmatrix} \end{array}, \quad HC = \begin{array}{c} r-l \\ l \end{array} \begin{array}{c} \begin{array}{cc} k-l & l \end{array} \\ \begin{bmatrix} C_{11} & 0 \\ C_{21} & C_{22} \end{bmatrix} \end{array}. \tag{6.9}$$

Then the compact factor representation (6.8) of X is

$$X = \mathcal{C}_{\mathcal{I}}\left(\left[\begin{array}{c|c} HC & 0 \\ \hline A & B \end{array}\right]\right) = \left[\begin{array}{cc|c} -C_{11}^*C_{11} - C_{21}^*C_{21} & -C_{21}^*C_{22} & A_1^* \\ -C_{22}^*C_{21} & -C_{22}^*C_{22} & A_2^* \\ \hline A_1 & A_2 & BB^* \end{array}\right].$$

Now we use Lemma 1.2.5 to convert between our two permuted Riccati representations. The pivot index set is $\mathcal{K} = \{k-l+1, \ldots, k\}$. Note that for this PPT to exist it must hold $r \geq l$. For the pivot submatrix $X_{\mathcal{K}\mathcal{K}} = -C_{22}^*C_{22}$ to be nonsingular, the square matrix C_{22} must be invertible. The diagonal sign change matrix D is the block diagonal matrix $D = \operatorname{diag}(I_{k-l}, -I_l, I_{n-k})$, and applying (1.13) to X to compute Y we get

$$X' = DYD = \left[\begin{array}{c|cc} -C_{11}^*C_{11} & -C_{21}^*C_{22}^{-*} & A_1^* - C_{21}^*C_{22}^{-*}A_2^* \\ \hline -C_{22}^{-1}C_{21} & (C_{22}^*C_{22})^{-1} & (C_{22}^*C_{22})^{-1}A_2^* \\ A_1 - A_2C_{22}^{-1}C_{21} & A_2(C_{22}^*C_{22})^{-1} & BB^* + A_2(C_{22}^*C_{22})^{-1}A_2^* \end{array}\right].$$

The matrix X' is $\mathcal{J}$-semidefinite (as follows by Theorem 6.2.2) and it is easy to check that it can be represented as $X' = \mathcal{C}_{\mathcal{J}}\left(\left[\begin{array}{c|c} C' & 0 \\ \hline A' & B' \end{array}\right]\right)$ for $A' \in \mathbb{C}^{(l+n-k)\times(k-l)}$, $B' \in \mathbb{C}^{(l+n-k)\times(l+t)}$, $C' \in \mathbb{C}^{(r-l)\times(k-l)}$ given by

$$A' = \begin{bmatrix} -C_{22}^{-1}C_{21} \\ A_1 - A_2C_{22}^{-1}C_{21} \end{bmatrix}, \quad B' = \begin{bmatrix} C_{22}^{-1} & 0 \\ A_2C_{22}^{-1} & B \end{bmatrix}, \quad C' = C_{11}. \tag{6.10}$$

6.4.2 Case 2.

Case 2 is very similar to Case 1. We again start from $A \in \mathbb{C}^{(n-k)\times k}$, $B \in \mathbb{C}^{(n-k)\times t}$, $C \in \mathbb{C}^{r\times k}$ and now take $1 \le m \le n-k$, with $m \le t$, to apply Lemma 1.2.5 to X from (6.8) for $\mathcal{J} = \{1, 2, \ldots, k, k+1, \ldots, k+m\}$, for simplicity. Let U be a unitary matrix such that

$$A = \begin{array}{c} \\ m \\ n-k-m \end{array}\begin{array}{c} \overset{k}{} \\ \left[\begin{array}{c} A_1 \\ A_2 \end{array}\right] \end{array}, \quad BU = \begin{array}{c} \\ m \\ n-k-m \end{array}\begin{array}{c} \overset{m \quad t-m}{} \\ \left[\begin{array}{cc} B_{11} & 0 \\ B_{21} & B_{22} \end{array}\right] \end{array}. \tag{6.11}$$

The compact factor representation (6.8) expands to

$$X = \mathcal{C}_{\mathcal{I}}\left(\left[\begin{array}{c|c} C & 0 \\ \hline A & BU \end{array}\right]\right) = \left[\begin{array}{c|cc} -C^*C & A_1^* & A_2^* \\ \hline A_1 & B_{11}B_{11}^* & B_{11}B_{21}^* \\ A_2 & B_{21}B_{11}^* & B_{21}B_{21}^* + B_{22}B_{22}^* \end{array}\right].$$

From Lemma 1.2.5 we have $\mathcal{K} = \{k+1, \ldots, k+m\}$ and $D = I_n$. The pivot submatrix is $X_{\mathcal{K}\mathcal{K}} = B_{11}B_{11}^*$ and B_{11} must be invertible for this PPT operation to be defined. If this is the case, we have

$$\begin{aligned} X' = DYD &= \left[\begin{array}{cc|c} -C^*C - A_1^*(B_{11}B_{11}^*)^{-1}A_1 & A_1^*(B_{11}B_{11}^*)^{-1} & A_2^* - A_1^*B_{11}^{-*}B_{21}^* \\ (B_{11}B_{11}^*)^{-1}A_1 & -(B_{11}B_{11}^*)^{-1} & B_{11}^{-*}B_{21} \\ \hline A_2 - B_{21}B_{11}^{-1}A_1 & B_{21}B_{11}^{-1} & B_{22}B_{22}^* \end{array}\right] \\ &= \mathcal{C}_{\mathcal{J}}\left(\left[\begin{array}{c|c} C' & 0 \\ \hline A' & B' \end{array}\right]\right), \end{aligned}$$

where $A' \in \mathbb{C}^{(n-k-m)\times(k+m)}$, $B' \in \mathbb{C}^{(n-k-m)\times(t-m)}$, and $C' \in \mathbb{C}^{(r+m)\times(k+m)}$ are given by

$$A' = \begin{bmatrix} A_2 - B_{21}B_{11}^{-1}A_1 & B_{21}B_{11}^{-1} \end{bmatrix}, \quad B' = B_{22}, \quad C' = \begin{bmatrix} C & 0 \\ -B_{11}^{-1}A_1 & B_{11}^{-1} \end{bmatrix}. \tag{6.12}$$

6.4.3 Case 3.

Case 3 is somewhat more complicated. We start from $A \in \mathbb{C}^{(n-k)\times k}$, $B \in \mathbb{C}^{(n-k)\times t}$, $C \in \mathbb{C}^{r\times k}$ and take $1 \le l \le k$ and $1 \le m \le n-k$, such that $l \le r$ and $m \le t$. For simplicity, we look at $\mathcal{J} = \{1, 2, \ldots, k-l\} \cup \{k+1, \ldots, k+m\}$. Let H and U be unitary matrices such that

$$BU = \begin{array}{c} \\ m \\ n-k-m \end{array} \begin{array}{c} \begin{array}{cc} m & t-m \end{array} \\ \begin{bmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix} \end{array}, \quad HC = \begin{array}{c} \\ r-l \\ l \end{array} \begin{array}{c} \begin{array}{cc} k-l & l \end{array} \\ \begin{bmatrix} C_{11} & 0 \\ C_{21} & C_{22} \end{bmatrix} \end{array}, \tag{6.13}$$

$$\text{and} \quad A = \begin{array}{c} \\ m \\ n-k-m \end{array} \begin{array}{c} \begin{array}{cc} k-l & l \end{array} \\ \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \end{array}.$$

In this case (6.8) is

$$\begin{aligned} X &= \mathcal{C}_{\mathcal{I}} \left(\left[\begin{array}{c|c} HC & 0 \\ \hline A & BU \end{array}\right] \right) \\ &= \left[\begin{array}{cc|cc} -C_{11}^*C_{11} - C_{21}^*C_{21} & -C_{21}^*C_{22} & A_{11}^* & A_{21}^* \\ -C_{22}^*C_{21} & -C_{22}^*C_{22} & A_{12}^* & A_{22}^* \\ \hline A_{11} & A_{12} & B_{11}B_{11}^* & B_{11}B_{21}^* \\ A_{21} & A_{22} & B_{21}B_{11}^* & B_{21}B_{21}^* + B_{22}B_{22}^* \end{array}\right]. \end{aligned} \tag{6.14}$$

From Lemma 1.2.5 we have $\mathcal{K} = \{k-l+1, \ldots, k, k+1, \ldots, k+m\}$ and the pivot submatrix whose inverse is required is the quasidefinite matrix

$$X_{\mathcal{K}\mathcal{K}} = \begin{bmatrix} -C_{22}^*C_{22} & A_{12}^* \\ A_{12} & B_{11}B_{11}^* \end{bmatrix}. \tag{6.15}$$

An inversion formula for quasidefinite matrices

It is not difficult to see that whenever a quasidefinite matrix is invertible, its inverse is quasidefinite, too [119, Thm. 1.1]. Hence, given A, B, C of conformal sizes, we can write

$$\begin{bmatrix} -C^*C & A^* \\ A & BB^* \end{bmatrix}^{-1} = \begin{bmatrix} -NN^* & K \\ K^* & L^*L \end{bmatrix} \tag{6.16}$$

for suitable matrices K, L, N. In this section, we describe a method to compute K, L, N directly from A, B, C. In principle, one can assemble the matrix in (6.16), invert

it, and then find the Cholesky factors of its diagonal blocks. However, this does not appear sound from a numerical point of view, since it means forming Gram matrices BB^* and C^*C and then factoring the corresponding blocks in the computed inverse (which may not be semidefinite due to numerical errors). It is a problem similar to the infamous normal equations for least-squares problems [55, Sec. 20.4]. The only condition appearing in Lemma 1.2.5 is that the pivot submatrix (6.15) is invertible and we wish to keep only that assumption for the existence of the PPT. Hence, formulae which rely on Schur complements [62, Sec. 0.7.3] cannot be used, since BB^* and C^*C are not guaranteed to have full rank (consider, e.g. the case $A = 1$, $B = C = 0$).

In the following, we present an alternative expression that relies heavily on unitary factorizations.

Theorem 6.4.1. *Let*

$$P = \begin{bmatrix} -C_{22}^* C_{22} & A_{12}^* \\ A_{12} & B_{11} B_{11}^* \end{bmatrix}, \quad A_{12} \in \mathbb{C}^{m \times l},\ B_{11} \in \mathbb{C}^{m \times m},\ C_{22} \in \mathbb{C}^{l \times l}$$

be an invertible matrix and let Q and H be unitary matrices such that

$$\begin{bmatrix} B_{11}^* \\ A_{12}^* \end{bmatrix} = Q \begin{bmatrix} R^* \\ 0 \end{bmatrix}, \quad Q = \begin{array}{c} \\ m \\ l \end{array} \begin{array}{c} \begin{array}{cc} m & l \end{array} \\ \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \end{array} \tag{6.17}$$

and

$$M = \begin{bmatrix} I_m & 0 \\ 0 & C_{22} \end{bmatrix} Q = H \begin{bmatrix} M_{11} & 0 \\ M_{21} & M_{22} \end{bmatrix}, \quad H = \begin{array}{c} \\ m \\ l \end{array} \begin{array}{c} \begin{array}{cc} m & l \end{array} \\ \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \end{array}. \tag{6.18}$$

Then,

1. *R and M_{22} are invertible.*
2. *We have*

$$P^{-1} = \begin{bmatrix} -NN^* & K \\ K^* & L^* L \end{bmatrix},$$

with

$$N = Q_{22} M_{22}^{-1}, \quad K = (Q_{21} - Q_{22} M_{22}^{-1} M_{21}) R^{-1}, \quad L = M_{11} R^{-1}.$$

3. The following relations hold:

$$C_{22}N = H_{22}, \qquad C_{22}K = H_{21}L, \tag{6.19}$$

$$LB_{11} = H_{11}^*, \qquad KB_{11} = -NH_{12}^*. \tag{6.20}$$

Proof. We use a few manipulations of quasidefinite matrices which are standard in the context of preconditioners for saddle-point matrices; see for instance [11, Sec. 10.4].

Note that P is the Schur complement of $-I_m$ in

$$T = \begin{bmatrix} -I_m & 0 & B_{11}^* \\ 0 & -C_{22}^*C_{22} & A_{12}^* \\ B_{11} & A_{12} & 0 \end{bmatrix},$$

so by the standard results on Schur complements T is nonsingular and

$$P^{-1} = \begin{bmatrix} 0 & I_l & 0 \\ 0 & 0 & I_m \end{bmatrix} T^{-1} \begin{bmatrix} 0 & 0 \\ I_l & 0 \\ 0 & I_m \end{bmatrix}.$$

Inserting factors $\hat{Q} = \operatorname{diag}(Q, I)$ and its inverse, we get

$$\begin{aligned} P^{-1} &= \begin{bmatrix} 0 & I_l & 0 \\ 0 & 0 & I_m \end{bmatrix} \hat{Q} \left(\hat{Q}^* \begin{bmatrix} -I_m & 0 & B_{11}^* \\ 0 & -C_{22}^*C_{22} & A_{12}^* \\ B_{11} & A_{12} & 0 \end{bmatrix} \hat{Q} \right)^{-1} \hat{Q}^* \begin{bmatrix} 0 & 0 \\ I_l & 0 \\ 0 & I_m \end{bmatrix} \\ &= \begin{bmatrix} Q_{21} & Q_{22} & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} * & * & R^* \\ * & * & 0 \\ R & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Q_{21}^* & 0 \\ Q_{22}^* & 0 \\ 0 & I \end{bmatrix}. \end{aligned}$$

The top–left 2×2 block which we have marked with asterisks is $-M^*M$, with M as in (6.18), so we can write it also as

$$P^{-1} = \begin{bmatrix} Q_{21} & Q_{22} & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} -M_{11}^*M_{11} - M_{21}^*M_{21} & -M_{21}^*M_{22} & R^* \\ -M_{22}^*M_{21} & -M_{22}^*M_{22} & 0 \\ R & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Q_{21}^* & 0 \\ Q_{22}^* & 0 \\ 0 & I \end{bmatrix}. \tag{6.21}$$

The middle matrix in (6.21) is equal to $\hat{Q}^*T\hat{Q}$, which is invertible. Hence R and M_{22} must be invertible, too, which proves our first statement. The inverse of this block

antitriangular matrix can be computed explicitly as

$$P^{-1} = \begin{bmatrix} Q_{21} & Q_{22} & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} 0 & 0 & R^{-1} \\ 0 & -M_{22}^{-1}M_{22}^{-*} & -M_{22}^{-1}M_{21}R^{-1} \\ R^{-*} & -R^{-*}M_{21}^{*}M_{22}^{-*} & R^{-*}M_{11}^{*}M_{11}R^{-1} \end{bmatrix} \begin{bmatrix} Q_{21}^{*} & 0 \\ Q_{22}^{*} & 0 \\ 0 & I \end{bmatrix}$$
$$= \begin{bmatrix} -Q_{22}M_{22}^{-1}M_{22}^{-*}Q_{22}^{*} & (Q_{21} - Q_{22}M_{22}^{-1}M_{21})R^{-1} \\ R^{-*}(Q_{21}^{*} - M_{21}^{*}M_{22}^{-*}Q_{22}^{*}) & R^{-*}M_{11}^{*}M_{11}R^{-1} \end{bmatrix} = \begin{bmatrix} -NN^{*} & K \\ K^{*} & L^{*}L \end{bmatrix},$$

which proves the second claim.

Expanding the multiplications in the second block column of (6.18), we get $C_{22}Q_{22} = H_{22}M_{22}$ and $C_{22}Q_{21} = H_{21}M_{11} + H_{22}M_{21}$, from which the two equations (6.19) follow easily. From the first block row of (6.17) we get $B_{11}^{*} = Q_{11}R^{*}$, and again from (6.18) we get

$$H^{*} \begin{bmatrix} I & 0 \\ 0 & C_{22} \end{bmatrix} = \begin{bmatrix} M_{11} & 0 \\ M_{21} & M_{22} \end{bmatrix} Q^{*},$$

whose first block column reads $H_{11}^{*} = M_{11}Q_{11}^{*}$, $H_{12}^{*} = M_{21}Q_{11}^{*} + M_{22}Q_{12}^{*}$. Putting together these relations, (6.20) follows. □

We now continue computing the factored version of the PPT in Case 3. Assuming that the matrix $X_{\mathcal{K}\mathcal{K}}$ from (6.15) is nonsingular, the symmetric principal pivot transform Y of X from (6.14) exists and we partition it as

$$Y = \left[\begin{array}{c|ccc} Y_{11} & Y_{21}^{*} & Y_{31}^{*} & Y_{41}^{*} \\ \hline Y_{21} & -Y_{22} & -Y_{32}^{*} & Y_{42}^{*} \\ Y_{31} & -Y_{32} & -Y_{33} & Y_{43}^{*} \\ \hline Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{array}\right]. \tag{6.22}$$

The middle block is $-X_{\mathcal{K}\mathcal{K}}^{-1}$ and from Theorem 6.4.1 defining K, L, N we have

$$X_{\mathcal{K}\mathcal{K}}^{-1} = \begin{bmatrix} -C_{22}^{*}C_{22} & A_{12}^{*} \\ A_{12} & B_{11}B_{11}^{*} \end{bmatrix}^{-1} = \begin{bmatrix} -NN^{*} & K \\ K^{*} & L^{*}L \end{bmatrix} = \begin{bmatrix} Y_{22} & Y_{32}^{*} \\ Y_{32} & Y_{33} \end{bmatrix}. \tag{6.23}$$

Lemma 6.4.2. *The remaining blocks of Y from (6.22) are given by*

$$\begin{aligned}
Y_{11} &= -C_{11}^*C_{11} - C_{21}^*C_{21} + C_{21}^*C_{22}NN^*C_{22}^*C_{21} + A_{11}^*K^*C_{22}^*C_{21} \\
&\quad + C_{21}^*C_{22}KA_{11} - A_{11}^*L^*LA_{11}, \\
Y_{21} &= NN^*C_{22}^*C_{21} + KA_{11}, \\
Y_{31} &= -K^*C_{22}^*C_{21} + L^*LA_{11}, \\
Y_{41} &= A_{21} - A_{22}NN^*C_{22}^*C_{21} - A_{22}KA_{11} + B_{21}B_{11}^*K^*C_{22}^*C_{21} - B_{21}B_{11}^*L^*LA_{11}, \\
Y_{42} &= -A_{22}NN^* + B_{21}B_{11}^*K^*, \\
Y_{43} &= A_{22}K + B_{21}B_{11}^*L^*L, \\
Y_{44} &= B_{21}B_{21}^* + B_{22}B_{22}^* + A_{22}NN^*A_{22}^* - B_{21}B_{11}^*K^*A_{22}^* \\
&\quad - A_{22}KB_{11}B_{21}^* - B_{21}B_{11}^*L^*LB_{11}B_{21}^*.
\end{aligned}$$

Proof. We get the above formulae after some tedious but straightforward algebra from the PPT formulae (1.13), the expression (6.14) for X, and (6.23). □

The sign change matrix D from Lemma 1.2.5 is $D = \operatorname{diag}(I_{k-l}, -I_l, I_m, I_{n-k-m})$, and we finally have

$$X' = DYD = \left[\begin{array}{c|ccc} Y_{11} & -Y_{21}^* & Y_{31}^* & Y_{41}^* \\ \hline -Y_{21} & -Y_{22} & Y_{32}^* & -Y_{42}^* \\ Y_{31} & Y_{32} & -Y_{33} & Y_{43}^* \\ \hline Y_{41} & -Y_{42} & Y_{43} & Y_{44} \end{array}\right],$$

where the blocks are defined in (6.23) and Lemma 6.4.2. What remains is to show that $X' = \mathcal{C}_{\mathcal{J}}\left(\left[\begin{smallmatrix} C' & 0 \\ A' & B' \end{smallmatrix}\right]\right)$, where $\mathcal{J} = \{1, 2, \ldots, k-l\} \cup \{k, k+1, \ldots, k+m\}$, by finding the factors $B' \in \mathbb{C}^{(n-k+l-m)\times(t-m+l)}$ and $C' \in \mathbb{C}^{(r-l+m)\times(k-l+m)}$ such that

$$\begin{bmatrix} -Y_{22} & -Y_{42}^* \\ -Y_{42} & Y_{44} \end{bmatrix} = B'(B')^* \quad \text{and} \quad \begin{bmatrix} -Y_{11} & -Y_{31}^* \\ -Y_{31} & Y_{33} \end{bmatrix} = (C')^*C'. \tag{6.24}$$

The factor $A' \in \mathbb{C}^{(n-k+l-m)\times(k+m-l)}$ is given by

$$A' = \begin{bmatrix} -Y_{21} & Y_{32}^* \\ Y_{41} & Y_{43} \end{bmatrix}. \tag{6.25}$$

Lemma 6.4.3. *The equalities (6.24) hold with*

$$B' = \begin{bmatrix} N & 0 \\ B_{21}H_{12} + A_{22}N & B_{22} \end{bmatrix} \quad \textit{and} \quad C' = \begin{bmatrix} C_{11} & 0 \\ H_{21}^*C_{21} - LA_{11} & L \end{bmatrix}, \tag{6.26}$$

where H_{12}, H_{21}, L, and N are defined in Theorem 6.4.1.

Proof. Define $Z = B_{21}H_{12} + A_{22}N$. Then

$$B'(B')^* = \begin{bmatrix} NN^* & NZ^* \\ ZN^* & ZZ^* + B_{22}B_{22}^* \end{bmatrix}$$

and we only need to check that these blocks match the blocks specified in (6.24). From (6.23) we have $NN^* = -Y_{22}$ and from (6.20) we get

$$ZN^* = B_{21}H_{12}N^* + A_{22}NN^* = -B_{21}B_{11}^*K^* + A_{22}NN^* = -Y_{42},$$

where we have used the formula for Y_{42} from Lemma 6.4.2 for the last equality. What remains is to show that

$$ZZ^* + B_{22}B_{22}^* = Y_{44}.$$

Multiplying out the left hand side and using (6.20) we see that the above equality holds if and only if

$$B_{21}H_{12}H_{12}^*B_{21}^* = B_{21}B_{21}^* - B_{21}H_{11}H_{11}^*B_{21}^*,$$

which is true because H from (6.18) is a unitary matrix and so $H_{11}H_{11}^* + H_{12}H_{12}^* = I$.

The proof involving the matrix C' uses (6.19) and is identical to the above. □

To summarize the results for Case 3, we have an $\mathcal{I}$-semidefinite matrix $X = \mathcal{C}_{\mathcal{I}}([\begin{smallmatrix} HC & 0 \\ A & BU \end{smallmatrix}])$ for $\mathcal{I} = \{1, \ldots, k\}$ and the factors A, BU, and HC as in (6.13), and we wish to transform it into a $\mathcal{J}$-semidefinite matrix X' for $\mathcal{J} = \{1, \ldots, k-l\} \cup \{k+1, \ldots, k+m\}$. Providing that the matrix $X_{\mathcal{K}\mathcal{K}}$ from (6.15) is invertible and its inverse defined by (6.23), X' can be represented as $X' = \mathcal{C}_{\mathcal{J}}\left(\left[\begin{smallmatrix} C' & 0 \\ A' & B' \end{smallmatrix}\right]\right)$, where the factor A' is given by (6.25) and the factors B' and C' are defined in Lemma 6.4.3.

6.4.4 Formulae for arbitrary index sets

Using a suitable permutation, we can reduce the general case (with arbitrary $\mathcal{I}$ and $\mathcal{J}$) to the ones we treated above. For instance, we show how to adapt Case 1 (the other two are analogous). Let P be the permutation matrix associated to a permutation π that maps

$$\pi(\mathcal{I} \cap \mathcal{J}) = \{1, 2, \ldots, k-l\},$$
$$\pi(\mathcal{I} \cap \mathcal{J}^c) = \{k-l+1, \ldots, k\},$$
$$\pi(\mathcal{I}^c) = \{k+1, \ldots, n\},$$

where $k = \operatorname{card}(\mathcal{I})$, $l = \operatorname{card}(\mathcal{I} \cap \mathcal{J}^c)$. Then the matrix

$$\hat{X} = PXP^T = \begin{bmatrix} X_{\mathcal{I}\mathcal{I}} & X_{\mathcal{I}\mathcal{I}^c} \\ X_{\mathcal{I}^c\mathcal{I}} & X_{\mathcal{I}^c\mathcal{I}^c} \end{bmatrix} = \begin{bmatrix} -C^*C & A^* \\ A & BB^* \end{bmatrix}$$

is $\{1, 2, \ldots, k\}$-semidefinite and as in Section 6.4.1 we get A', B', and C' defined by (6.10) as the factors of an $\{1, 2, \ldots, k-l\}$-semidefinite matrix

$$\hat{X}' = \begin{bmatrix} -(C')^*C' & (A')^* \\ A' & B'(B')^* \end{bmatrix}.$$

Then the $\mathcal{J}$-semidefinite matrix X' is obtained as $X' = P^T \hat{X}' P$.

6.5 PPTs with bounded elements

We now use the factor-based formulae for the PPT derived in Section 6.4 to compute an optimal permuted Riccati basis for a Lagrangian semidefinite subspace. From [79, Thm. 3.4], which is here stated as the final part of Theorem 1.2.2, we know that for a Lagrangian subspace $\operatorname{Im} U$ there exists at least one optimal permuted Riccati representation with X_{opt} satisfying

$$|(X_{\text{opt}})_{ij}| \leq \begin{cases} 1, & \text{if } \ i = j, \\ \sqrt{2}, & \text{otherwise.} \end{cases}$$

The above inequality is sharp, as can be seen from the example [79, Sec. 3] where $U = \left[\begin{smallmatrix} I_2 \\ X \end{smallmatrix}\right]$, $X = \left[\begin{smallmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 1 \end{smallmatrix}\right]$. However, a stronger version can be obtained under the additional hypothesis that $\operatorname{Im} U$ is Lagrangian semidefinite (instead of merely Lagrangian).

Theorem 6.5.1. *Let $U \in \mathbb{C}^{2n\times n}$ be such that $\operatorname{Im} U$ is Lagrangian semidefinite. Then, there exists $\mathcal{I}_{\text{opt}} \subseteq \{1, 2, \ldots, n\}$ such that $U \sim \mathcal{G}_{\mathcal{I}_{\text{opt}}}(X)$ and*

$$|x_{ij}| \leq 1 \quad \forall i, j. \tag{6.27}$$

Proof. Since $\operatorname{Im} U$ is Lagrangian, from the proof of [79, Thm. 3.4] it follows that there exists an index set $\mathcal{I}$ defining the symplectic swap $\Pi_{\mathcal{I}}$ such that $U \sim \mathcal{G}_{\mathcal{I}}(X)$, $X \in \mathbb{C}^{n\times n}$ is Hermitian, and

$$|x_{ii}| \leq 1, \quad \left| \det \begin{bmatrix} x_{ii} & x_{ij} \\ x_{ji} & x_{jj} \end{bmatrix} \right| \leq 1, \quad i, j = 1, 2, \ldots, n. \tag{6.28}$$

In addition, since $\operatorname{Im} U$ is Lagrangian semidefinite, from Theorem 6.2.2 it follows that X is $\mathcal{I}$-semidefinite. We prove that the choice $\mathcal{I}_{\text{opt}} = \mathcal{I}$ satisfies (6.27).

For $i = j$ this trivially follows from (6.28). When $i \neq j$, we distinguish four cases.

Case A: $i, j \in \mathcal{I}$ The block $X_{\mathcal{I}\mathcal{I}}$ is negative semidefinite so its submatrix $(X)_{\{i,j\}\{i,j\}}$ is negative semidefinite, too, and this implies

$$|x_{ij}|^2 = x_{ij}x_{ji} \leq |x_{ii}||x_{jj}| \leq 1.$$

Case B: $i, j \notin \mathcal{I}$ The proof is analogous to Case A, since $X_{\mathcal{I}^c\mathcal{I}^c}$ is positive semidefinite.

Case C: $i \in \mathcal{I}, j \notin \mathcal{I}$ By semidefiniteness it follows that $-1 \leq x_{ii} \leq 0$ and $0 \leq x_{jj} \leq 1$. Moreover, by the 2×2 case from (6.28) we get

$$|x_{ii}x_{jj} - |x_{ij}|^2| = |x_{ii}|x_{jj} + |x_{ij}|^2 \leq 1,$$

and hence $|x_{ij}| \leq 1$.

Case D: $i \notin \mathcal{I}, j \in \mathcal{I}$ The proof is analogous to Case C by swapping i and j. □

6.5.1 The optimization algorithm

Algorithm 6.5.2, which is a modified version of [79, Alg. 2], can be used to compute $\mathcal{I}_{\text{opt}}$ such that (6.27) holds. In each step, the original algorithm performs one symmetric PPT, where the pivot set consists of either an index of the diagonal element with the largest modulus, providing that this value is greater than a threshold $\tau_1 \geq 1$ or, if all diagonal elements are less than τ_1 in magnitude, then the pivot set contains indices of the off-diagonal element of largest modulus, if this is greater than $\tau_2 \geq \sqrt{2}$.

Note that due to Theorem 6.5.1 we need only one threshold value $\tau \geq 1$. Our algorithm is based on the following observations about the location of the element of the maximum modulus which defines the pivot set for the PPT. Since the blocks $X_{\mathcal{I}\mathcal{I}}$ and $X_{\mathcal{I}^c\mathcal{I}^c}$ of an $\mathcal{I}$-semidefinite matrix X with factors A, B, C are semidefinite, if the entry of X with maximum modulus occurs on the diagonal of X then

- either it is in the block $X_{\mathcal{I}\mathcal{I}} = -C^*C$: then it is the squared norm $\|C_{:,j}\|^2$ of a column of C;

- or it is in the block $X_{\mathcal{I}^c\mathcal{I}^c} = BB^*$: then it is the squared norm $\|B_{i,:}\|^2$ of a row of B.

Moreover, if all diagonal elements of X are reduced below τ so that the pivot set is defined by the indices of some off-diagonal element of X, due to definiteness, all off-diagonal elements of $X_{\mathcal{I}\mathcal{I}}$ and $X_{\mathcal{I}^c\mathcal{I}^c}$ will not exceed τ, and hence in this case we need only look in the block A for the element of maximum modulus.

Once the maximum modulus of elements of X is determined, when it exceeds τ, in each of the three cases we can perform a PPT that strictly reduces this maximal entry. For computational efficiency, the algorithm first attempts to find a pivot index among the columns of the factor C, when there are no such pivots, it attempts to find a pivot index among the rows of the factor B, and finally, if no diagonal pivots are found, it looks for an off-diagonal pivot indices among the elements of $|A_{i,j}|$. We repeat this procedure until all entries are smaller than τ.

Notice the use of the control flow instructions break and continue, defined as in C or MATLAB: the first exits prematurely from the for loop, the second resumes execution from its next iteration.

The algorithm terminates since each PPT uses a pivot matrix with the modulus of the determinant at least τ and hence $|\det X|$ is reduced by a factor at least τ at each step. This argument is similar to, but slightly different from, the one used in [79, Thm. 5.2], where a determinant argument is applied to U_1 in (1.9) rather than X.

Algorithm 6.5.2. *Given $\mathcal{I}_{\text{in}} \subseteq \{1, 2, \ldots, n\}$, and factors $A_{\text{in}}, B_{\text{in}}, C_{\text{in}}$ of an $\mathcal{I}_{\text{in}}$-semidefinite matrix $X_{\text{in}} = X_{\text{in}}^* \in \mathbb{C}^{n\times n}$ as defined by (6.7), and a threshold $\tau \geq 1$, this algorithm computes a bounded permuted Riccati basis $\mathcal{I}_{\text{out}} \subseteq \{1, 2, \ldots, n\}$, and factors $A_{\text{out}}, B_{\text{out}}, C_{\text{out}}$ of an $\mathcal{I}_{\text{out}}$-semidefinite matrix $X_{\text{out}} = X_{\text{out}}^* \in \mathbb{C}^{n\times n}$ such that $\mathcal{G}_{\mathcal{I}_{\text{in}}}(X_{\text{in}}) \sim \mathcal{G}_{\mathcal{I}_{\text{out}}}(X_{\text{out}})$ and $|(X_{\text{out}})_{ij}| \leq \tau$ for each i, j. The functions g_A, g_B, g_C specify the mapping between 'local' indices in A, B, C and 'global' indices in the full matrix X.*

1 $A = A_{\text{in}}$, $B = B_{\text{in}}$, $C = C_{\text{in}}$, $\mathcal{I} = \mathcal{I}_{\text{in}}$

2 for it $= 1, 2, \ldots,$ max_iterations
3 $\quad \hat{\jmath} = \arg\max \|C_{:,j}\|^2$
4 $\quad$ if $\|C_{:,\hat{\jmath}}\|^2 > \tau$
5 $\quad\quad$ use the formulae in Case 1 in Section 6.4.1, with $\mathcal{J} = \mathcal{I} \setminus \{g_C(\hat{\jmath})\}$,
$\quad\quad$ to update $(A, B, C, \mathcal{I}) \leftarrow (A', B', C', \mathcal{J})$
6 $\quad\quad$ continue
7 $\quad$ end
8 $\quad \hat{\imath} = \arg\max \|B_{i,:}\|^2$
9 $\quad$ if $\|B_{\hat{\imath},:}\|^2 > \tau$
10 $\quad\quad$ use the formulae in Case 2 in Section 6.4.2, with $\mathcal{J} = \mathcal{I} \cup \{g_B(\hat{\imath})\}$,
$\quad\quad$ to update $(A, B, C, \mathcal{I}) \leftarrow (A', B', C', \mathcal{J})$
11 $\quad\quad$ continue
12 $\quad$ end
13 $\quad \hat{\imath}, \hat{\jmath} = \arg\max |A_{i,j}|$
14 $\quad$ if $|A_{\hat{\imath},\hat{\jmath}}| > \tau$
15 $\quad\quad$ use the formulae in Case 3 in Section 6.4.3, with $\mathcal{J} = (\mathcal{I} \setminus \{g_A(\hat{\jmath})\}) \cup \{g_A(\hat{\imath})\}$,
$\quad\quad$ to update $(A, B, C, \mathcal{I}) \leftarrow (A', B', C', \mathcal{J})$
16 $\quad\quad$ continue
17 $\quad$ end
18 $\quad$ break
19 end
20 $A_{\text{out}} = A$, $B_{\text{out}} = B$, $C_{\text{out}} = C$, $\mathcal{I}_{\text{out}} = \mathcal{I}$

6.5.2 Special formulae for the scalar cases $l = 1$, $m = 1$

The pivot sets used in the algorithm have at most 2 elements, so some simplifications can be done to the general formulae (6.10), (6.12), (6.25), and (6.26).

For Case 1, the factor partition (6.9) is (up to the ordering of indices)

$$A = \begin{matrix} \\ n-k \end{matrix} \begin{matrix} \overset{k-1}{} & \overset{1}{} \\ \end{matrix} \begin{bmatrix} A_1 & a \end{bmatrix}, \quad HC = \begin{matrix} r-1 \\ 1 \end{matrix} \begin{matrix} \overset{k-1}{\phantom{C_{11}}} & \overset{1}{} \\ \end{matrix} \begin{bmatrix} C_{11} & 0 \\ c^* & \gamma \end{bmatrix},$$

and the PPT that gives the updated factors in (6.10) for $\mathcal{J} = \{1, \ldots, k-1\}$ is

$$\left[\begin{array}{c|c} HC & 0 \\ \hline A & B \end{array}\right] = \left[\begin{array}{cc|c} C_{11} & 0 & 0 \\ c^* & \gamma & 0 \\ \hline A_1 & a & B \end{array}\right] \mapsto \left[\begin{array}{c|cc} C_{11} & 0 & 0 \\ \hline -\gamma^{-1}c^* & \gamma^{-1} & 0 \\ A_1 - \gamma^{-1}ac^* & \gamma^{-1}a & B \end{array}\right] = \left[\begin{array}{c|c} C' & 0 \\ \hline A' & B' \end{array}\right].$$

Similarly, for Case 2 the starting factors (6.11) are now partitioned as

$$A = \begin{array}{c} \\ 1 \\ n-k-1 \end{array} \begin{array}{c} k \\ \left[\begin{array}{c} a^* \\ A_2 \end{array}\right] \end{array}, \quad BU = \begin{array}{c} \\ 1 \\ n-k-1 \end{array} \begin{array}{c} 1 \quad t-1 \\ \left[\begin{array}{cc} \beta & 0 \\ b & B_{22} \end{array}\right] \end{array},$$

and the updated factors (6.12) for $\mathcal{J} = \{1, \ldots, k, k+1\}$ correspond to the PPT

$$\left[\begin{array}{c|c} C & 0 \\ \hline A & BU \end{array}\right] = \left[\begin{array}{c|cc} C & 0 & 0 \\ \hline a^* & \beta & 0 \\ A_2 & b & B_{22} \end{array}\right] \mapsto \left[\begin{array}{cc|c} C & 0 & 0 \\ -\beta^{-1}a^* & \beta^{-1} & 0 \\ \hline A_2 - \beta^{-1}ba^* & \beta^{-1}b & B_{22} \end{array}\right] = \left[\begin{array}{c|c} C' & 0 \\ \hline A' & B' \end{array}\right].$$

For Case 3 the initial partition of factors (6.13) is

$$A = \begin{array}{c} \\ 1 \\ n-k-1 \end{array} \begin{array}{c} k-1 \quad 1 \\ \left[\begin{array}{cc} a^* & \alpha \\ A_{21} & d \end{array}\right] \end{array}, \quad BU = \begin{array}{c} \\ 1 \\ n-k-1 \end{array} \begin{array}{c} 1 \quad t-1 \\ \left[\begin{array}{cc} \beta & 0 \\ b & B_{22} \end{array}\right] \end{array}, \quad HC = \begin{array}{c} r-1 \\ 1 \end{array} \begin{array}{c} k-1 \quad 1 \\ \left[\begin{array}{cc} C_{11} & 0 \\ c^* & \gamma \end{array}\right] \end{array}.$$

The PPT for the updated factors for $\mathcal{J} = \{1, \ldots, k-1\} \cup \{k+1\}$ has the pivot set $\mathcal{K} = \{k, k+1\}$ and it requires the inverse of the 2×2 matrix $X_{\mathcal{K}\mathcal{K}} = \left[\begin{smallmatrix} -|\gamma|^2 & \overline{\alpha} \\ \alpha & |\beta|^2 \end{smallmatrix}\right]$, which can be computed explicitly as

$$X_{\mathcal{K}\mathcal{K}}^{-1} = \frac{1}{\Delta^2}\begin{bmatrix} -|\beta|^2 & \overline{\alpha} \\ \alpha & |\gamma|^2 \end{bmatrix}, \quad \Delta = \sqrt{|\alpha|^2 + |\beta\gamma|^2}.$$

Therefore, we can write $X_{\mathcal{K}\mathcal{K}}^{-1} = \begin{bmatrix} -NN^* & K \\ K^* & L^*L \end{bmatrix}$ for $N = \beta/\Delta$, $K = \overline{\alpha}/\Delta^2$ and $L = \gamma/\Delta$. Lemma 6.4.3 gives

$$B' = \begin{bmatrix} \beta/\Delta & 0 \\ (\beta d - \alpha b)/\Delta & B_{22} \end{bmatrix} \quad \text{and} \quad C' = \begin{bmatrix} C_{11} & 0 \\ (\alpha c^* - \gamma a^*)/\Delta & \gamma/\Delta \end{bmatrix}.$$

Finally, the factor update formula is

$$\left[\begin{array}{c|c} HC & 0 \\ \hline A & BU \end{array}\right] = \left[\begin{array}{cc|cc} C_{11} & 0 & 0 & 0 \\ c^* & \gamma & 0 & 0 \\ \hline a^* & \alpha & \beta & 0 \\ A_{21} & d & b & B_{22} \end{array}\right]$$

$$\mapsto \left[\begin{array}{cc|cc} C_{11} & 0 & 0 & 0 \\ (\alpha c^* - \gamma a^*)/\Delta & \gamma/\Delta & 0 & 0 \\ \hline -Y_{21} & Y_{32}^* & \beta/\Delta & 0 \\ Y_{41} & Y_{43} & (\beta d - \alpha b)/\Delta & B_{22} \end{array}\right] = \left[\begin{array}{c|c} C' & 0 \\ \hline A' & B' \end{array}\right],$$

where

$$\begin{aligned} Y_{21} &= (\overline{\gamma}|\beta|^2 c^* + \overline{\alpha} a^*)/\Delta^2, \\ Y_{41} &= A_{21} - \frac{1}{\Delta^2}\begin{bmatrix} d & b \end{bmatrix}\begin{bmatrix} \overline{\gamma}|\beta|^2 & \overline{\alpha} \\ -\alpha\overline{\beta}\overline{\gamma} & \overline{\beta}|\gamma|^2 \end{bmatrix}\begin{bmatrix} c^* \\ a^* \end{bmatrix}, \\ Y_{32} &= \overline{\alpha}/\Delta^2, \\ Y_{43} &= (\overline{\alpha} d + |\gamma|^2 \overline{\beta} b)/\Delta^2. \end{aligned}$$

6.6 Numerical experiments

In our first experiment with Algorithm 6.5.2 we use `randn` to generate random factors $C \in \mathbb{R}^{14\times 14}$, $A \in \mathbb{R}^{16\times 14}$, $B \in \mathbb{R}^{16\times 16}$, and a random index set $\mathcal{I}$ with $\operatorname{card}(\mathcal{I}) = 14$ defining the $\mathcal{I}$-semidefinite matrix X of order 30. The threshold parameter for the optimization algorithm is $\tau = 1.5$. In Figure 6.1 we display a color plot of the matrix $|X|$, where $(|X|)_{ij} = |x_{ij}|$ at the start of the optimization procedure, after 10 and 20 iterations, and the final matrix. The algorithm took 31 iterations and produced the matrix X with $\max|x_{ij}| = 0.46$ and $\operatorname{card}(\mathcal{I}_{\text{opt}}) = 16$. The effect of semidefinite blocks on the reduction can be seen in plots (b) and (c) of Figure 6.1, where the dark red stripes that appear are due to the fact that whenever a diagonal pivot is chosen (Case 1 or 2), all elements in the corresponding row and column of the matrices $-C^*C$ or BB^* are also reduced below τ.

For the same example, Figure 6.2 displays $\max_{i,j}|x_{ij}|$ and $|\det X|$ during the iterations. The quantity $\max_{i,j}|x_{ij}|$ is not guaranteed to decrease with iterations and we

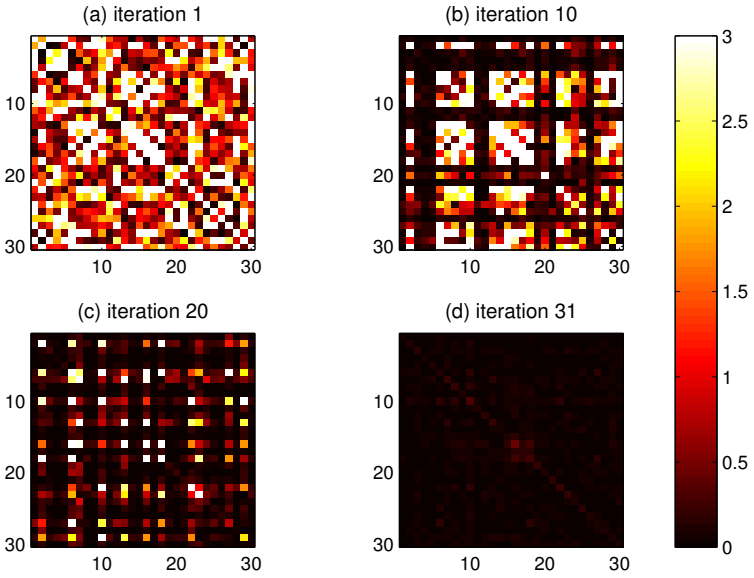

Figure 6.1: Snapshots of $|X|$ for the starting matrix, iterations 10 and 20, and the final matrix for Algorithm 6.5.2 applied to a random matrix X of order 30 with the factors $C \in \mathbb{R}^{14\times14}$, $A \in \mathbb{R}^{16\times14}$, and $B \in \mathbb{R}^{16\times16}$.

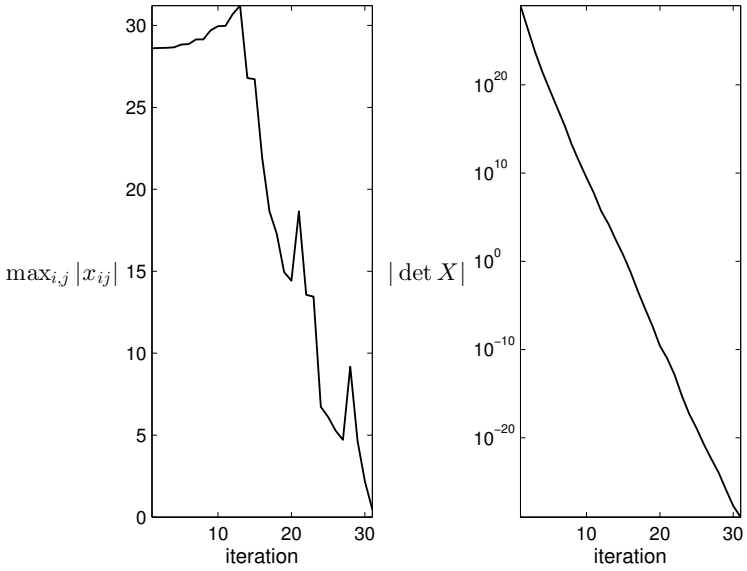

Figure 6.2: The changes in $\max_{i,j} |x_{ij}|$ and $|\det X|$ for Algorithm 6.5.2 applied to a random matrix X of order 30 with the factors $C \in \mathbb{R}^{14\times14}$, $A \in \mathbb{R}^{16\times14}$, and $B \in \mathbb{R}^{16\times16}$. In the figure on the right, a base-10 logarithmic scale is used for the y-axis.

see this behaviour on the left plot but $|\det X|$ must decrease with each iteration as we explained in Section 6.5.1, and this is evident in the log-lin graph on the right.

We next use the matrices from the examples in the benchmark test set [25] detailed in section 1.4 to construct a quasidefinite matrix X to which we then apply Algorithm 6.5.2 with the threshold $\tau = 1.5$. Each example contains factors (cf. Section 6.3) A, $G = G^T$, $Q = Q^T \in \mathbb{R}^{k\times k}$, $B \in \mathbb{R}^{k\times t}$, $C \in \mathbb{R}^{r\times k}$, $R = R^T \in \mathbb{R}^{t\times t}$, and

$\widetilde{Q} = \widetilde{Q}^T \in \mathbb{R}^{r\times r}$, with $r, t \leq k$, which define the Hamiltonian matrix

$$\mathcal{H} = \begin{bmatrix} A & -G \\ -Q & -A^T \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -C^T\widetilde{Q}C & -A^T \end{bmatrix}.$$

From these factors we construct the quasidefinite matrix

$$X = \begin{bmatrix} -Q & A^T \\ A & G \end{bmatrix} = \begin{bmatrix} -C_f^T C_f & A^T \\ A & B_f B_f^T \end{bmatrix},$$

where $B_f = BR_R^{-1} \in \mathbb{R}^{k\times t}$, $C_f = R_{\widetilde{Q}}C \in \mathbb{R}^{r\times k}$, and R_R and $R_{\widetilde{Q}}$ are the upper triangular Cholesky factors of the matrices R and $\widetilde{Q}$, respectively. Moreover, the matrix $\widetilde{Q} = \left[\begin{smallmatrix} 9 & 6 \\ 6 & 4 \end{smallmatrix}\right]$ in Example 2 in [25] is singular positive semidefinite and therefore we use $R_{\widetilde{Q}} = \left[\begin{smallmatrix} 3 & 2 \\ 0 & 0 \end{smallmatrix}\right]$ as its Cholesky factor. The pair $(\mathcal{I}, X)$, with $\mathcal{I} = \{1, 2, \ldots, k\}$, identifies a Lagrangian subspace of $\mathbb{C}^{4k}$ which is associated with the Hamiltonian pencil (6.3), as described in Section 6.3.

This construction eliminates Examples 3, 4, 17, and 18 from [25] due to the indefiniteness of the matrix $\widetilde{Q}$, and consequently the matrix Q, since we cannot form a quasidefinite matrix X from these factors.

In Table 6.1, for each of the remaining examples we present the dimensions k, t, and r defining the factors C_f, A, and B_f of the matrix X, the number of iterations `it` the optimization took, the 2-norm condition number κ of the starting matrix $\mathcal{G}_{\mathcal{I}}(X)$, the maximum modulus of the elements in X and the computed optimal reduced matrix X_{opt}, and the subspace distance between $\mathcal{G}_{\mathcal{I}}(X)$ and $\mathcal{G}_{\mathcal{I}_{\text{opt}}}(X_{\text{opt}})$ computed by MATLAB's `subspace`.

Small values for the subspaces distance indicate that the algorithm produced a representation of the same subspace associated with $\mathcal{G}_{\mathcal{I}}(X)$, which happens in almost all examples. The largest value for this quantity corresponds to the Example 22 where the starting representation $\mathcal{G}_{\mathcal{I}}(X)$ is ill-conditioned. Several examples already had X with elements bounded in modulus by 1 but we include them for completeness. In all other examples, the algorithm achieved the goal of reducing the modulus of all elements in X_{opt} below the threshold τ and the number of iterations required to do this was in general not large.

We also note that the factors B_f and C_f could have been formed from the matrices G and Q, for example by taking their Cholesky factors or semidefinite square roots.

Table 6.1: Algorithm 6.5.2 applied to the matrices defining the test examples from [25].

Ex.	k	t	r	$\kappa(\mathcal{G}_{\mathcal{I}}(X))$	Subspace dist.	$\max \lvert x_{ij} \rvert$	$\max \lvert (X_{\text{opt}})_{ij} \rvert$	it
1	2	1	2	2.41	4.71e-16	2.000	1.000	2
2	2	1	2	1.62e1	1.31e-15	9.000	9.231e-1	3
5	9	3	9	2.16e2	5.60e-15	1.472e2	7.961e-1	13
6	30	3	5	1.44e8	3.47e-13	1.440e8	1.377	29
7	2	1	1	2.03	7.67e-16	2.000	1.000	2
8	2	1	1	2.72	7.28e-16	2.000	1.000	2
9	2	2	1	1.01e4	1.99e-13	1.000e4	1.000e-1	2
10	2	2	1	1.01e6	4.07e-11	1.000e6	1.000e-1	3
11	2	1	2	1.62	4.71e-16	1.000	1.000	1
12	2	1	2	7.07e5	5.08e-16	1.000e6	1.000	2
13	2	1	2	1.41	4.71e-16	1.000	1.000	1
14	2	2	2	1.91	7.25e-16	2.000	4.000e-1	3
15	2	2	2	2.75	1.04e-15	1.000	1.000	1
16	2	2	2	2.75	1.86e-15	1.000	1.000	1
19	3	3	3	1.91	1.22e-15	2.333	1.486	3
20	3	3	3	3.54	1.12e-15	2.333e6	7.037e-7	7
21	4	1	2	1.91	1.60e-15	1.000	1.000	1
22	4	1	2	1.00e12	1.16e-10	1.000e12	1.000	3
23	4	1	1	4.16	1.09e-15	1.000	1.000	1
24	4	1	1	4.24	9.11e-16	1.000	1.000	1
25	77	39	38	1.00e1	6.44e-15	1.000e1	1.000	39
26	237	119	118	1.00e1	9.88e-15	1.000e1	1.000	119
27	397	199	198	1.00e1	1.31e-14	1.000e1	1.000	199
28	8	8	8	3.00	1.68e-15	2.000	1.400	5
29	64	64	64	3.00	6.11e-15	2.000	1.400	33
30	21	1	1	1.00	6.76e-16	1.000	1.000	1
31	21	1	1	1.00e2	6.76e-16	1.000e2	1.000	2
32	100	1	1	1.22e3	1.84e-13	4.481e2	9.985e-1	200
33	60	2	60	2.41	6.75e-15	1.000	1.000	1

We chose not to do this not only because B and C are readily available, but also since in most examples G and/or Q are singular, and moreover such B_f and C_f would be square, while those computed from B and C are rectangular, often with very small number of columns and rows, respectively.

Chapter 7

Summary

The worst thing you can do to a problem is solve it completely.
—Daniel Kleitman

The final chapter summarizes the work presented in this thesis and our findings, and describes a few further research lines.

In Chapter 3 we aimed to develop an alternative to the popular, but relatively expensive, approach of replacing the given matrix by the nearest positive semidefinite matrix or nearest correlation matrix. The motivation for this work was the growing number of applications in which matrices that are supposed to be (semi)definite turn out to be indefinite. We have shown that shrinking is an attractive way of restoring definiteness. The method is flexible as it allows the practitioner to choose a target matrix that best serves the needs of the application; all that is required is to make sure that the chosen matrix is positive semidefinite or, in the correlation matrix case, a correlation matrix. We have described how to define a target matrix in the case of fixed diagonal blocks, as occur in stress testing, for example and have shown that shrinking can also take advantage of this structure. Weighting is a popular feature of the nearest correlation matrix methods used in practice and we have shown that with shrinking weights can be incorporated in the target matrix without any effect on the solution techniques.

Shrinking can be achieved in at least three different ways, all of which are straightforward to implement. Of our three shrinking methods we favor the bisection and generalized eigenvalue methods. Bisection is perhaps preferable for convergence tolerances of 10^{-6} and larger, whereas the generalized eigenvalue method is preferable for more stringent tolerances, since its cost is essentially independent of the tolerance.

Bisection also has the advantage that it produces a numerically semidefinite matrix (one for which the Cholesky factorization succeeds).

The problems for which we believe shrinking is of interest range in size from order 10–100, as arise for example in foreign exchange trading, and which may need to be solved thousands of times in a simulation, to orders in the thousands or millions, as for example in bioinformatics [101]. In the case of invalid correlation matrices an attractive feature of shrinking is that it is at least an order of magnitude faster than computing the nearest correlation matrix. This is due to the fact that computing the nearest correlation matrix requires at least several full eigenvalue decompositions, while we can completely avoid computing any eigenvalues in the bisection method and need to compute only one for the generalized eigenvalue method.

Our analysis did not assume any special structure for the initial matrix M_0, except in the fixed block case, but in practice there are many structured covariance and correlation matrix problems [73]; the tridiagonal structure seems to be quite popular in some areas [105]. It would therefore be of interest to analyze shrinking in structured cases requiring structured targets, particularly since computing the nearest correlation matrix in general does not preserve zero patterns or block structure of the matrix.

Chapter 4 is the first thorough treatment of upper and lower bounds for the distance of a symmetric matrix A to the nearest correlation matrix. For the most common case in practice, in which A is indefinite with unit diagonal and $|a_{ij}| \leq 1$ for $i \neq j$, we have obtained upper bounds (4.8), (4.12), and (4.13) that differ from the lower bound (4.3) by a factor at most $1 + n\sqrt{n}$. For the sharpest bound (4.8) we found the ratio to be always less than 5 in our experiments with matrices of dimension up to 3120, so the upper bound was demonstrably of the correct order of magnitude in every case. The cost of computing the pair (4.3) and (4.8) is $17n^3/6$ flops, which is substantially less than the $70n^3/3$ or more flops required to compute $\mathrm{ncm}(A)$ by the preconditioned Newton algorithm of [18].

The upper bound (4.13) based on shrinking has about half the cost of (4.8) and, while less sharp than (4.8), it still performed well in our tests.

The modified Cholesky bound (4.20) has the attraction that it provides an inexpensive test for definiteness ($n^3/3$ flops) along with an upper bound (costing another $n^3/3$ flops) that, while sometimes two orders of magnitude larger than (4.8), can still

provide useful information.

We conclude that our bounds are well suited to gauging the size of $\mathrm{d}_{\mathrm{corr}}(A)$. The information they provide enables a user to identify an invalid correlation matrix relatively cheaply and to decide whether to revisit its construction or proceed to compute a replacement directly from it. A natural replacement is the nearest correlation matrix itself; an alternative is to use shrinking [60].

The main contribution of Chapter 5 is to show that Anderson acceleration with history length m equal to 2 or 3 works remarkably well in conjunction with the widely used alternating projections method of Higham [56] for computing the nearest correlation matrix, both in its original form and in the forms that allow elements of the matrix to be fixed or a lower bound to be imposed on the smallest eigenvalue. Since no Newton method is available for the nearest correlation matrix problem with fixed elements, Anderson acceleration of the alternating projections method is the method of choice for this problem variant. Our recommendation for m is based on the balance between the reduction in both the number of iterations and the computation time: even though larger values of m in some examples lead to a further decease in the number of iterations the computation time sometimes increases for m larger than 2 or 3.

Although Anderson acceleration is well established in quantum chemistry applications and has recently started to attract the attention of numerical analysts, the method is still not well known in the numerical analysis community. Indeed it has not, to our knowledge, previously been applied to alternating projections methods. The success of Anderson acceleration in the nearest correlation matrix context suggests the possibility of using it in conjunction with other projection algorithms, such as those for feasibility problems, that is, finding a point (not necessarily the nearest one) in the intersection of several convex sets. Such algorithms include the (uncorrected) alternating projections method and the Douglas–Rachford method [5]. Gould [46, p. 10] states that an efficient acceleration scheme is needed for projection methods if they are to be successfully applied to real-life convex feasibility problems. Our work suggests that Anderson acceleration could make projection methods competitive in this context.

The main motivation for the work in Chapter 6 was the fact that the definiteness

structure possessed by most matrices to which the PPT is applied in [79] is not used or enforced in general formulae (1.13), which means that it could be lost in computation due to numerical errors. These matrices of interest, which generalize the quasidefinite structure, are called $\mathcal{I}$-semidefinite.

We have shown that $\mathcal{I}$-semidefinite matrices define Lagrangian semidefinite subspaces which are associated with the standard form of Hamiltonian and symplectic pencils appearing in control theory, and this makes $\mathcal{I}$-semidefinite matrices ubiquitous in the field. We also proved that the elementwise bound on the entries of an optimal permuted Riccati representation can be improved for the case of a Lagrangian semidefinite subspace.

The central part of the chapter was dedicated to deriving factored versions of the general PPT formulae used in the optimization algorithm for computing this optimal representation. These formulae now exploit the structure of an $\mathcal{I}$-semidefinite matrix X by working on the (not necessarily square) factors defining the semidefinite blocks and guarantee the definiteness properties of the resulting matrix by construction. Working directly with the factors of X is additionally appealing in view of the fact that the factors B and C are often available a priori in control theory. Furthermore, in this way we avoid forming the Gram matrices C^*C and BB^* where a possible loss of accuracy might occur.

Permuted Riccati matrices proved to be a valuable tool in many applications (see [88]) since from the stability angle, they can often be used in place of unitary matrices but are much easier to work with in finite arithmetic. There are plenty of problems that can be reduced to computing a "good" basis for a subspace, such as preconditioning least-squares [6] or a null space method for solving saddle-point systems [11, Sec. 6], and it would be interesting to see how permuted Riccati matrices perform in this setting (for the least-squares problem some analysis has already been done in [6]).

Bibliography

> ***day off*** *n. (in Academia)*
> *A day spent doing something related to your project that can still be considered productive but which requires no mental effort.*
> *e.g. "I took a day off and sorted my references."*
> —PHD Comics, 12/7/2015

[1] Andrei A. Agrachev and Yuri L. Sachkov. *Control theory from the geometric viewpoint*, volume 87 of *Encyclopaedia of Mathematical Sciences*. Springer-Verlag, Berlin, 2004. xiv+412 pp. Control Theory and Optimization, II. ISBN 3-540-21019-9.

[2] Gregory Ammar and Volker Mehrmann. On Hamiltonian and symplectic Hessenberg forms. *Linear Algebra Appl.*, 149:55 – 72, 1991.

[3] Donald G. Anderson. Iterative procedures for nonlinear integral equations. *J. Assoc. Comput. Mach.*, 12(4):547–560, 1965.

[4] Greg Anderson, Lisa Goldberg, Alec N. Kercheval, Guy Miller, and Kathy Sorge. On the aggregation of local risk models for global risk management. *Journal of Risk*, 8(1):25–40, 2005.

[5] Francisco J. Aragón Artacho, Jonathan M. Borwein, and Matthew K. Tam. Douglas–Rachford feasibility methods for matrix completion problems. *The ANZIAM Journal*, 55:299–326, 2014.

[6] Mario Arioli and Iain S. Duff. Preconditioning linear least-squares problems by identifying a basis matrix. *SIAM J. Sci. Comput.*, 37(5):S544–S561, 2015.

[7] Jared L. Aurentz, Raf Vandebril, and David S. Watkins. Fast computation of the zeros of a polynomial via factorization of the companion matrix. *SIAM J. Sci. Comput.*, 35(1):A255–A269, 2013.

[8] Zhaojun Bai, James W. Demmel, Jack J. Dongarra, Axel Ruhe, and Henk A. Van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000. xxix+410 pp. ISBN 0-89871-471-0.

[9] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 15(9):820–826, 1972.

[10] Peter Benner, Alan J. Laub, and Volker Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: the continuous-time case. Technical Report SPC 95-22, Forschergruppe 'Scientific Parallel Computing', Fakultät für Mathematik, TU Chemnitz-Zwickau, 1995. Version dated February 28, 1996.

[11] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.

[12] Vineer Bhansali and Mark B. Wise. Forecasting portfolio risk in normal and stressed markets. *Journal of Risk*, 4(1):91–106, 2001.

[13] Dario A. Bini, Bruno Iannazzo, and Beatrice Meini. *Numerical Solution of Algebraic Riccati Equations*. Society for Industrial and Applied Mathematics, 2011.

[14] Ernesto G. Birgin and Marcos Raydan. Robust stopping criteria for Dykstra's algorithm. *SIAM J. Sci. Comput.*, 26(4):1405–1414, 2005.

[15] Åke Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996. xvii+408 pp. ISBN 0-89871-360-9.

[16] Madalyn S. Blondes, John H. Schuenemeyer, Ricardo A. Olea, and Lawrence J. Drew. Aggregation of carbon dioxide sequestration storage assessment units. *Stoch. Environ. Res. Risk Assess.*, 27(8):1839–1859, 2013.

[17] Rüdiger Borsdorf. A Newton algorithm for the nearest correlation matrix. M.Sc. Thesis, The University of Manchester, Manchester, UK, September 2007. 151 pp. MIMS EPrint 2008.49, Manchester Institute for Mathematical Sciences, The University of Manchester.

[18] Rüdiger Borsdorf and Nicholas J. Higham. A preconditioned Newton algorithm for the nearest correlation matrix. *IMA J. Numer. Anal.*, 30(1):94–107, 2010.

[19] Rüdiger Borsdorf, Nicholas J. Higham, and Marcos Raydan. Computing a nearest correlation matrix with factor structure. *SIAM J. Matrix Anal. Appl.*, 31(5): 2603–2622, 2010.

[20] James P. Boyle and Richard L. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In *Advances in Order Restricted Statistical Inference*, Richard Dykstra, Tim Robertson, and Farroll T. Wright, editors, volume 37 of *Lecture Notes in Statistics*, Springer New York, 1986, pages 28–47.

[21] Jan H. Brandts. Matlab code for sorting real Schur forms. *Numer. Linear Algebra Appl.*, 9(3):249–261, 2002.

[22] Claude Brezinski and Michela Redivo-Zaglia. *Extrapolation Methods: Theory and Practice*. Studies in Computational Mathematics 2. North-Holland, Amsterdam, 1991. ix+464 pp. ISBN 0-444-88814-4.

[23] Charles George Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19:577–593, 1965.

[24] Sheung Hun Cheng and Nicholas J. Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. *SIAM J. Matrix Anal. Appl.*, 19 (4):1097–1110, 1998.

[25] Delin Chu, Xinmin Liu, and Volker Mehrmann. A numerical method for computing the Hamiltonian Schur form. *Numer. Math.*, 105(3):375–412, 2007.

[26] Charles R. Crawford and Yiu Sang Moon. Finding a positive definite linear combination of two Hermitian matrices. *Linear Algebra Appl.*, 51:37–48, 1983.

[27] Michael J. Daniels and Robert E. Kass. Shrinkage estimators for covariance matrices. *Biometrics*, 57(4):1173–1184, 2001.

[28] Philip I. Davies, Nicholas J. Higham, and Françoise Tisseur. Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem. *SIAM J. Matrix Anal. Appl.*, 23(2):472–493, 2001.

[29] Hakan Demirtas, Donald Hedeker, and Robin J. Mermelstein. Simulation of massive public health data by power polynomials. *Statist. Med.*, 31(27):3337–3346, 2012.

[30] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997. xi+419 pp. ISBN 0-89871-389-7.

[31] Susan J. Devlin, R. Gnanadesikan, and J. R. Kettenring. Robust estimation and outlier detection with correlation coefficients. *Biometrika*, 62(3):531–545, 1975.

[32] Richard J. Duffin, Dov Hazony, and Norman Morrison. Network synthesis through hybrid matrices. *SIAM J. Appl. Math.*, 14(2):390–413, 1966.

[33] Richard L. Dykstra. An algorithm for restricted least squares regression. *J. Amer. Statist. Assoc.*, 78:837–842, 1983.

[34] René Escalante and Marcos Raydan. *Alternating Projection Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2011. ix+127 pp. ISBN 967-1-611071-93-4.

[35] V. Eyert. A comparative study on methods for convergence acceleration of iterative vector sequences. *J. Comput. Phys.*, 124(2):271 – 285, 1996.

[36] Haw-ren Fang and Dianne P. O'Leary. Modified Cholesky algorithms: a catalog with new approaches. *Math. Program.*, 115(2):319–349, 2008.

[37] Haw-ren Fang and Yousef Saad. Two classes of multisecant methods for nonlinear acceleration. *Numer. Linear Algebra Appl.*, 16(3):197–221, 2009.

[38] Heike Fassbender. *Symplectic methods for the symplectic eigenproblem*. Kluwer Academic/Plenum Publishers, New York, 2000. xvi+269 pp. ISBN 0-306-46478-0.

[39] Christopher C. Finger. A methodology to stress correlations. *RiskMetrics Monitor*, Fourth Quarter:3–11, 1997.

[40] Matthias Fripp. Greenhouse gas emissions from operating reserves used to backup large-scale wind power. *Environ. Sci. Technol.*, 45(21):9405–9412, 2011.

[41] Scott Gerlt, Wyatt Thompson, and Douglas J. Miller. Exploiting the relationship between farm-level yields and county-level yields for applied analysis. *Journal of Agricultural and Resource Economics*, 39(2):253–270, 2014.

[42] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[43] Gene H. Golub and James M. Ortega. *Scientific Computing and Differential Equations: An introduction to numerical methods*. Academic Press, 1992. xi+337 pp. ISBN 978-0-08-051669-1.

[44] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Fourth edition, Johns Hopkins University Press, Baltimore, MD, USA, 2013. xxi+756 pp. ISBN 978-1-4214-0794-4.

[45] James H. Goodnight. A tutorial on the SWEEP operator. *The American Statistician*, 33(3):149–158, 1979.

[46] Nicholas I. M. Gould. How good are projection methods for convex feasibility problems? *Comput. Optim. Appl.*, 40:1–12, 2008.

[47] Chun-Hua Guo, Nicholas J. Higham, and Françoise Tisseur. An improved arc algorithm for detecting definite Hermitian pairs. *SIAM J. Matrix Anal. Appl.*, 31(3):1131–1151, 2009.

[48] Sven Hammarling and Craig Lucas. Updating the QR factorization and the least squares problem. MIMS EPrint 2008.111, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, November 2008. 72 pp.

[49] Sven J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2:303–323, 1982.

[50] Sven J. Hammarling. Numerical solution of the discrete-time, convergent, non-negative definite Lyapunov equation. *Systems and Control Letters*, 17:137–139, 1991.

[51] Douglas M. Hawkins and W. J. R. Eplett. The Cholesky factorization of the inverse correlation or covariance matrix in multiple regression. *Technometrics*, 24(3):191–198, 1982.

[52] Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra Appl.*, 103:103–118, 1988.

[53] Nicholas J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, M. J. C. Gover and S. Barnett, editors, Oxford University Press, 1989, pages 1–27.

[54] Nicholas J. Higham. Analysis of the Cholesky decomposition of a semi-definite matrix. In *Reliable Numerical Computation*, M. G. Cox and S. J. Hammarling, editors, Oxford University Press, 1990, pages 161–185.

[55] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. xxx+680 pp. ISBN 0-89871-521-0.

[56] Nicholas J. Higham. Computing the nearest correlation matrix—A problem from finance. *IMA J. Numer. Anal.*, 22(3):329–343, 2002.

[57] Nicholas J. Higham. *J*-orthogonal matrices: Properties and generation. *SIAM Rev.*, 45(3):504–519, 2003.

[58] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. xx+425 pp. ISBN 978-0-898716-46-7.

[59] Nicholas J. Higham. The nearest correlation matrix. `https://nickhigham.wordpress.com/2013/02/13/the-nearest-correlation-matrix`, 2013.

[60] Nicholas J. Higham, Nataša Strabić, and Vedran Šego. Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block. MIMS EPrint 2014.54, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, November 2014. 19 pp. Revised March 2015. To appear in SIAM Rev.

[61] Leslie Hogben, editor. *Handbook of linear algebra*. Discrete Mathematics and its Applications (Boca Raton). Second edition, CRC Press, Boca Raton, FL, 2014. xxx+1874 pp. ISBN 978-1-4665-0728-9.

[62] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Second edition, Cambridge University Press, Cambridge, UK, 2013. xviii+643 pp. ISBN 978-0-521-83940-2.

[63] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1995. xiii+165 pp. ISBN 0-89871-352-8.

[64] Alec N. Kercheval. Optimal covariances in risk model aggregation. In *Proceedings of the Third IASTED International Conference on Financial Engineering and Applications*, Calgary, 2006, pages 30–35. ACTA Press.

[65] Daniel Kressner. Block algorithms for reordering standard and generalized Schur forms. *ACM Trans. Math. Software*, 32(4):521–532, 2006.

[66] Paul H. Kupiec. Stress testing in a value at risk framework. *J. Derivatives*, 6 (1):7–24, 1998. Reprinted as [67].

[67] Paul H. Kupiec. Stress testing in a value at risk framework. In *Risk Management: Value at Risk and Beyond*, M. A. H. Dempster, editor, Cambridge University Press, 2002, chapter 3, pages 76–99.

[68] Eric Kvaalen. A faster Broyden method. *BIT*, 31(2):369–372, 1991.

[69] Peter Lancaster and Leiba Rodman. *Algebraic Riccati equations*. Oxford University Press, Oxford, 1995. xviii+480 pp. ISBN 0-19-853795-6.

[70] A.J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Control*, 24(6):913–921, 1979.

[71] Olivier Ledoit and Michael Wolf. Honey, I shrunk the sample covariance matrix. *J. Portfolio Management*, 30(4):110–119, 2004.

[72] Olivier Ledoit and Michael Wolf. Nonlinear shrinkage estimation of large-dimensional covariance matrices. *Ann. Statist.*, 40(2):1024–1060, 2012.

[73] Lijing Lin, Nicholas J. Higham, and Jianxin Pan. Covariance structure regularization via entropy loss function. *Comput. Statist. Data Anal.*, 72:315–327, 2014.

[74] Wen-Wei Lin, Volker Mehrmann, and Hongguo Xu. Canonical forms for Hamiltonian and symplectic matrices and pencils. *Linear Algebra Appl.*, 302/303: 469–533, 1999.

[75] Williams López and Marcos Raydan. An acceleration scheme for Dykstra's algorithm. *Comput. Optim. Appl.*, 2015.

[76] Craig Lucas. Computing nearest covariance and correlation matrices. M.Sc. Thesis, University of Manchester, Manchester, England, October 2001. 68 pp.

[77] Jerome Malick. A dual approach to semidefinite least-squares problems. *SIAM J. Matrix Anal. Appl.*, 26(1):272–284, 2004.

[78] Volker Mehrmann. *The autonomous linear quadratic control problem*, volume 163 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 1991. vi+177 pp. Theory and numerical solution. ISBN 3-540-54170-5.

[79] Volker Mehrmann and Federico Poloni. Doubling algorithms with permuted Lagrangian graph bases. *SIAM J. Matrix Anal. Appl.*, 33(3):780–805, 2012.

[80] Aleksei Minabutdinov, Ilia Manaev, and Maxim Bouev. Finding the nearest valid covariance matrix: A FX market case. Working paper Ec-07/13, Department of Economics, European University at St. Petersburg, St. Petersburg, Russia, 2013. Revised June 2014.

[81] Hossein Mohammadi, Abbas Seifi, and Toomaj Foroud. A robust Kriging model for predicting accumulative outflow from a mature reservoir considering a new horizontal well. *Journal of Petroleum Science and Engineering*, 82-83:113–119, 2012.

[82] NAG Toolbox for MATLAB. NAG Ltd., Oxford, UK. http://www.nag.co.uk.

[83] NAG Library. NAG Ltd., Oxford, UK. http://www.nag.co.uk.

[84] M. Olshanskii and E. Tyrtyshnikov. *Iterative Methods for Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[85] Chris Paige and Charles Van Loan. A Schur decomposition for Hamiltonian matrices. *Linear Algebra Appl.*, 41:11 – 32, 1981.

[86] Sergio Pezzulli, Patrizio Frederic, Shanti Majithia, Sal Sabbagh, Emily Black, Rowan Sutton, and David Stephenson. The seasonal forecast of electricity demand: a hierarchical Bayesian model with climatological weather generator. *Appl. Stochastic Models Bus. Ind.*, 22(2):113–125, 2006.

[87] Joshua H. Plasse. The EM algorithm in multivariate Gaussian mixture models using Anderson acceleration. M.Sc. Thesis, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA, USA, May 2013. 51 pp. http://www.wpi.edu/Pubs/ETD/Available/etd-042513-091152/.

[88] Federico Poloni. Permuted graph matrices and their applications. In *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory*, Peter Benner, Matthias Bollhöfer, Daniel Kressner, Christian Mehl, and Tatjana Stykel, editors, Springer International Publishing, 2015, pages 107–129.

[89] Florian A. Potra and Hans Engler. A characterization of the behavior of the Anderson acceleration on linear problems. *Linear Algebra Appl.*, 438(3):1002–1011, 2013.

[90] Mohsen Pourahmadi. Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika*, 86(3):677–690, 1999.

[91] Péter Pulay. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chemical Physics Letters*, 73(2):393 – 398, 1980.

[92] Houduo Qi and Defeng Sun. A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM J. Matrix Anal. Appl.*, 28(2): 360–385, 2006.

[93] Houduo Qi and Defeng Sun. Correlation stress testing for value-at-risk: an unconstrained convex optimization approach. *Comput Optim Appl*, 45(2):427–462, 2010.

[94] Houduo Qi and Defeng Sun. An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem. *IMA J. Numer. Anal.*, 31:491–511, 2011.

[95] Adi Raveh. On the use of the inverse of the correlation matrix in multivariate data analysis. *The American Statistician*, 39(1):39–42, 1985.

[96] Richard D. Ray and Bruce C. Douglas. Experiments in reconstructing twentieth-century sea levels. *Progress in Oceanography*, 91:496–515, 2011.

[97] Riccardo Rebonato and Peter Jäckel. The most general methodology for creating a valid correlation matrix for risk management and option pricing purposes. *J. Risk*, 2(2):17–27, 2000.

[98] Jiří Rohn. Computing the norm $\|A\|_{\infty,1}$ is NP-hard. *Linear and Multilinear Algebra*, 47(3):195–204, 2000.

[99] Thorsten Rohwedder and Reinhold Schneider. An analysis for the DIIS acceleration method used in quantum chemistry calculations. *J. Math. Chem.*, 49(9): 1889–1914, 2011.

[100] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.

[101] Juliane Schäfer and Korbinian Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1), 2005. Article 32.

[102] Robert B. Schnabel and Elizabeth Eskow. A new modified Cholesky factorization. *SIAM J. Sci. Statist. Comput.*, 11(6):1136–1158, 1990.

[103] Robert B. Schnabel and Elizabeth Eskow. A revised modified Cholesky factorization algorithm. *SIAM J. Optim.*, 9(4):1135–1148, 1999.

[104] James R. Schott. *Matrix Analysis for Statistics*. Wiley, New York, 1997.

[105] Hyundong Shin, M. Z. Win, and M. Chiani. Asymptotic statistics of mutual information for doubly correlated MIMO channels. *IEEE Trans. Wireless Communications*, 7(2):562–573, 2008.

[106] Danny C. Sorensen and Yunkai Zhou. Direct methods for matrix Sylvester and Lyapunov equations. *J. Appl. Math*, 2003(6):277–303, 2003.

[107] G. W. Stewart and Ji Guang Sun. *Matrix perturbation theory*. Computer Science and Scientific Computing. Academic Press, Inc., Boston, MA, 1990. xvi+365 pp. ISBN 0-12-670230-6.

[108] I. D. Stewart, I. M. S. White, A. R. Gilmour, R. Thompson, J. A. Woolliams, and S. Brotherstone. Estimating variance components and predicting breeding values for eventing disciplines and grades in sport horses. *Animal*, 6(9):1377–1388, 2012.

[109] Michael Stewart and G. W. Stewart. On hyperbolic triangularization: Stability and pivoting. *SIAM J. Matrix Anal. Appl.*, 19(4):847–860, 1998.

[110] T. Toni and B. Tidor. Combined model of intrinsic and extrinsic variability for computational network design with application to synthetic biology. *PLoS Comput Biol*, 9(3):e1002960, 2013.

[111] Alex Toth and C. T. Kelley. Convergence analysis for Anderson Acceleration. *SIAM J. Numer. Anal.*, 53(2):805–819, 2015.

[112] M. V. Travaglia. A lower bound for the nearest correlation matrix problem based on the circulant mean. *Comp. Appl. Math.*, 33:27–44, 2014.

[113] William F. Trench. Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices. *SIAM J. Matrix Anal. Appl.*, 10(2):135–146, 1989.

[114] Michael J. Tsatsomeros. Principal pivot transforms: properties and applications. *Linear Algebra Appl.*, 307(1-3):151–165, 2000.

[115] Albert W. Tucker. Principal pivotal transforms of square matrices. *SIAM Rev.*, 5(3):305, 1963.

[116] Saygun Turkay, Eduardo Epperlein, and Nicos Christofides. Correlation stress testing for value-at-risk. *Journal of Risk*, 5(4):75–89, 2003.

[117] Rajesh Tyagi and Chandrasekhar Das. Grouping customers for better allocation of resources to serve correlated demands. *Computers & Operations Research*, 26 (10-11):1041–1058, 1999.

[118] U.S. Geological Survey Geologic Carbon Dioxide Storage Resources Assessment Team. *National Assessment of Geologic Carbon Dioxide Storage Resources—Results (Ver. 1.1, September 2013)*, September 2013.

[119] Robert J. Vanderbei. Symmetric quasidefinite matrices. *SIAM J. Optim.*, 5(1): 100–113, 1995.

[120] Giulia Vilone, Damien Comiskey, Fanny Heraud, and Cian O'Mahony. Statistical method to assess usual dietary intakes in the European population. *Food Additives & Contaminants: Part A*, 31(10):1639–1651, 2014.

[121] Homer F. Walker. Anderson acceleration: Algorithms and implementations. Technical Report MS-6-15-50, Mathematical Sciences Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA, June 2011. 14 pp.

[122] Homer F. Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.*, 49(4):1715–1735, 2011.

[123] 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 Resour. Res.*, 45(5), 2009.

[124] Xiaohui Wang, Eric Anderson, Peter Steenkiste, and Fan Bai. Simulating spatial cross-correlation in vehicular networks. In *IEEE Vehicular Networking Conference (VNC)*, 2014, pages 207–214.

[125] David S. Watkins. *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*. Society for Industrial and Applied Mathematics, 2007.

[126] Nanny Wermuth, Michael Wiedenbeck, and David R. Cox. Partial inversion for linear systems and partial closure of independence graphs. *BIT*, 46(4):883–901, 2006.

[127] J. H. Wilkinson. Kronecker's canonical form and the QZ algorithm. *Linear Algebra Appl.*, 28:285–303, 1979.

[128] Joong-Ho Won, Johan Lim, Seung-Jean Kim, and Bala Rajaratnam. Condition-number-regularized covariance estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):427–450, 2013.

[129] Kefeng Xu and Philip T. Evers. Managing single echelon inventories through demand aggregation and the feasibility of a correlation matrix. *Computers & Operations Research*, 30(2):297–308, 2003.

Index

Sitting down and indexing a book is—in our experience—the most painful, horrible, mind–numbing activity you could ever wish on your worst enemy. ... Until recently, it also required a large stack of note cards, highlighter pens, Post–It notes, and serious medication."
—From "Working with Long Documents in Adobe InDesign CS3: Indexes (or Indices)"