Preprocessing for Hard Optimization Problems Across Structurally Diverse Models

by

Woosuk Jung

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics & Optimization

Waterloo, Ontario, Canada, 2026

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Michael P. Friedlander Professor, Dept. of Mathematics University of British Columbia
Supervisors:	Walaa M. Moursi Assistant Professor, Dept. of Combinatorics & Optimization University of Waterloo Henry Wolkowicz Professor, Dept. of Combinatorics & Optimization University of Waterloo
Internal Members:	Vijay Bhattiprolu Assistant Professor, Dept. of Combinatorics & Optimization University of Waterloo Stephen Vavasis Professor, Dept. of Combinatorics & Optimization University of Waterloo
Internal-External Member:	Hans De Sterck Professor, Dept. of Applied Mathematics University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis consists in part of six manuscripts written for publication. For each of these chapters, Woosuk Leo Jung made a major contribution to the research and was responsible for writing and integrating the material into this thesis.

Research presented in Chapter 3 is based on the following manuscript:

"The ω-condition number: applications to preconditioning and low rank generalized Jacobian updating."
W. Jung, D. Torregrosa-Belen, and H. Wolkowicz.

Research presented in Chapter 4 is based on the following manuscript:

"New insights and algorithms for optimal diagonal preconditioning."
S. Ghadimi, **W. Jung**, A. Sujanani, D. Torregrosa-Belen, H. Wolkowicz.

Research presented in Chapter 5 is based on the following manuscript:

"On the local and global minimizers of the smooth stress function in Euclidean Distance Matrix problems."
M. Song, D. Goncalves, **W. Jung**, C. Lavor, A. Mucherino, H. Wolkowicz.

Research presented in Chapter 6 is based on the following manuscript:

"Single Element Error Correction in a Euclidean Distance Matrix."
A. Alfakih, **W. Jung**, H. Wolkowicz, T. Xu.

Research presented in Chapter 7 is based on the following manuscript:

"Projection, Degeneracy, and Singularity Degree for Spectrahedra."
H. Im, **W. Jung**, D. Torregrosa-Belen, H. Wolkowicz.

Research presented in Chapter 8 is based on the following manuscript:

"Exact Solutions for the NP-hard Wasserstein Barycenter Problem using a Doubly Nonnegative Relaxation and a Splitting Method."
W. Jung, H. Wolkowicz.

All co-authors provided intellectual input, guidance, and feedback during the development of the corresponding manuscripts.

Abstract

Difficulties in solving large-scale optimization problems often arise from structural pathologies such as ill-conditioning, Hadamard ill-posedness, and degeneracy, particularly due to the failure of constraint qualifications. While standard algorithms often struggle to address these issues, preprocessing based on structural analysis offers an effective strategy for overcoming such challenges. This thesis investigates several preprocessing methods targeting various sources of these difficulties.

In Part I, we study a nonclassical, average condition number of linear systems, the ω-condition number. Our results demonstrate several advantages of the ω-condition number over the classical κ-condition number. First, ω provides a more accurate measure of the conditioning of linear systems by more faithfully capturing the effects of perturbations observed in practice. Second, ω exhibits superior numerical stability compared to κ. Third, when used in preconditioner design, ω more effectively promotes eigenvalue clustering, which is crucial for the efficiency of iterative solvers. Finally, the analytical simplicity of ω enables the derivation of explicit optimality conditions, allowing for closed-form expressions of optimal preconditioners under various frameworks, including low rank updates of the generalized Jacobian for semismooth Newton methods and diagonal or block-diagonal scaling. For diagonal preconditioning, we further include a comparison between two distinct notions of conditioning.

In Part II, we first answer in the affirmative a long-standing open question of whether the smooth stress function admits local nonglobal minimizers. This quartic nonconvex objective function arises in the exact recovery of a Euclidean distance matrix (**EDM**) of a given embedding dimension. By eliminating the Hadamard ill-posedness caused by translation and rotation invariance, we stabilize Newton's method and avoid singular Hessians.
We then consider the single-element error correction problem as a case study. We first show that the projection onto the **EDM** cone fails to recover the correct **EDM** in this setting. We then introduce divide-and-conquer strategies based on facial reduction. Our approach efficiently recovers the correct **EDM** with high accuracy, and we further provide criteria characterizing the existence of multiple solutions.

In Part III, we relate **FR** to the analysis of the convergence behaviour of a semismooth Newton method for projection onto a spectrahedron, i.e., the intersection of a linear manifold and the semidefinite cone. In this process, we derive an explicit formula for the projection onto a face of the semidefinite cone obtained via regularization and analyze pathologies that arise in the absence of strict feasibility. We further show that ill-conditioning of the Jacobian near optimality characterizes the degeneracy of the projection point.
As an application, we consider a simplified Wasserstein barycenter problem, a well-known **NP**-hard problem. We compute the Wasserstein barycenter by exploiting the structure of the linear constraints to obtain a facially reduced doubly nonnegative (**DNN**) relaxation. This reduction provides a natural splitting for applying the symmetric alternating direction method of multipliers (**sADMM**). The resulting algorithm exploits structure in the subproblems to compute strong upper and lower bounds. In most of the instances, we achieve the small gap between these bounds, which means that the original problem is solved.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Henry Wolkowicz and Walaa Moursi. Their guidance, encouragement, and support throughout my doctoral studies taught me not only how to conduct research, but also how to enjoy it.

I am also grateful to the members of my examining committee, Michael Friedlander, Vijay Bhattiprolu, Stephen Vavasis, and Hans De Sterck, for their time, careful reading, and valuable feedback.

I would also like to thank my academic siblings, Haesol and Tyler, as well as my colleagues, Matt, Fernanda, Kartik, Andrew, Martin, and many others. Their friendship and support throughout my doctoral studies helped me in countless ways and reminded me that I was never alone.

To my best friends, David and Mengmeng, thank you for visiting Waterloo and for becoming such dear friends. The summer of 2023 that we spent together remains one of the most beautiful and cherished periods of my doctoral journey.

Last but certainly not least, I offer my deepest gratitude to the St. Paul Chung Community and to Fr. Yang for the boundless love and care they have shown me. Their unwavering kindness and support sustained me through the most difficult moments of this journey and gave me the strength to carry on.

Dedication

To my mother,
with my deepest love and gratitude.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In modern optimization, the importance of scalable methods has been steadily increasing. In many cases, the difficulty of large-scale optimization problems lies not in their size per se, but in the structural pathologies they exhibit. These pathologies include numerical ill-conditioning of operators, Hadamard ill-posedness, and failures of constraint qualifications in conic programs. In this thesis, we investigate various preprocessing techniques that enhance numerical stability in challenging optimization problems by addressing these structural pathologies.

The thesis consists of three parts. Part I focuses on the numerical stability of linear systems. In particular, we investigate the nonclassical, average-motivated ω-condition number as a practical alternative to the standard κ-condition number. We demonstrate that ω provides a more informative measure of the conditioning of linear systems while offering both numerical and analytical advantages. We first justify the use of the ω-condition number through an error analysis of linear systems. Our discussion emphasizes that ω is *not* inverse invariant and that, for linear systems $Ax = b$, the relevant quantity to minimize is $\sqrt{\omega((A^TA)^{-1})}$. Numerical experiments are conducted to support this observation. Specifically, we examine the correlation between the empirical condition number of a system, computed from sampled perturbations, and various notions of condition numbers. Our results clearly show that the ω-condition number more accurately captures the effects of perturbations. We further present discussions and numerical experiments concerning the conditioning of condition numbers and eigenvalue clustering. We then turn our attention to preconditioning based on ω. In this context, the analytical simplicity of ω facilitates optimal preconditioning and leads to superior performance in iterative solvers compared to κ. Unlike the κ-condition number, which is difficult to differentiate and often nonsmooth, the ω-condition number admits a simple analytical form, enabling the derivation of explicit formulas for optimal preconditioners under various frameworks. We include results on ω-optimal low rank updates, κ- and ω-optimal diagonal preconditioning, and their comparative performance.

In Part II, we resolve a long-standing open question in the Euclidean Distance Matrix (**EDM**) completion problem by proving the existence of local nonglobal minimizers of the smooth stress function. A key difficulty in this problem arises from the singularity of the Hessian caused by Hadamard ill-posedness. Our approach stabilizes Newton's method by eliminating translation and rotation invariance, allowing us to numerically identify neighborhoods containing local nonglobal minimizers. The existence result is then established analytically using the classical Kantorovich theorem.
We then turn our attention to the single-element error correction problem for a noisy **EDM** with

a known embedding dimension. This problem is of particular interest because the standard nearest **EDM** formulation fails in this setting. We develop divide-and-conquer strategies based on facial reduction (**FR**) to identify the minimal face of the centered Gram matrix that guarantees a unique solution to the reduced problem. Under the general position assumption, our approach efficiently recovers the correct **EDM** with high accuracy. We further provide criteria for characterizing the geometric conditions under which multiple solutions may arise due to the failure of the general position assumption.

Finally, Part III addresses structural degeneracy, specifically the failure of strict feasibility, in conic optimization problems. Chapter 7 discusses several computational consequences of the failure of strict feasibility in the best approximation problem for spectrahedra. When Slater's condition fails, the generalized Jacobian of the optimality conditions becomes singular, causing semismooth Newton methods to collapse. We show that applying **FR** eliminates redundant constraints, which in turn restores dual attainment. Our numerical experiments show that this preprocessing technique reduces the condition number of the Jacobian from the order of 10^{12} to 10^2. We further explore the relationship between the failure of strict feasibility and the degeneracy of the projected point. An application of **FR** to a distance geometry problem is also presented. In Chapter 8, we study a simplified Wasserstein barycenter problem that is known to be **NP**-hard. We formulate the problem as a binary-constrained quadratic program and obtain an exact solution by constructing a facially reduced doubly nonnegative (**DNN**) relaxation. This **FR** provides a natural splitting, allowing us to apply the Peaceman-Rachford splitting method. To promote low rank solutions without sacrificing computational efficiency, we utilize redundant constraints. Our algorithm achieves a provable zero duality gap for generic cases. We also provide a geometric characterization of the nontrivial duality gaps that occur when the problem exhibits highly symmetric configurations.

1.1 Notation

We include Index.

We follow the work and notation in [68, 117, 123, 142, 181]. We work in a *real Euclidean space* $\mathbb{R}^n$ of dimension n equipped with the *inner product* $\langle \cdot, \cdot \rangle$ and induced *norm* $\|x\| = \sqrt{\langle x, x \rangle}$. We denote by $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ the nonnegative and positive orthants, respectively. The space of $m \times n$ matrices is denoted $\mathbb{R}^{m \times n}$.

Let $\mathbb{S}^n$ denote the space of real symmetric matrices of order n, equipped with the trace inner product $\langle S, T \rangle = \operatorname{trace}(ST)$. The set $\mathbb{S}^n_+ \subseteq \mathbb{S}^n$ denotes the closed convex cone of positive semidefinite matrices, and $\mathbb{S}^n_{++}$ the cone of positive definite matrices. We write $X \succeq 0$ (resp., $X \succ 0$) to denote $X \in \mathbb{S}^n_+$ (resp., $X \in \mathbb{S}^n_{++}$). Additionally, $X \geqslant 0$ and $X > 0$ denote that all entries of X are non-negative and positive, respectively.

We let $\mathcal{N}^n := \{X \in \mathbb{S}^n : X \geqslant 0\}$ denote $n \times n$ nonnegative symmetric matrices. The cone of doubly nonnegative matrices is $\mathbf{DNN} = \mathbb{S}^n_+ \cap \mathcal{N}^n$.

We use the Kronecker product $A \otimes B$ and the Hadamard (elementwise) product $A \circ B$. The vectorization of a matrix $X \in \mathbb{R}^{n \times d}$ is denoted by $\operatorname{vec}(X)$, and $\operatorname{Mat} \cong \operatorname{vec}^*$ is the adjoint of vec, satisfying $\operatorname{Mat}(\operatorname{vec}(X)) = X$ for all $X \in \mathbb{R}^{n \times d}$. The linear operator $\operatorname{Diag} : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ maps a vector v to the diagonal matrix $\operatorname{Diag}(v)$ whose diagonal is v, and its adjoint is $\operatorname{diag} = \operatorname{Diag}^*$. We use the MATLAB notation blkdiag to denote the block diagonal matrix formed from the arguments.

For $A \in \mathbb{R}^{m \times n}$, the Moore-Penrose pseudoinverse is $A^\dagger$.

For $B \in \mathbb{S}^n$, eigenvalues are ordered as

$$\lambda_{\max}(B) = \lambda_1(B) \geqslant \cdots \geqslant \lambda_n(B) = \lambda_{\min}(B).$$

We often use the following matrix for a basis for the orthogonal complement $e_n^\perp$,

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{n-1} \\ -e_{n-1}^T \end{bmatrix}, \quad \text{range}(V) = e_n^\perp. \tag{1.1}$$

For integers s, t, and k, $s \leqslant t$, we define

$$[s,t] = \{s, s+1, \ldots, t\}, \quad [k] = \{1, 2, \ldots, k\},$$

and $t(k) = k(k+1)/2$, *triangular number*.

For a set $C \subseteq \mathbb{R}^n$, we denote the interior and relative interior by $\text{int}(C)$ and $\text{relint}(C)$. The open ball centered at x with radius ε is $\mathrm{B}(x, \varepsilon)$.

We denote the zero vector in $\mathbb{R}^n$ by 0_n and the zero matrix in $\mathbb{R}^{n\times n}$ by $0_{n\times n}$. The identity matrix is I (or I_n when size matters). The vector of all ones is e, and e_i denotes the i-th unit vector. Moreover, we denote $E_{jk} = e_j e_k^T + e_k e_j^T$, *unit matrix*.

For index sets $\mathcal{I}, \mathcal{J}$, we denote submatrices by

$$A_{(:,\mathcal{I})}, \quad A_{(\mathcal{J},:)}, \quad A_{(\mathcal{J},\mathcal{I})}.$$

For $x \in \mathbb{R}^n$, we define

$$x_+ = (\max\{0, x_i\})_{i=1}^n, \quad x_- = (\min\{0, x_i\})_{i=1}^n.$$

For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we write ∇f. If $n = 1$, we use f'. A differentiable function $f : \Omega \to \mathbb{R}$ on an open set Ω is *pseudoconvex* if

$$\nabla f(x)^T (y - x) \geqslant 0 \implies f(y) \geqslant f(x), \quad \forall x, y \in \Omega.$$

In this case, $\nabla f(x) = 0$ characterizes global minimizers on convex domains. See, e.g., [126].

The *Fenchel subdifferential*, $\partial h(x)$, of a convex function h is

$$\partial h(x) = \{v : \langle v, y - x\rangle \leqslant h(y) - h(x),\ \forall y\},$$

and the *Clarke subdifferential*, $\partial_C h(x)$, of a locally Lipschitz, not necessarily convex, function h is defined as

$$\partial_C h(x) = \text{conv}\left\{s : \exists x^k \to x, \nabla h(x^k) \text{ exists, and } \nabla h(x^k) \to s\right\}, \tag{1.2}$$

where conv denotes the convex hull of a set. It is well known that the Fenchel subdifferential and the Clarke subdifferential coincide for convex functions.

A function $f : S \to \mathbb{R}$ defined on a convex subset $S \subseteq \mathbb{R}^n$ is *quasiconvex* if

$$f(\lambda x + (1-\lambda)y) \leqslant \max\{f(x), f(y)\}, \qquad \forall x, y \in S, \lambda \in [0,1].$$

Now we define a special subdifferential called the quasisubdifferential for a quasiconvex function f. Define the *strict sublevel set* or *inner slice* of f as:

$$\bar{S}(x) := \big\{y \in \operatorname{int} \bar{D} : f(y) < f(x)\big\},$$

where $\operatorname{int} \bar{D}$ denotes the interior of the domain of f. The *quasisubdifferential,* $\bar{\partial}^\circ f(x)$, of a quasiconvex function f relative to the above sublevel set is defined as

$$\bar{\partial}^\circ f(x) := \big\{g : \langle g, y - x\rangle < 0, \quad \forall y \in \bar{S}(x)\big\}. \tag{1.3}$$

For column and row index subset $\alpha \subseteq [n]$ we let D_α denote the corresponding principal submatrix of D, and

$$\mathcal{P}_\alpha : \mathbb{S}^n \to \mathbb{S}^\alpha,\ \mathcal{P}_\alpha(D) = D_\alpha,$$

is the projection, or *coordinate shadow*, corresponding to α. We abuse notation by using $\mathbb{S}^\alpha$ rather than $\mathbb{S}^{|\alpha|}$ to emphasize the actual coordinates used. The inverse image is

$$\mathcal{P}_\alpha^{-1}(\bar{D}) = \{D \in \mathbb{S}^n : D_\alpha = \bar{D}\}.$$

We extend the coordinate shadow for rectangular matrices

$$P \in \mathbb{R}^{n\times d}, \quad \mathcal{P}_\alpha(P) \in \mathbb{R}^{|\alpha|\times d},$$

corresponds to the rows indexed by α.

Further notation will be introduced as needed.

1.2 Contributions and Organization

We now summarize the organization and main contributions of the thesis. We begin in Chapter 2 by reviewing the necessary background on spectral functions, including eigenvalue functions and projection operators, as well as convergence analysis for quasiconvex optimization, facial reduction, semidefinite programming, and Euclidean distance matrices. The main results of the thesis are divided into three parts.

In Part I, we study the ω-condition number and preconditioning for linear systems. The objective of this part is to demonstrate the advantages of using the ω-condition number over the classical κ-condition number. Chapter 3 discusses various properties of the ω-condition number and explains why these properties are favorable for analyzing the conditioning of linear systems. The contributions of this chapter include:

- An error analysis of linear systems that justifies the use of the ω-condition number and, in particular, highlights the importance of minimizing

$$\omega_{\text{inv}}(A) := \sqrt{\omega((A^TA)^{-1})};$$

- Empirical evidence that ω exhibits improved computational stability and more effectively promotes eigenvalue clustering; and

- ω-optimal low rank updates that exploit the analytical simplicity of ω.

Chapter 4 focuses on optimal diagonal preconditioning and provides a careful comparison between κ- and ω-based approaches.The main contributions are as follows:

- A new approach for computing κ-optimal diagonal preconditioners, based on an affine reformulation obtained by exploiting the invariance of eigenvalues;
- A theoretical validation of existing preconditioners, including Jacobi preconditioning and matrix balancing methods such as Sinkhorn–Knopp scaling, through the lens of ω-optimal preconditioning under various formulations; and
- A numerical comparison between ω-optimal and κ-optimal preconditioners for iterative methods, including PCG and LSQR.

Part II addresses **EDM** completion problems with a given embedding dimension. In Chapter 5, we answer in the affirmative a long-standing open question concerning the existence of local nonglobal minimizers of the smooth stress function arising in **EDM** completion. The main contributions of this chapter include:

- A characterization of the condition in which every second-order stationary point of the smooth stress function is a global minimizer; and
- The construction of local nonglobal minimizers, first identified numerically by a stabilized Newton method and then certified analytically using Kantorovich's theorem.

The methodology for the second contribution consists of:

- Stabilizing Newton's method by eliminating translation and rotation invariance;
- Searching for nearly optimal points using second-order methods; and
- Certifying the existence of local nonglobal minimizers using the Kantorovich theorem.

Then Chapter 6 considers a single element error correction problem for **EDM**s, where the embedding dimension is known and exactly one distance entry is corrupted. This problem is of particular interest because the standard nearest **EDM** formulation fails generically in this setting. Instead, we develop divide-and-conquer approaches based on facial reduction to recover the correct **EDM** efficiently and accurately. The main contributions of this chapter are:

- A characterization of the exceptional cases in which the nearest **EDM** formulation recovers the correct **EDM** , showing that this occurs only in a highly degenerate setting;
- Divide-and-conquer algorithms based on three forms of facial reduction: exposing vectors, facial vectors, and Gale transforms;
- A clarification of the relationships among these facial reduction techniques, including their connection to Gale transforms;

- Complexity analysis and numerical experiments demonstrating that the proposed methods solve very large instances to near machine precision; and
- A characterization of hard cases in which the general position assumption fails, including conditions under which a single-entry perturbation preserves both the **EDM** property and the prescribed embedding dimension, and examples where multiple solutions arise.

Part III then provides a more thorough discussion of facial reduction in conic programming. In Chapter 7, we examine the effect of the failure of strict feasibility on projection problems over spectrahedra. In particular, we study the convergence behaviour of a semismooth Newton method for projection onto a spectrahedron in relation to **FR**. The contributions of this chapter include:

- An explicit formula for the projection onto a face of the semidefinite cone:

$$P_f(u) = V\left(P_{\mathbb{S}^r_+}(V^T u V)\right)V^T,$$

  where $f = V\mathbb{S}^r_+V^T$ and $V^TV = I$;
- The identification of two pathologies that arise in the absence of strict feasibility: failure of dual attainment and unboundedness of the dual optimal set;
- A proof of the equivalence between degeneracy of the optimal projection point and singularity of the Jacobian; and
- A connection between the eigenpairs associated with small eigenvalues of the Jacobian and the construction of exposing vectors for **FR**.

We then extend our approach from the positive semidefinite cone to the doubly nonnegative cone. In Chapter 8, we study the simplified Wasserstein barycenter problem, which is known to be **NP**-hard. By applying **FR** , we obtain a facially reduced **DNN** relaxation, which naturally leads to a splitting method. The main contributions of this chapter include:

- The symmetric alternating direction method of multipliers based on the facially reduced **DNN** relaxation;
- The use of redundant constraints to promote low-rank solutions and improve algorithmic performance;
- Provable lower and upper bounds that certify optimality when the duality gap vanishes; and
- A characterization of instances with nontrivial duality gaps.

Our algorithm solves most instances by producing a zero duality gap. We further characterize nontrivial duality gaps that arise in highly symmetric configurations.

Chapter 2

Background

We first present some background on spectral functions, convergence results for quasiconvex optimization, **EDM**s, and facial reduction.

2.1 Spectral Functions

2.1.1 Eigenvalue Functions

We now recall the well-known facts about the largest and smallest eigenvalues of symmetric matrices and include a proof for completeness as it is useful further below.

Lemma 2.1.1. *The maximum and minimum eigenvalue functions on* $\mathbb{S}^n$, $\lambda_{\max}, \lambda_{\min} : \mathbb{S}^n \to \mathbb{R}$, *are convex, concave, respectively.*

Proof. Letting $B \in \mathbb{S}^n$ and using the Rayleigh quotient, we get $\lambda_{\max}(B) = \max\{x^T Bx : \|x\| = 1\}$, which is a maximum of linear (so convex) functions in B, and therefore is convex. We get the concavity part similarly from $\lambda_{\min}(B) = \min\{x^T Bx : \|x\| = 1\}$. □

2.1.2 Projection Operators and Their Derivatives

We work with $f : \mathbb{S}^n \to \mathbb{R} \cup \{+\infty\}$, a closed proper extended valued convex function on $\mathbb{S}^n$. We denote P_S, *projection onto a nonempty closed convex set* S, i.e.,

$$P_S(W) = \operatorname*{argmin}_{X \in S} \frac{1}{2}\|W - X\|^2.$$

And for a convex set S, we denote the *indicator function,* ι_S. The *Moreau regularization of* $\iota_{\mathbb{S}^n_-}$, $\mathbb{S}^n_- := -\mathbb{S}^n_+$, (see [142]) is given by

$$\Delta(X) = \min_{Z \in \mathbb{S}^n} \left\{ \frac{1}{2}\|X - Z\|^2 + \iota_{\mathbb{S}^n_-}(Z) \right\}. \tag{2.1}$$

A *spectral function* $g : \mathbb{S}^n \to \mathbb{R} \cup \{+\infty\}$ is one that is invariant under orthogonal conjugation (congruence)

$$g(X) = g(U^T X U), \, \forall \, X \in \mathbb{S}^n, \forall \, U \in \mathcal{O}^n,$$

where $\mathcal{O}^n$ is the set of orthogonal matrices of order n. In [123, Lemmas 2.3-4], it is shown that the Moreau regularization, $\Delta(X)$, of $\iota_{\mathbb{S}^n_-}$ is a *spectral function* with its gradient given as $\nabla \Delta(X) = P_{\mathbb{S}^n_+}(X)$. See Lemma 2.1.3 below. Let $\lambda : \mathbb{S}^n \to \mathbb{R}^n$ denote the eigenvalue function, i.e., $\lambda(X)$ is the vector of eigenvalues of $X \in \mathbb{S}^n$ in nonincreasing order.

Lemma 2.1.2 ([123, Lemma 2.3]). *The function Δ in (2.1) is the spectral function $\Delta = \delta \circ \lambda$, where the function δ on $x \in \mathbb{R}^n$ is:*

$$\delta(x) = \frac{1}{2} \sum_{i=1}^{n} \max\{0, x_i\}^2.$$

Proof. We include the proof from [123, Lemma 2.3] for completeness. For any $X \in \mathbb{S}^n$ we have

$$\begin{array}{rcl} \Delta(X) & = & \frac{1}{2}\|X - P_{\mathbb{S}^n_-}(X)\|^2 \\ & = & \frac{1}{2}\|P_{\mathbb{S}^n_+}(X)\|^2 \\ & = & \frac{1}{2}\sum_{i=1}^n \max\{0, \lambda_i(X)\}^2 \\ & = & \delta(\lambda(X)). \end{array}$$

□

Lemma 2.1.3 ([123, Lemma 2.4]). *The function Δ in (2.1) is convex and differentiable. Moreover, its gradient at $X \in \mathbb{S}^n$ is $P_{\mathbb{S}^n_+}(X)$, i.e.,*

$$\Delta(X)'(dX) = \langle \nabla \Delta(X), dX \rangle = \langle P_{\mathbb{S}^n_+}(X), dX \rangle = \operatorname{trace} P_{\mathbb{S}^n_+}(X) dX.$$

From spectral function theory e.g., [68,117,123], we know that the differentiability in Lemma 2.1.3 follows from the differentiability of $\delta \circ \lambda$. The formula for the derivative follows from the spectral function formula with $X = U \operatorname{Diag}(\lambda(X)) U^T$:

$$\nabla(\delta \circ \lambda)(X) = U \left(\operatorname{Diag} \nabla \delta(\lambda(X))\right) U^T, \; U \in \mathcal{O}^n.$$

The derivative (Jacobian) of the projection can be found from the Hessian of the regularization function

$$P'_{\mathbb{S}^n_+}(X) = \nabla^2 \Delta(X).$$

2.2 Convergence Analysis for Quasiconvex Optimization

In this section, we present a known convergence result for subgradient methods in quasiconvex optimization problems [109]. Let $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$ be an extended real-valued function with *domain* $\bar{D}$. The following minimization problem is considered:

$$f_* := \inf \{f(x) : x \in X\}, \tag{2.2}$$

with the following assumptions on f and X:

A1 : $\operatorname{int} \bar{D}$ is nonempty and convex.

A2 : $\overline{\lim}_{t\downarrow 0} f(x + t(y - x)) \leqslant f(x)$, $\forall x \in \bar{D}, y \in \operatorname{int} \bar{D}$.

A3 : f is upper semicontinuous (usc) on $\operatorname{int} \bar{D}$, i.e., $f(x) = \lim_{\epsilon\downarrow 0} \sup_{B(x,\epsilon)} f, \quad \forall x \in \operatorname{int} \bar{D}$ where $B(x, \epsilon) := \{y : \|y - x\| \leqslant \epsilon\}$.

A4 : f is quasiconvex on $\operatorname{int} \bar{D}$, i.e., the set $\{x \in \operatorname{int} \bar{D} : f(x) \leqslant \alpha\}$ is convex $\forall \alpha \in \mathbb{R}$.

A5 : The constraint set $X \subset \mathbb{R}^n$ is closed convex, and $X \cap \operatorname{int} \bar{D} \neq \varnothing$.

The following lemma from [109] discusses that fractional programs with certain structures are quasiconvex and also provides one characterization of their quasisubgradients.

Lemma 2.2.1 ([109, Lemma 4]). *Suppose $f(x) = a(x)/b(x)$ for all $x \in \operatorname{int} \bar{D}$, where a is a convex function, b is finite and positive on $\operatorname{int} \bar{D}$, $\operatorname{int} \bar{D}$ is convex, and one of the following conditions holds:*

(a) b is affine;

(b) a is nonnegative on $\operatorname{int} \bar{D}$ and b is concave;

(c) a is nonpositive on $\operatorname{int} \bar{D}$ and b is convex.

Then f is quasiconvex on $\operatorname{int} \bar{D}$ and for each $x \in \operatorname{int} \bar{D}$, if $\alpha := f(x)$ is finite then $a - \alpha b$ is convex and $\partial[a - \alpha b](x) \subset \bar{\partial}^{\circ} f(x)$.

The following lemma can be found in [109] and is useful for showing that assumptions (A1)-(A4) hold for our set-up in (4.10) since $\kappa(\mathcal{D}(d))$ is the ratio of a convex and concave function that is positive on Ω.

Lemma 2.2.2 ([109, Remark 2]). *Suppose a and $\tilde{b} := -b$ are proper convex functions on $\operatorname{int} \bar{D}^a \cap \operatorname{int} \bar{D}^b$, a is nonnegative on the (open and convex set) $C := \left\{y \in \operatorname{int} \bar{D}^a \cap \operatorname{int} \bar{D}^b : b(y) > 0\right\} \neq \varnothing$, and*

$$f(x) := \begin{cases} a(x)/b(x), & \text{if } x \in C, \\ \sup_{y \in C} \overline{\lim}_{t\downarrow 0} f(x + t(y - x)), & \text{if } x \in \operatorname{bd} C, \\ \infty, & \text{if } x \notin \operatorname{cl} C \end{cases}$$

*where $\operatorname{cl} C$ denotes the closure, $\operatorname{bd} C$ denotes the boundary of C, and $\bar{D}^a$ and $\bar{D}^b$ denote the domains of a and b, respectively. Then assumptions **A1-A4** hold with $\operatorname{int} \bar{D} = C$.*

2.3 Euclidean Distance Matrices

We introduce the necessary notation and background from distance geometry. Further details are provided in [5]. For a set of n points $p_i \in \mathbb{R}^d, i = 1, \ldots, n$, we denote the matrix of points, the configuration matrix, $P = \begin{bmatrix} p_1 & p_2 & \ldots & p_n \end{bmatrix}^T \in \mathbb{R}^{n \times d}$. The Euclidean distance matrix (of squared distances) is $D = \left(\|p_i - p_j\|^2\right) \in \mathcal{E}^n \subset \mathbb{S}^n$. We denote the closed convex *cone of **EDM***, $\mathcal{E}^n$. Here $0 < d = \operatorname{embdim}(D)$ is the embedding dimension of the **EDM**. Without loss of generality, we can

translate the points so that they are centered, i.e., with *vector of ones,* e, we have $P^T e = 0$. Note that $v := \frac{1}{n} P^T e$ is the barycenter of the points. The translation is then given by

$$P^T \leftarrow P^T - ve^T. \tag{2.3}$$

Our approach works well under the assumption that the points are in *general position*, i.e., no $d+1$ of them lie in a proper hyperplane; or, equivalently, that every subset of $d+1$ points are affinely independent, e.g., [4, 6]. This guarantees that our algorithms succeed as it is based on identifying whether a Gram matrix corresponding to chosen principal submatrices has the correct rank d. However, we have also extended our algorithms to allow for the case when the general position assumption fails.

We denote the corresponding *Gram matrix,* $G = PP^T$. Then the classical result of Schoenberg [157] relates the matrix of squared distances, the *Euclidean distance matrix,* ***EDM***, with a Gram matrix by applying $\mathcal{K}(G)$,

$$D = \mathcal{K}(G) = \operatorname{diag}(G)e^T + e\operatorname{diag}(G)^T - 2G \in \mathcal{E}^n. \tag{2.4}$$

Moreover, this mapping is one-one and onto between the *centered subspace,* $\mathcal{S}_C^n$, and *hollow subspace,* $\mathcal{S}_H^n$,

$$\mathcal{S}_C^n = \{X \in \mathbb{S}^n \,:\, Xe = 0\}, \quad \mathcal{S}_H^n = \{X \in \mathbb{S}^n \,:\, \operatorname{diag}(X) = 0\}. \tag{2.5}$$

We ignore the dimension n when the meaning is clear. Note that the centered assumption $P^T e = 0 \implies G \in \mathcal{S}_C^n$. Let

$$J = I - \frac{1}{n} ee^T \tag{2.6}$$

be the orthogonal projection onto the orthogonal complement $e^\perp$. Note that the translation in (2.3) is equivalent to

$$P \leftarrow JP = P - \frac{1}{n} ee^T P.$$

We let offDiag: $\mathbb{S}^n \to \mathbb{S}^n$ denote the orthogonal projection onto $\mathcal{S}_H^n$, the hollow matrices. Then the *Moore-Penrose generalized inverse,* $\mathcal{K}^\dagger$, is

$$\mathcal{K}^\dagger(D) = -\frac{1}{2} J \operatorname{offDiag}(D) J.$$

2.3.1 Commutativity of $\mathcal{P}_\alpha, \mathcal{K}$

We first include a useful and very interesting observation about the commutativity of $\mathcal{K}, \mathcal{P}_\alpha, \alpha \subset [n]$.

Lemma 2.3.1. *Let $M \in \mathbb{R}^{n\times n}, \alpha \subset [n]$. Then, by abuse of notation on the dimensions of e and the transformations $\mathcal{K}, \mathcal{P}_\alpha$, we get*

$$\mathcal{P}_\alpha \mathcal{K}(M) = \mathcal{K}\mathcal{P}_\alpha(M).$$

Proof. We note that $\mathcal{P}_\alpha(\operatorname{diag}(M)) = \operatorname{diag}(\mathcal{P}_\alpha(M))$. Therefore,

$$\begin{aligned} \mathcal{P}_\alpha \mathcal{K}(M) &= \mathcal{P}_\alpha \left(\operatorname{diag}(M)e^T + e\operatorname{diag}(M) - 2M\right) \\ &= \operatorname{diag}(\mathcal{P}_\alpha(M))e^T + e(\operatorname{diag}(\mathcal{P}_\alpha M)) - 2\mathcal{P}_\alpha(M) \\ &= \mathcal{K}\mathcal{P}_\alpha(M). \end{aligned}$$

□

We note that symmetry for M is not needed.

2.3.2 General Position

Lemma 2.3.2. *Let $P \in \mathbb{R}^{n\times d}$ be a configuration matrix with*

$$G = PP^T,\ D = \mathcal{K}(G) \in \mathcal{E}^n,\ d = \operatorname{embdim}(D).$$

Then the general position assumption is equivalent to

$$\alpha \subset [n], |\alpha| \geqslant d+1 \implies \operatorname{rank}(\mathcal{K}^\dagger(D_\alpha)) = d. \tag{2.7}$$

Proof. The general position assumption states that the dimension of the affine hull of the subset of points in the columns of P_α^T is full dimension d. Since a translation does not change this statement nor change the distances between points, we can assume without loss of generality that the points are centered, $P_\alpha^T e = 0$, and general position is equivalent to $\operatorname{span}(P_\alpha^T) = \mathbb{R}^d$, or equivalently $\operatorname{rank}(P_\alpha^T) = d$. We now note that

$$G_\alpha := \mathcal{K}^\dagger(D_\alpha) =: P_\alpha P_\alpha^T,$$

where G_α is the centered Gram matrix. The result now follows from $\operatorname{rank}(P_\alpha^T) = \operatorname{rank}(P_\alpha P_\alpha^T)$.

An equivalent result using the Gale matrix of D is given in [5, Cor. 3.1]. Further connections with Gale matrices is given below, Section 6.3.3. □

Moreover, we find the orthogonal projection $\mathcal{P}_{\operatorname{range}(K^\dagger)}$ as

$$\mathcal{K}^\dagger\mathcal{K}(S) = JSJ. \tag{2.8}$$

Note that the orthogonal projection satisfies $\mathcal{P}_{\operatorname{range}(K^\dagger)} = \mathcal{P}_{\mathcal{S}_C^n}$. Throughout we abuse notation and use $\mathcal{K}, \mathcal{P}, J$ without indicating the dimension of the space that is involved.

2.4 Facial Reduction

The facial structure of cones plays an essential role when analyzing various stability concepts. In this section we study various properties that arise from the absence of strict feasibility. Section 2.4.1 presents the basics of the facial structure of $\mathbb{S}_+^n$ and the facial reduction process for $\mathcal{F}$. In Section 2.4.1 we revisit known notions of singularities and the length of the facial reduction process that connects to the dimension of the solution set of our problem in Part III.

2.4.1 Facial Reduction Process

We first review the basic facial structure of $\mathbb{S}_+^n$. Recall that the convex cone $f \subseteq K$ is a face of a convex cone $K \subseteq \mathbb{S}^n$, denoted by $f \trianglelefteq K$, if

$$x, y \in K, z = x + y, z \in f \implies x, y \in f.$$

The cone f is a proper face if $\{0\} \subsetneq f \subsetneq K$. Here we denote f^Δ, *conjugate face of* f, defined as the intersection of the nonnegative polar cone with the orthogonal complement, $f^\Delta = f^\perp \cap K^+$, where $K^+ := \{\phi : \langle \phi, k\rangle \geqslant 0, \forall k \in K\}$ is the *nonnegative polar cone* of K. The facial structure of $\mathbb{S}^n_+$ is well-studied and has an intuitive characterization. For any convex set $C \subseteq \mathbb{S}^n_+$, the minimal face of $\mathbb{S}^n_+$ containing C, i.e., the intersection of all faces of $\mathbb{S}^n_+$ containing C, is denoted $\text{face}(C, \mathbb{S}^n_+)$. For the singleton $C = \{X\} \subseteq \mathbb{S}^n_+$, we get

$$\text{face}(X, \mathbb{S}^n_+) = \{Y \in \mathbb{S}^n_+ : \text{range}(Y) \subseteq \text{range}(X)\}.$$

Consider the following conic programming problem:

$$\begin{array}{ll} \min & g(X) \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \in \mathbb{S}^n_+, \end{array} \tag{2.9}$$

where $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a linear map defined by $\mathcal{A}(X) = (\text{trace}\, A_i X) \in \mathbb{R}^m$ for given $A_i \in \mathbb{S}^n$, $i = 1, \ldots, m$. Without loss of generality, we assume that $\mathcal{A}$ is surjective and A_i's are linearly independent. Facial reduction (**FR**) for $\mathcal{F} := \{X \in \mathbb{S}^n_+ : \mathcal{A}(X) = b\}$, the constraint system in (2.9), is the process of identifying the minimal face of $\mathbb{S}^n_+$ containing $\mathcal{F}$. It is known that a point $\hat{X}$ in the relative interior of $\mathcal{F}$, $\text{relint}(\mathcal{F})$, provides the following characterization

$$\text{face}(\hat{X}, \mathbb{S}^n_+) = \text{face}(\mathcal{F}, \mathbb{S}^n_+).$$

Furthermore, $\text{face}(\hat{X}, \mathbb{S}^n_+) = V\mathbb{S}^r_+V^T$ for some rank-r matrix V satisfying $\text{range}(V) = \text{range}(\hat{X})$. Such a matrix V is called a *facial range vector* for $\text{face}(\hat{X}, \mathbb{S}^n_+)$.

Finding $\text{face}(\mathcal{F}, \mathbb{S}^n_+)$ for an arbitrary $\mathcal{F}$ analytically is a challenging task. An alternative approach that is often used is to find the minimal face numerically. Proposition 2.4.1 below provides a tool for the **FR** process. Below, $\mathcal{A}^*$ refers to the *adjoint* of $\mathcal{A}$.

Proposition 2.4.1 (theorem of the alternative)**.** *For the feasible constraint system $\mathcal{F}$ of (2.9), exactly one of the following statements holds:*

(i) there exists $X \succ 0$ such that $X \in \mathcal{L}$;

(ii) there exists $\lambda \in \mathbb{R}^m$ that satisfies the auxiliary system

$$0 \neq Z = \mathcal{A}^*\lambda \succeq 0,\ \langle b, \lambda\rangle = 0. \tag{2.10}$$

The vector Z in (2.10) is called an *exposing vector* for the face $(Z^\perp \cap \mathbb{S}^n_+) \supseteq \mathcal{F}$, as $X \in \mathcal{F}$ implies

$$0 = \langle b, \lambda\rangle = \langle \mathcal{A}X, \lambda\rangle = \langle X, \mathcal{A}^*\lambda\rangle = \langle X, Z\rangle.$$

Here the face $(Z^\perp \cap \mathbb{S}^n_+)$ is exposed by the hyperplane $Z^\perp = \{Z\}^\perp$, and this allows for a simplified expression of feasible points. This restriction results in an equivalent strictly smaller dimensional problem. Finding a solution to (2.10) does not necessarily lead directly to the minimal face, i.e., we can have $(Z^\perp \cap \mathbb{S}^n_+) \supsetneq \text{face}(\mathcal{F}, \mathbb{S}^n_+)$. For such a case, the process is reapplied until $\text{face}(\mathcal{F}, \mathbb{S}^n_+)$ is found. The number of times the system (2.10) needs to be solved varies, though it is at most $\min\{m, n\}$, see Section 2.4.1 below. A reader may refer to Example 7.2.5 for a brief illustration of how the theorem of the alternative is used for the **FR** process.

Three Notions of Singularity Degree

We now exhibit some properties that originate from the number of **FR** iterations.

Definition 2.4.2. *The* singularity degree *of* $\mathcal{F}$*, denoted* $\mathrm{sd}(\mathcal{F})$*, is the minimum number of* ***FR*** *iterations for finding* $\mathrm{face}(\mathcal{F}, \mathbb{S}^n_+)$*. The* maximum singularity degree *of* $\mathcal{F}$*, denoted* $\mathrm{maxsd}(\mathcal{F})$*, is the maximum number of nontrivial* ***FR*** *iterations for finding* $\mathrm{face}(\mathcal{F}, \mathbb{S}^n_+)$*.*

The singularity degree is often used to relate error bounds to explain the difficulty of solving problems numerically; see [162, 166]. It is known that a high singularity degree results in a worse *forward error bound* relative to the backward errors. The maximum singularity degree is a relatively new notion, and this motivates the idea of *implicit problem singularity,* $\mathrm{ips}(\mathcal{F})$. Every nontrivial step of **FR** results in redundant linear constraints. More specifically, **FR** reveals redundant equalities in $\mathcal{A}(\cdot) = b$; see [100]. The total number of these implicitly redundant constraints is called $\mathrm{ips}(\mathcal{F})$ and a short argument shows that $\mathrm{ips}(\mathcal{F}) \geqslant \mathrm{maxsd}(\mathcal{F})$. Proposition 2.4.3 below uses $\mathrm{maxsd}(\mathcal{F})$ to extend the result in [161, Lemma 3.5.2] and shows an interesting property that a **FR** sequence generates.[1]

Proposition 2.4.3. *[121, Theorem 1],[161, Lemma 3.5.2] Let the exposing vector obtained in the* i*-th* ***FR*** *iteration be nontrivial,* $Z^i = \mathcal{A}^*(\lambda^i) \neq 0, i = 1, \ldots, k \leqslant \mathrm{maxsd}(\mathcal{F})$*. Then the vectors,* $\lambda^1, \lambda^2, \ldots, \lambda^k$*, are linearly independent.*

We relate Proposition 2.4.3 to the dimension of the set of the roots that arise in the algorithm for solving the **BAP**, e.g., see Theorem 7.2.4.

2.4.2 Degeneracy and Relations to Strict Feasibility

Many of the concepts of degeneracy arise from the early work with the simplex method for linear programs. The *stalling* phenomenon of the simplex method is a well-known subject of research [26, 37, 129] and many methods are proposed to overcome this difficulty. In this section we use a generalized definition of degeneracy proposed by Pataki [144] to extend the discussion to spectrahedra. We then examine a connection between the *Slater constraint qualification* (strict feasibility), and degeneracy of feasible points. We identify a type of degeneracy that inevitably arises in the absence of strict feasibility.

Definition 2.4.4. *[144, Definition 3.3.1] A point* $X \in \mathcal{F}$ *is called nondegenerate if*

$$\mathrm{lin}(\mathrm{face}(X, \mathbb{S}^n_+)^\Delta) \cap \mathrm{range}(\mathcal{A}^*) = \{0\}.$$

Definition 2.4.4 immediately yields Lemma 2.4.5.

Lemma 2.4.5. *[144, Corollary 3.3.2] Let* $X \in \mathcal{F}$ *and let*

$$X = \begin{bmatrix} V & \bar{V} \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V & \bar{V} \end{bmatrix}^T, \; D \succ 0, \tag{2.11}$$

[1] We note that the concepts of $\mathrm{maxsd}(\mathcal{F}), \mathrm{ips}(\mathcal{F})$ did not yet exist in [161]. Moreover, it is shown empirically in [100] that ips is directly related to the forward error for **LP**s.

be a spectral decomposition of X*. Then*

$$\begin{array}{c} X \text{ is a nondegenerate point of } \mathcal{F} \\ \text{if, and only if,} \\ \left\{\begin{bmatrix} V^TA_iV & V^TA_i\bar{V} \\ \bar{V}^TA_iV & 0 \end{bmatrix}\right\}_{i=1}^{m} \text{ is a set of linearly independent matrices in } \mathbb{S}^n. \end{array} \tag{2.12}$$

Remark 2.4.6. *The degeneracy of a point* $X \in \mathcal{F}$ *can be identified by constructing a matrix using the characterization* (2.12)*. We let* $W_i := \begin{bmatrix} V^TA_iV & V^TA_i\bar{V} \\ \bar{V}^TA_iV & 0 \end{bmatrix}$, $i = 1, \ldots, m$ *and denote* $t(n) = n(n+1)/2$*, triangular number. We let* vec *denote the vectorization of a matrix by column; and* $\operatorname{svec} : \mathbb{S}^k \to \mathbb{R}^{t(k)}$ *denotes the isometry that vectorizes a symmetric matrix using the upper triangular part after multiplying the strict upper-triangular part by* $\sqrt{2}$*. We construct the following matrix* $L \in \mathbb{R}^{t(n)\times m}$ *with an appropriate permutation matrix* Π *and the* i*-th column:*

$$Le_i = \Pi \operatorname{svec} W_i = \begin{pmatrix} \operatorname{svec} V^TA_iV \\ \sqrt{2}\operatorname{vec} V^TA_i\bar{V} \\ \operatorname{svec} 0 \end{pmatrix}, \ \forall i \in \{1, \ldots, m\},$$

where V *and* $\bar{V}$ *are given in Lemma 2.4.5. Consider the matrix* $\bar{L}$*, with the* i*-th column* $\bar{L}e_i = \begin{pmatrix} \operatorname{svec} V^TA_iV \\ \sqrt{2}\operatorname{vec} V^TA_i\bar{V} \\ \operatorname{svec} \bar{V}^TA_i\bar{V} \end{pmatrix}$*. Note that* $\bar{L}$ *is full-column rank given* $\mathcal{A}$ *is surjective. The matrix* L *is obtained after zeroing out the last* $t(\operatorname{nullity}(X))$ *rows of* $\bar{L}$*. We note that* $\operatorname{rank}(L) < \operatorname{rank}(\bar{L})$ *(i.e., degeneracy holds) if, and only if, the orthogonal complement of the span of the first* $t(n) - t(\operatorname{nullity}(X))$ *rows of* $\bar{L}$ *has nonzero intersection with the span of the remaining rows that are then changed to* 0*. Therefore, if* $t(n) > m + t(\operatorname{nullity}(X))$*, then generically nondegeneracy holds.*

Theorem 2.4.7. *Suppose that* $\mathcal{F}$ *fails strict feasibility and let* $X = VDV^T \in \mathcal{F}$ *be found using* (2.11)*. Then the set* $\{A_iV\}_{i=1}^m$ *is linearly dependent. In particular, any solution* λ *to* (2.10) *certifies the linear dependence of the set* $\{A_iV\}_{i=1}^m$*. Thus every point in* $\mathcal{F}$ *is degenerate.*

Proof. Suppose that $\mathcal{F}$ fails strict feasibility. Let $X = VDV^T \in \mathcal{F}$ and let λ be a solution to the *auxiliary system* (2.10). Then $\mathcal{A}^*(\lambda)$ is an exposing vector and hence the inner product $\langle \mathcal{A}^*(\lambda), X\rangle = 0$ which yields

$$0 = \mathcal{A}^*(\lambda)X = (\mathcal{A}^*(\lambda)V)\,DV^T \implies 0 = \mathcal{A}^*(\lambda)V = \sum_{i=1}^{m} \lambda_i\,(A_iV)\,. \tag{2.13}$$

Since λ is a nonzero vector, (2.13) shows the linear dependence.

The linear dependence of the set $\{A_iV\}_{i=1}^m$ shown above allows for verifying *total* degeneracy that occurs in the absence of strict feasibility of $\mathcal{F}$. Then the above provides $\sum_{i=1}^m \lambda_iA_iV = 0, \lambda \neq 0$. We observe that

$$\begin{array}{rcccl} 0 & = & V^T\left(\sum_{i=1}^m \lambda_iA_iV\right) & = & \sum_{i=1}^m \lambda_iV^TA_iV, \\ 0 & = & \bar{V}^T\left(\sum_{i=1}^m \lambda_iA_iV\right) & = & \sum_{i=1}^m \lambda_i\bar{V}^TA_iV. \end{array}$$

This immediately implies that the matrices in (2.12) are linearly dependent and hence X is degenerate. □

In Corollary 2.4.8 we now connect nondegeneracy to strict feasibility.

Corollary 2.4.8. *Let $\mathcal{F}$ be given. Then the following holds:*

(i) If $\mathcal{F}$ contains a nondegenerate point, then strict feasibility holds.

(ii) Every $X \in \mathcal{F} \cap \mathbb{S}^n_{++}$ is nondegenerate.

Proof. Item (i) is the contrapositive of Theorem 2.4.7. Item (ii) is immediate from the definition of nondegeneracy, Definition 2.4.4, since $\operatorname{face}(X, \mathbb{S}^n_+)^\Delta = 0$, for all $X \succ 0$. □

Propositions 2.4.9 and 2.4.10 below provide sufficient conditions for the nondegeneracy of certain points of spectrahedra. They enable identifying nearest points where the semismooth Newton method for the **BAP** performs effectively.

Proposition 2.4.9. *Let $X_1, X_2 \in \mathcal{F}$ and let X_1 be a nondegenerate point. Then, $\gamma X_1 + (1-\gamma)X_2$ is a nondegenerate point of $\mathcal{F}$, for all $\gamma \in (0, 1]$.*

Proof. Let $X_1, X_2 \in \mathcal{F}$ and let X_1 be a nondegenerate point. Let $\gamma \in (0, 1]$ and $X' = \gamma X_1 + (1-\gamma)X_2$. We observe that

$$\begin{array}{rrcl} & \operatorname{face}(X_1, \mathbb{S}^n_+) & \subseteq & \operatorname{face}(X', \mathbb{S}^n_+) \\ \implies & \operatorname{face}(X_1, \mathbb{S}^n_+)^\Delta & \supseteq & \operatorname{face}(X', \mathbb{S}^n_+)^\Delta \\ \implies & \operatorname{lin}\left(\operatorname{face}(X_1, \mathbb{S}^n_+)^\Delta\right) & \supseteq & \operatorname{lin}\left(\operatorname{face}(X', \mathbb{S}^n_+)^\Delta\right) \\ \implies & \operatorname{lin}\left(\operatorname{face}(X_1, \mathbb{S}^n_+)^\Delta\right) \cap \operatorname{range}(\mathcal{A}^*) & \supseteq & \operatorname{lin}\left(\operatorname{face}(X', \mathbb{S}^n_+)^\Delta\right) \cap \operatorname{range}(\mathcal{A}^*). \end{array}$$

Since X_1 is a nondegenerate point, we have $\operatorname{lin}\left(\operatorname{face}(X_1, \mathbb{S}^n_+)^\Delta\right) \cap \operatorname{range}(\mathcal{A}^*) = \{0\}$. Thus, X' is a nondegenerate point. □

Proposition 2.4.10. *Let f be a face of $\mathcal{F}$ containing a nondegenerate point. Then every point in* $\operatorname{relint}(f)$ *is nondegenerate.*

Proof. Let $X_1 \in f$ be a nondegenerate point. For any $X \in \operatorname{relint}(f)$ there exists X_2 such that X belongs to the segment (X_1, X_2). The nondegeneracy of X then follows from Proposition 2.4.9. □

2.4.3 Facial Reduction for EDMs

Here we provide some properties of **FR** for **EDM**s. Recall that $f = \operatorname{face}(X, K) \trianglelefteq K$ denotes the minimal face of K ontaining X. We denote $f = \operatorname{face}(X)$ when $K = \mathbb{S}^n_+$. For $X \in \mathbb{S}^n_+$ with spectral decomposition $X = \begin{bmatrix} V & U \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V & U \end{bmatrix}^T$, $\Lambda \in \mathbb{S}^r_{++}$, we get

$$\operatorname{face}(X) = V\mathbb{S}^r_+ V^T, \ \operatorname{face}(X)^\Delta = U\mathbb{S}^{n-r}_+ U^T,$$

where $\operatorname{face}(X)^\Delta$ is the conjugate face of $\operatorname{face}(X)$. See e.g., [63].

We follow notation and definitions in [5, 63]. First we recall that both $\mathbb{S}^n_+, \mathcal{E}^n$ are closed convex cones in $\mathbb{S}^n$. We let W, V be two full column rank matrices that satisfy

$$\text{range}(V) = \text{range}(X),\ \text{range}(W) = \text{null}(X),\ Z = WW^T,\ d = \text{rank}(X).$$

Then $X \in \text{relint}(f)$ and we have two representations for f:

$$f = V\mathbb{S}^d_+V^T = \mathbb{S}^n_+ \cap Z^\perp. \tag{2.14}$$

We call V, Z a facial and exposing vector, respectively.

If we choose the facial vector to satisfy $\begin{bmatrix} V & e \end{bmatrix}$ nonsingular and $V^Te = 0$, then we can characterize the face of centered Gram matrices

$$\mathcal{S}^n_C \cap \mathbb{S}^n_+ = V\mathbb{S}^{n-1}_+V^T = \mathcal{K}^\dagger(\mathcal{E}^n) \trianglelefteq \mathbb{S}^n_+.$$

This is used in [9] to regularize the **EDM** completion problem using $\mathcal{K}_V(X) := \mathcal{K}(VXV^T)$, i.e., $\mathcal{K}_V : \mathbb{S}^{n-1}_+ \to \mathcal{E}^n$ and strict feasibility is satisfied for **EDM** completion problems. Here V is a *centered facial vector*, i.e., $V^Te = 0$. Since we work with centered Gram matrices, we often use centered facial vectors below, e.g., for the nearest **EDM** problem, **NEDM**, below.

Our divide and conquer methods use **EDM**s, D_n, principal submatrices D_α of D_n and corresponding Gram matrices, with ordered integers $\alpha = [i, i+k], 1 \leqslant i \leqslant i+k \leqslant n$. The first **FR** method finds exposing vectors for each submatrix and then adds these up to get an exposing vector for the entire Gram matrix G, i.e., we exploit Lemma 2.4.11.

Lemma 2.4.11 ([63]). *Let*

$$G \in \mathbb{S}^n_+,\ Z_i \in \mathbb{S}^n_+, \text{trace}(GZ_i) = 0, i = 1, \ldots, k.$$

Then $Z = \sum_{i=1}^k Z_i \succeq 0$ *and* $GZ = 0$.

The second **FR** method uses adjacent pairs of principal submatrices in order to do **FR** by intersecting pairs of faces. We first consider the representation of **FR** for a single principal submatrix, i.e., the representation of the smallest face obtained using the Gram matrix corresponding to D_α. Without loss of generality we use the first, top left, principal submatrix. We modify the notation in Theorem 2.4.12 to match the notation herein. Recall that $\mathcal{P}^{-1}_\alpha$ denotes the projection inverse image.

Theorem 2.4.12 ([113, Theorem 2.3]). *Let*

$$D \in \mathcal{E}^n, d = \text{embdim}(D),\ \alpha = [k],\ \bar{D} = D_\alpha, t = \text{embdim}(\bar{D}).$$

Let

$$\bar{G} = \mathcal{K}^\dagger(\bar{D}) = \bar{U}_G S \bar{U}_G^T,\ S \in \mathbb{S}^t_{++},$$

where

$$\bar{U}_G \in \mathbb{R}^{k\times t},\ \bar{U}_G^T\bar{U}_G = I_t,\ \bar{U}_G^Te = 0.$$

Furthermore, let $U_G := \begin{bmatrix} \bar{U}_G & \frac{1}{\sqrt{k}}e \end{bmatrix}$, $U := \begin{bmatrix} U_G & 0 \\ 0 & I_{n-k} \end{bmatrix}$, *and* $\begin{bmatrix} V & \frac{1}{\|U^Te\|}U^Te \end{bmatrix} \in \mathbb{R}^{n-k+t+1\times n-k+t+1}$

be orthogonal. Then

$$\operatorname{face}\left(\mathcal{K}^{\dagger}\left(\mathcal{P}_{\alpha}^{-1}(\bar{D})\cap\mathcal{E}^{n}\right)\right)=\left(U\mathbb{S}_{+}^{n-k+t+1}U^{T}\right)\cap\mathcal{S}_{C}^{n}=\left((UV)\mathbb{S}_{+}^{n-k+t}(UV)^{T}\right).$$

The matrix UV in Theorem 2.4.12 provides a facial vector for the minimal face corresponding to the block $\bar{D}$. If we have two overlapping principal submatrices with the overlap having the proper embedding dimension d, then we can efficiently find a facial vector for the intersection of the two corresponding faces. This is given in [113, Lemma 2.9] and we have added that option to our code and include the details here in Lemma 2.4.13 for completeness.

Lemma 2.4.13 ([113, Lemma 2.9])**.** *Let*

$$U_1:=\begin{array}{c}\\ s_1\\ k\end{array}\begin{array}{c}r+1\\ \left[\begin{array}{c}U_1'\\ U_1''\end{array}\right]\end{array},\quad U_2:=\begin{array}{c}\\ k\\ s_2\end{array}\begin{array}{c}r+1\\ \left[\begin{array}{c}U_2''\\ U_2'\end{array}\right]\end{array},\quad \hat{U}_1:=\begin{array}{c}\\ s_1\\ k\\ s_2\end{array}\begin{array}{c}r+1\quad s_2\\ \left[\begin{array}{cc}U_1' & 0\\ U_1'' & 0\\ 0 & I\end{array}\right]\end{array},\quad \hat{U}_2:=\begin{array}{c}\\ s_1\\ k\\ s_2\end{array}\begin{array}{c}s_1\quad r+1\\ \left[\begin{array}{cc}I & 0\\ 0 & U_2''\\ 0 & U_2'\end{array}\right]\end{array}$$

be appropriately blocked with $U_1'', U_2'' \in \mathcal{M}^{k\times(r+1)}$ full column rank and $\operatorname{range}(U_1'')=\operatorname{range}(U_2'')$. *Furthermore, let*

$$\bar{U}_1:=\begin{array}{c}\\ s_1\\ k\\ s_2\end{array}\begin{array}{c}r+1\\ \left[\begin{array}{c}U_1'\\ U_1''\\ U_2'(U_2'')^{\dagger}U_1''\end{array}\right]\end{array},\quad \bar{U}_2:=\begin{array}{c}\\ s_1\\ k\\ s_2\end{array}\begin{array}{c}r+1\\ \left[\begin{array}{c}U_1'(U_1'')^{\dagger}U_2''\\ U_2''\\ U_2'\end{array}\right]\end{array}. \tag{2.15}$$

Then $\bar{U}_1$ and $\bar{U}_2$ are full column rank and satisfy

$$\operatorname{range}(\hat{U}_1)\cap\operatorname{range}(\hat{U}_2)=\operatorname{range}\left(\bar{U}_1\right)=\operatorname{range}\left(\bar{U}_2\right).$$

Moreover, if $e_{r+1}\in\mathbb{R}^{r+1}$ is the $(r+1)^{\text{st}}$ standard unit vector, and $U_ie_{r+1}=\alpha_ie$, for some $\alpha_i\neq 0$, for $i=1,2$, then $\bar{U}_ie_{r+1}=\alpha_ie$, for $i=1,2$.

Remark 2.4.14. *In Lemma 2.4.13, if we have two overlapping blocks $D_{s_1\cup k}, D_{k\cup s_2}$ we can get facial vectors from bases of the corresponding Gram matrices in U_1, U_2, respectively. These are matched up in $\hat{U}_1, \hat{U}_2$. Then a facial vector for the intersection of the minimal faces containing these blocks is given in either of the formulae in (2.15). In our implementation, we choose the one for which U_1'', U_2'' is better conditioned. Thus we use the two facial vectors and find a new facial vector for the intersection of the overlapping ranges.*

This emphasizes the importance of the conditioning of the overlap in U_1'', U_2''. In fact, it is essential that the rank of each overlap is eventually d, the embedding dimension of the problem. But there is no reason that the ordering of the nodes (rows) of D cannot be changed. Therefore, if we find a well-conditioned block of correct rank d, we can include that in the overlap for further iterations, i.e., we can always save and use the best conditioned block with the largest rank in the overlap.

Part I

Preconditioning for Linear Systems

Chapter 3

Condition Numbers

In numerical analysis, a *condition number* of a matrix A is the main tool in the study of error propagation in the problem of solving a system of linear equations:

$$Ax = b,\ A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m.$$

The linear system $Ax = b$ is said to be well-conditioned when A has a small condition number. It aims to be a measure of how strongly a relative error in the data affects the relative error in the solution. Therefore, in simplified form the condition number of the linear system, it can be estimated as the ratio

$$\mathrm{cond}(A) := \frac{\|\Delta x\|/\|x\|}{\|\Delta b\|/\|b\|}, \quad \text{(rel. error output/rel. error input)} \tag{3.1}$$

e.g., [164, Sect. 1.3]. Note that the above ratio depends on the choice of the perturbation Δb, as well as on the choice of the norms. In general, iterative algorithms that solve $Ax = b$ require a large number of iterations to achieve a solution with sufficient accuracy if the problem is not well-conditioned, i.e., is ill-conditioned. In the literature $\kappa = \kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$ is taken as a *worst case* estimator of the condition number as the inequality

$$\mathrm{cond}(A) = \frac{\|\Delta x\|\|b\|}{\|x\|\|\Delta b\|} \leqslant \kappa(A),$$

holds for all $\Delta b \in \mathbb{R}^m$. See, e.g., [163], [87] or [165, Chapter 7] for further details. From the discussions in [51, 91], the (*worst case*) condition number for the linear system $Ax = b$ is

$$\begin{aligned} \mathrm{cond}(A,b) &:= \lim_{\epsilon\downarrow 0} \sup_{\substack{\|\Delta A\|\leqslant\epsilon\|A\| \\ \|\Delta b\|\leqslant\epsilon\|b\|}} \frac{\|(A+\Delta A)^{-1}(b+\Delta b)-A^{-1}b\|}{\epsilon\|A^{-1}b\|} \\ &= \kappa(A) + \frac{\|A^{-1}\|\|b\|}{\|A^{-1}b\|}. \end{aligned} \tag{3.2}$$

We emphasize that though many heuristics are given, the main measure of conditioning, e.g. [146], is κ. However, in our view, κ has the misleading property that it is *inverse invariant.*[1] In contrast, the ω-*condition number* distinguishes between the conditioning of A and A^{-1}. This observation

[1] $\kappa(A) = \|A\|\|A^{-1}\| = \kappa(A^{-1})$.

leads us to the discussion in Section 3.1, where we demonstrate that ω is a valid condition number. In particular, Remark 3.1.3 suggests that we should use the measure $\sqrt{\omega((A^TA)^{-1})}$, denoted ω_{inv}, rather than $\kappa(A)$ or $\omega(A)$.[2] As an empirical evidence, Section 3.1.1 provides numerical test that ω and ω_{inv} provide significantly better estimates for the conditioning of a linear system than κ. Moreover, it is expensive to evaluate the κ-condition number as it uses both $\|A\|, \|A^{-1}\|$. For large scale, one often uses the ℓ_1 approximation in [87]. We show in Section 3.1.2 that we can find the exact value of ω when a Cholesky or LU factorization is done.

Then our interest naturally move to the discussions on *preconditioners*. In order to improve the conditioning of a problem, preconditioners are employed for obtaining equivalent systems with better condition number. A recent survey on preconditioning is given in [146]. For example, in [47] a preconditioner that minimizes κ is obtained in the Broyden family of rank-two updates. Also, for applications to inexact Newton methods see [23, 24], where it is emphasized that the goal is to improve the *clustering of eigenvalues* around 1. The ω-condition number in particular uses *all* the eigenvalues, rather than just the largest and smallest as in the classical κ.[3] In Section 3.1.3, our empirics show that ω is more effective in promoting the important property of clustering of eigenvalues. The final and most important feature of the ω-condition number concerns optimal preconditioning. Both κ and ω are pseudoconvex over the open convex cone of positive definite matrices, $\mathbb{S}^n_{++}$; thus a local minimum is a global minimum. But, κ is differentiable if, and only if, both largest and smallest eigenvalues are singletons, while ω is differentiable on all of $\mathbb{S}^n_{++}$. This facilitates obtaining explicit formulae for optimal preconditioners and avoids expensive calculations., see e.g., [52]. In particular, our original motivation is to find well conditioned, ω-optimal, low rank updates of the positive definite generalized Jacobian that arises in nonsmooth Newton methods e.g., [35]. We use optimality conditions to find *explicit formulae* for these low rank updates. See Section 3.2. We continue the discussion on the optimal diagonal preconditioners in Chapter 4.

3.1 ω-Condition Number

The nonstandard condition number ω, defined as the ratio of the arithmetic and geometric means of the singular values, was proposed in [52]. Interestingly, the authors show that inverse-sized BFGS and sized DFP [137] arise as optimal quasi-Newton updates with respect to this measure.

For our purposes, we restrict attention to positive definite matrices $M(= A^TA) \succ 0$. Indeed, the solution to a linear system $Ax = b$[4] is given by $x^* = A^\dagger b = (A^TA)^{-1}A^Tb$ when A is overdetermined. Consequently, the conditioning of a linear system is closely related to the conditioning of $M^{-1} = (A^TA)^{-1}$ rather than that of A itself. See Remark 3.1.3 for further details.

The κ-, and ω-condition number for $M \succ 0$ are, respectively, given by ratios of eigenvalues

$$\kappa(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}; \quad \omega(M) = \frac{\sum_i \lambda_i(M)/n}{\prod_i (\lambda_i(M))^{1/n}} = \frac{\text{trace}(M)/n}{\det(M)^{1/n}}. \tag{3.3}$$

[2]Though ω_{inv} is currently for theoretical purposes only as it involves calculating $(A^TA)^{-1}$.

[3]Given a positive definite matrix $M \succ 0$, $\omega(M)$ is defined with eigenvalues rather than singular values. Throughout this thesis, we are working on a (possibly overdetermined) matrix $A \in \mathbb{R}^{m\times n}$, $m \geqslant n$, with full column rank. If A is symmetric, we use $\omega(A)$. If not, we use $\omega(A^TA)$ instead. This explains how we can safely use eigenvalues without modifying the definition of the condition numbers.

[4]Or to a least squares problem $\min \|Ax - b\|^2$ when the system is inconsistent.

The goal of this section is to demonstrate that the ω-condition number offers several advantages over the classical κ-condition number. In particular, ω provides a more accurate measure of conditioning, exhibits superior numerical stability, and is more sensitive to eigenvalue clustering. Moreover, its analytical simplicity facilitates effective preconditioner design, enabling the explicit construction of optimal low-rank updates.

In Section 3.1.1 we analytically and empirically illustrate that ω is a better indicator of conditioning for linear equations. Remark 3.1.3 provides the justification for using $\omega_{\text{inv}} := \sqrt{\omega((A^TA)^{-1})}$ as a measure and emphasizing the advantage over κ of not being inverse invariant. Efficiency and accuracy of computing ω is studied in Section 3.1.2. Indeed *the condition number of the condition number is the condition number* holds for κ but not for ω indicating that numerical calculations of ill-conditioned κ can be very inaccurate in contrast to ω. See Remark 3.1.4 Finally, we include empirical results for better clustering of eigenvalues, Figure 3.5 in Section 3.1.3.

We first observe some basic properties of ω. The following Proposition 3.1.1 allows us to find the ω-optimal preconditioners using the first-order optimality condition.

Proposition 3.1.1 ([52, Prop. 2.1 (iv)])**.** *The measure ω is pseudoconvex on the cone of symmetric positive definite matrices; thus every stationary point is a global minimizer of ω.*

We now include the gradients of the condition numbers for use in the definitions below. In the case of κ, for simplicity and to avoid the use of subgradients, we assume that the largest and smallest eigenvalues are singletons.

Lemma 3.1.2. *Let $M \in \mathbb{S}^n_{++}$ with eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_{n-1} \geqslant \lambda_n$, with corresponding orthonormal eigenvectors $v_1, \ldots, v_n$. Then:*

1. $$\begin{aligned}\nabla\omega(M) &= \tfrac{1}{n\det(M)^{1/n}}\left(I - \tfrac{\operatorname{trace} M}{n}M^{-1}\right) \text{ is indefinite,}\\ &\quad \text{with } \|\nabla\omega(M)\| = \tfrac{1}{n\det(M)^{1/n}}\max\left\{1 - \tfrac{\operatorname{trace} M}{n\lambda_1}, \tfrac{\operatorname{trace} M}{n\lambda_n} - 1\right\}.\end{aligned}$$

2. *In addition, assume that the largest and smallest eigenvaules of M are singletons, i.e., $\lambda_1 > \lambda_2 \geqslant \ldots \geqslant \lambda_{n-1} > \lambda_n$. Then:*

$$\nabla\kappa(M) = \frac{1}{\lambda_n}\left(v_1v_1^T - \kappa(M)v_nv_n^T\right), \text{ is indefinite, with } \|\nabla\kappa(M)\| = \frac{\kappa(M)}{\lambda_n}.$$

Proof. 1. The gradient is

$$\begin{aligned}\nabla\omega(M) &= \tfrac{1}{n\det(M)^{2/n}}\left(\det(M)^{1/n}I - \tfrac{\operatorname{trace} M}{n}\det(M)^{\frac{1}{n}-1}\operatorname{adj} M\right)\\ &= \tfrac{1}{n\det(M)^{1/n}}\left(I - \tfrac{\operatorname{trace} M}{n}M^{-1}\right),\end{aligned}$$

where adj M is the adjunct, the matrix of cofactors. The last expression follows from $M^{-1} = \frac{1}{\det(M)}\operatorname{adj} M$. The indefiniteness and norm follow from:

$$\begin{aligned}\lambda_{\max}(I - \tfrac{\operatorname{trace} M}{n}M^{-1}) &= \max_{\|x\|=1} x^T(I - \tfrac{\operatorname{trace} M}{n}M^{-1})x\\ &= 1 - \tfrac{\operatorname{trace} M}{n}\min_{\|x\|=1} x^TM^{-1}x\\ &= 1 - \tfrac{\operatorname{trace} M}{n\lambda_1}, \qquad \text{with attainment at } x = v_1,\\ &> 0;\end{aligned}$$

$$\begin{aligned}
\lambda_{\min}(I - \tfrac{\operatorname{trace} M}{n} M^{-1}) &= \min_{\|x\|=1} x^T (I - \tfrac{\operatorname{trace} M}{n} M^{-1}) x \\
&= 1 + \tfrac{\operatorname{trace} M}{n} \min_{\|x\|=1} (-x^T M^{-1} x) \\
&= 1 - \tfrac{\operatorname{trace} M}{n} \max_{\|x\|=1} x^T M^{-1} x \\
&= 1 - \tfrac{\operatorname{trace} M}{n\lambda_n}, \qquad \text{with attainement at } x = v_n, \\
&< 0.
\end{aligned}$$

2. Since the eigenvalues are singletons, they are differentiable with gradients $v_1 v_1^T, v_n v_n^T$, respectively. The result follows from the definitions of the gradient of the fractional function κ, the spectral norm, and orthonormality of the eigenvectors.

□

3.1.1 Error Analysis for Linear System

Recall that we are interested in understanding how small changes in the data affect the solution of the system. Let $x + \Delta x$ be a solution of the perturbed system

$$A(x + \Delta x) = b + \Delta b, \tag{3.4}$$

where $\Delta x, \Delta b \in \mathbb{R}^n$. The following remark provides new insight on the estimation of ill-conditioning of linear systems.

Remark 3.1.3. *Recall that we are interested in solving a linear system $Ax = b$, where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. Without loss of generality, we can assume that A is overdetermined, i.e., $m \geqslant n$ and A has full column rank. The solution is then given by $x^* = A^\dagger b$. Hence, the conditioning of the linear system can be understood as the conditioning of the operator $A^\dagger(\,\cdot\,)$.*

Let $A = USV^T, U \in \mathcal{O}^m$, $S \in \mathbb{R}^{m\times n}$ diagonal, and $V \in \mathcal{O}^n$, be the singular value decomposition of A. We get $A^\dagger = VS^\dagger U^T$, and in particular, $A^\dagger u_i = \frac{1}{\sigma_i} v_i$, for all $i \in [n]$. We can represent the right-hand side and its perturbation as linear combinations of u_i's: $b = \sum_{i=1}^m \beta_i u_i$, $\Delta b = \sum_{i=1}^m \delta_i u_i$. Then,

$$\begin{aligned}
\|x\|^2 &= \|A^\dagger b\|^2 \\
&= \|\textstyle\sum_{i=1}^m \beta_i A^\dagger u_i\|^2 \\
&= \|\textstyle\sum_{i=1}^n \frac{\beta_i}{\sigma_i} v_i\|^2 \\
&= \textstyle\sum_{i=1}^n \left(\frac{\beta_i}{\sigma_i}\right)^2 \\
&= \textstyle\sum_{i=1}^n \frac{\beta^2}{\lambda_i} \\
&= \tfrac{1}{n} \textstyle\sum_i \beta_i^2 \cdot \sum_i \lambda_i^{-1} + \operatorname{Cov}(\beta^2, \tfrac{1}{\lambda}),
\end{aligned}$$

where $\operatorname{Cov}(\beta^2, \frac{1}{\lambda})$ is the covariance of the vectors $\beta^2 := (\beta_1^2, \ldots, \beta_n^2)^T$ and $\frac{1}{\lambda} := (\lambda_1^{-1}, \ldots, \lambda_n^{-1})^T$. Generally, it is reasonable to assume that β^2 is uncorrelated with $\frac{1}{\lambda}$. Consequently, the covariance $\operatorname{Cov}(\beta^2, \frac{1}{\lambda})$ is assumed to be negligible. Under this assumption, we obtain the following approximation:

$$\begin{aligned}
\|x\|^2 &= \tfrac{1}{n} \textstyle\sum_i \beta_i^2 \cdot \sum_i \lambda_i^{-1} + \operatorname{Cov}(\beta^2, \tfrac{1}{\lambda}) \\
&\approx \tfrac{1}{n} \textstyle\sum_i \beta_i^2 \cdot \sum_i \lambda_i^{-1} = \|\beta\|^2 \textstyle\sum_i \lambda_i^{-1}/n.
\end{aligned}$$

Similarly,

$$\|\Delta x\|^2 = \|A^\dagger \Delta b\|^2 \approx \|\delta\|^2 \sum_i \lambda_i^{-1}/n \geqslant \|\delta\|^2 \prod_i \lambda_i^{-1/n}.$$

The last inequality is from the AM-GM inequality. From (3.1) (squared) we have:

$$\begin{aligned} \mathrm{cond}(A)^2 &= \frac{\|b\|^2}{\|x\|^2}\frac{\|\Delta x\|^2}{\|\Delta b\|^2} \\ &\lesssim \frac{\|b\|^2}{\|\Delta b\|^2}\frac{\|\delta\|^2\sum_i \lambda_i^{-1}/n}{\|\beta\|^2\prod_i \lambda_i^{-1/n}} \\ &= \frac{\sum_i \lambda_i^{-1}/n}{\prod_i \lambda_i^{-1/n}} = \omega((A^TA)^{-1}) \end{aligned} \tag{3.5}$$

Moreover, from [52, Prop. 2.1 (i)] and the inverse invariance of κ we have

$$1 \leqslant \sqrt{\omega((A^TA)^{-1})} \leqslant \sqrt{\kappa((A^TA)^{-1})} = \kappa(A) < \sqrt{4\omega^n((A^TA)^{-1})} = 2\sqrt{\omega^n((A^TA)^{-1})},$$

i.e., the measure $\sqrt{\omega((A^TA)^{-1})}$ (denoted $\omega_{\mathrm{inv}}(A)$ below) is a valid condition number.

The following numerical tests appear to confirm that it is indeed a more informative estimate of ill-conditioning. To see this, we study the correlation between the three estimates of cond: (i) κ, (ii) ω, and (iii) ω_{inv}, with the estimate of cond evaluated by sampling.

We sample as follows:

1. Generate 500 linear systems $A_ix = b_i, i \in [500]$, where the overdetermined $A_i \in \mathbb{R}^{300\times 200}$ are randomly generated with either uniformly or normally distributed positive singular values and the column vectors are set as $b_i := A_ix_i$, with x_i sampled from the standard normal distribution.
2. For each $i \in [500]$, we generate 1000 perturbations $\{\Delta b_j\}_{j\in[1000]}$ of norm 10^{-6} and set $\Delta x_j := A_i^\dagger \Delta b_j$. Then, for each $j \in [1000]$ we compute the relative residual ratio in (3.1). We then average over j to yield an estimate of the condition number, $\mathrm{cond}(A_i)$, of the i-th system.
3. We then check the resulting correlation between the following vectors (i) $(\kappa(A_i))_{i=1}^{500}$, (ii) $(\omega(A_i))_{i=1}^{500}$, and (iii) $(\omega_{\mathrm{inv}}(A_i))_{i=1}^{500}$, with $(\mathrm{cond}(A_i))_{i=1}^{500}$, by comparing the corresponding linear regression models.

Figures 3.1 and 3.2 show the linear regression models between the condition number, cond, empirically estimated from perturbed samples, and different notions of condition numbers $(\kappa, \omega, \omega_{\mathrm{inv}})$ for a random matrix A with uniformly, or normally, distributed singular values respectively. This numerical test reveals a significant linear correlation between cond and ω, ω_{inv}; whereas in contrast, cond and κ are not linearly correlated. The correlation coefficients are given in Table 3.1.

Distribution	κ	ω	ω_{inv}
Normal	0.0413	0.4012	0.7244
Uniform	0.4339	0.9182	0.9462

Table 3.1: Correlation between empirical condition estimate and different condition numbers.

3.1.2 Estimation of Condition Numbers

Since eigenvalue decompositions can be expensive, one issue with $\kappa(A)$ is how to estimate it efficiently when the size of matrix A is large. A survey of estimates and, in particular, estimates using

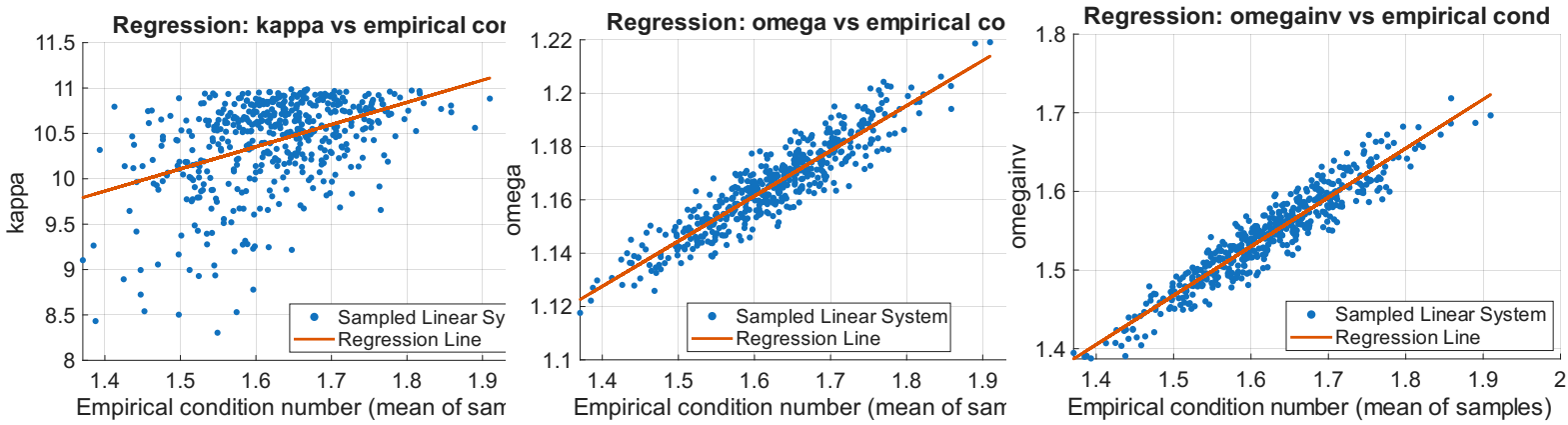

Figure 3.1: Linear regression models between cond and: κ, ω, ω_{inv}, respectively; uniformly distributed singular values.

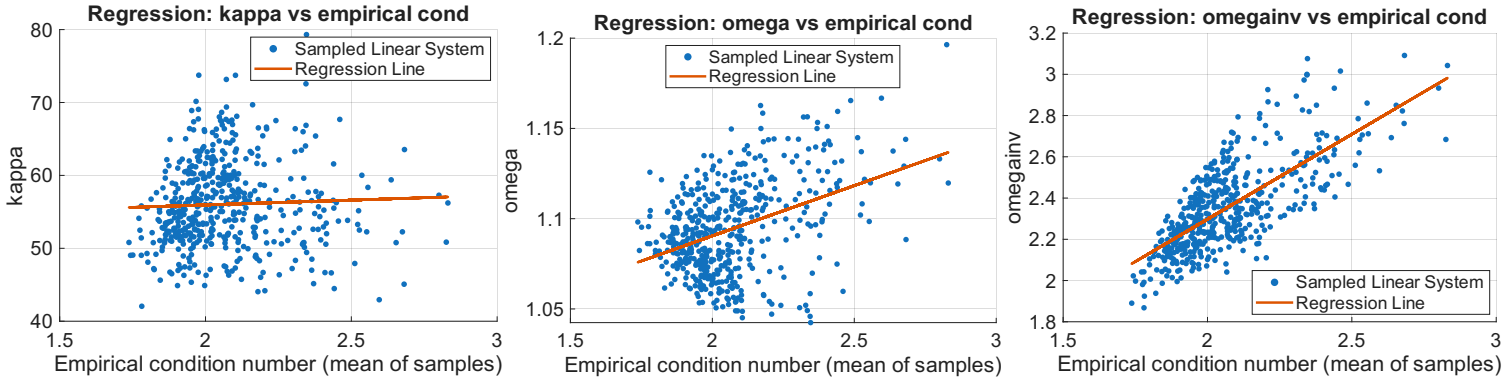

Figure 3.2: Linear regression models between cond and: κ, ω, ω_{inv}, respectively; normally distributed singular values.

the ℓ_1-norm, is given in [87, 92]. Extensions to sparse matrices and block-oriented generalizations are given in [84, 93]. Results from these papers form the basis of the *condest* command in MATLAB. More recently [71] deals with scalable methods for finding the κ-optimal diagonal preconditioner. This illustrates the difficulty in accurately estimating $\kappa(A)$.

On the other hand, the measure $\omega(A)$ can be calculated using the trace and determinant function that do not require eigenvalue decompositions. However, for large n, the determinant is also numerically difficult to compute as it could easily result in an overflow $+\infty$ or an underflow 0 due to the limits of finite precision arithmetic, e.g., if the order of A is $n = 50$ and the eigenvalues $\lambda_i = .5, \forall i$, then the determinant $.5^n$ is zero to machine precision. A similar problem arises for e.g., $\lambda_i = 2, \forall i$ with overflow. In order to overcome this problem, we take the n-th root first and then the product, i.e., we define the value obtained from the spectral factorization as

$$\omega_{\text{eig}}(A) = \frac{\sum_{i=1}^{n} \lambda_i(A)/n}{\prod_{i=1}^{n} (\lambda_i(A)^{1/n})}.$$

This quantity can be evaluated using either the Cholesky or LU factorization. Let $A = R^T R = LUP$ denote the Cholesky and LU factorizations, respectively, with appropriate permutation matrix P.

We assume that L is unit lower triangular. Therefore,

$$\det(A)^{1/n} = \det(R^T R)^{1/n} = \det(R)^{2/n} = \prod_{i=1}^{n} \left(R_{ii}^{2/n} \right). \tag{3.6}$$

Similarly,

$$\det(A)^{1/n} = \det(LUP)^{1/n} = \prod_{i=1}^{n} \left(|U_{ii}|^{1/n} \right). \tag{3.7}$$

Therefore, we find $\omega(A)$ with numerator $\text{trace}(A)/n$ and denominator given in (3.6) and (3.7), respectively:

$$\omega_R(A) = \frac{\text{trace}(A)/n}{\prod_{i=1}^{n} \left(R_{ii}^{2/n} \right)}, \quad \omega_{LU}(A) = \frac{\text{trace}(A)/n}{\prod_{i=1}^{n} \left(|U_{ii}|^{1/n} \right)}.$$

Tables 3.2 and 3.3 provide comparisons on the time and precision from the three different factorization methods. Each column presents different order of κ-condition number, while each row corresponds to different decompositions with different size n of the problem. We form the random matrix using $A = QDQ^T$ for random orthogonal Q and positive definite diagonal D. We then symmetrize $A \leftarrow (A + A^T)/2$ to avoid roundoff error in the multiplications. Therefore, we consider the evaluation using D as the *exact value* of $\omega(A)$, i.e.,

$$\omega(A) = \frac{\sum_{i=1}^{n} (D_{ii})/n}{\prod_{i=1}^{n} \left(D_{ii}^{1/n} \right)}.$$

Table 3.3 shows the absolute value of the difference between the exact ω-condition number and the ω-condition numbers obtained by making use of each factorization, namely, ω_{eig}, ω_R and ω_{LU}. Surprisingly, we see that both the Cholesky and LU decompositions give better results than the eigenvalue decomposition.

n	Fact.	order κ 1e2	order κ 1e3	order κ 1e4	order κ 1e5	order κ 1e6	order κ 1e7	order κ 1e8	order κ 1e9
500	eig	5.5267e-02	5.7766e-02	5.2747e-02	5.9256e-02	6.0856e-02	6.2197e-02	5.5592e-02	5.7626e-02
	R	1.1218e-02	8.0907e-03	7.5172e-03	8.4705e-03	9.2774e-03	8.5553e-03	8.1462e-03	7.9027e-03
	LU	2.2893e-02	1.8159e-02	1.8910e-02	2.0902e-02	2.0057e-02	2.0308e-02	1.9060e-02	1.8879e-02
1000	eig	3.0664e-01	2.8968e-01	2.6095e-01	2.7796e-01	5.7083e-01	5.9007e-01	5.8351e-01	5.9630e-01
	R	2.9328e-02	2.8339e-02	2.7869e-02	3.1909e-02	5.8628e-02	6.0873e-02	6.2429e-02	6.1074e-02
	LU	7.5011e-02	7.2666e-02	7.0497e-02	7.6778e-02	1.6313e-01	1.7313e-01	1.7666e-01	1.7326e-01
2000	eig	3.4794e+00	3.4804e+00	3.1916e+00	3.4386e+00	3.4235e+00	3.4766e+00	3.2327e+00	3.3704e+00
	R	3.5644e-01	3.5989e-01	2.9556e-01	3.6375e-01	3.5847e-01	3.5972e-01	3.2629e-01	3.4227e-01
	LU	9.0136e-01	9.0537e-01	7.1161e-01	8.7445e-01	8.6420e-01	8.8027e-01	8.1990e-01	8.1383e-01

Table 3.2: CPU sec. for evaluating $\omega(A)$, averaged over the same 10 random instances; eig, R, LU are eigenvalue, Cholesky, LU decompositions, respectively.

Condition number of condition numbers

If we regard the condition numbers κ and ω as operators, we can define and compute the *condition number of condition numbers*; see, e.g., [50,91]. The following Remark 3.1.4 discusses the first order condition number of κ and ω. In particular, for κ, it is well known that *the condition number of the condition number is the condition number.*

n	Fact.	order κ 1e2	order κ 1e3	order κ 1e4	order κ 1e5	order κ 1e6	order κ 1e7	order κ 1e8	order κ 1e9
500	eig	1.5632e-13	2.7853e-12	2.2618e-10	1.2695e-08	8.9169e-07	5.4109e-05	2.2610e-03	1.7349e-01
	R	1.7053e-13	2.5580e-12	1.0039e-10	1.1339e-08	4.9818e-07	2.6470e-05	1.3173e-03	1.6217e-01
	LU	1.5987e-13	2.4585e-12	1.0652e-10	1.1987e-08	5.1592e-07	2.1372e-05	1.3641e-03	1.4268e-01
1000	eig	2.1316e-13	2.1032e-12	8.7653e-11	4.6271e-09	3.1477e-07	1.9602e-05	9.9290e-04	7.6469e-02
	R	4.2633e-13	1.5632e-12	4.2235e-11	3.9297e-09	2.9562e-07	1.1498e-05	9.1506e-04	5.3287e-02
	LU	4.4054e-13	1.4850e-12	3.7858e-11	3.8287e-09	2.7390e-07	1.3820e-05	6.0492e-04	4.8568e-02
2000	eig	2.4336e-13	4.1780e-12	4.2019e-10	2.0080e-08	7.7358e-07	6.4819e-05	5.5339e-03	3.7527e-01
	R	4.3698e-13	2.0819e-12	5.0704e-11	2.3442e-09	1.8376e-07	8.9575e-06	5.5255e-04	4.8842e-02
	LU	4.3165e-13	2.2595e-12	2.3249e-11	2.5057e-09	1.5020e-07	6.0479e-06	5.4228e-04	4.4205e-02

Table 3.3: Precision of evaluation of $\omega(A)$ averaged over the same 10 random instances. eig, R, LU are eigenvalue, Cholesky, LU decompositions, respectively.

Remark 3.1.4. *If we consider b as the input to a function G with output x, then a Taylor type argument gives to first order condition number as in* (3.1)

$$\text{cond}(G) = \frac{\|b\|}{\|G(b)\|}\frac{\|\Delta G\|}{\|\Delta b\|} \cong \|\nabla G(b)\|\frac{\|b\|}{\|G(b)\|}, \tag{3.8}$$

a first order approximation for the condition number of G. Using the result in Lemma 3.1.2, we compute the condition number of κ:

$$\text{cond}(\kappa) = \|\nabla\kappa(A)\|\frac{\|A\|}{\kappa(A)} = \frac{\kappa(A)}{\lambda_n}\frac{\lambda_1}{\kappa(A)} = \kappa(A).$$

We have observed empirically that the condition number of ω is significantly smaller than the condition number of κ. Figure 3.3 illustrates the difference in magnitude between the condition numbers of κ and ω over random matrices of size $10 \leqslant n \leqslant 300$. Figure 3.4 illustrates a comparison

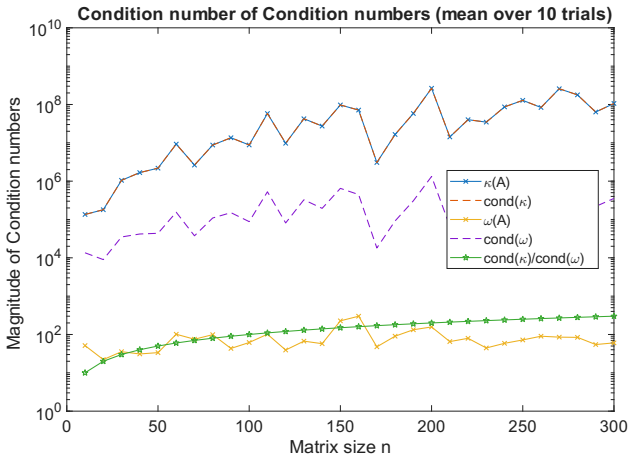

Figure 3.3: Condition number of condition numbers

of accuracy in evaluations of ω, κ. We use one positive definite matrix with spectral decomposition $A = QDQ^T$, and $n = 1000$, density $= 10^{-4}$ with various κ from 10 to 1000. We use perturbations of the eigenvalues $\|D(\epsilon) - D\|/\|D\| = 1e-8$ and reform $A(\epsilon) = QD(\epsilon)Q^T$. Figure 3.4 clearly shows that ω is calculated more accurately as the ill-conditioning grows. This relates to the *condition number of the condition number* in Remark 3.1.4.

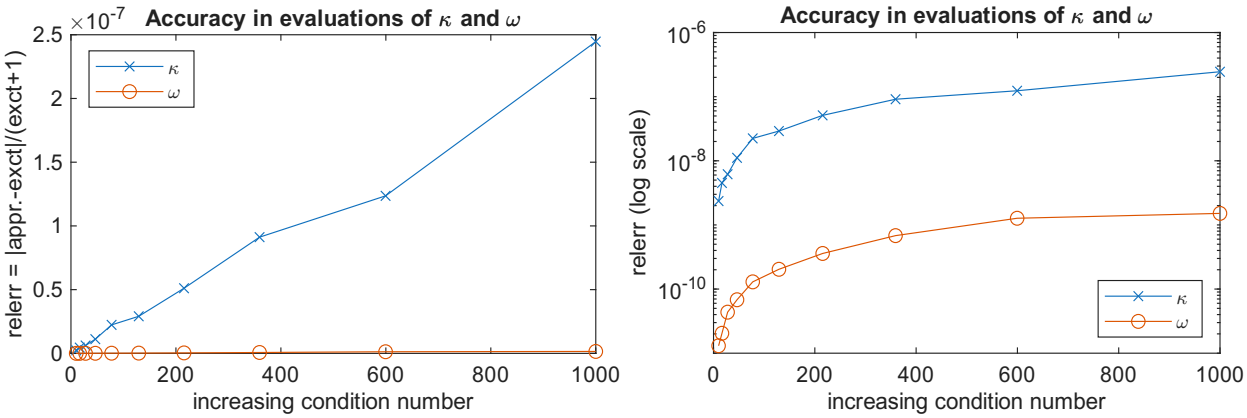

Figure 3.4: Comparing accuracy under perturbations for calculating ω, κ.

3.1.3 Eigenvalue Clustering

As mentioned above, the clustering of eigenvalues is crucial for iterative methods for solving linear systems. Many convergence results depend on the degree of eigenvalue clustering; see, e.g., [82].

A typical comparison of eigenvalue clustering for $W \succ 0$ after optimal diagonal preconditioning with respect to κ, ω, and ω_{inv} is shown in Figure 3.5. See Chapter 4 for further details on optimal diagonal preconditioning. The left panel illustrates the distribution of eigenvalues, while the right panel shows the number of eigenvalues (out of 60) contained within bands of varying widths centered at the mean eigenvalue. We interpret tighter clustering as having more eigenvalues contained within narrower bands.

The left panel clearly demonstrates improved eigenvalue clustering under ω-based preconditioning. While κ-optimal preconditioning primarily rescales the spectrum to reduce the ratio $\lambda_{\max}/\lambda_{\min}$, both ω-based measures drive the eigenvalues closer to 1, thereby promoting stronger clustering. This improvement is reflected in the right panel, where more eigenvalues are concentrated near the mean in the ω-optimal preconditioned matrices.

Table 3.4 further empirically illustrates that the ω-condition number correlates better with the number of iterations needed for **pcg**. In the table we fix κ by fixing the smallest $\lambda_{\min}$ and largest $\lambda_{\max}$ eigenvalues and also fix the orthogonal matrix of eigenvectors Q in the spectral decomposition $A = Q\,\text{Diag}(\lambda)Q^T$. We generate 5 random sparse matrices whose $(n-2)$-non extremal eigenvalues have the same mean but follow different distributions, i.e., they can be either uniformly distributed or normally distributed with different standard deviations, resulting in different values of ω. The first column of Table 3.4 reports the distribution used for the non-extremal eigenvalues, the second column gives the corresponding standard deviation, and the third column lists the resulting ω-condition number of each matrix. Next, we generate a column vector b which is in the range of the sum of the 5 matrices plus a random error term of order 10^{-7}. Finally, we use MATLAB's **pcg** with a tolerance set at 10^{-6} and solve the 5 linear systems given by b and each of the 5 matrices. No preconditioner is used in any of the experiments. The fourth column reports the number of PCG iterations required for convergence for each system. The correlation (monotonicity) between the std, ω, iteration count in Table 3.4 is clear. We include the fifth and sixth columns to highlight the significant correlation in the percentage reductions relative to the worst case, namely the first row of the table. Let ω_1 and iter_1 denote the ω value and iteration count in the first row, and let ω_i

and iter_i denote the corresponding quantities for the i-th row. The fifth column reports the relative reduction in ω,

$$\frac{(\omega_1 - 1) - (\omega_i - 1)}{\omega_1 - 1} \times 100\%,$$

which measures how much closer ω_i is to the ideal value 1 compared with the worst case. Similarly, the sixth column reports the relative reduction in the iteration count,

$$\frac{\text{iter}_1 - \text{iter}_i}{\text{iter}_1} \times 100\%,$$

which measures the percentage decrease in the number of iterations required by **pcg** relative to the first row.

Distribution	std	ω	# Iterations	% Reduct. ω	% Reduct. Iter.	Rel. Resid.
Uniform	1.163e+11	1.3682	264	-	-	9.738e-07
Normal	4.031e+10	1.0308	14	91.63	94.70	5.596e-07
Normal	2.022e+10	1.0143	8	96.11	96.97	9.834e-07
Normal	9.205e+09	1.0101	6	97.25	97.73	4.432e-07
Normal	6.095e+09	1.0095	5	97.42	98.11	6.067e-07

Table 3.4: correlations: eigenvals standard deviation (std); ω; iterations

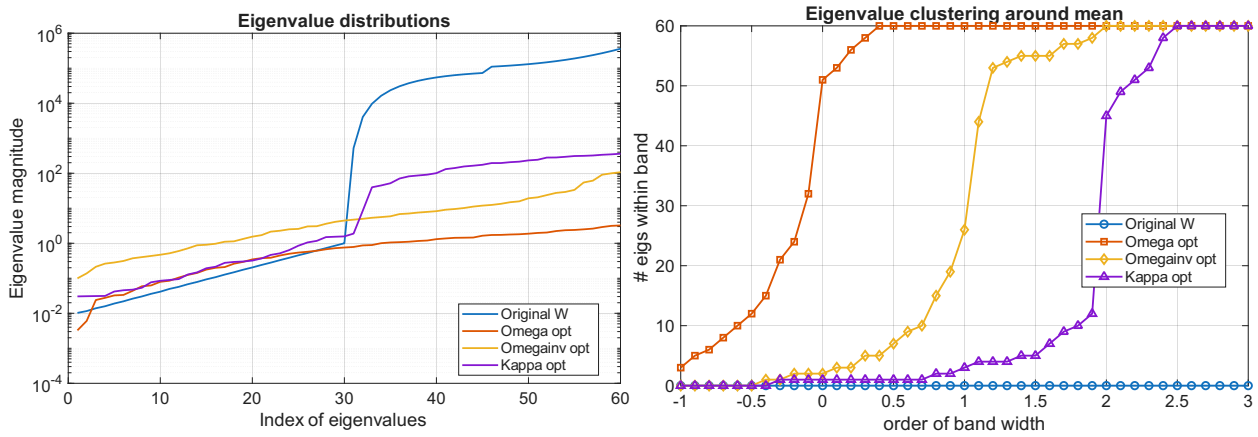

Figure 3.5: Comparison for clustering of eigenvalues pre-post preconditioning

3.2 ω-Optimal Low Rank Updating

In the second part of this chapter, we consider the problem of improving conditioning for low rank updates of very ill-conditioned (close to singular) positive definite matrices. The analytic simplicity of the definition of ω-condition number allows us to exploit the derivative of ω to derive optimality conditions.

3.2.1 Preliminaries

Given a positive definite matrix $A \in \mathbb{S}^n_{++}$ and a matrix $U \in \mathbb{R}^{n\times t}$ with $t << n$, we aim to find $\gamma \in \mathbb{R}^t$ so as to minimize the condition number of the low rank update

$$A + U\,\mathrm{Diag}(\gamma)U^T. \tag{3.9}$$

This kind of updating arises when finding generalized Jacobians in nonsmooth optimization. We provide insight on the problem in the following Example 3.2.1.

Example 3.2.1 (Generalized Jacobians)**.** *In many nonsmooth and semismooth Newton methods one aims to find a root of a function $F : \mathbb{R}^n \to \mathbb{R}^n$ of the form*

$$F(y) := B(v + B^T y)_+ - c,$$

where $B \in \mathbb{R}^{n\times m}$, $v \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $(\,\cdot\,)_+$ denotes the projection onto the nonnegative orthant, e.g., [35, 98, 148]. At every iteration of these algorithms a generalized Jacobian of F of the form

$$J := \sum_{i\in\mathcal{I}_+} B_i B_i^T + \sum_{j\in\mathcal{I}_0} \gamma_j B_j B_j^T, \ \text{with } \gamma_j \in [0,1],$$

is computed. Here B_i and B_j denote columns of B over the set of indices

$$\begin{array}{rcl} \mathcal{I}_+ & := & \{i \in [m] : (v + B^T y)_i > 0\};\ \text{and} \\ \mathcal{I}_0 & := & \{j \in [m] : (v + B^T y)_j = 0 \text{ and } (B_j)_{j\in\mathcal{I}_0} \text{ is a maximal linearly independent set}\}. \end{array}$$

The generalized Jacobian J, is usually singular. It is used to obtain a Newton direction $d \in \mathbb{R}^n$ by solving a least-square problem for the system $(J + \epsilon I)\,d = -F(y)$, where ϵI, with $\epsilon > 0$, is analogous to the regularization term of the well-known Levenberg–Marquardt method*. Thus, this linear system is in general very ill-conditioned. This makes preconditioning by optimal updating appropriate.*

The optimal preconditioned update can be done in our framework as we start with

$$A := \sum_{i\in\mathcal{I}_+} B_i B_i^T + \epsilon I, \quad U = [B_j]_{j\in\mathcal{I}_0}, \tag{3.10}$$

and then find an optimal low rank update as in (3.9); done with additional box constraints on γ, namely, $\gamma \in [0,1]^t$.

Similar conditioning questions also appear in the normal equations matrix, ADA^T, in interior point methods, e.g., modifying the weights in D appropriately to avoid ill-conditioning [42,77]. For other related work on minimizing condition numbers for low rank updates see, e.g., [39, 83].

Here, we propose obtaining an optimal conditioning of the update (3.9) by using the ω-condition number of [52], instead of the classic κ-condition number. The ω-condition number presents some advantages with respect to the classic condition number, since it is differentiable and pseudoconvex in the interior of the positive semidefinite cone, which facilitates addressing minimization problems involving it. Our empirical results show a significant decrease in the number of iterations required for a requested accuracy in the residual.

3.2.2 Optimal Conditioning for Rank One Updates

We first consider the special case where the update is rank one. Related eigenvalue results for rank one updates are well known in the quasi-Newton literature, e.g., [53, 156]. We include this special rank one case as it yields interesting results. The general rank-t update is studied in Section 3.2.3, below.

Theorem 3.2.2. *Suppose we have a given $A \in \mathbb{S}^n_{++}$ and $u \in \mathbb{R}^n$. Let $A = QDQ^T$ be the (orthogonal) spectral decomposition of A. Let $U = uu^T$ and define the rank one update*

$$A(\gamma) = A + \gamma U, \ \gamma \in \mathbb{R}.$$

Set

$$w = D^{-1/2}Q^T u, \tag{3.11}$$

and

$$\gamma^* = \frac{\operatorname{trace}(A)\|w\|^2 - n\|u\|^2}{(n-1)\|u\|^2\|w\|^2}. \tag{3.12}$$

Then, $\gamma^* \in \left]-\|w\|^{-2}, +\infty\right[$ and it provides the ω-optimal conditioning, i.e.,

$$\gamma^* = \operatorname*{argmin}_{A(\gamma)\succ 0} \omega(\gamma). \tag{3.13}$$

Proof. Let

$$f(\gamma) := \operatorname{trace}(A(\gamma))/n \quad \text{and} \quad g(\gamma) := \det(A(\gamma))^{1/n}.$$

We want to find the optimal γ to minimize the condition number

$$\omega(\gamma) = f(\gamma)/g(\gamma)$$

subject to $A(\gamma)$ being positive definite. By Proposition 3.1.1, $\omega : \mathbb{R} \to \mathbb{R}$; $\gamma \to \omega(\gamma)$ is pseudoconvex as long as $A(\gamma) \succ 0$. We prove that the latter is true for γ belonging to an open interval in the real line. Indeed, let $A = QDQ^T$ be the spectral decomposition of A and define

$$w = D^{-1/2}Q^T u \quad \text{and} \quad W = ww^T = D^{-1/2}Q^T uu^T QD^{-1/2}. \tag{3.14}$$

Then we can rewrite

$$A(\gamma) = QD^{1/2}(I + \gamma W)D^{1/2}Q^T, \tag{3.15}$$

which is positive definite if and only if the rank one update of I, $I + \gamma W$, belongs to the cone of positive definite matrices. Now, note that the eigenvalues of this term are $\lambda_1 = 1$, with multiplicity $n-1$, and $\lambda_2 = 1 + \gamma\|w\|^2$ with multiplicity 1. We then conclude that

$$A(\gamma) \in \mathbb{S}^n_{++} \iff \gamma \in \left]-\frac{1}{\|w\|^2}, +\infty\right[,$$

in which case $\lambda_2 > 0$. Moreover, $\omega(\gamma)$ tends to ∞ as γ approaches the extreme of the above interval. Therefore ω possesses a minimizer in the open interval, $\gamma^* \in \left]-\|w\|^{-2}, +\infty\right[$, that satisfies $\omega'(\gamma^*) = 0$. Note that since ω is pseudoconvex the fact that its derivative is equal to zero is also a sufficient condition for global optimality (see Fact 3.2.7 below).

In the following we obtain an explicit expression for the (unique) minimizer of (3.13), γ^*, by studying the zeros of ω'. Using the notation introduced in (3.14), f and its derivative are expressed as

$$f(\gamma) = \left(\operatorname{trace}(A) + \gamma\|u\|^2\right)/n \quad \text{and} \quad f'(\gamma) = \|u\|^2/n,$$

respectively. By making use of (3.15), g becomes

$$g(\gamma) := \left(\det(A)\det(I+\gamma W)\right)^{1/n},$$

since $\det(D) = \det(A)$. As explained above the eigenvalues of $I + \gamma W$, are $\lambda_1 = 1 + \gamma\|w\|^2$, and the others are all 1, which yields that

$$g(\gamma) = \left(\det(A)(1+\gamma\|w\|^2)\right)^{1/n} = \det(A)^{1/n}(1+\gamma\|w\|^2)^{1/n}.$$

We get

$$g'(\gamma) = \frac{1}{n}\det(A)^{1/n}\|w\|^2(1+\gamma\|w\|^2)^{(1-n)/n}.$$

The derivative of ω is then obtained as follows

$$\begin{aligned}
\omega'(\gamma) &= \frac{f'(\gamma)g(\gamma) - f(\gamma)g'(\gamma)}{g(\gamma)^2} \\
&= \frac{1}{g(\gamma)^2}\left[\frac{\|u\|^2}{n}\det(A)^{1/n}\left(1+\gamma\|w\|^2\right)^{1/n}\right. \\
&\quad \left. -\frac{\|w\|^2}{n^2}\left(\operatorname{trace}(A)+\gamma\|u\|^2\right)\det(A)^{1/n}\left(1+\gamma\|w\|^2\right)^{(1-n)/n}\right] \\
&= \frac{\det(A)^{1/n}}{g(\gamma)^2 n^2}\left(1+\gamma\|w\|^2\right)^{(1-n)/n}\left[n\|u\|^2 + (n-1)\gamma\|u\|^2\|w\|^2 - \operatorname{trace}(A)\|w\|^2\right].
\end{aligned} \tag{3.16}$$

A simple computation shows that this derivative is 0 only when γ attains the value

$$\gamma^* = \frac{\operatorname{trace}(A)\|w\|^2 - n\|u\|^2}{(n-1)\|u\|^2\|w\|^2}, \tag{3.17}$$

that has to be in the open interval $\left]-\|w\|^{-2}, +\infty\right[$. Since ω is pseudoconvex, we conclude that γ^* is the ω-optimal conditioning that solves (3.13). □

Equivalently, we can deduce an expression for the ω-optimal conditioning by making use of the Cholesky decomposition of A instead of the spectral decomposition. This is gathered in our next corollary. The proof follows from the same calculations than Theorem 3.2.2 and thus is omitted.

Corollary 3.2.3. *Given A and U as in Theorem 3.2.2. Let $A = LL^T$ be the Cholesky decomposition of A. Then, the formula for the ω-optimal conditioning γ^* in (3.12) holds with the replacement*

$$w \leftarrow L^{-1}u.$$

As shown in Example 3.2.1, in some applications the preconditioner multiplier γ is required to take values in the interval $[0, 1]$. In the following, we analyze the optimal ω-preconditioner for the rank 1 update subject to this interval constraint.

Corollary 3.2.4. *Let the assumptions of Theorem 3.2.2 hold and let $\bar{\gamma}$ be the ω-optimal conditioning in the interval $[0,1]$, i.e.,*

$$\bar{\gamma} = \arg \min_{\substack{0 \leqslant \gamma \leqslant 1 \\ A(\gamma) > 0}} \omega(\gamma).$$

Then, if $\gamma^* \in \left]-\|w\|^2, +\infty\right[$ is the ω-optimal "unconstrained" conditioning obtained in Theorem 3.2.2, the following hold:

(i) If $\gamma^* \in [0,1] \implies \bar{\gamma} = \gamma^*$;

(ii) If $\gamma^* < 0 \implies \bar{\gamma} = 0$;

(iii) If $\gamma^* > 1 \implies \bar{\gamma} = 1$.

Proof. Item (i) In this case, since γ^* is the global optimum of ω in $\left]-\|w\|^2, +\infty\right[$, it would also be so in the interval $[0,1]$.

For Items (ii) and (iii), it suffices to observe that, by (3.16) and (3.17), when $\gamma^* < 0$ (respectively, $\gamma^* > 1$) the derivative of ω is monotonically increasing (respectively, decreasing) in the interval $[0,1]$. □

3.2.3 Optimal Conditioning with a Low Rank Update

We now consider the case where the update is low rank. We need the following notations. For a matrix $Z \in \mathbb{R}^{n \times t}$, we use MATLAB notation and define the function $\text{norms}(Z) : \mathbb{R}^{n \times t} \to \mathbb{R}^t$ as the (column) vector of column 2-norms of Z. We let $\text{norms}^\alpha(Z)$ denote the vector of column norms with each norm to the power α.

Theorem 3.2.5 (Rank t-update)**.** *Let $A \in \mathbb{S}^n_{++}$, $U = [u_1, \ldots, u_t] \in \mathbb{R}^{n \times t}$, be given with $n > t \geqslant 2$, and $\text{norms}(U) > 0$. We further assume that $U^T A^{-1} U$ is diagonal.*[5] *Set*

$$A(\gamma) = A + U \operatorname{Diag}(\gamma) U^T, \text{ for } \gamma \in \mathbb{R}^t.$$

Let the spectral decomposition of A be given by $A = QDQ^T$, define $w_i = D^{-1/2} Q^T u_i, i \in [t]$, as in (3.11), with $W = \begin{bmatrix} w_1 \ldots w_t \end{bmatrix}$. Let

$$\begin{aligned} K(U) &= \left[n \operatorname{Diag}\left(\text{norms}^2(U)\right) - e\, \text{norms}^2(U)^T \right], \\ b(U) &= \left(\text{trace}(A) e - n \operatorname{Diag}\left(\text{norms}^2(W)\right)^{-1} \text{norms}^2(U) \right), \end{aligned} \tag{3.18}$$

where e denotes the vector of all ones. Then, the ω-optimal conditioning,

$$\gamma^* = \operatorname*{argmin}_{A(\gamma) > 0} \omega(\gamma), \tag{3.19}$$

[5] This assumption is not restrictive for the motivation in Example 3.2.1. An invertible change of coordinates can be applied to the columns of B and the vectors v and c so that the corresponding analogue of $U^T A^{-1} U$ is diagonal. Consequently, the problem of finding a zero of $F(y)$ remains equivalent up lto a reparameterization.

is given component-wise for $i \in [t]$ *by*

$$\begin{aligned}(\gamma^*)_i &= (K(U)^{-1}b(U))_i \\ &= \frac{1}{(n-t)\|u_i\|^2}\left(\operatorname{tr}(A) - (n-t)\frac{\|u_i\|^2}{\|w_i\|^2} - \sum_{j=1}^{t}\frac{\|u_j\|^2}{\|w_j\|^2}\right).\end{aligned} \tag{3.20}$$

Proof. Let $A \succ 0$ and

$$U = \begin{bmatrix} u_1 & \dots & u_t \end{bmatrix} \in \mathbb{R}^{n\times t}, \qquad \text{with } n > t \geqslant 2.$$

We consider the update of the form

$$A(\gamma) = A + U\operatorname{Diag}(\gamma)U^T = A + \sum_{i=1}^{t}\gamma_i u_i u_i^t, \quad \gamma \in \mathbb{R}^t.$$

Same than in Theorem 3.2.2, we start characterizing an open subset of $\mathbb{R}^t$ where $A(\gamma)$ is positive definite. In order to do this, we again transform the problem using the spectral decomposition of A, $A = QDQ^T$, and setting

$$w_i = D^{-1/2}Q^T u_i \quad \text{and} \quad W_i = w_i w_i^T \quad \text{for } i \in [t].$$

From the assumption that $U^T A^{-1} U$ is diagonal, we know that the vectors w_i, $i = 1, \dots, t$, are mutually orthogonal. Then, we can express $A(\gamma)$ as

$$\begin{aligned} A(\gamma) &= A + U\operatorname{Diag}(\gamma)U^T \\ &= QD^{1/2}\left(I + D^{-1/2}Q^T U\operatorname{Diag}(\gamma)U^T QD^{-1/2}\right)D^{1/2}Q^T \\ &= QD^{1/2}\left(I + \sum_{i=1}^{t}\gamma_i(D^{-1/2}Q^T u_i)(u_i^T QD^{-1/2})\right)D^{1/2}Q^T \\ &= QD^{1/2}\left(I + \sum_{i=1}^{t}\gamma_i W_i\right)D^{1/2}Q^T. \end{aligned}$$

Hence, we obtain the following expression for the determinant of $A(\gamma)$

$$\det\left(A(\gamma)\right) = \det(A)\prod_{i=1}^{t}(1 + \gamma_i\|w_i\|^2). \tag{3.21}$$

Consequently, $A(\gamma)$ is nonsingular and, by continuity of the eigenvalues, positive definite for γ belonging to the set

$$\Omega := \left]-\frac{1}{\|w_1\|^2}, +\infty\right[\times \left]-\frac{1}{\|w_2\|^2}, +\infty\right[\times \dots \times \left]-\frac{1}{\|w_t\|^2}, +\infty\right[. \tag{3.22}$$

Now, note that the constraint $A(\gamma) \succ 0$ is a positive definite constraint, so it is convex. Therefore, if there exists some γ outside of Ω such that $A(\gamma) \succ 0$, we would lose the convexity of the feasible set, since $A(\gamma)$ is singular on the boundary of Ω. This implies that

$$A(\gamma) \succ 0 \iff \gamma \in \Omega.$$

Moreover, since $\omega(\gamma) \to +\infty$ as γ tends to the border of Ω or to $+\infty$, we can ensure that γ

has a minimizer in Ω. Since the function is pseudoconvex on this open set, the global minimum is attained at a point γ^* such that $\nabla\omega(\gamma^*) = 0$. Next, we prove that γ^* is given by (3.20).

For this, note that $f(\gamma)$ can be expressed as

$$f(\gamma) = \frac{1}{n}\operatorname{trace}(A + U\operatorname{Diag}(\gamma)U^T) = \frac{1}{n}\left(\operatorname{trace}(A) + \sum_{i=1}^{t}\gamma_i\|u_i\|^2\right) = \frac{1}{n}\left(\operatorname{trace}(A) + \gamma^T\operatorname{norms}^2(U)\right),$$

and its gradient is $\nabla f(\gamma) = \frac{1}{n}\operatorname{norms}^2(U)$. On the other hand, by (3.21) $g(\gamma)$ can be expressed as

$$g(\gamma) = \det(A)^{1/n}\left(\prod_{i=1}^{t}(1+\gamma_i\|w_i\|^2)\right)^{1/n}.$$

The gradient of $g(\gamma)$ is then given component-wise by

$$\begin{aligned}\frac{\partial g(\gamma)}{\partial\gamma_j} &= \frac{1}{n}\det(A)^{1/n}\left(\prod_{i=1}^{t}(1+\gamma_i\|w_i\|^2)\right)^{(1-n)/n}\left(\prod_{i=1,i\neq j}^{t}(1+\gamma_i\|w_i\|^2)\right)\|w_j\|^2\\ &= \frac{1}{n}\det(A)^{1/n}\left(\prod_{i=1}^{t}(1+\gamma_i\|w_i\|^2)\right)^{1/n}(1+\gamma_j\|w_j\|^2)^{-1}\|w_j\|^2\\ &= \frac{g(\gamma)}{n}\frac{\|w_j\|^2}{1+\gamma_j\|w_j\|^2}, \qquad j\in[t].\end{aligned}$$

We make use of these expressions in order to compute the partial derivatives of ω. For every $j\in[t]$, we have

$$\begin{aligned}\frac{\partial\omega(\gamma)}{\partial\gamma_j} &= \frac{1}{g(\gamma)^2}\left[\frac{\partial f(\gamma)}{\partial\gamma_j}g(\gamma) - f(\gamma)\frac{\partial g(\gamma)}{\partial\gamma_j}\right]\\ &= \frac{1}{n^2 g(\gamma)}\left[n\|u_j\|^2 - \frac{\|w_j\|^2\left(\operatorname{tr}(A)+\gamma^T\operatorname{norms}^2(U)\right)}{1+\gamma_j\|w_j\|^2}\right].\end{aligned} \tag{3.23}$$

Since $n^2 g(\gamma) > 0$, the j-th partial derivative of ω is zero if, and only if,

$$n\|u_j\|^2 - \frac{\|w_j\|^2\left(\operatorname{tr}(A)+\gamma^T\operatorname{norms}^2(U)\right)}{1+\gamma_j\|w_j\|^2} = 0.$$

Therefore, the minimum of the pseudoconvex function is obtained as the solution of the linear system defined by the t equations

$$(n-1)\|u_k\|^2\gamma_k - \sum_{i=1,i\neq k}^{t}\|u_i\|^2\gamma_i = \operatorname{tr}(A) - n\frac{\|u_k\|^2}{\|w_k\|^2}, \quad k\in[t].$$

Equivalently,

$$\begin{bmatrix}(n-1)\|u_1\|^2 & -\|u_2\|^2 & \dots & & -\|u_t\|^2\\ -\|u_1\|^2 & (n-1)\|u_2\|^2 & -\|u_3\|^2 & \dots & -\|u_t\|^2\\ \dots & & & & \\ -\|u_1\|^2 & \dots & \dots & -\|u_{t-1}\|^2 & (n-1)\|u_t\|^2\end{bmatrix}\gamma = \begin{pmatrix}\operatorname{trace}(A) - n\|u_1\|^2/\|w_1\|^2\\ \dots\\ \operatorname{trace}(A) - n\|u_t\|^2/\|w_t\|^2\end{pmatrix}.$$

This is further equivalent to

$$\left[n\,\mathrm{Diag}(\mathrm{norms}^2(U)) - e\,\mathrm{norms}^2(U)^T\right]\gamma = \left(\mathrm{trace}(A)e - n\,\mathrm{Diag}\left(\mathrm{norms}^2(W)\right)^{-1}\mathrm{norms}^2(U)\right),$$

which is the system $K(U)\gamma = b(U)$ using the notation in (3.18).

Now we derive an explicit expression for the optimal γ. In order to do this, note that $K(U)$ is given as the sum of an invertible matrix, $n\,\mathrm{Diag}(\mathrm{norms}^2(U))$, and an outer product of vectors, $-e\,\mathrm{norms}^2(U)^T$. By the *Sherman-Morrison formula*, this sum is invertible if and only if

$$1 - \frac{1}{n}\,\mathrm{norms}^2(U)^T\,\mathrm{Diag}(\mathrm{norms}^2(U))^{-1}e \neq 0.$$

This is always true for $t < n$. Indeed, we have

$$1 - \frac{1}{n}\,\mathrm{norms}^2(U)^T\,\mathrm{Diag}(\mathrm{norms}^2(U))^{-1}e = 1 - \frac{1}{n}e^Te = 1 - \frac{t}{n} > 0.$$

Moreover, we obtain the following expression for the inverse

$$\begin{aligned}
&\left(n\ \mathrm{Diag}(\mathrm{norms}^2(U)) - e\,\mathrm{norms}^2(U)^T\right)^{-1} \\
&\quad= \frac{1}{n}\,\mathrm{Diag}(\mathrm{norms}^2(U))^{-1} + \frac{1}{\left(1-\frac{t}{n}\right)n^2}\,\mathrm{Diag}(\mathrm{norms}^2(U))^{-1}e\,\mathrm{norms}^2(U)^T\,\mathrm{Diag}(\mathrm{norms}^2(U))^{-1} \\
&\quad= \frac{1}{n}\,\mathrm{Diag}(\mathrm{norms}^{-2}(U)) + \frac{1}{(n-t)n}\,\mathrm{Diag}(\mathrm{norms}^{-2}(U))ee^T.
\end{aligned}$$

Therefore, the inverse of $K(U)$ in matrix form is given by

$$K(U)^{-1} = \frac{1}{n}\begin{bmatrix} \frac{1}{\|u_1\|^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\|u_2\|^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\|u_t\|^2} \end{bmatrix} + \frac{1}{(n-t)n}\begin{bmatrix} \frac{1}{\|u_1\|^2} & \frac{1}{\|u_1\|^2} & \dots & \frac{1}{\|u_1\|^2} \\ \frac{1}{\|u_2\|^2} & \frac{1}{\|u_2\|^2} & \dots & \frac{1}{\|u_2\|^2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\|u_t\|^2} & \frac{1}{\|u_t\|^2} & \dots & \frac{1}{\|u_t\|^2} \end{bmatrix}.$$

Finally, we obtain γ^* by calculating the product $\gamma^* = K(U)^{-1}b(U)$ which yields

$$\gamma_i^* = \frac{1}{(n-t)\|u_i\|^2}\left(\mathrm{tr}(A) - (n-t)\frac{\|u_i\|^2}{\|w_i\|^2} - \sum_{j=1}^{t}\frac{\|u_j\|^2}{\|w_j\|^2}\right), \tag{3.24}$$

for all $i \in [t]$. Since γ^* is the unique zero of the gradient of ω, we conclude that it belongs to Ω and solves (3.19). □

We note that the ω-optimal conditioning for the rank one update in Theorem 3.2.2 is obtained from (3.20) when $t = 1$. On the other hand, we can also employ the Cholesky decomposition of A to derive the ω-optimal conditioning in Theorem 3.2.5. We state this in the following corollary.

Corollary 3.2.6. *Given A and U as in Theorem 3.2.5. Let $A = LL^T$ be the Cholesky decomposition of A. Then, the formula for the ω-optimal conditioning γ^* in (3.20) holds with the replacement*

$$w_i \leftarrow L^{-1}u_i, \quad i \in [t].$$

Proof. The proof follows similarly to the one of Theorem 3.2.5 and thus is omitted. □

With the same assumptions as in Theorem 3.2.5, we now consider the problem of finding the ω-optimal conditioning in the box $[0,1]^t$, i.e.,

$$\bar{\gamma} = \arg \min_{\substack{\gamma \in [0,1]^t \\ A(\gamma) \succ 0}} \omega(\gamma). \tag{3.25}$$

For the rank one update ($t = 1$), Corollary 3.2.4 shows that the solution to (3.25) can be obtained by first computing the minimum of the unconstrained problem, whose explicit expression was given in Theorem 3.2.2, and then projecting onto the box constraint, which in that case was the interval $[0, 1]$. However, this simple projection can fail in general for the low rank update, as we now show in Example 3.2.8 below.

The illustration of this phenomenon will require considering a constrained pseudoconvex minimization problem. In the following Fact 3.2.7, see, e.g., [126, Chapter 10], we recall the sufficient optimality conditions for this class of optimization problems. We note that no constraint qualification is needed for *sufficiency*.

Fact 3.2.7 (Sufficient optimality conditions for pseudoconvex programming)**.** *Let $\Omega \subseteq \mathbb{R}^n$ be nonempty open and convex. Let $f : \Omega \to \mathbb{R}$ be a pseudoconvex function and $(g_i)_{i=1}^m : \Omega \to \mathbb{R}$ a family of differentiable and quasiconvex functions. Consider the optimization problem*

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_i(x) \leqslant 0, \quad i \in [m], \\ & x \in \Omega. \end{array} \tag{3.26}$$

Let $\bar{x} \in \Omega, \bar{\lambda} \in \mathbb{R}^m$, be a KKT primal-dual pair, i.e., the following KKT conditions hold:

$$\begin{aligned} \nabla f(\bar{x}) + \sum_{i=1}^m \bar{\lambda}_i \nabla g_i(\bar{x}) &= 0 \\ \bar{\lambda}_i &\geqslant 0, \quad i \in [m], \\ \bar{\lambda}_i g_i(\bar{x}) &= 0, \quad i \in [m], \\ \bar{x} \in \Omega \text{ and } g_i(\bar{x}) &\leqslant 0, \quad i \in [m]. \end{aligned} \tag{3.27}$$

Then $\bar{x}$ solves (3.26).

Example 3.2.8 (Failure of projection for constrained problem (3.25))**.** *Let $n = 3$, $t = 2$ and consider the following initial data for the ω-minimization problem:*

$$A := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \text{and} \quad U := \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{-1}{\sqrt{2}} & 0 \\ 0 & 1 \end{bmatrix}.$$

Then, we get the following:

- *From (3.24) and Theorem 3.2.5, the ω-optimal preconditioner is $\gamma^* = \frac{1}{3}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$;*

- *projecting onto* $[0,1]^2$ *yields* $\gamma_p^* = \frac{1}{3}\begin{pmatrix}1\\0\end{pmatrix}$, *where* $\omega(\gamma_p^*) = 16/(9\sqrt[3]{5})$;

- *however, with* $\bar{\gamma} := \frac{1}{2}\begin{pmatrix}1\\0\end{pmatrix}$, *we get a lower value:*

$$\omega(\bar{\gamma}) = 1/\left(3\sqrt[3]{(2/11)^2}\right) \approx 1.0386 < 1.0397 \approx 16/(9\sqrt[3]{5});$$

and $\bar{\gamma}$ *is the* ω*-optimal preconditioner in* $[0,1]^2$, *as we now show.*

To prove the last statement, note that (3.25) *can be written as the pseudoconvex program in* (3.26) *by setting* $f := \omega : \mathbb{R}^2 \to \mathbb{R}$, $g_1(\gamma) = -\gamma_1$, $g_2(\gamma) = -\gamma_2$, $g_3(\gamma) = \gamma_1 - 1$, $g_4(\gamma) = \gamma_2 - 1$ *and* Ω *defined as in* (3.22). *In particular, the only active constraint for* $\bar{\gamma} = (1/2, 0)^T$ *is* $g_2(\gamma) = 0$, *so the KKT conditions become*

$$0 = \frac{\partial \omega(\bar{\gamma})}{\partial \gamma_1},$$
$$0 = \frac{\partial \omega(\bar{\gamma})}{\partial \gamma_2} - \bar{\lambda}_2,$$

for some $\bar{\lambda}_2 \geqslant 0$. *This can be verified by simply substituting using the expressions of the partial derivatives of* ω *obtained in* (3.23). *By Fact 3.2.7, we conclude that for the given data,* $\bar{\gamma}$ *is the solution of* (3.25).

As done in the previous example, obtaining the ω-optimal preconditioner in the box $[0,1]^t$ would require obtaining a KKT point for the constrained pseudoconvex problem (3.25). This is not an easy task. To the author's knowledge, closed formulas for this kind of box constrained minimization problems are not known even when the objective is a quadratic. Nevertheless, using the projection of γ^* onto $[0,1]^t$ as an approximation to $\bar{\gamma}$ appears to give good results in practice. We see this in our numerical tests in Section 3.2.4.

Finally, observe that the computation of γ^* in formula (3.20) might be as expensive as finding the Newton direction without preconditioning, since it requires the spectral or Cholesky decomposition. However, under the framework of Example 3.2.1, we now see in Proposition 3.2.9 that we can get the following inexpensive and effective approximation $\gamma = \gamma^*_{\text{apr}}$, given by the expression

$$\left[\gamma^*_{\text{apr}}\right]_i := \frac{\operatorname{tr}(A)}{(n-t)\|u_i\|^2}, \quad \forall i \in [t]. \tag{3.28}$$

Proposition 3.2.9. *Let* $A \in \mathbb{S}^n_+, B \in \mathbb{R}^{n\times m}, U \in \mathbb{R}^{n\times t}$, *with* $t := |\mathcal{I}_0|$, *be defined as in Example 3.2.1. For simplicity define the columns of the matrix* $\bar{B} := \left[B_i\right]_{i\in\mathcal{I}_+}$ *with* $r := \operatorname{rank}(\bar{B}) < n$. *Let* $j \in \mathcal{I}_0$ *and let* u_j, w_j *be defined as in Theorem 3.2.5, and let* $\epsilon > 0$ *be the Levenberg–Marquardt regularization parameter in* (3.10). *Then,*

$$u_j \notin \operatorname{range}(\bar{B}) \implies \frac{\|u_j\|^2}{\|w_j\|^2} \leqslant \epsilon \frac{\|u_j\|^2}{\operatorname{dist}(u_j, \operatorname{range}(\bar{B}))^2}.$$

We conclude that (3.28) *provides an efficient estimate of* γ^* *in formula* (3.20).

Proof. Let $x \in \operatorname{range}(\bar{B})$, $y \in \operatorname{null}(\bar{B}^T)$ such that $u_j = x + y$. Then,

$$\begin{aligned}
\|w_j\|^2 &= u_j^T Q D^{-1} Q^T u_j \\
&= (x+y)^T Q D^{-1} Q^T (x+y) \\
&= x^T Q D^{-1} Q^T x + 2x^T Q D^{-1} Q^T y + y^T Q D^{-1} Q^T y \\
&= \sum_{\ell=1}^{r} \frac{1}{\lambda_\ell + \epsilon} (q_\ell^T x)^2 + 0 + \sum_{\ell=r+1}^{n} \frac{1}{\epsilon} (q_\ell^T y)^2 \\
&\geqslant \sum_{\ell=r+1}^{n} \frac{1}{\epsilon} (q_\ell^T y)^2 \\
&= \frac{1}{\epsilon} \sum_{\ell=r+1}^{n} (q_\ell^T y)^2 = \frac{1}{\epsilon} \sum_{\ell=1}^{n} (q_\ell^T y)^2 = \frac{1}{\epsilon} \|y\|^2,
\end{aligned}$$

where q_ℓ is ℓ-th column of Q. Since $\|y\| = \operatorname{dist}(u_j, \operatorname{range}(\bar{B}))$,

$$\frac{\|u_j\|^2}{\|w_j\|^2} \leqslant \epsilon \frac{\|u_j\|^2}{\operatorname{dist}(u_j, \operatorname{range}(\bar{B}))^2}.$$

□

Remark 3.2.10. *We note that the approximate ω-optimal conditioning obtained in (3.28) is strictly related to a popular choice of γ appearing in the literature (see, e.g., [35]), namely, taking*

$$\gamma_i := \frac{1}{\|u_i\|^2}, \quad \forall i \in [t]. \tag{3.29}$$

Indeed, the updates resulting from (3.28) and (3.29) only differ in a scaling given by $\operatorname{trace}(A)/(n-t)$. We will see in Proposition 4.2.1 that the selection of (3.29) corresponds to the ω-optimal diagonal preconditioner of the matrix U, i.e., it aims to minimize the ω-condition number of only the update term. In contrast, our proposed approach aims to minimize ω for the whole matrix $A(\gamma) = A + U \operatorname{Diag}(\gamma) U^T$. Numerical comparisons between both approaches are presented in the numerics in Table 3.5 and Figure 3.6, below.

3.2.4 Numerical Tests

We now present tests with different choices of γ for efficient iterative solutions of linear systems of the form $\boxed{A(\gamma)x = b}$, where $A(\gamma)$ is given in (3.30). We use MATLAB's builtin preconditioned conjugate gradient function **pcg**. We focus our attention on the case where $A(\gamma) \in \mathbb{S}^n_{++}$ is a low rank update that appears in choosing subgradients in nonsmooth Newton methods, see Example 3.2.1. Our aim is to improve conditioning to improve convergence, thus we call this γ-*conditioning*.

Problem Generation and Definitions of γ

Specifically, we generate random instances as follows:

- Define

$$A(\gamma) := A + \epsilon I + U \operatorname{Diag}(\gamma) U^T; \tag{3.30}$$

- ϵ is a random number in the interval $[10^{-7}, 10^{-9}]$;
- $A = A_0^T A_0$ with $A_0 \in \mathbb{R}^{r\times n}$ a normally distributed random sparse matrix with density at most $0.5/\log(n)$; $r \in [n/2+1, n-1]$ is a random integer;
- $t \in [2, r/2]$ is the randomly chosen rank of the update, $U \in \mathbb{R}^{n\times t}$ is a normally distributed random sparse matrix of density at most $1/\log(n)$;
- The right hand side, b, is chosen as the sum of two random vectors in the range of A and U, respectively. More precisely,
$$b = A\, b^1 + U\, b^2,$$
with $b^1 \in \mathbb{R}^n$ and $b^2 \in \mathbb{R}^t$ vectors randomly generated using the standard normal distribution.

As explained in Example 3.2.1, in this application the γ for conditioning is required to belong to the hypercube $[0,1]^t$. Therefore, in our experiments we test the performance of four different choices of γ-conditioning:

- $(\gamma = 0)$: the zero vector ;
- $(\gamma = e)$: the vector of ones;
- $(\gamma = u^{-2})$: projection onto $[0,1]^t$ of the ω-optimal diagonal preconditioner for the last term, $U\,\mathrm{Diag}(\gamma)U^T$, of (3.30). Recall from (3.29) (also, Proposition 4.2.1) that γ is given by
$$\gamma_i = \min\{1, 1/\|u_i\|^2\},\ i \in [t],$$
where u_i denotes the ith column of U;
- $(\gamma = \gamma_p^*)$ projection of γ^*, obtained in Theorem 3.2.5, onto $[0,1]^t$;
- $(\gamma = \gamma_{\mathrm{apr}}^*)$: projection of the approximated ω-optimal obtained in (3.28), onto $[0,1]^t$.

Descriptions of Parameters and Outputs

For each dimension $n \in \{1000, 2000, 3000, 5000\}$, we generate 10 instances of random problems and solve the corresponding systems with MATLAB's **pcg** and with the five different choices of γ-conditioning.

Table 3.5 shows the average over the 10 instances of: κ- and ω-condition numbers of every $A(\gamma)$; relative residual; number of iterations; time used by **pcg** for every choice of γ; and time for computing each (nontrivial) γ. Both the spectral and Cholesky decompositions are implemented for computing $\gamma = \gamma_p^*$ but the table only contains the time for more efficient approach. We indicated the corresponding approach in the last column as well. We stop if a tolerance of 10^{-12} is reached or the maximum $50,000$ iterations is exceeded. We use the origin as our initial starting point.

We also use performance profiles, e.g., [60], to compare the different choices of γ. Let P denote the set of problems, and now set $\Gamma := \{0, e, u^{-2}, \gamma_p^*, \gamma_{\mathrm{apr}}^*\}$ as the set of γ conditioners.

$$\begin{aligned} t_{p,\gamma} = &\{\text{time for computing the preconditioner}\} \\ &+ \{\text{time for solving the preconditioned problem by } \mathbf{pcg}\}. \end{aligned}$$

n	γ	$\kappa(A(\gamma))$	$\omega(A(\gamma))$	Rel.Res	Iter	T. Total	T. Solve	T. γ^*
1000	0	2.3249e+09	1.0173e+04	1.5146e-01	2734.30	0.0218	0.0218	-
	e	2.2193e+11	4.7689e+03	1.7750e-06	4187.40	0.8181	0.8181	-
	u^{-2}	8.5539e+09	2.2114e+03	3.8172e-07	2730.10	0.0810	0.0807	0.0003
	γ_p^*	1.0722e+10	2.2095e+03	1.8169e-07	2788.10	0.0880	0.0748	0.0132 (C)
	γ_{apr}^*	1.0432e+10	2.2095e+03	1.5855e-07	2821.80	0.0757	0.0752	0.0005
2000	0	2.0729e+12	2.2127e+04	1.8829e-07	230.70	0.5887	0.5887	-
	e	3.4235e+12	6.5292e+02	9.2055e-13	425.50	1.4527	1.4527	-
	u^{-2}	1.8491e+12	8.9700e+02	9.1536e-13	1364.40	0.7545	0.7541	0.0003
	γ_p^*	2.0726e+12	5.6538e+02	9.1902e-13	376.20	0.6021	0.2319	0.3702 (S)
	γ_{apr}^*	2.0644e+12	5.6538e+02	9.1737e-13	376.30	0.2391	0.2368	0.0023
3000	0	4.4961e+12	1.9718e+04	7.4824e-08	586.20	3.3928	3.3928	-
	e	3.6078e+12	3.9795e+03	9.3498e-13	261.70	1.5414	1.5414	-
	u^{-2}	3.6699e+12	4.3747e+03	9.1465e-13	917.80	1.0956	1.0954	0.0002
	γ_p^*	3.6544e+12	3.9795e+03	9.4326e-13	261.60	1.6990	0.3219	1.3770 (S)
	γ_{apr}^*	3.5688e+12	3.9795e+03	9.4326e-13	261.60	0.3247	0.3199	0.0048
5000	0	1.0028e+13	3.0421e+04	2.6256e-07	698.80	11.5600	11.5600	-
	e	1.1709e+13	8.9242e+02	9.3629e-13	362.90	5.9142	5.9142	-
	u^{-2}	8.2778e+12	1.4249e+03	1.2583e-09	1563.00	6.3804	6.3783	0.0021
	γ_p^*	8.7440e+12	8.2755e+02	9.6946e-13	344.30	8.3604	1.4008	6.9596 (S)
	γ_{apr}^*	8.8089e+12	8.2755e+02	9.5456e-13	344.30	1.4175	1.4026	0.0149

Table 3.5: For different dimensions n, every choice of γ for updating, average of 10 instances: κ- and ω-condition numbers of $A(\gamma)$; residual; number of iterations; total time (in seconds); solve time; time for computing γ^*, (S) stands for spectral and (C) for Cholesky decomposition.

Then, for every problem $p \in P$ and every $\gamma \in \Gamma$, we define the performance ratio as

$$r_{p,\gamma} := \begin{cases} \frac{t_{p,\gamma}}{\min\{t_{p,\gamma} : \gamma \in \Gamma\}} & \text{if convergence test passed,} \\ +\infty & \text{if convergence test failed.} \end{cases}$$

In our experiments, a convergence test *passed* if it succeeded in solving the linear system with the required relative residual tolerance in less than $50,000$ iterations, and otherwise it *failed*. Note that the best performing preconditioner with respect to the measure under study (time or number of iterations), say $\tilde{\gamma}$, for problem p will have performance ratio $r_{p,\tilde{\gamma}} = 1$. In contrast, if the preconditioner γ underperforms in comparison with $\tilde{\gamma}$, but still manages to pass the test, then

$$r_{p,\gamma} = \frac{t_{p,\gamma}}{t_{p,\tilde{\gamma}}} > 1$$

is the ratio between the overall time (resp., number of iterations) required for solving the problem p for this particular choice and the time (resp., number of iterations) employed by $\tilde{\gamma}$. Consequently, the larger the value of $r_{p,\gamma}$, the worse the preconditioner γ performed for problem p.

Finally, the performance profile of $\gamma \in \Gamma$ is defined as

$$\rho_\gamma(\tau) := \frac{1}{|P|} \text{size}\left\{p \in P \,:\, r_{p,\gamma} \leqslant \tau\right\},$$

where $|P|$ is the number of problems in P. This can be understood as the relative portion of times that the performance ratio $r_{p,\gamma}$ is within a factor of $\tau \geqslant 1$ of the best possible performance ratio. In particular, $\rho_\gamma(1)$ represents the number of problems where γ is the best choice. Also, the existence of a $\tau \geqslant 1$ such that $\rho_\gamma(\tau) = 1$, indicates that γ passed the convergence test for every single problem in P. In Figure 3.6, we display our performance profiles, with $\log_2$ scale on τ.

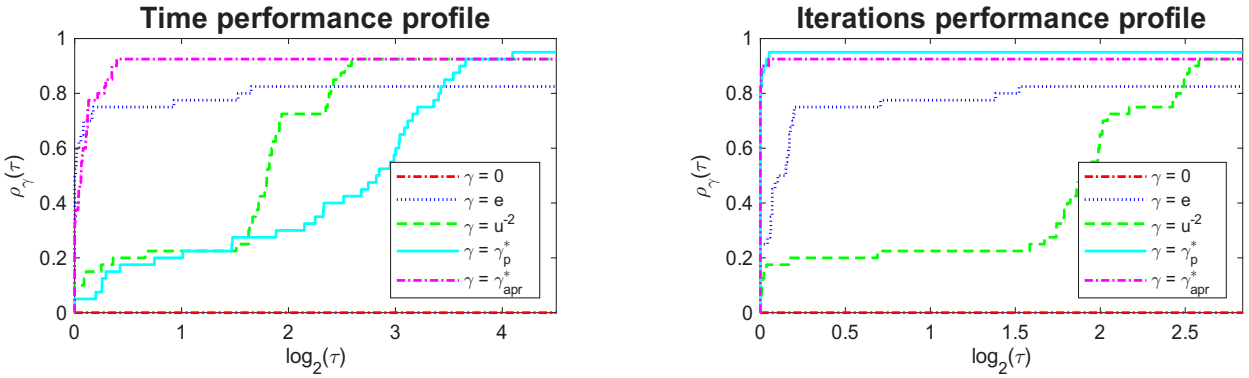

Figure 3.6: Time, iterations performance profiles; for system $A(\gamma)x = b$ with different choices of γ in Section 3.2.4; using MATLAB's **pcg**.

Summary of Empirics

Firstly, we observe that **pcg** with no γ-conditioning (γ=0) fails to achieve the desired residual, so the problems under consideration are sufficiently ill-conditioned. The performance profiles reveal that, in more than 90% of the tested instances, the ω-optimal conditioning leads to a problem

that can be solved with the least number of iterations. However, the computation of the optimal conditioning γ_p^* is too time expensive (see Table 3.5) which does not make it advantageous in general[6]. Nonetheless, we observe that the approximated ω-optimal conditioning γ_{apr}^* maintains the benefits of γ_p^* in terms of both solve time and number of iterations. In addition, it can be computed very efficiently which makes it the best option among all γ-conditionings.

[6]Regarding the different approaches for computing $\gamma = \gamma_p^*$, we want to mention that although obtaining the Cholesky decomposition $A = LL^T$ is in general less costly than computing its eigenvalue decomposition, the computation of the ω-optimal conditioning in this case requires solving the system $LW = U$, see Corollary 3.2.6. This means that, for larger dimensions, employing the spectral decomposition for computing γ^* seems to be more time efficient.

Chapter 4

Diagonal Preconditioners

Preconditioning is essential in iterative and direct solutions of large-scale linear systems e.g., [38,67, 70,71,76,79,149]. All of these works focus on studying the classical κ-condition number of a matrix A. On the other hand, another line of research has focused on studying the ω-condition number. Moreover, it is known that the simple scaling diagonal preconditioner using the norms of the columns of A is the optimal diagonal preconditioner with respect to ω and is efficient in practice, see [52,147], i.e., ω validates the use of this specific diagonal preconditioner.[1] Although κ-optimal preconditioners minimize the worst-case condition number, our results show that ω-optimal preconditioners are cheaper to compute and are more predictive of PCG/LSQR performance. We emphasize that ω provides a more average indication rather than a worst case measure of conditioning.

For simplicity, we restrict our comparisons to diagonal and block diagonal preconditioning for large scale sparse positive definite and general invertible linear systems. The positive definite case arises from many diverse applications including: finite element analysis, sparse regression, Newton type optimization algorithms, and also from e.g., in [71, 74]: (i) the so-called normal equations from interior point methods in solving linear programs and (ii) the Hessians in minimizing logistic regression.

Recall that the κ- and ω-condition numbers for the positive definite case $M \succ 0$ are, respectively, given by ratios of eigenvalues

$$\kappa(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}; \quad \omega(M) = \frac{\text{trace}(M)/n}{\det(M)^{1/n}} = \frac{\sum_i \lambda_i(M)/n}{\prod_i (\lambda_i(M))^{1/n}}. \tag{4.1}$$

Preconditioning with D uses

$$Ax = b \iff D(A(D(D^{-1}x)) = Db,$$

and reduces a condition number of DAD. For our purposes we exploit the fact that the following functions have the same values for nonsingular D:

$$\kappa(DAD) = \frac{\lambda_{\max}(DAD)}{\lambda_{\min}(DAD)} = \frac{\lambda_{\max}(ADD)}{\lambda_{\min}(ADD)}; \qquad \omega(DAD) = \frac{\text{trace}(DAD)/n}{\det(DAD)^{1/n}} = \frac{\text{trace}(ADD)/n}{\det(ADD)^{1/n}}. \tag{4.2}$$

[1] In [71] the motivation for numerically finding diagonal κ-optimal preconditioners was the lack of theoretical validation. See also the near optimality results in [173]. Validation using ω is now provided in Proposition 4.2.1.

Our algorithms exploit the functions with argument affine in $\bar{D} = DD$ rather than quadratic in D.

Main Contributions

The main contributions of this chapter consist of both theoretical and numerical analyses of the κ and ω-condition numbers. First, we provide an affine based pseudoconvex reformulation of the κ-optimal diagonal preconditioning problem that allows for an elegant characterization of its optimality conditions. There are three advantages of this reformulation: all its stationary points are global minima, its subgradients are inexpensive to compute, and its optimization variable is just an $n \times 1$ vector rather than an $n \times n$ matrix as in *semidefinite programming, SDP* [149]. Our extensive numerical experiments show that subgradient methods based on our reformulation can effectively converge to a κ-optimal diagonal preconditioner more efficiently in time and accuracy than current SDP-based approaches in the literature [71, 149].

Second, we provide and exploit the ability to find explicit formulae for ω-optimal diagonal and block diagonal preconditioners. We demonstrate that many popular classic preconditioners, such as the Jacobi preconditioner and row/column normalization preconditioners are ω-optimal preconditioners. We also display that matrix balancing schemes reduce ω at every iteration and converge to a stationary point of the two-sided ω-optimal diagonal preconditioning problem. To the best of our knowledge, this is the first time that such a comprehensive characterization of optimality conditions for both condition numbers is provided.

Finally, we conduct additional extensive numerical experiments that compare the efficiency of the κ- and ω-condition numbers. Our results show that since ω-optimal preconditioners typically have closed-form solutions, they are much cheaper to construct than κ-optimal preconditioners. Our results further demonstrate that PCG iterations and LSQR iterations for solving linear systems are, on average, more correlated with the ω-condition number than the κ-condition number.

We now continue with the organization of this chapter and more details on the main contributions. In Section 4.1, we present several formulations for minimizing κ along with the derivatives and optimality conditions. The formulations are presented to take advantage of the affine approach and then avoid the positive homogeneity of the problem in order to improve stability of the problem. In particular, we include Theorem 4.1.7 that presents a characterization of a κ-optimally diagonally preconditioned A that is based on the *largest/smallest* eigenpairs. This leads to an efficient subgradient algorithm and we include convergence results. In Section 4.2, we characterize ω-optimal preconditioners with special structures. A characterization for a ω-optimal diagonal preconditioner is simple as it corresponds to the classical Jacobi preconditioner for symmetric positive definite matrices and the row/column normalization preconditioner for general nonsingular matrices. We also show that the matrix balancing algorithm, Sinkhorn-Knopp, linearly converges to a stationary point of the two-sided ω-optimal diagonal preconditioning problem. Finally, we include results on extending from diagonal to block diagonal preconditioning in Section 4.2.3. We show that the right-sided ω-optimal block diagonal preconditioner for full column rank matrices A comes from taking Q-less QR decompositions of the blocks of A. We also give a characterization for the left-sided ω-optimal block diagonal preconditioner and show that when A is square that it is related to applying the right-sided optimal preconditioner to A^T.

Extensive numerical tests are then conducted in Section 4.3. We demonstrate that our subgradient methods are more scalable and efficient at computing κ-optimal diagonal preconditioners than existing SDP-based approaches in the literature [71, 149]. Moreover, the resulting preconditioners

yield more substantial improvements in PCG performance for solving linear systems compared to those produced by [71, 149].

We further compare optimal diagonal preconditioning strategies based on the condition numbers κ and ω. In particular, we observe that the Sinkhorn-Knopp algorithm, which converges to a stationary point of the two-sided ω-optimal preconditioning problem, produces preconditioners that significantly outperform the two-sided κ-optimal method of [149] in terms of LSQR iteration counts for minimizing $\|b-Ax\|$. While Sinkhorn-Knopp dramatically reduces ω, the method of [149] achieves a stronger reduction in κ. This is notable as despite this difference, Sinkhorn-Knopp yields better LSQR performance on the majority of instances while also being computationally cheaper.

Finally, we demonstrate that applying ω-optimal diagonal scaling to positive definite linear systems that are already κ-optimally scaled leads to significant improvements in PCG performance. Overall, our results highlight two key advantages of ω-optimal preconditioners: they are inexpensive to compute and substantially reduce the iteration counts of iterative methods.

4.1 κ-Optimal Diagonal Preconditioning

Preconditioning (or scaling) plays a fundamental role in numerical linear algebra and optimization; see, for example, the surveys [22, 41, 176]. In this section, we study the problem of computing a **κ-optimal diagonal preconditioner** for positive definite linear systems

$$Mx = b, \quad M = A^T A, \quad A \in \mathcal{M}^n \text{ nonsingular.}$$

Our objective is to find a diagonal matrix $D \succ 0$ that minimizes the classical condition number $\kappa(D^{1/2}MD^{1/2})$.

Rather than viewing this as a nonlinear problem in $D^{1/2}$, we exploit the eigenvalue invarience in (4.2) to reformulate to the simpler affine mapping MD and still minimize $\kappa(D^{1/2}MD^{1/2})$. The reformulation yields a pseudoconvex minimization problem with several advantages:

- all stationary points are global minimizers,
- subgradients are easy to compute,
- the composite functions are used to avoid homogeneity and design very efficient and scalable algorithms.

Note that [71] also considers minimizing $\kappa(D^{1/2}MD^{1/2})$, although they aim to find a diagonal preconditioner that is a convex combination of a small basis or list of diagonal preconditioners.

4.1.1 Optimality Conditions

Let $M \in \mathbb{S}^n_{++}$. We denote the quadratic type scaling as follows:

$$\mathcal{D}_{\mathrm{q}} : \mathbb{R}^n_{++} \to \mathbb{S}^n, \quad \mathcal{D}_{\mathrm{q}}(d) = \operatorname{Diag}(d)^{1/2} M \operatorname{Diag}(d)^{1/2}.$$

By abuse of notation, we let the argument determine the function,

$$\kappa(d) = \kappa(\mathcal{D}_{\mathrm{q}}(d)) = (\kappa \circ \mathcal{D}_{\mathrm{q}})(d).$$

Similarly,

$$\lambda_{\max}(d) = \lambda_{\max}(\mathcal{D}_{\mathrm{q}}(d)) = (\lambda_{\max} \circ \mathcal{D}_{\mathrm{q}})(d), \quad \lambda_{\min}(d) = \lambda_{\min}(\mathcal{D}_{\mathrm{q}}(d)) = (\lambda_{\min} \circ \mathcal{D}_{\mathrm{q}})(d).$$

In the literature, e.g., [70, 71, 149], κ-optimal diagonal preconditioning refers to solving

$$d^* \in \operatorname{argmin} \left\{ \kappa(\mathcal{D}_{\mathrm{q}}(d)) = \frac{\lambda_{\max}(\mathcal{D}_{\mathrm{q}}(d))}{\lambda_{\min}(\mathcal{D}_{\mathrm{q}}(d))} : d \in \mathbb{R}^n_{++} \right\}. \tag{4.3}$$

Here we restrict to $d \in \mathbb{R}^n_{++}$ as we are taking square roots, see also Remark 4.1.4, below.

We show below, that for our purposes, we can use the equivalent *linear in* d transformation

$$\mathcal{D} : \mathbb{R}^n_{++} \to \mathbb{S}^n, \ \mathcal{D}(d) = M \operatorname{Diag}(d). \tag{4.4}$$

With $D = \operatorname{Diag}(d)$, we first note the relationships in the eigenpairs of $\mathcal{D}_{\mathrm{q}}(d) = D^{1/2} M D^{1/2}$ and the nonsymmetric but similar $\mathcal{D}(d) = MD$, and $\mathcal{D}(d)^T = DM$; as well we have the convexity and concavity of the maximum and minimum eigenvalues of these nonsymmetric matrices MD.[2]

Lemma 4.1.1. *Let $M, D = \operatorname{Diag}(d) \in \mathbb{S}^n_{++}$, and $(u_i, \lambda_i), i = 1, \ldots, n$, be orthogonal eigenpairs of $\mathcal{D}_{\mathrm{q}}(d)$, i.e., $\langle u_i, u_j \rangle = 0, \forall i \neq j$. Define*

$$x_i = D^{-1/2} u_i, \forall i, \ X = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}, Y = X^{-T}, \ \Lambda = \operatorname{Diag}(\lambda).$$

Then X and Y are matrices of right and left eigenvectors of $\mathcal{D}(d)$, respectively, with corresponding matrix of eigenvalues Λ; and Y is given by $Y = DX(X^T DX)^{-1}$.
Moreover, $\lambda_{\max}(d) = \lambda_1(d)$ (respectively, $\lambda_{\min}(d) = \lambda_n(d)$) is a convex (respectively, concave) function of $d \in \mathbb{R}^n_{++}$.

Proof. Let $U := [u_1 \ \ldots \ u_n]$ be the corresponding matrix with *orthogonal*, not necessarily orthonormal, eigenvectors. We have $X = D^{-1/2} U$ and thus $U^T U = X^T D X$. This brings us to

$$U^{-T} = U(X^T D X)^{-1}.$$

Notice that $X^T D X$ is a diagonal matrix with its diagonal entries given by $\|u_i\|^2$, which are strictly positive as eigenvectors are nonzero. Thus the inverse is well-defined. Now, observe that

$$\begin{aligned} Y &= X^{-T} = D^{1/2} U^{-T} \\ &= D^{1/2} U (X^T D X)^{-1} \\ &= D X (X^T D X)^{-1} \end{aligned}$$

as desired. Now,

$$\begin{aligned} D^{1/2} M D^{1/2} U = U\Lambda \quad &\Longrightarrow \quad MDD^{-1/2}U = D^{-1/2}U\Lambda \\ &\Longrightarrow \quad MDX = X\Lambda \\ &\Longrightarrow \quad X^T DM = \Lambda X^T \\ &\Longrightarrow \quad DMX^{-T} = X^{-T}\Lambda \\ &\Longrightarrow \quad DMY = Y\Lambda. \end{aligned}$$

[2] These are a special case of so-called K-pd matrices in the literature, i.e., a product of two symmetric matrices where at least one is positive definite. The product of two positive definite matrices $P = AB$ arises in the optimality conditions in semidefinite programming. One of the difficulties is that the symmetrization of P is *not* necessarily positive definite, see e.g., [170].

The second and last equation show that X and Y are right, and left, eigenvectors of MD, respectively. Moreover, we note that $M \succ 0$ has a positive definite square root and $\lambda_i(M \operatorname{Diag}(d)) = \lambda_i(M^{1/2} \operatorname{Diag}(d) M^{1/2})$. Therefore,

$$\max \lambda_i(\mathcal{D}(d)) = \max \lambda_i(M^{1/2} \operatorname{Diag}(d) M^{1/2}) = \max_{\|x\|=1} x^T M^{1/2} \operatorname{Diag}(d) M^{1/2} x.$$

As in the proof of Lemma 2.1.1, the latter is a maximum of linear functions in d and therefore is convex. The concavity result follows similarly. □

Corollary 4.1.2. *Let $M \in \mathbb{S}^n_{++}, d \in \mathbb{R}^n_{++}$. Then*

$$\lambda_{\max}(d) = \lambda_{\max}(\mathcal{D}_{\mathrm{q}}(d)) = \lambda_{\max}(\mathcal{D}(d)), \quad \lambda_{\min}(d) = \lambda_{\min}(\mathcal{D}_{\mathrm{q}}(d)) = \lambda_{\min}(\mathcal{D}(d));$$

and

$$\kappa(d) = \frac{\lambda_{\max}(\mathcal{D}_{\mathrm{q}}(d))}{\lambda_{\min}(\mathcal{D}_{\mathrm{q}}(d))} = \frac{\lambda_{\max}(\mathcal{D}(d))}{\lambda_{\min}(\mathcal{D}(d))},$$

where recall $\mathcal{D}(d)$ is as in (4.4).

Proof. The proof follows immediately from Lemma 4.1.1. □

This allows us to consider the equivalent simplified view of finding a κ-optimal diagonal preconditioner, i.e., we need to solve the fractional pseudoconvex[3] minimization problem

$$\bar{d} \in \operatorname{argmin} \left\{ \kappa(d) = \frac{\lambda_{\max}(\mathcal{D}(d))}{\lambda_{\min}(\mathcal{D}(d))} : d \in \mathbb{R}^n_{++} \right\}. \tag{4.5}$$

The following Lemma 4.1.3 shows that for $M \succ 0$, the matrix MD having all positive eigenvalues is equivalent to the positive definiteness of D. Combining this observation with Lemma 4.1.1, we obtain that $\kappa(d)$ is the ratio of a convex function and a positive concave function on its domain, and is therefore pseudoconvex.

Lemma 4.1.3. *Let $M \in \mathbb{S}^n_{++}, d \in \mathbb{R}^n, D = \operatorname{Diag}(d)$. Then*

$$\lambda_i(MD) > 0, \forall i \iff d \in \mathbb{R}^n_{++}.$$

Proof. We note that the eigenvalues of MD are the same as the eigenvalues of $M^{1/2} D M^{1/2}$. Sylvester's Lemma of inertia implies that the number of negative eigenvalues of $M^{1/2} D M^{1/2}$ is the same as that of D and so it is the same as the number of negative elements in d. □

Remark 4.1.4. *Note that the two equivalent problems (4.3) and (4.5) are both essentially unconstrained, and the optima are attained and characterized as stationary points in $\mathbb{R}^n_{++}$. This is not obvious but follows since we have the ratios of convex and concave functions using $\lambda_{\max}, \lambda_{\min}$ and this is over the open cone constraint $d \in \mathbb{R}^n_{++}$. If we add the constraints $d^T e_n = n, d > 0$, or equivalently that $d = e_n + Vv > 0$, with $V \in \mathbb{R}^{n \times (n-1)}$ a matrix whose columns contain a basis for $e_n^\perp$ as done above, then we have a bounded problem in $v \in \mathbb{R}^{n-1}$ and the spectral radius $\lambda_{\max}(MD) \leqslant \|M\| \|D\|$*

[3] A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is said to be pseudoconvex if for all $x, y \in X$, $\nabla f(x) \cdot (y - x) \geqslant 0 \implies f(y) \geqslant f(x)$. A key property of pseudoconvex functions is that every stationary point is a global minimizer.

is bounded above. Bounded below away from 0 *follows from applying the greedy solution to the knapsack problem with constraints* $\sum_i d_i = n, d \geqslant 0$. *We get*

$$\lambda_{\max}(d) \quad \geqslant \quad \operatorname{trace}(MD)/n = \textstyle\sum_i M_{ii} d_i/n \geqslant \min_i M_{ii} > 0.$$

Therefore, with the denominator going to 0, *we have* κ *going to* ∞ *as* $d = e_n + Vv$ *approaches the boundary of the simplex. That is, minimizing* κ *provides a* self-barrier function *for this boundary.*

Moreover, $\alpha > 0 \implies \kappa(d) = \kappa(\alpha d)$, *i.e., we have positive homogeneity of degree zero. Therefore, the opimal* d *is* not *unique. By pseudoconvexity, the optimal set is a convex set. This set can be large, see Proposition 4.1.8.*

Characterization of κ-Optimal Scaling

Using eigenvalue derivative formulas in Lemmas 4.1.5 and 4.1.6, we compute the gradient of $\kappa(d)$ in the smooth setting. When the largest and smallest eigenvalues of $\mathcal{D}(d)$ are simple, the gradient takes an explicit form in terms of the corresponding eigenvectors. The main result Theorem 4.1.7 provides a characterization for κ-optimality.

Lemma 4.1.5 ([172, (3.1)])**.** *Let* $B : \mathbb{R} \to \mathcal{M}^n$ *be differentiable with derivative* $\dot{B} = \dot{B}(t)$, *and, at* $t = \bar{t}$, *let* $B(\bar{t})$ *be diagonalizable with real eigenpairs (ignoring the argument* $\bar{t}$*)*

$$BX = X\Lambda,\ B^T Y = Y\Lambda, \quad \Lambda = \operatorname{Diag}(\lambda).$$

And make the choice $Y = X^{-T}$. *Let* $1 \leqslant k \leqslant n$ *and* λ_k *be a singleton eigenvalue with right and left eigenvector* x_k, y_k, *respectively, taken from the corresponding columns of* X, Y, *respectively. Then the derivative of eigenvalues is given by*

$$\dot{\lambda}_k(\bar{t}) = y_k^T(\bar{t})\dot{B}(\bar{t})x_k(\bar{t}) = \operatorname{trace} \dot{B} x_k y_k^T.^{4} \tag{4.6}$$

Proof. The proof is in [172, Pg 303].[5] □

Lemma 4.1.6. *The Fréchet derivative of the linear transformation* $\mathcal{D}$ *at* d *acting on* Δd *is (simply)*

$$\mathcal{D}'(d)(\Delta d) = M \operatorname{Diag}(\Delta d).$$

Now, let λ *be a singleton eigenvalue of* $\mathcal{D}(d)$ *with a right eigenvector* x. *Then the gradient of the composite function at* d *is*

$$\nabla\lambda(\mathcal{D}(d)) = \frac{\lambda}{x^T D x} x \circ x.$$

Proof. The derivative of the linear transformation is clear.

Suppose as above that we have a linear transformation $\mathcal{D}(d)$ with singleton eigenvalue λ. Let $y = Dx/x^T Dx$ be the corresponding left eigenvector given in Lemma 4.1.1. Then the derivative of

[4] If Y is an invertible matrix of right eigenvectors not chosen using X^{-T}, then the normalization scaling needs to be added explicitly as $\dot{\lambda}_k(\bar{t}) = y_k^T(\bar{t})\dot{B}(\bar{t})x_k(\bar{t})/(y_k(\bar{t})^T x(\bar{t}))$.

[5] The derivative of eigenvectors is included in the reference along with a useful normalization that allows for stability of the evaluations of the eigenvectors.

the composite function at d acting on Δd is

$$\begin{aligned}\langle \nabla(\lambda \circ \mathcal{D})(d), \Delta d\rangle &= \lambda'\left(\mathcal{D}'(d)(\Delta d)\right) = y^T\left(\mathcal{D}'(d)(\Delta d)\right)x \\ &= \tfrac{1}{x^T D x} x^T \left(DM\,\mathrm{Diag}(\Delta d)\right)x \\ &= \tfrac{\lambda}{x^T D x} x^T \mathrm{Diag}(\Delta d)x = \tfrac{\lambda}{x^T D x}\langle x \circ x, \Delta d\rangle.\end{aligned}$$

□

Theorem 4.1.7 shows that at optimal preconditioning, the extreme eigenvectors must "spread mass equally" across coordinates. This balance condition is what characterizes optimal diagonal preconditioners.

Theorem 4.1.7 (Characterization of κ-optimal diagonal preconditioning)**.** *Consider $M \in \mathbb{S}^n_{++}, d \in \mathbb{R}^n_{++}$ be given. Let $D = \mathrm{Diag}(d)$ and let (λ_1, x_1) (resp. (λ_n, x_n)) be a singleton maximum (resp. minimum) eigenpair of $\mathcal{D}(d)$. Then the gradient of the composite function $\kappa(d)$ is*

$$\nabla\kappa(d) = \kappa(d)\left(\frac{1}{x_1^T D x_1}(x_1 \circ x_1) - \frac{1}{x_n^T D x_n}(x_n \circ x_n)\right),$$

where recall that $\kappa(d)$ is as in Corollary 4.1.2 . Hence, M is κ-optimally diagonally preconditioned if, and only if, the orthonormal eigenvector pair satisfies

$$x_1 \circ x_1 = x_n \circ x_n. \tag{4.7}$$

Equivalently, after a permutation of the elements to account for the sign,

$$x_1 = \begin{pmatrix} u \\ v \end{pmatrix},\ x_n = \begin{pmatrix} u \\ -v \end{pmatrix},\ \|u\| = \|v\|. \tag{4.8}$$

In the nonsmooth case, we can choose the normalized x_1, respectively x_n, in the eigenspace of $\lambda_{\max}(\mathcal{D}(d))$, respectively $\lambda_{\min}(\mathcal{D}(d))$.

Proof. From Lemma 4.1.6, we can find the gradient as follows:

$$\begin{aligned}\nabla\kappa(d) &= \tfrac{1}{\lambda_n^2}\left(\lambda_n\dot{\lambda}_1 - \lambda_1\dot{\lambda}_n\right) = \tfrac{1}{\lambda_n^2}\left(\tfrac{\lambda_n\lambda_1}{x_1^T D x_1}x_1 \circ x_1 - \tfrac{\lambda_1\lambda_n}{x_n^T D x_n}x_n \circ x_n\right) \\ &= \kappa(d)\left((x_1 \circ x_1)/(x_1^T D x_1) - (x_n \circ x_n)/(x_n^T D x_n)\right).\end{aligned}$$

The characterization for κ-optimally diagonally preconditioned matrix A follows from solving $\nabla\kappa(e_n) = 0$. □

Because $\kappa(d)$ is positively homogeneous of degree zero, the minimizer is not unique. Proposition 4.1.8 shows that this nonuniqueness is structural rather than pathological, i.e., when the eigenspace for the largest and smallest eigenvalues is unique of dimension 2, the orthogonal complement is unique. When the extreme eigenvectors do not uniquely determine the eigenspace of the orthogonal complement, entire continua of optimal diagonal scalings exist. This observation motivates the reformulation in Section 4.1.3, where homogeneity is removed.

Proposition 4.1.8 (Nonuniqueness of κ-optimal diagonal preconditioning)**.** *Let $M \succ 0$ be a κ-optimal diagonal preconditioned matrix as given in Theorem 4.1.7 with $x_i, \lambda_i, i = 1, n$, being two,*

singleton, eigenpairs that satisfy the optimality conditions in Theorem 4.1.7. Thus we have

$$\lambda_1 > \lambda_2 \geqslant \ldots \geqslant \lambda_{n-1} > \lambda_n > 0, \quad \lambda = (\lambda_i) \in \mathbb{R}^n_{++}, \, \Lambda = \mathrm{Diag}(\lambda),$$

and we let $Q = \begin{bmatrix} x_1 & \bar{Q} & x_n \end{bmatrix}$ *be an orthogonal matrix and* $M = Q\Lambda Q^T$ *from the spectral theorem. Let* V *be the basis for* $e_n^\perp$ *defined in (1.1). Suppose in addition that the* lack *of strict complementarity condition holds, i.e., there exists* v *such that*

$$0 \neq v \in \{u \in \mathbb{R}^n_- : 0 = Vu \circ x_1 = Vu \circ x_n\}. \tag{4.9}$$

Then with $\Delta D = \mathrm{Diag}(Vv)$*, there exists* $\epsilon > 0$ *such that*

$$\kappa(M) = \kappa\left((I + t\Delta D)M(I + t\Delta D)\right), \, \forall |t| \leqslant \epsilon.$$

Proof. The result follows from expanding

$$(I + \epsilon\Delta D)M(I + \epsilon\Delta D) = M + O(\epsilon)$$

and using the continuity of eigenvalues and the fact that the optimality conditions imply

$$\begin{array}{rcl} Vv \circ x_i = 0, i = 1, n & \iff & \mathrm{Diag}(Vv)x_i = 0, i = 1, n \\ & \iff & \Delta D x_i = 0, i = 1, n. \end{array}$$

Specifically, let

$$M_2 = \bar{Q}\,\mathrm{Diag}((\lambda_2, \ldots, \lambda_{n-1})^T)\bar{Q}^T, \quad M_1 = M - M_2.$$

The above equation then implies $\Delta D M_1 = M_1 \Delta D = 0$. Moreover, let $D = (I + \epsilon\Delta D)$, then

$$DMD = M_1 + DM_2D = M + \epsilon\Delta D M_2 + \epsilon M_2 \Delta D + \epsilon^2 \Delta D M_2 \Delta D.$$

Since $\mathrm{range}(\Delta D) \subseteq \mathrm{null}(M_1) = \mathrm{range}(M_2)$, the range of the perturbation of M above is restricted to the eigenspace of M_2, i.e., to the $\mathrm{span}(\{x_2, x_3, \ldots, x_{n-1}\})$. This means that after the perturbation, with $\epsilon > 0$ sufficiently small, λ_1 and λ_n remain the largest and smallest eigenvalues of DMD respectively. As stated above, we are using the continuity of eigenvalues and the orthogonality $M_1 M_2 = 0$ that arises using the spectral theorem and $M_1, M_2 \succeq 0$. □

Note that if condition (4.9) holds, then we get a nonsingleton set in $\mathbb{R}^n_-$ of solutions. Moreover, the structure of V and support of v implies that (4.9) restricts the support of the eigenspace $\mathrm{span}(\{x_1, x_n\})$ and we can permute to get the support in the last entries.[6]

4.1.2 Projected Subgradient Method

In Theorem 4.1.7, we derived the gradient and optimality conditions for minimizing the pseudoconvex function $\kappa(d)$ in (4.5), over the open set $d > 0$. Convergence of subgradient methods for pseudoconvex minimization typically requires optimizing over a closed set. Hence, we consider the following optimization problem:

[6] Necessity of (4.9) for nonuniqueness is still an open problem.

$$\bar{d} \in \operatorname{argmin}\left\{\kappa(d) = \frac{\lambda_{\max}(\mathcal{D}(d))}{\lambda_{\min}(\mathcal{D}(d))} : d \in \Omega := \{d : d \geqslant \delta e_n\}\right\}, \quad 0 < \delta \ll 1. \tag{4.10}$$

We now exploit the pseudoconvex structure that guarantees all stationary points are global minimizers making subgradient methods natural. The search directions used in Algorithm 4.1 are shown to lie in the quasisubdifferential of κ, allowing us to directly invoke classical convergence results for quasiconvex minimization. We first treat the unconstrained formulation in Algorithm 4.1 and address homogeneity in Section 4.1.3.

Algorithm 4.1 A Subgradient Method for Minimizing $\kappa(d)$ over Ω

Inputs: symmetric positive definite matrix $M \succ 0$; sequence of positive stepsizes $\{t_k\} \to 0$ with $\sum_{k=1}^{\infty} t_k = \infty$, scalar $\delta \in (0,1)$; tolerance tol; rule for the stopping criterion, stopcrit.

set $k \leftarrow 0$;

set stopcrit $\leftarrow \infty$;

set $d_1 = e_n \in \mathbb{R}^n$;

1: **while** stopcrit>tol **do** (main outer loop)

2: set $k \leftarrow k + 1$;

3: compute min eigenpair (λ_n^k, x_n^k) and max eigenpair (λ_1^k, x_1^k) of $M\,\mathrm{Diag}(d_k)$;

4: compute direction

$$s_k = \frac{\lambda_1^k}{\lambda_n^k}\left(\frac{1}{\langle x_1^k, d_k \circ x_1^k\rangle}\left(x_1^k \circ x_1^k\right) - \frac{1}{\langle x_n^k, d_k \circ x_n^k\rangle}\left(x_n^k \circ x_n^k\right)\right); \tag{4.11}$$

5: perform projected gradient step

$$d_{k+1} = \max\left\{d_k - t_k \frac{s_k}{\|s_k\|}, \delta e_n\right\}; \tag{4.12}$$

6: update stopcrit;

7: **end while**(main outer loop)

Output: $\hat{D} := \mathrm{Diag}(d_{k+1})$.

Remark 4.1.9. *For Algorithm 4.1, we use a popular choice for stepsize sequence $\{t_k\}$ in subgradient methods, $t_k = 1/k$, where k is the iteration index [109]. We note that since Algorithm 4.1 is a subgradient method, in general it is not a descent method. Finally, the algorithm is general in that it does not specify the stopping rule stopcrit. There are several possible rules that the user can employ in practice for updating stopcrit in step 6. For example, at every iteration the user can take stopcrit as $\|s_k\|$ or $|\kappa(d_{k+1}) - \kappa(d_k)|$.*

We now briefly describe each of the steps of Algorithm 4.1. First, step 3 computes a maximal and minimal eigenpair of $M\,\mathrm{Diag}(d_k)$ to construct the search direction s_k. It should be noted that the user never has to form the matrix $M\,\mathrm{Diag}(d_k)$ to compute the eigenpairs. Instead, the user only has to construct subroutines which compute $M\,\mathrm{Diag}(d_k) * x$ and $(M\,\mathrm{Diag}(d_k))\backslash x$ efficiently, where $x \in \mathbb{R}^n$. Second, it will be shown in Section 4.1.2 that the direction s_k computed in step 4 lies in the quasisubdifferential (also to be defined in Section 4.1.2) of $\kappa(d_k)$. Finally, using this direction, step 5 performs the projected gradient update $d_{k+1} = \Pi_\Omega(d_k - t_k \frac{s_k}{\|s_k\|})$, where $\Pi_\Omega(x) := \operatorname{argmin}_{y\in\Omega} \|x - y\|$ denotes the projection onto Ω.

Asymptotic Convergence Analysis

Our convergence analysis of Algorithm 4.1 relies mostly on [109] where efficient subgradient methods for minimizing quasiconvex functions are presented. Under various assumptions in addition to quasiconvexity, the author establishes asymptotic convergence of subgradient methods with decaying stepsizes. Since our function $\kappa(d)$ is pseudoconvex, it is quasiconvex. For a more detailed discussion related to the assumptions of Kiwiel [109] and how our set-up satisfies these assumptions, the reader is referred to Section 2.2. In the remaining part of this subsection, we show that Algorithm 4.1 asymptotically converges by showing that it is an instance of the subgradient framework proposed by Kiwiel.

To minimize a quasiconvex function f over a closed convex set X, Kiwiel proposes in [109] the following basic subgradient algorithm:

$$x_{k+1} := \Pi_X(x_k - t_k \hat{g}_k), \quad \hat{g}_k := g_k/\|g_k\|, \quad g_k \in \bar{\partial}^\circ f(x_k), \quad k = 1, 2, \ldots, \quad x_1 \in X,$$

where $\bar{\partial}^\circ f$ is the quasisubdifferential of f defined in (1.3) and $t_k > 0$ are the stepsizes.

Hence, we need to characterize vectors that lie in the quasisubdifferential of $\kappa(d)$ to show that Algorithm 4.1 is an instance of Kiwiel's subgradient framework. The result below presents one characterization of vectors that lie in the quasisubdifferential of fractional programs like the one in our set-up (4.10).

Lemma 4.1.10. *Suppose $f(x) := a(x)/b(x)$ is well-defined for all $x \in X$ where $a(x)$ is a convex function that is positive on X, $b(x)$ is a concave function that is positive on X. If for $x \in X$, $B := 1/b(x)$ and $A := a(x)/b^2(x)$, then*

$$B[\partial a(x)] + A[\partial(-b)(x)] \subseteq \bar{\partial}^\circ f(x).$$

Proof. Let $x \in X$ and consider B and A as in the assumptions of the lemma. It follows from the fact that a and $-b$ are convex functions, B and A are positive scalars, and Fenchel subdifferential calculus rules that

$$B[\partial a(x)] + A[\partial(-b)(x)] = \partial(Ba)(x) + \partial(-Ab)(x) \subseteq \partial(Ba - Ab)(x).$$

Now, let $g \in B[\partial a(x)] + A[\partial(-b)(x)] \subseteq \partial(Ba - Ab)(x)$, and suppose $y \in \bar{S}(x)$, i.e., y is in the interior of the domain of f and $f(y) < f(x)$. It then follows that

$$\begin{aligned} \langle g, y - x\rangle &\leqslant Ba(y) - Ab(y) - Ba(x) + Ab(x) \\ &= \frac{a(y)}{b(x)} - \frac{a(x)b(y)}{b^2(x)} < 0, \end{aligned}$$

where the first inequality follows from the definition of Fenchel subdifferential, the equality follows from the definitions of A and B, and the last inequality follows from the fact that $a(y)/b(y) < a(x)/b(x)$ and $b(y)$ and $b(x)$ are positive. It then follows from the definition of quasisubdifferential in (1.3) that $g \in \bar{\partial}^\circ f(x)$. □

Corollary 4.1.11 constructs a vector that lies in the quasisubdifferential of $\kappa(d)$.

Corollary 4.1.11. *Let $d \in \Omega$ and let (λ_1, x_1) and (λ_n, x_n) be a maximal and minimal eigenpair of $\mathcal{D}(d)$, respectively. Then, it holds that*

$$\frac{\lambda_1}{\lambda_n}\left(\frac{1}{x_1^T(d \circ x_1)}(x_1 \circ x_1) - \frac{1}{x_n^T(d \circ x_n)}(x_n \circ x_n)\right) \in \bar{\partial}^\circ\left(\kappa(d)\right).$$

Proof. It follows from Lemma 4.1.1 that $\lambda_{\max}(\mathcal{D}(d))$ and $\lambda_{\min}(\mathcal{D}(d))$ are convex and concave functions of d, respectively. Also, it is easy to see that $\lambda_{\max}(\mathcal{D}(d))$ and $\lambda_{\min}(\mathcal{D}(d))$ are positive on Ω. Hence, it follows from Lemma 4.1.10 that

$$\frac{1}{\lambda_{\min}(\mathcal{D}(d))}\partial\lambda_{\max}(\mathcal{D}(d)) + \frac{\lambda_{\max}(\mathcal{D}(d))}{\lambda_{\min}^2(\mathcal{D}(d))}\partial\left(-\lambda_{\min}(\mathcal{D}(d))\right) \subseteq \bar{\partial}^\circ\left(\kappa(d)\right) \tag{4.13}$$

holds for $d \in \Omega$. Moreover, it follows from Lemma 4.1.6, the definition of Clarke subdifferential in (1.2), and the fact that the Fenchel and Clarke subdifferentials coincide for convex functions that the following inclusions hold.

$$\frac{\lambda_1}{x_1^T\left(d \circ x_1\right)}\left(x_1 \circ x_1\right) \in \partial\lambda_{\max}(\mathcal{D}(d)), \quad -\frac{\lambda_n}{x_n^T\left(d \circ x_n\right)}\left(x_n \circ x_n\right) \in \partial\left(-\lambda_{\min}(\mathcal{D}(d))\right), \tag{4.14}$$

where here (λ_1, x_1) and (λ_n, x_n) are maximal and minimal eigenpairs of $\mathcal{D}(d)$, respectively. The result then follows from (4.13) and (4.14).

□

Remark 4.1.12. *It follows from Corollary 4.1.11 that the update rule (4.12) in step 5 is of the form*

$$d_{k+1} = \Pi_\Omega\left(d_k - t_k\frac{s_k}{\|s_k\|}\right), \text{ where } s_k \in \bar{\partial}^\circ\left(\kappa(d_k)\right).$$

We are now ready to present the main theorem that shows Algorithm 4.1 converges asymptotically.

Theorem 4.1.13 (Asymptotic convergence)**.** *Let $\{d_k\}$ be generated by Algorithm 4.1. Let*

$$\kappa_* := \min\left\{\kappa(d) : d \in \Omega\right\}; \quad \kappa_*^k = \min\{\kappa(d_j), j = 1\ldots, k\}.$$

Then

$$\underline{\lim}_{k\to\infty}\kappa(d_k) = \kappa_*; \quad \text{and } \kappa_*^k \downarrow \kappa_*.$$

Proof. It follows from the last remark in Lemma 4.1.1 and the definitions of $\kappa(d)$ and Ω in (4.10), that $\lambda_{\max}(\mathcal{D}(d))$ (resp. $\lambda_{\min}(\mathcal{D}(d))$) is a convex (resp. concave) function that is positive on Ω. It then follows from this observation, Lemma 2.2.2, and the definition of $\kappa(d)$, that assumptions **A1-A4** in Section 2.2 hold for the minimization problem in (4.10). Clearly, also assumption **A5** in Section 2.2 also holds since Ω in (4.10) is a closed set and the intersection of Ω with the interior of the domain of $\kappa(d)$ is clearly nonempty. The result of the theorem then immediately follows from this observation, Remark 4.1.12, the facts that $t_k \to 0$, and $\sum_{k=1}^\infty t_k = \infty$, and Theorem 1 in [109]. □

4.1.3 Avoiding Positive Homogeneity

As mentioned in Remark 4.1.4, $\kappa(d)$ is a positively homogenous function (of degree zero). Thus there are multiple optimal solutions and our problem is *Hadamard ill-posed*. From a computational stability perspective, it is more efficient to minimize an equivalent smaller dimensional formulation that is not positively homogeneous. With this in mind, we let $M \succ 0$ and and $e_n \in \mathbb{R}^n$ be the vector of all ones. We consider the function

$$\mathring{\mathcal{V}}(v) := M \operatorname{Diag}(e_n + Vv), \tag{4.15}$$

where $v \in \mathbb{R}^{n-1}$ and V is as in (1.1). In this subsection, we consider the following formulation

$$\min \left\{ \kappa(v) := \frac{\lambda_{\max}(\mathring{\mathcal{V}}(v))}{\lambda_{\min}(\mathring{\mathcal{V}}(v))} : e_n + Vv \geqslant \hat{\delta} e_n,\ v \in \mathbb{R}^n_- \right\}, \tag{4.16}$$

where $\hat{\delta} \in (0,1)$ is sufficiently small. It is easy to see that with the choice of V in (1.1), the above (4.16) can be rewritten as

$$\min \left\{ \kappa(v) : v \in \hat{\Omega} \right\}, \tag{4.17}$$

where

$$\hat{\Omega} := \left\{ v \in \mathbb{R}^{n-1} : \sum_{i=1}^{n-1} v_i \leqslant -\sqrt{2}(\hat{\delta} - 1), \quad v_i \geqslant \sqrt{2}(\hat{\delta} - 1), \quad i = 1, \dots n-1 \right\}. \tag{4.18}$$

Note that the set $\hat{\Omega}$ is convex and compact. Projecting onto it is also easy since it is just a simplex. Also, since $\hat{\Omega}$ is bounded, we will be able to get nonasymptotic convergence guarantees for the subgradient method that we propose to minimize (4.17).

The following lemma will be useful for developing our subgradient method. It gives useful characterizations of the derivatives of $\mathring{\mathcal{V}}(v)$ and $\kappa(v)$ in the smooth setting when the maximum and minimum eigenvalues of $\mathring{\mathcal{V}}(v)$ have multiplicity one.

Lemma 4.1.14 (Derivatives of $\mathring{\mathcal{V}}(v), \kappa(v)$)**.** *Let $M \succ 0, v \in \mathbb{R}^n_-$ be given and set $w = e_n + Vv \in \mathbb{R}^n_{++}$ and $D = \operatorname{Diag}(w)$. Also, let (λ_1, x_1) and (λ_n, x_n) be maximal and minimal eigenpairs of $\mathring{\mathcal{V}}(v)$, respectively, where λ_1 and λ_n have multiplicity one and x_1 and x_n are assumed to be normalized. Then the following hold:*

1. *The derivative at v acting on $\Delta v \in \mathbb{R}^n_-$ is*

$$\dot{\mathring{\mathcal{V}}}(v)(\Delta v) := \mathring{\mathcal{V}}'(v)(\Delta v) = M \operatorname{Diag}(V \Delta v).$$

2. *The gradient of the composite function $\kappa(v) := \kappa(\mathring{\mathcal{V}}(v))$ is*

$$\nabla \kappa(v) = \kappa(v) V^T \left(\frac{1}{x_1^T (w \circ x_1)} (x_1 \circ x_1) - \frac{1}{x_n^T (w \circ x_n)} (x_n \circ x_n) \right). \tag{4.19}$$

Proof. 1. The proof follows from the function being affine.

2. For a singleton eigenvalue λ with normalized right eigenvector x, we have the left eigenvector $y = Dx/(x^T Dx)$ as in Lemma 4.1.1, which satisfies $x^T y = 1$. Then

$$\begin{aligned}\langle \nabla(\lambda \circ \mathring{\mathcal{V}})(v), \Delta v\rangle &= y^T \dot{\mathring{\mathcal{V}}}(v)(\Delta v)x = y^T(M\,\mathrm{Diag}(V\Delta v))x \\ &= \langle V^T \,\mathrm{diag}(\lambda D^{-1} y x^T), \Delta v\rangle \\ &= \tfrac{1}{x^T Dx}\langle V^T \,\mathrm{diag}(\lambda D^{-1} Dx x^T), \Delta v\rangle \\ &= \tfrac{1}{x^T(w\circ x)}\langle V^T \,\mathrm{diag}(\lambda x x^T), \Delta v\rangle = \tfrac{1}{x^T(w\circ x)}\langle \lambda V^T(x\circ x), \Delta v\rangle.\end{aligned}$$

Therefore the gradient of $\kappa(v) := \kappa(\mathring{\mathcal{V}}(v))$ is:

$$\begin{aligned}\nabla\kappa(v) &= \tfrac{\lambda_n\dot{\lambda}_1 - \lambda_1\dot{\lambda}_n}{\lambda_n^2} = \tfrac{1}{\lambda_n^2}\left(\tfrac{\lambda_n}{x_1^T(w\circ x_1)}\lambda_1 V^T(x_1\circ x_1) - \tfrac{\lambda_1}{x_n^T(w\circ x_n)}\lambda_n V^T(x_n\circ x_n)\right) \\ &= \kappa(v)V^T\left(\tfrac{1}{x_1^T(w\circ x_1)}(x_1\circ x_1) - \tfrac{1}{x_n^T(w\circ x_n)}(x_n\circ x_n)\right).\end{aligned} \tag{4.20}$$

□

The following remark shows that, as in Lemma 4.1.14, $\nabla\kappa(v)$ lies in the quasisubdifferential of $\kappa(v)$.

Remark 4.1.15. *Let $M \succ 0$ and $v \in \mathbb{R}^{n-1}$. Also, suppose that $w = e_n + Vv$ and let (x_i, λ_i), $i = 1, n$, be eigenpairs of $\mathring{\mathcal{V}}(v)$, where $\|x_i\| = 1$. It then follows from Lemma 4.1.10 and a similar argument as in the proof of Corollary 4.1.11 that*

$$\kappa(v)V^T\left(\frac{1}{x_1^T(w\circ x_1)}(x_1\circ x_1) - \frac{1}{x_n^T(w\circ x_n)}(x_n\circ x_n)\right) \in \bar{\partial}^\circ\left(\kappa(v)\right) \tag{4.21}$$

where recall that $\kappa(v) := \kappa(\mathring{\mathcal{V}}(v))$.

We now present our subgradient algorithm for minimizing (4.17).

Algorithm 4.2 A Subgradient Method for Minimizing $\kappa(v) := \kappa(\mathring{\mathcal{V}}(v))$ over $\hat{\Omega}$

Inputs: symmetric positive definite matrix $M \succ 0$; $V \in \mathbb{R}^{n\times(n-1)}$ a basis matrix for the orthogonal complement $e^{\perp}$; sequence of stepsizes $\{t_k\} = 1/\sqrt{k}$; scalar $\hat{\delta} \in (0,1)$; a tolerance tol> 0; a rule for the stopping criterion, stopcrit.

set $k \leftarrow 0$;

set stopcrit $\leftarrow \infty$;

set $v_1 = 0 \in \mathbb{R}^{n-1}$ and $w_1 = e_n \in \mathbb{R}^n_+$;

while stopcrit>tol **do** (main outer loop)

set $k \leftarrow k+1$.

compute min eigenpair (λ_n^k, x_n^k) and max eigenpair (λ_1^k, x_1^k) of $M\,\mathrm{Diag}(w_k)$;

compute direction

$$g_k = \frac{\lambda_1^k}{\lambda_n^k} V^T \left(\frac{1}{\langle x_1^k, w_k \circ x_1^k \rangle} \left(x_1^k \circ x_1^k \right) - \frac{1}{\langle x_n^k, w_k \circ x_n^k \rangle} \left(x_n^k \circ x_n^k \right) \right) \tag{4.22}$$

perform projected gradient step

$$v_{k+1} = \Pi_{\hat{\Omega}} \left(v_k - t_k \frac{g_k}{\|g_k\|} \right) \tag{4.23}$$

where $\hat{\Omega}$ is as in (4.18) and set

$$w_{k+1} = e + V v_{k+1}; \tag{4.24}$$

update stopcrit;

end while(main outer loop)

Output: $\hat{D} := \mathrm{Diag}(w_{k+1})$.

Several remarks about Algorithm 4.2 are now given. First, the matrix V does not need to be stored in memory and is actually not needed as input. All the user needs to input is a subroutine that outputs Vv given a vector $v \in \mathbb{R}^{n-1}$. Likewise, the matrix $M\,\mathrm{Diag}(w_k)$ does not need to be stored. Second, it follows from Remark 4.1.15 that $g_k \in \bar{\partial}^{\circ}(\kappa(v_k))$, which is defined in (1.3). Finally, the projected gradient step in (4.23) can be performed very efficiently since projecting onto $\hat{\Omega}$ just involves computing the root of a simple equation.

Nonasymptotic Convergence Rate Analysis for Minimizing $\kappa(v)$ on $\hat{\Omega}$

In this subsection, we show a nonasymptotic convergence rate for Algorithm 4.2 for minimizing $\kappa(v)$ over $\hat{\Omega}$. More specifically, we show that

$$\min_{1\leqslant k\leqslant K} \kappa(v_k) - \kappa_* \leqslant \epsilon$$

holds for $K = \mathcal{O}(1/\epsilon^2)$, where $\kappa_* = \min_{v\in\hat{\Omega}} \kappa(v)$. Our nonasymptotic convergence analysis of Algorithm 4.2 relies mostly on the results from [109] and [99]. The function $\kappa(v)$ is pseudoconvex since it is the ratio of a convex function and a positive concave function. Also, observe that the set $\hat{\Omega}$ is just a box so it is a convex compact set that is easy to project onto. The only other assumption

that needs to be verified to be able to show a nonasymptotic convergence rate of Algorithm 4.2 is that the function $\kappa(v)$ is Lipschitz continuous on $\hat{\Omega}$. This result is proved in the following proposition.

Proposition 4.1.16. *The function $\kappa(v)$ is Lipschitz continuous on $\hat{\Omega}$, i.e.,*

$$\|\kappa(v_1) - \kappa(v_2)\| \leqslant L\|v_1 - v_2\|, \quad \forall v_1, v_2 \in \hat{\Omega}$$

where $L > 0$ and $\hat{\Omega}$ is as in (4.18).

Proof. First, it is easy to see that $\lambda_{\max}(\mathring{\mathcal{V}}(v))$ (resp. $\lambda_{\min}(\mathring{\mathcal{V}}(v))$) is a convex (resp. concave) function. It then follows from this observation, the fact $\hat{\Omega}$ is a compact set that is contained in the domain of both functions, and Proposition A.48(b) in [116] that $\lambda_{\max}(\mathring{\mathcal{V}}(v))$ (resp. $\lambda_{\min}(\mathring{\mathcal{V}}(v))$) is L_1-Lipschitz (resp. L_2-Lipschitz) on $\hat{\Omega}$. Consider now the composite function $h(v) = g(\lambda_{\min}(\mathring{\mathcal{V}}(v)))$ where $g(y) = 1/y$. Note that by the choice of $\hat{\Omega}$, $\lambda_{\min}(\mathring{\mathcal{V}}(v)) > 0$ for all $v \in \hat{\Omega}$. The above conclusion and the fact that $g(y) = 1/y$ is Lipchitz continuous on any positive open interval then imply that h is L_3-Lipschitz continuous on $\hat{\Omega}$, where $L_3 > 0$.

We now show that $\kappa(v)$ is Lipschitz continuous. First, it holds that $|\lambda_{\max}(v)| \leqslant M_1$ and $|h(v)| \leqslant M_2$ for $v \in \hat{\Omega}$ since a Lipschitz function on a compact set is bounded. Then, for any $v_1 \in \hat{\Omega}$ and $v_2 \in \hat{\Omega}$, the following holds:

$$\begin{aligned}
\|\kappa(\mathring{\mathcal{V}}(v_1)) - \kappa(\mathring{\mathcal{V}}(v_2))\| &= \|\lambda_{\max}(\mathring{\mathcal{V}}(v_1))h(v_1) - \lambda_{\max}(\mathring{\mathcal{V}}(v_2))h(v_2)\| \\
&= \|\lambda_{\max}(\mathring{\mathcal{V}}(v_1))h(v_1) - \lambda_{\max}(\mathring{\mathcal{V}}(v_1))h(v_2) + \lambda_{\max}(\mathring{\mathcal{V}}(v_1))h(v_2) - \lambda_{\max}(v_2)h(v_2)\| \\
&\leqslant M_1\|h(v_1) - h(v_2)\| + M_2\|\lambda_{\max}(\mathring{\mathcal{V}}(v_1)) - \lambda_{\max}(\mathring{\mathcal{V}}(v_2))\| \\
&\leqslant M_1L_3\|v_1 - v_2\| + M_2L_1\|v_1 - v_2\| = (M_1L_3 + M_2L_1)\|v_1 - v_2\|,
\end{aligned}$$

which immediately implies the statement of the proposition with $L = M_1L_3 + M_2L_1$.

□

We now state Theorem 4.1.17, which displays that Algorithm 4.2 is able to find an ϵ-approximate optimal solution of (4.17) with a sublinear $\mathcal{O}(1/\epsilon^2)$ rate of convergence.

Theorem 4.1.17. *Let $\epsilon > 0$ be a given tolerance and suppose that $K = \mathcal{O}(1/\epsilon^2)$. It then holds that*

$$\min_{1 \leqslant k \leqslant K} \kappa(v_k) - \kappa_* \leqslant \epsilon, \tag{4.25}$$

where $\kappa_* = \min_{v \in \hat{\Omega}} \kappa(v)$ and $\kappa(v) := \kappa(\mathring{\mathcal{V}}(v))$.

Proof. It is immediate to see that $\hat{\Omega}$ is a convex compact set and that $\kappa(v)$ is a quasiconvex function since it is the ratio of a convex and a positive concave function. It follows from Proposition 4.1.16 that $\kappa(v)$ is Lipschitz continuous on $\hat{\Omega}$. Finally, it follows from Remark 4.1.15 that the update (4.23) in step 5 of Algorithm 4.2 is of the form $v_{k+1} = \Pi_{\hat{\Omega}}\left(v_k - t_k \frac{g_k}{\|g_k\|}\right)$ where $g_k \in \bar{\partial}^\circ(\kappa(v_k))$. Hence, it follows from these observations, the fact that $\{t_k\} = 1/\sqrt{k}$, and Theorem 3.2(ii) in [99] with $s = 1/2$, $\delta = \epsilon$, and $p = 1$ that the statement of Theorem 4.1.17 holds.

□

An Efficient Line-Search Subgradient Method

This section presents an efficient line-search subgradient method for minimizing $\kappa(v) := \kappa(\mathring{\mathcal{V}}(v))$ where $\mathring{\mathcal{V}}(v)$ is as in (4.15).

Let $w = e + Vv \in \mathbb{R}^n_{++}$ correspond to the current iterate v. Also, let σ be a small scalar between 0 and 1 and let $\Delta w = V\Delta v$ where $\Delta v = -\nabla\kappa(v)$ where $\kappa(v)$ is as in (4.16). From Lemma 4.1.3, we can use a ratio test and guarantee positive definiteness from $w + tV\Delta v = w + t\Delta w > \sigma w$, or equivalently $t\Delta w > (\sigma - 1)w$. Therefore, we conclude that the maximum step with safeguarding, so as not to get too close to the boundary, is

$$t_{\max}(\sigma) \cong t_{\max} := \min\left\{\frac{(\sigma-1)w_i}{\Delta w_i} : \Delta w_i < 0\right\} > 0. \tag{4.26}$$

Recall the gradient $\nabla\kappa(v) = \nabla(\kappa \circ \mathring{\mathcal{V}}(v))$ in Lemma 4.1.14. Our line search in Algorithm 4.3 maintains:

LineSearch 4.1.18 (for Algorithm 4.3)**.** *Backtrack and maintain: (i) $t \leqslant t_{\max}$ from (4.26); (ii) monotonic nonincrease of the objective $\kappa(v + t\Delta v)$; and nonpositivity of the directional derivative $\nabla\kappa(v + t\Delta v)^T\Delta v \leqslant 0$.*

Algorithm 4.3 is presented below.

Algorithm 4.3 An Efficient Line-Search Subgradient Method for Minimizing $\kappa(v)$.

Inputs: A symmetric positive definite matrix $A \succ 0$, a max iteration count mainmaxiter, stopping tolerances maintoler> 0 and linesrchtoler> 0, and a small scalar $0 < \sigma \ll 1$.
set $k \leftarrow 0$, $v_1 = 0 \in \mathbb{R}^{n-1}$, $w_1 = e_n \in \mathbb{R}^n$, and $t_k \leftarrow \infty$;
compute a min eigenpair (λ_n^1, x_n^1) and max eigenpair (λ_1^1, x_1^1) of A;
compute $\kappa(v_1)$ and $\nabla\kappa(v_1)$ according to (4.16) and (4.19), respectively;
set relnormg$= \|\nabla\kappa(v_1)\|^2/(1 + \kappa(v_1)^2)$;
while relnormg $>$ maintoler & $t_k >$ linesrchtoler & $k \leqslant$ mainmaxiter **do** (main outer loop)
  set $k \leftarrow k + 1$;
  set $\Delta v_k \leftarrow -\nabla\kappa(v_k)$;
  use (4.26) with $\Delta w := V\Delta v_k$ and $w := w_k$ to evaluate $t_{\max}(\sigma)$;
  perform LineSearch 4.1.18 to find t_k;
  set $v_{k+1} = v_k + t_k\Delta v_k$ and $w_{k+1} = e_n + Vv_{k+1}$;
  compute min eigenpair $(\lambda_n^{k+1}, x_n^{k+1})$ and max eigenpair $(\lambda_1^{k+1}, x_1^{k+1})$ of $A\,\mathrm{Diag}(w_{k+1})$;
  compute $\kappa(v_{k+1})$ and $\nabla\kappa(v_{k+1})$ according to (4.16) and (4.19), respectively;
  update relnormg$= \|\nabla\kappa(v_{k+1})\|^2/(1 + \kappa(v_{k+1})^2)$;
end while(main outer loop)
Output: $\hat{D} := \mathrm{Diag}(w_{k+1})$.

We emphasize that LineSearch 4.1.18 does not perform an exact line search and is intended only as a heuristic. In practice, we set linesearchtol to 10^{-7} and maintoler to 10^{-4}. Numerical results are presented in Section 4.3.1.

4.2 ω-Optimal Structured Preconditioning

This section focuses on obtaining ω-optimal preconditioners of special structure for finding least squares solutions of $Ax = b$, where A might not be symmetric nor square. In what follows, A will be assumed to have full column rank, so that the matrix $M := A^T A$ is positive definite. Recall that for $M \succ 0$, we define $\omega(M)$ as

$$\omega(M) = \frac{\text{trace}(M)/n}{\det(M)^{1/n}},$$

i.e., it is the ratio of the arithmetic to geometric means of the eigenvalues of M. Unlike the classical condition number κ, the quantity ω is smooth, and closely related to eigenvalue clustering. These properties make it particularly well suited to deriving explicit solutions to optimality conditions for structured preconditioners.

This section is divided into three main subsections.

- Section 4.2.1 treats one-sided diagonal preconditioning, distinguishing between right- and left-sided scalings.
- Section 4.2.2 addresses two-sided diagonal scaling, establishing optimality conditions and linking them to matrix balancing.
- Section 4.2.3 extends the analysis to block-diagonal preconditioners, including triangular block structures.

A unifying theme of this section is that many classical preconditioners—Jacobi scaling, row and column normalization, and Sinkhorn–Knopp balancing—arise naturally as exact solutions or stationary points of ω-optimal problems. Table 4.1 summarizes several of these connections.

Special Structure	Preconditioner	Location	Literature
Right Diagonal	Column Normalization	Proposition 4.2.1	[52, 57]
Diagonal for Symmetric $M \succ 0$	Jacobi Preconditioner	Corollary 4.2.2	[104]
Left Diagonal	Row Normalization	Theorem 4.2.3	
Two-Sided Diagonal	Sinkhorn-Knopp/Matrix Balancing	Theorem 4.2.7	[110, 154, 160]
Right Block-Diagonal	Block QR	Corollary 4.2.11	[57, 107]
Left Block-Diagonal	Characterization	Theorem 4.2.12	

Table 4.1: Relation of ω-Optimal Preconditioners to the Literature

4.2.1 Right- and Left-Sided Diagonal

Suppose that $\text{diag}(A)$ has no zero entries. The classical Jacobi preconditioner [104], [152, Sect. 10.2] is defined by:

$$P^{-1}A = P^{-1}b, \quad P = \text{Diag}(\text{diag}(A)). \tag{4.27}$$

We note that this can be found by using: a splitting $A = M - N$ so that $A = \text{Diag}(\text{diag}(A)) - N$; or the following variational problem that implicitly finds the best approximation of the identity using diagonal matrices:

$$\min_d \|A - \text{Diag}(d)\|_F, \quad \text{Diag}(d)^{-1}A \approx I.$$

In what follows, we reinterpret such classical constructions through the lens of the ω-condition number.

Right-Sided Diagonal

The following result shows that a ω-optimal right-sided diagonal scaling of an overdetermined full rank matrix A is formed using $(\pm)$ column normalization, see [52].

Proposition 4.2.1. *Let A be $m \times n$ full column rank, and $D := \operatorname{Diag}(\operatorname{diag}(A^T A))^{-1}$. Then $AD^{1/2}$ is a ω-optimal right-sided diagonal scaling i.e.,*

$$D \in \operatorname{argmin} \left\{ \omega(D^{1/2} A^T A D^{1/2}) : \operatorname{diag}(D) \in \mathbb{R}^n_{++} \right\}.$$

Proof. The proof of the result can be found in [52, Proposition 2.1(v)] and a modified proof in [107, Proposition 2.1(3)]. □

As an immediate specialization, we obtain the following classical result.

Corollary 4.2.2. *Let $M \in \mathbb{S}^n_{++}$. Then the Jacobi preconditioner, $D = \operatorname{Diag}(\operatorname{diag}(M))^{-1}$, is the ω-optimal diagonal scaling, i.e.,*

$$D \in \operatorname{argmin} \left\{ \omega(D^{1/2} M D^{1/2}) : \operatorname{diag}(D) \in \mathbb{R}^n_{++} \right\}.$$

Proof. The proof is immediate from Proposition 4.2.1 with $A^T A$ replaced by $M \in \mathbb{S}^n_{++}$. □

Thus, in contrast to its heuristic origins, Jacobi scaling is optimal in the precise sense of minimizing ω.

Left-Sided Diagonal

In this section we let $A \in \mathcal{M}^n$ be invertible, and we consider the optimal left-sided diagonal preconditioning problem:

$$\begin{array}{ll} \min & \omega\big((\operatorname{Diag}(c)^{1/2} A)^T \operatorname{Diag}(c)^{1/2} A\big) \\ \text{s.t.} & c \in \mathbb{R}^n_{++}. \end{array} \tag{4.28}$$

For simplicity, let $C := \operatorname{Diag}(c)$. We can characterize the ω-optimal left-sided diagonal preconditioner using the above results in Section 4.2.1 for the right preconditioner.

Theorem 4.2.3. *Let $A \in \mathcal{M}^n$ be nonsingular. Then a ω-optimal left-sided diagonal preconditioner minimizing (4.28) is given by $C = \operatorname{Diag}(\operatorname{diag}(AA^T))^{-1}$.*

Proof. Note that

$$\omega\big((\operatorname{Diag}(c)^{1/2} A)^T \operatorname{Diag}(c)^{1/2} A\big) = \omega\big(A^T \operatorname{Diag}(c) A\big) = \omega\big(\operatorname{Diag}(c)^{1/2} A A^T \operatorname{Diag}(c)^{1/2}\big).$$

Hence, Proposition 4.2.1 and the above equivalence implies that

$$\begin{array}{rcl} C = \operatorname{Diag}(\operatorname{diag}(AA^T))^{-1} & \in & \operatorname{argmin}\left\{\omega(C^{1/2}AA^TC^{1/2}) : C = \operatorname{diag}(c) \in \mathbb{S}^n_{++}\right\} \\ & = & \operatorname{argmin}\left\{\omega\big((\operatorname{Diag}(c)^{1/2}A)^T \operatorname{Diag}(c)^{1/2}A\big) : C = \operatorname{diag}(c) \in \mathbb{S}^n_{++}\right\}. \end{array}$$

□

4.2.2 Two-Sided Diagonal

We now consider the problem of finding the two-sided ω-optimal diagonal scaling for nonsingular matrices $A \in \mathcal{M}^n$. This subsection is broken up into two smaller parts. The first part derives the optimality conditions for the two-sided ω-optimal scaling problem. The second part presents an iterative matrix balancing scheme that reduces ω at every iteration and whose output is proved to be a stationary point of the two-sided ω-optimal scaling problem.

Optimality Conditions for Two-Sided Problem

This part presents the formulation of the two-sided ω-optimal diagonal scaling problem and derives its optimality conditions. Namely, we consider

$$\begin{array}{ll} \min & \omega((\operatorname{Diag}(c)^{1/2}A\operatorname{Diag}(d)^{1/2})^T(\operatorname{Diag}(c)^{1/2}A\operatorname{Diag}(d)^{1/2})) \\ \text{s.t.} & c, d \in \mathbb{R}^n_{++}, \end{array} \tag{4.29}$$

where $A \in \mathcal{M}^n$ is nonsingular. For simplicity, let $C := \operatorname{Diag}(c)$ and $D := \operatorname{Diag}(d)$. We also define

$$\mathcal{Q}(c,d) := A^T \operatorname{Diag}(c) A \operatorname{Diag}(d), \;\; f(c,d) := \frac{1}{n}\operatorname{trace}(\mathcal{Q}(c,d)), \;\; g(c,d) := \det(\mathcal{Q}(c,d))^{1/n}. \tag{4.30}$$

Therefore we have the following equivalent problem to (4.29):

$$\begin{array}{ll} \min & \omega_{\mathcal{Q}}(c,d) := \frac{f(c,d)}{g(c,d)} \\ \text{s.t.} & c, d \in \mathbb{R}^n_{++}. \end{array} \tag{4.31}$$

Finally, for notational convenience, we define $\operatorname{inv} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ by

$$\operatorname{inv}(c,d) = \begin{pmatrix} \operatorname{diag}\left(\operatorname{Diag}(c)^{-1}\right) \\ \operatorname{diag}\left(\operatorname{Diag}(d)^{-1}\right) \end{pmatrix}. \tag{4.32}$$

The following result provides the gradient of $\omega_{\mathcal{Q}}(c,d)$.

Proposition 4.2.4. *The gradient of $\omega_{\mathcal{Q}}(c,d)$ is given by*

$$\nabla\omega_{\mathcal{Q}}(c,d) = \frac{1}{ng(c,d)}\begin{pmatrix} \operatorname{diag}(A\operatorname{Diag}(d)A^T) \\ \operatorname{diag}(A^T\operatorname{Diag}(c)A) \end{pmatrix} - \frac{1}{n}\omega_{\mathcal{Q}}(c,d)\operatorname{inv}(c,d), \tag{4.33}$$

where $g(c,d)$ is as in (4.30), $\omega_{\mathcal{Q}}(c,d)$ is defined in (4.31), and $\operatorname{inv}(c,d)$ is as in (4.32).

Proof. To compute $\nabla\omega_{\mathcal{Q}}(c,d)$, we have to compute $\nabla f(c,d)$ and $\nabla g(c,d)$. It is easy to see from

the definition of $f(c,d)$ in (4.30) that $\nabla f(c,d)$ can be computed as

$$\nabla f(c,d) = \frac{1}{n}\begin{pmatrix}\operatorname{diag}(A\operatorname{Diag}(d)A^T)\\ \operatorname{diag}(A^T\operatorname{Diag}(c)A)\end{pmatrix}. \tag{4.34}$$

To compute $\nabla g(c,d)$, observe that it follows from the definition of $g(c,d)$ in (4.30) and the formula for the gradient of the determinant that

$$\begin{aligned}\langle \nabla g(c,d), (\Delta c, \Delta d)\rangle &= \det(A^TA)^{1/n}\langle \nabla \det\left(CD\right)^{1/n}, (\Delta c, \Delta d)\rangle\\ &= \det(A^TA)^{1/n}\left\langle \tfrac{1}{n}\det\left(CD\right)^{\frac{1}{n}-1}\operatorname{adj}\left(CD\right), (C\Delta D + \Delta C D)\right\rangle\\ &= \tfrac{1}{n}g(c,d)\left\langle \left(CD\right)^{-1}, (C\Delta D + \Delta C D)\right\rangle\\ &= \tfrac{1}{n}g(c,d)\operatorname{trace}\left(C^{-1}\Delta C + D^{-1}\Delta D\right).\end{aligned}$$

Hence, it follows from the above equations

$$\nabla g(c,d) = \frac{1}{n}g(c,d)\begin{pmatrix}\operatorname{diag}\left(\operatorname{Diag}(c)^{-1}\right)\\ \operatorname{diag}\left(\operatorname{Diag}(d)^{-1}\right)\end{pmatrix} = \frac{1}{n}g(c,d)\operatorname{inv}(c,d). \tag{4.35}$$

It then follows from (4.34), (4.35) and the quotient rule that $\nabla\omega_{\mathcal{Q}}(c,d)$ can be computed as

$$\begin{aligned}\nabla\omega_{\mathcal{Q}}(c,d) &= \frac{1}{g(c,d)^2}\left(g(c,d)\nabla f(c,d) - f(c,d)\nabla g(c,d)\right) = \frac{1}{g(c,d)}\nabla f(c,d) - \frac{\omega_{\mathcal{Q}}(c,d)}{g(c,d)}\nabla g(c,d)\\ &= \frac{1}{ng(c,d)}\begin{pmatrix}\operatorname{diag}(A\operatorname{Diag}(d)A^T)\\ \operatorname{diag}(A^T\operatorname{Diag}(c)A)\end{pmatrix} - \frac{1}{n}\omega_{\mathcal{Q}}(c,d)\operatorname{inv}(c,d).\end{aligned}$$

□

Remark 4.2.5. *It is easy to see that $\nabla\omega_{\mathcal{Q}}(c,d)$ can be equivalently written in the useful form:*

$$\nabla\omega_{\mathcal{Q}}(c,d) = \frac{1}{n}\begin{bmatrix}0 & A\circ A\\ (A\circ A)^T & 0\end{bmatrix}\begin{pmatrix}c\\ d\end{pmatrix} - \frac{1}{n}\omega_{\mathcal{Q}}(c,d)\operatorname{inv}(c,d).$$

The following lemma proves the convexity of our objective after a logarithmic reparameterization. This observation was communicated to us by the external examiner.

Lemma 4.2.6. *The objective $\omega_{\mathcal{Q}}(c,d)$ in (4.31) is convex in (u,v) after the logarithmic change of variables $c_i = e^{u_i}$ and $d_j = e^{v_j}$.*

Proof. We note that the trace is a posynomial, and thus a change of variables $c_i = e^{u_i}$ and $d_j = e^{v_j}$ gives

$$nf(c,d) = \log\operatorname{tr}(\mathcal{Q}(c,d)) = \log\sum_{ij}\exp(u_i + v_j + \log a_{ij}^2), \tag{4.36}$$

where $f(c,d)$ and $\mathcal{Q}(c,d)$ are in (4.30) . Note that (4.36) is convex in (u,v) since it is a log-sum-exp function. On the other hand, observe that

$$\det(\mathcal{Q}(c,d)) = \det(A^T\operatorname{Diag}(c)A\operatorname{Diag}(d)) = \det(A)^2\prod_i e^{u_i}\prod_j e^{v_j},$$

and its log is affine in (u, v). Therefore, $\log(\omega_{\mathcal{Q}}(c, d))$ turns out to be the difference between a convex and an affine function. This completes the proof. □

We now state our main theorem of this subsection on a necessary condition for a vector to be globally optimal for (4.29). This relates to optimal matrix balancing discussed below.

Theorem 4.2.7. *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular, and let $e_{2n} \in \mathbb{R}^{2n}$ denote the vector of all ones. Then $c^*, d^* \in \mathbb{R}^n_{++}$ form a pair of globally optimal solutions to (4.29) if, and only if,*

$$\begin{pmatrix} \operatorname{diag}(C^* A D^* A^T) \\ \operatorname{diag}(D^* A^T C^* A) \end{pmatrix} = \alpha e_{2n}, \quad \text{for some } \alpha > 0, \tag{4.37}$$

where $C^* = \operatorname{Diag}(c^*)$, and $D^* = \operatorname{Diag}(d^*)$. Hence, if c^*, d^* is a global optimal solution of (4.29), then all rows and columns of $\sqrt{C^*} A \sqrt{D^*}$ have identical Euclidean norms, i.e., optimal two-sided ω-scaling produces a balanced matrix.

Proof. For $c^*, d^* \in \mathbb{R}^n_{++}$ to be a global optimal solution of (4.29), it must satisfy the necessary condition $\nabla \omega_{\mathcal{Q}}(c^*, d^*) = 0$. Thus, it follows from (4.33) that c^*, d^* must satisfy

$$0 = \frac{1}{ng(c^*, d^*)} \begin{pmatrix} \operatorname{diag}(AD^*A^T) \\ \operatorname{diag}(A^T C^* A) \end{pmatrix} - \frac{1}{n} \omega_{\mathcal{M}}(c^*, d^*) \operatorname{inv}(c^*, d^*).$$

It follows from taking the Hadamard product with the vector $\begin{pmatrix} c^* \\ d^* \end{pmatrix}$ on both sides of the above equation as well as multiplying both sides by $ng(c^*, d^*)$ that the following necessary condition must hold

$$f(c^*, d^*) e_{2n} = \begin{pmatrix} \operatorname{diag}(C^* A D^* A^T) \\ \operatorname{diag}(D^* A^T C^* A) \end{pmatrix}. \tag{4.38}$$

Observe that the definition of $f(c, d)$ in (4.30) implies that if c^*, d^* satisfies the above relation then any positive scalar multiple of c^*, d^* also satisfies (4.38). It then follows from this observation and (4.38) that a globally optimal solution c^*, d^* must satisfy the necessary condition (4.37) for some scalar $\alpha > 0$. The last observation of the theorem then immediately follows from the condition in (4.37).

□

Square-Root Sinkhorn-Knopp Algorithm for Two Sided Problem

Let a nonsingular matrix $A \in \mathcal{M}^n$ be given. We have seen above that the ω-optimal right-sided preconditioner is equivalent to the column normalization preconditioner. Similarly, the ω-optimal left-sided preconditioner is equivalent to the row normalization preconditioner. Therefore, we can alternate between ω-optimal right- and left-sided preconditioners and decrease the value of ω at each iteration. This yields the following computationally efficient matrix balancing scheme where we alternate between column and row normalization. This algorithm can be seen as equivalent to the Square-Root Sinkhorn-Knopp algorithm and is related to other balancing methods in the literature [110, 111, 154, 160].

We prove below in Theorem 4.2.9 that the Square-Root Sinkhorn-Knopp algorithm, namely Algorithm 4.4, converges and the outputted two-sided diagonal preconditioner satisfies the necessary

condition in (4.37) for global optimality of a two-sided ω-optimal preconditioner. This continues the theme that the ω-measure provides justification for heuristics used in the literature.

Algorithm 4.4 Square-Root Sinkhorn-Knopp Algorithm for Two Sided ω-Optimal Preconditioner

Inputs: A square nonsingular matrix $A \in \mathcal{M}^n$, a tolerance tol> 0, and a rule for the stopping criterion, stopcrit.

set stopcrit $\leftarrow \infty$;
set $A_0 \leftarrow A$;
set $k \leftarrow 0$;
1: **while** stopcrit>tol **do** (main outer loop)
2: set $k \leftarrow k + 1$;
3: compute $(\sqrt{d_k})_i = 1./\|(A_{k-1})_{:,i}\|$, $i = 1, \dots n$, where $(A_{k-1})_{:,i}$ is the i-th column of A_{k-1};
4: set $\tilde{A}_k \leftarrow A_{k-1}\,\mathrm{Diag}(\sqrt{d_k})$;
5: compute $\sqrt{c_k} \in \mathbb{R}^n$ with $(\sqrt{c_k})_i = 1./\|(\tilde{A}_k)_{i,:}\|$, $i = 1, \dots n$, where $(\tilde{A}_k)_{i,:}$ is the i-th row of $\tilde{A}_k$;
6: update stopcrit;
7: set $A_k \leftarrow \mathrm{Diag}(\sqrt{c_k})\tilde{A}_k$;
8: **end while**(main outer loop)

Output:

$$\hat{C}^{1/2} := \mathrm{Diag}\left(\prod_{j=1}^{k}\sqrt{c_j}\right) \text{ and } \hat{D}^{1/2} := \mathrm{Diag}\left(\prod_{j=1}^{k}\sqrt{d_j}\right).$$

There several viable options for the choice of stopcrit in Algorithm 4.4. One natural choice is

$$\text{stopcrit} = \max\left(\max_i \left|\|(A_k)_{i,:} - 1\|\right|, \max_j \left|\|(A_k)_{:,j} - 1\|\right|\right).$$

It is shown in Theorem 4.2.9 that the sequence A_k converges to a matrix that is balanced, i.e., one that has column and row norms all equal to one. The remark below also discusses that the ω values of the iterates are monotonically decreasing.

Remark 4.2.8. *At each iteration, $\tilde{A}_k$ (resp. A_k) is computed using the ω-optimal right-sided (resp. left-sided) preconditioner. Hence, the ω values of the iterates are monotonically decreasing, i.e., $\omega(A_k A_k^T) \leqslant \omega(\tilde{A}_k \tilde{A}_k^T) \leqslant \omega(A_{k-1} x A_{k-1}^T)$ for all $k \geqslant 1$. The output of Algorithm 4.4 thus satisfies $\omega((\hat{C}^{1/2} A \hat{D}^{1/2})^T(\hat{C}^{1/2} A \hat{D}^{1/2})) \leqslant \omega(A^T A)$.*

Theorem 4.2.9 now provides a convergence guarantee for Algorithm 4.4 and shows that the necessary condition (4.37) for a two-sided ω-optimal diagonal preconditioner holds in the limit for Algorithm 4.4.

Theorem 4.2.9. *Let A be such that $A \circ A$ has total support.*[7] *Then the sequence A_k converges*

[7] A nonnegative square matrix B is said to have total support if $B \neq 0$ and all its nonzero elements lie on a positive diagonal. That is, for each nonzero entry $b > 0$ of B, there exists a permutation σ of $[n] = \{1, \dots, n\}$ such that $B_{i,\sigma(i)} > 0$ for all $i \in [n]$, and $b = B_{j,\sigma(j)}$ for some $j \in [n]$.

linearly to a matrix $\bar{A} := \bar{C}^{1/2} A \bar{D}^{1/2}$ *that has column and row norms all equal to 1. That is,*

$$\lim_{k\to\infty} \operatorname{Diag}\left(\prod_{j=1}^{k} \sqrt{c_j}\right) A \operatorname{Diag}\left(\prod_{j=1}^{k} \sqrt{d_j}\right) = \bar{C}^{1/2} A \bar{D}^{1/2}$$

with a linear rate of convergence. Hence, it follows from Theorem 4.2.7 that limiting matrices $\bar{C}$ *and* $\bar{D}$ *satisfy the necessary condition (4.37) for a two-sided* ω*-optimal diagonal preconditioner.*

Proof. Let $\tilde{P}_k = \tilde{A}_k \circ \tilde{A}_k$ and let $P_k = A_k \circ A_k$. It follows that $\tilde{P}_k = (A_{k-1} \circ A_{k-1}) \operatorname{Diag}(d_k) = P_{k-1} \operatorname{Diag}(d_k)$ where $(d_k)_i = 1./\|(A_{k-1})_{:,i}\|^2$. Note $\|(A_{k-1})_{:,i}\|^2$ is exactly the i-th column sum of P_{k-1} so $(d_k)_i = 1./\sum_{l=1}^{n} (P_{k-1})_{l,i}$. Moreover, also observe that $P_k = \operatorname{Diag}(c_k)(\tilde{A}_k \circ \tilde{A}_k) = \operatorname{Diag}(c_k)\tilde{P}_k$. Note $(c_k)_i = 1./\sum_{l=1}^{n} (\tilde{P}_k)_{i,l}$. Hence, the sequences of iterates $\tilde{P}_k$ and P_k can be viewed as the iterates of Sinkhorn-Knopp Algorithm applied to $A \circ A$. By [110], we know then that the sequence P_k converges linearly to a matrix $\bar{P}$ where $\bar{P} := \bar{C}(A \circ A)\bar{D}$ is a doubly stochastic matrix. The sequences of iterates $\tilde{A}_k$ and A_k are just the square roots of the iterates of the Sinkhorn-Knopp algorithm applied to $A \circ A$. Therefore, it follows from the above argument that the sequence A_k converges linearly to a matrix $\bar{A} := \bar{C}^{1/2} A \bar{D}^{1/2}$ that has column and row norms all equal to 1. The last conclusion of the theorem follows from this observation and the necessary condition (4.37) in Theorem 4.2.7.

□

4.2.3 Right-, Left-Sided Block Diagonal

We now show that the above results extend directly to finding *block diagonal* preconditioners for least squares solutions of full rank possibly overdetermined linear systems $Ax = b, A \in \mathbb{R}^{m\times n}, m \geqslant n$. The preconditioner is extended from diagonal to block diagonal (block upper-triangular). This relates to sparse QR preconditioning and extends the results for ω-optimal preconditioning with partial Cholesky structure in [107, Theorem 2.7] and the block diagonal ω-optimal preconditioner in [57, Prop. 3 part 3].

Right-Sided Block Diagonal

We let the linear transformation $B = \text{Blkdiag}(\mathcal{B}) \in \mathcal{M}^n$ denote the block diagonal matrix with blocks formed from the set of square matrices $\mathcal{B} = \{B_i\}_{i=1}^k$ of order $n_i, \sum_{i=1}^k n_i = n$. For our application we restrict the blocks to be of upper-triangular structure denoted $\mathcal{R} = \{R_i\}_{i=1}^k$, and see that the best, with respect to the ω measure, comes from Q-less QR decompositions of the corresponding blocks of A.

First we choose the number of blocks k and the block sizes $n_i, \sum_{i=1}^k n_i = n$. Thus we get the block structure

$$A = \begin{bmatrix} A_1 & A_2 & \dots & A_k \end{bmatrix}, \quad A/\operatorname{Blkdiag}(\mathcal{R}) = \begin{bmatrix} A_1/R_1 & A_2/R_2 & \dots & A_k/R_k \end{bmatrix},$$

where we use the MATLAB notation that / denotes matrix division $A/R = AR^{-1}$. Optimally, we want the sizes of the blocks chosen so that the matrices $A_i^T A_i$ are chordal so that there is no loss of sparsity. Moreover, we can allow a preliminary permutation of the columns $A \leftarrow AP$ so that we can increase the effect of the preconditioner.

To extend the diagonal preconditioner results we solve:

$$\min_{\mathcal{B}}\{\omega((AB)^T(AB)) : B = \text{Blkdiag}(\mathcal{B})\}.$$

The basic result we use follows.

Proposition 4.2.10 ([57, Prop. 3 part 3],[58, Prop. 2.2])**.** *Let*

$$A = \begin{bmatrix} A_1 & A_2 & \dots & A_k \end{bmatrix}, \quad A_i \in \mathbb{R}^{m\times n_i}, \forall i,$$

be a full rank $m \times n$ matrix, $m \geqslant n$. Then an optimal block diagonal scaling

$$\mathcal{B} := \{B_1, B_2, \dots, B_k\},\ B_i \in \mathbb{R}^{n_i\times n_i}, \qquad B := \text{Blkdiag}(\mathcal{B}),$$

that minimizes the measure ω, i.e.,

$$\min\{\omega((AB)^T(AB)) : B = \text{Blkdiag}(\mathcal{B})\},$$

is found by satisfying the factorization

$$B_iB_i^T = (A_i^TA_i)^{-1}, \quad i = 1, \dots, k.$$

Corollary 4.2.11. *Let $A, \mathcal{B}, B$ be as in Proposition 4.2.10.*
(i) If we restrict the diagonal blocks B_i to be diagonal themselves, then we get the optimal diagonal preconditioner in Proposition 4.2.1.
(ii) If we restrict the diagonal blocks B_i to be upper-triangular, then we get $B_i = R_i^{-1}, A_i = Q_iR_i$ from the QR decomposition of A_i (up to sign of rows of R).

Left-Sided Block Diagonal

We continue with A full column rank but with left-sided preconditioning. We present a nonlinear equation that characterizes optimality.

Theorem 4.2.12. *Let $A \in \mathbb{R}^{m\times n}$ be full column rank with block structure given by*

$$A = \begin{bmatrix} \bar{A}_1 \\ \bar{A}_2 \\ \vdots \\ \bar{A}_\ell \end{bmatrix}, \bar{A}_j \in \mathbb{R}^{m_j\times n}, \text{rank}(\bar{A}_j) = m_j,\ \forall j.$$

Then the ω-optimal left-sided block diagonal preconditioner

$$E := \text{Blkdiag}(E_1, E_2, \dots, E_\ell),\ E_j \in \mathbb{R}^{m_j\times m_j},$$

that minimizes the measure ω, i.e.,

$$\min \omega((EA)^TEA),$$

is characterized by:

$$\bar{A}_j\bar{A}_j^T = \frac{1}{n}\left(\text{trace}\sum_{t=1}^{\ell}\left(\bar{A}_t^T(E_t^TE_t)\bar{A}_t\right)\right)\ \bar{A}_j\left(A^T(E^TE)A\right)^{-1}\bar{A}_j^T,\ \forall j = 1, \dots \ell.$$

Proof. To begin, we substitute $K := E^T E$. That is, we solve

$$\min\{\omega(A^T K A) : K = \text{Blkdiag}(K_1, K_2, \ldots, K_\ell),\ K_j \in \mathbb{R}^{m_j \times m_j}\}.$$

Notice that $A^T K A = \sum_{j=1}^{\ell} \bar{A}_j^T K_j \bar{A}_j$. Define the linear transformation $\mathcal{G}(K) = \sum_{j=1}^{\ell} \bar{A}_j^T K_j \bar{A}_j$. This yields

$$\omega_{\mathcal{G}}(K) := \frac{\text{trace}(A^T K A)/n}{\det(A^T K A)^{1/n}} = \frac{\sum_{j=1}^{\ell} \text{trace}(\bar{A}_j^T K_j \bar{A}_j)/n}{\det(A^T K A)^{1/n}}.$$

Hence, for convenience and for the remainder of the proof define

$$f(K) := \frac{1}{n} \sum_{j=1}^{\ell} \text{trace}(\bar{A}_j^T K_j \bar{A}_j) \quad \text{and} \quad g(K) := \det(A^T K A)^{1/n}.$$

The partial derivative of g with respect to the block K_j is given by

$$\begin{aligned}
\langle \nabla_{K_j} g(K), \Delta K_j \rangle &= \langle \nabla \det(\mathcal{G}(K))^{1/n}, \Delta K_j \rangle \\
&= \langle \frac{1}{n} \det(\mathcal{G}(K))^{\frac{1}{n}-1} \text{adj}(\mathcal{G}(K)), \bar{A}_j^T \Delta K_j \bar{A}_j \rangle \\
&= \frac{1}{n} g(K) \langle \mathcal{G}(K)^{-1}, \bar{A}_j^T \Delta K_j \bar{A}_j \rangle \\
&= \frac{1}{n} g(K) \langle \bar{A}_j \mathcal{G}(K)^{-1} \bar{A}_j^T, \Delta K_j \rangle.
\end{aligned}$$

Therefore, we have

$$\nabla_{K_j} f(K) = \frac{1}{n} \bar{A}_j \bar{A}_j^T \quad \text{and} \quad \nabla_{K_j} g(K) = \frac{1}{n} g(K) \bar{A}_j (A^T K A)^{-1} \bar{A}_j.$$

These in turn lead to the partial derivative of $\omega_{\mathcal{G}}$ with respect to K_j given by

$$\begin{aligned}
\nabla_{K_j} \omega_{\mathcal{G}}(K) &= \frac{1}{g(K)^2} (g(K) \nabla_{K_j} f(K) - f(K) \nabla_{K_j} g(K)) \\
&= \frac{1}{n g(K)} (\bar{A}_j \bar{A}_j^T - f(K) \bar{A}_j (A^T K A)^{-1} \bar{A}_j^T)
\end{aligned}$$

Therefore

$$\nabla \omega_{\mathcal{G}}(K) = 0 \quad \Longleftrightarrow \quad \bar{A}_j \bar{A}_j^T = \frac{1}{n} \sum_{t=1}^{\ell} \text{trace}(\bar{A}_t \bar{A}_t^T K_t)\, \bar{A}_j (A^T K A)^{-1} \bar{A}_j^T, \quad \forall j.$$

We can then recover the E_j from K using factorizations of the blocks $K_j = E_j^T E_j$. □

Finding an explicit solution for E in Theorem 4.2.12 is an open question. However, if we assume further that A is a square matrix, the following Corollary 4.2.13 connects the result from Corollary 4.2.11 to this left sided preconditioning.

Corollary 4.2.13. *Let $A \in \mathcal{M}^n$ be nonsingular. Let the block of rows, $\bar{A}_j \in \mathbb{R}^{n_j \times n}$, be given by $A^T = [\bar{A}_1^T\ \bar{A}_2^T\ \ldots\ \bar{A}_\ell^T]$, $\sum_j n_j = n$. Define $\mathcal{E} := \{E_1, E_2, \ldots, E_k\}$ by $E_i := \bar{R}_i^{-T}$, where $\bar{A}_i^T := \bar{Q}_i \bar{R}_i$*

is the QR decomposition of $\bar{A}_i^T$. *Then,* $\mathcal{E}$ *is the optimal lower triangular block diagonal scaling for*

$$\min\{\omega((EA)^T(EA)) : E = \text{Blkdiag}(E_1, E_2, \ldots, E_k)\}.$$

Proof. Since A and E are square matrices, we observe

$$\omega\big((EA)^T EA\big) = \omega\big(A^T E^T EA\big) = \omega\big(EAA^T E^T\big),$$

and thus Corollary 4.2.11 implies $E_i = \bar{R}_i^{-T}$, where $\bar{A}_i\bar{A}_i^T := \bar{R}_i^T\bar{R}_i$ as defined above. □

Remark 4.2.14. *The result in Corollary 4.2.13 agrees with our characterization in Theorem 4.2.12. Indeed, by defining* $E_i := \bar{R}_i^{-T}$ *with* $\bar{A}_i\bar{A}_i^T = \bar{R}_i^T\bar{R}_i$, *we get that* $\text{trace}(\bar{A}_t\bar{A}_t^T E_t^T E_t) = \text{trace}(\bar{R}_t^T\bar{R}_t\bar{R}_t^{-1}\bar{R}_t^{-T}) = \text{trace}(I_{n_t}) = n_t$. *Thus,*

$$\sum_{t=1}^{\ell} \text{trace}(\bar{A}_t\bar{A}_t^T E_t^T E_t) = \sum_{t=1}^{\ell} n_t = n.$$

Now, define $\mathcal{I}_j := \begin{bmatrix} 0 & I_{n_j} & 0 \end{bmatrix} \in \mathbb{R}^{n_j \times n}$, *with the identity* $I_{n_j} \in \mathcal{M}^{n_j}$ *of order* n_j, *appropriately located such that* $\mathcal{I}_j A = \bar{A}_j$. *Then,*

$$\begin{aligned}
\frac{1}{n}\left(\text{trace}\sum_{t=1}^{\ell}\left(\bar{A}_t^T E_t^T E_t \bar{A}_t\right)\right)\bar{A}_j\left(A^T E^T EA\right)^{-1}\bar{A}_j^T &= \bar{A}_j(A^T E^T EA)^{-1}\bar{A}_j^T \\
&= \mathcal{I}_j AA^{-1}E^{-1}E^{-T}A^{-T}A^T\mathcal{I}_j^T \\
&= \mathcal{I}_j E^{-1}E^{-T}\mathcal{I}_j^T \\
&= E_j^{-1}E_j^{-T} = \bar{R}_j^T\bar{R}_j = \bar{A}_j\bar{A}_j^T.
\end{aligned}$$

4.3 Computational Experiments

Our computational experiments are organized into four parts, each designed to evaluate different aspects of the proposed methods.

In Section 4.3.1, we compare the computational efficiency of our subgradient-based algorithms with existing SDP-based approaches for computing approximate κ-optimal diagonal preconditioners.
In Section 4.3.2, we assess the impact of these preconditioners on PCG performance for solving symmetric positive definite systems.
In Section 4.3.3, we study two-sided preconditioning for LSQR and compare ω-optimal scaling with κ-optimal approaches.
Finally, in Section 4.3.4, we investigate whether applying ω-optimal scaling to κ-optimally preconditioned systems can further improve PCG performance.

All experiments in Sections 4.3.1 to 4.3.4 are run on a 2023 Macbook Pro with an 8-core CPU and 128 GB of memory using MATLAB 2025b. The latest codes are available at this clickable-link or with URL `https://github.com/asujanani6/Optimal_Preconditioning`.

4.3.1 Minimizing κ Efficiently

In this subsection, we study the practical performance of our subgradient methods, Algorithms 4.2 and 4.3, for computing approximate κ-optimal diagonal preconditioners. We first compare the

proposed methods with the SDP-based approaches of [71, 149] on moderate-scale instances. We then investigate the scalability of the proposed methods on substantially larger problems for which SDP-based approaches become impractical.

Throughout this subsection, condition numbers are evaluated using MATLAB's EIGS routine with tolerance 10^{-10}. CPU times include all preprocessing and algorithmic overhead. Unless otherwise stated, the default parameter settings from [71,149] are used. Algorithm 4.2 uses tolerance 10^{-4}, maximum iteration limit 500, sets stopcrit=$2|\kappa(d_{k+1}) - \kappa(d_k)|/|\kappa(d_{k+1})) + \kappa(d_k)|$ at every iteration, and uses $\hat{\delta} = 10^{-3}$. Algorithm 4.3 sets a maximum of 80 iterations and uses the same tolerance 10^{-4}.

Moderate Sizes

In this subsection, we compare our Algorithms 4.2 and 4.3, with the SDP-based approaches of [71, 149] on moderate-scale SuiteSparse and randomly generated instances. The precise details of our line-search variant Algorithm 4.3 are given in Section 4.1.3.

Our methods yield an optimal diagonal matrix D that minimizes $\kappa(D^{1/2}MD^{1/2})$ for given $M \succ 0$. Convergence is proved for our Algorithm 4.2 that often achieved stronger reductions in κ than Algorithm 4.3; our line-search variant Algorithm 4.3 was often slightly faster in practice than Algorithm 4.2. In contrast, the method in [71] solves an SDP over a prescribed subspace of diagonal preconditioners, while [149] computes a right-scaling E for M by minimizing $\kappa((BE)^T(BE))$, with $M = B^TB$ for given nonsingular B. Hence, when implementing this latter method, we take B as a Cholesky factorization of M.

Table 4.2 reports results on eighteen matrices from the SuiteSparse Matrix Collection [48], while Table 4.3 considers randomly generated instances produced using MATLAB's SPRANDSYM routine.

The proposed methods consistently achieved substantially larger reductions in κ than the SDP-based approaches. On the SuiteSparse instances, Algorithm 4.2 and Algorithm 4.3 reduced κ on average by 79.7% and 68.5%, respectively, compared to 30.3% and 21.1% for [149] and [71]. Moreover, both proposed methods were substantially faster, with Algorithm 4.3 often providing the best runtime performance.

For the random instances, a time limit of 3600 seconds was imposed on [71] due to its substantially higher computational cost. Across all tested instances, Algorithm 4.2 and Algorithm 4.3 consistently reduced κ more effectively and more efficiently than the competing methods.

Dim & Density, & $\kappa(M)$			% Reduction in κ				CPU Time (seconds)				CPU Ratios	
n	density	$\kappa(M)$	[149]	[71]	Algorithm 4.2	Algorithm 4.3	[149]	[71]	Algorithm 4.2	Algorithm 4.3	[149]/Algorithm 4.2	[149]/Algorithm 4.3
1074	1.1e-02	2.6e+07	0.0e+00	0.0e+00	9.9e+01	9.9e+01	48.203	51.700	8.469	5.644	5.7	8.5
2003	2.1e-02	1.1e+10	0.0e+00	0.0e+00	9.8e+01	6.8e+01	330.887	324.334	12.968	0.120	25.5	2752.3
3600	2.1e-03	1.8e+07	0.0e+00	0.0e+00	6.0e+01	6.0e+01	510.244	1102.270	24.644	9.538	20.7	53.5
3134	4.6e-03	2.6e+12	-9.7e-04	0.0e+00	7.5e+01	7.0e+01	178.360	1347.551	2.542	0.410	70.2	435.3
3562	1.3e-02	1.9e+11	0.0e+00	0.0e+00	8.6e+01	7.8e+01	1621.648	948.059	36.940	1.694	43.9	957.4
1922	8.2e-03	1.7e+08	0.0e+00	0.0e+00	9.6e+01	8.8e+01	182.146	448.809	4.524	0.405	40.3	450.0
4410	1.1e-02	9.5e+08	0.0e+00	8.7e+01	9.6e+01	8.3e+01	3170.182	2602.683	4.481	2.784	707.5	1138.7
588	6.2e-02	2.8e+04	9.9e+01	0.0e+00	9.0e+01	9.4e+01	19.195	7.152	0.463	1.379	41.5	13.9
494	6.8e-03	2.4e+06	9.0e+01	0.0e+00	9.1e+01	3.3e+01	9.222	2.959	0.832	0.116	11.1	79.8
662	5.6e-03	7.9e+05	9.5e+01	0.0e+00	7.4e+01	4.5e+01	16.293	5.947	0.324	0.203	50.2	80.3
1824	1.2e-02	1.9e+06	0.0e+00	0.0e+00	8.7e+01	7.1e+01	182.642	165.698	3.468	1.958	52.7	93.3
2146	1.6e-02	1.7e+03	0.0e+00	5.3e+01	5.5e+01	4.9e+01	428.711	137.240	2.604	16.508	164.6	26.0
2910	2.1e-02	6.0e+06	0.0e+00	0.0e+00	9.1e+01	7.5e+01	619.758	613.440	8.306	3.731	74.6	166.1
237	1.8e-02	2.0e+07	2.1e+01	5.5e+00	7.9e+01	6.5e+01	2.178	0.620	0.276	0.056	7.9	38.8
957	4.5e-03	5.1e+09	2.8e+01	3.5e+01	6.0e+01	6.5e+01	10.107	8.676	6.419	0.145	1.6	69.7
100	5.9e-02	1.6e+03	3.8e+01	3.1e+01	3.5e+01	3.5e+01	0.800	0.297	0.036	0.970	22.2	0.8
468	2.4e-02	1.1e+04	8.3e+01	7.9e+01	7.7e+01	7.1e+01	15.428	2.957	1.081	1.298	14.3	11.9
729	8.7e-03	2.4e+09	9.2e+01	8.9e+01	8.6e+01	8.4e+01	39.553	6.439	0.987	1.108	40.1	35.7

Table 4.2: Comparison of Algorithms for Min κ on SuiteSparse

Dim & Density, & $\kappa(A)$			% Reduction in κ				CPU Time (seconds)				CPU Ratios	
n	density	$\kappa(M)$	[149]	[71]	Algorithm 4.2	Algorithm 4.3	[149]	[71]	Algorithm 4.2	Algorithm 4.3	[149]/Algorithm 4.2	[149]/Algorithm 4.3
2000	1.0e-03	2.0e+06	-4.8e+01	4.8e+01	5.3e+01	6.6e+01	14.498	76.446	4.470	1.154	3.2	12.6
2500	8.1e-04	2.5e+06	-5.0e+01	2.9e+01	5.2e+01	5.2e+01	21.500	229.744	9.068	1.233	2.4	17.4
3000	6.7e-04	3.0e+06	-1.8e+01	0.0e+00	1.9e+01	1.3e+01	32.757	358.866	5.011	0.476	6.5	68.8
3500	5.8e-04	3.5e+06	-1.9e+01	2.5e+01	4.3e+01	4.9e+01	43.032	565.527	8.757	1.961	4.9	21.9
4000	5.0e-04	4.0e+06	-4.4e+01	0.0e+00	2.8e+01	6.3e+00	56.246	631.303	11.281	0.519	5.0	108.4
4500	4.5e-04	4.5e+06	-3.0e+01	0.0e+00	3.3e+01	4.2e+01	69.000	888.172	3.774	2.202	18.3	31.3
5000	4.0e-04	5.0e+06	-4.4e+01	0.0e+00	3.9e+01	3.9e+01	85.069	2009.496	13.994	2.341	6.1	36.3
5500	3.7e-04	5.5e+06	-3.5e+01	1.4e+01	2.8e+01	2.8e+01	104.179	2512.849	2.763	2.614	37.7	39.8
6000	3.3e-04	6.0e+06	-4.3e+01	0.0e+00	3.0e+01	3.4e+01	127.060	3680.562	5.383	2.748	23.6	46.2
6500	3.1e-04	6.5e+06	-2.9e+01	1.9e+01	2.7e+01	3.2e+01	146.476	3755.564	4.694	3.124	31.2	46.9
7000	2.9e-04	7.0e+06	-2.9e+01	2.4e+01	3.3e+01	3.0e+01	168.726	3838.763	11.441	3.243	14.7	52.0
7500	2.7e-04	7.5e+06	-4.1e+01	2.5e+01	3.2e+01	2.8e+01	193.993	3879.417	12.929	3.649	15.0	53.2
8000	2.5e-04	8.0e+06	-1.9e+01	3.0e+01	3.2e+01	2.7e+01	215.347	3790.944	16.234	3.645	13.3	59.1
8500	2.4e-04	8.5e+06	-2.8e+01	3.6e+01	2.5e+01	2.6e+01	246.213	3929.044	8.086	4.716	30.5	52.2
9000	2.2e-04	9.0e+06	-1.5e+01	-9.9e+00	9.1e+00	3.2e+00	271.863	3903.914	7.354	0.931	37.0	292.1
9500	2.1e-04	9.5e+06	-4.2e+01	3.6e+01	2.6e+01	2.3e+01	323.748	4147.347	13.329	5.681	24.3	57.0
10000	2.0e-04	1.0e+07	-3.2e+01	1.2e+01	2.2e+01	1.8e+01	341.504	4304.042	8.240	5.232	41.4	65.3
10500	1.9e-04	1.0e+07	-1.2e+01	0.0e+00	2.7e+01	2.1e+01	384.690	4302.243	20.255	6.627	19.0	58.0
11000	1.8e-04	1.1e+07	-4.1e+01	3.0e+00	2.2e+01	1.9e+01	463.158	4273.218	11.945	6.085	38.8	76.1
11500	1.7e-04	1.1e+07	-3.4e+01	0.0e+00	2.3e+01	2.0e+01	487.216	4000.550	9.915	6.705	49.1	72.7
12000	1.7e-04	1.2e+07	-3.0e+01	0.0e+00	2.2e+01	1.9e+01	511.686	4581.828	10.963	6.709	46.7	76.3
12500	1.6e-04	1.3e+07	-3.1e+01	0.0e+00	4.0e+00	3.2e+00	546.856	4059.532	3.496	2.080	156.4	263.0
13000	1.5e-04	1.3e+07	-2.9e+01	0.0e+00	2.4e+01	1.8e+01	593.252	4586.045	40.099	7.383	14.8	80.4
13500	1.5e-04	1.4e+07	-3.0e+01	0.0e+00	2.2e+01	1.8e+01	665.554	5145.811	15.941	8.300	41.8	80.2
14000	1.4e-04	1.4e+07	-3.0e+01	0.0e+00	2.2e+01	1.7e+01	719.336	4122.784	14.430	7.864	49.9	91.5
14500	1.4e-04	1.4e+07	-3.4e+01	0.0e+00	2.1e+01	1.2e+01	901.989	4606.777	24.572	6.888	36.7	130.9
15000	1.3e-04	1.5e+07	-4.1e+01	0.0e+00	1.6e+01	1.1e+01	920.641	5115.021	14.827	6.354	62.1	144.9

Table 4.3: Comparison of Algorithms for Min κ on Random

Large Sizes

Tables 4.4 and 4.5 illustrate the scalability of Algorithms 4.2 and 4.3 on substantially larger instances for which the SDP-based methods of [71, 149] became computationally impractical. Table 4.4 considers matrices from the SuiteSparse Matrix Collection, while Table 4.5 considers randomly generated matrices with very large condition numbers.

On the SuiteSparse instances, Algorithm 4.2 and Algorithm 4.3 reduced κ by averages of 45.7% and 41.9%, respectively, while often requiring less than one minute. On the random instances, which satisfy $\kappa(M) > 10^{11}$, the corresponding average reductions were 11.6% and 4.4%.

Dim & Density, & $\kappa(M)$			% Reduction in κ		CPU Time	
n	density	$\kappa(M)$	Algorithm 4.2	Algorithm 4.3	Algorithm 4.2	Algorithm 4.3
14822	3.3e-03	2.0e+06	3.9e+01	3.0e+01	7.850	283.641
15439	1.1e-03	4.4e+12	8.0e+01	7.2e+01	8.734	2.185
15439	6.5e-05	6.1e+09	4.2e+01	5.4e+01	2.283	2.725
17361	1.1e-03	2.5e+06	1.3e+01	1.2e+01	1.097	9.086
17361	3.4e-03	1.1e+09	5.6e+01	3.0e+01	19.675	9.306
23052	2.2e-03	7.4e+11	9.0e+01	7.2e+01	503.668	6.933
30401	5.1e-04	5.8e+03	9.1e+01	9.3e+01	204.718	14.951
36441	4.3e-04	2.6e+03	8.5e+01	9.2e+01	459.869	59.632
40806	1.2e-04	8.1e+01	4.5e-02	1.1e-01	29.901	16.301
48962	2.1e-04	1.6e+05	6.2e+00	4.1e+00	22.109	43.123
150102	3.2e-05	1.3e+07	4.6e-03	1.5e+00	53.450	864.384

Table 4.4: Comparison of Algorithms for Min κ on Large SuiteSparse

Dim & Density, & $\kappa(M)$			% Reduction in κ		CPU Time	
n	density	$\kappa(M)$	Algorithm 4.2	Algorithm 4.3	Algorithm 4.2	Algorithm 4.3
50000	3.8e-05	1.0e+11	1.7e+01	7.7e+00	244.577	74.798
60000	3.2e-05	1.2e+11	1.5e+01	6.5e+00	400.467	103.404
70000	2.7e-05	1.4e+11	1.4e+01	5.6e+00	492.698	123.284
80000	2.4e-05	1.6e+11	1.1e+01	5.0e+00	436.111	198.894
90000	2.1e-05	1.8e+11	1.0e+01	4.3e+00	550.245	223.299
100000	1.9e-05	2.0e+11	1.2e+01	4.0e+00	986.187	250.367
110000	1.7e-05	2.2e+11	1.1e+01	3.7e+00	1162.321	293.284
120000	1.6e-05	2.4e+11	1.1e+01	3.4e+00	1172.119	332.887
130000	1.5e-05	2.6e+11	8.5e+00	3.1e+00	1274.404	386.352
140000	1.4e-05	2.8e+11	9.1e+00	2.9e+00	1500.467	480.042
150000	1.3e-05	3.0e+11	8.7e+00	2.7e+00	1677.360	498.512

Table 4.5: Comparison of Algorithms for Min κ on Large Random

In summary, the results show that the proposed subgradient-based methods not only scale substantially better than existing SDP-based approaches, but also achieve significantly larger reductions in κ at a fraction of the computational cost.

4.3.2 PCG Comparison for Solving Linear Systems

In this subsection, we evaluate the effect of preconditioning on PCG performance for solving symmetric positive definite systems $Mx = b$. We compare the diagonal preconditioners computed by Algorithm 4.3 with those obtained using the κ-optimal scaling method from [149]. PCG is run with a relative residual tolerance of 10^{-4} and a maximum of 5×10^6 iterations. The results are reported in Table 4.6. All test matrices are randomly generated using MATLAB's `sprandsym` function with varying dimensions, densities, and κ values.

Table 4.6 shows that, over the 47 test instances, PCG requires on average 442,147 iterations with the preconditioner from Algorithm 4.3, compared to 549,002 iterations with the preconditioner from [149]. Averaged over all test instances, PCG with Algorithm 4.3 requires approximately 20% fewer iterations while reducing total CPU time by nearly an order of magnitude compared with the method of [149].

Dim, Density, & $\kappa(A)$			% Reduction in κ		PCG Iterations		PCG CPU Time		Total CPU Time	
n	density	$\kappa(M)$	[149]	Algorithm 4.3	[149]	Algorithm 4.3	[149]	Algorithm 4.3	[149]	Algorithm 4.3
1000	3.3e-03	1.0e+09	-2.6e+01	3.6e+01	115264	99023	1.14	0.99	5.33	1.29
1300	2.5e-03	1.3e+09	-2.6e+01	6.0e+01	153760	101197	1.89	1.23	8.41	1.49
1600	2.0e-03	1.6e+09	-4.3e+01	-1.0e-13	180929	179469	2.67	2.67	12.17	2.71
1900	1.7e-03	1.9e+09	-3.5e+01	6.9e+01	202276	121629	3.43	2.07	16.17	2.87
2200	1.4e-03	2.2e+09	-4.9e+01	7.1e+01	245521	126567	4.82	2.50	21.54	3.39
2500	1.2e-03	2.5e+09	-2.3e+01	2.7e+01	252757	221181	5.43	4.69	29.06	5.07
2800	1.1e-03	2.8e+09	-4.2e+01	1.9e+01	289351	248376	6.69	5.74	33.60	6.12
3100	9.9e-04	3.1e+09	-2.9e+01	6.6e+01	306195	174755	8.02	4.56	45.28	6.01
3400	9.0e-04	3.4e+09	-3.3e+01	5.4e+01	325838	215488	9.07	6.05	54.35	7.46
3700	8.3e-04	3.7e+09	-2.5e+01	4.4e+01	330583	256787	9.89	7.67	63.61	9.21
4000	7.6e-04	4.0e+09	-4.4e+01	5.7e+01	366269	234646	11.60	7.40	65.77	9.21
4300	7.1e-04	4.3e+09	-2.9e+01	5.4e+01	378370	252209	13.49	9.00	88.27	10.97
4600	6.6e-04	4.6e+09	-4.2e+01	4.1e-13	408179	384608	15.52	14.65	89.55	14.74
4900	6.2e-04	4.9e+09	-1.1e+01	8.7e+00	399371	382842	15.87	15.31	108.88	15.75
5200	5.8e-04	5.2e+09	-3.7e+01	-3.7e-14	458497	418282	19.04	17.32	115.01	17.43
5500	5.5e-04	5.5e+09	-3.3e+01	3.9e+01	460157	340511	19.97	14.79	128.37	16.99
5800	5.2e-04	5.8e+09	-3.2e+01	4.6e+01	469674	336957	21.25	15.32	143.80	17.73
6100	4.9e-04	6.1e+09	-5.0e+01	4.3e+01	506736	353612	23.89	16.68	157.42	19.34
6400	4.7e-04	6.4e+09	-2.2e+01	4.3e+01	492739	361695	23.98	17.59	166.18	20.56
6700	4.4e-04	6.7e+09	-4.0e+01	4.1e+01	526574	378065	25.65	18.54	189.81	21.40
7000	4.2e-04	7.0e+09	-2.9e+01	3.9e+01	533440	394622	28.16	20.13	202.50	22.93
7300	4.1e-04	7.3e+09	-5.3e+01	2.7e+00	572826	511612	32.32	27.95	234.05	28.40
7600	3.9e-04	7.6e+09	-2.9e+01	-3.8e-14	555453	528584	31.27	29.77	232.71	29.89
7900	3.7e-04	7.9e+09	-2.2e+01	3.5e+01	569653	434589	33.24	25.44	274.74	29.14
8200	3.6e-04	8.2e+09	-6.6e+01	3.5e+01	614529	448278	37.09	26.97	323.07	30.42
8500	3.5e-04	8.5e+09	-2.7e+01	3.4e+01	573590	458996	35.49	28.28	292.58	32.18
8800	3.3e-04	8.8e+09	-3.6e+01	1.6e+01	625430	532398	40.04	34.12	322.79	35.85
9100	3.2e-04	9.1e+09	-1.8e+01	1.5e+01	597815	541439	39.21	35.60	324.73	37.50
9400	3.1e-04	9.4e+09	-2.0e+01	1.1e+01	635767	565196	42.77	37.92	374.03	39.77
9700	3.0e-04	9.7e+09	-2.9e+01	3.1e+01	639683	508698	44.28	35.17	391.15	40.05
10000	2.9e-04	1.0e+10	-4.5e+01	3.1e+01	687294	521357	48.91	37.36	454.89	41.62
10300	2.8e-04	1.0e+10	-2.5e+01	2.9e+01	667536	538749	49.08	39.58	506.66	44.78
10600	2.8e-04	1.1e+10	-4.4e+01	2.9e+01	725311	548238	54.63	41.32	510.40	46.14
10900	2.7e-04	1.1e+10	-3.9e+01	2.6e+01	725604	566231	55.80	43.64	491.97	49.26
11200	2.6e-04	1.1e+10	-2.4e+01	2.8e+01	693639	569480	55.03	44.93	498.31	49.79
11500	2.5e-04	1.1e+10	-8.9e+00	2.4e+01	673910	597322	54.77	48.20	521.32	53.85
11800	2.5e-04	1.2e+10	-3.9e+01	1.7e+01	757749	627850	62.92	52.02	581.98	55.67
12100	2.4e-04	1.2e+10	-2.9e+01	2.6e+01	748235	601448	63.64	51.45	640.96	57.59
12400	2.3e-04	1.2e+10	-4.3e+01	2.5e+01	777435	614047	68.12	54.42	653.76	60.37
12700	2.3e-04	1.3e+10	-3.3e+01	2.5e+01	775871	620193	70.07	56.35	686.85	63.40
13000	2.2e-04	1.3e+10	-3.2e+01	1.1e+01	771101	681834	73.02	65.22	703.04	68.91
13300	2.2e-04	1.3e+10	-2.7e+01	2.4e+01	766793	648685	75.93	64.69	727.64	71.98
13600	2.1e-04	1.4e+10	-3.5e+01	1.0e+01	814120	708483	82.37	72.15	752.62	75.16
13900	2.1e-04	1.4e+10	-5.5e+01	2.3e+01	863835	662119	89.35	68.57	824.58	76.07
14200	2.0e-04	1.4e+10	-4.4e+01	2.3e+01	858674	673736	90.12	71.18	887.64	78.25
14500	2.0e-04	1.5e+10	-4.3e+01	9.4e+00	872710	741010	94.29	80.77	869.02	84.41
14800	2.0e-04	1.5e+10	-3.0e+01	8.5e+00	836054	747797	91.59	82.12	874.94	84.96

Table 4.6: PCG Comparison Using Preconditioners Found by Algorithm in [149] and Algorithm 4.3

4.3.3 LSQR Comparison: Algorithm 4.4 vs. Two-Sided κ-optimal Scaling in [149]

In Section 4.3.3, we compare our two-sided Algorithm 4.4, which decreases ω at every iteration, with the two-sided κ-optimal preconditioner of [149] for solving $\min_x \|b - Ax\|$ with LSQR.

The results are reported in Tables 4.7 to 4.9. Table 4.7 and Table 4.8 consider SuiteSparse and randomly generated matrices, respectively, with $n \leqslant 300$, since the method of [149] could not handle larger instances. In these experiments, LSQR is run with tolerance 10^{-8} and a maximum of 5000 iterations. Table 4.9 considers SuiteSparse matrices with $n \geqslant 2000$ and compares LSQR with and without the preconditioner from Algorithm 4.4. Here, LSQR is run with tolerance 10^{-6} and a maximum of 100000 iterations.

In all tables, $S := (\hat{C}^{1/2} A \hat{D}^{1/2})^T (\hat{C}^{1/2} A \hat{D}^{1/2})$ and $T := (D_1^{1/2} A D_2^{-1/2})^T (D_1^{1/2} A D_2^{-1/2})$ where $\hat{C}^{1/2}, \hat{D}^{1/2}$ and $D_1^{1/2}, D_2^{-1/2}$ are the preconditioners computed by Algorithm 4.4 and [149], respectively. Reported CPU times include both preconditioning and LSQR runtimes.

Dim & Density, & $\kappa(A)$			Ratio of Condition Numbers		CPU Time for Prec		LSQR Iterations		Total CPU Time	
n	density	$\kappa(A)$	$\kappa(S)/\kappa(T)$	$\omega(S)/\omega(T)$	Algorithm 4.4	[149]	Algorithm 4.4	[149]	Algorithm 4.4	[149]
300	6.6e-03	5.2e+03	1.3e+00	1.0e+00	0.006	23.358	5	4	0.015	23.36
130	6.1e-02	6.1e+10	1.1e-01	8.2e-01	0.002	27.943	9	35	0.005	27.95
200	2.0e-02	2.4e+03	1.0e+00	1.0e+00	0.000	19.308	750	756	0.022	19.32
104	9.2e-02	5.5e+03	1.6e+00	9.7e-01	0.001	10.056	301	288	0.004	10.06
216	9.3e-02	3.3e+04	8.9e-01	4.6e-01	0.002	309.028	56	175	0.004	309.03
115	9.3e-02	3.7e+02	1.5e+00	6.7e-01	0.001	14.324	267	363	0.004	14.33
185	2.8e-02	1.8e+05	5.7e+00	7.5e-01	0.003	40.382	242	328	0.006	40.39
216	1.7e-02	1.0e+02	1.3e+00	9.0e-01	0.001	17.672	183	215	0.004	17.67
207	1.3e-02	1.4e+08	3.9e-01	3.4e-01	0.003	19.862	122	709	0.006	19.87
137	2.1e-02	1.8e+04	1.2e+00	7.9e-01	0.002	12.505	61	95	0.003	12.51
225	2.6e-02	7.1e+06	2.5e+00	8.1e-01	0.003	146.868	78	86	0.004	146.87
198	1.4e-01	3.0e+03	1.3e+00	5.8e-01	0.002	88.701	631	973	0.010	88.71
265	2.5e-02	1.4e+03	1.6e+00	5.0e-01	0.002	38.714	735	1379	0.010	38.73
100	4.0e-02	1.5e+04	1.0e+00	9.7e-01	0.000	8.944	114	132	0.002	8.95
105	8.0e-02	7.2e+02	1.5e+00	6.6e-01	0.000	14.751	163	227	0.002	14.75
135	3.6e-02	9.2e+05	1.0e+00	2.3e-01	0.000	17.484	164	850	0.002	17.49
120	6.0e-02	4.3e+08	3.1e-03	2.7e-02	0.001	24.967	77	845	0.003	24.97
100	7.1e-02	2.4e+12	5.8e-02	2.4e-01	0.001	28.979	19	108	0.003	28.98
136	2.6e-02	2.5e+05	1.9e+00	8.0e-01	0.001	19.051	132	175	0.003	19.05
100	4.0e-02	1.3e+04	2.3e+00	9.3e-01	0.000	10.800	339	385	0.003	10.80
300	3.5e-02	8.5e+05	2.2e+00	6.0e-01	0.002	581.259	1195	1983	0.016	581.28
132	2.4e-02	4.2e+11	2.1e+00	1.0e-01	0.000	22.553	93	1356	0.002	22.56

Table 4.7: LSQR Comparison on Small Suitesparse Matrices: Algorithm 4.4 vs [149]

Dim & Density, & $\kappa(A)$			Ratio of Condition Numbers		CPU Time for Prec		LSQR Iterations		Total CPU Time	
n	density	$\kappa(A)$	$\kappa(S)/\kappa(T)$	$\omega(S)/\omega(T)$	Algorithm 4.4	[149]	Algorithm 4.4	[149]	Algorithm 4.4	[149]
50	2.5e-01	1.0e+03	1.2e+00	9.4e-01	0.007	16.583	18	18	0.02	16.59
60	2.4e-01	1.0e+03	5.2e+00	5.1e-01	0.001	11.914	46	67	0.01	11.92
70	2.3e-01	1.1e+03	1.4e+00	8.3e-01	0.001	17.909	22	27	0.00	17.91
80	2.2e-01	1.1e+03	2.0e+00	2.6e-01	0.002	13.848	51	191	0.00	13.85
90	2.2e-01	1.2e+03	2.2e+00	5.0e-01	0.002	18.674	43	90	0.00	18.68
100	2.1e-01	1.2e+03	5.2e+00	4.9e-01	0.003	27.400	52	72	0.00	27.40
110	2.1e-01	1.3e+03	2.2e+00	6.4e-01	0.003	45.059	52	52	0.00	45.06
120	2.0e-01	1.4e+03	2.0e+00	7.0e-01	0.002	37.248	63	57	0.00	37.25
130	2.0e-01	1.5e+03	7.3e+00	4.0e-01	0.002	42.239	96	140	0.00	42.24
140	1.9e-01	1.6e+03	1.1e+01	5.4e-01	0.002	55.521	54	83	0.00	55.52
150	1.9e-01	1.7e+03	2.4e+00	8.0e-01	0.001	102.427	46	35	0.00	102.43
160	1.9e-01	1.8e+03	6.2e+00	5.3e-01	0.002	90.146	66	81	0.00	90.15
170	1.9e-01	1.9e+03	2.4e+01	5.1e-01	0.003	101.889	84	111	0.01	101.89
180	1.8e-01	2.1e+03	9.3e+00	4.3e-01	0.001	128.101	90	109	0.00	128.10
190	1.8e-01	2.2e+03	8.1e+00	3.4e-01	0.001	116.720	113	219	0.00	116.72
200	1.8e-01	2.5e+03	9.1e+00	4.7e-01	0.001	147.824	94	124	0.00	147.83
210	1.8e-01	2.7e+03	1.2e+01	5.2e-01	0.002	191.344	121	113	0.00	191.35
220	1.8e-01	3.1e+03	4.6e+00	5.3e-01	0.002	293.214	94	75	0.00	293.22
230	1.7e-01	3.5e+03	2.2e+00	3.8e-01	0.001	175.652	105	508	0.00	175.66
240	1.7e-01	4.0e+03	6.1e+00	4.5e-01	0.001	287.572	129	116	0.00	287.57
250	1.7e-01	4.8e+03	1.1e+01	5.4e-01	0.002	433.302	107	92	0.00	433.30
260	1.7e-01	5.9e+03	4.0e+00	5.1e-01	0.002	384.679	139	166	0.01	384.68
270	1.7e-01	7.8e+03	3.6e+00	6.3e-01	0.002	431.242	121	75	0.00	431.24
280	1.7e-01	1.1e+04	1.5e+01	3.9e-01	0.002	498.242	153	298	0.01	498.25
290	1.7e-01	2.0e+04	1.6e+01	4.5e-01	0.002	750.635	178	128	0.01	750.64
300	1.7e-01	1.0e+05	1.0e+01	2.8e-01	0.002	751.572	197	264	0.01	751.58

Table 4.8: LSQR Comparison on Small Random Matrices: Algorithm 4.4 vs [149]

Dim & Density, & $\omega(A)$			Ratio of Omega & Prec CPU Time		LSQR Iter		Total CPU Time	
n	density	$\omega(A)$	$\omega(S)/\omega(A)$	Algorithm 4.4 CPU	No Prec	Algorithm 4.4	No Prec	Algorithm 4.4
14214	1.3e-03	5.1e+01	3.2e-02	1.0e-01	100000	8963	3.63e+01	3.43e+00
4119	2.1e-03	4.1e+05	3.6e-06	1.7e-02	5461	946	4.98e-01	1.04e-01
7479	1.2e-03	2.8e+04	4.8e-05	3.0e-02	9393	1758	1.85e+00	3.81e-01
9271	1.4e-03	8.2e+06	1.7e-07	4.1e-02	1798	377	4.44e-01	1.37e-01
4562	6.3e-03	2.3e+02	7.4e-03	1.3e-02	6497	1070	1.11e+00	2.00e-01
3072	1.3e-02	1.9e+00	1.0e+00	9.8e-03	1358	1357	5.43e-01	6.06e-01
11341	7.6e-04	6.0e+01	5.0e-02	3.9e-02	26647	3050	7.55e+00	9.09e-01
2048	2.9e-03	2.8e+00	9.3e-01	1.6e-03	4730	4187	2.10e-01	1.93e-01
8192	7.3e-04	2.8e+00	9.4e-01	6.1e-03	19233	16963	3.15e+00	2.79e+00
14734	4.4e-04	2.3e+00	7.2e-01	4.2e-02	10631	6883	3.43e+00	2.26e+00
2283	9.2e-03	3.6e+01	5.6e-02	9.1e-03	17604	1305	1.15e+00	9.74e-02
2000	2.0e-03	2.5e+00	1.0e+00	7.9e-04	1578	1578	6.33e-02	6.54e-02
2000	5.9e-03	4.2e+00	5.9e-01	3.5e-03	9674	3842	4.76e-01	1.95e-01
5000	2.4e-03	4.2e+00	5.7e-01	6.9e-03	29362	7483	4.56e+00	1.15e+00
7000	1.7e-03	4.3e+00	5.7e-01	9.3e-03	50422	13747	9.93e+00	2.82e+00
7000	1.7e-03	4.2e+00	5.7e-01	9.4e-03	48882	11272	9.71e+00	2.23e+00
3175	8.4e-03	3.0e+01	7.0e-02	9.3e-03	16004	2157	2.06e+00	2.90e-01
5952	6.3e-04	3.7e+02	4.6e-03	1.8e-02	13984	5192	1.55e+00	5.96e-01
13694	3.9e-04	1.8e+02	7.1e-03	8.0e-02	100000	2551	3.08e+01	8.70e-01
5832	9.0e-03	1.7e+00	1.0e+00	1.0e-02	2397	2390	6.25e-01	6.36e-01
9801	9.1e-04	1.5e+00	1.0e+00	2.6e-03	3132	3036	7.93e-01	7.72e-01
4000	5.5e-04	1.3e+09	7.5e-10	7.3e-03	7194	5	5.05e-01	8.63e-03
5940	2.4e-03	6.6e+00	4.6e-01	3.6e-02	100000	44539	1.74e+01	7.78e+00
4326	3.3e-03	1.1e+05	1.7e-05	1.4e-02	34107	2486	4.85e+00	3.69e-01
9800	6.0e-04	2.6e+00	9.9e-01	6.4e-03	18382	18177	4.37e+00	4.29e+00

Table 4.9: LSQR Comparison on Large Suitesparse Matrices: Algorithm 4.4 vs No Preconditioning

The results in Tables 4.7 and 4.8 show that although the two-sided κ-optimal preconditioner in [149] often achieves a stronger reduction in κ, Algorithm 4.4 consistently yields lower ω values at a significantly lower computational cost. Importantly, reductions in ω correlate more strongly with LSQR iteration counts, resulting in faster convergence on the majority of tested instances.

In summary, these experiments reinforce that ω-optimal preconditioning offers a more practical and computationally efficient alternative to κ-optimal scaling for LSQR, particularly in large-scale and nonsymmetric settings.

4.3.4 From κ-optimal M to "Improve PCG" using ω-Optimal Scaling

We study the effect of applying ω-optimal (Jacobi) diagonal scaling to a κ-optimally scaled matrix $M \succ 0$.[8] To this end, we construct the matrix M using Theorem 4.1.7: the extreme eigenpairs satisfy (4.7) and (4.8), while the remaining eigenvalues are spread uniformly in (λ_n, λ_1) with orthonormal eigenvectors chosen in the orthogonal complement of $\text{span}\{x_1, x_n\}$. The remaining eigenvectors are generated via a sparse QR factorization, so the density of M cannot be prescribed exactly in advance. A more detailed description of the construction is provided in our code.

After constructing M, we apply the ω-optimal (Jacobi) scaling $J = D^{1/2}MD^{1/2}$ where $D = \text{Diag}(1./\,\text{diag}(M))$, and compare PCG applied to $Mx = b$ and to the scaled system $J(D^{-1/2}x) =$

[8]For the large problems in Table 4.11 we used the fastlinux server, cpu155.math.private, a Dell PowerEdge R650 with two Intel Xeon Gold 6334 8-core 3.6 GHz (Ice Lake) processors and 256 GB RAM.

$D^{1/2}b$. Iteration counts and runtimes are averaged over five initial points. The results are reported in Tables 4.10 and 4.11.

Dim & Density		Ratios in conds (J, M)		J: ω-opt of M		Ratios M/J	
n	density	$\kappa(J)/\kappa(M)$	$\omega(J)/\omega(M)$	iters	cpu	iters	cpu
5000	9.40e-03	2.98e+00	7.368e-01	19.4	3.927e-03	25.2	13.9
10000	7.44e-03	3.31e+00	7.362e-01	18.8	6.698e-03	36.6	30.5
15000	6.02e-03	2.64e+00	7.365e-01	17.0	1.418e-02	41.6	39.0
20000	4.57e-03	3.56e+00	7.363e-01	20.6	2.861e-02	41.6	41.1
25000	3.47e-03	3.71e+00	7.370e-01	20.6	8.006e-02	30.9	29.1
30000	2.42e-03	3.52e+00	7.362e-01	20.2	4.327e-02	48.0	43.0
35000	1.56e-03	2.77e+00	7.371e-01	16.0	4.174e-02	36.5	36.1
40000	8.93e-04	2.89e+00	7.364e-01	18.0	2.535e-02	46.7	38.9
45000	4.15e-04	3.93e+00	7.370e-01	21.8	1.820e-02	28.5	27.0
50000	4.12e-04	4.12e+00	7.361e-01	24.0	2.679e-02	49.5	43.7

Table 4.10: PCG tol. $1e-7$; Medium; κ-opt **VS** J, ω-opt of M

Dim & Density		Ratios in conds (J, M)		J: ω-opt of M		Ratios M/J	
n	density	$\kappa(J)/\kappa(M)$	$\omega(J)/\omega(M)$	iters	cpu	iters	cpu
50000	9.05e-03	3.26e+00	7.366e-01	15.6	3.332e-01	48.7	42.5
52000	8.17e-03	4.34e+00	7.362e-01	20.8	4.245e-01	51.7	46.9
54000	7.49e-03	3.55e+00	7.365e-01	17.0	3.578e-01	47.5	42.4
56000	6.71e-03	3.73e+00	7.363e-01	18.0	3.672e-01	57.1	51.0
58000	5.98e-03	3.33e+00	7.369e-01	16.8	3.367e-01	38.5	34.1
60000	5.24e-03	3.90e+00	7.362e-01	17.6	3.316e-01	62.1	55.4
62000	4.63e-03	4.43e+00	7.371e-01	18.0	3.233e-01	33.2	29.7
64000	4.01e-03	4.37e+00	7.365e-01	22.0	3.582e-01	38.9	35.5
66000	3.46e-03	3.24e+00	7.369e-01	16.0	2.457e-01	38.9	34.4
68000	2.87e-03	4.31e+00	7.361e-01	20.8	2.756e-01	59.8	54.0
70000	2.40e-03	4.25e+00	7.363e-01	20.4	2.445e-01	46.5	42.0
72000	1.96e-03	4.05e+00	7.369e-01	17.8	1.883e-01	36.4	32.1
74000	1.54e-03	3.68e+00	7.362e-01	17.2	1.438e-01	64.8	56.9
76000	1.22e-03	3.95e+00	7.366e-01	18.0	1.194e-01	43.6	38.4
78000	9.02e-04	2.98e+00	7.361e-01	16.4	8.235e-02	72.1	62.2
80000	6.44e-04	3.50e+00	7.366e-01	17.0	6.009e-02	43.4	36.6
82000	4.20e-04	3.81e+00	7.366e-01	19.2	4.669e-02	40.2	33.9
84000	2.46e-04	4.12e+00	7.363e-01	20.2	3.227e-02	46.2	37.8
86000	1.20e-04	2.92e+00	7.370e-01	17.0	1.930e-02	35.8	27.8
88000	4.12e-05	3.25e+00	7.366e-01	21.6	1.801e-02	36.7	23.2
90000	1.14e-05	1.81e+00	7.366e-01	18.8	1.303e-02	38.2	21.9

Table 4.11: PCG tol. $1e-7$; Large; κ-opt **VS** J, ω-opt of M

The experiments show that surprisingly, despite increasing the κ-condition number, the ω-optimal (Jacobi) scaling consistently yields dramatic improvements in PCG convergence. Across all medium and large instances presented in Tables 4.10 and 4.11, PCG requires between 96–98.6% fewer iterations and correspondingly less CPU time after ω-optimal scaling. This improvement is especially pronounced on the large-scale instances presented in Table 4.11.

In summary, our computational experiments in Section 4.3 provide compelling evidence that the ω-condition number is a stronger predictor of iterative solver performance than κ, and that

ω-optimal preconditioners strike a superior balance between computational cost and practical effectiveness.

Part II

EDM Completion

Chapter 5

Local and Global Minimizers of the Smooth Stress Function

5.1 Introduction

EDM (completion) problems have long been studied in the scientific literature, see e.g., the surveys, book collections, and some recent papers [3,5,15,114,119,132]. It is well known that one can obtain the Gram matrix $\bar{G}$ from a given **EDM** $\bar{D}$. Then, a configuration matrix $\bar{P}$ of points $\bar{p}_i \in \mathbb{R}^d$ such that $\bar{D}_{ij} = \|\bar{p}_i - \bar{p}_j\|^2, i, j = 1, \ldots, n$, can be obtained from a full rank factorization

$$\bar{G} = \bar{P}\bar{P}^T, \quad \bar{P}^T = \left[\bar{p}_1, \ldots, \bar{p}_n\right] \in \mathbb{R}^{d\times n}.$$

In this chapter, we consider the question of exact recovery from the unconstrained minimization problem

$$\min_{P\in\mathbb{R}^{n\times d}} \quad \|\mathcal{K}(PP^T) - \bar{D}\|_F^2, \tag{5.1}$$

where $\mathcal{K} : \mathbb{S}^n \to \mathbb{S}^n$ is the Lindenstrauss operator on symmetric matrix space:

$$\mathcal{K}(G) = \operatorname{diag}(G)e^T + e\operatorname{diag}(G)^T - 2G.$$

Here e is the vector of ones and $\|\cdot\|_F$ denote Frobenius norm.

The optimization problem (5.1) is a Euclidean distance geometry problem (DGP), see e.g., [119]. DGP includes the **EDM** Completion Problem, where $\bar{D}$ can have missing entries as well as noisy entries. This latter problem has been proven to be **NP**-hard [153].

The objective function of (5.1) is denoted and expressed as a quartic in P:

$$\sigma_2(P) = \|\mathcal{K}(PP^T) - \bar{D}\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n \left(\|p_i - p_j\|^2 - \|\bar{p}_i - \bar{p}_j\|^2\right)^2,$$

which is referred to as the *smooth stress* in the multidimensional scaling (MDS) literature. (Throughout this chapter, for notational convenience, we use $f(P) = \frac{1}{2}\sigma_2(P)$ as our objective function.)

Since the function $\sigma_2(P)$ is nonconvex, and optimization methods generally find local minima, we investigate the possibility for such a quartic function to have all local minimizers as global

minimizers. This question has been considered as open and has been widely studied in the MDS literature; see for example, [124, 125, 128, 141]. One of the motivations for the research about the local nonglobal minima (**lngm**) of $\sigma_2(P)$ is that the characterization of the **lngm** is critical to developing efficient algorithms without resorting to convex (semidefinite programming) relaxations. This is also closely related to tightness results for low-rank semidefinite programming reformulations, where a central question is whether nonconvex formulations admit spurious local minima or whether all local minima are globally optimal [17, 31, 175].

The question about the existence of a **lngm** was also considered for another type of stress function, called the *raw stress*:

$$\sigma_1(P) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\|p_i - p_j\| - \|\bar{p}_i - \bar{p}_j\|\right)^2 .$$

Trosset and Mathar [128] analytically verified that the raw stress function $\sigma_1(P)$ admits a **lngm**, where $\bar{P}$ is a square configuration with vertices at the four points $\bar{p}_1 = [1/2, 1/2], \bar{p}_2 = [-1/2, 1/2], \bar{p}_3 = [-1/2, -1/2]$, and $\bar{p}_4 = [1/2, -1/2]$. The authors of [174] applied this example to the smooth function $\sigma_2(P)$ but did not find a **lngm**. Instead, they present an example of the *inexact* **EDM** recovery problem having a **lngm**; specifically, problem (5.1) with $\bar{D}$ replaced by $\Delta \in \mathbb{S}^n$ that is *not* an **EDM** . However, the question about the existence of a **lngm** for the exact problem $\sigma_2(P)$ remained open. Addressing this challenge involves two main difficulties: (1) the apparent lack of simple examples exhibiting **lngm**; and (2) the inherent complexity in proving the existence of a **lngm** for such a complex problem.

In this chapter, we provide a definitive answer to this open question. We prove that all second-order stationary points are global minimizers whenever $n \leqslant d + 1$. For $n > d + 1$, we present an example in dimension $d = 1$, with a very special structure, for which we can analytically exhibit a **lngm**. For more general cases, we find examples where the function $\sigma_2(P)$ has a **lngm**, and we provide analytic verification using techniques from the local convergence proof of Newton's method. The examples are obtained using the trust region approach with random initializations.

The rest of this chapter is arranged as follows. We continue in Section 5.2 with a description of our main unconstrained minimization problem (5.1). We introduce two additional equivalent problems with reduced numbers of variables. The reduction allows for strict second-order sufficient optimality conditions and thus is necessary for the analytic existence proof. In Section 5.3, we include various linear transformations, derivatives, and adjoints. Many of these are used throughout this chapter. We suggest that they provide a useful addition to the literature on **EDM**, as it emphasizes the use of matrix transformations rather than individual elements or points. In Section 5.4, we study the optimality conditions and establish that $\sigma_2(P)$ has no **lngm** if $n \leqslant d + 1$. In Section 5.5.1, we give a special example with $d = 1$ where a **lngm** can be explictly illustrated. In Section 5.5.2, we provide two examples and use the Kantorovich theorem to (numerically) prove the existence of **lngm**s in this more general setting.

5.2 Notation and Main Problem Formulations

We follow notations in Section 2.3. For a set of points $p_i \in \mathbb{R}^d,\ i \in [n] := \{1, 2, \cdots, n\}$, denote the *configuration matrix* by

$$P = \begin{bmatrix} p_1 & p_2 & \dots & p_n \end{bmatrix}^T \in \mathbb{R}^{n \times d}.$$

Here d is the embedding dimension. Denote the quadratic mapping $\mathcal{M} : \mathbb{R}^{n\times d} \to \mathbb{S}^n$, $\mathcal{M}(P) = PP^T$. Recall that e denotes the column vector of ones of appropriate dimension. Then, the classical result of Schöenberg [157] relates an **EDM** , $\mathcal{D}(P)$, with the corresponding *Gram matrix,* $G = \mathcal{M}(P)$, by applying the linear operator $\mathcal{K} : \mathcal{S}_C^n \to \mathcal{S}_H^n$:

$$\mathcal{K}(G) = \operatorname{diag}(G)e^T + e\operatorname{diag}(G)^T - 2G = \left(\|p_i - p_j\|^2\right)_{ij} =: \mathcal{D}(P), \tag{5.2}$$

where the *centered subspace,* $\mathcal{S}_C^n$ and the *hollow subspace,* $\mathcal{S}_H^n$ are defined by

$$\mathcal{S}_C^n = \{S \in \mathbb{S}^n \,:\, Se = 0\}, \quad \mathcal{S}_H^n = \{S \in \mathbb{S}^n \,:\, \operatorname{diag}(S) = 0\}.$$

Denote $S_e : \mathbb{R}^n \to \mathbb{S}^n$: $S_e(v) = ve^T + ev^T$. Then, $\mathcal{K}(G) = S_e(\operatorname{diag}(G)) - 2G$. Note that the centered assumption $P^Te = 0 \Leftrightarrow G = PP^T \in \mathcal{S}_C^n$. Also, when the domain of $\mathcal{K}$ is restricted to be $\mathcal{S}_C^n$, the mapping $\mathcal{K}$ is a bijection between $\mathcal{S}_C^n$ and $\mathcal{K}(\mathcal{S}_C^n)$.

Further detailed properties and a list of (non)linear transformations and adjoints are given in Section 5.3.

5.2.1 Main Problem Formulations

Suppose that we are given a centered configuration matrix $\bar{P} \in \mathbb{R}^{n\times d}, \bar{P}^Te = 0$. This gives rise to the corresponding Gram matrix $\bar{G} = \bar{P}\bar{P}^T \in \mathcal{S}_C^n$ and **EDM**, $\bar{D} = \mathcal{K}(\bar{G})$. We now present the main problem and two reformulations that reduce the size of variables and help with stability.

Problem 5.2.1. *Let $\bar{D} = \mathcal{D}(\bar{P})$ be a **EDM** obtained from some given configuration matrix $\bar{P}$. Consider the* nonconvex *minimization problem of recovering a corresponding configuration matrix $\hat{P}$ given by*

$$\hat{P} \in \operatorname*{argmin}_{P\in\mathbb{R}^{n\times d}} f(P) := \frac{1}{2}\|\mathcal{K}(PP^T) - \bar{D}\|_F^2 =: \frac{1}{2}\|F(P)\|_F^2; \tag{5.3}$$

thus defining the function $F : \mathbb{R}^{n\times d} \to \mathcal{S}_H^n$.

Problem 5.2.1 is a nonlinear least squares problem. It has nd variables. By taking advantage of symmetry and the zero-diagonal constraints, the objective function can be seen as a sum of squares of $t(n-1)$ quadratic functions, where $t(n-1) := n(n-1)/2$ is the *triangular number*. Note that $\bar{P}$ is a global minimizer for (5.3) with the optimal value $f(\bar{P}) = 0$. We study whether all stationary points where the second-order necessary optimality conditions hold are global minimizers.

Note that the distance matrix is invariant under translations and rotations of P. Without loss of generality, we assume P is centered ($P^Te = 0$). Let $V \in \mathbb{R}^{n\times(n-1)}$ be such that

$$V^TV = I_{n-1}, \quad V^Te = 0. \tag{5.4}$$

By the fact that VV^T is the orthogonal projection onto $e^\perp$ (the orthogonal complement of e), we have $P^Te = 0$ if, and only if, $P = VL$ for some $L \in \mathbb{R}^{(n-1)\times d}$.[1] We exploit this property for deducing an equivalent problem formulation having a smaller dimension.

Problem 5.2.2. *Let $\bar{P}, \bar{D}$ be as given in Problem 5.2.1, and let V be as in (5.4). Consider the* nonconvex *minimization problem of recovering a corresponding centered configuration matrix*

[1] This is similar to the application of *facial reduction* for the semidefinite relaxation, see [9].

$\hat{P} = V\hat{L}$ *by finding*

$$\hat{L} \in \underset{L\in\mathbb{R}^{(n-1)\times d}}{\operatorname{argmin}} f_L(L) := \frac{1}{2}\|\mathcal{K}(VL(VL)^T) - \bar{D}\|_F^2 =: \frac{1}{2}\|F_L(L)\|_F^2; \tag{5.5}$$

thus defining the function $F_L : \mathbb{R}^{(n-1)\times d} \to \mathcal{S}_H^n$.

Let $\mathcal{O} = \{Q \in \mathbb{R}^{d\times d} : Q^TQ = I_d\}$ be the orthogonal group of order d. Note that $LL^T = LQQ^TL^T$ holds for all $Q \in \mathcal{O}$. If $L^T = QR$ is the QR factorization, then

$$R^T = LQ \Rightarrow f_L(L) = f_L(R^TQ^T) = f_L(R^T),$$

where $R \in \mathbb{R}^{d\times(n-1)}$ is upper triangular (trapezoidal when $d < n-1$). The problem can be further reduced using rotation invariance: $f_L(LQ) = f_L(L)$, $\forall Q \in \mathcal{O}$.

Recall that the linear transformation $\operatorname{svec} : \mathbb{S}^n \to \mathbb{R}^{t(n)}$ is a generalization of the vectorization vec applied to symmetric matrices that avoids the duplication of the lower triangular part. We now extend this idea to triangular (trapezoidal) matrices to avoid the zeros. We define the linear operator that maps a vector $\ell \in \mathbb{R}^{t_\ell}$ to a lower triangular (trapezoidal) matrix in $\mathbb{R}^{(n-1)\times d}$ given by

$$\mathcal{L}\mathrm{Triag}(\ell)_{(i,j)} = \begin{cases} \ell_{nj-n-t(j)+i+1}, & \text{if } j \leqslant i \\ 0, & \text{otherwise,} \end{cases} \tag{5.6}$$

where

$$t_\ell = \begin{cases} t(n-1), & \text{if } d \geqslant n-1 \\ (n-1)d - t(d-1), & \text{otherwise.} \end{cases} \tag{5.7}$$

For $L \in \mathbb{R}^{(n-1)\times d}$ lower trapezoidal, t_ℓ counts its entries where nonzero values are allowed. For $d < n-1$, $\mathcal{L}\mathrm{Triag}(\ell)$ has $t(d-1)$ zero elements at the top right; whereas for $d \geqslant n-1$, $\mathcal{L}\mathrm{Triag}(\ell)$ has $t(n-1)$ nonzero elements at the bottom left.

Using the above definition, we define

$$f_\ell : \mathbb{R}^{t_\ell} \to \mathbb{R}, \quad f_\ell(\ell) := f_L(\mathcal{L}\mathrm{Triag}(\ell)). \tag{5.8}$$

Notice that the adjoint of $\mathcal{L}\mathrm{Triag}$, $\mathcal{L}\mathrm{Triag}^* : \mathbb{R}^{(n-1)\times d} \to \mathbb{R}^{t_\ell}$, takes the lower triangular (trapezoidal) part of $L \in \mathbb{R}^{(n-1)\times d}$ and maps it to the corresponding vector $\ell \in \mathbb{R}^{t_\ell}$, such that $\mathcal{L}\mathrm{Triag}^*\,\mathcal{L}\mathrm{Triag}(\ell) = \ell$. Moreover, $\mathcal{L}\mathrm{Triag}\,\mathcal{L}\mathrm{Triag}^*(L)$ is the projection of L onto the subspace of lower triangular (trapezoidal) matrices.

Problem 5.2.3. *Let* $\bar{P}, \bar{D}$ *be as given in Problem 5.2.2 (and in Problem 5.2.1), and let* $V, t_\ell, f_\ell(\ell)$, *be as in (5.4), (5.7) and (5.8), respectively. Consider the* nonconvex *minimization problem of recovering a corresponding centered configuration matrix* $\hat{P} = V\hat{L} = V\,\mathcal{L}\mathrm{Triag}(\hat{\ell})\hat{Q}^T$, *with* $\hat{Q} \in \mathcal{O}$, *by finding*

$$\begin{aligned} \hat{\ell} \in \operatorname{argmin}_{\ell\in\mathbb{R}^{t_\ell}} f_\ell(\ell) \quad &:= \quad \tfrac{1}{2}\|\mathcal{K}(V\,\mathcal{L}\mathrm{Triag}(\ell)(V\,\mathcal{L}\mathrm{Triag}(\ell))^T) - \bar{D}\|_F^2 \\ &= \tfrac{1}{2}\|F_L(\mathcal{L}\mathrm{Triag}(\ell))\|_F^2 =: \tfrac{1}{2}\|F_\ell(\ell)\|_F^2; \end{aligned} \tag{5.9}$$

thus defining the function $F_\ell : \mathbb{R}^{t_\ell} \to \mathcal{S}_H^n$.

Remark 5.2.4. *Compared to Problem 5.2.1, Problem 5.2.3 is a nonlinear least squares problem, with fewer variables and the same number of quadratic terms* $\left(\mathcal{K}(V\,\mathcal{L}\mathrm{Triag}(\ell)(V\,\mathcal{L}\mathrm{Triag}(\ell))^T) - \bar{D}\right)_{ij}, i < j$. *For* $d < n-1$, *the underlying system of equations is overdetermined, as* $t_\ell < t(n-1)$. *For*

$d \geqslant n-1$, from (5.7), the number of variables is $t(n-1)$, the same as the number of quadratic equations. Thus we no longer have the singularity that arises for the Jacobian of an underdetermined nonlinear least squares problem.

In order to determine whether a **lngm** exists, the next section provides useful formulae for linear transformations and derivatives.

5.3 Properties and auxiliary results

We now provide appropriate notation and formulae for transformations, adjoints and derivatives involved in **EDM**, and then give the equivalence relationships among local minimizers of the above three reformulations.

5.3.1 Transformations, Derivatives, Adjoints, Range and Null Spaces

Lemma 5.3.1 below presents a list of auxiliary results. It concerns the following vectors, matrices and functions:

$$P \in \mathbb{R}^{n\times d},\ p = \mathrm{vec}(P) \in \mathbb{R}^{nd}, \Delta P \in \mathbb{R}^{n\times d},\ \Delta p = \mathrm{vec}(\Delta P) \in \mathbb{R}^{nd},$$
$$L \in \mathbb{R}^{(n-1)\times d},\ \ell \in \mathbb{R}^{t_\ell}, t_\ell \text{ in (5.7)},\ S, T \in \mathbb{S}^n;$$

$$\mathcal{M} : \mathbb{R}^{n\times d} \to \mathbb{S}^n,\ \mathcal{K} : \mathbb{S}^n \to \mathbb{S}^n,\ F : \mathbb{R}^{n\times d} \to \mathcal{S}^n_H,\ f : \mathbb{R}^{n\times d} \to \mathbb{R},$$
$$\mathcal{L}\mathrm{Triag} : \mathbb{R}^{t_\ell} \to \mathbb{R}^{(n-1)\times d},\ S_e : \mathbb{R}^n \to \mathbb{S}^n,\ F_L : \mathbb{R}^{(n-1)\times d} \to \mathcal{S}^n_H,\ f_L : \mathbb{R}^{(n-1)\times d} \to \mathbb{R},$$
$$F_\ell : \mathbb{R}^{t_\ell} \to \mathcal{S}^n_H,\ f_\ell : \mathbb{R}^{t_\ell} \to \mathbb{R}.$$

Lemma 5.3.1. *We have the following first and second Fréchet derivatives and adjoints:*

1. $\mathcal{M}'(P)(\Delta P) = P\Delta P^T + \Delta P P^T$, $\mathcal{M}''(P)(\Delta P, \Delta P) = 2\Delta P \Delta P^T$.
2. $\mathcal{M}'(P)^*(S) = 2SP$.
3. $S_e^*(S) = 2Se$.
4. $\mathcal{K}(G) = S_e(\mathrm{diag}(G)) - 2G$, $\mathrm{range}(\mathcal{K}) = \mathcal{S}^n_H$, $\mathrm{null}(\mathcal{K}) = \mathrm{range}(S_e)$.
5. $\mathcal{K}^*(S) = 2(\mathrm{Diag}(Se) - S)$, $\mathrm{range}(\mathcal{K}^*) = \mathcal{S}^n_C$, $\mathrm{null}(\mathcal{K}^*) = \mathrm{Diag}(\mathbb{R}^n)$. *Moreover,* $S \geqslant (\leqslant) 0 \implies \mathcal{K}^*(S) \geq (\leq) 0$.
6. $\mathcal{D}(P) = S_e(\mathrm{diag}(\mathcal{M}(P))) - 2\mathcal{M}(P)$.
7. $\mathcal{D}'(P)(\Delta P) = S_e(\mathrm{diag}(\mathcal{M}'(P)(\Delta P)) - 2\mathcal{M}'(P)(\Delta P)$.
8. $F'(P)(\Delta P) = \mathcal{K}\left(\mathcal{M}'(P)(\Delta P)\right)$, $F''(P)(\Delta P, \Delta P) = \mathcal{K}\left(\mathcal{M}''(P)(\Delta P, \Delta P)\right)$.
9. $F'(P)^*(S) = \mathcal{M}'(P)^*(\mathcal{K}^*(S)) = 4(\mathrm{Diag}(Se) - S)P$.
10. *We have*
$$f'(P) = F'(P)^*(F(P)) = 4[\mathrm{Diag}(F(P)e) - F(P)]P, \tag{5.10}$$

and

$$\begin{aligned} f''(P)(\Delta P, \Delta P) &= \langle \mathcal{K}(P\Delta P^T + \Delta P P^T), \mathcal{K}(P\Delta P^T + \Delta P P^T)\rangle \\ &\quad +2\langle F(P), \mathcal{K}(\Delta P \Delta P^T)\rangle. \end{aligned} \tag{5.11}$$

Proof. 1. It follows directly from the expansion

$$\begin{aligned} \mathcal{M}(P+\Delta P) &= (P+\Delta P)(P+\Delta P)^T \\ &= PP^T + \Delta P P^T + P\Delta P^T + \Delta P \Delta P^T \\ &= \mathcal{M}(P) + \mathcal{M}'(P)(\Delta P) + \tfrac{1}{2}\mathcal{M}''(P)(\Delta P, \Delta P). \end{aligned}$$

2. Note that

$$\begin{aligned} \langle \mathcal{M}'(P)(\Delta P), S\rangle &= \langle P\Delta P^T + \Delta P P^T, S\rangle \\ &= \operatorname{trace}(P\Delta P^T S + \Delta P P^T S) \\ &= \operatorname{trace}(SP\Delta P^T + SP\Delta P^T) \\ &= \langle \mathcal{M}'(P)^*(S), \Delta P\rangle. \end{aligned}$$

3. From the trace inner product,

$$\langle S_e(v), S\rangle = \operatorname{trace}(ev^T S + ve^T S) = \operatorname{trace}(v^T Se) + \operatorname{trace}(Sev^T) = \langle 2Se, v\rangle.$$

4. See [2, Prop. 2.2].

5. See [2, Prop. 2.2] for the characterization of the nullspace of $\mathcal{K}^*$. The identity $\mathcal{K}^*(S) = 2(\operatorname{Diag}(Se) - S)$ follows from

$$\begin{aligned} \langle \mathcal{K}(T), S\rangle &= \langle \operatorname{diag}(T)e^T + e\operatorname{diag}(T)^T - 2T, S\rangle \\ &= 2\operatorname{trace}(e^T S \operatorname{diag}(T)) - 2\operatorname{trace}(TS) \\ &= 2\langle Se, \operatorname{diag}(T)\rangle - 2\langle T, S\rangle \\ &= 2\langle \operatorname{Diag}(Se) - S, T\rangle, \end{aligned}$$

   where the last equality is due to $\operatorname{Diag} = \operatorname{diag}^*$. Moreover, for $S \in \mathbb{S}^n$, we have by diagonal dominance that $S \succcurlyeq (\preccurlyeq) 0 \implies \mathcal{K}^*(S) \succeq (\preceq) 0$.

6. It follows directly from Item 5.

7. This follows from the linearity of diag and S_e.

8. Both follow from the definitions and linearity of $\mathcal{K}$.

9. It follows from

$$\begin{aligned} \langle F'(P)(\Delta P), S\rangle &= \langle \mathcal{K}(\mathcal{M}'(P)(\Delta P)), S\rangle \\ &= \langle \mathcal{M}'(P)(\Delta P), \mathcal{K}^*(S)\rangle \\ &= \langle \Delta P, \mathcal{M}'(P)^*(\mathcal{K}^*(S))\rangle, \end{aligned}$$

   and $\mathcal{M}'(P)^*$ and $\mathcal{K}^*(S)$ presented in Items 2 and 5.

10. From the expansion of $f(P+\Delta P)$,

$$\begin{aligned} & f(P+\Delta P) \\ = \; & \tfrac{1}{2}\langle F(P+\Delta P), F(P+\Delta P)\rangle \\ = \; & \tfrac{1}{2}\|F(P)+F'(P)(\Delta P)+\tfrac{1}{2}F''(P)(\Delta P,\Delta P)+o(\|\Delta P\|^2)\|^2, \\ = \; & \tfrac{1}{2}\langle F(P),F(P)\rangle+\langle F(P),F'(P)(\Delta P)\rangle \\ & +\tfrac{1}{2}\langle F'(P)(\Delta P),F'(P)(\Delta P)\rangle+\tfrac{1}{2}\langle F(P),F''(P)(\Delta P,\Delta P)\rangle+o(\|\Delta P\|^2), \end{aligned}$$

we get (5.10). Then we obtain

$$\begin{aligned} & f''(P)(\Delta P,\Delta P) \\ = \; & \langle F'(P)(\Delta P),F'(P)(\Delta P)\rangle+\langle F(P),F''(P)(\Delta P,\Delta P)\rangle \\ = \; & \langle \mathcal{K}(\mathcal{M}'(P)(\Delta P)),\mathcal{K}(\mathcal{M}'(P)(\Delta P))\rangle+\langle F(P),\mathcal{K}(\mathcal{M}''(P)(\Delta P,\Delta P))\rangle \\ = \; & \langle \mathcal{K}(P\Delta P^T+\Delta PP^T),\mathcal{K}(P\Delta P^T+\Delta PP^T)\rangle+2\langle F(P),\mathcal{K}(\Delta P\Delta P^T)\rangle, \end{aligned} \tag{5.12}$$

where the second equality follows from Item 8. Define $\Delta p := \mathrm{vec}(\Delta P)$. Now, we can isolate the matrix representation with

$$\begin{aligned} f''(P)(\Delta P,\Delta P) &= \langle f''(P)(\mathrm{Mat}\,\mathrm{vec}(\Delta P)),\mathrm{Mat}\,\mathrm{vec}(\Delta P)\rangle \\ &= \langle [\mathrm{vec}\, f''(P)\,\mathrm{Mat}]\,(\Delta p)\,,(\Delta p)\rangle. \end{aligned}$$

Denote the symmetrization $\mathcal{S}:\mathbb{R}^{n\times n}\to\mathbb{S}^n, \mathcal{S}(K)=(K+K^T)/2$, and let $\mathcal{T}$ be the self-adjoint transpose operator. The first term in (5.12) is

$$\begin{aligned} & 4\langle \mathcal{K}(\mathcal{S}(P(\mathrm{Mat}\,\mathrm{vec}\,\Delta P)^T)),\mathcal{K}(\mathcal{S}(P(\mathrm{Mat}\,\mathrm{vec}\,\Delta P)^T))\rangle \\ = \; & 4\langle (P^T\mathcal{S}^*\mathcal{K}^*\mathcal{K}\mathcal{S}P)((\mathrm{Mat}\,\mathrm{vec}\,\Delta P)^T),(\mathrm{Mat}\,\mathrm{vec}\,\Delta P)^T\rangle \\ = \; & 4\langle (P^T\mathcal{S}^*\mathcal{K}^*\mathcal{K}\mathcal{S}P)(\mathcal{T}\,\mathrm{Mat}\,\mathrm{vec}\,\Delta P),(\mathcal{T}\,\mathrm{Mat}\,\mathrm{vec}\,\Delta P)\rangle \\ = \; & 4\left\langle \left[\mathrm{vec}\,\mathcal{T}^*P^T\mathcal{S}^*\mathcal{K}^*\mathcal{K}\mathcal{S}P\mathcal{T}\,\mathrm{Mat}\right]\Delta p,\Delta p\right\rangle. \end{aligned} \tag{5.13}$$

The second term in (5.12) is

$$\begin{aligned} & 2\langle F(P),\mathcal{K}\left(\Delta P\Delta P^T\right)\rangle \\ = \; & 2\left\langle \mathcal{K}^*\left(F(P)\right),\Delta P\Delta P^T\right\rangle \\ = \; & 2\langle \Delta P,\mathcal{K}^*(F(P))\Delta P\rangle \\ = \; & 2\left\langle [\mathrm{vec}\,\mathcal{K}^*F(P)\,\mathrm{Mat}]\,\Delta p,\Delta p\right\rangle. \end{aligned} \tag{5.14}$$

Recall that $F'(P)(\Delta P)=\mathcal{K}(\mathcal{M}'(P)(\Delta P))$. We combine (5.13) and (5.14) and obtain the matrix representation of the Hessian (not necessarily positive semidefinite) :

$$\begin{aligned} [\mathrm{vec}\, f''(P)\,\mathrm{Mat}] &= 4\left[\mathrm{vec}\,\mathcal{T}^*P^T\mathcal{S}^*\mathcal{K}^*\mathcal{K}\mathcal{S}P\mathcal{T}\,\mathrm{Mat}\right] \\ &\quad +2\left[\mathrm{vec}\,\mathcal{K}^*F(P)\,\mathrm{Mat}\right] \\ &= 4\left[J^*J\right]+2\left[\mathrm{vec}\,\mathcal{K}^*F(P)\,\mathrm{Mat}\right], \end{aligned} \tag{5.15}$$

where

$$J(\Delta p) := \mathcal{K}\mathcal{S}P\mathcal{T}\,\mathrm{Mat}\,\Delta p. \tag{5.16}$$

□

Theorem 5.3.2. *The second-order necessary optimality conditions for (5.3) are:*

$$0 \quad = \quad f'(P) = F'(P)^*(F(P)) = 2\mathcal{K}^*(F(P))P, \tag{5.17}$$

$$0 \quad \preceq \quad [\operatorname{vec} f''(P)\operatorname{Mat}] = 4\,[J^*J] + 2\,[\operatorname{vec}\mathcal{K}^*(F(P))\operatorname{Mat}]\,. \tag{5.18}$$

Proof. The second equality in (5.17) follows from (5.10), and the third equality in (5.17) follows from Items 2 and 9 of Lemma 5.3.1. In particular, we have

$$f'(P) = F'(P)^*(F(P)) = \mathcal{M}'(P)^*(\mathcal{K}^*(F(P))) = 2\mathcal{K}^*(F(P))P. \tag{5.19}$$

The expression for the second-order term in (5.18) follows from (5.15) in Item 10 of Lemma 5.3.1. □

Throughout this chapter, we denote the following two matrices in $\mathbb{S}^{nd}$:

$$H_1 = [J^*J],\ H_2 = [\operatorname{vec}(\mathcal{K}^*(F(P)))\operatorname{Mat}]. \tag{5.20}$$

By abuse of notation, H_1 and H_2 represent both the linear maps and their matrix representations. The meaning will be clear from the context. We call P a *stationary point* if (5.17) holds, and we call P a *second-order stationary point* if both (5.17) and (5.18) hold.

5.3.2 Optimality Conditions of Three Problem Formulations

According to the chain rule, the derivatives and optimality conditions of f_L defined in (5.5) and f_ℓ defined in (5.9) can be easily obtained from that of f.

Proposition 5.3.3. *The derivatives of $f_L(L) : \mathbb{R}^{(n-1)\times d} \to \mathbb{R}$ are*

$$f_L'(L) = V^T f'(VL), \tag{5.21}$$

and

$$f_L''(L) = V^T f''(VL)V. \tag{5.22}$$

Proposition 5.3.4. *The derivatives of $f_\ell(\ell) : \mathbb{R}^{t_\ell} \to \mathbb{R}$ are*

$$f_\ell'(\ell) = \mathcal{L}\mathrm{Triag}^*\, f_L'(\mathcal{L}\mathrm{Triag}(\ell)), \tag{5.23}$$

and

$$f_\ell''(\ell) = \mathcal{L}\mathrm{Triag}^*\, f_L''(\mathcal{L}\mathrm{Triag}(\ell))\,\mathcal{L}\mathrm{Triag}\,. \tag{5.24}$$

In the following, we show that any local minimizer of (5.5) corresponds to a family of local minimizers of (5.3), obtained by translations. Similarly, any local minimizer of (5.9) corresponds to a family of local minimizers of (5.5), derived from rotations.

Proposition 5.3.5. *The configuration matrix $P_* \in \mathbb{R}^{n\times d}$ is a local minimizer of the function f (see (5.3)) if, and only if, all configurations in $\{P_* + ev^T : v \in \mathbb{R}^d\}$ are local minimizers of the function f.*

Proof. We exploit the fact that the function f is invariant w.r.t. translations: for any point P, we have

$$f(P) = f(P + ev^T).$$

If P_* is a local minimizer of f, there must be a $\delta > 0$ such that:

$$\forall P \,:\, \|P_* - P\|_F \leqslant \delta, \quad f(P_*) \leqslant f(P).$$

Then, for all $\hat{P}$ such that $\|\hat{P} - (P_* + ev^T)\|_F \leqslant \delta$, we have $\|(\hat{P} - ev^T) - P_*\|_F \leqslant \delta$, thus

$$f(\hat{P}) = f(\hat{P} - ev^T) \geqslant f(P_*) = f(P_* + ev^T)$$

implying that $P_* + ev^T$ is also a local minimizer. The other implication follows similarly. □

Proposition 5.3.6. *The configuration matrix $L_* \in \mathbb{R}^{(n-1)\times d}$ is a local minimizer of the function f_L (see (5.5)) if, and only if, all configurations in $\{L_*Q : Q \in \mathcal{O}\}$ are local minimizers of f_L.*

Proof. We now exploit the fact that the function f_L is invariant w.r.t. rotations: for any configuration L, and $Q \in \mathcal{O}$, we have

$$f_L(L) = f_L(LQ).$$

If L_* is a local minimizer of f_L, there must be a $\delta > 0$ such that:

$$\forall L \,:\, \|L_* - L\|_F \leqslant \delta, \quad f_L(L_*) \leqslant f_L(L).$$

Then, for all $\hat{L}$ such that $\|\hat{L} - L_*Q\|_F \leqslant \delta$, we have

$$\|\hat{L}Q^T - L_*\|_F = \|\hat{L} - L_*Q\|_F \leqslant \delta,$$

thus

$$f_L(\hat{L}) = f_L(\hat{L}Q^T) \geqslant f_L(L_*) = f_L(L_*Q)$$

implying that L_*Q is also a local minimizer. The other implication follows similarly. □

The local minimizers of the two functions in equations (5.3) and (5.5) have the following relationships.

Theorem 5.3.7. *Let $P_* \in \mathbb{R}^{n\times d}$ and V be as defined in (5.4). Denote*

$$v_* = \frac{1}{n}P_*^T e \in \mathbb{R}^d,\ P_{v_*} = P_* - ev_*^T,\ L_* = V^T P_{v_*}.$$

Then, L_* is a local minimizer of (5.5) if, and only if, P_{v_*} and P_* are local minimizers of (5.3).

Proof. First, recall that VV^T is the orthogonal projection onto $e^\perp$ and that the columns of P_{v_*} are centered. Thus, we have $VL_* = VV^T P_{v_*} = P_{v_*}$. Sufficiency: Let P_{v_*} be a local minimizer of (5.3). Then, there exists $\delta > 0$ such that

$$f(P_{v_*}) \leqslant f(P),\ \forall P \,:\, \|P - P_{v_*}\|_F \leqslant \delta. \tag{5.25}$$

For any $L \in \mathbb{R}^{(n-1)\times p}$ such that $\|L - L_*\|_F \leqslant \delta$, let $\hat{P} = VL$. Then, we have

$$f_L(L_*) = f(VL_*) = f(P_{v_*}) \leqslant f(\hat{P}) = f(VL) = f_L(L),$$

where the inequality is due to $\|\hat{P} - P_{v_*}\|_F = \|VL - VL_*\|_F = \|L - L_*\|_F \leqslant \delta$ and (5.25), and the equalities hold by the definition of f_L.
Necessity: Suppose L_* is a local minimizer of $f_L(L)$, meaning there exists $\delta > 0$ such that

$$f_L(L_*) \leqslant f_L(L), \ \forall L \ : \ \|L - L_*\|_F \leqslant \delta. \tag{5.26}$$

For any configuration P with $\|P - P_{v_*}\|_F \leqslant \delta$, define its centroid $v = P^T e/n$. Then, there exists $L \in \mathbb{R}^{(n-1)\times d}$ such that the centered configuration can be expressed as $P = VL + ev^T$. This implies that $P - P_{v_*} = V(L - L_*) + ev^T$. As $V(L - L_*)$ and ev^T are orthogonal, we get

$$\|L - L_*\|_F^2 = \|V(L - L_*)\|_F^2 = \|P - P_{v_*}\|_F^2 - \|ev^T\|_F^2 \leqslant \delta^2. \tag{5.27}$$

Now, from (5.26) and (5.27), we have

$$f(P) = f(VL + ev^T) = f(VL) \geqslant f(VL_*) = f(P_{v*}),$$

implying that P_{v*} is a local minimizer of $f(P)$. According to Proposition 5.3.5, P_* is also a local minimizer of $f(P)$. □

From Proposition 5.3.6, we know that for the case of $d \geqslant 2$, if L is a local minimizer of $f_L(L)$, then $\{LQ : Q \in \mathcal{O}\}$ is a local minimizer of $f_L(L)$. This means that when $d \geqslant 2$, any local minimizer of $f_L(L)$ is nonisolated and has a singular Hessian matrix.

Next, we consider the correspondence between the local minimizers of (5.5) and its rotation-reduced formulation (5.9).

Theorem 5.3.8. *The following statements hold.*

1. *If L_* is a local minimizer of f_L, then any ℓ_* satisfying*

$$L_* = \mathcal{L}\mathrm{Triag}(\ell_*)Q^T, \tag{5.28}$$

   for some $Q \in \mathcal{O}$, is a local minimizer of f_ℓ.

2. *If ℓ_* is a local minimizer of f_ℓ, and the first d rows of $\mathcal{L}\mathrm{Triag}(\ell_*)$ are linearly independent, then $L_* = \mathcal{L}\mathrm{Triag}(\ell_*)$ is a local minimizer of f_L.*

Proof. 1. Suppose L_* is a local minimizer of f_L, meaning there exists $r > 0$ such that

$$f_L(L) \geqslant f_L(L_*), \ \forall L \ : \ \|L - L_*\|_F \leqslant r. \tag{5.29}$$

For any $\ell \in \mathbb{R}^{t_\ell}$ satisfying $\|\ell - \ell_*\| \leqslant r$, we let $L = \mathcal{L}\mathrm{Triag}(\ell)Q^T$ and we have

$$\|L - L_*\|_F = \|\,\mathcal{L}\mathrm{Triag}(\ell) - \mathcal{L}\mathrm{Triag}(\ell_*)\|_F = \|\ell - \ell_*\| \leqslant r.$$

Then, from (5.28) and (5.29) we have

$$f_\ell(\ell) = f_L(\mathcal{L}\mathrm{Triag}(\ell)) = f_L(L) \geqslant f_L(L_*) = f_L(\mathcal{L}\mathrm{Triag}(\ell_*)) = f_\ell(\ell_*).$$

Therefore, ℓ_* is a local minimizer of f_ℓ.

2. We prove Item 2 by contradiction. Suppose $L_* = \mathcal{L}\text{Triag}(\ell_*)$ is not a local minimizer of f_L. Then there exists a sequence $L_k, k = 1, 2, \ldots$ such that

$$\lim_{k\to+\infty} L_k = L_*,\ f_L(L_k) < f_L(L_*). \tag{5.30}$$

Consider the QR decompositions of $L_k^T, k = 1, 2, \ldots$, i.e., there exist $Q_k \in \mathcal{O}, k = 1, 2, \ldots$, and upper triangular matrices $R_k, k = 1, 2, \ldots$, such that

$$L_k^T = Q_k R_k, k = 1, 2, \ldots. \tag{5.31}$$

Since $\|Q_k\|_2 = 1$ for all $k = 1, 2, \ldots$, the bounded sequence $\{Q_k\}$ has a convergent subsequence. Without loss of generality, we directly assume that $\lim_{k\to+\infty} Q_k = Q_*$. According to (5.30) and (5.31), we have

$$R_*^T := L_* Q_* = \lim_{k\to+\infty} L_k Q_k = \lim_{k\to+\infty} R_k^T. \tag{5.32}$$

By (5.32), R_*^T is a triangular matrix. Since the first d rows of $L_* = \mathcal{L}\text{Triag}(\ell_*)$ are linearly independent, the QR factorization of L_*^T is unique except for signs in each dimension. Thus, Q_* is a diagonal matrix with diagonal elements being -1 or 1. Let

$$\ell_k = \mathcal{L}\text{Triag}^*(R_k^T Q_*^T), k = 1, 2, \ldots.$$

By (5.32), we get

$$\lim_{k\to+\infty} \ell_k = \ell_*.$$

By (5.30), we have

$$\begin{aligned} f_\ell(\ell_k) = f_L(R_k^T) = f_L(L_k) < f_L(L_*) = f_L(R_*^T Q_*^T) \\ = f_L(R_*^T) = f_\ell(\ell_*). \end{aligned}$$

Thus, ℓ_* is not a local minimizer of f_ℓ, a contradiction.

□

The optimality conditions of (5.3) and (5.5) also have an equivalence relationship. To this end, we first note that the directional derivatives of $f(P)$ are zero in any translation.

Lemma 5.3.9. *For any $v \in \mathbb{R}^d$, we have*

$$\langle f'(P), ev^T \rangle = 0 \text{ and } f''(P)(ev^T, ev^T) = 0.$$

Proof. For $t \in \mathbb{R}$, we have

$$f(P + tev^T) \quad = \quad f(P) + t\langle f'(P), ev^T \rangle + \tfrac{t^2}{2}\langle f''(P)(ev^T), ev^T \rangle + o(t^2).$$

Since $f(P + tev^T) = f(P)$ holds for all $v \in \mathbb{R}^d$ and $t \in \mathbb{R}$, we get

$$\langle f'(P), ev^T \rangle = \langle f''(P)(ev^T), ev^T \rangle = 0.$$

Moreover, we claim that

$$f''(P)(ev^T) = 0. \tag{5.33}$$

According to (5.15) and (5.16), we have

$$f''(P)(ev^T) \quad = \quad 4\,[J^*J]\,\mathrm{vec}(ev^T) + 2\,[\mathrm{vec}\,\mathcal{K}^*F(P)\,\mathrm{Mat}]\,\mathrm{vec}(ev^T).$$

Since $\mathrm{null}(\mathcal{K}) = \mathrm{range}(S_e)$ and $\mathrm{range}(\mathcal{K}^*) = \{S \in \mathbb{S}^n : \ Se = 0\}$ in Items 4 and 5 of Lemma 5.3.1, we have

$$J(\mathrm{vec}(ev^T)) = \mathcal{K}\left(\frac{Pve^T + ev^TP^T}{2}\right) = 0, \ \mathcal{K}^*F(P)(ev^T) = 0.$$

Thus, (5.33) holds. □

Theorem 5.3.10. *For $P \in \mathbb{R}^{n\times d}$, denote $v = (P^Te)/n \in \mathbb{R}^d$, $P_v = P - ev^T$, $L = V^TP_v$ where V is defined in (5.4), denote $L^T = QR$ where R is upper triangular and $Q \in \mathbb{R}^{d\times d}$ is orthogonal. Then, the following are equivalent:*

(i) *the first (resp., second)-order necessary conditions of (5.3) hold at P;*

(ii) *the first (resp., second)-order necessary conditions of (5.3) hold at P_v;*

(iii) *the first (resp., second)-order necessary conditions of (5.5) hold at L;*

(iv) *the first (resp., second)-order necessary conditions of (5.5) hold at R^T.*

Proof. Since

$$\begin{aligned} & f(P + t\Delta P) = f(P) + t\langle f'(P), \Delta P\rangle + \frac{t^2}{2}\langle f''(P)(\Delta P), \Delta P\rangle + o(t^2) \\ = \quad & f(P_v + t\Delta P) = f(P_v) + t\langle f'(P_v), \Delta P\rangle + \frac{t^2}{2}\langle f''(P_v)(\Delta P), \Delta P\rangle + o(t^2) \end{aligned}$$

holds for all $\Delta P \in \mathbb{R}^{n\times d}$ and $t \in \mathbb{R}$, we have

$$f'(P_v) = f'(P), \ f''(P_v) = f''(P). \tag{5.34}$$

By

$$\begin{aligned} & f_L(L + t\Delta L) = f_L(L) + t\langle f_L'(L), \Delta L\rangle + \frac{t^2}{2}\langle f_L''(L)(\Delta L), \Delta L\rangle + o(t^2) \\ = \quad & f_L(R^T + t\Delta LQ) = f_L(R^T) + t\langle f_L'(R^T), \Delta LQ\rangle + \frac{t^2}{2}\langle f_L''(R^T)(\Delta LQ), \Delta LQ\rangle + o(t^2), \end{aligned}$$

we have

$$f_L'(R^T) = 0 \Leftrightarrow f_L'(L) = 0, \ f_L''(R^T) \succeq 0 \Leftrightarrow f_L''(L) \succeq 0.^2$$

Thus, (i)⇔(ii) and (iii)⇔(iv).

Now we prove (ii)⇔(iii). First, we prove the equivalence of their first-order necessary conditions. According to (5.19) and $\mathrm{range}(\mathcal{K}^*) = \mathcal{S}_C^n$ (Lemma 5.3.1, Item 5), we have

$$e^Tf'(P) = 2e^T\mathcal{K}^*(F(P))P = 0. \tag{5.35}$$

[2]Note that $f_L''(L)$ is a positive semidefinite linear operator on $\mathbb{R}^{(n-1)\times d}$.

By Proposition 5.3.3, (5.34), (5.35), the definition of V, and Item 5 of Lemma 5.3.1, we obtain

$$f'(P_v) = 0 \iff f'_L(L) = V^T f'(P_v) = 0.$$

Secondly, we prove the equivalence of their second-order necessary optimality conditions. According to (5.22), for any $\Delta L \in \mathbb{R}^{(n-1)\times d}$, we have

$$f''_L(L)(\Delta L, \Delta L) = V^T f''(VL)V(\Delta L, \Delta L) = f''(VL)(V\Delta L, V\Delta L). \tag{5.36}$$

According to (5.33) in Lemma 5.3.9 and (5.36), we have $f''_L(L)(\Delta L, \Delta L) \succcurlyeq 0$ if, and only if,

$$f''(P_v)(\Delta P, \Delta P) = f''(P_v)(V\Delta L + ev^T, V\Delta L + ev^T) \succcurlyeq 0.$$

(Note that we have proved a slightly stronger statement as the semidefinite condition is treated separately from stationarity.) □

Remark 5.3.11. *The reduction from (5.5) to (5.9) may introduce additional stationary points. Let $\mathcal{L}\mathrm{Triag}(\ell_*) = R_*^T$. According to (5.23), $f'_\ell(\ell) = 0$ holds if, and only if, the lower triangular part of $f'_L(R_*^T)$ is zero. Moreover, to have a local minimizer correspondence, the assumption that the first d rows of R_*^T is linear independent in Theorem 5.3.8, Item 2 is needed.*

5.4 Second-Order Optimality Conditions

In this section, we present the optimality conditions and derive a sufficient condition such that there is no **lngm**. First of all, the necessary and sufficient characterization for the global minimizer is clear.

Lemma 5.4.1. *A matrix $P \in \mathbb{R}^{n\times d}$ is a* **global minimizer** *of (5.3) if, and only if, $\mathcal{D}(P) = \bar{D}$.*

Proof. Since $f(P) \geqslant 0$ holds for all $P \in \mathbb{R}^{n\times d}$ and $f(\bar{P}) = 0$, the global minimum of f is 0. By the definition of f and property of norms , $f(P) = 0$ holds if, and only if, $F(P) = \mathcal{D}(P) - \bar{D} = 0$. □

In order to further characterize the second-order optimality conditions, we discuss essential properties of the matrices

$$H_1 = [J^*J] \quad \text{and} \quad H_2 = [\mathrm{vec}\,(\mathcal{K}^*\,(F(P)))\,\mathrm{Mat}].$$

Lemma 5.4.2. *The matrix H_1 defined in (5.20) is always positive semidefinite. For H_2, the following holds:*

- *$H_2 \succeq 0$ when $F(P)$ is element-wise nonnegative,*
- *$H_2 \preceq 0$ when $F(P)$ is element-wise nonpositive.*

Proof. For any $x \in \mathbb{R}^{nd}$,

$$x^T H_1 x = \langle x, J^*Jx\rangle = \langle Jx, Jx\rangle \geqslant 0.$$

Thus, H_1 is always positive semidefinite. By Lemma 5.3.1, Item 5, if $F(P) \succcurlyeq (\preccurlyeq)\ 0$, then $\mathcal{K}^*(F(P)) \succeq (\preceq)\ 0$, which implies

$$x^T H_2 x = \langle x, \mathrm{Mat}^*\, \mathcal{K}^* F(P)\, \mathrm{Mat}\, x\rangle = \langle \mathrm{Mat}\, x, \mathcal{K}^*(F(P))\, \mathrm{Mat}\, x\rangle \succcurlyeq (\preccurlyeq)\ 0$$

for all $x \in \mathbb{R}^{nd}$. Thus, $H_2 \succeq (\preceq)\ 0$ if $F(P) \succcurlyeq (\preccurlyeq)\ 0$. □

Lemma 5.4.3. *The matrix H_2 is the zero matrix if, and only if, $F(P) = 0$ holds, which is equivalent to P being a global minimizer of (5.3).*

Proof. By Lemma 5.3.1, Item 5, $\mathcal{K}^*(S) = 2(\mathrm{Diag}(Se) - S)$ and $\mathrm{null}(\mathcal{K}^*) = \mathrm{Diag}(\mathbb{R}^n)$. Since $\mathrm{diag}(F(P)) = \mathrm{diag}(\mathcal{D}(P)) - \mathrm{diag}(\bar{D}) = 0$ is always true, $\mathcal{K}^*(F(P)) = 0$ holds if, and only if, $F(P) = 0$. □

Lemma 5.4.4. *Let $\bar{P}$, with $\bar{P}^T e = 0$, be a global minimizer of (5.3). Suppose that P is a stationary point for (5.3) but is not a global optimizer. Then H_2 is not positive semidefinite. Specifically, $\mathrm{vec}(\bar{P})^T H_2\, \mathrm{vec}(\bar{P}) < 0$.*

Proof. By (5.17), we have $\langle P, \mathcal{K}^* F(P) P\rangle$=0, and then

$$\begin{array}{rcl} \mathrm{vec}(\bar{P})^T H_2\, \mathrm{vec}(\bar{P}) & = & \langle \bar{P}, \mathcal{K}^* F(P)\bar{P}\rangle - \langle P, \mathcal{K}^* F(P) P\rangle \\ & = & \langle \mathcal{K}(\bar{P}\bar{P}^T), F(P)\rangle - \langle \mathcal{K}(PP^T), F(P)\rangle \\ & = & \langle \mathcal{K}(\bar{P}\bar{P}^T) - \mathcal{K}(PP^T), F(P)\rangle \\ & = & \langle \bar{D} - \mathcal{D}(P), F(P)\rangle \\ & = & -\langle F(P), F(P)\rangle \\ & < & 0. \end{array}$$

The last inequality holds because P is not a global minimizer, which implies $F(P) \neq 0$ according to Lemma 5.4.1. □

Under the condition of Lemma 5.4.4, we have known that

$$\langle \bar{P}, \mathcal{K}^* F(P)\bar{P}\rangle < 0,$$

which implies that

$$\mathcal{K}^* F(P) \not\succeq 0. \tag{5.37}$$

We analyze the extreme case of $\bar{D} = 0$.

Corollary 5.4.5. *If $\bar{D} = 0$, then every stationary point is a global minimizer.*

Proof. Suppose P is a stationary point. As $\bar{D} = 0$, we get $\bar{p}_1 = \ldots = \bar{p}_n$ holds. Since $\mathcal{K}^* F(P)$ is a Laplacian (sum of its columns is zero), we have $\mathcal{K}^* F(P)\bar{P} = 0$. Combining this with the first-order condition (5.17), we have

$$\begin{array}{rcl} \mathrm{vec}(\bar{P})^T H_2\, \mathrm{vec}(\bar{P}) & = & \langle \bar{P}, \mathcal{K}^*(F(P))\bar{P}\rangle - \langle P, \mathcal{K}^*(F(P))P\rangle \\ & = & 0. \end{array}$$

From Lemma 5.4.4, we conclude that any stationary point P for $f(P)$ is a global minimizer. □

Next, we consider another extreme case of $\bar{D} \neq 0, \mathcal{D}(P) = 0$.

Theorem 5.4.6. *For $\bar{D} \neq 0$ and P with $\mathcal{D}(P) = 0$ (i.e., all $p_i \equiv p$), P is a stationary point and has nontrivial negative semidefinite Hessian:*

$$0 \neq 4H_1 + 2H_2 \preceq 0.$$

Proof. Since $p_1 = \ldots = p_n = p$ and $\mathcal{K}^*(F(P))$ is a Laplacian, P satisfies the first-order optimality condition (5.17). Since $f(P) = \|\bar{D} - \mathcal{D}(P)\|_F^2 = \|\bar{D}\|_F^2 > 0$, P is not a global minimizer. By

$$P\Delta P^T + \Delta P P^T = ep^T \Delta P^T + \Delta P p e^T = e(\Delta P p)^T + (\Delta P p)e^T,$$

and $\mathrm{null}(\mathcal{K}) = \mathrm{range}(S_e)$ (Lemma 5.3.1, Item 4), $J = 0$ defined in (5.16) holds, and then $H_1 = 0$. Since $\mathcal{D}(P) = 0$ and $\bar{D} \geqslant 0$, $F(P) \leqslant 0$ holds. According to Lemma 5.4.2 and Lemma 5.4.3, $0 \neq H_2 \preceq 0$ holds. Therefore, the Hessian matrix satisfies $0 \neq 4H_1 + 2H_2 \preceq 0$. □

Remark 5.4.7. *If P is a local* **maximizer** *of f, then necessarily $\mathcal{D}(P) = 0$ ($p_1 = \cdots = p_n$). To see this, observe that $t = 1$ must locally maximize $g(t) = f(tP)$. By the second-order necessary conditions, we have $g'(1) = 0$ and $g''(1) \leqslant 0$. Since $g'(t) = 4t^3\|\mathcal{D}(P)\|_F^2 - 4t\langle \bar{D}, \mathcal{D}(P)\rangle$, $0 = g'(1)$ implies that $\|\mathcal{D}(P)\|_F^2 = \langle \bar{D}, \mathcal{D}(P)\rangle$. Then, $0 \geqslant g''(1) = 8\|\mathcal{D}(P)\|_F^2$ and we conclude that $\mathcal{D}(P) = 0$.*

In the following, we present the condition under which there is no **lngm**. Recall the equivalence between local minimizers of (5.5) and (5.3) in Theorem 5.3.7, we analyze (5.5) for convenience.

Theorem 5.4.8. *Any stationary point L of (5.5) satisfying $\mathrm{rank}(L) = n-1$ is a global minimizer.*

Proof. Since $0 = f_L'(L) = 2V^T\mathcal{K}^*F(VL)VL$, where the last equality follows from (5.21) and (5.19), the span of columns of L is an $n-1$ dimensional eigenvector space corresponding to the zero eigenvalue of the $(n-1)\times(n-1)$ matrix $V^T\mathcal{K}^*F(VL)V$. Therefore $V^T\mathcal{K}^*(F(P))V = 0$. Combining this with $\mathrm{range}(\mathcal{K}^*) = \mathcal{S}_C^n$ from Lemma 5.3.1, Item 5, we conclude that $\mathcal{K}^*(F(P)) = 0$, and then $H_2 = 0$. Thus, L is a global minimizer according to Lemma 5.4.3. □

As $L \in \mathbb{R}^{(n-1)\times d}$, the condition in Theorem 5.4.8 holds in the case that $d \geqslant n-1$ and L is of full row rank. Next, we consider another case where L is not full column rank.

Theorem 5.4.9. *Suppose that L is a non-globally-optimal stationary point of (5.5) and*

$$\mathrm{rank}(L) < d. \tag{5.38}$$

Then, the second-order necessary optimality conditions fail at L.

Proof. Denote $P = VL$. According to Lemma 5.4.4 and the subsequent discussion, (5.37) holds. Thus there exists $a \in \mathbb{R}^n$ such that $a^T\mathcal{K}^*F(P)a < 0$. Then, for any nonzero $w \in \mathbb{R}^d$,

$$\begin{aligned}
\mathrm{vec}(aw^T)^T H_2\,\mathrm{vec}(aw^T) &= \langle aw^T, \mathcal{K}^*F(P)(aw^T)\rangle \\
&= \mathrm{tr}(\mathcal{K}^*F(P)(aw^T)(aw^T)^T) \\
&= w^Tw\,\mathrm{tr}(\mathcal{K}^*F(P)aa^T) \\
&= w^Twa^T\mathcal{K}^*F(P)a \\
&< 0.
\end{aligned}$$

By (5.38), there exists a nonzero $w \in \mathbb{R}^d$ such that $w \in \text{null}(L)$, meaning,

$$Lw = 0. \tag{5.39}$$

We claim that $H_1 \operatorname{vec}(aw^T) = 0$ holds. First, we have

$$J \operatorname{vec}(aw^T) = \mathcal{KSV}L\mathcal{T}(aw^T).$$

By (5.39), we have

$$L\mathcal{T}(aw^T) = L\begin{bmatrix} a_1 w & a_2 w & \dots & a_{n-1} w \end{bmatrix} = 0.$$

Thus, $H_1 \operatorname{vec}(aw^T) = 0$ holds. In sum, we have

$$\operatorname{vec}(aw^T)^T (4H_1 + 2H_2) \operatorname{vec}(aw^T) < 0,$$

which implies the second-order necessary optimality condition (5.18) fails. □

Combining Theorem 5.4.8 and Theorem 5.4.9, we present the main result of this section.

Theorem 5.4.10. *If $n \leqslant d + 1$, then any stationary point satisfying the second-order necessary optimality conditions is a global minimizer.*

Proof. Suppose that $n \leqslant d + 1$, and L is a stationary point satisfying the second-order necessary optimality condition. If $\operatorname{rank}(L) = n-1$, then L is globally optimal by Theorem 5.4.8. If $\operatorname{rank}(L) < n - 1$, then $\operatorname{rank}(L) < d$. If we assume L is not a global minimizer, according to Theorem 5.4.9, L does not satisfy the second-order necessary optimality condition, a contradiction. □

Recalling Remark 5.2.4, we note that $n \leqslant d+1$ is exactly the condition such that the underlying system of equations is square. When $n > d+1$ (overdetermined), it is possible to find local nonglobal minimizers.

5.5 lngms Examples

We now provide instances with **lngm**s. The data and codes are available at GitHub repository.

We first provide an analytical example with $d = 1$ with a specific simple structure for the points in $\mathbb{R}^d$, see Section 5.5.1. Then in Section 5.5.2 we give two examples where we numerically obtain approximate second-order stationary points. Then, we analytically prove that the assumptions of the Kantorovich theorem hold at these two points, i.e., this implies that there exists **lngm**s in the neighborhoods. We consider the sensitivity analysis needed to analytically prove that our examples have **lngm**s. We exploit the strength of the classical Kantorovich theorem for the convergence of Newton's Method to exact stationary points when using the approximate stationary points that we found as starting points. See Theorem 5.5.6 and Corollary 5.5.10, below.

5.5.1 An Explicit Example with a lngm with $d = 1$

We now present a simple explicit example with an **lngm**, where we can analytically verify the **lngm** .

Example 5.5.1. *Let*

$$d = 1, n > 6; \quad \bar{P}^T = [\bar{p}_1, \cdots, \bar{p}_n],\ \tilde{P}^T = [\tilde{p}_1, \cdots, \tilde{p}_n] \in \mathbb{R}^{d\times n},$$

with

$$\bar{p}_1 = 2,\ \bar{p}_2 = 0,\ \bar{p}_3 = \cdots = \bar{p}_n = 1 \quad \textit{and} \quad \tilde{p}_1 = \tilde{p}_2 = 0,\ \tilde{p}_3 = \cdots = \tilde{p}_n = 1.$$

We now continue and illustrate that $\tilde{P}$ is a **lngm** of (5.3) in Problem 5.2.1. Let $E_{2,n-2}$ be the $2 \times n-2$ matrix of all ones, and $[0]$ denote the matrix of zeros of appropriate size. From the definitions of $F(\cdot)$, $\mathcal{K}(\cdot)$ and $\mathcal{K}^*(\cdot)$, we have:

$$\begin{aligned} F(\tilde{P}) &= \mathcal{D}(\tilde{P}) - \mathcal{D}(\bar{P}) \\ &= \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & E_{2,n-2} \\ E_{2,n-2}^T & [0] \end{bmatrix} - \begin{bmatrix} \begin{bmatrix} 0 & 4 \\ 4 & 0 \end{bmatrix} & E_{2,n-2} \\ E_{2,n-2}^T & [0] \end{bmatrix} \qquad (5.40) \\ &= \begin{bmatrix} \begin{bmatrix} 0 & -4 \\ -4 & 0 \end{bmatrix} & 0_{2,n-2} \\ 0_{2,n-2}^T & [0] \end{bmatrix}; \qquad (5.41) \end{aligned}$$

$$\mathcal{K}^*(F(\tilde{P})) = 2 \begin{bmatrix} \begin{bmatrix} -4 & 4 \\ 4 & -4 \end{bmatrix} & [0] \\ [0] & [0] \end{bmatrix}.$$

Moreover, we have

$$\begin{aligned} & \langle \mathcal{K}(\tilde{P}\Delta P^T + \Delta P \tilde{P}^T), \mathcal{K}(\tilde{P}\Delta P^T + \Delta P \tilde{P}^T) \rangle \\ = \ & 4 \sum_{i\neq j} [(\tilde{p}_i - \tilde{p}_j)(\Delta p_i - \Delta p_j)]^2 \\ = \ & 8\Delta P^T \begin{bmatrix} \sum_{i=1}^n (\tilde{p}_1 - \tilde{p}_i)^2 & -(\tilde{p}_1 - \tilde{p}_2)^2 & \cdots & -(\tilde{p}_1 - \tilde{p}_n)^2 \\ -(\tilde{p}_1 - \tilde{p}_2)^2 & \sum_{i=1}^n (\tilde{p}_2 - \tilde{p}_i)^2 & \cdots & -(\tilde{p}_2 - \tilde{p}_n)^2 \\ \vdots & \vdots & \cdots & \vdots \\ -(\tilde{p}_1 - \tilde{p}_2)^2 & -(\tilde{p}_2 - \tilde{p}_n)^2 & \cdots & \sum_{i=1}^n (\tilde{p}_n - \tilde{p}_i)^2 \end{bmatrix} \Delta P \\ = \ & 8\Delta P^T \begin{bmatrix} (n-2)I & -E_{2,n-2} \\ -E_{2,n-2}^T & 2I \end{bmatrix} \Delta P. \end{aligned}$$

Then we can compute from (5.10) and (5.11) that the derivative (gradient) is

$$f'(\tilde{P}) = 4[\mathrm{Diag}(F(\tilde{P})e) - F(\tilde{P})]\tilde{P} = 0. \qquad (5.42)$$

And the Hessian quadratic form is

$$\begin{aligned} & f''(\tilde{P})(\Delta P, \Delta P) \\ = \ & \langle \mathcal{K}(\tilde{P}\Delta P^T + \Delta P \tilde{P}^T), \mathcal{K}(\tilde{P}\Delta P^T + \Delta P \tilde{P}^T) \rangle + 2\langle F(\tilde{P}), \mathcal{K}(\Delta P \Delta P^T) \rangle \\ = \ & 8\Delta P^T \left(\begin{bmatrix} (n-2)I & -E_{2,n-2} \\ -E_{2,n-2}^T & 2I \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix} & [0] \\ [0] & [0] \end{bmatrix} \right) \Delta P. \end{aligned}$$

The Hessian $f''(\tilde{P}) = \nabla^2 f(\tilde{P})$ is a rank 3 update of $16I$. It is positive semidefinite with nullspace span(e) if, and only if, $n \geqslant 7$. (It is indefinite when $n < 6$.) Thus, $\tilde{P}$ is a second-order stationary point of the problem with data given above. Next, we prove that $\tilde{P}$ is a **lngm** by considering the reduced formulation (5.5).

To center as done in Theorem 5.3.7, we let

$$\tilde{v} = \frac{\tilde{P}^T e}{n} \in \mathbb{R}, \ \tilde{P}_* = \tilde{P} - e\tilde{v}^T, \ \tilde{L} = V^T \tilde{P}_*,$$

to obtain $\tilde{L}$. According to Theorem 5.3.10, $\tilde{L}$ is a second-order stationary point of the function $f_L(L)$. By Proposition 5.3.3, we have $f_L''(\tilde{L}) = V^T f''(V\tilde{L})V = V^T f''(\tilde{P})V$. Since $f''(\tilde{P})$ has a one-dimensional nullspace (span$\{e\}$), its restriction to span$\{e\}^\perp$ is positive definite. Thus, $f_L''(\tilde{L})$ is positive definite. This implies that $\tilde{L}$ satisfies the second-order sufficient optimality conditions for a strict local minimum of $f_L(L)$. By Theorem 5.3.7 and Lemma 5.4.1, $\tilde{P}$ is in fact a **lngm** of $f(P)$. Note that the objective function value $f(\tilde{P}) = 16 > 0 = f(\bar{P})$, thus confirming that $\tilde{P}$ is not a global minimum.

5.5.2 Examples via Kantorovich Theorem and Sensitivity Analysis

We now present Examples 5.5.2 and 5.5.7, with $d = 1, 2$, respectively, where we first find an approximate second-order stationary point $\tilde{L}$ numerically that has a sufficiently large (positive) objective value; and then we prove that there is a ***lngm*** *nearby* using the Kantorovich theorem and sensitivity analysis.

Case $d = 1$

In this case we analyze a configuration matrix $\tilde{P} = V\tilde{L} \in \mathbb{R}^{n\times 1}, \tilde{L} \in \mathbb{R}^{(n-1)\times 1}$, satisfying:

$$\begin{array}{rlll} \text{Objective value}: & f(\tilde{P}) = f_L(\tilde{L}) > \tilde{f}_L, & \text{for some (large) } \tilde{f}_L > 0; \\ \text{Near stationarity}: & \|\nabla f_L(\tilde{L})\| < \tilde{g}_L, & \text{for some (small) } \tilde{g}_L > 0; \\ \text{Local Convexity}: & \lambda_{\min}(\nabla^2 f_L(\tilde{L})) > \tilde{\lambda}_L, & \text{for some (large) } \tilde{\lambda}_L > 0. \end{array} \tag{5.43}$$

While theoretically exact, the floating-point representations introduce round-off errors in finite-precision computations. Our MATLAB implementation performs complete finite-precision arithmetic analysis in order to compute the rigorous bounds $\tilde{f}_L$, $\tilde{g}_L$, and $\tilde{\lambda}_L$ (see Footnote 3).

We apply the classical Kantorovich theorem, e.g., [54, Thm 5.3.1], to show that there is a point *nearby* that satisfies: (i) it is an *exact* stationary point; (ii) the function value is positive; and (iii) the Hessian is still positive definite. This provides an analytic proof that we have a theoretically verified **lngm** near $\tilde{L}$.

Example 5.5.2. *An example with $n = 50, d = 1$ is given, with data $\bar{P} = V\bar{L}$, $\tilde{P} = V\tilde{L} \in \mathbb{R}^{n\times d}$. (See the footnote below.[3]) Matrix $\bar{D}$ is the distance matrix obtained from $\bar{L}$ by $\bar{D} = \mathcal{K}(V\bar{L}(V\bar{L})^T)$.*

[3] The data and codes are available at https://github.com/MengmengSong97/EDM-code

Thus, $\bar{L}$ is a global minimizer. $\tilde{L}$ is a numerically convergence point obtained by a trust region method with random initialization, where the objective value is

$$f_L(\tilde{L}) > 2.65 \times 10^3, \tag{5.44}$$

the absolute and relative gradient norms are

$$\|\nabla f_L(\tilde{L})\| < 10^{-7}, \ \frac{\|\nabla f_L(\tilde{L})\|}{1 + f_L(\tilde{L})} < 3.53 \times 10^{-18}, \tag{5.45}$$

and the least eigenvalue of the Hessian matrix is

$$2.12 \times 10^2 > \lambda_{\min}(\nabla^2 f_L(\tilde{L})) > 2.10 \times 10^2. \tag{5.46}$$

In the following, we will verify that $f_L(L)$ has a **lngm**.

Remark 5.5.3. *The problem to find a **lngm** is a* nonlinear least squares *problem. The standard approach for nonlinear least squares is to use the Gauss-Newton method, which simplifies the Hessian $4H_1 + 2H_2$ (see (5.15) and (5.20)) by keeping only $4H_1$ and omitting $2H_2$ (same for L, cf. (5.22)). This approximation relies on the assumption that $f_L(L)$ is near zero, a condition we intentionally avoid, as this would yield the global minimum.*

Now, we find an estimate for the Lipschitz constant $\gamma > 0$ for the Hessian matrix of f_L. From our numerical output, we know that the smallest eigenvalue $\lambda_{\min}(\nabla^2 f_L(\tilde{L})) > 0$. By continuity of eigenvalues, we are guaranteed that this holds in a neighbourhood of $\tilde{L}$, which is now estimated in Proposition 5.5.4.

Proposition 5.5.4. *Let $r > 0$ and $\tilde{L} \in \mathbb{R}^{(n-1)\times d}$ be given. If*

$$\gamma \geqslant 24\sqrt{2}\left(\sum_{i,j} \|(V\tilde{L})[i,:] - (V\tilde{L})[j,:]\|_F + 2n^{3/2}r\right), \tag{5.47}$$

then γ is a Lipschitz constant for the Hessian of f_L in the radius-r neighborhood of $\tilde{L}$, i.e.,

$$\|\nabla^2 f_L(\hat{L}) - \nabla^2 f_L(\check{L})\|_2 \leqslant \gamma \|\hat{L} - \check{L}\|_F, \quad \text{for all } \hat{L}, \check{L} \in B_r(\tilde{L}). \tag{5.48}$$

Moreover,

$$\lambda_{\min}(\nabla^2 f_L(L)) \geqslant \lambda_{\min}(\nabla^2 f_L(\tilde{L})) - \gamma r, \quad \text{for all } L \in B_r(\tilde{L}). \tag{5.49}$$

Proof. By the definition of the induced norm, (5.48) is equivalent to

$$|f_L''(\hat{L})(\Delta L, \Delta L) - f_L''(\check{L})(\Delta L, \Delta L)| \leqslant \gamma \|\hat{L} - \check{L}\|, \tag{5.50}$$

for all $\hat{L}, \check{L} \in B_r(\tilde{L}), \|\Delta L\| = 1$. Let

$$\hat{P} = V\hat{L}, \check{P} = V\check{L}, \Delta P = V\Delta L, \tilde{P} = V\tilde{L}.$$

According to (5.12) and Proposition 5.3.3, we have

$$\begin{aligned} f''_L(\hat{L})(\Delta L, \Delta L) &= f''(\hat{P})(\Delta P, \Delta P) \\ &= \|\mathcal{K}(\hat{P}\Delta P^T + \Delta P \hat{P}^T)\|_F^2 + 2\langle F(\hat{P}), \mathcal{K}(\Delta P \Delta P^T)\rangle \\ &= \textstyle\sum_{i,j}(2\hat{p}_i^T\Delta p_i + 2\hat{p}_j^T\Delta p_j - 2\hat{p}_i^T\Delta p_j - 2\hat{p}_j^T\Delta p_i)^2 \\ &\quad +2\textstyle\sum_{i,j}\|\hat{p}_i - \hat{p}_j\|^2\|\Delta p_i - \Delta p_j\|^2 \\ &= 4\textstyle\sum_{i,j}[(\hat{p}_i - \hat{p}_j)^T(\Delta p_i - \Delta p_j)]^2 + 2\textstyle\sum_{i,j}\|\hat{p}_i - \hat{p}_j\|^2\|\Delta p_i - \Delta p_j\|^2. \end{aligned}$$

The calculations about $\check{L}$ are similar, implying

$$\begin{aligned} & f''_L(\hat{L})(\Delta L, \Delta L) - f''_L(\check{L})(\Delta L, \Delta L) \\ = \;& 4\textstyle\sum_{i,j}[(\hat{p}_i - \hat{p}_j)^T(\Delta p_i - \Delta p_j)]^2 - [(\check{p}_i - \check{p}_j)^T(\Delta p_i - \Delta p_j)]^2 \\ & +2\textstyle\sum_{i,j}(\|\hat{p}_i - \hat{p}_j\|^2 - \|\check{p}_i - \check{p}_j\|^2)\|\Delta p_i - \Delta p_j\|^2 \\ = \;& 4\textstyle\sum_{i,j}(\hat{p}_i - \hat{p}_j - \check{p}_i + \check{p}_j)^T(\Delta p_i - \Delta p_j)(\hat{p}_i - \hat{p}_j + \check{p}_i - \check{p}_j)^T(\Delta p_i - \Delta p_j) \\ & +2\textstyle\sum_{i,j}(\hat{p}_i - \hat{p}_j - \check{p}_i + \check{p}_j)^T(\hat{p}_i - \hat{p}_j + \check{p}_i - \check{p}_j)\|\Delta p_i - \Delta p_j\|^2. \end{aligned}$$

Then,

$$\begin{aligned} & |f''_L(\hat{L})(\Delta L, \Delta L) - f''_L(\check{L})(\Delta L, \Delta L)| \\ \leqslant \;& 6\textstyle\sum_{i,j}\|\hat{p}_i - \hat{p}_j - \check{p}_i + \check{p}_j\|\|\hat{p}_i - \hat{p}_j + \check{p}_i - \check{p}_j\|\|\Delta p_i - \Delta p_j\|^2. \end{aligned}$$

Since $\|\Delta P\|_F = \|V\Delta L\|_F = \|\Delta L\|_F = 1$,

$$\|\Delta p_i - \Delta p_j\|^2 \leqslant 2(\|\Delta p_i\|^2 + \|\Delta p_j\|^2) \leqslant 2.$$

From $\hat{L}, \check{L} \in B_r(\tilde{L})$, it follows that $\hat{P}, \check{P} \in B_r(\tilde{P})$. Then, we have

$$\begin{aligned} \|\hat{p}_i - \hat{p}_j - \check{p}_i + \check{p}_j\| &\leqslant \|\hat{p}_i - \check{p}_i\| + \|\hat{p}_j - \check{p}_j\| \\ &\leqslant \sqrt{2}\sqrt{\|\hat{p}_i - \check{p}_i\|^2 + \|\hat{p}_j - \check{p}_j\|^2} \\ &\leqslant \sqrt{2}\|\hat{L} - \check{L}\|_F, \end{aligned}$$

where the first inequality follows from the triangle inequality, the second from the Cauchy-Schwarz inequality, and the third from the fact that

$$\|\hat{P} - \check{P}\|_F = \|V\hat{L} - V\check{L}\|_F = \|\hat{L} - \check{L}\|_F.$$

We also have

$$\begin{aligned} & \textstyle\sum_{i,j}\|\hat{p}_i - \hat{p}_j + \check{p}_i - \check{p}_j\| \\ = \;& \textstyle\sum_{i,j}\|2(\tilde{p}_i - \tilde{p}_j) + (\hat{p}_i - \tilde{p}_i) - (\hat{p}_j - \tilde{p}_j) + (\check{p}_i - \tilde{p}_i) - (\check{p}_j - \tilde{p}_j)\| \\ \leqslant \;& 2\textstyle\sum_{i,j}\|\tilde{p}_i - \tilde{p}_j\| + \textstyle\sum_{i,j}\|\hat{p}_i - \tilde{p}_i\| + \textstyle\sum_{i,j}\|\hat{p}_j - \tilde{p}_j\| + \\ & \textstyle\sum_{i,j}\|\check{p}_i - \tilde{p}_i\| + \textstyle\sum_{i,j}\|\check{p}_j - \tilde{p}_j\| \\ = \;& 2\textstyle\sum_{i,j}\|\tilde{p}_i - \tilde{p}_j\| + n\textstyle\sum_i\|\hat{p}_i - \tilde{p}_i\| + n\textstyle\sum_j\|\hat{p}_j - \tilde{p}_j\| + \\ & n\textstyle\sum_i\|\check{p}_i - \tilde{p}_i\| + n\textstyle\sum_j\|\check{p}_j - \tilde{p}_j\| \\ \leqslant \;& 2\textstyle\sum_{i,j}\|\tilde{p}_i - \tilde{p}_j\| + 4n^{3/2}r, \end{aligned}$$

where the last inequality follows from the Hölder inequality. Thus,

$$\begin{aligned} |f''_L(\hat{L})(\Delta L, \Delta L) - f''_L(\check{L})(\Delta L, \Delta L)| &\leqslant 12\sqrt{2}\|\hat{L} - \check{L}\|_F\textstyle\sum_{i,j}\|\hat{p}_i - \hat{p}_j + \check{p}_i - \check{p}_j\| \\ &\leqslant 24\sqrt{2}\|\hat{L} - \check{L}\|_F\left(\textstyle\sum_{i,j}\|\tilde{p}_i - \tilde{p}_j\| + 2n^{3/2}r\right), \end{aligned}$$

applying (5.47) turns out (5.48). By (5.50), we have

$$\begin{aligned} f_L''(L)(\Delta L, \Delta L) = \quad & f_L''(\tilde{L})(\Delta L, \Delta L) - (f_L''(\tilde{L})(\Delta L, \Delta L) - f_L''(L)(\Delta L, \Delta L)) \\ \geqslant \quad & f_L''(\tilde{L})(\Delta L, \Delta L) - |f_L''(L)(\Delta L, \Delta L) - f_L''(\tilde{L})(\Delta L, \Delta L)| \\ \geqslant \quad & \lambda_{\min}(\nabla^2 f_L(\tilde{L})) - \gamma \|\hat{L} - \tilde{L}\|_F \\ \geqslant \quad & \lambda_{\min}(\nabla^2 f_L(\tilde{L})) - \gamma r, \quad \text{for all } L \in B_r(\tilde{L}), \|\Delta L\|_F = 1. \end{aligned}$$

Thus, we obtain (5.49). □

To verify the existence of a **lngm** for Example 5.5.2, we calculate the Lipschitz constant estimated in Proposition 5.5.4. Let $r = 10^{-3}$. Since

$$\sum_{i,j} \|(V\tilde{L})[i,:] - (V\tilde{L})[j,:]\| < 2.13 \times 10^3,$$

(5.47) gives

$$\gamma = 7.24 \times 10^4.$$

Moreover, by (5.49), we have

$$\lambda_{\min}(\nabla^2 f_L(L)) \geqslant 211 - 7.24 \times 10^4 \times r = 138.6 > 0, \quad \text{for all } L \in B_r(\tilde{L}). \tag{5.51}$$

That is, we find a neighbourhood where the Hessian stays positive semidefinite. Next, we prove that the objective stays sufficiently positive in a region around $\tilde{L}$.

Lemma 5.5.5. *Let the configuration $\tilde{P} = V\tilde{L} \in \mathbb{R}^{(n-1)\times d}, \tilde{L} \in \mathbb{R}^{(n-1)\times d}$ and positive parameters $\bar{f}_L, r \in \mathbb{R}_{++}$, be given. Suppose that $f_L(\tilde{L}) > \bar{f}_L$ and that the Hessian $\nabla^2 f_L$ is uniformly positive definite in the r-ball around $\tilde{L}$, i.e.,*

$$\lambda_{\min}(\nabla^2 f_L(L)) > 0, \quad \textit{for all } L \in B_r(\tilde{L}). \tag{5.52}$$

Then f_L is positively uniformly bounded from below in $B_r(\tilde{L})$, i.e.,

$$f_L(L) > \bar{f}_L > 0, \quad \textit{for all } \|L - \tilde{L}\|_F \leqslant \min\left\{ r, \frac{f_L(\tilde{L}) - \bar{f}_L}{\|\nabla f_L(\tilde{L})\|_F} \right\}.$$

Proof. By the positive definiteness assumption of the Hessian in the r-ball $B_r(\tilde{L})$, we can apply convexity of f_L in the ball. Therefore, for all $L \in \mathbb{R}^{(n-1)\times d}$ such that $\|L - \tilde{L}\|_F \leqslant \min\left\{r, (f_L(\tilde{L}) - \bar{f}_L)/\|\nabla f_L(\tilde{L})\|_F\right\}$, we have

$$\begin{aligned} f_L(L) \quad & \geqslant \quad f_L(\tilde{L}) + \langle \nabla f_L(\tilde{L}), L - \tilde{L} \rangle \\ & \geqslant \quad f_L(\tilde{L}) - \|\nabla f_L(\tilde{L})\|_F \|L - \tilde{L}\|_F \\ & > \quad \bar{f}_L > 0. \end{aligned}$$

□

By (5.44), (5.45) and considering $\bar{f}_L = 10^3$, we get

$$\frac{f_L(\tilde{L}) - \bar{f}_L}{\|\nabla f_L(\tilde{L})\|} > \frac{2.65 \times 10^3 - 10^3}{10^{-7}} > r.$$

According to Lemma 5.5.5 with (5.51) we conclude that

$$f_L(L) > \bar{f}_L > 0, \quad \text{for all } L \in B_r(\tilde{L}). \tag{5.53}$$

We now apply the classical Kantorovich theorem to prove the existence of a unique **lngm** point within a certain neighborhood. We reword the version in [54, Thm 5.3.1].

Theorem 5.5.6. *Let the configuration matrix $\tilde{P} = V\tilde{L} \in \mathbb{R}^{n\times d}$, $\tilde{L} \in \mathbb{R}^{(n-1)\times d}$ be given. Let $r \in \mathbb{R}_{++}$ be found such that*

$$\nabla^2 f_L(L) \succ 0, \quad \textit{for all } L \in B_r(\tilde{L}),$$

and $\bar{f}_L$ satisfying

$$f_L(L) > \bar{f}_L > 0, \quad \textit{for all } L \in B_r(\tilde{L}).$$

Let γ be a Lipschitz constant for the Hessian of f_L in the r-ball about $\tilde{L}$. Set

$$\beta := \|\nabla^2 f_L(\tilde{L})^{-1}\|_2 \quad \textit{and} \quad \eta := \|\nabla^2 f_L(\tilde{L})^{-1}\nabla f_L(\tilde{L})\|.$$

Define $\gamma_R = \beta\gamma$ and $\alpha = \gamma_R\eta$. If $\alpha \leqslant \frac{1}{2}$ and $r \geqslant r_0 := \frac{1-\sqrt{1-2\alpha}}{\beta\gamma}$, then the sequence $L_0 = \tilde{L}, L_1, L_2, \ldots$, produced by

$$L_{k+1} = L_k - \nabla^2 f_L(L_k)^{-1}\nabla f_L(L_k),\ k = 0, 1, \ldots$$

is well defined and converges to L_*, a unique root of the gradient ∇f_L in the closure of $B_{r_0}(\tilde{L})$. If $\alpha < \frac{1}{2}$, then L_* is the unique zero of ∇f_L in $B_{r_1}(\tilde{L})$, where

$$r_1 := \min\left\{r, \frac{1+\sqrt{1-2\alpha}}{\beta\gamma}\right\},$$

*and $\|L_k - L_*\|_F \leqslant (2\alpha)^{2k}\frac{\eta}{\alpha}$, $k = 0, 1, \ldots$. Moreover, L_* is a **lngm**.*

Proof. The proof is a direct application of the Kantorovich theorem, e.g., [54, Thm 5.3.1], along with the above lemmas and corollaries in this section. □

As mentioned previously in this section, the conditions required in Lemma 5.5.5 and Theorem 5.5.6 are fulfilled with

$$r = 10^{-3},\ \gamma = 7.24 \times 10^4,\ \bar{f}_L = 10^3.$$

Plugging them and (5.44), (5.45), (5.46) into Theorem 5.5.6, we have

$$\begin{aligned}
&1/212 < \beta < 1/210,\\
&\eta \leqslant \|\nabla^2 f_L(\tilde{L})^{-1}\|_2\|\nabla f_L(\tilde{L})\| \leqslant 1/210 \times 10^{-7},\\
&\gamma_R = \beta\gamma < 1/210 \times 7.24 \times 10^4,\\
&\alpha = \gamma_R\eta < (1/210 \times 7.24 \times 10^4) \times (1/210 \times 10^{-7}) < 1/2,\\
&r_0 = \tfrac{(1-\sqrt{1-2\alpha})}{\beta\gamma} < \tfrac{1-\sqrt{1-2\times 1/210^2 \times 7.24\times 10^{-3}}}{1/212\times 7.24\times 10^4} < r.
\end{aligned}$$

Combining these with (5.51) and (5.53) via Theorem 5.5.6, we conclude that f_L has a **lngm** in $B_r(\tilde{L})$.

In Figure 5.1 we plot the known centered global minimizer $\bar{P}$ and the centered numerically found $\tilde{P}$ near which it has been proved there exists a **lngm**. It is interesting to observe that the $\tilde{p}_i \approx \bar{p}_i$ for all $i \neq i_0$, all except the one $\bar{p}_{i_0}$ with the biggest absolute value, i.e., $|\bar{p}_{i_0}| > |\bar{p}_i|$, for all $i \neq i_0$; the corresponding $\tilde{p}_{i_0}$ has the biggest absolute value among all $\tilde{p}_i$ for $i = 1, \ldots, n = 50$, and has an opposite sign to $\bar{p}_{i_0}$.

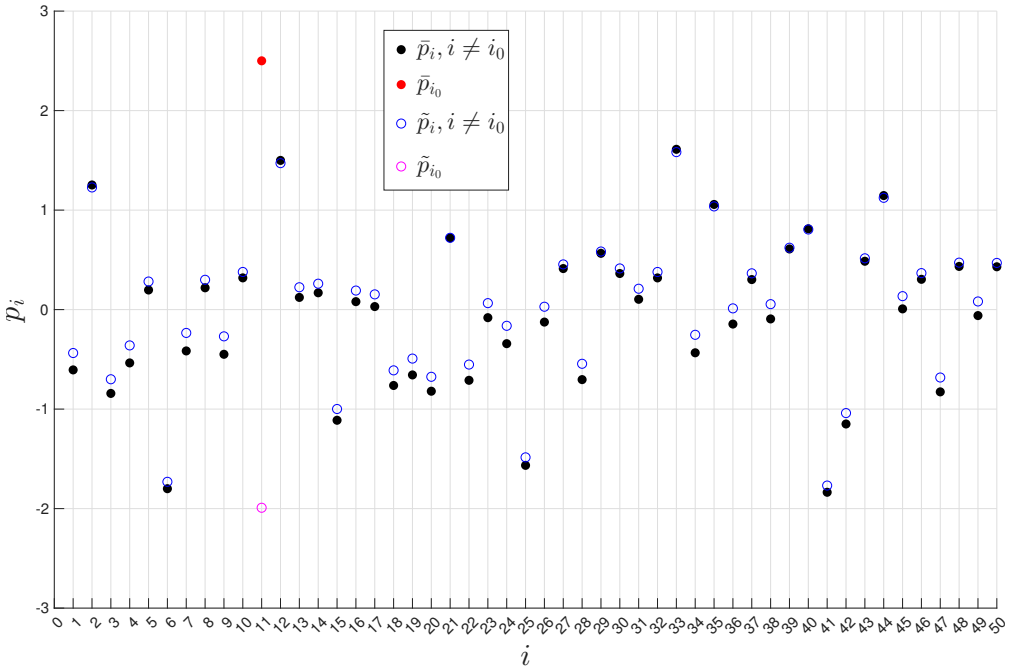

Figure 5.1: values of global min $\overline{P} \in \mathbb{R}^{50\times 1}$ and numerical configuration near **lngm** $\tilde{P} \in \mathbb{R}^{50\times 1}$ in Example 5.5.2.

Based on these observations, we generated examples with $d = 1$ and $n = 100$. We randomly generated $\bar{L}$, then get centered $\bar{P}$ by setting $\bar{P} = V\bar{L}$. Next, we find $\bar{p}_{i_0}$, which has the biggest absolute value among all $\bar{p}_i$ for $i = 1, \ldots, n$. We define the starting point $\hat{P}$ by setting $\hat{p}_i = \bar{p}_i$ for $i \neq i_0$ and $\hat{p}_{i_0} = -\bar{p}_{i_0}$. Starting from $\hat{L} = V^T\hat{P}$, the trust region method converges to a nonglobal second-order stationary point of $f_L(L)$ with high frequency.

Case $d = 2$

As mentioned in Section 5.2 for the case of $d > 1$, all local minimizers of $f_L(L)$ are nonisolated, implying that the Hessian matrix, at any local minimizer of $f_L(L)$, is singular. We consider the model $f_\ell(\ell)$ for an example with $d = 2$, and we analyze a configuration $\tilde{\ell} \in \mathbb{R}^{t_\ell}$ satisfying equivalent conditions to (5.43) but for f_ℓ:

$$\begin{array}{rlll} \text{Objective value :} & f_\ell(\tilde{\ell}) = f_\ell(\tilde{\ell}) > \tilde{f}_\ell, & \text{for some (large) } \tilde{f}_\ell > 0; \\ \text{Near stationarity :} & \|\nabla f_\ell(\tilde{\ell})\| < \tilde{g}_\ell, & \text{for some (small) } \tilde{g}_\ell > 0; \\ \text{Local Convexity :} & \lambda_{\min}(\nabla^2 f_\ell(\tilde{\ell})) > \tilde{\lambda}_\ell, & \text{for some (large) } \tilde{\lambda}_\ell > 0. \end{array} \tag{5.54}$$

We use Kantorovich Theorem to verify that this example has a strict local nonglobal minimizer ℓ_*, where the first 2 rows of $\mathcal{L}\text{Triag}(\ell_*)$ are linearly independent. According to Theorems 5.3.7 and 5.3.8, $\mathcal{L}\text{Triag}(\ell_*)$ is a local nonglobal minimizer of $f_L(L)$, and $V\,\mathcal{L}\text{Triag}(\ell_*)$ is a local nonglobal minimizer of $f(P)$.

Example 5.5.7. *An example with $n = 100, d = 2$ is given, with data $\bar{\ell}, \tilde{\ell} \in \mathbb{R}^{197}$ presented, see Footnote 3. Matrix $\bar{D}$ is the distance matrix obtained from $\bar{\ell}$ by*

$$\bar{D} = \mathcal{K}(V\,\mathcal{L}\text{Triag}(\bar{\ell})\,\mathcal{L}\text{Triag}(\bar{\ell})^T V^T).$$

Thus, $\bar{\ell}$ is a global minimizer; $\tilde{\ell}$ is a numerically convergence point obtained by a trust region method with random initialization. The objective value is

$$f_\ell(\tilde{\ell}) > 9.99 \times 10^3, \tag{5.55}$$

the absolute and relative gradient norms are

$$\|\nabla f_\ell(\tilde{\ell})\| < 9.23 \times 10^{-8}, \quad \frac{\|\nabla f_\ell(\tilde{\ell})\|}{1 + f_\ell(\tilde{\ell})} < 9.24 \times 10^{-12}, \tag{5.56}$$

and the least eigenvalue of the Hessian matrix is

$$8.58 > \lambda_{\min}(\nabla^2 f_\ell(\tilde{\ell})) > 7.93. \tag{5.57}$$

In the following Corollary 5.5.8, we verify that there exists a **lngm** of $f_\ell(\ell)$.

Corollary 5.5.8. *Let $r > 0, \tilde{\ell} \in \mathbb{R}^{(n-1)\times d}$ be given. If*

$$\gamma \geqslant 24\sqrt{2}\left(\sum_{i,j} \|(V\,\mathcal{L}\text{Triag}(\tilde{\ell}))[i,:] - (V\,\mathcal{L}\text{Triag}(\tilde{\ell}))[j,:]\|_F + 2n^{3/2} r\right), \tag{5.58}$$

then γ is a Lipschitz constant for the Hessian of f_ℓ in the radius-r neighborhood of $\tilde{\ell}$:

$$\|\nabla^2 f_\ell(\hat{\ell}) - \nabla^2 f_\ell(\check{\ell})\|_2 \leqslant \gamma \|\hat{\ell} - \check{\ell}\|, \quad \textit{for all } \hat{\ell}, \check{\ell} \in B_r(\tilde{\ell}).$$

Moreover,

$$\lambda_{\min}(\nabla^2 f_\ell(\ell)) \geqslant \lambda_{\min}(\nabla^2 f_\ell(\tilde{\ell})) - \gamma r, \quad \textit{for all } \ell \in B_r(\tilde{\ell}).$$

Proof. The results follow from the fact that, according to (5.24), the Hessian matrix of f_ℓ at $\tilde{\ell}$ is a submatrix of the Hessian matrix of f_L at $\mathcal{L}\text{Triag}(\tilde{\ell})$. The steps are similar to the proof of Proposition 5.5.4. □

In Example 5.5.7, we have

$$\sum_{i,j} \|(V\,\mathcal{L}\text{Triag}(\tilde{\ell}))[i,:] - (V\,\mathcal{L}\text{Triag}(\tilde{\ell}))[j,:]\| < 1.803 \times 10^4.$$

Let $r = 10^{-5}$. Then $\gamma = 6.12 \times 10^5$ satisfies (5.58). According to Corollary 5.5.8 and (5.57), we have

$$\lambda_{\min}(\nabla^2 f_\ell(\ell)) \geqslant \lambda_{\min}(\nabla^2 f_\ell(\tilde{\ell})) - \gamma r > 0, \quad \text{for all } \ell \in B_r(\tilde{\ell}). \tag{5.59}$$

We now continue to extend the results from Section 5.5.2 to this $d = 2$ case.

Corollary 5.5.9. *Suppose that $f_\ell(\tilde{\ell}) > \bar{f}_\ell$ and that the Hessian $\nabla^2 f_\ell$ is uniformly positive definite in the r-ball around $\tilde{\ell}$:*

$$\lambda_{\min}(\nabla^2 f_\ell(\ell)) > 0, \quad \textit{for all } \ell \in B_r(\tilde{\ell}).$$

Then, f_ℓ is positively uniformly bounded below in $B_r(\tilde{\ell})$:

$$f_\ell(\ell) > \bar{f}_\ell > 0, \quad \textit{for all } \|\ell - \tilde{\ell}\| \leqslant \min\left\{r, \frac{f_\ell(\tilde{\ell}) - \bar{f}_\ell}{\|\nabla f_\ell(\tilde{\ell})\|}\right\}.$$

Proof. The results follow as in Lemma 5.5.5. □

Let $\bar{f}_L = 10^3$. By (5.55) and (5.56), we have

$$\frac{f_\ell(\tilde{\ell}) - \bar{f}_\ell}{\|\nabla f_\ell(\tilde{\ell})\|} > \frac{8.99 \times 10^3}{9.23 \times 10^{-8}} > r.$$

Thus, according to Corollary 5.5.9, we have

$$f_\ell(\ell) > \bar{f}_\ell > 0, \quad \text{for all } \ell \in B_r(\tilde{\ell}). \tag{5.60}$$

Corollary 5.5.10. *Let $\tilde{\ell} \in \mathbb{R}^{t_\ell}$ be given and $r \in \mathbb{R}_{++}$ be found such that*

$$\nabla^2 f_\ell(\ell) \succ 0, \quad \textit{for all } \ell \in B_r(\tilde{\ell}), \tag{5.61}$$

and $\bar{f}_\ell$ satisfy

$$f_\ell(\ell) > \bar{f}_\ell > 0, \quad \textit{for all } \ell \in B_r(\tilde{\ell}).$$

Let γ be a Lipschitz constant for the Hessian of f_ℓ in the r-ball about $\tilde{\ell}$. Set

$$\beta := \|\nabla^2 f_\ell(\tilde{\ell})^{-1}\|_2, \quad \textit{and} \quad \eta := \|\nabla^2 f_\ell(\tilde{\ell})^{-1} \nabla f_\ell(\tilde{\ell})\|.$$

Define $\gamma_R = \beta\gamma$ and $\alpha = \gamma_R \eta$. If $\alpha \leqslant \frac{1}{2}$ and $r \geqslant r_0 := \frac{1-\sqrt{1-2\alpha}}{\beta\gamma}$, then the sequence $\ell_0 = \tilde{\ell}, \ell_1, \ell_2, \ldots$, produced by

$$\ell_{k+1} = \ell_k - \nabla^2 f_\ell(\ell_k)^{-1} \nabla f_\ell(\ell_k),\ k = 0, 1, \ldots,$$

is well defined and converges to ℓ_*, a unique root of the gradient ∇f_ℓ in the closure of $B_{r_0}(\tilde{\ell})$. If $\alpha < \frac{1}{2}$, then ℓ_* is the unique zero of ∇f_ℓ in the closure of $B_{r_1}(\tilde{\ell})$,

$$r_1 := \min\left\{r, \frac{1 + \sqrt{1 - 2\alpha}}{\beta\gamma}\right\}$$

and

$$\|\ell_k - \ell_*\| \leqslant (2\alpha)^{2k} \frac{\eta}{\alpha}, \quad k = 0, 1, \ldots.$$

*Moreover, ℓ_* is a **lngm**.*

Proof. As in Theorem 5.5.6, the proof is a direct application of the Kantorovich theorem. □

Plugging (5.55), (5.56), (5.57), and

$$r = 10^{-5},\ \gamma = 6.12 \times 10^5,\ \bar{f}_L = 10^3,$$

into Corollary 5.5.10, we have

$$\begin{aligned}
&1/7.93 > \beta > 1/8.58,\\
&\eta \leqslant \|\nabla^2 f_\ell(\tilde{\ell})^{-1}\|_2 \|\nabla f_\ell(\tilde{\ell})\| < 1/7.93 \times 9.23 \times 10^{-8},\\
&\gamma_R = \beta\gamma < 1/7.93 \times 6.12 \times 10^5,\\
&\alpha = \gamma_R \eta < (1/7.93 \times 6.12 \times 10^5) \times (1/7.93 \times 9.23 \times 10^{-8}) < 1/2,\\
&r_0 = \tfrac{1-\sqrt{1-2\alpha}}{\beta\gamma} < \tfrac{1-\sqrt{1-2\times 1/7.93^2 \times 6.12 \times 9.23 \times 10^{-3}}}{1/8.58 \times 6.12 \times 10^5} < r.
\end{aligned}$$

Combining this with (5.59) and (5.60), we conclude from Corollary 5.5.10 that there exists a **lngm** in $B_r(\tilde{\ell})$.

In Figure 5.2 we plot: the points $\bar{p}_i \in \mathbb{R}^2,\ i = 1, \ldots, n = 100$, of the global configuration $\bar{P} \in \mathbb{R}^{100\times 2}$, and the corresponding points $\tilde{p}_i \in \mathbb{R}^2,\ i = 1, \ldots, n$, of the numerical configuration $\tilde{P} \in \mathbb{R}^{100\times 2}$ near a proven **lngm**. We note that $\bar{p}_i \approx \tilde{p}_i, \forall i = 1, \ldots, n$, except for two indices i_0 and i_1; while $\tilde{p}_{i_0}$ and $\tilde{p}_{i_1}$ appear to be the reflections of $\bar{p}_{i_0}$ and $\bar{p}_{i_1}$. This interesting observation is similar to what happens in Example 5.5.2 as seen in Figure 5.1.

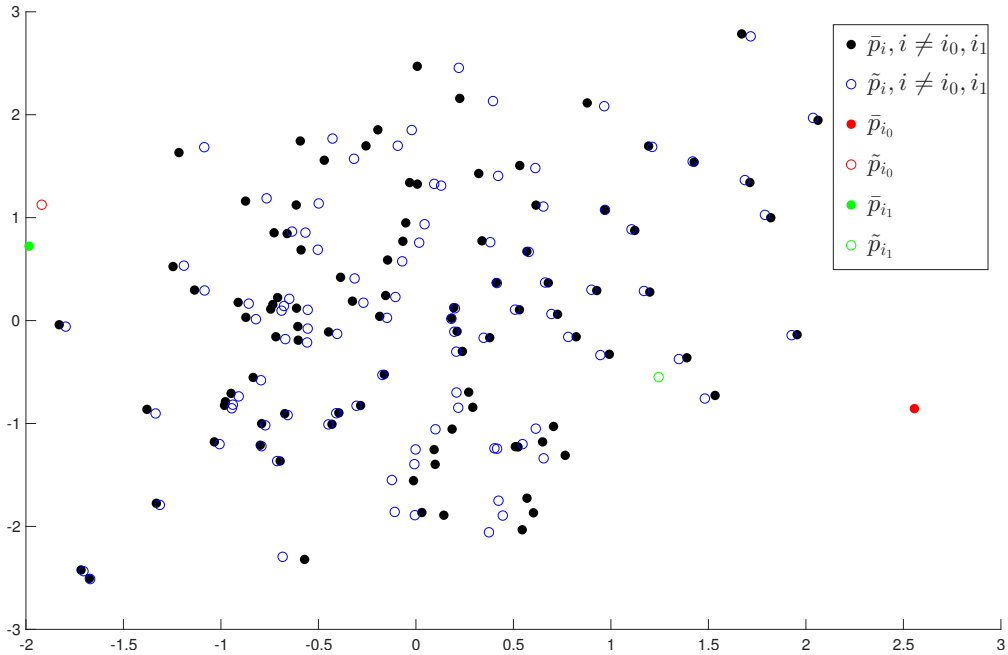

Figure 5.2: coordinates of global min $\overline{P} \in \mathbb{R}^{100\times 2}$ and numerical configuration near **lngm** $\tilde{P} \in \mathbb{R}^{100\times 2}$ in Example 5.5.7.

Chapter 6

Single Element Error Correction in an EDM

6.1 Introduction

We consider error correction for a given *Euclidean distance matrix,* ***EDM***, D, where the elements are the squared distances between n points in $\mathbb{R}^d$ with d the embedding dimension and $d \geqslant 1$. For our problem, we know the embedding dimension, and it is given that *exactly one* distance is corrupted with nonzero noise. But we do not know the magnitude or position of the noisy distance. Thus our problem is a **EDM** completion problem with known embedding dimension where the position of the noisy unknown element is not given. This means that there is a hard combinatorial element to the problem. We use three different *divide and conquer,* ***D&C***, approaches in combination with three different versions of *facial reduction,* ***FR***, one of which involves Gale transforms. These methods solve this problem efficiently and accurately in all but two hard cases. The results are related to *yielding* in an **EDM** studied in [6], i.e., characterizing when a perturbation of one element yields an **EDM**. We also show that the intuitive approach of finding the nearest **EDM** only works in the trivial case where the original distance is 0, degenerate, and the perturbation is negative.

The first **D&C** approach divides the matrix into two overlapping principal submatrix blocks. We then either identify a block that contains the corrupted element and continue with the smaller problem for that block; otherwise we use **EDM** completion to identify the correct position and magnitude of the noise. The second approach divides the matrix into the largest number of overlapping principal submatrix blocks with minimal overlap that allows for **EDM** completion. Again, we either identify one block to further subdivide the problem or we complete the **EDM** to locate the correct position and identify the noise. The third approach divides the matrix into principal submatrix blocks of smallest size that can be completed and we again apply the divide and conquer approach.

The three **D&C** approaches are used in combination with three different types of facial reduction: (i) finding exposing vectors Z_i for each block and then using the exposing vector $Z = \sum_i Z_i$ to identify the face $f = \mathbb{S}^n_+ \cap Z^\perp \trianglelefteq \mathbb{S}^n_+$; (ii) finding a *facial vector,* ***FV***, denoted V_i, for each block, and then finding the intersection $\text{range}(V) = \cap_i \text{range}(V_i)$ efficiently so as to identify the face $f = V\mathbb{S}^d_+V^T \trianglelefteq \mathbb{S}^n_+$, of the semidefinite cone $\mathbb{S}^n_+$; (iii) a combination of exposing and facial vectors

first finds a Gale matrix N so that $Z = NN^T$ is an exposing vector and yields a facial vector V to identify the face f. Identifying the correct face allows one to find the correct full rank factorization of the Gram matrix to complete the **EDM**.

Included are several interesting results on: the commutativity of some operators for this distance geometry problem; and properties of having exactly one noisy element. Our main result is three successful algorithms that efficiently find the position and magnitude of the corrupted distance. The algorithms scale well. The best algorithm can solve problems of the order of $100,000$ points in approximately one minute, and to machine precision. With embedding dimension d, the data is then size $100,000d$ with resulting $D \in \mathbf{EDM}$ of size $n = 100,000$. Theoretical difficulties arise when the points are not in *general position*. Note that the n points are in general position if the affine hull of every subset of $k \geqslant d+1$ points is all of $\mathbb{R}^d$. In fact, we prove that difficulties arise only when all but at most $d+1$ points are in a linear manifold of dimension $d-2$.[1]

In the process of this study, we review the relationships between three methods of facial reduction and Gale transforms. In addition, we show that the only time that the intuitive approach, the nearest **EDM** problem, solves our problem is the case where the original distance is 0, and the perturbation is negative, i.e., a trivial case.

6.1.1 Related Literature

The literature for **EDM** and more generally *distance geometry* is vast and includes many surveys and books with many applications in multiple areas of mathematics, engineering, health sciences. Classical results on **EDM** completion problems appear in e.g., [16, 18, 105, 106, 115] and relate to applications in protein folding and molecular conformations, e.g., [34, 158], as well as sensor network localization, e.g., [34, 113].

Completing a partial **EDM** using facial reduction, **FR**, and exploiting cliques and facial vectors was presented in [113]. This approach was extended to the noisy case in [61] by exploiting the exposing vector representation for **FR**. Other approaches that do not necessarily use **FR** are discussed in e.g., [8, 56, 112] and the more recent surveys [59, 120] and the references therein. The paper [86] discusses perturbation analysis for **EDM** completion and convergence of algorithms under perturbations. Matrix completion with noise and *outliers* is discussed in [178]. Further applications of **EDM** for finding the so-called *kissing number* are found in sphere packing that has further applications to error correcting codes from the fields of communications. A recent discussion on **EDM** completion with noisy data is in [133]. More recently, finding the nearest **EDM** in the ℓ_0 norm, the least number of changed elements, is studied in [44]. Many applications are included as well.

An application to finding a single node in the graph where the edges to the node are known but can be noisy was done in [145]. This is closely related to our problem and the paper includes applications to real world problems, e.g., when the unknown node is a cellular phone and the other nodes are cellular towers. Our application would consider the case when one tower is partially obstructed and provides a noisy distance to the cellular phone. Further results related to our problem is [5, Section 7.1] that deals with the completion problem for *one* missing entry of an **EDM**. See also [16].

[1] These exact conditions for difficulties arise in finding examples of local nonglobal minima for configuration matrix problems presented in [159, Example 5.1].

Further, we relate **FR** to Gale transforms. Details for Gale transforms are presented in [69] and discussed further in e.g., [7].

6.1.2 Outline and Main Results

In Section 6.2 we present the main problem description. This includes the relationship with semidefinite programming, **SDP** and the Gram matrix. We include the background on **FR** and results on commutativity involving the mapping between the semidefinite cone $\mathbb{S}^n_+$ and the **EDM** cone. Further definitions and results are presented as needed.

In addition, in Section 6.2.2 we show that the only case when the nearest **EDM** formulation, **NEDM**,, solves our problem is when the original element (distance) is zero and the perturbation is negative. This emphasizes the difference in our approach and that of **FR** with a **NEDM** approach. Solving **FR** for **NEDM** allows for reduction in the size of the problem while also improving robustness, e.g., [61]. **FR** for our approach also reduces the size of the problem but it allows for recognizing the location of the *single* corrupted element in an efficient way. We note that moving to more than one corrupted element greatly increases the difficulty of the problem, as one moves from characterizing intervals to higher dimensional objects that maintain the **EDM** property as well as the embedding dimension. In Section 6.3 we present the three algorithms: Section 6.3.1 presents the bisection **D&C** with facial reduction using exposing vectors, **BIEV**; Section 6.3.2 presents the multi-block case with **FV**s, **MBFV**, and is based on an efficient technique for finding the intersection of faces. A proof of finite convergence for this algorithm can be given based on the characterization of so-called *restricted yielding* in Theorem 6.5.2, i.e., characterizing an interval of perturbation of an element that preservers both the **EDM** property as well as the embedding dimension d. Finite convergence for the other approaches can be proved similarly. Section 6.3.3 presents an equivalent approach that uses small blocks with Gale transforms, **SBGT**. This section also highlights the interesting relationships, equivalences, between facial reduction and Gale transforms.

In Section 6.4 we present a complexity analysis of the methods as well as present the empirical results. We solve small, large, huge problems.

In Section 6.5 we discuss the two hard cases where the algorithms can fail. We present a characterization in Theorem 6.5.2 for detecting one noisy distance. Here we extend the results on *yielding* in [5, Section 7.2], i.e., characterizing when the **EDM** property is preserved under a single element perturbation. We provide a characterization when a perturbation yields (preserves) not only an **EDM** but also maintains embedding dimension d. The result states that all but two points are on a linear manifold of dimension $d-2$; and this is particularly elegant for $d=2$. We include empirics for the hard case.

6.2 Problem Description

We recall basic results about **EDM** and **SDP** in Section 2.3. In particular, Lemma 2.3.1 provides a useful commutativity result for $\mathcal{K}, \mathcal{P}_\alpha$, while Lemma 2.4.11 recalls the result that a sum of exposing vectors is an exposing vector. Theorem 2.4.12 shows how to find a (centered) **FV** for the face corresponding to a given principal submatrix D_α. Then Lemma 2.4.13 illustrates an efficient and accurate way to find the intersection of two given faces by finding a **FV** from the corresponding

two given **FV**s. Many of these can be found in classical references e.g, [5,46,114] and the references therein.

The results here use the *general position* assumption. Further results are in Theorem 6.5.2 that characterizes when the perturbation of a single element in an **EDM** with embedding dimension d can still be an **EDM** with embedding dimension d. This is important for our general algorithm in identifying the location of the corrupted element in the **EDM**. The main problem is presented in Section 6.2.1.

6.2.1 Main Problem Description and Model

We begin with a description of the *main problem.*

Problem 6.2.1 (Find magnitude/location of noise in **EDM**)**.** *Let the original data $D_0 \in \mathcal{E}^n$, the cone of **EDM**s, and let $d = \text{embdim}(D_0) \geqslant 1$. Let $1 \leqslant i < j \leqslant n$ be unknown fixed indices and let the unknown noise magnitude be $0 \neq \epsilon \in \mathbb{R}$. Define the corrupted data matrix $D_n = D_0 + \epsilon E_{ij}$. Then:*

> *Find the position i,j and the value of the noise ϵ from the given corrupted data matrix D_n with one noisy data element.*

Remark 6.2.2 (Naive Discrete Model)**.** *The fact that we have a single noisy element and a known embedding dimension leads to the following rank constrained mixed integer **MIP** and **SDP**:*

$$
\begin{array}{rccl}
 & p^* = & \min & \|H \circ (\mathcal{K}(G) - D_n)\|_F^2 \\
 & & s.t. & G \succeq 0,\ \text{rank}(G) = d \\
(\textbf{\textit{MIP}}, \textbf{\textit{SDP}}) & & & \text{diag}(H) = 0 \\
 & & & H \in \mathbb{S}^n \cap \{0,1\}^{n \times n} \\
 & & & e^T H e = n^2 - n - 2,
\end{array}
\tag{6.1}
$$

where $H \circ D$ denotes the Hadamard product and H is the adjacency matrix of the complete graph where one edge is missing as characterized by the last constraint. Since exactly one element is noisy, we know that the optimal value $p^* = 0$ when the correct position i, j is chosen for H. This problem looks extremely difficult as we deal with binary variables as well as semidefinite and rank constraints. Our approach in this chapter takes advantage of the structure to get extremely accurate solutions very quickly, while avoiding the difficult nature of the hard discrete problem (6.1).

6.2.2 Nearest EDM, NEDM, Formulation and Solving Main Problem 6.2.1

It is natural to try and solve Problem 6.2.1 by finding the nearest **EDM**, (**NEDM**, see (6.3) below) to the given noisy D_n, i.e., relax the **MIP** formulation (6.1). This emphasizes the difficulty of Problem 6.2.1 when the noisy data $D_n \in \mathcal{E}^n$. We now see that the **NEDM** approach works only in the case where the corrupted element arises from a negative noise at a zero distance in the original $D_0 \in \mathcal{E}^n$, i.e., the degenerate case of one point on top of another.

We now choose a *facial vector* V that allows for facial reduction for **EDM** completion problems. We let V be chosen so that $[V\ e]$ is an orthogonal matrix and for $X \in \mathbb{S}^{n-1}$ let $\mathcal{K}_V(X) := \mathcal{K}(VXV^T)$,

where $\mathcal{K}$ is given in (2.4). Results on adjoints, ranges and nullspaces are given in [2]. For example, the adjoints are:

$$\mathcal{K}^*(D) = 2(\mathrm{Diag}(De) - D),\ \mathcal{K}_V^*(D) = V^T\mathcal{K}^*(D)V,$$

Moreover,

$$0 = VXV^T \iff 0 = V^TVXV^TV \iff X = 0,$$

and therefore, for $D = \mathcal{K}(VXV^T)$ we get that VXV^T is a centered matrix by definition of V, and so D is a hollow matrix and

$$X \neq 0 \implies \langle X, \mathcal{K}_V^*(\mathcal{K}_V(X))\rangle = \|\mathcal{K}_V(X)\|_F^2 = \|D\|_F^2 \neq 0 \implies \mathcal{K}_V^*\mathcal{K}_V \succ 0, \tag{6.2}$$

i.e., $\mathcal{K}_V^*(\mathcal{K}_V(\cdot)) \succ 0$, is a positive definite linear operator. The facially reduced **NEDM** then has a unique solution and is:

$$\bar{X}(D_n) = \bar{X} = \underset{X\geq 0}{\mathrm{argmin}}\ \frac{1}{2}\|\mathcal{K}_V(X) - D_n\|_F^2. \tag{6.3}$$

Lemma 6.2.3 ($\mathcal{N}_{\mathbb{S}_+^n}(\bar{X})$, *normal cone*)**.** *Let*

$$\bar{X} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} \bar{\Lambda} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^T \in \mathbb{S}_+^n,\ \bar{\Lambda} \in \mathbb{S}_{++}^r,$$

with $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ *orthogonal. Let* $\lambda \in \mathbb{R}_+^n$ *be the vector of eigenvalues of* $\bar{X}$*. Then the normal cone*

$$\begin{aligned} \mathcal{N}_{\mathbb{S}_+^n}(\bar{X}) &= Q\,\mathrm{Diag}\left(\mathcal{N}_{\mathbb{R}_+^n}(\lambda)\right)Q^T \\ &= \{X \in -\mathbb{S}_+^n : X\bar{X} = 0\} \\ &= \left\{X \in -\mathbb{S}_+^n : X = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \hat{\Lambda} \end{bmatrix} \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^T, \hat{\Lambda} \in -\mathbb{S}_+^{n-r}\right\} \\ &= -\left(\mathbb{S}_+^n - \bar{X}\right)^+ \quad (\textit{nonnegative polar cone}) \\ &= -\,\mathrm{face}(\bar{X})^\Delta, \quad (\textit{the conjugate face}). \end{aligned}$$

Proof. Results on normal cones for **SDP** are well known, e.g., [25, 144]. We include a proof for completeness.

The proof follows from using the Eckart-Young-Mirsky, e.g., [65], and the Moreau decomposition, e.g., [131], theorems. That is $\bar{X}$ is the nearest point in $\mathbb{S}_+^n$ to a given $Y \in \mathbb{S}^n$ if, and only if, $Y - \bar{X} \in \mathcal{N}_{\mathbb{S}_+^n}(\bar{X})$, and Y has an orthogonal decomposition

$$Y = Y_+ + Y_-,\ Y_+, -Y_- \in \mathbb{S}_+^n,\ Y_+Y_- = 0,$$

using the projections Y_+, Y_- onto the nonnegative and nonpositive semidefinite cones, respectively. Therefore $Y - Y_+ = Y_-, Y_+ = \bar{X}$ yields the result. Note that the Eckart-Young-Mirsky theorem characterizes the projections as being the same as using the Moreau decomposition. □

We now present a characterization for when the nearest point finds the original **EDM** after a single element is corrupted. Note that for our problem the i, j are not known.

Theorem 6.2.4 (When **NEDM** finds D_0)**.** *We define* $D_0 \in \mathcal{E}^n, E_{ij} \in \mathbb{S}^n$*, indices* i, j *as above, and*

let the corrupted $D_n = D_0 + \epsilon E_{ij} \notin \mathcal{E}^n$. *Further let*

$$\begin{array}{rcl} Y := \mathcal{K}_V^*(E_{ij}) & = & V^T \mathcal{K}^*(E_{ij}) V \\ & = & 2\left[v_i^T v_i + v_j^T v_j - v_i^T v_j - v_j^T v_i\right], \end{array}$$

where v_i *is the* i*-th row of* V. *Then* $\epsilon \neq 0$, $0 \neq Y \succeq 0$, *and*

$$\begin{array}{rcll} D_0 \text{ is the nearest } \textbf{\textit{EDM}} \text{ to } D_n & \iff & D_0 = \mathcal{K}_V(\bar{X}), \text{ for } \bar{X} \text{ from (6.3)}, & \\ & \iff & X_0 := \mathcal{K}_V^\dagger(D_0) \in Y^\Delta := \mathbb{S}_+^{n-1} \cap Y^\perp, \text{ and } \epsilon < 0, & \\ & \iff & X_0 := \mathcal{K}_V^\dagger(D_0) \in Y^\Delta := \mathbb{S}_+^{n-1} \cap Y^\perp, \ \forall \epsilon < 0, & \\ & \iff & (D_0)_{ij} = 0, \quad \text{and } \epsilon < 0. & \end{array} \tag{6.4}$$

Proof. We consider the nearest **EDM** problem in (6.3) with $X_0 = \mathcal{K}_V^\dagger(D_0)$ as the original data. We want to characterize D_0, or equivalently X_0, so that the optimum $\bar{X}$ in (6.3) equals X_0, i.e, the **NEDM** problem solves our Problem 6.2.1.

First we note from (6.2) that $\mathcal{K}_V^* \mathcal{K}_V \succ 0$ so that the objective in (6.3) is strongly convex and we have a unique optimum. The optimality conditions for X_0 for the nearest point problem are that the gradient of the objective function is in the polar cone (negative normal cone) of the **SDP** cone at X, see Lemma 6.2.3, i.e., we have the equivalences

$$\begin{array}{rcl} \mathcal{K}_V^*(\mathcal{K}_V(X_0) - D_n) \in (\mathbb{S}_+^{n-1} - X_0)^+ & \iff & \mathcal{K}_V^*(\mathcal{K}_V \mathcal{K}_V^\dagger(D_0) - D_n) \in (\mathbb{S}_+^{n-1} - X_0)^+ \\ & \iff & \mathcal{K}_V^*(D_0 - D_n) \in (\mathbb{S}_+^{n-1} - X_0)^+ \\ & \iff & \epsilon \mathcal{K}_V^*(-E_{ij}) \in (\mathbb{S}_+^{n-1} - \mathcal{K}_V^\dagger(D_0))^+ \\ & \iff & -\epsilon \mathcal{K}_V^*(E_{ij}) \in (\mathbb{S}_+^{n-1} - \mathcal{K}_V^\dagger(D_0))^+. \end{array}$$

Note that $\text{range}(\mathcal{K}) = \text{range}(\mathcal{K}_V) = \mathcal{S}_H^n$, the hollow subspace defined in (2.5). Therefore $\text{null}(\mathcal{K}_V^*) = (\mathcal{S}_H^n)^\perp = \text{range}(\text{Diag})$. As E_{ij} is not diagonal, we get $\mathcal{K}_V^*(E_{ij}) \neq 0$. Alternatively, we recall that the choice of V implies $\bar{V} := \begin{bmatrix} V & \frac{1}{\sqrt{n}} e \end{bmatrix}$ is orthogonal. Therefore, the rows satisfy

$$v_i^T v_j + \frac{1}{n} = 0, v_i^T v_i + \frac{1}{n} = 1 \implies v_i^T v_i + v_j^T v_j - v_i^T v_j - v_j^T v_i = 2\left(1 - \frac{1}{n} + \frac{1}{n}\right) = 2 \neq 0,$$

i.e., we see again that $\mathcal{K}_V^*(E_{ij}) \neq 0$.

We now observe that necessarily $\epsilon \leqslant 0$. Without loss of generality, we assume $\epsilon < 0$. To have the nearest matrix be the original data D_0, X_0, we need to have

$$\begin{array}{rcl} \mathcal{K}_V^*(E_{ij}) & \in & (\mathbb{S}_+^{n-1} - \mathcal{K}_V^\dagger(D_0))^+ \\ & = & (\mathbb{S}_+^{n-1} - X_0)^+ \\ & = & \{X \in \mathbb{S}_+^{n-1} : X X_0 = 0\} \\ & = & \text{face}(X_0)^\Delta, \end{array}$$

i.e., the conjugate face to $\text{face}(X_0)$.

Next we see the equivalence using v_i, the rows of V:

$$\begin{aligned} Y = \mathcal{K}_V^*(E_{ij}) &= V^T \mathcal{K}^*(E_{ij}) V \\ &= 2V^T(\mathrm{Diag}(E_{ij}e) - E_{ij})V, \qquad (\succeq 0, \text{ since congruence of a Laplacian matrix}) \\ &= 2V^T(e_i e_i^T + e_j e_j^T - e_i e_j^T + e_j e_i^T)V \\ &= 2\left[v_i^T v_i + v_j^T v_j - v_i^T v_j - v_j^T v_i\right]. \end{aligned}$$

Since $Y, X_0 \in \mathbb{S}_+^{n-1}$, we have

$$Y \in \mathrm{face}(X_0)^\Delta \iff X_0 \in \mathrm{face}(Y)^\Delta \iff YX_0 = 0 \iff \mathrm{trace}\, YX_0 = 0.$$

Therefore, the first two equivalences in (6.4) follow as $D_0 = \mathcal{K}_V(X_0)$.

Finally, we note that the conjugate face condition is equivalent to

$$0 = \mathrm{trace}\, X_0 Y = \mathrm{trace}\, X_0 V^T \mathcal{K}^*(E_{ij}) V = \mathrm{trace}\, \mathcal{K}(VX_0V^T)E_{ij} = \mathrm{trace}\, D_0 E_{ij}.$$

Therefore, we can only have a negative perturbation from $(D_0)_{ij} = 0$. □

6.3 Three D&C and Three FR Methods

We have three different **D&C** methods: bisection; minimum block overlap; and minimum block size. And we have three different **FR** methods: exposing vector; facial vector; and Gale transform. The result is nine possible algorithms. We now pair each **D&C** method with exactly one **FR** method and describe and implement these three methods. For simplicity, we assume that the *general position* assumption holds, i.e., we can easily find a principal submatrix of the Gram matrix with rank d, see Remark 6.3.1.

6.3.1 Bisection D&C with Exposing Vector for FR

The first algorithm combines the bisection approach for **D&C** with the exposing vector approach for **FR**, denoted **BIEV**. We divide the corrupted data matrix D_n into 2 properly overlapping blocks corresponding to principal submatrices. We first identify whether the noisy element is within one of these two principal blocks and reduce the problem to that block. Or if it is outside both chosen principal blocks, then we apply **FR** explicitly to solve the problem and find the position (i, j) and noise ϵ.

Recall that d is the embedding dimension. We assume that $n \gg d$, i.e., is sufficiently larger than d. Then we use the two blocks indexed by columns (rows)

$$I_1 = \{1, \ldots, \lceil (n + d + 2)/2 \rceil\}, \quad I_2 = \{\lfloor (n - d - 2)/2 \rfloor, \ldots, n\}, \tag{6.5}$$

i.e., we have the two principal submatrices $D_1 = D_{I_1}, D_2 = D_{I_2}$ which overlap in the block of size at least $\frac{1}{2}((n + d + 2) - (n - d - 2)) = d + 2$. In addition, the assumption $n \gg d$ implies each block is at least size $d + 1$.

Remark 6.3.1. *The overlap corresponds to the sets of points*

$$I := I_1 \cap I_2 = \{\lfloor (n - d - 2)/2 \rfloor, \ldots, \lceil (n + d + 2)/2 \rceil\}.$$

If the corresponding (centered) Gram matrix has rank d, then we know that the overlap of the graph is rigid, i.e., the missing distances in $I_1 \cup I_2$ can be uniquely found using e.g., ***FR****, see e.g., [113, Lemma 2.9] and further clarification in Section 6.3.2, below. If this is not the case, then we need to permute the columns of D_n in order to obtain a rigid overlap.*

This raises the hard *question of how to find the best overlap. The simple case occurs if the overlap is in general position, i.e., each principal submatrix $D_\alpha, \alpha \subset [n], |I| \geqslant d+1$ corresponds to a Gram matrix with rank d.*

We now find the *supposed* Gram matrices $G_i = \mathcal{K}^\dagger(D_i), i = 1, 2$, i.e., these are indeed *centered* Gram matrices with rank d if D_i is an **EDM**. There are now three cases to consider for this current approach.

Case 1: Reducing Size of Problem

Suppose that one of $G_i, i = 1, 2$, is not positive semidefinite of rank d, i.e., as a result of the noise in the data, the corresponding submatrix D_i is not an **EDM** or does not have embedding dimension d. Then we can continue on the **D&C** approach and reduce our problem to that submatrix. And then we continue the division, i.e., we have reduced the problem by a factor of roughly 2. We then redefine n and D_n appropriately and return to dividing the indices in (6.5).

Case 2: EDM Completion using FR

We first recall that for $G \succeq 0$, face(G) denotes the smallest face containing G. Let V, N denote matrices with columns that are orthonormal basis for range(G), null(G), respectively, e.g., made up of an orthonormal set of eigenvectors. Let $d = \text{rank}(G)$. Then as stated above,

$$\text{face}(G) = V\mathbb{S}^d_+V^T = \mathbb{S}^n_+ \cap (NN^T)^\perp. \tag{6.6}$$

V is a *facial vector*, [103], while $Z = NN^T$ is an *exposing vector*, e.g., [61]. And moreover, the sum of exposing vectors is an exposing vector, Lemma 2.4.11:

$$Z_i \succeq 0, \text{trace}\, Z_iG = 0, i = 1, \ldots, k \implies \sum_i Z_i \succeq 0, \text{trace}\left(\left(\sum_i Z_i\right) G\right) = 0. \tag{6.7}$$

Suppose both $G_i, i = 1, 2$, are *centered* Gram matrices with rank d. Then we know the corrupted element/distance is outside the principal blocks. We now continue to completely solve the problem using **FR** as we can now complete the partial **EDM** formed from the two blocks. We now give the details.[2]

1. Using the spectral decomposition of $G_i, i = 1, 2$, given above, we obtain orthonormal bases of *centered* null vectors for the nullspaces of the Gram matrices $G_i, i = 1, 2$,

$$G_iN_i = 0,\ N_i^Te = 0,\ N_i^TN_i = I,\ i = 1, 2. \tag{6.8}$$

[2] See e.g., [61, Algor. 1, Pg. 2308] for more details.

Let $Z_i = N_i N_i^T \succeq 0, i = 1, 2$, be the corresponding *centered* exposing vectors, i.e., we have

$$Z_i G_i = 0, G_i e = 0,\ Z_i e = 0, G_i + Z_i + ee^T \succ 0,\ i = 1, 2. \tag{6.9}$$

By centering N_i we get that both G_i, Z_i are centered and thus we add the missing element e to get positive definite.

2. Let W_i be zero matrices of order n and set

$$(W_i)_{I_i} = Z_i,\ i = 1, 2,$$

i.e., we place the *centered* exposing vectors Z_i into the correct blocks. Now each $W_i, i = 1, 2$, is a *centered* exposing vector for the true centered Gram matrix G. As the sum of exposing vectors is an exposing vector, we form the *centered* exposing vector of the true centered Gram matrix G,

$$Z = W_1 + W_2. \tag{6.10}$$

This yields a *maximum rank exposing vector*

$$GZ = 0,\ Ge = 0,\ Ze = 0,\ Z \succeq 0,\ G + Z + ee^T \succ 0. \tag{6.11}$$

3. We choose $V, V^T V = I$, to be full column rank and to span null$([Z\ e]^T)$.[3] This completes the **FR** as we have

$$G = VRV^T,\ R \in \mathbb{S}^d_{++},\ Ge = 0. \tag{6.12}$$

4. We denote the *adjacency matrix,* H_α to be the matrix of zeros with ones in the positions indexed by α. Recall that $H \circ D$ denotes the Hadamard product. We solve the cone least squares problem that does not include the noisy element:

$$\min_{R \geq 0} \|H_{I_1 \cup I_2} \circ \mathcal{K}(VRV^T) - H_{I_1 \cup I_2} \circ D_n\|_F^2. \tag{6.13}$$

Since R is order d and at the start $d << n$, this can be a very overdetermined problem and can be helped by using a *sketch matrix*, see e.g. [61].

5. In the case of random data, we expect all non-noisy blocks of size at least $d + 1$ to be proper **EDM**s. Therefore, the optimal solution R in (6.13) is unique and in $\mathbb{S}^d_{++}$. Therefore, we can solve this least squares problem as an unconstrained problem and improve on the accuracy and speed. There is one constraint, linear equation, for each distinct pair in the set

$$\mathcal{I}_R := \{(i, j)\ :\ (i, j) \in (I_1 \times I_1) \cup (I_2 \times I_2),\ i < j\}\,.$$

We let svec : $\mathbb{S}^n \to \mathbb{R}^{t(n)}$ denote the isometric mapping that vectorizes a symmetric matrix columnwise with multiplying the off-diagonal elements by $\sqrt{2}$. The inverse (and adjoint) is sMat = svec*. Similarly, usvec : $\mathbb{S}^n \to \mathbb{R}^{t(n-1)}$ is for the strict triangular part with usMat = usvec*. Let the unknown variable be $r = \text{svec}(R) \in \mathbb{R}^{t(d)}$, where we denote *triangular number,* $t(d) := d(d+1)/2$. The $m_R \times n_R$ system, $m_R = t(|I_1| - 1) + t(|I_2| - 1) - t(|I_1 \cap I_2| - 1) \times (n_R := t(d))$ linear system to solve for r is

$$\text{usvec}\, H_{I_1 \cup I_2} \circ \mathcal{K}(V\, \text{sMat}(r) V^T) = \text{usvec}\, H_{I_1 \cup I_2} \circ D_n, \quad (: \mathbb{R}^{t(d)} \to \mathbb{R}^{t(n-1)}). \tag{6.14}$$

[3] As noted following (6.6), this is called a *facial vector* [103]. In fact, this is a centered facial vector.

The columns of the matrix representation are obtained by replacing r with unit vectors $e_i, i = 1, \ldots t(d)$. We note that the rows corresponding to i, j not in the union of the two blocks are zero and can be discarded.

Alternatively, we could find the transpose of the matrix representation by taking the adjoint of the left-hand side in (6.14) and working on $\mathbb{R}^{t(n)} \to \mathbb{R}^{t(d)}$. We use $g = e_\iota \in \mathbb{R}^{t(n-1)}$, $\mathrm{usMat}(g)$. The adjoint is given by

$$\mathrm{svec}\left(V^T \mathcal{K}^* \left[H_{I_1 \cup I_2} \circ \mathrm{usMat}(g)\right] V\right). \tag{6.15}$$

We first simplify $H_{I_1 \cup I_2} \circ \mathrm{usMat}(e_\iota)$ to get the row of the matrix representation. For each $\iota \in [t(n-1)]$, we can find $i, j \in [n]$ such that $\iota = i + \sum_{\ell=0}^{j-2} \ell$ and observe

$$H_{I_1 \cup I_2} \circ \mathrm{usMat}(e_\iota) = \begin{cases} \frac{1}{\sqrt{2}}\left(e_i e_j^T + e_j e_i^T\right) & \text{if } H_{ij} = 1; \\ 0 & \text{otherwise,} \end{cases}$$

where H_{ij} denotes the i, j-th entry of $H_{I_1 \cup I_2}$. Restricting g to the case when $H_{ij} = 1$, gives

$$\begin{aligned} \mathcal{K}^* \left[H_{I_1 \cup I_2} \circ \mathrm{usMat}(e_\iota)\right] &= \mathcal{K}^* \left[\tfrac{1}{\sqrt{2}}\left(e_i e_j^T + e_j e_i^T\right)\right] \\ &= \sqrt{2}\left[\mathrm{Diag}\left((e_i e_j^T + e_j e_i^T)e\right) - (e_i e_j^T + e_j e_i^T)\right] \\ &= \sqrt{2}\left[\mathrm{Diag}\left(e_i + e_j\right) - e_i e_j^T - e_j e_i^T\right] \\ &= \sqrt{2}\left(e_i e_i^T + e_j e_j^T - e_i e_j^T - e_j e_i^T\right). \end{aligned}$$

Plugging this into (6.15), we obtain

$$\begin{aligned} \sqrt{2}\,\mathrm{svec}\left(v_i^T v_i + v_j^T v_j - v_i^T v_j - v_j^T v_i\right) &= \sqrt{2}\,\mathrm{svec}\left(v_i^T (v_i - v_j) - v_j^T (v_i - v_j)\right) \\ &= \sqrt{2}\,\mathrm{svec}\left((v_i - v_j)^T (v_i - v_j)\right) \end{aligned} \tag{6.16}$$

as a column of the matrix representation of the adjoint. Here v_i is the i-th row vector of V. The advantage here is that we can use unit vectors for g restricted to the indices corresponding to $I_1 \cup I_2$ when finding the columns.

In our implementations, the calculation of the matrix representation was the most expensive step. We now show how to avoid this step.

This provides a simplification for solving the least squares problem in (6.13).

Corollary 6.3.2. *Let*

$$D \in \mathcal{E}^n,\ G = \mathcal{K}^\dagger(D),$$

and let $\alpha \subset [n], |I| \geqslant d + 1$. *Define*

$$D_\alpha := \mathcal{P}_\alpha(D),\ G_\alpha := \mathcal{K}^\dagger(D_\alpha).$$

Suppose that $\mathrm{rank}(G_\alpha) = \mathrm{rank}(G) = d$. *Then*

$$\begin{aligned} \mathcal{K}^\dagger \mathcal{P}_\alpha \mathcal{K}(G) &= \mathcal{K}^\dagger \mathcal{K}(\mathcal{P}_\alpha(G)) \\ &= J(\mathcal{P}_\alpha(G))J. \end{aligned}$$

Moreover, if $G = WW^T, G_\alpha = W_\alpha W_\alpha^T$ *are full rank factorizations, and* V, V_α *are centered facial*

vectors with

$$\text{range}(V) = \text{range}(G),\ \text{range}(V_\alpha) = \text{range}(G_\alpha),$$

then

$$Q = V^\dagger W = (JV_\alpha)^\dagger W_\alpha, \quad W = W_\alpha,$$

*and the **EDM** can be recovered with* $D = \mathcal{K}(VQQ^TV^T)$.

Proof. Let $G \in \mathbb{S}^n_+ \cap \mathcal{S}^n_C, face(G) = V\mathbb{S}^d_+V^T \trianglelefteq \mathbb{S}^n_+$, V a given centered, $V^Te = 0$, facial vector of full rank. The first part of the corollary is a direct consequence from Lemma 2.3.1. Let $G = WW^T$ be a full rank factorization and let Q be a solution of

$$J\mathcal{P}_\alpha VQ = W_\alpha.$$

Then,

$$\begin{aligned} \mathcal{K}^\dagger\mathcal{P}_\alpha\mathcal{K}(VQQ^TV^T) &= J(\mathcal{P}_\alpha(VQQ^TV^T))J \\ &= JV_\alpha QQ^TV_\alpha^TJ \\ &= JV_\alpha Q(JV_\alpha Q)^T \\ &= W_\alpha W_\alpha^T = G_\alpha. \end{aligned}$$

Therefore,

$$\mathcal{P}_\alpha\mathcal{K}(VQQ^TV^T) = \mathcal{K}(G_\alpha) = D_\alpha$$

and thus

$$\mathcal{P}_\alpha\big(\mathcal{K}(VQQ^TV^T) - D\big) = 0.$$

□

Using the above lemma we can avoid finding the matrix representation and find the full rank factoriazation of the small R instead.

Case 3: Small Remaining Block

Suppose that there is a single last block I with G_α that is not a proper Gram matrix with the correct rank but is too small to divide further, e.g., $< 2d+2$. Without loss of generality, we assume $I = \{1, 2, \ldots, \ell\}$. Then there are $\ell(\ell-1)/2$ possible elements that are noisy. We can now use other distances to find the noisy one, i.e., for $i, j \in I, i \neq j$, we set

$$I_{ij} = \{i, j, \ell+1, \ell+2, \ldots \ell+t\},\ |I_{ij}| = d+1,$$

and verify whether or not $K^\dagger\left(D_{I_{ij}}\right)$ is a Gram matrix with rank d. As soon as we find the one that is not, then we have found the noisy position i, j. We then choose a *well conditioned* block $I_0, |I_0| \geqslant d+1$ and set

$$I_1 = I_0 \cup \{i\},\ I_2 = I_0 \cup \{j\}.$$

We then use Section 6.3.1 to find the true value of D_{ij}.

6.3.2 Multi-Blocks with Facial Vectors, FV, for FR, MBFV

The second approach uses overlapping principal submatrices, matrix blocks, where the overlap is minimal size $\geqslant d+1$ but with embedding dimension d. Therefore, we have a larger number of

principal submatrices but they are significantly smaller. If at anytime we find a block that is not a **EDM**, then we stop and find the noise for this small problem by using submatrices with the correct embedding dimension. We denote this approach **MBFV** .

Rather than using exposing vectors as done above in the bisection approach Section 6.3.1, the **FR** is done by efficiently finding the intersection of two faces corresponding to two blocks by finding the **FV** using the approach in [113, Lemma 2.9]. We find a **FV** for the small blocks using Theorem 2.4.12. We then find the new **FV** that represents the intersection of faces, i.e., the union of blocks, using Lemma 2.4.13. If we end up with a small final block, we use the same strategy as in Section 6.3.1.

Remark 6.3.3. *Notice that once the problem is divided into principal submatrices, it becomes an **EDM** completion problem. But it is clear that we could permute the columns and rows $D_n \leftarrow QD_nQ^T$, where $Q \in \Pi$ is a permutation matrix, before applying the subdivisions and the algorithm. In fact, we could use overlapping cliques in the graph as long as we maintain a chordal structure, as chordalty allows for **EDM** completion, see e.g., [16, 106].*

6.3.3 Equivalent Approach Using Gale Transforms

In this section we present an alternative approach, using Gale transforms that is equivalent to that of **FR** and exposing vectors discussed above. We use the smallest blocks along with Gale transforms and denote this as **SBGT**. We note that the notion of Gale transform [69, 85] is well known and widely used in the theory of polytopes. Our approach reveals interesting relationships between **FR** and Gale transforms. We only consider the multi-block case presented in Section 6.3.2.

Recall that we have points $p_1, \ldots, p_n$ in $\mathbb{R}^d$ and we assume that the affine hull of these points has full dimension d. (In fact after centering as in (2.3), we can assume they are centered and span $\mathbb{R}^d$.) Recall that the $n \times d$ matrix of points P with $P^T = \begin{bmatrix} p_1 & \ldots & p_n \end{bmatrix}$, is called the *configuration matrix* of these points. Note that P has full column rank, and the Gram and **EDM** matrices defined by these points are $G = PP^T$ and $D = \mathcal{K}(G)$, respectively. The *Gale space of D*, $\mathrm{gal}(D)$, is given by

$$\mathrm{gal}(D) = \mathrm{null}\left(\begin{bmatrix} P^T \\ e^T \end{bmatrix}\right) = \mathrm{null}\left(\begin{bmatrix} G \\ e^T \end{bmatrix}\right). \tag{6.17}$$

Any $n \times (n-d-1)$ matrix N such that the columns of N form a basis of $\mathrm{gal}(D)$ is called a *Gale matrix* of D. (This is the matrix N_i for G_i in (6.8).) The i-th row of N is called a Gale transform of p_i. Note that N is not unique. In addition, we recall that a face $f \trianglelefteq \mathbb{S}^n_+$ is characterized by the nullspace or range space of any $X \in \mathrm{relint}(f)$ and $f = \mathrm{face}(X)$, i.e., f is the *minimal face containing* X. We know that the full rank factorization $X = PP^T$ yields a configuration matrix P. Now we note Lemma 6.3.4 that a maximum rank exposing vector $Z \in f^\Delta$ and a full rank factorization $Z = NN^T$ yields a Gale matrix N.

Lemma 6.3.4. *Let $G \in \mathbb{S}^n_+ \cap \mathcal{S}^n_C$[4] be a centered Gram matrix of the **EDM** D. Let Z be a maximum rank centered exposing vector for G as given in (6.11). Equivalently, $Z \in \mathrm{relint}(\mathrm{face}(G)^\Delta \cap \mathcal{S}^n_C)$, where $\cdot^\Delta$ denotes conjugate face. Then any full rank factorization $Z = NN^T$ yields a Gale matrix N of D. Conversely, if N is a Gale matrix of D, then $Z = NN^T$ is a maximum rank centered exposing vector of G.*

[4]The centered subset $\mathcal{S}^n_C$ is defined in (2.5).

Proof. This follows from the definitions in (6.11) and (6.17). □

Recall that a set of points $\{p_1, \ldots, p_k\}$ in $\mathbb{R}^d$ is said to be *affinely independent* if

$$\text{null}\left(\begin{bmatrix} p_1 & \cdots & p_k \\ 1 & \cdots & 1 \end{bmatrix}\right) = \{0\}.$$

It is easy to see that the columns of the Gale matrix N encode the affine dependency of the points $p_1, \ldots, p_n$.

Given $D \in \mathcal{E}^n$ of embedding dimension d, and $j \in [n-d-1]$, let $D_j = D_{[j,j+d+1]}$ be the principal submatrix of D induced by the columns (rows) $[j, j+d+1]$. Therefore, each D_j is an **EDM** of order $d+2$. Furthermore, for $j \in [n-d-2]$, the submatrices D_j and D_{j+1} overlap in $d+1$ columns (rows).

Now suppose we have an incomplete $D \in \mathcal{E}^n$ where only the principal diagonal blocks $D_i, i \in [n-d-1]$ are known, while the entries of D outside these diagonal submatrices are not known or are noisy. The problem addressed in this chapter is how to recover all the entries of D, i.e., how to complete the **EDM**. We now show how to recover D by computing P^0, a configuration matrix of D, using the Gale matrix N. Recall that we have assumed that the points $p_1, \ldots, p_n$ that generate D are in *general position*. We now build N using only information from the diagonal submatrices $D_i, i \in [n-d-1]$. We compare this to (6.8) and see the relation between the Gale matrix and exposing vectors.

Recall that $\mathcal{K}^\dagger(D_j) = -\frac{1}{2}JD_jJ$. Let

$$G_j = \mathcal{K}^\dagger(D_j) \tag{6.18}$$

be the Gram matrix of D_j. Then a Gale matrix N_j corresponding to D_j is the $(d+2)$-vector which forms a basis of[5]

$$\text{null}\left(\begin{bmatrix} G_j \\ e^T \end{bmatrix}\right). \tag{6.19}$$

Moreover, it follows from Lemma 2.3.2 that all the entries of N_j are nonzero. Now a Gale matrix N for the entire matrix D can be built one column at a time[6] as follows: the entries of the jth column of N are zeros except in the positions $i = j, \ldots j+d+1$ which are equal to those of N_j. Obviously, by construction, N is $n \times (n-d-1)$ matrix of full column rank.

Now let V be the $n \times d$ matrix whose columns form a basis of[7]

$$\text{null}\left(\begin{bmatrix} N^T \\ e^T \end{bmatrix}\right).$$

Note that a configuration matrix of D is given by

$$P^0 = VQ, \tag{6.20}$$

for some nonsingular Q of order d, since the columns of both P^0 and V are both bases of the same space. Once we find Q, both P^0 and consequently $D = \mathcal{K}(P^0 P^{0T})$ can be recovered. Note that

[5] See also (6.8) where N_i is not necessarily a single column as the size of G_i can be larger than G_j in (6.19).

[6] This step is equivalent to the summation of exposing vectors in (6.10). It is shown in [61] that the summation reduces noise if the data is random and from a (Gaussian) normal distribution.

[7] This is equivalent to the V found in (6.12).

(6.20) is an overdetermined system. We see next how to find Q by considering only the first $d+2$ equations of (6.20) that we denote by:

$$P_1^0 = V_1 Q. \tag{6.21}$$

Here both P_1^0 and V_1 are $(d+2) \times d$.[8]

Now (6.21) cannot be solved as is since both P_1^0 and Q are unknown. In order to overcome this hurdle, we multiply both sides of (6.21) from the left with J and let $V_1^0 := JP_1^0$. Thus we get

$$V_1^0 = JP_1^0 = JV_1 Q. \tag{6.22}$$

Note that V_1^0 is a configuration matrix corresponding to D_1 and $(V_1^0)^T e_{d+2} = 0$. Recall that D_1 is the principal submatrix of D induced by the rows (columns): $1, \ldots, d+2$. Now let B_1 be the Gram matrix corresponding to D_1. Then V_1^0 be can be found by the full-rank factorization of B_1, i.e., $B_1 = V_1^0 {V_1^0}^T$. Thus, Equation (6.22) can be solved since Q is the only unknown. Then Lemma 6.3.5 shows that (6.21) and (6.22) are equivalent.

.

Lemma 6.3.5. *Let V_1, P_1^0 and V_1^0 be as defined above. Under the general position assumption, consider the two systems in the $d \times d$ variable matrix X*

$$\textit{System I: } V_1 X = P_1^0 \qquad \textit{System II: } JV_1 X = JP_1^0 = V_1^0.$$

Then both systems have the same unique solution.

Proof. First note that, by relabeling the points if necessary, we can assume without loss of generality that P_1^0 has full column rank. Moreover, it follows from (6.21) that rank (V_1) = rank (P_1^0) and rank $([V_1\ e])$ = rank $([P_1^0\ e])$ since $[P_1^0\ e] = [V_1\ e] \begin{bmatrix} Q & 0 \\ 0 & 1 \end{bmatrix}$. Thus, V_1 has full column rank.

Now Q is the unique solution of System I. It is easy to see that the solution of System II is $Q+Y$ where the columns of Y are in null(JV_1). Let y be a nonzero vector in null(JV_1), then $V_1 y = \gamma e$ for some scalar γ. Thus rank $([V_1\ \ e])$ = rank $([P_1^0\ \ e]) = d$ which contradicts Lemma 2.3.2 since we assume that $p^1, \ldots, p^n$ are in general position. Thus null(JV_1) is trivial and the result follows. □

An immediate consequence of Lemma 6.3.5 is that Q in (6.20) can be calculated from System II, for example $Q = ({V_1}^T J V_1)^{-1} {V_1}^T V_1^0$. Consequently, $P^0 = VQ$ and $D = \mathcal{K}(P^0 {P^0}^T)$.

[8] Finding Q is equivalent to solving the system for r in (6.14). In fact, we have the equivalence $R = QQ^T, G = VQQ^TV^T = VRV^T$. Using the smaller system with V_1 would be equivalent to not using the entire overdetermined system in (6.14).

Example Using Gale Transforms

Example 6.3.6. *Consider the following* ***EDM*** D *of embedding dimension* $d = 2$,

$$D = \begin{bmatrix} 0 & 2 & 5 & 9 & 5 & 2 \\ & 0 & 1 & 5 & 5 & 4 \\ & & 0 & 2 & 4 & 5 \\ & & & 0 & 2 & 5 \\ & & & & 0 & 1 \\ & & & & & 0 \end{bmatrix}.$$

(For both the ***EDM*** *and Gram matrices we only provide the upper triangular parts.) And assume that noise is added to the entries* d_{15}, d_{16} *and* d_{26}. *Then the Gram matrix corresponding to* D_1 *is* $G_1 = \frac{1}{2}\begin{bmatrix} 5 & 1 & -2 & -4 \\ & 1 & 0 & -2 \\ & & 1 & 1 \\ & & & 5 \end{bmatrix}$. *Therefore,* V_1^0, *a configuration matrix of* D_1, *is obtained by the full-rank factorization of* G_1, *i.e.,* $V_1^0 = \frac{1}{2}\begin{bmatrix} 1 & -3 \\ -1 & -1 \\ -1 & 1 \\ 1 & 3 \end{bmatrix}$. *Furthermore, a Gale matrix for* D *is*

$N = \begin{bmatrix} -1 & 0 & 0 \\ 3 & -2 & 0 \\ -3 & 3 & -1 \\ 1 & -2 & 2 \\ 0 & 1 & -3 \\ 0 & 0 & 2 \end{bmatrix}$. *Hence,* V, *the matrix whose columns form a basis of* $\text{null}\left(\begin{bmatrix} N^T \\ e^T \end{bmatrix}\right)$, *is*

$V = \begin{bmatrix} 3 & 0 \\ 0 & 1 \\ -2 & 1 \\ -3 & 0 \\ 0 & -1 \\ 2 & -1 \end{bmatrix}$ *and hence* $V_1 = \begin{bmatrix} 3 & 0 \\ 0 & 1 \\ -2 & 1 \\ -3 & 0 \end{bmatrix}$ *and* $JV_1 = \frac{1}{2}\begin{bmatrix} 7 & -1 \\ 1 & 1 \\ -3 & 1 \\ -5 & -1 \end{bmatrix}$. *Hence, solving* (6.22) $V_1^0 = JV_1Q$, *we get* $Q = \frac{1}{2}\begin{bmatrix} 0 & -1 \\ -2 & -1 \end{bmatrix}$ *Therefore,* $P^0 = VQ = \frac{1}{2}\begin{bmatrix} 0 & -3 \\ -2 & -1 \\ -2 & 1 \\ 0 & 3 \\ 2 & 1 \\ 2 & -1 \end{bmatrix}$ *and the full* ***EDM*** D *is recovered by using* $D = \mathcal{K}(P^0 P^{0T})$.

6.4 Empirics and Complexity under General Position Assumption

We generate random problems based on: the number of points n; the embedding dimension d; the magnitude of the noise.[9] Table 6.2 compares three methods as outline in Table 6.1:

method #	3 **D&C** methods	3 **FR** methods
1	2 blocks	exposing vectors
2	min multi-blocks FR	facial vectors
3	max multi-blocks	Gale transforms

Table 6.1: Three combinations of **D&C** and **FR**.

The first uses two blocks and **FR** with exposing vectors; the second uses a minimum number of multiblocks with facial vectors; and the third uses Gale transforms with the maximum number of multiblocks. In this way we illustrate all three **D&C** and all three **FR** methods. The Gale transform method goes naturally with small blocks and so we used the maximum number of blocks. We found that the facial vector goes best with the minimum number of blocks. Our output indicates that all the problems were solved successfully and this follows from the fact that the general position property holds generically. The output includes: the relative error for the accurate **EDM** found; and the time in cpu seconds. We discuss the hard cases below in Section 6.5.

6.4.1 Random Problems

Tables 6.2 and 6.3 (on pages 120, 121) illustrate the high efficiency of the algorithms for speed, accuracy, and size. The noise ϵ was a normal random variable with nonzero absolute value greater than .01. Both the position and a near machine precision accurate value for the noise was found in 100% of the instances.

6.4.2 Complexity Estimates

We now look at theoretical complexity estimate results and compare them to the empirical output. For randomly generated problems, we have plotted dimension versus solution time in Figure 6.1, page 121. This agrees with the estimates in (6.23) to (6.25) for the three methods, respectively:

$$\mathbf{BIEV}, \mathbf{MBFV}, \mathbf{SBGT}: \quad O(n^3), O(n), O(n^3).$$

Note that the expense for generating the random problems is $O(n^2)$ as the main work is the multiplication using the configuration matrix PP^T.

Bisection with Exposing Vectors, BIEV

At each step, we divide an $n \times n$ matrix into two blocks with size $\frac{n+d+2}{2}$ so they overlap in a block of size $d+2$. The most time consuming calculation in each iteration is calculating the spectral

[9] We used MATLAB version R2022b on the two servers at University of Waterloo: biglinux, cpu149.math.private Dell PowerEdge R840 four Intel Xeon Gold 6230 20-core 2.1 GHz (Cascade Lake) 768 GB; and fastlinux, cpu157.math.private Dell PowerEdge R660 Two Intel Xeon Gold 6434 8-core 3.7 GHz (Sapphire Rapids) 256 GB.

Data specifications			**BIEV**		**MBFV**		**SBGT**	
n	d	noise	rel-error	time(s)	rel-error	time(s)	rel-error	time(s)
1000	5	0.289	1.86e-12	0.200	8.44e-14	0.022	3.10e-12	0.121
2000	5	-0.235	1.22e-12	0.916	1.28e-13	0.044	3.84e-13	0.364
3000	5	-0.843	2.33e-12	1.713	3.36e-13	0.130	1.28e-13	0.638
4000	5	0.570	7.01e-13	1.842	3.41e-13	0.195	2.74e-13	1.342
5000	5	0.517	1.10e-12	4.394	8.80e-14	0.292	2.07e-13	1.847
6000	5	0.659	1.30e-12	6.861	3.07e-13	0.414	1.31e-12	2.889
7000	5	0.200	2.16e-12	11.759	7.95e-14	0.631	2.01e-13	3.895
8000	5	-0.240	1.71e-12	10.993	1.12e-13	0.768	2.01e-13	5.338
9000	5	-0.294	4.63e-13	18.939	1.73e-13	0.974	1.54e-13	7.299
10000	5	0.197	2.01e-12	23.177	1.63e-13	1.179	5.51e-13	9.529
11000	5	-0.405	4.42e-12	18.598	7.43e-14	1.383	3.15e-13	11.282
12000	5	0.085	4.99e-12	20.521	3.88e-13	1.732	3.16e-13	14.150
13000	5	0.311	1.42e-12	44.017	3.35e-13	2.097	2.55e-13	17.511
14000	5	-0.390	4.50e-12	53.028	7.69e-14	2.201	2.14e-11	20.961
15000	5	-0.348	5.31e-12	54.837	3.78e-13	2.383	6.69e-12	25.517
16000	5	-0.294	6.14e-12	51.610	1.30e-13	2.780	6.97e-13	29.842
17000	5	0.063	3.08e-12	64.764	2.12e-13	3.176	3.78e-13	33.774
18000	5	-0.064	1.33e-11	92.478	2.20e-13	3.485	1.27e-12	38.898
19000	5	0.001	2.81e-11	99.526	3.89e-13	3.986	2.99e-13	43.750
20000	5	-0.004	9.56e-13	97.926	9.13e-13	4.229	4.21e-13	51.621
21000	5	0.368	1.09e-12	130.590	1.93e-13	4.749	9.99e-13	58.367
22000	5	0.035	1.21e-11	177.391	7.86e-14	5.232	1.34e-13	66.882
23000	5	0.018	5.76e-12	173.445	2.44e-13	5.786	1.06e-12	73.720
24000	5	1.000	2.69e-12	160.173	1.38e-13	5.970	2.18e-13	82.715
25000	5	0.139	4.08e-12	242.229	3.11e-13	6.804	3.07e-12	91.045
26000	5	-0.385	1.91e-12	91.091	1.28e-13	6.946	3.23e-13	102.973
27000	5	-0.131	7.17e-12	173.206	9.43e-14	7.791	2.47e-13	112.248
28000	5	0.022	1.18e-11	245.353	3.61e-13	8.199	2.10e-10	124.508
29000	5	0.109	1.05e-11	264.581	6.11e-13	8.728	3.85e-12	134.517
30000	5	0.089	2.68e-12	299.627	2.58e-13	8.793	7.09e-13	149.871

Table 6.2: Fastlinux; 3 methods: n = 1K to 30K; mean of 3 instances per row

decomposition of the corresponding Gram matrices G_1, G_2 with runtime $2O\left(\left(\frac{n+d+2}{2}\right)^3\right) = O(n^3)$, as $n \gg d$. The number of points outside of the two principal submatrices is $2\left(\frac{n-d-2}{2}\right)^2 = \frac{(n-d-2)^2}{2}$. If the noisy element is outside of the two principal submatrices, the algorithm will terminate in this step. Thus, the probability to continue to another step is

$$1 - \frac{(n-d-2)^2}{2n^2} \approx \frac{1}{2}.$$

If the noisy element is in one of the principal submatrices, then we continue with the divide and conquer algorithm. After the division, the smaller matrix has size $\frac{n+d+2}{2}$, i.e., we reduce the size of problem by approximately half. After i divisions, the size of the matrix is $\frac{n+2^{i-1}(d+2)}{2^i}$. The total runtime of this algorithm is $O(n^3)$. We now drop the d and constants in the analysis as $n \gg d$. Let $T(n)$ be the total runtime of the algorithm, and $f(n)$ be the runtime of one iteration of the subproblem. Then, we have the recurrence

$$T(n) = \frac{1}{2}T\left(\frac{n}{2}\right) + f(n) = \frac{1}{2}T\left(\frac{n}{2}\right) + O(n^3).$$

Data specifications				**MBFV**	
n	d	noise	gen-time	rel-error	time(s)
60000	5	0.436	90.767	1.32e-13	53.049
70000	5	0.026	109.262	3.37e-13	72.006
80000	5	0.550	155.565	3.12e-13	93.529
90000	5	0.435	196.015	8.91e-13	114.084
100000	5	0.420	243.331	9.21e-13	144.727
110000	5	0.330	315.411	3.01e-13	196.053
120000	5	0.205	385.318	7.16e-13	237.335

Table 6.3: BigLinux; Multi-block solver with gen time; mean of 3 instances per row

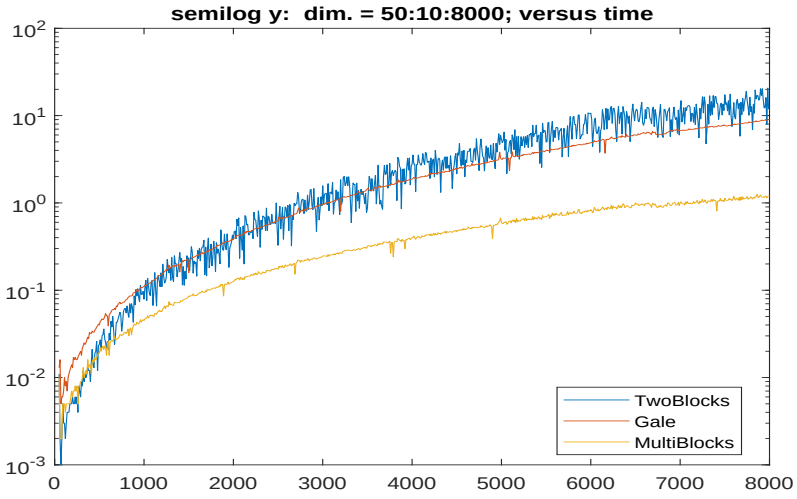

Figure 6.1: Semilogy plot; dimension versus solution time

The $O(n^3)$ term dominates the recursive relationship. We get:

$$\text{total runtime is } O(n^3). \tag{6.23}$$

Multi-Block with Facial Vectors, MBFV

We build up the full facial vector from overlapping small principal submatrices of size $2d+6$. Each overlaps with the previous one with a block of size $d+3$. Solving for the final connecting matrix Q also involves solving a small system of equations using the first block, thus costing a constant time. Finding the facial vector of each small matrix has constant runtime as well, since n is large. The number of small blocks we look at is roughly $\frac{n}{d+2}$. Therefore, we get the

$$\text{total runtime is } O(n). \tag{6.24}$$

Small Blocks with Gale Transform, SBGT

We calculate the Gale matrix of each small principal submatrix of size $d+2$, and then build up the Gale matrix for the entire matrix D. The most time consuming step is in finding $V \in \mathbb{R}^{n\times d}$, the **FV** , whose columns form a basis of null $\left(\begin{bmatrix} N^T \\ e^T \end{bmatrix}\right)$. This has runtime complexity of:

$$\text{total runtime is } O(n^3). \tag{6.25}$$

6.5 Hard Cases; No General Position Assumption

We now consider problems where the general position assumption may not hold. This can lead to hard cases for our algorithm. The hard cases are related to problems where all but 2 points are in a linear manifold of dimension $d-2$. We present examples as well as empirics for the hard cases.

Our modified algorithm for the hard case finds a block with rank d by checking all *consecutive* principal diagonal blocks $D_{[i,i+d]}$ of size $d+1$. If we can find a block with the correct embedding dimension d from the principal diagonal blocks, we can proceed with the Gale algorithm as before. Otherwise, we calculate the Gale matrix

$$N_i = \begin{bmatrix} n_i^T \\ \vdots \\ n_{i+d}^T \end{bmatrix}, \text{ whose columns form a basis for null } \left(\begin{bmatrix} G_i \\ e^T \end{bmatrix}\right),$$

for each small block $D_i := D_{[i,i+d]}$. Since the G_i does not have rank d, the points $p_i, \cdots, p_{i+d}$ are not in general position. We identify the indices where the corresponding Gale transform results in a row of zeros. If $n_j = 0$, then p_j is not in the affine hull of $\{p_i, \cdots, p_{i+d}\}\backslash\{p_j\}$ [5, Section 7.2.1]. We collect these indices in a set I, so $j \in I$ if there is some $i \in \{1, \cdots, n-d\}$ such that $j \in \{i, \cdots, i+d\}$ and $n_j = 0$ in N_i. Additionally, let $I' = \{1, \cdots, n\}\backslash I$. We repeat the process on $D_{I\cup I'(1:d+1)}$ until we have $d+1$ points that are in general position.

There are certain cases where we can not solve the problem, so-called *hard cases*. The results in Section 6.5.1 show that critical to our conclusions is the difficult case when all but two points are in a linear manifold of dimension $d-2$.

Definition 6.5.1 (good, bad blocks)**.** *Let D_n be the noisy **EDM** as above. Let (i,j) be the corrupted index.*

1. *By a good block we mean a principal submatrix $(D_n)_\alpha$, such that the corresponding $\mathcal{K}^\dagger\left((D_n)_\alpha\right)$ is positive semidefinite with rank d.*
2. *By a bad block we mean a principal submatrix $(D_n)_\alpha$, such that the corresponding $\mathcal{K}^\dagger\left((D_n)_\alpha\right)$ is either not positive semidefinite or the rank is greater than d.*
3. *An uncorrupted good block signifies a principal submatrix $(D_n)_\alpha$, such that the corresponding $\mathcal{K}^\dagger\left((D_n)_\alpha\right)$ is positive semidefinite with rank d, and α does not contain either corrupted index i or j.*

Suppose that there exists an *uncorrupted good block*, then we never encounter an unsolvable case. After we found a block D_α of size $d+1$ with embedding dimension d, we calculate the Gale matrix for $D_{\alpha\cup\{i\}}$ for each $i \in \{1,\cdots,n\}\backslash\alpha$. If D_α contains the corrupted entry, then by the result in Section 6.5.1, the only case we cannot recognize $D_{\alpha\cup\{i\}}$ to be a corrupted **EDM** is if all but two points in this set are in a linear manifold of dimension $d-2$. However, under the assumption of the data, there exists $d+1$ points: $\{p_{i_1},\cdots,p_{i_{d+1}}\}$ in general position and the corresponding block is uncorrupted. Therefore, it cannot be that $D_{\alpha\cup\{i_1\}},\cdots,D_{\alpha\cup\{i_{d+1}\}}$ all result in the case of all but two points are in a linear manifold of dimension $d-2$. Thus, if D_α contains the corrupted entry, we can recognize it as not being a proper **EDM**.

6.5.1 Characterizing Good and Bad Blocks; $\mathrm{embdim}(D) = 2$ and Beyond

We know that an **EDM** D with embedding dimension d corresponds to a Gram matrix $G = \mathcal{K}^\dagger(D) \geq 0$ with $\mathrm{rank}(G) = d$. The question that arises is whether, when a single element is perturbed in D, can this corrupted block *always* be verified correctly by checking that there are no longer exactly d nonzero and positive eigenvalues for the corresponding Gram matrix G? More precisely, does a corrupted block always have a negative eigenvalue and/or more than d positive eigenvalues. We show that indeed this can always be detected except when all but two points of a corresponding configuration matrix P lie in a linear manifold of dimension $d-2$. In addition, we show how to handle these cases by extending the notion of *yielding* in [5, Section 7.2]. Note that the 2 in the results arises from the fact that the perturbation matrix ϵE_{ij} is rank 2.

The case when the embedding dimension $d = 2$ is special. We conclude that no failures in detection can occur if the distances are positive, an easy check. For this special case $d = 2$, we do not have to assume that the problem is in general position to recognize a corrupt entry.

In our algorithms we have to identify whether the corrupted element is within a principal submatrix D_α, where for **MBFV** the cardinality $|\alpha| \geqslant d+2$ and is often relatively small $|\alpha| << n$. We do not assume general position, see Lemma 2.3.2. We present the characterization in the general d case in Theorem 6.5.2. This is related to the results on *yielding intervals* in [5, Section 7.2], [6], i.e., characterizations of intervals of perturbations that maintain the **EDM** property but the embedding dimension is not fixed. Also, this is related to the general question of when a matrix pencil is positive semidefinite. The interval for this is studied recently in [134]. Our case is special in that we look at rank two updates

$$D(\epsilon) := D + \epsilon E_{ij} \in \mathcal{E}^n, D \in \mathcal{E}^n,\ G(\epsilon) := G + \epsilon\mathcal{K}^\dagger(E_{ij}) \geq 0, G \geq 0.$$

The specific case of a rank two perturbation as in our case is studied in [33]. However, we have the additional condition that the rank is maintained at d.

Essentially we characterize the cases when the perturbation $D(\epsilon)$ results in a corresponding Gram matrix with the correct rank, thus fooling the algorithm. Recall that blkdiag denotes the block diagonal matrix formed from the arguments.

Theorem 6.5.2. *Let $D \in \mathcal{E}^k$ with $\mathrm{embdim}(D) = d \leqslant k-2$. Suppose that there is a single, nonzero corrupted distance $D_{ij}, i < j$, and the noisy **EDM** is denoted by the singular matrix pencil*

$$D(\epsilon) := D + \epsilon E_{ij},\ \epsilon \in \mathbb{R}\backslash\{0\}.$$

Let $G := \mathcal{K}^\dagger(D) = Q\Lambda Q^T$ *be the Gram matrix with its spectral decomposition and, without loss of generality,*

$$\Lambda = \operatorname{blkdiag}(0, \Lambda_+),\ \Lambda_+ \in \mathbb{S}^d_{++}.$$

Denote $G_E := \mathcal{K}^\dagger(E_{ij}) = -\frac{1}{2} J E_{ij} J$ *and let* $\bar{G}_E$ *be defined and appropriately blocked with size* d*:*

$$\bar{G}_E := Q^T G_E Q = \begin{bmatrix} \bar{G}_{11} & \bar{G}_{12} \\ \bar{G}_{12}^T & \bar{G}_{22}, \end{bmatrix},\ \bar{G}_{22} \in \mathbb{S}^d.$$

Define the open interval I_ϵ *for maintaining* $\Lambda_+ + \epsilon \bar{G}_{22} \succ 0$*, equivalently for maintaining* $I_d + \epsilon \Lambda_+^{-1/2} \bar{G}_{22} \Lambda_+^{-1/2} \succ 0$*, as*

$$I_\epsilon = \left(\frac{-1}{\max\left\{0, \lambda_{\max}\left(\Lambda_+^{-1/2} \bar{G}_{22} \Lambda_+^{-1/2}\right)\right\}}, \frac{-1}{\min\left\{0, \lambda_{\min}\left(\Lambda_+^{-1/2} \bar{G}_{22} \Lambda_+^{-1/2}\right)\right\}} \right) \subseteq (-\infty, +\infty), \quad (6.26)$$

where a 0 *in the denominator results in* $-\infty, +\infty$*, for the left/right bound of the interval, appropriately. Define the condition:*

$$\textbf{\textit{EDM}}\ \textit{condition:} \quad \boxed{\epsilon \in I_\epsilon\ \textit{and}\ \epsilon\left(\bar{G}_{11} - \epsilon \bar{G}_{12}\left(\Lambda_+ + \epsilon \bar{G}_{22}\right)^{-1} \bar{G}_{21}\right) \succeq 0}. \quad (6.27)$$

Then:

1. *The* ***EDM*** *condition* (6.27) *defines the convex yielding interval for* D_{ij}*. More precisely, suppose that the* ***EDM*** *condition* (6.27) *holds. Then:*

$$D(\epsilon) \in \mathcal{E}^k\ \textit{and}\ \operatorname{embdim}(D(\epsilon)) \geqslant d; \quad (6.28a)$$

$$\bar{G}_{11} = 0 \implies D(\epsilon) \in \mathcal{E}^k, \operatorname{embdim}(D(\epsilon)) = d; \quad (6.28b)$$

$$\bar{G}_{11} - \epsilon \bar{G}_{12}\left(\Lambda_+ + \epsilon \bar{G}_{22}\right)^{-1} \bar{G}_{21} = 0 \implies D(\epsilon) \in \mathcal{E}^k, \operatorname{embdim}(D(\epsilon)) = d. \quad (6.28c)$$

2. *Conversely, we have the following necessary conditions for restricted yielding. Suppose that there exists* $\delta \in \mathbb{R}_{++}$ *such that*

$$D(\epsilon) \in \mathcal{E}^k, \operatorname{embdim}(D(\epsilon)) = d,\ \forall \epsilon \in (-\delta, \delta).$$

Then the configuration matrix P *of* D*,* $PP^T = G = \mathcal{K}^\dagger(D)$*, has* $k - 2 \geqslant d$ *points that are in a linear manifold* $\mathcal{L}$ *of dimension* $d - 2$*. Necessarily, the two points* i, j *outside the linear manifold define the corrupted distance* D_{ij}*.*

Proof. We define the *matrix pencil*

$$G(\epsilon) := \mathcal{K}^\dagger(D(\epsilon)) = G + \epsilon G_E.$$

Our results depend on identifying when the perturbed Gram matrix, the matrix pencil, maintains: $G(\epsilon) = G + \epsilon G_E \succeq 0$ with $\operatorname{rank} G(\epsilon) = d$. We use the spectral decompositions and the *Sylvester law*

of inertia. The latter identifies when positive semidefiniteness and rank d are maintained under a congruence.

To begin, we need to consider the eigenpairs for the two nonzero eigenvalues of

$$G_E := \mathcal{K}^\dagger(E_{ij}) = -\frac{1}{2} J E_{ij} J.$$

An orthogonal pair of eigenvectors of E_{ij} is $e_i \pm e_j$. We have the properties $E_{ij} \neq 0, \text{trace}(E_{ij}) = 0$ and

$$\lambda_1(E_{ij}) = 1 > 0 = \lambda_2(E_{ij}) = \ldots = \lambda_{n-1}(E_{ij}) = 0 > \lambda_n(E_{ij}) = -1.$$

Therefore, the above singular congruence $JE_{ij}J = \mathcal{P}_{\mathcal{S}^n_C}(E_{ij})$, by (2.8), implies

$$\lambda_1(G_E) > 0 = \lambda_2(G_E) = \ldots = \lambda_{n-1}(G_E) = 0 > \lambda_n(G_E),$$

i.e., there are exactly two nonzero eigenvalues of G_E; verified as well in the following. Define the two orthogonal vectors

$$e_{ij} := e_i - e_j,\ e_{ijc} := \frac{2}{k} e - (e_i + e_j) \in e^\perp \subset \mathbb{R}^k. \tag{6.29}$$

One can verify that e_{ij} and e_{ijc} are eigenvectors of G_E and the corresponding eigenvalues are:

$$\lambda_1(G_E) = \frac{1}{2} > 0 > \lambda_k(G_E) = \frac{2-k}{2k}. \tag{6.30}$$

Notice that signs of the eigenvalues are strictly positive and strictly negative. The congruences with Q and the interlace theorem for eigenvalues yield $\lambda_{\max}(\bar{G}_{22}) \geqslant 0 \geqslant \lambda_{\min}(\bar{G}_{22})$. The congruence with $\Lambda_+^{-1/2}$ and the definition (6.26) of I_ϵ bring us to $0 \in I_\epsilon$ and we conclude that

$$\begin{array}{rcrcl} \epsilon \in I_\epsilon & \iff & I_d + \epsilon \Lambda_+^{-1/2} \bar{G}_{22} \Lambda_+^{-1/2} & \succ & 0 \\ & \iff & \Lambda_+ + \epsilon \bar{G}_{22} & \succ & 0, \end{array}$$

and that

$$G + \epsilon G_E = Q(\Lambda + \epsilon \bar{G}_E) Q^T \succeq 0 \iff \Lambda + \epsilon \bar{G}_E \succeq 0.$$

1. Assuming (6.27), we have $\Lambda_+ + \epsilon \bar{G}_E \succ 0$. By the Schur complement theorem,

$$\epsilon \left(\bar{G}_{11} - \epsilon \bar{G}_{12} \left(\Lambda_+ + \epsilon \bar{G}_{22} \right)^{-1} \bar{G}_{21} \right) \succeq 0 \iff G(\epsilon) \succeq 0.$$

   Under the assumptions and definitions on $D(\epsilon), I_\epsilon$, and by continuity of eigenvalues and of the linear transformation $\mathcal{K}^\dagger$, we get that the d positive eigenvalues of G are perturbed but stay positive in $G(\epsilon)$, i.e., recalling the assumption that $\text{embdim}(D) = d$, we have

$$\text{rank}(G(\epsilon)) \geqslant \text{rank}(G) = d \implies \text{embdim}(D(\epsilon)) \geqslant d.$$

   This proves (6.28a).

(6.28b) and (6.28c) follow from a Schur complement argument. That is,

$$\begin{aligned}\Lambda + \epsilon\bar{G}_E &= \begin{bmatrix}\epsilon\bar{G}_{11} & \epsilon\bar{G}_{12}\\ \epsilon\bar{G}_{12}^T & \Lambda_+ + \epsilon\bar{G}_{22}\end{bmatrix}\\ &\cong \begin{bmatrix}\epsilon\left(\bar{G}_{11} - \epsilon\bar{G}_{12}\left(\Lambda_+ + \epsilon\bar{G}_{22}\right)^{-1}\bar{G}_{21}\right) & \epsilon\bar{G}_{12}\\ 0 & \Lambda_+ + \epsilon\bar{G}_{22}\end{bmatrix}.\end{aligned}$$

Since $\operatorname{rank}(\Lambda_+ + \epsilon\bar{G}_{22}) = d$, $\bar{G}_{11} - \epsilon\bar{G}_{12}\left(\Lambda_+ + \epsilon\bar{G}_{22}\right)^{-1}\bar{G}_{21} = 0$ implies

$$\operatorname{rank}(\Lambda + \epsilon\bar{G}_E) = \operatorname{rank}(G + \epsilon G_E) = d.$$

Similarly, $\bar{G}_{11} = 0$ leads to $\bar{G}_{12} = 0$ and thus $\operatorname{rank}(G + \epsilon G_E) = d$. Therefore (6.28c) holds.

2. Suppose that there exists $\delta > 0$ such that, for all $|\epsilon| < \delta$, we have $D(\epsilon) \in \mathcal{E}^k$ and $\operatorname{embdim}(D(\epsilon)) = d$. This is equivalent to

$$G + \epsilon G_E \succeq 0 \text{ and } \operatorname{rank}(G + \epsilon G_E) = d.$$

We want to show that

$$G + \epsilon G_E \succeq 0,\ \operatorname{rank}(G + \epsilon G_E) = \operatorname{rank}(G) \implies \operatorname{range}(G_E) \subset \operatorname{range}(G).$$

Recall that $G + \epsilon G_E \succeq 0 \iff \Lambda + \epsilon\bar{G}_E \succeq 0$ and that

$$\Lambda + \epsilon\bar{G}_E \cong \begin{bmatrix}\epsilon\left(\bar{G}_{11} - \epsilon\bar{G}_{12}\left(\Lambda_+ + \epsilon\bar{G}_{22}\right)^{-1}\bar{G}_{21}\right) & \epsilon\bar{G}_{12}\\ 0 & \Lambda_+ + \epsilon\bar{G}_{22}\end{bmatrix}.$$

Since $0 \in I_\epsilon$, we can find small enough $|\epsilon| > 0$, namely $\epsilon \in I_\epsilon \cap (-\delta, \delta)$, so that $\Lambda_+ + \epsilon\bar{G}_{22} \succ 0$. Hence $\operatorname{rank}(\Lambda_+ + \epsilon\bar{G}_{22}) = d$ and note that the rank of the entire matrix is the sum of the ranks of the diagonal blocks. Hence, the top left block after the elimination has to have rank 0, by $\operatorname{rank}(G + \epsilon G_E) = d$ assumption. Therefore,

$$\bar{G}_{11} - \epsilon\bar{G}_{12}\left(\Lambda_+ + \epsilon\bar{G}_{22}\right)^{-1}\bar{G}_{21} = 0. \tag{6.31}$$

(6.31) is violated for small enough $|\epsilon|$ unless $\bar{G}_{11} = 0$. Hence $\bar{G}_{11} = 0$, which also implies $\bar{G}_{12} = 0$. Hence,

$$\operatorname{range}(\bar{G}_E) \subseteq \operatorname{range}(\operatorname{blkdiag}(0, I_d)) = \operatorname{range}(\operatorname{blkdiag}(0, \Lambda_+)) = \operatorname{range}(\Lambda),$$

which is equivalent to

$$\operatorname{range}(G_E) = \operatorname{range}(Q\bar{G}_E Q^T) \subseteq \operatorname{range}(Q\Lambda Q^T) = \operatorname{range}(G).$$

This implies that the facial vector V for $G = \mathcal{K}^\dagger(D)$ must have range that contains the span of the two eigenvectors of $G_E = \mathcal{K}^\dagger(E_{ij})$. Otherwise, at least one of the following happens: (i) we lose positive semidefiniteness; (ii) the number of positive eigenvalues increases. Therefore, we must have

$$\operatorname{range}\left(\begin{bmatrix}e_{ij} & e_{ijc}\end{bmatrix}\right) \subset \operatorname{range}(V), \quad V = \begin{bmatrix}e_{ij} & e_{ijc} & \bar{V}\end{bmatrix}, \quad \begin{bmatrix}e_{ij} & e_{ijc}\end{bmatrix}^T \bar{V} = 0. \tag{6.32}$$

Thus, the ℓ-th row of V is

$$V(\ell,:) = \begin{bmatrix} 0 & \frac{2}{k} & \bar{V}(\ell,:) \end{bmatrix} \in \{0\} \times \{2/k\} \times \mathbb{R}^{d-2},$$

for all $\ell \neq i, j$. This shows that for $R = I_d$, all but two points are in a manifold of dimension $d-2$. This means the same is true for general $R \in \mathbb{S}^d_{++}$. Without loss of generality, let $i = n-1, j = n$. Then,

$$V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \in \mathbb{R}^{k \times d}, V_1 \in \mathbb{R}^{(k-2)\times d}, V_2 \in \mathbb{R}^{2\times k}.$$

Note that rows of V_1 are in a $d-2$ dimensional manifold if, and only if, $\operatorname{rank} V_1 \leqslant d-2$. Since $R > 0$, $V_1 R^{1/2}$ maintains the same rank as V_1, i.e., the rows of $V_1 R^{1/2}$ are in a manifold of dimension $d-2$ where

$$\begin{aligned} G &= VRV^T = [VR^{1/2}][VR^{1/2}]^T \\ &= \begin{bmatrix} V_1 R^{1/2} \\ V_2 R^{1/2} \end{bmatrix} \begin{bmatrix} V_1 R^{1/2} \\ V_2 R^{1/2} \end{bmatrix}^T. \end{aligned}$$

□

Corollary 6.5.3. *Let $D \in \mathcal{E}^k$ with $\operatorname{embdim}(D) = 2 < k-1$. Suppose that there is a corrupted position $i < j$ and the noisy **EDM** is*

$$D_n = D + \epsilon E_{ij},\ \epsilon \neq 0. \tag{6.33}$$

If $D_n \in \mathcal{E}^k, \operatorname{embdim}(D_n) \leqslant d$, and $|\epsilon|$ is sufficiently small, then the (centered) configuration matrix P of D, $PP^T = G = \mathcal{K}^\dagger(D)$, has (at least) $k-2 \geqslant 2$ points that are equal, i.e., the rows $P_{s:} = p \in \mathbb{R}^d, \forall s \in [k], s \neq i, s \neq j$. Thus

$$D_{st} = (D_n)_{st} = 0, \forall s, t \in [k], s \neq i, t \neq j.$$

Proof. We apply Theorem 6.5.2, Item 2 for this special $d = 2$ case with $|\epsilon|$ small to guarantee we are within the interval. □

6.5.2 Example of Hard Case; Multiple Solutions

The following is an example where we have determined a *bad* block but where we cannot determine the *position* of the entry that is corrupted. Consider the following corrupted **EDM** in embedding dimension 3.

$$D(18) := D_n = \begin{bmatrix} 0 & 4 & 16 & 8 & 6 & 14 \\ 4 & 0 & 4 & 4 & 6 & 6 \\ 16 & 4 & 0 & 8 & 14 & 6 \\ 8 & 4 & 8 & 0 & \boxed{18} & 14 \\ 6 & 6 & 14 & \boxed{18} & 0 & 20 \\ 14 & 6 & 6 & 14 & 20 & 0 \end{bmatrix} \quad G_n = \mathcal{K}^\dagger(D_n);\ \lambda(G(18)) = \lambda(G_n) \approx \begin{pmatrix} -0.7252 \\ 0.0000 \\ 0.0000 \\ 4.7157 \\ 7.4003 \\ 13.2759 \end{pmatrix}.$$

This is not an **EDM** , as its corresponding $G_n = \mathcal{K}^\dagger(D_n)$ is not positive semidefinite. However, there is more than one way to change only one entry of this matrix and obtain an **EDM** . We can change the 18 in the $(4,5)$ entry in two different ways, to 14 and 6/5, respectively, and get **EDM**s $D_1 := D(14), D_2 := D(6/5)$, with corresponding positive semidefinite $G(\alpha) = \mathcal{K}^\dagger(D(\alpha))$:

$$\lambda(G(14)) \approx \begin{pmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 4.876894 \\ 6.000000 \\ 13.123106 \end{pmatrix}, \qquad \lambda\left(G\left(\frac{6}{5}\right)\right) \approx \begin{pmatrix} 0.000000 \\ 0.000000 \\ 0.000000 \\ 1.562857 \\ 6.189609 \\ 14.114201 \end{pmatrix}.$$

However, if we use a value in the middle of the changes $[6/5, 14]$, e.g., $D(5)$, then the corresponding $G(5)$ is indeed a Gram matrix but has $4 > d$ positive eigenvalues. This corresponds with Theorem 6.5.2 and means that we have the end points of a yielding interval where $D(6/5) + \epsilon E_{45} \in \mathcal{E}^6$ if, and only if, $\epsilon \in [0, 12.8]$. This also means that the two eigenvectors of $G_E = \mathcal{K}^\dagger(E_{45})$ are not in range$(G(6/5)$ or range$(G(14))$. In fact, by checking the rank of $[G\ v]$, with v one of the eigenvectors for G_E, we can see that one eigenvector is in the range while the other is not, thus explaining the finite yielding intervals. The example can be extended to higher embedding dimension using yelding intervals formed with the eigenvectors in the appropriate range.

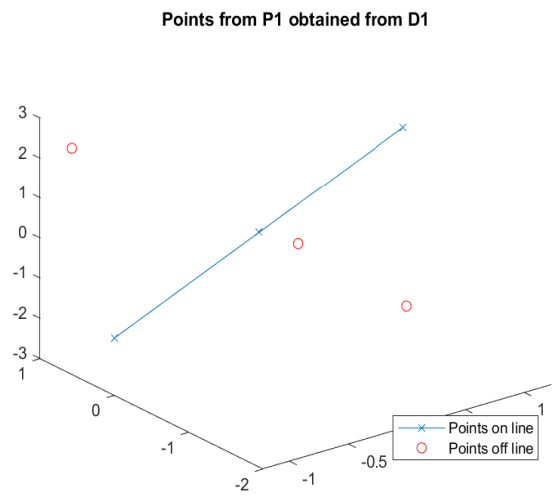

Figure 6.2: Three points off the line

Or we can change the $(4,6)$ entry in 2 different ways and obtain a proper **EDM**

$$D_3 = \begin{bmatrix} 0 & 4 & 16 & 8 & 6 & 14 \\ 4 & 0 & 4 & 4 & 6 & 6 \\ 16 & 4 & 0 & 8 & 14 & 6 \\ 8 & 4 & 8 & 0 & 18 & \boxed{\frac{42}{5}} \\ 6 & 6 & 14 & 18 & 0 & 20 \\ 14 & 6 & 6 & \boxed{\frac{42}{5}} & 20 & 0 \end{bmatrix} \quad D_4 = \begin{bmatrix} 0 & 4 & 16 & 8 & 6 & 14 \\ 4 & 0 & 4 & 4 & 6 & 6 \\ 16 & 4 & 0 & 8 & 14 & 6 \\ 8 & 4 & 8 & 0 & 18 & \boxed{2} \\ 6 & 6 & 14 & 18 & 0 & 20 \\ 14 & 6 & 6 & \boxed{2} & 20 & 0 \end{bmatrix}$$

Similarly, changing the $(5,6)$ entry can also make D_n into an **EDM**

$$D_5 = \begin{bmatrix} 0 & 4 & 16 & 8 & 6 & 14 \\ 4 & 0 & 4 & 4 & 6 & 6 \\ 16 & 4 & 0 & 8 & 14 & 6 \\ 8 & 4 & 8 & 0 & 18 & 14 \\ 6 & 6 & 14 & 18 & 0 & \boxed{14} \\ 14 & 6 & 6 & 14 & \boxed{14} & 0 \end{bmatrix} \quad D_6 = \begin{bmatrix} 0 & 4 & 16 & 8 & 6 & 14 \\ 4 & 0 & 4 & 4 & 6 & 6 \\ 16 & 4 & 0 & 8 & 14 & 6 \\ 8 & 4 & 8 & 0 & 18 & 14 \\ 6 & 6 & 14 & 18 & 0 & \boxed{6} \\ 14 & 6 & 6 & 14 & \boxed{6} & 0 \end{bmatrix}$$

This situation occurs in general when all but 3 points are in a manifold of dimension $d-2$. In this case, points 1-3 are on a line in $\mathbb{R}^3$, and point 4-6 are off the line.
For example, the configuration of D_1, see Figure 6.2, page 128, is

$$P_1 = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \\ -2 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & -2 & -1 \end{bmatrix}$$

If we perturb the distance between points 4 and 5, then we no longer have a **EDM** . However, if we ignore the existence of point 6, then all but 2 points are on a manifold of dimension $d-2$ so the $(4,5)$ entry is yielding. Thus, the submatrix formed by points 1 to 5 is still a proper **EDM** . Thus we can fix the corrupted **EDM** by changing the $(4,6)$ entry instead.

6.5.3 Empirics for the Hard Case

We now generate random hard problems where some points are generated to be on a linear manifold of dimension less than d. Thus, many points are not in general position. Under the Data specification columns, the fourth column indicates the number of points not in general position. The fifth column indicate the dimension of manifold these points live in. We run the problems with $n = 100$ and d from 2 to 10. For each embedding dimension, we run 50 problems. Our output demonstrate the algorithm works well for the hard problems.

Data specifications					Hard Gale Transform with Multi Blocks		
n	d	noise	# pts not in general position	dim of pts	tol-attained	rel-error	time(s)
100	2	0.016	79	1	100%	1.64e-12	0.058
100	3	0.071	86	1	100%	1.62e-13	0.050
100	4	-0.472	82	3	100%	5.08e-13	0.034
100	5	-0.328	78	3	100%	6.98e-13	0.032
100	6	0.185	89	1	100%	1.08e-12	0.048
100	7	0.334	91	1	100%	3.09e-12	0.045
100	8	-0.193	78	2	100%	3.98e-12	0.041
100	9	0.394	76	8	100%	9.31e-12	0.028
100	10	0.222	83	5	100%	1.91e-11	0.047

Table 6.4: Average of 50 hard problems; hard problems where many point are generated to be on a manifold of dimension less than the embedding dimension d

Part III

Conic Programming

Chapter 7

Projection onto Spectrahedra

7.1 Introduction

Facial reduction, **FR**, is a finite step process that regularizes convex programs where the strict feasibility constraint qualification, **CQ**, fails. This CQ holds generically for linear conic programs, see e.g., [64]. However, failure is pervasive in applications such as semidefinite programming, **SDP**, relaxations of hard discrete optimization problems, e.g., [63]. The minimum number of **FR** steps is denoted as the *singularity degree of* $\mathcal{F}$, $\mathrm{sd}(\mathcal{F})$, of the program with feasible set $\mathcal{F}$. It has been shown to be related to stability, error analysis, and convergence rates, see e.g., [55, 62, 162, 166]. Further generalized notions of singularity degree such as the maximum number of **FR** steps are studied in [100, 103] and shown to also relate to stability and convergence rates. In this chapter we study $\mathrm{sd}(\mathcal{F})$ and relations to the projection problem, or best approximation problem (**BAP**), onto a *spectrahedron*, the intersection of a linear manifold and the positive semidefinite cone in symmetric matrix space.

Our main purpose is to examine the effect of failure of strict feasibility on the projection problem. In the absence of strict feasibility, we find surprising relationships between the eigenpairs of small eigenvalues of the Jacobian in a newly proposed Newton method for the projection problem and finding exposing vectors for **FR**. Furthermore, we provide a characterization that links the degeneracy of a point $\bar{X}$ in the spectrahedron, which pertains solely to the feasible set, to the singularity of the Jacobian computed in the course of the Newton method applied to the **BAP**, where $\bar{X}$ is the projected point. This establishes a connection between the degeneracy and the behaviour of the Newton method. We apply the results, both theoretically and empirically, to the problem of finding nearest points to the sets of: (i) correlation matrices or the elliptope; and (ii) semidefinite relaxations of permutation matrices or the vontope, i.e., the feasible sets for the semidefinite relaxations of the max-cut and quadratic assignment problems, respectively.

7.1.1 Projection Problem

Let the *data,* $W \in \mathbb{S}^n$, be given. The projection, or basic *best approximation problem,* ***BAP***, is

$$\begin{array}{rccll} X^* = & \operatorname{argmin} & \frac{1}{2}\|X - W\|^2, & & p^* = \frac{1}{2}\|X^* - W\|^2, \\ & \text{s.t.} & X \in \mathcal{F} := \mathcal{L} \cap \mathbb{S}^n_+, & & \end{array} \tag{7.1}$$

where $\mathcal{L} \subseteq \mathbb{S}^n$ is a linear manifold; and, p^*, X^* are the optimal value and optimum, respectively. The representation of the linear manifold is essential in algorithms and different representations can result in different stability properties for the problem, e.g., [171]. We let $\mathcal{L} = \{X \in \mathbb{S}^n : \mathcal{A}X = b\}$, where $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a given surjective (without loss of generality) linear transformation; $\mathcal{A}X = (\operatorname{trace} A_i X) \in \mathbb{R}^m$ for given fixed linearly independent $A_i \in \mathbb{S}^n, i = 1, \ldots, m$. We assume that $\mathcal{F}$ is a nonempty feasible set; it is called a *spectrahedron*. Here the data of **BAP** is $W, \mathcal{A}, b$.

Nearest point problems are pervasive in the literature and are often the essential step in feasiblity seeking problems, e.g., [20, 36, 127]. We study these problems and show that they reveal hidden structure and information about the stability and conditioning of feasible sets and the degeneracy of optimal points. Related convergence analysis and new types of singularity degree are given in [62, 103]. Recall that a *correlation matrix* is a positive semidefinite matrix with diagonal all one. The set of correlation matrices is often called the *elliptope*. Finding the nearest correlation matrix is one application [27, 94, 95] that arises in many areas, e.g., finance. In addition, we specifically look at the feasible set of the **SDP** relaxation of the quadratic assignment problems **QAP**, which we call the *vontope*. We characterize degeneracy of nearest points and the resulting effects on stability of the nearest point algorithm for these two special instances.

Related Results

The **BAP** for the polyhedral case is studied in [35] with application to linear programming. (In the linear programming, **LP**, case, $\mathbb{S}^n \leftarrow \mathbb{R}^n, \mathbb{S}^n_+ \leftarrow \mathbb{R}^n_+$.) Generalized Jacobians play a critical role, though the relation to stability is not studied. The **SDP** case is studied in e.g., [90, 122]. They use a quasi-Newton method to solve a dual problem similar to our dual problem; though we use a regularized semismooth Newton method with a generalized Jacobian and illustrate fast quadratic convergence for well-posed problems. Further related results on spectral functions, projections, and Jacobians, appear in [123].

In [101] it is shown that *any* conic program that fails strict feasibility has implicit redundancies and every point is degenerate. Relationships with the Barvinok-Pataki bound and strengthened bound [19, 102, 143] for conic programs is discussed. Further discussions on degeneracy related to loss of strict complementarity appear in [49].

The paper [62] provides a sublinear upper bound based on the singularity degree for the convergence rate of the method of alternating projections applied to spectrahedra. The paper [135] furnishes analytic formulas for the sequence generated by alternative projection that reveal that this upper bound can fail to be tight. Although this work is not restricted to the singleton case, much of the analysis is developed with emphasis on the singleton setting. Further results on accuracy and differentiability appear in [78, 123].

7.1.2 Outline

Section 7.2 presents the equation to be solved using the semismooth Newton root finding method, which is derived through a detailed examination of the optimality conditions along with notions on facial structure. We also derive a useful formula for the projection onto a *face* of the semidefinite cone. In Section 7.3 we introduce the semismooth Newton method for the **BAP** and derive Jacobians associated with the Newton method. In Section 7.4, we investigate various phenomena arising in the absence of strict feasibility. This includes the relationships between the eigenpairs

of small eigenvalues of the Jacobian in the Newton method and identification of exposing vectors for **FR** (Section 7.4.1); the singularity of the Jacobian and its link to degeneracy (Section 7.4.2); and applications to the set of correlation matrices and the feasible set of the **SDP** relaxations of the **QAP** (Section 7.4.3). We present numerical experiments in Section 7.5 with results and observations discussed in Section 7.4.

7.2 Characterization of Optimality for the BAP

We now study the optimality conditions of the **BAP** (7.1) and present several properties, including an equation for the application of Newton's method.

Theorem 7.2.1. *Consider the projection problem (7.1). Then the following hold:*

(i) p^* *is finite and the optimum* X^* *exists and is unique.*

(ii) There is a zero duality gap between the primal and the dual problem of (7.1), where the Lagrangian dual is the maximization of the dual functional, $\phi(y, Z)$*, i.e.,*

$$p^* = d^* \quad = \quad \max_{Z\geq 0, y\in\mathbb{R}^m} -\tfrac{1}{2}\|Z + \mathcal{A}^*y\|^2 + \langle y, b - \mathcal{A}W\rangle - \langle Z, W\rangle. \tag{7.2}$$

(iii) Strong duality (zero duality gap and dual attainment) holds in (7.1) if, and only if, there exists a root $\bar{y}, F(\bar{y}) = 0$*, of the function*

$$F(y) := \mathcal{A}P_{\mathbb{S}^n_+}(W + \mathcal{A}^*y) - b. \tag{7.3}$$

Moreover, in this case the solution to the primal problem is given by

$$X^* = P_{\mathbb{S}^n_+}(W + \mathcal{A}^*\bar{y}).$$

Proof. Item (i): The primal problem (7.1) is the minimization of a strongly convex function over a nonempty closed convex set. This yields that the optimal value is finite and is attained at a unique point.

Item (ii): Since the primal objective function is coercive, there is a zero duality gap, see e.g., [14, Theorem 5.4.1]. Let $Z \geq 0$. The Lagrangian function of problem (7.1) and its gradient are given by

$$L(X, y, Z) = \frac{1}{2}\|X - W\|^2 + \langle y, b - \mathcal{A}X\rangle - \langle Z, X\rangle, \quad \nabla_X L(X, y, Z) = X - W - \mathcal{A}^*y - Z.$$

It follows that X is a stationary point of the Lagrangian if

$$X = W + \mathcal{A}^*y + Z. \tag{7.4}$$

By means of this equality, we can eliminate X and write the Lagrangian dual as the following

maximization problem in Z, y.

$$\begin{aligned} d^* &= \max_{Z\succeq 0, y} \min_X L(X, y, Z) \\ &= \max_{Z\succeq 0, y} \min_X \frac{1}{2}\|X - W\|^2 + \langle y, b - \mathcal{A}X\rangle - \langle Z, X\rangle \\ &= \max_{Z\succeq 0, y, X} \left\{ \frac{1}{2}\|X - W\|^2 + \langle y, b - \mathcal{A}X\rangle - \langle Z, X\rangle : \nabla_X L(X, y, Z) = 0 \right\} \\ &= \max_{Z\succeq 0, y} -\frac{1}{2}\|Z + \mathcal{A}^* y\|^2 + \langle y, b - \mathcal{A}W\rangle - \langle Z, W\rangle. \end{aligned}$$

Item (iii): Let $\bar{X}$ be the unique optimal solution, as found by the above. Then strong duality holds if, and only if, there exists $(\bar{y}, \bar{Z})$ such that the following *Karush-Kuhn-Tucker,* ***KKT*** conditions hold:

$$\begin{array}{lll} \bar{X} - W - \mathcal{A}^*\bar{y} - \bar{Z} = 0, & \bar{Z} \succeq 0, & \text{(dual feasibility)}, \\ \mathcal{A}\bar{X} - b = 0, & \bar{X} \succeq 0, & \text{(primal feasibility)}, \\ \langle \bar{Z}, \bar{X}\rangle = 0, & & \text{(complementary slackness)}. \end{array} \tag{7.5}$$

Note that the complementary slackness condition and the fact that $\bar{X}, \bar{Z} \succeq 0$ yield

$$P_{\mathbb{S}^n_+}(W + \mathcal{A}^*\bar{y}) = \bar{X} \text{ and } P_{\mathbb{S}^n_-}(W + \mathcal{A}^*\bar{y}) = -\bar{Z}, \tag{7.6}$$

due to $\bar{X} + (-\bar{Z}) = W + \mathcal{A}^*\bar{y}$ being the Moreau decomposition. Finally, substituting $\bar{X} = P_{\mathbb{S}^n_+}(W + \mathcal{A}^*\bar{y})$ in the primal feasibility condition, we conclude that the **KKT** conditions imply $F(\bar{y}) = \mathcal{A}P_{\mathbb{S}^n_+}(W + \mathcal{A}^*\bar{y}) - b = 0$.

Conversely, directly from the Moreau decomposition theorem, if we are given some $\bar{y} \in Y$ satisfying $F(\bar{y}) = 0$, then the tuple $(\bar{X}, \bar{y}, \bar{Z})$, with $\bar{X}$ and $\bar{Z}$ defined as in (7.6), satisfies the above **KKT** conditions. □

Theorem 7.2.1 (iii) shows how to obtain a solution to the primal problem (7.1) from a root of F. In addition, the pair $(\bar{y}, \bar{Z})$, with

$$\bar{Z} = -P_{\mathbb{S}^n_-}(W + \mathcal{A}^*\bar{y}),$$

is a dual optimal solution of the dual problem (7.2). This fact immediately follows from the proof of Theorem 7.2.1 (iii), where we showed that the tuple $(\bar{X}, \bar{y}, \bar{Z})$ satisfies the **KKT** conditions of problem (7.1).

7.2.1 Regularization for Strong Duality

The projection problem (7.1) always admits a solution given that the feasible set $\mathcal{F}$ is nonempty. If the dual (7.2) of (7.1) has an optimal solution, one can verify that the system given by the function in (7.3), i.e.,

$$F(y) = \mathcal{A}\left(P_{\mathbb{S}^n_+}(W + \mathcal{A}^* y)\right) - b$$

has a root $y \in \mathbb{R}^m$. However, when (7.3) does not have a root, then strong duality fails. We elaborate on this pathology further in Section 7.4.1. One way to guarantee that dual optimal set is

nonempty is to *regularize* (7.1) using **FR**. In Theorem 7.2.2 below, we list some properties induced by **FR** properties that guarantee strong duality.

Theorem 7.2.2. *Consider the projection problem (7.1) with data $W, \mathcal{A}, b$. Denote $f := \text{face}(\mathcal{F}, \mathbb{S}^n_+)$, the minimal face of $\mathbb{S}^n_+$ containing $\mathcal{F}$. Let $\hat{X} \in \text{relint } f$ and let V be a full column rank facial range vector with orthonormal columns with $\text{range } V = \text{range } \hat{X}$, $\text{rank}(V) = r$. Let $\overline{W} = V^T W V \in \mathbb{S}^r$. Define the linear transformation*

$$\mathcal{V}(R) := VRV^T, \ R \in \mathbb{S}^r.$$

Let $\bar{\mathcal{A}}, \bar{b}$ define the affine constraints obtained from $(\mathcal{A} \circ \mathcal{V})(\cdot), b$ after deleting redundant constraints. Then the following hold:

(i) A facially reduced problem of (7.1) in the original space $\mathbb{S}^n$ is

$$\begin{array}{rlll} X^*(W) := & \text{argmin} & \frac{1}{2}\|X - W\|^2 & \\ & s.t. & \mathcal{A}X = b, \ X \in f & (X \succeq_f 0, \ f \trianglelefteq \mathbb{S}^n_+). \end{array} \tag{7.7}$$

*The **KKT** conditions hold at $X^*(W)$ with optimal dual pair $y^* \in \mathbb{R}^m, Z^* \in f^+$.*

(ii) A facially reduced problem of (7.1) in the smaller space $\mathbb{S}^r$ with surjective constraint $\bar{\mathcal{A}} : \mathbb{S}^r_+ \to \mathbb{R}^{\bar{m}}$ is

$$\mathcal{V}^\dagger(X^*(W)) = R^*(\overline{W}) := \text{argmin} \left\{ \tfrac{1}{2}\|R - \overline{W}\|^2 \ : \ \bar{\mathcal{A}}R = \bar{b}, \ R \in \mathbb{S}^r_+ \right\}, \tag{7.8}$$

*where we denote $\mathcal{V}^\dagger$ for the Moore-Penrose generalized inverse of $\mathcal{V}$.[1] The **KKT** conditions hold at $R^*(\overline{W})$ with optimal dual pair $y^* \in \mathbb{R}^{\bar{m}}, Z^* \in \mathbb{S}^r_+$.*

*(iii) Strong duality holds for the **FR** problems (7.7) and (7.8). Moreover:*

$$X^* = \mathcal{V}(R^*(\overline{W})) \text{ solves the original problem (7.1);}$$

*and, $\bar{R} = R^*(\overline{W})$ is a solution to the **FR** primal problem (7.8) if, and only if,*

$$\bar{R} = P_{\mathbb{S}^r_+}(\overline{W} + \bar{\mathcal{A}}^* \bar{y}),$$

where $\bar{y}$ is a root of the function

$$\bar{F}(y) := \bar{\mathcal{A}} P_{\mathbb{S}^r_+}(\overline{W} + \bar{\mathcal{A}}^* y) - \bar{b}. \tag{7.9}$$

Equivalently, $X^*(W)$ is a solution to the original primal problem (7.1) if, and only if, there exists $\hat{y}$ such that

$$0 = F_f(\hat{y}) := \mathcal{A} P_f(W + \mathcal{A}^* \hat{y}) - b, \quad X^*(W) = P_f(W + \mathcal{A}^* \hat{y}), \tag{7.10}$$

where P_f stands for the projection onto $f := \text{face}(\mathcal{F}, \mathbb{S}^n_+)$ given by:

$$P_f(u) = V \left(P_{\mathbb{S}^r_+}(V^T(u)V) \right) V^T \quad \left(= \mathcal{V} \left(P_{\mathbb{S}^r_+} \mathcal{V}^*(u) \right), \quad \mathcal{V}^* \mathcal{V} = I \right). \tag{7.11}$$

[1] The Moore-Penrose generalized inverse of a linear transformation $\mathcal{V}$ is the unique linear transformation satisfying the four Penrose equations, e.g., [21].

Proof. The proof for Items (i) and (ii) follows from the regularization in [28]. Recall that the **KKT** conditions for conic programming problems are:

$$\begin{array}{lll} X - W - \mathcal{A}^* y - Z = 0, & Z \succeq_{f^+} 0, & \text{(dual feasibility)}, \\ \mathcal{A}X - b = 0, & X \succeq_f 0, & \text{(primal feasibility)}, \\ \langle Z, X \rangle = 0, & & \text{(complementary slackness)}. \end{array} \tag{7.12}$$

The first-order optimality follows directly from [28, Theorem 6.1] by taking $\Omega = \mathbb{S}^n$, $S = 0_n \times (-\mathbb{S}^n_+)$, where $0_n \in \mathbb{R}^n$ denotes the vector of zeros, and $g : \Omega \to \mathbb{R}^n \times \mathbb{S}^n$ defined by

$$g(X) = \begin{pmatrix} b - \mathcal{A}x \\ -X \end{pmatrix}.$$

Item (ii) follows from the substitution $X \leftarrow VRV^T$. We note that the number of variables in the objective function reduces since V has orthonormal columns and the norm is orthogonally invariant.

Item (iii): We first show the *elegant* projection formula (7.11). To show that the expression for $P_f(u)$ solves the nearest point problem defined as $P_f(u) = \arg\min_{v \in f} \frac{1}{2}\|v - u\|^2$, we now verify the optimality conditions [96, Theorem 3.1.1.]:

$$\operatorname{trace}\{(P_f(u) - u)(x - P_f(u))\} \geqslant 0, \ \forall x \in f,$$

i.e., for each $x \in f$, there is $R \in \mathbb{S}^r_+$ such that $x = VRV^T$ and thus,

$$\begin{array}{rl} & \operatorname{trace}\Big\{(V(P_{\mathbb{S}^r_+}(V^T(u)V))V^T - u)(x - V(P_{\mathbb{S}^r_+}(V^T(u)V))V^T)\Big\} \\ = & \operatorname{trace}\Big\{(V(P_{\mathbb{S}^r_+}(V^T(u)V))V^T - u)(VRV^T - V(P_{\mathbb{S}^r_+}(V^T(u)V))V^T)\Big\} \\ = & \operatorname{trace}\Big\{V(P_{\mathbb{S}^r_+}(V^T(u)V))V^TVRV^T + uV(P_{\mathbb{S}^r_+}(V^T(u)V))V^T \\ & \quad -V(P_{\mathbb{S}^r_+}(V^T(u)V))V^TV(P_{\mathbb{S}^r_+}(V^T(u)V))V^T - uVRV^T\Big\} \\ = & \operatorname{trace}\Big\{(P_{\mathbb{S}^r_+}(V^T(u)V))(V^TVRV^TV) + (V^TuV)(P_{\mathbb{S}^r_+}(V^T(u)V)) \\ & \quad -(P_{\mathbb{S}^r_+}(V^T(u)V))(P_{\mathbb{S}^r_+}(V^T(u)V)) - uVRV^T\Big\} \\ = & \operatorname{trace}\Big\{(P_{\mathbb{S}^r_+}(V^T(u)V))(R) + (V^TuV)(P_{\mathbb{S}^r_+}(V^T(u)V)) \\ & \quad -(P_{\mathbb{S}^r_+}(V^T(u)V))(P_{\mathbb{S}^r_+}(V^T(u)V)) - V^TuVR\Big\} \\ = & \operatorname{trace}\Big\{(P_{\mathbb{S}^r_+}(V^TuV) - (V^TuV))(R - P_{\mathbb{S}^r_+}(V^TuV))\Big\} \geqslant 0, \end{array}$$

where the last inequality comes from the projection. This completes the proof of (7.11) and establishes the formula for the projection onto the face.

We continue to study the case where the CQ (strict feasibility) fails. With P being the projection that makes the linear transformation onto, we get (7.9) is equivalent to:

$$\begin{array}{rcl} \bar{F}(y) & = & \bar{\mathcal{A}} P_{\mathbb{S}^r_+}(\overline{W} + \bar{\mathcal{A}}^* y) - \bar{b} \\ & = & (P\mathcal{A} \circ \mathcal{V}) P_{\mathbb{S}^r_+}(V^TWV + (P\mathcal{A} \circ \mathcal{V})^* y) - Pb, \text{ for given data } W \\ & = & (P\mathcal{A} \circ \mathcal{V}) P_{\mathbb{S}^r_+}(V^TWV + (\mathcal{V}^* \circ \mathcal{A}^* P^T)(y)) - Pb \\ & = & (P\mathcal{A}) \left(V \left[P_{\mathbb{S}^r_+}(V^T(W + \mathcal{A}^* P^T y)V)\right] V^T\right) - Pb \\ & = & (P\mathcal{A}) \left(P_f(W + \mathcal{A}^* P^T y)\right) - Pb \\ & = & (P\mathcal{A}) \left(P_f(W + (P\mathcal{A})^* y)\right) - Pb, \end{array}$$

where we have used the elegant formula (7.11).

This shows that we can work in the original space if we have done facial reduction. Moreover,

$$P_f(W + \mathcal{A}^* P^T y) = \mathcal{V}\left[P_{\mathbb{S}^r_+}(\mathcal{V}^*(W + \mathcal{A}^* P^T y))\right].$$

Recall that $V^T V = I$. In summary, necessity of (7.9) is clear. Therefore necessity of (7.10) follows from

$$\begin{aligned} 0 &= \bar{\mathcal{A}} P_{\mathbb{S}^r_+}(\overline{W} + \bar{\mathcal{A}}^* y) - \bar{b} \\ &= (P\mathcal{A})\mathcal{V} P_{\mathbb{S}^r_+}(\mathcal{V}^*(W + \mathcal{A}^* P^T y)) - Pb \\ &= (P\mathcal{A}) P_f(W + \mathcal{A}^* P^T y) - Pb. \end{aligned}$$

We can remove P in the last line and ignore the redundant constraints. □

We emphasize that strong duality holds in Theorem 7.2.1, Item (iii), only when there is a dual optimal solution that attains the dual optimal value d^*. However, strong duality is *guaranteed* to hold in Theorem 7.2.2, Item (iii), as a result of the regularization.

Remark 7.2.3. *The proof of Theorem 7.2.2 above provides the following elegant formula for the projection of $u \in \mathbb{S}^n$ onto the face $f = V\mathbb{S}^r_+ V^T, V^T V = I$,*

$$\boxed{P_f(u) = V\left(P_{\mathbb{S}^r_+}(V^T(u)V)\right)V^T = \mathcal{V}\left(P_{\mathbb{S}^r_+}\mathcal{V}^*(u)\right), \quad \mathcal{V}^*\mathcal{V} = I,} \tag{7.13}$$

i.e., the work of finding the projection onto the face f is reduced to the well-known projection onto the smaller dimensional proper cone $\mathbb{S}^r_+$.

We now consider dual optimal sets

$$\mathcal{S} := \{y \in \mathbb{R}^m : F(y) = 0\} \text{ and } \mathcal{S}_f := \{y \in \mathbb{R}^m : F_f(y) = 0\}, \tag{7.14}$$

where F is defined in (7.3) and F_f is defined in (7.10). In fact, when strict feasibility fails, $\mathcal{S}$ is unbounded given that $\mathcal{S} \neq \varnothing$ (see Theorem 7.4.4 (i) below). Moreover, $\mathcal{S}$ and $\mathcal{S}_f$ are convex and $\mathcal{S} \subseteq \mathcal{S}_f$.

Theorem 7.2.4. *The facially reduced problem (7.10) admits at least* $\operatorname{maxsd}(\mathcal{F})$ *number of affinely independent dual solutions y.*

Proof. Let $(\bar{X}, \bar{y}, \bar{Z})$ be a triple satisfying (7.5). Let $\lambda^1, \lambda^2, \ldots, \lambda^{\operatorname{maxsd}(\mathcal{F})}$ be vectors generated by **FR** iterations. By (2.10), each λ^i satisfies $\mathcal{A}^*(\lambda^i) \succeq 0$ and $\langle \bar{X}, \mathcal{A}^*(\lambda^i)\rangle = 0$. Since $\bar{y}$ satisfies $\bar{X} - W - \mathcal{A}^*\bar{y} - \bar{Z} = 0$, it follows that $\bar{X} - W - \mathcal{A}^*(\bar{y} - \lambda^i) - (\bar{Z} + \mathcal{A}^*(\lambda^i)) = 0$. Moreover, $\bar{Z} + \mathcal{A}^*(\lambda^i) \succeq 0$ and $\langle \bar{X}, \bar{Z} + \mathcal{A}^*(\lambda^i)\rangle = 0$. Thus the vectors in the following set

$$\mathcal{S}_\lambda := \bar{y} - \{\lambda^1, \ldots, \lambda^{\operatorname{maxsd}(\mathcal{F})}\} = \{\bar{y} - \lambda^1, \ldots, \bar{y} - \lambda^{\operatorname{maxsd}(\mathcal{F})}\}$$

are solutions to (7.10) as well. Consequently, the result follows from the linear independence in Proposition 2.4.3. □

We show in Example 7.2.5 that $\mathcal{S}$ and $\mathcal{S}_f$ can differ, i.e., the containment $\mathcal{S} \subseteq \mathcal{S}_f$ can be strict.

Example 7.2.5 ($\mathcal{S} \subsetneq \mathcal{S}_f$)**.** *Consider the instance $\mathcal{F}$ given by the data:*

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and } b = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

*The singularity degree of $\mathcal{F}$ is 2, $\mathrm{sd}(\mathcal{F}) = 2$. The first **FR** iteration yields a face that strictly contains the minimal face and corresponds to $\lambda^1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ with the facial range vector $V_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$; and the second **FR** iteration yields $\lambda^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ with the facial range vector $V_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Thus, the minimal facial range vector V for $\mathcal{F}$ is $V = V_1 V_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$. The facially reduced system is $\{R \in \mathbb{S}^1_+ : \begin{bmatrix} 1 \end{bmatrix} R = 1\}$. We note that $\mathcal{F}$ contains a unique point $e_1 e_1^T$.*

*We now consider the **BAP** (7.1) with $W = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & 0 \end{bmatrix}$. We consider the triple $(\bar{X}, \bar{Z}, \bar{y})$ where*

$$\bar{X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \bar{Z} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \text{ and } \bar{y} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}.$$

The triple $(\bar{X}, \bar{Z}, \bar{y})$ satisfies the first-order optimality conditions.

We note that $\bar{y} - \lambda^1$ and $\bar{y} - \lambda^2$ are solutions to (7.10). However, $\bar{y} - \lambda^2$ is not a solution to (7.3) since

$$W + \mathcal{A}^*(\bar{y} - \lambda^2) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & -2 & -1 \\ -1 & -1 & -2 \end{bmatrix},$$

and $\bar{X} - W - \mathcal{A}^*(\bar{y} - \lambda^2)$ has a negative eigenvalue.

It is of interest that the containment relation $\mathcal{S} \subsetneq \mathcal{S}_f$ in Example 7.2.5 stems from the solutions to (2.10). This can also be seen from the optimality characterization (7.5) with $Z \succeq 0$ and the regularized characterization (7.12) with $Z \succeq_{f^+} 0$, where the failure of strict feasibility implies $f \subsetneq \mathbb{S}^n_+, f^+ \supsetneq \mathbb{S}^n_+$.

7.3 A Basic Newton Method

We design a Newton-like method that solves for a root $\bar{y}$, $F(\bar{y}) = 0$, where

$$F(y) = \mathcal{A} P_{\mathbb{S}^n_+}(W + \mathcal{A}^* y) - b.$$

Rather than applying an optimization algorithm to solve the dual as in [122], we highlight that we solve a system of equations of the form $F(y) = 0$ to find points satisfying the first-order optimality conditions as is done in [29, 35, 130]. Given that $F(\bar{y}) = 0$, it follows that the optimum of the **BAP** (7.1) is $\bar{X} = P_{\mathbb{S}^n_+}(W + \mathcal{A}^*\bar{y})$. We note that $P_{\mathbb{S}^n_+}$ is found using the Eckart-Young Theorem [65], i.e., we use a spectral decomposition and set the negative eigenvalues to 0. Primal feasibility is immediate from the definitions and the projection. An application of the Moreau theorem yields the dual feasibility and complementarity.

We now present the pseudo-code of our Semi-Smooth Newton Method for the **BAP** (7.1) in Algorithm 7.1. The algorithm proceeds as follows: at each iteration, the search direction is computed by forming a (Clarke generalized) Jacobian J_k of F at the current iterate y_k, and the triple (X_k, Z_k, y_k) is then updated using this search direction. We resort to evaluations of unit vectors to construct J_k, the computational steps are discussed in Section 7.3.1

Algorithm 7.1 Semi-Smooth Newton Method for **BAP** for Spectrahedra

Inputs: $(W \in \mathbb{S}^n, \mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m, b \in \mathbb{R}^m)$, $(y_0 \in \mathbb{R}^m, \varepsilon > 0,$ maxiter $\in \mathbb{N})$

Initialization: $k \leftarrow 0$, $F_0 \leftarrow \mathcal{A}(X_0) - b$, stopcrit $\leftarrow \|F_0\|/(1 + \|b\|)$,
$X_0 \leftarrow P_{\mathbb{S}^n_+}(W + \mathcal{A}^*y_0)$, $Z_0 \leftarrow (X_0 - W - \mathcal{A}^*y_0)$
while (stopcrit $> \varepsilon$) & ($k \leqslant$ maxiter) **do**
  construct Jacobian J_k by evaluating each column of $J(y_k)$ using (7.21)
  choose a regularization parameter $\sigma \geqslant 0$ for positive definite $\bar{J} = (J_k + \sigma I_m)$
  solve system $\bar{J}d = -F_k$ ▷ (Obtain Newton direction d)
  update:
    $y_{k+1} \leftarrow y_k + d$
    $X_{k+1} \leftarrow P_{\mathbb{S}^n_+}(W + \mathcal{A}^*y_{k+1})$
    $Z_{k+1} \leftarrow X_{k+1} - (W + \mathcal{A}^*y_{k+1})$
    $F_{k+1} \leftarrow \mathcal{A}(X_{k+1}) - b$
    stopcrit $\leftarrow \|F_{k+1}\|/(1 + \|b\|)$
    $k \leftarrow k + 1$
end while
Output: Primal-dual near optimum: $X_k, (y_k, Z_k)$

Remark 7.3.1 (On the convergence of Algorithm 7.1)**.** *Convergence guarantees for Algorithm 7.1 are derived from the so-called* inexact Semismooth Newton's Algorithm *in Facchinei and Pang's book; see [66, Algorithm 7.5.4]. Specifically, Facchinei and Pang consider an inexact Newton's method for obtaining a zero of a semismooth operator $F : \mathbb{R}^m \to \mathbb{R}^m$, with search direction d^k defined by*

$$F(x^k) + J_k d^k = r^k, \quad \forall k,$$

where J_k is an element of the Clarke generalized Jacobian of F at x^k, and the error r^k satisfies

$$\|r^k\| \leqslant \eta_k \|F(x^k)\|$$

for some nonnegative sequence $\{\eta_k\}$. According to [66, Theorem 7.5.5], if the generalized Jacobian is nonsingular at a point x^* satisfying $F(x^*) = 0$, and if there exists a constant $\bar{\eta} > 0$ such that $\eta_k \leqslant \bar{\eta}$ for all k, then the method converges locally to x^*.

Our algorithm can be regarded as a realization of this inexact Semismooth Algorithm where

$r^k = -\sigma d^k$, *with* $\sigma > 0$, *and therefore* $d^k = -(J_k + \sigma I_m)^{-1} F(x^k)$. *Hence, we may choose* η_k *as*

$$\eta_k = \frac{\|r^k\|}{\|F(x^k)\|} = \frac{\|(J_k + \sigma I_m)^{-1} F(x^k)\|}{\|F(x^k)\|}\sigma \leqslant \frac{\sigma}{\lambda_{\min}(J_k) + \sigma} \leqslant 1, \quad \forall k,$$

where $\lambda_{min}(J_k)$ *stands for the smallest eigenvalue of* J_k. *Therefore, [66, Theorem 7.5.5] provides local convergence guarantees for Algorithm 7.1.*

7.3.1 Alternative Directional Derivative Formulation

We now outline the computation of the Jacobian J_k at line 4 in Algorithm 7.1. In principle, obtaining a Clarke generalized Jacobian of F in our semismooth Newton method would require computing an element in the Clarke generalized Jacobian of $P_{\mathbb{S}^n_+}$. Every element in the generalized Jacobian is a 4-*tensor* on $\mathbb{R}^n$, whose complete formulation can be found in [123]. In matrix form this would be expressed as a square matrix of order n^4. The memory requirements for storing a matrix of such dimension can be too demanding even for moderate values of n. In particular, MATLAB software would have problems with size $n \geqslant 150$.

In order to overcome the memory deficiency, we make use of an elegant formula for evaluating Clarke generalized Jacobians of $P_{\mathbb{S}^n_+}$ derived by Sun in [167], which builds upon earlier developments in [139, 168]. In what follows we show this formula can be leveraged to efficiently compute a Clarke generalized Jacobian of F through evaluations involving only the unit vectors $e_i \in \mathbb{R}^m$. We begin by introducing the following notation.

Let $S = U\Lambda U^T \in \mathbb{S}^n$ be the spectral decomposition with vector of eigenvalues λ given as above. We partition the index sets of the eigenvalues based on their signs:

$$\alpha = \{i : \lambda_i > 0\},\ \beta = \{i : \lambda_i = 0\},\ \gamma = \{i : \lambda_i < 0\}.$$

Accordingly, the eigenvalue and eigenvector matrices can be block-partitioned as

$$\Lambda = \text{blkdiag}(\Lambda_\alpha, 0, \Lambda_\gamma), \quad \text{and} \quad U = [U_\alpha\, U_\beta\, U_\gamma].$$

Next, we define $\Omega \in \mathbb{S}^n$ entry-wise by

$$\Omega_{ij} = \frac{\max(\lambda_i, 0) + \max(\lambda_j, 0)}{|\lambda_i| + |\lambda_j|},\ \forall i, j, \tag{7.15}$$

where the convention $\frac{0}{0} := 1$ is adopted. By [167, Proposition 2.2], for any matrix $S = U\Lambda U^T$, a linear map G_1 belongs to the Clarke generalized Jacobian $\partial_C P_{\mathbb{S}^n_+}(S)$ if and only if there exists $G_2 \in \partial_C P_{\mathbb{S}^{|\beta|}_+}(0)$ such that for all $H \in \mathbb{S}^n$,

$$G_1(H) = U \begin{bmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & \Omega_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}^T_{\alpha\beta} & G_2(\tilde{H}_{\beta\beta}) & 0 \\ \tilde{H}^T_{\alpha\gamma} \circ \Omega^T_{\alpha\gamma} & 0 & 0 \end{bmatrix} U^T, \tag{7.16}$$

where $\tilde{H} = U^T H U$. Equation (7.16) indicates that any element of the Clarke generalized Jacobian $\partial_C P_{\mathbb{S}^n_+}(S)$ is determined by a Clarke generalized Jacobian in $G_2 \in \partial_C P_{\mathbb{S}^{|\beta|}_+}(0)$. Consequently, the problem reduces to finding an element in $\partial_C P_{\mathbb{S}^{|\beta|}_+}(0)$.

In particular, by [123, Example 3.8], the Clarke subdifferential of $P_{\mathbb{S}_+^{|\beta|}}$ at 0 is characterized by the set of 4-tensors on $\mathbb{R}^{|\beta|}$ given by

$$\partial_C P_{\mathbb{S}_+^{|\beta|}}(0) = \operatorname{conv}\left(\mathcal{O}^{|\beta|} \cdot (\operatorname{Diag}^{(12)} \mathcal{D}_{\{01\}}(|\beta|))\right),$$

where $\operatorname{conv}(S)$ stands for the convex hull of a set S and

(i) $\mathcal{D}_{\{01\}}(|\beta|)$ is the set of $|\beta| \times |\beta|$ symmetric matrices with entries in $\{0, 1\}$ such that the entries in each row (from left to right) or column (from top to bottom) form a noincreasing sequence;

(ii) for any matrix M in $\mathbb{R}^{|\beta|\times|\beta|}$, $\operatorname{Diag}^{(12)} M$ is defined as the 4-tensor on $\mathbb{R}^{|\beta|}$ whose components are given by

$$(\operatorname{Diag}^{(12)} M)_{\substack{i_1 i_2 \\ j_1 j_2}} = \begin{cases} M_{i_1 i_2}, & \text{if } i_1 = j_2 \text{ and } i_2 = j_1, \\ 0, & \text{otherwise.} \end{cases}$$

We note that, since the zero matrix of size $|\beta| \times |\beta|$ belongs to $D_{\{01\}}(|\beta|)$, the zero tensor from $\mathbb{R}^{|\beta|\times|\beta|}$ to $\mathbb{R}^{|\beta|\times|\beta|}$ belongs to $\partial_C P_{\mathbb{S}_+^{|\beta|}}(0)$.

Hence, in Algorithm 7.1, we can always choose the generalized Jacobian $G_1 \in \partial_C P_{\mathbb{S}_+^n}(S)$ associated with $G_2 = 0$. Under this choice, the evaluation of G_1 at any $H \in \mathbb{S}^n$ simplifies to

$$G_1(H) = U \begin{bmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & \Omega_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & 0 & 0 \\ \tilde{H}_{\alpha\gamma}^T \circ \Omega_{\alpha\gamma}^T & 0 & 0 \end{bmatrix} U^T, \quad \forall H \in \mathbb{S}^n, \tag{7.17}$$

where $\tilde{H} = U^T H U$. Finally, observe that a Clarke Jacobian for our function F is a matrix that can be explicitly constructed by evaluating this operator along the standard basis vectors.

The following lemma provides an explicit expression for evaluating this particular Jacobian $J \in \partial_C F(y)$ for any $y \in \mathbb{R}^n$ along any direction.

Lemma 7.3.2. *Let $y \in \mathbb{R}^m$ be such that $Y := W + \mathcal{A}^* y \in \mathbb{S}^n$. Let $Y := U \operatorname{Diag}(\lambda(Y)) U^T$ be a spectral decomposition of Y such that the eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n$ are sorted in nonincreasing order, and denote with α, β and γ the sets of indices associated with positive, zero and negative eigenvalues, respectively, i.e., $\alpha := \{i : \lambda_i > 0\}$, $\beta := \{i : \lambda_i = 0\}$ and $\gamma = \{i : \lambda_i < 0\}$. Then there exists a Clarke generalized Jacobian J in $\partial_C F$ at y such that its evaluation at any direction $\Delta y \in \mathbb{R}^m$ is given by*

$$J(\Delta y) = \mathcal{A}\left(U \begin{bmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & \Omega_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & 0 & 0 \\ \tilde{H}_{\alpha\gamma}^T \circ \Omega_{\alpha\gamma}^T & 0 & 0 \end{bmatrix} U^T\right), \tag{7.18}$$

where $\tilde{H} := U^T (\mathcal{A}^* \Delta y) U$.

Proof. It suffices to recall that the chain rule for Clarke subdifferential applied to F yields

$$\partial_C F(y) = \mathcal{A} \partial_C P_{\mathbb{S}_+^n}(W + \mathcal{A}^* y) \mathcal{A}^*,$$

where the equality above is by [43, Theorem 2.3.10], since semismooth implies Clarke regular. Hence, by taking $G_1 \in \partial_C P_{\mathbb{S}^n_+}(S)$ given above for $S = W + \mathcal{A}^* y$, the equation in (7.17) provides (7.18). □

We now outline the steps for computing the Clarke generalized Jacobian J_k at line 4 in Algorithm 7.1. Specifically, by Lemma 7.3.2, the Jacobian of F evaluated at $y \in \mathbb{R}^m$, $J(y)$, is computed following the steps below.

(i) Let $Y = W + \mathcal{A}^* y \in \mathbb{S}^n$. Consider the spectral decomposition of Y given by

$$Y = \begin{bmatrix} V_\alpha & V_\beta & V_\gamma \end{bmatrix} \operatorname{Diag}(\lambda(Y)) \begin{bmatrix} V_\alpha & V_\beta & V_\gamma \end{bmatrix}^T,$$

where V_α, V_β and V_γ are the matrices of eigenvectors associated with the positive, zero and negative eigenvalues of Y, respectively.

(ii) Define the rotation $\mathcal{R}_Y : \mathbb{S}^n \to \mathbb{S}^n$, by

$$\mathcal{R}_Y(\rho) := \begin{bmatrix} V_\alpha & V_\beta & V_\gamma \end{bmatrix} \rho \begin{bmatrix} V_\alpha & V_\beta & V_\gamma \end{bmatrix}^T; \tag{7.19}$$

(iii) For each $j = 1, \ldots, m$, compute

$$T_j := \begin{bmatrix} V_\alpha^T A_j V_\alpha & V_\alpha^T A_j V_\beta & \Omega_{\alpha\gamma} \circ V_\alpha^T A_j V_\gamma \\ (V_\alpha^T A_j V_\beta)^T & 0 & 0 \\ \left(\Omega_{\alpha\gamma} \circ V_\alpha^T A_j V_\gamma^T\right)^T & 0 & 0 \end{bmatrix} \in \mathbb{S}^n; \tag{7.20}$$

(iv) The j-th column of the Jacobian at y, $J(y)$, is

$$\mathcal{A}(\mathcal{R}_Y(T_j)) =: A \operatorname{svec}\left(\mathcal{R}_Y(T_j)\right). \tag{7.21}$$

7.4 Failure of Regularity and Degeneracy

This section examines various aspects of Algorithm 7.1 that result from the failure of strict feasibility. The failure of regularity is known to result in pathologies in conic programs, both on theoretical and practical aspects. We show that Algorithm 7.1 is not an exception to this phenomenon.

This section is organized in three parts. In Section 7.4.1 we discuss two types of pathologies. One well-known pathology is the possibility of failure of strong duality. Since the primal and dual optimal values agree (Theorem 7.2.1 (ii)), the only difficulty left is that the dual optimal value may not be attained by any dual feasible point. We identify a condition where this occurs (see Lemma 7.4.2) and show how to construct instances where strong duality fails. Another well-known consequence of the absence of strict feasibility is that the dual optimal set is unbounded [72]. We explain why Algorithm 7.1 experiences difficulties in this case in Section 7.4.1. The second part in Section 7.4.2 is devoted to understanding the properties of the Jacobian of F computed near the optimal point as seen through the lens of degeneracy. We recall the discussions from Section 2.4.2 to help explain the behaviour of Algorithm 7.1. In particular, we rely on the fact that *every point in $\mathcal{F}$ is degenerate in the absence of strict feasibility*. We conclude in Section 7.4.3 with the application of degeneracy identification to two real-world examples: the elliptope and the vontope.

7.4.1 Pathologies Arise in the Absence of Strict Feasibility

In this section we discuss pathologies that arise as a result of the absence of strict feasibility. We provide a method of constructing instances where the dual optimal value is not attained. In addition, assuming that the dual optimal value is attained, we provide members that certify the unbounded dual optimal set; and we examine the behaviour of Algorithm 7.1.

Unattained Dual Optimal Value

Theorem 7.2.1 states that there is always a *zero duality gap*, $p^* = d^*$ and the solution value of the primal problem, p^*, is attained. However, in the absence of strict feasibility, the dual attainment does not necessarily hold. Example 7.4.1 below illustrates that strong duality can fail for (7.1) when strict feasibility fails.

Example 7.4.1 (Failure of strong duality)**.** *Consider the following instance of* ***BAP*** (7.1) *given by*

$$\min_X \left\{ \frac{1}{2} \left\| X - \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \right\|^2 \ : \ X_{11} = 0, \ X \succeq_{\mathbb{S}^2_+} 0 \right\}. \tag{7.22}$$

The set of feasible solutions of (7.22) *is* $\{X \in \mathbb{S}^2 : X_{11} = X_{12} = X_{21} = 0, X_{22} \geqslant 0\}$*. Therefore, the optimal value of the problem is*

$$1 = \min_{X_{22} \geqslant 0} \frac{1}{2} \left\| \begin{bmatrix} 0 & 1 \\ 1 & X_{22} \end{bmatrix} \right\|^2 = \frac{1}{2} \left(2 + X_{22}^2\right),$$

which is attained when $X_{22} = 0$*. In other words, the optimal solution of the best approximation problem is attained at* $\bar{X} = 0$*.*

Now, note that the primal constraint in (7.22) *is given by* $\operatorname{trace}(E_{11}X) = \mathcal{A}X = 0$*, and therefore* $\mathcal{A}^* y = yE_{11}$ *for all* $y \in \mathbb{R}$*. Thus, dual feasibility of the optimality conditions (see* (7.5)*) implies*

$$-\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} - \bar{y}E_{11} = \begin{bmatrix} -\bar{y} & 1 \\ 1 & 0 \end{bmatrix} \in \mathbb{S}^2_+, \ \text{for some } \bar{y} \in \mathbb{R}.$$

However this does not hold for any $\bar{y} \in \mathbb{R}$*. Thus attainment fails for the dual.*

Example 7.4.1 above illustrates that strong duality may fail in the absence of strict feasibility; the linear manifold defined by $X_{11} = 0$ entirely consists of singular matrices. We note that strong duality can hold even in the absence of strict feasibility. Remark 7.4.3 presents a constructive approach for generating instances that fail strong duality. We first recall the following.

Lemma 7.4.2 ([150, Lemma 2.2])**.** *Suppose that* $0 \neq K \trianglelefteq \mathbb{S}^n_+$*, is a proper face of* $\mathbb{S}^n_+$*. Then*

$$\mathbb{S}^n_+ + K^\perp = \overline{\mathbb{S}^n_+ + \operatorname{span} K^\Delta}.$$

Furthermore,

$$\mathbb{S}^n_+ + \operatorname{span} K \ \textit{is not closed.} \tag{7.23}$$

Remark 7.4.3 (Constructing instances that fail strong duality)**.** *The dual feasibility of the first-order optimality conditions* (7.5) *states:*

$$\bar{X} - W \in \operatorname{range}(\mathcal{A}^*) + \mathbb{S}^n_+.$$

From (7.23)*, we can choose any proper face* $K \trianglelefteq \mathbb{S}^n_+$ *and construct a linear map* $\mathcal{A}$ *to satisfy* $\operatorname{range}(\mathcal{A}^*) = \operatorname{span} K$*. Therefore,*

$$\bar{X} - W \in \overline{\operatorname{range}(\mathcal{A}^*) + \mathbb{S}^n_+} \setminus \left(\operatorname{range}(\mathcal{A}^*) + \mathbb{S}^n_+\right), \ \bar{X} \succeq 0,$$

results in the failure of (7.5)*. Example 7.4.1 indeed falls into this category. Note that we can always choose* $b = \mathcal{A}\bar{X}$ *so that we still have a zero duality gap.*

Unbounded Dual Optimal Set and Singular Jacobian

We now discuss a property of the dual optimal set that, if it exists, results in a poor behaviour of Algorithm 7.1. Recall that the absence of strict feasibility of $\mathcal{F}$ implies the existence of a solution λ of the auxiliary system (2.10). We use the solution λ of (2.10) to derive two properties of the dual solution set $\mathcal{S} = \{y \in \mathbb{R}^m : F(y) = 0\}$ defined in (7.14):

$$\text{Slater fails, } \mathcal{S} \neq \varnothing \implies \begin{cases} \text{(i): solution set } \mathcal{S} \text{ is unbounded;} \\ \text{(ii) Jacobian at any solution } \bar{y} \in \mathcal{S} \text{ is singular.} \end{cases} \tag{7.24}$$

Theorem 7.4.4 below clarifies the conditions that result in the unbounded dual solution set in (7.24)(i); it then explains why we get an ill-conditioned Jacobian and thus provides a rationale for regularization of the search direction at line 4 in Algorithm 7.1.

Theorem 7.4.4. *Suppose that strict feasibility fails for the (primal) spectrahedron* (7.1) but *strong duality holds. Let* $\bar{y} \in \mathcal{S}$ *and let* λ *be any solution to* (2.10)*. Then the following hold:*

(i) The solution set $\mathcal{S}$ *is unbounded. Moreover,* λ *provides a recession direction,* $F(\bar{y} - t\lambda) = 0, \forall t \in \mathbb{R}^p$*.*

(ii) The directional derivative of F *at* $\bar{y}$ *along* λ *exists and is equal to zero.*

(iii) In addition suppose that F *is differentiable at* $\bar{y} \in \mathcal{S}$*. Then the Jacobian* $F'(\bar{y})$ *is singular. Moreover,* $\lambda \in \operatorname{null} F'(\bar{y})$*.*

Proof. Item (i): Let $(\bar{X}, \bar{y}, \bar{Z})$ be a triple that satisfies the optimality conditions in (7.5). We now let λ be a solution to the auxiliary system (2.10) and $Z := \mathcal{A}^*\lambda \succeq 0$. We aim to show that, for any $t > 0$, the triple $(\bar{X}, \bar{y} - t\lambda, \bar{Z} + tZ)$ also satisfies the optimality conditions. Indeed, for all $t > 0$, we have $\bar{Z} + tZ \succeq 0$ and

$$\begin{aligned} 0 &= \bar{X} - W - \mathcal{A}^*\bar{y} - \bar{Z} \\ &= \bar{X} - W - \mathcal{A}^*(\bar{y} - t\lambda) - (\bar{Z} + tZ). \end{aligned}$$

The verification of primal feasibility is trivial. Finally complementarity follows:

$$\langle \bar{Z} + tZ, \bar{X} \rangle = t\langle Z, \bar{X} \rangle = t\langle \lambda, \mathcal{A}\bar{X} \rangle = t\langle \lambda, b \rangle = 0, \quad \forall t > 0,$$

where the last equality follows from (2.10). Finally, by the proof of Theorem 7.2.1 (iii) we conclude that $\bar{y} - t\lambda$ is a root of F for all $t > 0$, or equivalently,

$$\{\bar{y} - t\lambda : t \in \mathbb{R}_+\} \subseteq \mathcal{S}.$$

Item (ii): This directly follows from the fact that $F(\bar{y}) = F(\bar{y} - t\lambda)$ for all $y \in \mathcal{S}$ and $t \in \mathbb{R}_+$.

Item (iii): Suppose F is differentiable at a point $\bar{y} \in \mathcal{S}$. Then the partial derivative of F at $\bar{y}$ in the direction of λ is given by

$$F'(\bar{y})\lambda = 0,$$

where $F'(\bar{y})$ denotes the Jacobian of F at $\bar{y}$. □

We note that the system (2.10) may contain multiple linearly independent solutions. Let $\{\lambda^1, \ldots, \lambda^k\}$ be a set of linearly independent solutions to (2.10). Hence by Theorem 7.4.4 we deduce that the solution set $\mathcal{S}$ contains a k-dimensional recession cone. Moreover, if the differentiability of F at $\bar{y}$ is further assumed, null $F'(\bar{y})$ contains at least k zero singular values. Another interesting consequence of Theorem 7.4.4 is that if $F'(\bar{y})$ is nonsingular, then strict feasibility holds for $\mathcal{F}$.

The unboundedness of the set $\mathcal{S}$ immediately translates into the unboundedness of the set of optimal solutions of the dual problem (7.2). The proof of Theorem 7.4.4 Item (i) shows that the triple $(\bar{X}, \bar{y} - t\lambda, \bar{Z} + t\mathcal{A}^*\lambda)$ satisfies the optimality conditions (7.5) for all $t \in \mathbb{R}_+$. Therefore, the unbounded set

$$\{(\bar{y}, \bar{Z}) + t(-\lambda, \mathcal{A}^*\lambda) \ : \ t \in \mathbb{R}_+\}$$

constitutes recession directions of the set of dual solutions.

Furthermore, as stated in [72] (and related papers), the dual optimal set is unbounded when Mangasarian-Fromovitz type constraint qualifications fail. Here this means that the dual optimal set is unbounded when strict feasibility fails for the primal (7.1), since we assume that $\mathcal{A}$ is surjective. Our dual Algorithm 7.1 can encounter numerical difficulties in this setting. We now provide the details why the norms of the dual variables typically diverge.

Let $\bar{y} \in \mathcal{S}$. Suppose that we are at a point $\hat{y}$ such that $0 \neq \epsilon = F(\hat{y})$, $\hat{y} = \bar{y} + \Delta y$. We note that

$$\epsilon = F(\bar{y} + \Delta y) - F(\bar{y}) \approx F'(\bar{y})\Delta y.$$

When $\|\epsilon\|$ is small, Δy is close to being in null$(F'(\bar{y}))$. We have shown in Theorem 7.4.4 that a solution λ to (2.10) always satisfies

$$F(\bar{y} + \lambda) = 0 \text{ and } F'(\bar{y})\lambda = 0.$$

Therefore, *large* choices for Δy that include a large component from the nullspace is possible. This is in particular true when the Jacobian is singular at the optimum and the regularization parameter is converging to zero.

A typical behaviour of Algorithm 7.1 in the absence of strict feasibility is illustrated in Figure 7.1, i.e., we see the growth of the norms of the dual variables.

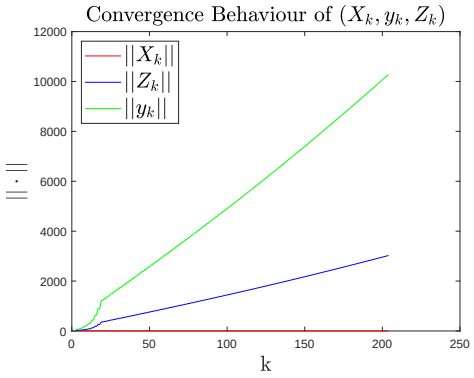

Figure 7.1: Typical behaviour of $\{(X_k, y_k, Z_k)\}$ from Algorithm 7.1 in the absence of strict feasibility.

7.4.2 Jacobian Behaviour Near-Optimum and Degeneracy

In this section we study properties of the Jacobian of F computed near the optimal point and relate its behaviour to the degeneracy status of the optimal point. We show that the degeneracy status of the optimal point *characterizes* the singularity of the Jacobian matrix.

We extend the discussion of computing the Jacobian presented in Lemma 7.3.2 and elaborate the computational steps. Let $(\bar{X}, \bar{y}, \bar{Z})$ be an optimal triple that solves (7.5). We further assume that $\bar{X}$ and $\bar{Z}$ satisfy *strict complementarity*. Since $\bar{X}$ and $\bar{Z}$ are mutually orthogonally diagonalizable, we obtain

$$\bar{X} - \bar{Z} = W + \mathcal{A}^*(\bar{y}) = \begin{bmatrix} V & \bar{V} \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & -S \end{bmatrix} \begin{bmatrix} V & \bar{V} \end{bmatrix}^T, \quad R \succ 0,\, S \succ 0,$$

where $\bar{X} = VRV^T$ and $\bar{Z} = \bar{V}S\bar{V}^T$.

Recall the steps for computing the Jacobian in Section 7.3.1 and the rotation operator in (7.19). Since we are now assuming that the matrix $\bar{X} - \bar{Z} = W + \mathcal{A}^*(\bar{y})$ is nonsingular, then the matrices of eigenvectors in (7.19) are given by $V_\alpha = V$, $V_\beta = 0$ and $V_\gamma = \bar{V}$. Hence, the rotation operator we are interested in now is $\mathcal{R}_{\bar{X}-\bar{Z}}(\cdot) = \begin{bmatrix} V & \bar{V} \end{bmatrix}(\cdot)\begin{bmatrix} V & \bar{V} \end{bmatrix}^T$. We now closely observe how the (i,j)-th element of the Jacobian in (7.21) is evaluated. Let T_j be the matrix defined in (7.20). Then

$$\begin{aligned} & \operatorname{trace}(A_i \mathcal{R}_{\bar{X}-\bar{Z}}(T_j)) \\ = & \left\langle A_i, \begin{bmatrix} V & \bar{V} \end{bmatrix} T_j \begin{bmatrix} V & \bar{V} \end{bmatrix}^T \right\rangle \\ = & \left\langle \begin{bmatrix} V & \bar{V} \end{bmatrix}^T A_i \begin{bmatrix} V & \bar{V} \end{bmatrix}, T_j \right\rangle \\ = & \left\langle \begin{bmatrix} V^T A_i V & V^T A_i \bar{V} \\ \bar{V}^T A_i V & \bar{V}^T A_i \bar{V} \end{bmatrix}, \begin{bmatrix} V^T A_j V & \Omega_{\alpha\gamma} \circ V^T A_j \bar{V} \\ \left(\Omega_{\alpha\gamma} \circ V^T A_j \bar{V}\right)^T & 0 \end{bmatrix} \right\rangle \\ = & \left\langle \begin{bmatrix} V^T A_i V & V^T A_i \bar{V} \\ \bar{V}^T A_i V & 0 \end{bmatrix}, \begin{bmatrix} V^T A_j V & \Omega_{\alpha\gamma} \circ V^T A_j \bar{V} \\ \left(\Omega_{\alpha\gamma} \circ V^T A_j \bar{V}\right)^T & 0 \end{bmatrix} \right\rangle. \end{aligned} \tag{7.25}$$

Note that the two arguments in the last trace inner product in (7.25) share the analogous block structure, with the difference arising from both the use of A_i versus A_j and an element-wise (Hadamard) scaling applied to the off-diagonal block. Lemma 7.4.5 below links the degeneracy of the optimal point $\bar{X}$ to the invertibility of the Jacobian of F at $\bar{y}$.

Lemma 7.4.5. *Let $D \in \mathbb{S}^n_{++}$ be a diagonal matrix, and let $\{x_1, \ldots, x_m\} \subseteq \mathbb{R}^n$ be given. Let $U = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}$. Then* $\operatorname{rank}(U) = \operatorname{rank}(U^T U) = \operatorname{rank}(U^T D U)$.

Now we use (7.25), Lemma 7.4.5, and Lemma 2.4.5 to characterize the singularity of the Jacobian of F evaluated at an optimal solution. We note that [10, Theorem 3.1] addresses nonsingularity of Jacobians of linear **SDP**s under nondegeneracy assumptions.

Theorem 7.4.6. *Let $(\bar{X}, \bar{y}, \bar{Z})$ satisfy the optimality condition (7.5) and the strict complementarity condition. Then $\bar{X}$ is degenerate if, and only if, the Jacobian of F at $\bar{y}$ is singular.*

Proof. Let $\bar{X}$ be the optimal point of (7.1). Let

$$D = \operatorname{Diag}\left(\operatorname{svec}\left(\begin{bmatrix} M & \frac{1}{\sqrt{2}}\Omega_{\alpha\gamma}(\bar{X}) \\ \frac{1}{\sqrt{2}}\Omega_{\alpha\gamma}(\bar{X})^T & M \end{bmatrix}\right)\right) \in \mathbb{S}^{t(n)}_{++},$$

where $M = \frac{1}{\sqrt{2}}E + \left(1 - \frac{1}{\sqrt{2}}\right) I$ and where we include the dependence on $\bar{X}$ of Ω given as in (7.15). Let $X_i := \begin{bmatrix} V^T A_i V & V^T A_i \bar{V} \\ \bar{V}^T A_i V & 0 \end{bmatrix}$, and let $x_i := \operatorname{svec}(X_i)$. We recall the definition of T_j in (7.20) and note that

$$\operatorname{svec}(T_j) = D x_j.$$

We then observe the last inner product in (7.25):

$$\langle X_i, T_j \rangle = \langle \operatorname{svec}(X_i), \operatorname{svec}(T_j) \rangle = \langle x_i, D x_j \rangle.$$

Now we form $U := \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \in \mathbb{R}^{t(n) \times m}$. Then, $\forall i, j$, we have

$$(U^T D U)_{i,j} = \left(\begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix} \begin{bmatrix} D x_1 \cdots D x_m \end{bmatrix} \right)_{i,j} = x_i^T D x_j = \operatorname{trace}(A_i \mathcal{R}_{\bar{X}}(T_j)).$$

Therefore we conclude

$$\begin{array}{lll} \bar{X} \text{ is degenerate} & \iff & \operatorname{rank}(U) < m, \qquad \text{by (2.12)}, \\ & \iff & U^T D U \text{ is singular}, \qquad \text{by Lemma 7.4.5}, \\ & \iff & \text{Jacobian of } F \text{ at } \bar{X} \text{ is singular}. \end{array}$$

□

Recall the sufficient conditions for producing a nondegenerate solution given in Propositions 2.4.9 and 2.4.10. Therefore, any projection point $\bar{X}$ that satisfies the conditions in Propositions 2.4.9 and 2.4.10 yields a nonsingular Jacobian.

7.4.3 Nondegeneracy of the Elliptope and Degeneracy of the Vontope

We study degeneracies of two classes of spectrahedra; the elliptope (the set of correlation matrices), and the vontope (feasible region of the **SDP** relaxation of the quadratic assignment problem, **QAP**). We introduce these sets to illustrate how degeneracy interacts with the performance of

Algorithm 7.1 in Section 7.5.2. We exhibit the result from [144] that the elliptope has only nondegenerate points; however all vertices of the vontope are degenerate before **FR**, and some vertices of the vontope are degenerate even after **FR**.

Example 7.4.7 (Elliptope, [144, Thm 3.4.2]). *We consider the problem of finding the nearest correlation matrix:*

$$\min \left\{ \frac{1}{2}\|X - W\|_F^2 \,:\, \operatorname{diag}(X) = e,\, X \succeq 0 \right\},$$

*where e is the vector of all ones. The feasible region of the above problem is called the elliptope. Note that the elliptope is the feasible region of the **SDP** relaxation of the max-cut problem. Every point in the elliptope is nondegenerate. A semismooth Newton method for this problem is developed in [148].*

Example 7.4.8 (Vontope, [182]). *Let Π_n be the set of n-by-n permutation matrices. For $X \in \Pi_n$, let*

$$Y_X = y_X y_X^T, \text{ where } y_X = \begin{pmatrix} 1 \\ \operatorname{vec} X \end{pmatrix} \in \mathbb{R}^{n^2+1},$$

*be the lifted matrix. Here we index the rows and columns of a matrix starting from 0. The lifting process gives rise to the following feasible region for the **SDP** relaxation:*

$$\mathcal{F}_{\boldsymbol{QAP}} := \left\{ Y \in \mathbb{S}^{n^2+1}_+ : \begin{array}{l} G_J(Y) = e_0, \mathrm{b}^0 \operatorname{diag}(Y) = I_n, \mathrm{o}^0 \operatorname{diag}(Y) = I_n, \\ Y_{0,j} = Y_{j,j}, \forall j = 1, \dots, n^2+1 \end{array} \right\}. \tag{7.26}$$

Here, $G_J : \mathbb{S}^{n^2+1} \to \mathbb{R}^{|J|}$ is a linear map that chooses the elements in the index set J that correspond to the $(0,0)$-element of Y, the off-diagonal elements of the n-by-n diagonal blocks, and the diagonal elements of the n-by-n off-diagonal blocks; $\mathrm{b}^0 \operatorname{diag} : \mathbb{S}^{n^2+1} \to \mathbb{S}^n$ is the linear map that sums the n-by-n diagonal blocks; and $\mathrm{o}^0 \operatorname{diag} : \mathbb{S}^{n^2+1} \to \mathbb{S}^n$ is the linear map defined by $\mathrm{o}^0 \operatorname{diag}(Y) = \left(\operatorname{trace}(\hat{Y}^{i,j})\right)_{i,j}$, where $\hat{Y}^{i,j}$ is the (i,j)-th n-by-n block submatrix in Y; see [182] for details on the construction of $G_J, \mathrm{b}^0 \operatorname{diag}$ and $\mathrm{o}^0 \operatorname{diag}$.

*We remark that the expression in (7.26) contains redundant linear constraints. It is well-known that the **SDP** relaxation of the **QAP** fails strict feasibility [182] and so we employ **FR** and work in a smaller space. Let*

$$H = \begin{bmatrix} e^T \otimes I_n \\ I_n \otimes e^T \end{bmatrix} \in \mathbb{R}^{2n \times n^2}, \; K = \begin{bmatrix} -e & H \end{bmatrix} \in \mathbb{R}^{2n \times (n^2+1)},$$

*and let $\hat{V} \in \mathbb{R}^{(n^2+1) \times ((n-1)^2+1)}$ be the matrix with orthonormal columns that spans $\operatorname{null}(K)$.[2] **FR** leads to the following constraints:*

$$\mathcal{F}^{\boldsymbol{FR}}_{\boldsymbol{QAP}} := \{R \in \mathbb{S}^{(n-1)^2+1} : G_{\hat{J}}(\hat{V} R \hat{V}^T) = e_0,\, R \succeq 0\}, \tag{7.27}$$

*where $G_{\hat{J}} : \mathbb{S}^{n^2+1} \to \mathbb{R}^{|\hat{J}|}$ a newly defined surjective linear map that chooses indices in $\hat{J}$ such that $\hat{J} \subsetneq J$. This aligns with the fact that **FR** reveals implicit redundant constraints. It is known that the number of equality constraints reduces to $n^3 - 2n^2 + 1$ after **FR**; see [182].*[3]

[2] Note that the last row of K is linearly dependent and, for efficiency and accuracy, is best ignored when finding the nullspace.

[3] The last column of off-diagonal blocks and the $(n-2, n-1)$ off-diagonal block are linearly dependent and are ignored, see [80, 182].

We now discuss the degeneracy of each lifted matrix $Y_X = y_X y_X^T = \hat{V} R_X \hat{V}^T$, *where* $X \in \Pi_n$. *Due to the orthonormality of* $\hat{V}$, *we get*

$$R_X = \hat{V}^T Y_X \hat{V} \in \mathbb{S}^{(n-1)^2+1}.$$

We note that $\operatorname{rank}(R_X) = 1$. *We let* $\{A_i\}_{i=1}^{n^3-2n^2+1} \subseteq \mathbb{S}^{(n-1)^2+1}$ *be the set of matrices defining data matrices for the affine constraints. Hence the linear dependence of the matrices of the set* (2.12) *can be analyzed by considering their first columns. For* $n \geqslant 3$, *we observe that the vectors*

$$\left\{ \begin{pmatrix} V_X^T A_i V_X \\ \bar{V}_X^T A_i V_X \end{pmatrix} \right\}_{i=1}^{n^3-2n^2+1} \subseteq \mathbb{R}^{(n-1)^2+1}, \quad n^3 - 2n^2 + 1 > (n-1)^2 + 1, \ n \geqslant 3,$$

are linearly dependent. This proves that the rank-one vertices that arise from Π_n *remains degenerate after* ***FR****.*

Remark 7.4.9. *If we replace* $\mathbb{S}^n_+$ *with* $\mathbb{R}^n_+$, *the set* $\mathcal{F}$ *reduces to a polyhedron and the discussion on the degeneracy simplifies. The degeneracy status of a point* x *in a polyhedron can be confirmed by evaluating the rank of* $A(:, \operatorname{supp}(x))$, *where* $\operatorname{supp}(x) = \{i : x_i \neq 0\}$ *denotes the support of* x; *see [144]. The performance of the proposed algorithm in [35] is also affected by the degeneracy of the optimal point. Moreover every point of* $\mathcal{F}$ *as a polyhedron is degenerate in the absence of strict feasibility.*

7.5 Numerical Experiments

To illustrate the effects on convergence and degeneracy, we now present multiple experiments using diverse spectrahedra $\mathcal{F}$ with various ranges of values for the *singularity degree,* $\operatorname{sd}(\mathcal{F})$, and for the *implicit problem singularity,* $\operatorname{ips}(\mathcal{F})$. In our algorithm, dual feasibility and complementary slackness are satisfied exactly. Therefore, we use the following $\boldsymbol{\varepsilon}^k \in \mathbb{R}_+$ to denote the relative residual of the optimality conditions at iteration k:

$$\boldsymbol{\varepsilon}^k := \min\left\{1, \frac{\|F(y^k)\|}{1 + \|b\|}\right\} =: \alpha_k 10^{-t_k}, \ 1 \leqslant \alpha_k < 10.$$

We denote the condition number of the Jacobian of F at y^k as $\operatorname{cond}(J_k)$, and let

$$\operatorname{cond}(J_k) = \beta_k 10^{s_k}, \ 1 \leqslant \beta_k < 10.$$

We stop Algorithm Algorithm 7.1 once

$$(i) \ \ \boldsymbol{\varepsilon}^k \leqslant 10^{-13} \text{ or } (ii) \ \ s_k + t_k > 16 \text{ or } (iii) \ \ k > 2000.$$

If condition (i) holds, then we consider the **BAP** problem is solved. If condition (ii) holds, then we consider the optimal solution of the **BAP** problem as being degenerate. In our algorithm, if (ii) or (iii) hold, then we conclude that a small eigenvalue for the Jacobian exists and we assume that strict feasibility fails.[4] And, by looking at the nonzero elements of an eigenvector associated to the *smallest eigenvalue* we get information on an exposing vector; and we identify constraints

[4]Note that by Remark 2.4.6, nondegeneracy holds for our problem generically.

that give rise to the failure of strict feasibility. This solves an auxiliary system for a **FR** step, see Proposition 2.4.1. Using the information on the exposing vector, we then solve a *reduced auxiliary system*, using a *Gauss–Newton* approach[5]. This results in a **FR** step. Following this, we remove the redundant constraints that arise from the **FR** step. We repeat until strict feasibility holds.

Numerical experiments are conducted with MATLAB R2023b on a Windows 11 PC with Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz, RAM 16.0GB.

7.5.1 Comparison With(out) Strict Feasibility

As expected, our tests in Table 7.1, show that Algorithm 7.1 performs exceptionally well for instances with strict feasibility but struggles when strict feasibility fails. In fact, we observe that Algorithm 7.1 achieves the relative precision of 10^{-7} in under 7 iterations when strict feasibility holds. In contrast, when strict feasibility fails and Algorithm 7.1 converges, hundreds of iterations are needed to reach the desired precision. In Table 7.2, we repeat the same experiment setting a relative precision tolerance of 10^{-13} and allowing 2000 iteration limit. In this case, Algorithm 7.1 never reached the desired relative precision within the maximum number of iterations.

n	10	20	50	100
Slater	100%	100%	100%	100%
No Slater	55%	50%	50%	25%

Table 7.1: 20 randomly generated problems (7.1); % converged $\varepsilon^k \leqslant 10^{-8}, k \leqslant 1000$.

n	10	20	50	100
Slater	100%	100%	100%	100%
No Slater	0%	0%	0%	0%

Table 7.2: 20 randomly generated problems (7.1); % converged $\varepsilon^k \leqslant 10^{-13}, k \leqslant 2000$.

We now look at the case where the singularity degree $\operatorname{sd}(\mathcal{F}) = 1$, while the implicit singularity $\operatorname{ips}(\mathcal{F})$ varies.

$\operatorname{ips}(\mathcal{F}) = 1$

We use a spectrahedra with singularity degree 1 and $n = 15, m = 7$. The singularity degree is obtained by constructing an exposing vector as a linear combination of 5 out of the 7 constraints of the problem. Algorithm 7.1 is used to monitor the eigenvalues of the Jacobian of F at every iteration k, see Figure 7.2. We observe that only one of the eigenvalues tends to 0. After 452 iterations the method reaches a relative residual of 9.9567×10^{-8}, while the condition number of the Jacobian is 7.0236×10^{12}. Therefore the algorithm stops and indicates which of the constraints are causing strict feasibility to fail. After applying **FR** and removing the single (implicit) redundant

[5] https://github.com/j5im/FacialReductionSpectrahedron

constraint found, the algorithm now succeeds and converges to a point with a relative residual of 1.0231×10^{-15} in only 8 iterations, see Table 7.3.

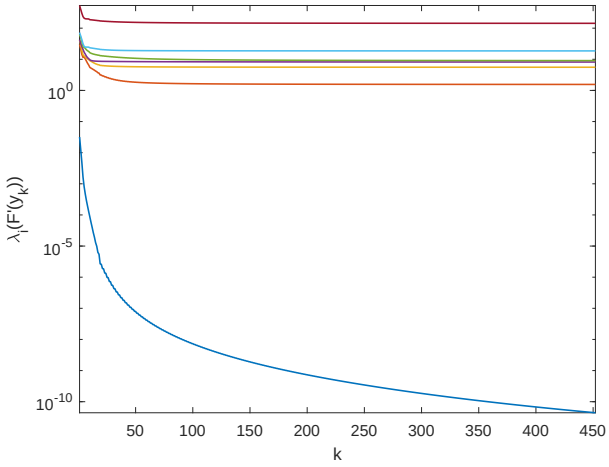

Figure 7.2: Changes in eigenvalues of Jacobian of F for spectrahedron in Section 7.5.1.

	n	m	ε^k(rel. res.)	cond($F'(y_k)$)	$\lambda_n(y_k)$	k
Before **FR**	15	7	9.9567e-08	7.0236e+12	-1.7238e-16	452
After **FR**	15	6	1.0231e-15	198.08	2.5515e-17	8

Table 7.3: Spectrahedron in Section 7.5.1; at final iteration k; before and after **FR** iters

ips($\mathcal{F}$) > 1

In our second experiment, see Table 7.4, we work with data obtained from an **SDP** relaxation of the protein side-chain positioning problems, e.g., [32]. The spectrahedron we are considering has singularity degree 1, but the implicit problem singularity is greater than 1, i.e., there are more than 1 redundant constraints after applying **FR**. In particular, the dimension of the space is $n = 35$ and the number of constraints is $m = 75$. By running our algorithm, we observe that a large number of eigenvalues of the Jacobian tend to 0 along the iterations (see Figure 7.3). After applying **FR**, we reduced the dimension of the problem to $n = 10$ and the number of constraints to $m = 22$. In the next run of the algorithm, only one eigenvalue of the Jacobian tends to 0, but we detect that a second iteration of **FR** is needed. This time, we reduce n to 9 and we remove 6 more redundant constraints, resulting in $m = 16$. The next time we apply our algorithm, the method converges to the solution in 18 iterations, see Table 7.4.

7.5.2 Experiments with Elliptope and Vontope

In this section we address the importance of strict feasibility and degeneracy on the performance of Algorithm 7.1. We consider the elliptope and vontope cases. Furthermore we compare the

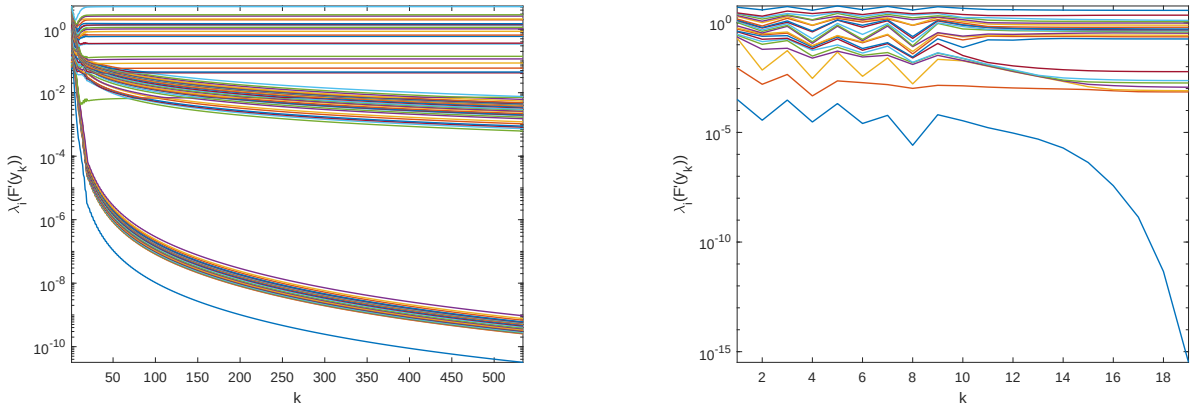

Figure 7.3: Iterations k vs eigenvalues; spectrahedron in Section 7.5.1; before and after one **FR** iteration

	n	m	ε^k(rel. res.)	cond($F'(y_k)$)	$\lambda_n(y_k)$	k
Before **FR**	35	76	8.5351e-08	1.0060e+12	-1.6941e-15	534
After **FR** 1	10	22	7.6363e-04	1.8739e+16	-5.2097e-16	19
After **FR** 2	9	16	8.7202e-14	16103.37	-5.6900e-16	18

Table 7.4: Spectrahedron in Section 7.5.1; at final iteration k; before and after **FR** iters

performance of Algorithm 7.1 with the interior point solver SDPT3[6].

From Section 7.4.3 we recall that the **MC** problem satisfies strict feasibility and every point of the elliptope, the feasible set, is nondegenerate; see Example 7.4.7. The results from the **MC** problem are displayed in the line labelled 'Elliptope' in Table 7.5. As for the **QAP**, without **FR**, the **SDP** relaxation of **QAP** fails strict feasibility and all the feasible points are degenerate. Hence in our tests, we consider two models of the same set of instances: $\mathcal{F}_{QAP}$ obtained directly by the lifting of the variables (see (7.26)); and $\mathcal{F}^{FR}_{QAP}$ obtained after **FR** is applied to $\mathcal{F}_{\mathbf{QAP}}$ (see (7.27)). In Table 7.5, **QAP** (**QAP**$_{\mathbf{FR}}$, resp.) indicates the results obtained from $\mathcal{F}_{\mathbf{QAP}}$ ($\mathcal{F}^{\mathbf{FR}}_{\mathbf{QAP}}$, resp.).

We used two settings for the choice of W in the objective function. The first setting for W forces the optimal solution $\bar{X}$ to be rank 1. Recall that rank-one optimal solutions for **QAP** are degenerate and thus lead to ill-conditioned Jacobians as can be seen by the huge condition numbers demonstrated later in Table 7.5. The second setting chooses a random W. Followed by the discussions in Remark 2.4.6 and Proposition 2.4.9, $\bar{X}$ is generically nondegenerate when strict feasibility holds.

For SDPT3 we provided the following second-order cone formulation of (7.1):

$$\min_{X,y,t} \{\ t\ : \operatorname{svec}(X) + y = \operatorname{svec}(W),\ \|y\|_2 \leqslant t,\ X \in \mathcal{F}\ \}.$$

[6] https://www.math.cmu.edu/~reha/sdpt3.html, version SDPT3 4.0 [169].

The default settings for SDPT3 were used for the tests.

Each line of Table 7.5 reports on the average of 10 instances, problem order $n = 10$. The meaning of the header names used in Table 7.5 is as follows:

(i) The headers 'pf', 'df' and 'cs' under Semi-Smooth Newton refer to the average of the primal feasibility, dual feasibility and complementarity, respectively, introduced in (7.12). The df includes both the linear dual feasibility and the violation of semidefiniteness. Both are essentially zero up to roundoff error of the arithmetic. Note that the values 10^{-15} and smaller for pf and df are essentially zero (machine precision). The headers pf, df and cs under SPDT3 refer to the solver outputs, pinfeas, dinfeas and gap, respectively.

(ii) k is the average number of iterations.

(iii) time is the average run time in cpu-seconds.

(iv) cond$(F'(y^k))$ is the average condition number of the Jacobian $(F'(y^k))$; we only have this metric for the semismooth Newton method.

W Generation	Problem	Semi-Smooth Newton						SDPT3				
		pf	df	cs	k	time	cond$(F'(y^k))$	pf	df	cs	k	time
W, rank$(\bar{X}) = 1$	Elliptope	9e-13	9e-16	2e-16	6.8	4e-02	3e+00	4e-12	6e-12	2e-07	15.5	2e-01
	$\mathbf{QAP_{FR}}$	4e-07	2e-15	1e-16	7.5	7e+00	4e+15	5e-10	1e-09	9e-06	17.9	7e+01
	QAP	8e-09	3e-15	1e-16	8.6	2e+01	4e+14	5e-10	5e-09	1e-05	18.9	6e+01
random W	Elliptope	3e-12	1e-15	6e-17	6.3	1e-02	2e+00	1e-11	6e-12	3e-08	11.5	9e-02
	$\mathbf{QAP_{FR}}$	2e-12	3e-15	7e-17	20.6	2e+01	3e+05	5e-10	5e-10	7e-07	13.9	5e+01
	QAP	1e-07	5e-13	3e-16	537.9	1e+03	6e+11	1e-08	2e-09	1e-06	17.3	7e+01

Table 7.5: Algorithm 7.1 and SDPT3 on: Elliptope and Vontope; $n = 10$;

We now discuss the results in Table 7.5. We start with the Semi-Smooth Newton, Algorithm 7.1. The 'pf' column clearly indicates that the degeneracy of the optimal point $\bar{X}$ plays a significant role. Except for random W with $\mathbf{QAP_{FR}}$, the 'pf' values for the **QAP** are poor. This correlates with the condition number values; see also the discussion in Section 7.4. The condition numbers of the Jacobian near optimal points, cond$(F'(y^k))$, become ill-conditioned when strict feasibility fails and the optimal solution is degenerate. The good measures for 'df' and 'cs' of Semi-Smooth Newton method is attributed to the construction of Algorithm 7.1. We observe that the average number of iterations for **QAP** is high when a random W is constructed, since the algorithm's difficulty in achieving the stopping tolerance ϵ, set to 10^{-7}.

SDPT3 exhibits overall strong performance on all instances. However, the 'df' and 'cs' values under SDPT3 are weaker compared to the Semi-Smooth Newton. This is due to interior point methods aiming to satisfy the first-optimality condition simultaneously throughout the process. We also observe that the number of iterations is higher when the optimal solutions are set to be degenerate.

Algorithm 7.1 has a superior performance for **MC** problems as all components of the optimality conditions are satisfied with near machine accuracy. This confirms that the status of the optimal solution plays an important role when it comes to the performance of Algorithm 7.1. In addition, preprocessing the instances so that they satisfy strict feasibility is important as seen by problems failing strict feasibility only contain degenerate points; see Theorem 2.4.7.

7.5.3 Data Availability Statement

The results and data used in this chapter is generated using our MATLAB codes. These are publicly available at the link:
https:\www.math.uwaterloo.ca/~hwolkowi/henry/reports/CodesProjDegSingDegJul2024.d/

Chapter 8

Simplified Wasserstein barycenter problem

8.1 Introduction

We consider the so-called *simplified Wasserstein barycenter,* ***WBP***, problem of finding the optimal barycenter of k points, where exactly one point is chosen from k sets of points, each set consisting of n points. This is a simplification of more general problems of *optimal mass transportation* and the problems of summarizing and combining probability measures that occur in e.g., statistics and machine learning. The latter is studied from a theoretical perspective in [1], while scalable numerical methods based on entropic regularization are proposed in [45]. The descrete problem considered in this chapter is called the *cheapest-hub problem* in [11, Def. 1.4]. Moreover, [11] derives a reduction to **WBP** from the *k-clique problem* thus proving NP-hardness.[1] Algorithms for **WBP** with exponential dependence in the *embedding dimension,* d, are discussed in [11, Sect. 1.3.1].[2] There are many important applications in molecular conformation e.g., [32], clustering [180], supervised and unsupervised learning, etc. For additional details on the theory and applications of optimal transport theory see e.g., [12,73,138], lecture link, Introduction to Transportation Problems link, and the many references therein.

The purpose of this chapter is twofold. First, we provide a successful framework for handling quadratic hard discrete optimization problems; and second, we illustrate the surprising success when applied to our specific **WBP**.

We model our problem as a quadratic objective, quadratic constrained $\{0, 1\}$ discrete optimization problem, i.e., we obtain a *binary quadratic* model where the binary constraints are equivalent to quadratic constraints $x_i^2 - x_i = 0$. We then get a convex relaxation for this hard discrete optimization problem by lifting to the doubly nonnegative, **DNN**, cone, the cone of nonnegative elementwise, positive semidefinite symmetric matrices. Strict feasibility fails for the relaxation, so we apply *facial reduction,* ***FR***. This results in many constraints becoming redundant and also gives rise to a *natural splitting* that can be exploited by the symmetric alternating directions method

[1] Recall that the *k-clique problem* is the problem of finding k vertices in a graph such that each pair is close in the sense of being *adjacent*.

[2] We discuss this further below as the complexity of our algorithm does *not* depend on the embedding dimension d.

of multipliers (**sADMM**). We exploit the structure, and include redundant constraints on the subproblems of the splitting and on the dual variables. The **sADMM** algorithm allows for efficient upper and provable lower bounding techniques for the original hard **WBP** problem. This enables earlier termination of the algorithm.

Our extensive tests on random problems are surprisingly efficient and successful, i.e., the relaxation with the upper and lower bounding techniques provide a provable optimal solution to the original hard **WBP** for *surprisingly many* instances; essentially for all our randomly generated instances. For example, to solve the original hard discrete optimization problem to optimality for our algorithm for a random problem with $k = n = 25$ in dimension $d = 25$ took on the order of 10 seconds. In contrast, using the well-known solver GUROBI, on three problems, with sizes $n = k = 5, 7, 8$, respectively, it took approximately: $2, 570, 46692$ seconds, respectively (using the same laptop for both tests). (Detailed numerics for our algorithm are provided below in Section 8.4.4.)

Though the proposed approach recovers exact solutions for nearly all randomly generated instances considered in our experiments, the **DNN** relaxation can fail to find the exact solution for problems with special structure, i.e., there can be a positive duality gap between the optimal value of the original hard problem and the lower bound found from the **DNN** relaxation. In Corollary 8.5.2 we include a constructive proof for a general hard $\{0, 1\}$ problem that a sufficient number of linearly independent optimal solutions results in a duality gap between the original hard problem and its **DNN** relaxation. This is specialized to the **WBP** in Corollary 8.5.3. A specific instance is included. Note that we consider that we have found an optimal solution to **WBP** if the upper and lower bounds are equal to machine precision as any other feasible solution cannot have a smaller objective value within machine precision.

8.1.1 Outline

The main Problem 8.2.1 and connections to Euclidean distance matrices, **EDM**, are given in Section 8.2. A regularized, facially reduced, doubly nonnegative, **DNN**, relaxation is derived in Section 8.3. The **FR** in the relaxation fits *naturally* with applying a splitting approach. This is presented in Section 8.4 along with special bounding techniques and heuristics on the dual multipliers for accelerating the splitting algorithm. The algorithm provides *provable* lower and upper bounds for the original problem in Problem 8.2.1 that are attained by dual and primal feasible solutions, respectively. Thus a *zero* gap (called a duality gap) proves optimality. Our empirics are given in Section 8.4.4. In Section 8.5 we prove that multiple optimal solutions can lead to duality gaps. We include specific instances.

Background

The **SDP** relaxation for quadratic problems with discrete variables has been successful in many applications including the classic application for the max-cut problem, e.g., [75]. The application that uses **SDP** and exploits facial reduction, **FR**, and gangster constraints, as done in this thesis, appears in e.g., [177, 182]. Then adding on nonnegativity, i.e., using the **DNN** relaxation, and exploiting the natural splitting that arises from facial reduction, appears in e.g., [32, 118, 136].

Our work combines the techniques in the latter references above and applies it to our specific **WBP** . We exploit the special **EDM** structure and illustrate the strength of applying the natural

splitting that arises with **FR**. The approach surprisingly solves generic problems to optimality; but we also find special structures that give rise to positive duality gaps, i.e., for which our technique fails to find the optimal solution.

8.2 Simplified Wasserstein barycenters, WBP

The general Wasserstein barycenter problem involves finding the mean of a collection of probability distributions by minimizing the sum of their Wasserstein distances to a target distribution, see e.g., [11]. The simplified version has a finite collection and a finite discrete distribution. We now present this as our main problem along with the connections to Euclidean distance matrices, **EDM**. We follow the notation in [11, Sect. 1.2] and refer to our main problem as the simplified Wasserstein barycenter problem, or Wasserstein barycenter for short.

8.2.1 Main problem and EDM connection

Our main optimization problem is to find a point in each of k sets to obtain an optimal barycenter. We can think of this as finding a *hub of hubs*. That is, suppose that there are k areas with n airports in each area.[3] We want to choose exactly one airport to act as a minor hub in each of the k areas so that the barycenter for these k minor hubs would serve as a major (best) hub for the k minor hubs. .

Problem 8.2.1 (*simplified Wasserstein barycenter,* ***WBP***). *Suppose that we are given a finite number of sets $S_1, ..., S_k$, each consisting of n points in $\mathbb{R}^d$. Find the optimal barycenter point y after choosing exactly one point from each set:*

$$p^*_W := \min_{\substack{p_i \in S_i \\ i \in [k]}} \min_{y \in \mathbb{R}^d} \sum_{i \in [k]} \|p_i - y\|^2 =: \min_{\substack{p_{j_i} \in S_i \\ i \in [k]}} F(p_{j_1}, p_{j_2}, \ldots, p_{j_k}), \tag{8.1}$$

thus defining F. Here

$$P^T = \begin{bmatrix} p_1 & \ldots & p_N \end{bmatrix} \in \mathbb{R}^{d \times N}, D, G, \tag{8.2}$$

denote the corresponding (configuration) matrix of points, ***EDM****, and Gram matrix, respectively. We allow the set sizes, $n_j, j \in [k]$, to vary and let $N = \sum_{j \in [k]} n_j$.*

By Lemma 8.2.2 below, the optimal Wasserstein barycenter is the standard barycenter of the k optimal points. It is known [11, Sect. 1.2] that the problem can be phrased using inter-point squared distances. We include a proof to emphasize the connection between Gram and Euclidean distance matrices.[4] We start by recording the following minimal property of the standard barycenter with respect to sum of squared distances.

Lemma 8.2.2. *Suppose that we are given k points $q_i \in \mathbb{R}^d, i = 1, \ldots k$. Let $\bar{y} = \frac{1}{k} \sum_{i=1}^k q_i$ denote the barycenter. Then*

$$\bar{y} = \operatorname*{argmin}_y \sum_{i=1}^k \frac{1}{2} \|q_i - y\|^2.$$

[3] In this hub of hubs problem, the extension allows for different sizes for the sets.

[4] As noted earlier, This is called the *cheapest-hub* problem in [11, Sect. 1.2].

Proof. The result follows from the stationary point equation $\sum_{i=1}^{k}(q_i - \bar{y}) = 0$. □

We now have the following useful lemma.

Lemma 8.2.3. *Let $Q^T = [q_1 \dots q_k] \in \mathbb{R}^{d\times k}$ and let G_Q and D_Q be, respectively, the Gram and the **EDM** matrices corresponding to the columns in Q^T. Further, let $y = \frac{1}{k}Q^T e$ be the barycenter. Then*

$$2k\sum_{i=1}^{k}\|q_i - y\|^2 = e^T D_Q e = 2k\,\mathrm{trace}(G_Q) - 2e^T G_Q e. \tag{8.3}$$

Proof. Let $J = I - ee^T/k$ be the orthogonal projection onto $e^{\perp}$. Hence, $J^2 = J^T = J$. Moreover, the i-th row $(JQ)_i = (Q - \frac{1}{k}ee^TQ)_i = (q_i - y)^T$. Now

$$\sum_{i=1}^{k}\|q_i - y\|^2 = \mathrm{trace}(JQQ^TJ) = \mathrm{trace}(JG_Q) = \mathrm{trace}(G_Q) - \frac{1}{k}e^TG_Qe.$$

But $D_Q = \mathcal{K}(G_Q) = e\,\mathrm{diag}(G_Q)^T + \mathrm{diag}(G_Q)e^T - 2G_Q$. Therefore, $e^TD_Qe = 2k\,\mathrm{trace}(G_Q) - 2e^TG_Qe$. □

The following Corollary 8.2.4 illustrates the connection between the simple Wasserstein barycenter problem[5] of finding the optimal barycenter and the k-clique problem of finding k pairwise adjacent vertices.

Corollary 8.2.4. *Let $N = \sum_{j\in[k]} n_j$, and let $S_j = \{p_{j_i}\}_{i=1}^{n_j}, j \in [k]$ be given. Consider the main problem (8.1) and let y denote the optimal Wasserstein barycenter. The problem of finding y is equivalent to finding exactly one point in each set that minimizes the sum of squared distances in the following:*

$$(WIQP) \qquad 2Np_W^* = p^* := \min_{\substack{p_t\in S_t\\ t\in[k]}} \sum_{i,j\in[k]} \|p_i - p_j\|^2. \tag{8.4}$$

Proof. Suppose that $P^T = \begin{bmatrix} p_1 & \dots & p_k \end{bmatrix} \in \mathbb{R}^{d\times k}$ is a fixed matrix of optimal solution vectors to (8.1), and let y be the barycenter. Without loss of generality, since distances do not change after a translation, we translate all the points $p_t, t \in [k]$, by y and obtain $y = 0$. This implies that the corresponding Gram matrix $Ge = PP^Te = 0$. This combined with (8.1) and (8.3) and the corresponding distance matrix D yield

$$\begin{array}{rcl} \sum_{i,j\in[k]} \|p_i - p_j\|^2 & = & e^TDe \\ & = & e^T(\mathrm{diag}(G)e^T + e\,\mathrm{diag}(G)^T - 2G)e \\ & = & 2N\,\mathrm{diag}(G)^Te \\ & = & 2N\,\mathrm{trace}\,G \\ & = & 2N\sum_{i\in[k]}\|p_i\|^2 \\ & = & 2Np_W^*, \end{array} \tag{8.5}$$

where the last equality follows from Lemma 8.2.2. □

[5] We refer to this as the Wasserstein barycenter problem.

8.2.2 A reformulation using a Euclidean distance matrix

We work with the optimal value p^* and now provide a reformulation of (8.4) using an **EDM**. Define

$$x := \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix} \in \mathbb{R}^N,\ v_i \in \mathbb{R}^{n_i}, \quad A = I \otimes e^T \in \mathbb{R}^{k \times kn}, \tag{8.6}$$

where we use the *Kronecker product*, $\otimes$. Note that we get $A^T e = e$. Then, the constraints of picking exactly one point from each set can be recast as:

$$Ax = e,\ x \in \{0,1\}^N. \tag{8.7}$$

Recalling Corollary 8.2.4 and (8.5) in the proof, we see that (8.1) can be formulated as a binary-constrained quadratic program (***BCQP***) using the Euclidean distance matrix D formed from all N points $P^T = \begin{bmatrix} p_1 & \dots & p_{n_1} & p_{n_1+1} & \dots & p_{n_1+n_2} & \dots & \dots & p_N \end{bmatrix}$:

$$\begin{array}{lllll} & p^* = & \min & x^T D x = \langle D, xx^T \rangle \\ (\mathbf{BCQP}) & & \text{s.t.} & Ax = e \\ & & & x \in \{0,1\}^N. \end{array} \tag{8.8}$$

For simplicity in the sequel we often assume that the cardinality of all sets are equal.

Remark 8.2.5 (difficulty of the Wasserstein barycenter problem)**.** *We first note that A in (8.6) is totally unimodular, i.e., every square submatrix has $\det(A_I) \in \{0, \pm 1\}$. Therefore, the basic feasible solutions, vertices of the feasible set, of the LP relaxation $Ax = e$, $x \geqslant 0$, are $\{0,1\}$ variables. Therefore, these discrete optimization problems with a linear objective yield vertices as optimal solutions and can be solved with simplex-type methods while yielding $\{0,1\}$ solutions.*

For our problem we have a quadratic objective function. And, by the properties of distance matrices, it is concave on the span of the feasible set of the LP relaxation. Therefore, minima are attained at extreme points, i.e., at vertices, at $\{0,1\}$ points. But solving the concave minimization problem is a hard problem.

In summary, the problem appears to be difficult due to the minimization of a quadratic function, [140], and the binary $0,1$ constraints. This is in contrast to the linear programming approaches for the generalized transportation problems solved in e.g., [73] and the references therein. However, the properties of total unimodularity and concavity both promote binary valued optimal points.

In addition, our problem is known to be NP-hard in the embedding dimension d, as mentioned in Section 8.1. Note that the width *of P, i.e., the embedding dimension of D, is hidden in the rank when forming the Gram matrix $G = PP^T$, and so it is hidden in the rank of the rank-two update of the **EDM***

$$D = \operatorname{diag}(G)e^T + e \operatorname{diag}(G)^T - 2G, \quad \operatorname{rank}(D) \in [\operatorname{rank}(G), \operatorname{rank}(G) + 2].$$

8.3 Facially reduced DNN relaxation

We now introduce a regularized convex relaxation to the hard binary quadratic constrained problem introduced in (8.8). We start with the **SDP** relaxation using a standard lifting approach along

with facial reduction and a gangster constraint as done in [32, Sect. 2.2]. We regularize using *facial reduction,* ***FR***, and include the so-called *gangster constraint.* We then strengthen this by including nonnegativity constraints, i.e., we get the *doubly nonnegative,* ***DNN*** relaxation.

As stated above, for simplicity of notation we consider all sets to have the same cardinality, $n_1 = \ldots = n_k, N = \sum_j n_j$.

8.3.1 Semidefinite programming (SDP) relaxation

We begin with deriving an **SDP** relaxation of our formulation in (8.8). We start with a feasible vector $x \in \mathbb{R}^{kn}$ and set $\begin{pmatrix} x_0 \\ x \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix}$. We then lift the vector to a rank-1 matrix $Y_x := \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$. The convex hull of the lifted vertices of the feasible set of (8.8) yields an equivalent polyhedral set in $\mathbb{S}^{N+1}$. It is difficult and expensive to find this polyhedral set. To obtain a tractable convex relaxation, we relax the implicit nonconvex rank-1 constraint on Y_x and linearize the objective function. Let

$$\hat{D} := \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \in \mathbb{S}^{kn+1}. \tag{8.9}$$

The objective function of (8.8) now becomes $\langle D, xx^T \rangle = \langle \hat{D}, Y_x \rangle$. After the lifting, we impose the constraints that we have from x onto Y, e.g., these include the $\{0,1\}$-constraints $x_i^2 - x_i = 0$, and the linear constraints $Ax = e$.

SDP reformulation

Define the linear transformation

$$\text{arrow} : \mathbb{S}^{n+1} \to \mathbb{R}^{n+1} : \begin{bmatrix} s_0 & s^T \\ s & \bar{S} \end{bmatrix} \mapsto \begin{pmatrix} s_0 \\ \text{diag}(\bar{S}) - s \end{pmatrix}.$$

For convenience, we define

$$\text{arrow}_0 : \mathbb{S}^{n+1} \to \mathbb{R}^{n+1} : \begin{bmatrix} s_0 & s^T \\ s & \bar{S} \end{bmatrix} \mapsto \begin{pmatrix} 0 \\ \text{diag}(\bar{S}) - s \end{pmatrix}.$$

We now show that: (i) the binary constraint on vector x is equivalent to
(ii) the rank-1 constraint with the arrow constraint on the lifted matrix, $\text{arrow}(Y_x) = e_0$.

Proposition 8.3.1. *The following holds:*

$$\left\{ Y \in \mathbb{S}_+^{kn+1} : \text{rank}(Y) = 1, \text{arrow}(Y) = e_0 \right\} = \left\{ Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T : x \in \{0,1\}^{kn} \right\}.$$

Proof. ($\supseteq$): This is clear from the definitions.
($\subseteq$): Since Y is symmetric, positive semidefinite and has rank 1, there exist $x_0 \in \mathbb{R}$ and $x \in \mathbb{R}^{kn}$ such that $Y = \begin{pmatrix} x_0 \\ x \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix}^T$. Since $\text{arrow}(Y) = e_0$, $x_0^2 = 1$ and $x \circ x = x_0 x$. If $x_0 = 1, x \in \{0,1\}^{kn}$;

otherwise $x_0 = -1$ and $x \in \{0, -1\}^n$ and it is easy to verify that

$$\left\{ \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T : x \in \{0,1\}^{kn} \right\} = \left\{ \begin{pmatrix} -1 \\ x \end{pmatrix} \begin{pmatrix} -1 \\ x \end{pmatrix}^T : x \in \{0,-1\}^{n} \right\}.$$

□

For the "only-one-element-from-each-set" linear equality constraint $Ax = e$ (see (8.7)), we use the following positive semidefinite matrix

$$K := \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T \in \mathbb{S}_+^{kn+1}. \tag{8.10}$$

We observe that

$$\begin{aligned} Ax = e \quad &\iff \quad \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix} = 0 \\ &\iff \quad Y_x K = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T = 0 \\ &\iff \quad KY_x = 0, \end{aligned} \tag{8.11}$$

i.e., $\mathrm{range}(Y_x) \subseteq \mathrm{null}(K) = \mathrm{null}\left(\begin{bmatrix} -e & A \end{bmatrix}\right)$. Moreover, this emphasizes that strict feasibility fails for feasible Y even if we ignore the rank-1 constraint. If we choose V full column rank so that $\mathrm{range}(V) = \mathrm{null}(K)$, then we can *facially reduce* the problem using the substitution

$$Y \leftarrow VRV^T \in V\mathbb{S}_+^{nk+1-k}V^T \trianglelefteq \mathbb{S}_+^{kn+1}, \tag{8.12}$$

where $\trianglelefteq$ denotes *face of*. This makes the constraint $KY = 0$ redundant. More detailed discussion for constructing the facial vector V is provided later in Section 8.3.2.

Then the rank restricted **SDP** reformulation of (8.8) becomes

$$(\textbf{SDP}\ ) \qquad \begin{array}{lll} p^* = & \min & \langle \hat{D}, Y \rangle \\ & & \mathrm{arrow}(Y) = e_0 \\ & & \mathrm{rank}(Y) = 1 \\ & & KY = 0 \\ & & Y \in \mathbb{S}_+^{kn+1}. \end{array}$$

Relaxing the rank-1 constraint

Since the **NP**-hardness of the **SDP** formulation comes from the rank-1 constraint, we now relax the problem by deleting this constraint. The **SDP** relaxation of the above model is

$$(\textbf{SDP}\ \ \text{relax}) \qquad \begin{array}{lll} p^* = & \min_{Y \in \mathbb{S}^{kn+1}} & \langle \hat{D}, Y \rangle \\ & & \mathrm{arrow}(Y) = e_0 \\ & & KY = 0 \\ & & Y \succeq 0. \end{array} \tag{8.13}$$

However, the improved processing efficiency of this convex relaxation trades off with the accuracy of solving the original NP-hard problem. The rank of an optimal Y can now be greater than one. The idea now is to impose a "correct" amount of redundant constraints in the **SDP** model that reduces the rank of an optimal solution as much as possible, but does not hurt the processing efficiency of the model too much. Note that if we have ignored the rank one constraint and replaced the $KY = 0$ constraint using facial reduction and substituting $Y = VRV^T$, then strict feasibility holds, i.e., the barycenter of the the lifted vertices of the feasible set of (8.8) yields an $\hat{R}, Y = V\hat{R}V^T$, that satisfies strict feasibility, e.g., [63].

The gangster constraint

The *gangster constraint* is essentially a trivial projection that fixes at 0 (shoots holes at) certain entries of the matrix. The entries are given in the *gangster index,* $\mathcal{J}$. This constraint for **SDP** relaxations first appeared in [177, 182] and more recently in many other applications, see e.g., [63] for summaries. In combination with **FR** , it has been shown to provide significant strengthenings for **SDP** relaxations. In the **SDP** case, redundant gangster constraints are identified and discarded to avoid ill-conditioning of the linear system. For our subproblems it is just important to impose the constraints efficiently.

The gangster constraint in our case comes from the linear constraint $Ax = e$ combined with the binary constraint on x. We include a (novel) proof for completeness. We let $S \circ T$ denote the Hadamard (elementwise) product.

Proposition 8.3.2. *Let x be feasible for **BCQP**. Then*

$$[A^TA - I] \circ xx^T = 0,$$

and $A^TA - I \geqslant 0, xx^T \geqslant 0$.

Proof. Recall that $x \in \mathbb{R}^{kn}_+$. We now use basic properties of the Kronecker product, e.g., [155], and see that

$$A = I_k \otimes e^T,\ A^T = I_k \otimes e, \quad A^TA = I_k \otimes ee^T,$$

i.e., A^TA, has a block diagonal structure, and the columns of A are unit vectors. Therefore $A^Te_k = e_{nk}$ and $\mathrm{Diag}(\mathrm{diag}(A^TA)) = I_{kn}$. The nonnegativity results follow from the definition, as does $Y_{00} = 1$.

Then

$$\begin{aligned}
Ax = e &\implies A^TAx = A^Te \\
&\implies A^TAx - Ix = A^Te - Ix \\
&\implies (A^TA - I)x = e_{nk} - x \\
&\implies (A^TA - I)xx^T = (e - x)x^T = ex^T - xx^T \\
&\implies \mathrm{trace}[(A^TA - I)xx^T] = \mathrm{trace}[ex^T - xx^T] = \textstyle\sum_{i=1}^{kn} x_i - x_i^2 = 0 \\
&\implies (A^TA - I) \circ xx^T = 0.
\end{aligned}$$

The final conclusion now follows from the nonnegativities in the Hadamard product. □

Define the gangster indices

$$\mathcal{J} := \left\{(i,j) : \left(A^T A - I\right)_{ij} > 0\right\}.$$

The gangster constraint on Y in (8.13) is $Y_{00} = 1$ and

$$\mathcal{J}(Y) = Y_{\mathcal{J}} = 0 \in \mathbb{R}^{|\mathcal{J}|}.$$

From Proposition 8.3.2, we see that the *gangster indices,* $\mathcal{J}$, are the nonzeros of the matrix $A^T A - I$, i.e., the set of off-diagonal indices of the n-by-n diagonal blocks of, all but the zeroth row and column, of Y_x. We define the *gangster constraint mapping,* $\mathcal{G}_{\mathcal{J}}$:

$$\mathcal{G}_{\mathcal{J}}(Y) = Y(\mathcal{J}) \in \mathbb{R}^{|\mathcal{J}|},$$

i.e., the elements of Y indexed by the index set $\mathcal{J}$.

Now the **SDP** relaxation model becomes

$$\begin{array}{rll} p^* = & \min_{Y \in \mathbb{S}^{kn+1}} & \langle \hat{D}, Y \rangle \\ & & Y_{00} = 1 \\ & & \text{arrow}_0(Y) = 0 \\ & & \mathcal{G}_{\mathcal{J}}(Y) = 0 \\ & & KY = 0 \\ & & Y \succeq 0. \end{array} \tag{8.14}$$

Proposition 8.3.3. *Consider the* ***SDP*** *relaxation* (8.14) *but without the* arrow_0 *constraint. Then every optimal solution satisfies the constraint*

$$\text{arrow}_0(Y) = 0,$$

i.e., it was a redundant constraint.

Proof. It follows from [32, Thm 2.1]. □

Our empirical tests on random problems without the arrow_0 constraint confirmed this result. However, the extra redundant constraint is useful for the subproblems in the splitting approach below.

8.3.2 Doubly nonnegative (DNN) relaxation

We now split the problem by using two variables $\{Y, R\}$ and apply a doubly nonnegative relaxation to (8.14). This *natural splitting* uses the facial reduction obtained in (8.12) but with orthonormal columns chosen for the facial vector V.

Formulating facial vector V for sparsity

In this subsection, we suggest strategies for constructing sparse versions of the *facial vector,* V. Recall that the columns of the facial vector V form an orthonormal basis for the nullspace of K.

For a typical matrix V see Figure 8.1 that is constructed using Lemma 8.3.4, below. Alterna-

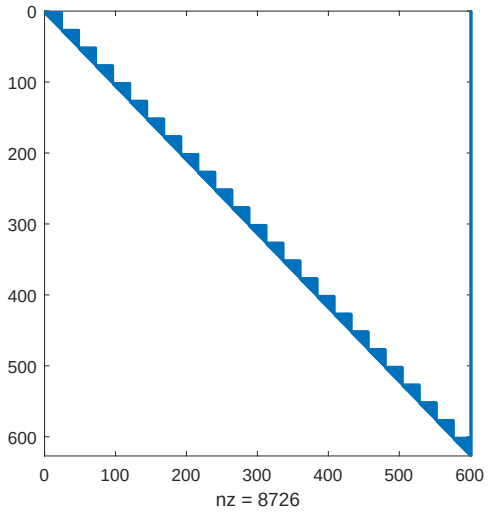

Figure 8.1: V matrix for $k = 25, n = 25$

tively, we can use the MATLAB QR algorithm (with A specified as sparse) $[q, \sim] = qr(-e \quad A)$ and use the last part of q for the nullspace. This results in a relatively sparse orthonormal basis for the nullspace.

Lemma 8.3.4. *Let k, n be given positive integers and from above let*

$$A = \left[I_k \otimes e_n^T\right], \; B = \left[-e_k \quad A\right].$$

Let $\mathcal{O} \in \mathbb{R}^{n-1\times n-1}$ be the strictly upper triangular matrix of ones of order $n-1$. Set

$$v = \left(\frac{1}{\sqrt{j+j^2}}\right)_j \in \mathbb{R}^{n-1}, \; \bar{v} = \left(\frac{j}{\sqrt{j+j^2}}\right)_j \in \mathbb{R}^{n-1}, \; \beta = -1/\sqrt{n^2+nk}, \text{ and } \alpha = n\beta.$$

Let $\tilde{\mathcal{O}} = -\mathcal{O}\,\mathrm{Diag}(v) + \mathrm{Diag}(\bar{v})$ and set

$$\bar{\mathcal{O}} = \begin{bmatrix} -v^T \\ \tilde{\mathcal{O}} \end{bmatrix} = \begin{bmatrix} -v_1 & -v_2 & -v_3 & \cdots & -v_{n-1} \\ \bar{v}_1 & -v_2 & -v_3 & \cdots & -v_{n-1} \\ 0 & \bar{v}_2 & -v_3 & \cdots & -v_{n-1} \\ 0 & 0 & \bar{v}_3 & \cdots & -v_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{v}_{n-1} \end{bmatrix}.$$

Then we have

$$V = \begin{bmatrix} 0 & \alpha \\ I_k \otimes \bar{\mathcal{O}} & \beta e \end{bmatrix} \in \mathbb{R}^{nk+1\times(n-1)k+1}, \quad V^T V = I, \; BV = 0.$$

Proof. Denote the j-th column of V by V_j and define $J_s := \{j_1^s, j_2^s, \ldots, j_{n-1}^s\}$, where $j_r^s = (n-1)(s-1)+r$. Notice that J_s is the index set of columns of V in s-th block. $j \in J_{k+1}$ means V_j is the last column of V.

We first prove that $V^TV = I$, i.e., column vectors of V are orthonormal. Let $i, j \in \{1, \ldots, (n-1)k+1\}$. We consider the following cases:
If $j \leqslant (n-1)k$, then

$$V_j^T V_j = jv_j^2 + \bar{v}_j^2 = \frac{j}{j+j^2} + \frac{j^2}{j+j^2} = 1.$$

If $j = (n-1)k+1$, then

$$V_j^T V_j = \alpha^2 + nk\beta^2 = (n^2+nk)\beta^2 = 1.$$

Now let $i < j$. If $i, j \in J_s$ for some $s \leqslant k$. Then,

$$\begin{aligned} V_i^T V_j &= iv_i v_j - \bar{v}_i v_j \\ &= i \cdot \frac{1}{\sqrt{i+i^2}} \frac{1}{\sqrt{j+j^2}} - \frac{i}{\sqrt{i+i^2}} \frac{1}{\sqrt{j+j^2}} = 0. \end{aligned}$$

If $j = (n-1)k+1$. Then,

$$V_i^T V_j = -iv_i\beta + \bar{v}_i\beta = (-iv_i + iv_i)\beta = 0.$$

If $i \in J_s$, $j \in J_t$ with $s < t \leqslant k$. For each row, at least one of the vectors has 0 entry, so trivially $V_i^T V_j = 0$. This proves the orthonormality.

Secondly, we observe $BV = 0$, i.e., $\operatorname{range} V \subseteq \operatorname{null}(B)$. To this end, we will see that $BV_j = 0$ for each $j = 1, \ldots, (n-1)k+1$. Fix $s \in \{1, \ldots, k\}$. If $j = (n-1)k+1$,

$$\left(BV_j\right)_s = -\alpha + n\beta = -n\beta + n\beta = 0,$$

Now assume that $j \leqslant (n-1)k$. If $j \in J_s$, then

$$\left(BV_j\right)_s = -jv_j + \bar{v}_j = -jv_j + jv_j = 0, \text{ for each } i = 1, \ldots, k.$$

Otherwise, trivially $\left(BV_j\right)_s = 0$. This justifies $BV = 0$. □

We leave open the question on how to exploit the structure of V to obtain efficient matrix-matrix multiplications of the form VRV^T needed in our algorithm.

DNN reformulation via facial reduction

Recall that the lifting for Y_x has the form

$$Y_x = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T, \quad x \in \{0,1\}^{kn}.$$

Hence, we can impose the redundant elementwise **LP** relaxation bound constraint $0 \leqslant Y \leqslant 1$. Moreover, the constraint $KY = 0$ is equivalent to applying **FR**, i.e., we get

$$Y \succeq 0, KY = 0 \iff Y = VRV^T, R \in \mathbb{S}_+^{nk+1-k}.$$

We now observe a useful redundant trace constraint on Y and transform it onto R.

Lemma 8.3.5. *Let* $Y \in \mathbb{S}^{kn+1}, Y = VRV^T, R \in \mathbb{S}^{nk+1-k}$. *Then*

$$KY = 0, \text{arrow}(Y) = e_0 \implies \text{trace}(Y) = \text{trace}(R) = k + 1.$$

Proof. Recall that $K := \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T$. Since $\text{null}(K) = \text{null}\left(\begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T \right)$, we have

$$0 = KY \iff 0 = \begin{bmatrix} -1 & e^T & \cdots & 0^T \\ \vdots & \vdots & \ddots & \vdots \\ -1 & 0^T & \cdots & e^T \end{bmatrix} \begin{bmatrix} Y_{0.0} & \cdots & Y_{0,nk} \\ \vdots & \ddots & \vdots \\ Y_{nk,0} & \cdots & Y_{nk,nk} \end{bmatrix}.$$

By expanding the first column of the product, we get $\sum_{i=1}^{n} Y_{jn+i,0} = 1, \forall j \in \{0, ..., k-1\}$. Since $\text{arrow}(Y) = e_0$, this implies that $\text{trace}(Y) = Y_{0,0} + \sum_{j=0}^{k-1} \sum_{i=1}^{n} Y_{jn+i,0} = 1 + k$. Since we choose the facial vector V to have orthonormal columns, the facial constraint yields

$$1 + k = \text{trace}(Y) = \text{trace}(VRV^T) = \text{trace}(RV^TV) = \text{trace}(R),$$

□

Next, we incorporate all these constraints into the **SDP** relaxation model to form the **DNN** relaxation model. Define the two convex set constraints

$$\mathcal{Y} := \{Y \in \mathbb{S}^{nk+1} : Y_{00} = 1, \mathcal{G}_{\mathcal{J}}(Y) = 0, \text{arrow}_0(Y) = 0, 0 \leqslant Y \leqslant 1\}, \ \mathcal{R} := \{R \in \mathbb{S}^{nk+1-k}_{+} : \text{trace}(R) = k+1\}. \tag{8.15}$$

Our ***DNN*** *relaxation* model is:

$$(\textbf{DNN}) \qquad \begin{array}{lll} p^*_{\textbf{DNN}} := \min_{R,Y} & \langle \hat{D}, Y \rangle \\ \text{s.t.} & Y = VRV^T \\ & Y \in \mathcal{Y} \\ & R \in \mathcal{R}. \end{array} \tag{8.16}$$

Observe that every feasible Y is both element-wise nonnegative and ***PSD***, i.e., this is a **DNN** relaxation. Moreover, by construction, **DNN** is trivially a feasible problem and the constraint $Y - VRV^T = 0$ is trivially surjective. Therefore, we have a regularized problem.

The splitting allows for the two cones to be handled separately. Combining them into one and applying e.g., an interior point approach is known to be very costly. And, one cannot get high dual feasibility accuracy and therefore the approximate optimal value we get is not a provable lower bound for Problem 8.2.1. In summary, the expense does not scale well with N, and we cannot apply weak duality and use the dual solution as both primal and dual feasibility are not highly accurate. We overcome this problem for the splitting method in Section 8.4.1 below.

The first-order optimality conditions are also given in [32]. We provide the proof of the optimality conditions in the following Proposition 8.3.6.

Proposition 8.3.6 (characterization of optimality for **DNN** in (8.16))**.** *The primal-dual pair*

$(\bar{Y}, \bar{R}, \bar{Z})$ *is optimal for* (8.16) *if, and only if,*

$$\bar{R} = \mathcal{P}_{\mathcal{R}}(\bar{R} + V^T \bar{Z} V) \tag{8.17a}$$
$$\bar{Y} = \mathcal{P}_{\mathcal{Y}}(\bar{Y} - \hat{D} - \bar{Z}) \tag{8.17b}$$
$$\bar{Y} = V \bar{R} V^T. \tag{8.17c}$$

Proof. Let the Lagrangian function be:

$$\mathcal{L}(Y, R, Z) = \langle \hat{D}, Y \rangle + \langle Z, Y - V R V^T \rangle.$$

For any given Z, the domain of $\mathcal{L}$ is the entire space $\mathbb{S}^{nk+1} \times \mathbb{S}^{nk+1-k}$. This provides a valid constraint qualification, **CQ**, for the simple constrained minimization of the Lagrangian function. Therefore, see [151], the optimality conditions for (8.16) with the normal cone, $\mathcal{N}_{\mathcal{Y}\times\mathcal{R}}(\bar{Y}, \bar{R})$, are:

$$\begin{array}{ll} \nabla_{Y,R}\mathcal{L}(\bar{Y}, \bar{R}, \bar{Z}) \in -\mathcal{N}_{\mathcal{Y}\times\mathcal{R}}(\bar{Y}, \bar{R}). & \text{(dual feasibility)} \\ \bar{Y} = V \bar{R} V^T \ \bar{Y} \in \mathcal{Y}, \ \bar{R} \in \mathcal{R} & \text{(primal feasibility)} \end{array}$$

In addition, the normal cone at $(\bar{Y}, \bar{R})$ satisfies:

$$\begin{array}{rcl} \mathcal{N}_{\mathcal{Y}\times\mathcal{R}}(\bar{Y}, \bar{R}) & = & \{(Y, R) : \langle (Y, R), (\hat{Y} - \bar{Y}, \hat{R} - \bar{R}) \rangle \leqslant 0, \ \forall (\hat{Y}, \hat{R}) \in \mathcal{Y} \times \mathcal{R}\} \\ & = & \{Y : \langle Y, \hat{Y} - \bar{Y} \rangle \leqslant 0, \ \forall \hat{Y} \in \mathcal{Y}\} \times \{R : \langle R, \hat{R} - \bar{R} \rangle \leqslant 0, \ \forall \hat{R} \in \mathcal{R}\} \\ & = & \mathcal{N}_{\mathcal{Y}}(\bar{Y}) \times \mathcal{N}_{\mathcal{R}}(\bar{R}). \end{array}$$

Therefore the first-order optimality conditions to the problem in (8.16) are: a primal-dual pair $(\bar{Y}, \bar{R}, \bar{Z})$ is optimal if, and only if,

$$0 \in -V^T \bar{Z} V + \mathcal{N}_{\mathcal{R}}(\bar{R}) \qquad \text{(dual } \bar{R} \text{ feasibility)} \tag{8.18a}$$
$$0 \in \hat{D} + \bar{Z} + \mathcal{N}_{\mathcal{Y}}(\bar{Y}) \qquad \text{(dual } \bar{Y} \text{ feasibility)} \tag{8.18b}$$
$$\bar{Y} = V \bar{R} V^T, \quad \bar{R} \in \mathcal{R}, \ \bar{Y} \in \mathcal{Y} \qquad \text{(primal feasibility)} \tag{8.18c}$$

By the definition of the normal cone, we conclude that (8.18) holds if, and only if, (8.17) holds. □

8.4 sADMM algorithm; bounding; empirics

The augmented Lagrangian corresponding to the **DNN** relaxation (8.16) with parameter $\beta > 0$ is

$$\mathcal{L}_\beta(Y, R, Z) := \langle \hat{D}, Y \rangle + \langle Z, Y - V R V^T \rangle + \frac{\beta}{2} \|Y - V R V^T\|_F^2 + \iota_{\mathcal{Y}}(Y) + \iota_{\mathcal{R}}(R). \tag{8.19}$$

To solve our **DNN** relaxation in (8.16), we use the symmetric alternating directions method of multipliers **sADMM** that has intermediate updates of dual multipliers Z_t: one dual update after the R-update and then another after the Y-update. This approach has been used successfully in [32, 118]. Hence, both the R-update and the Y-update take into account newly updated dual variable information. (We include the details here for completeness.)

Let $Y_0 \in \mathbb{S}^{nk+1}, Z_0 \in \mathbb{S}^{nk+1}$. The updates for all nonnegative integers $k \in \mathbb{Z}_+$ are:

$$\begin{array}{rcl} R_{k+1} & = & \operatorname{argmin}_{R \in \mathbb{S}^{nk+1-k}} \mathcal{L}_\beta(Y_k, R, Z_k) \\ Z_{k+\frac{1}{2}} & = & Z_k + \beta(Y_k - V R_{k+1} V^T) \\ Y_{k+1} & = & \operatorname{argmin}_{Y \in \mathbb{S}^{nk+1}} \mathcal{L}_\beta(Y, R_{k+1}, Z_{k+\frac{1}{2}}) \\ Z_{k+1} & = & Z_{k+\frac{1}{2}} + \beta(Y_{k+1} - V R_{k+1} V^T). \end{array} \tag{8.20}$$

In our **DNN** model (8.16), the objective function is continuous and the feasible set is compact. By the Weierstrass Theorem, an optimal primal pair (Y^*, R^*) always exists. As seen above, the constraint is linear and surjective and strong duality holds for the generalized **KKT** conditions. (See the optimality conditions in Proposition 8.3.6). In fact, in our application we modify the dual multiplier update using a projection, see Lemma 8.4.1 and Algorithm 8.1.

Explicit Primal updates for R, Y

The success of our splitting method is dependent on efficiently solving the subproblems. We start with using a spectral decomposition, implicitly defined below, to get the:

$$\begin{array}{rcll} R-\text{update} & = & \operatorname{argmin}_{R \in \mathbb{S}^{nk+1-k}} \mathcal{L}_\beta(Y_k, R, Z_k) & \\ & = & \operatorname{argmin}_{R \in \mathcal{R}} \|Y_k - V R V^T + \frac{1}{\beta} Z_k\|_F^2, & \text{by completing the square} \\ & = & \operatorname{argmin}_{R \in \mathcal{R}} \|V^T Y_k V - R + \frac{1}{\beta} V^T Z_k V\|_F^2, & \text{since } V^T V = I \\ & = & \operatorname{argmin}_{R \in \mathcal{R}} \|R - V^T(Y_k + \frac{1}{\beta} Z_k) V\|_F^2 & \\ & = & \mathcal{P}_{\mathcal{R}}[V^T(Y_k + \frac{1}{\beta} Z_k) V] & \\ & = & U \operatorname{Diag}[\mathcal{P}_{\Delta_{k+1}}(\lambda)] U^T, & \text{spectral decomposition} \end{array}$$

where the $U \operatorname{Diag}(\lambda) U^T$ provides the spectral decomposition of $V^T(Y_k + \frac{1}{\beta} Z_k) V$ and then $\mathcal{P}_{\Delta_{k+1}}$ denotes the projection onto the *simplex* $\Delta_{k+1} := \{x \in \mathbb{R}^n_+ : \langle e, x \rangle = 1 + k\}$, see e.g., [40].

Next for the Y-update, we define the *gangster set* $\hat{\mathcal{G}}_{\mathcal{J}} := \{Y \in \mathbb{S}^{nk+1} : Y_{00} = 1, \mathcal{G}_{\mathcal{J}}(Y) = 0\}$. Then,

$$\begin{array}{rcll} Y-\text{update} & = & \operatorname{argmin}_{Y \in \mathbb{S}^{nk+1}} \mathcal{L}_\beta(Y, R_{k+1}, Z_{k+\frac{1}{2}}) & \\ & = & \operatorname{argmin}_{Y \in \mathcal{Y}} \|Y - [V R_{k+1} V^T - \frac{1}{\beta}(\hat{D} + Z_{k+\frac{1}{2}})]\|_F^2 & \text{by completing the square} \\ & = & \mathcal{P}_{\mathcal{Y}} \left(V R_{k+1} V^T - \frac{1}{\beta}(\hat{D} + Z_{k+\frac{1}{2}}) \right) & \\ & = & \mathcal{P}_{\text{arrowbox}} \left(\mathcal{P}_{\hat{\mathcal{G}}_{\mathcal{J}}}[V R_{k+1} V^T - \frac{1}{\beta}(\hat{D} + Z_{k+\frac{1}{2}})] \right), & \end{array}$$

where $\mathcal{P}_{\text{arrowbox}}$ projects onto the polyhedral set $\{Y \in \mathbb{S}^{nk+1} : Y_{ij} \in [0,1], \operatorname{arrow}(Y) = e_0\}$.

Dual updates

The correct choice of the Lagrange dual multiplier Z is important in the progress of the algorithm and in obtaining strong lower bounds. In addition, if the set of dual multipliers for all iterations is compact, then it indicates the stability of the primal problem. Lagrange multipliers are used to replace constraints by adding the appropriate expression into the Lagrangian function. If an optimal Z^* for (8.16) is known in advance, then it makes sense that we do not need the corresponding primal

feasibility constraint $Y = VRV^T$. Hence, following the idea of exploiting redundant constraints, we now aim to identify certain properties of an optimal dual multiplier and impose that property at each iteration of our algorithm.

Lemma 8.4.1. *Let*

$$\mathcal{Z}_A := \left\{ Z \in \mathbb{S}^{kn+1} : (Z + \hat{D})_{i,i} = 0, (Z + \hat{D})_{0,i} = 0, (Z + \hat{D})_{i,0} = 0, i = 1, ..., nk \right\}.$$

Let (Y^*, R^*, Z^*) *be an optimal primal-dual pair for the* ***DNN*** *in* (8.16)*. Then,* $Z^* \in \mathcal{Z}_A$*.*

Proof. The proof of this fact uses the dual Y feasibility condition (8.18b) and a reformulation of the Y-feasible set. The details are in [81, Thm 2.1] and [32]. □

In view of Lemma 8.4.1 we propose the following modification of the symmetric **ADMM** algorithm, e.g., [89]. Our modification is in the way we update the multiplier. At every initial or intermediate update of the multiplier we project the dual variable onto $\mathcal{Z}_A$, i.e:

- $Z_{j+\frac{1}{2}} := Z_j + \gamma\beta\mathcal{P}_{\mathcal{Z}_A}(Y_j - VR_{j+1}V^T)$;
- $Z_{j+1} := Z_{j+\frac{1}{2}} + \gamma\beta\mathcal{P}_{\mathcal{Z}_A}(Y_{j+1} - VR_{j+1}V^T)$.

Note that a convergence proof using the modified updates is given in [81, Thm 3.2]. Therefore, in view of the **ADMM** updates (8.20) we propose the following Algorithm 8.1 with modified Z updates. The $\gamma \in (0, 1)$ is the dual steplength.

Algorithm 8.1 sADMM, modified symmetric **ADMM**

Initialization: $j = 0, Y_j = 0 \in S^{nk+1}, Z_j = \mathcal{P}_{\mathcal{Z}_A}(0), \beta = \max(\lfloor \frac{nk+1}{k} \rfloor, 1), \gamma = 0.9$
while termination criteria are not met **do**
  $R_{j+1} = U\,\mathrm{Diag}[\mathcal{P}_{\Delta_{j+1}}(d)]U^T$ where $U\,\mathrm{Diag}(d)U^T = \mathrm{eig}(V^T(Y_j + \frac{1}{\beta}Z_j)V)$
  $Z_{j+\frac{1}{2}} = Z_j + \gamma\beta\mathcal{P}_{\mathcal{Z}_A}(Y_j - VR_{j+1}V^T)$
  $Y_{j+1} = \mathcal{P}_{\text{arrowbox}}[\mathcal{P}_{\hat{\mathcal{G}}_{\mathcal{J}}}(VR_{j+1}V^T - \frac{1}{\beta}(\hat{D} + Z_{j+\frac{1}{2}}))]$
  $Z_{j+1} = Z_{j+\frac{1}{2}} + \gamma\beta\mathcal{P}_{\mathcal{Z}_A}(Y_{j+1} - VR_{j+1}V^T)$
  $j = j + 1$
end while

Remark 8.4.2. *For convergence in Algorithm 8.1 we can choose any* $\gamma \in (0, 1)$ *and* $\beta > 0$ *as discussed in [88, 118]. In our numerical experiments for Algorithm 8.1 we used an adaptive* β *based on the discussion in Section 8.4.3. The convergence rate is linear for* β *fixed and remains linear for adaptive* β *in the worst case analysis.*

8.4.1 Bounding and duality gaps

Strong upper and lower bounds allow for early stopping conditions as well as proving optimality. We now provide provable upper and lower bounds to machine precision.

Provable lower bound to NP-hard problem

A provable lower bound using a special Lagrangian duality for an **sADMM** type algorithm was successfully used in [136]. Another example where accurate lower bounds are critical is given in [97].

Here the Lagrangian dual function $g : \mathbb{S}^{nk+1} \to \mathbb{R}$ to the **DNN** model that we use is

$$\begin{array}{rcl} g(Z) & = & \min_{R\in\mathcal{R},Y\in\mathcal{Y}}\langle \hat{D}, Y\rangle + \langle Z, Y - VRV^T\rangle \\ & = & \min_{Y\in\mathcal{Y},R\in\mathcal{R}}\langle \hat{D} + Z, Y\rangle - \langle Z, VRV^T\rangle \\ & = & \min_{Y\in\mathcal{Y}}\langle \hat{D} + Z, Y\rangle + \min_{R\in\mathcal{R}}(-\langle V^TZV, R\rangle) \\ & = & \min_{Y\in\mathcal{Y}}\langle \hat{D} + Z, Y\rangle - \max_{R\in\mathcal{R}}\langle V^TZV, R\rangle \\ & = & \min_{Y\in\mathcal{Y}}\langle \hat{D} + Z, Y\rangle - \max_{\|v\|^2=(k+1)} v^TV^TZVv \\ & = & \min_{Y\in\mathcal{Y}}\langle \hat{D} + Z, Y\rangle - (k+1)\lambda_{\max}(V^TZV). \end{array}$$

Hence, at iteration j, and applying weak duality, a lower bound to the optimal value of the **DNN** model (8.16) is

$$\begin{array}{rcl} p^*_{\mathbf{DNN}} & \geqslant & \max_Z g(Z) \\ & \geqslant & \min_{Y\in\mathcal{Y}}\langle \hat{D} + Z_j, Y\rangle - (k+1)\lambda_{\max}(V^TZ_jV). \end{array} \tag{8.21}$$

Note that from the definition of $\mathcal{Y}$ in (8.15), this bound is found from solving: an **LP** with a simplex-type feasible set; and an eigenvalue problem. Thus both values can be found accurately and efficiently. Moreover, since **DNN** is a relaxation, weak duality implies that this lower bound is a *provable lower bound* for the original NP-hard Problem 8.2.1.

Upper bounds

As for the upper bound, we consider two strategies for finding feasible solutions to the **BCQP** in (8.8). The 0-column approach is to take all but the first element of this zeroth column $Y(1\!:\!\text{end}, 0)$ and compute its nearest feasible solution to **BCQP**. It is equivalent to the greedy approach of using only the maximum weight index for each consecutive block of length n, see [32, Section 3.2.2].

Alternatively, we use the eigenvector of Y corresponding to the largest eigenvalue. The Perron-Frobenius Theorem implies this eigenvector is nonnegative, as Y is nonnegative. We then compute the nearest feasible solution to **BCQP**. It is again equivalent to the greedy approach but with using the eigenvector.

Then, we compare the objective values for both approaches and select the upper bound with smaller magnitude. The relative duality gap at the current iterate j is defined to be $\frac{UB_j - LB_j}{|UB_j| + |LB_j| + 1}$ where UB_j, LB_j denote the current best upper, and lower bound, respectively.

8.4.2 Stopping criterion

By Proposition 8.3.6, we can define the primal and dual residuals of the **sADMM** algorithm at iterate j as follows:

- Primal residual $r_j := \|Y_j - VR_jV^T\|$;
- Dual-R residual $s_j^R := \|R_j - \mathcal{P}_{\mathcal{R}}\left(R_j + V^TZ_jV\right)\|$;

- Dual-Y residual $s_j^Y := \left\| Y_j - \mathcal{P}_{\mathcal{Y}} \left(Y_j - \hat{D} - Z_{j+\frac{1}{2}} \right) \right\|$.

We terminate the algorithm once one of the following conditions is satisfied:

- The maximum number of iterations (maxiter) := $10^4 + k(nk+1)$ is reached;
- The relative duality gap is less or equal to ϵ, a given tolerance;
- KKTres := $\max\{r_j, s_j^R, s_j^Y\} < \eta$, a given tolerance.
- Both the least upper bound and the greatest lower bound have not changed for boundCounterMax := 200 times (stalling).

8.4.3 Heuristics for algorithm acceleration

Adaptive step size

We apply the heuristic idea presented in [30], namely we bound the gap between the primal and dual residual norms within a factor of $\mu := 2$ as they converge to 0. This guarantees that they converge to 0 at about the same rate and one residual does not overshoot the other residual by too much. Since a large penalty β prioritizes primal feasibility over dual feasibility and a small penalty β prioritizes dual feasibility over primal feasibility, we scale β by a factor of $\tau^{\text{incr}} := 2$ if the primal residual overshoots the dual residual by a factor of μ and scale β down by a factor of $\tau^{\text{decr}} := 2$ if the dual residual overshoots the primal residual by a factor of μ. Otherwise, we keep β unchanged. Namely,

$$\beta_{j+1} := \begin{cases} \tau^{\text{incr}} \beta_j, & \|r_j\|_2 > \mu \|s_j\|_2 ; \\ \frac{\beta_j}{\tau^{\text{decr}}}, & \|s_j\|_2 > \mu \|r_j\|_2 ; \\ \beta_j, & \text{otherwise.} \end{cases}$$

Transformation and scaling

In this subsection, we consider translating and scaling the objective function, i.e., $\hat{D}$. Define the orthogonal projection map $P_V := VV^T$. Then,

$$\begin{aligned} \langle \hat{D}, Y \rangle &:= \langle \hat{D} + \alpha I, Y \rangle - (n+1)\alpha \\ &= \langle \hat{D} + \alpha I, P_V Y P_V \rangle - (n+1)\alpha \\ &= \langle (P_V \hat{D} P_V + \alpha I), Y \rangle - (n+1)\alpha. \end{aligned} \tag{8.22}$$

Hence, when finding the optimal Y,

$$\begin{aligned} \langle \hat{D}, Y \rangle \text{ is minimized} &\iff \delta \langle \hat{D}, Y \rangle = \langle \delta(P_V \hat{D} P_V + \alpha I), Y \rangle - (n+1)\delta\alpha \text{ is minimized} \\ &\iff \langle \delta(P_V \hat{D} P_V + \alpha I), Y \rangle \text{ is minimized.} \end{aligned}$$

This lets us transform $\hat{D}$ into $\delta(P_V \hat{D} P_V + \alpha I)$ without changing the optimal solutions. Numerical experiments show that once we scale $\hat{D}$ by some $\delta > 0$, the convergence becomes faster for the aforementioned input data distributions, e.g., providing a normalization for the objective function matrix.

8.4.4 Numerical tests

We now illustrate the efficiency of our algorithm on medium and large scale randomly generated problems. We observe that our **sADMM** approach finds the *exact* solution with relative duality gap < 1e-13, i.e., in machine precision, in almost all of the instances we tried. Sometimes, our algorithm gets stuck at the relative duality gap ≈ 1e-13 and runs until it reaches the max iteration. These indeterminate cases appear as outliers in the numerical tests. Further discussion about the nonzero duality gap follows in Section 8.5.

We use MATLAB version 2025a on (fastlinux in caption): Dell PowerEdge, Two Intel Xeon Gold 6244 8-core 3.6 GHz (Cascade Lake), 192 GB for the tests in Section 8.4.4 and Section 8.4.4. In Section 8.4.4, we use MATLAB version 2022a on two linux servers: (i) fastlinux: greyling22 Dell R840 4 Intel Xeon Gold 6254, with 3.10 GHz, 72 core and 384 GB for Table 8.1; and (ii) (biglinux in caption): Dell PowerEdge R6625, two AMD EPYC 9754 128-core 2.25 GHz, 1.5 TB for Table 8.2.

NP-hardness of WBP

To illustrate the difficulty in solving the original **NP**-hard problem, we solve **BCQP** in (8.8) using the commercial GUROBI solver, though total enumeration could be more competitive at times. We set $n = (2:1:8)$, $k = (2:1:8)$ for the tests, i.e., each of the k sets has the same n number of points. And we take the average of three tests for each problem size. For each test, a random **EDM** $D \in \mathbb{S}_+^{nk}$ is generated with embedding dimension $d = 2$. We can clearly see exponential growth

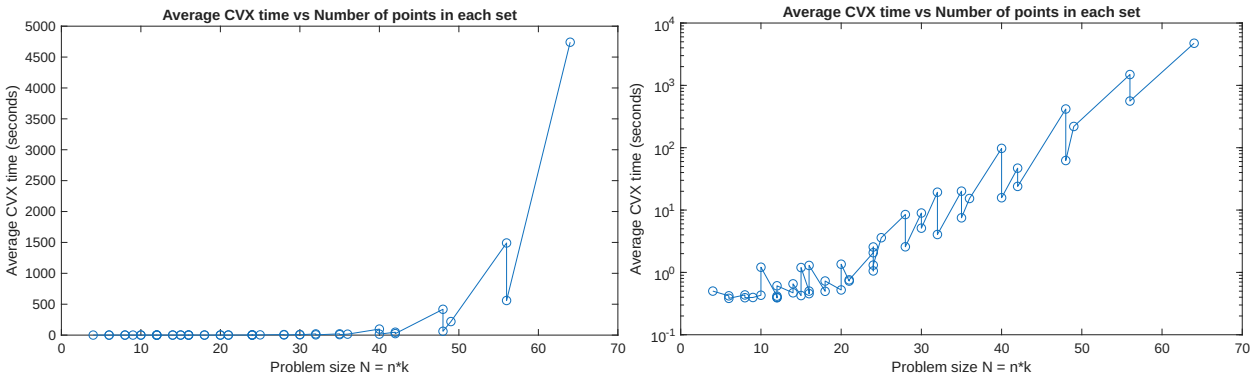

Figure 8.2: GUROBI: size $N = kn$ versus cpu time; illustrating exponential time

in time from Figure 8.2. Note that the CVX time is related to the number of feasible points, n^k. Hence, even if we have the same problem size, e.g., $(n, k) = (7, 8), (8, 7)$, the number of vertices, e.g., $n^k = 7^8, 8^7$, can be significantly different. Thus it appears in the figures that we have two values for the same problem size.

In contrast, we see the slow (linear) growth for **sADMM**, other than outliers, for the computation time versus the size $N = \sum_i n_i$, where n_i are the varying set sizes chosen randomly in the interval of width 5 about the given expected value n. See Figure 8.3, page 173. The figure on the right is log-log scale which reduces the effect of outliers, so we can clearly see the linear relation. (Note that an outlier is generally a result of one of five instances having a positive duality gap.) This was with $d = (3:3:6), k = (10:1:20), n = (20:1:30)$, and each set has the same size. We run five problems for each data instance and take the average time.

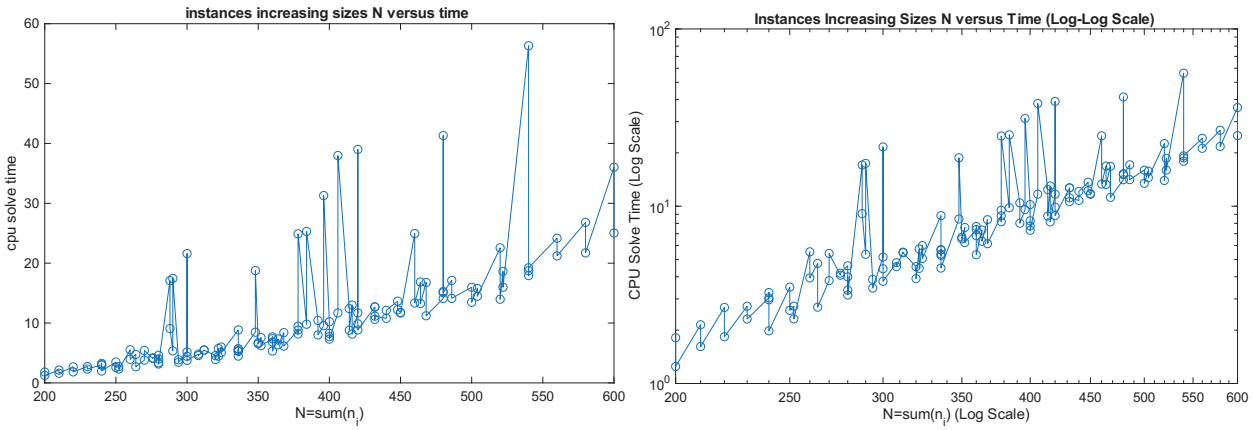

Figure 8.3: **sADMM**: size $N = kn$ versus cpu time; illustrating linear time

Success of sADMM approach

The results below detail the efficiency and surprising success of our algorithm in finding the exact solution of the original NP-hard problem.

In Table 8.1, page 174, we see a comparison between using the **sADMM** approach and CVX with the Mosek optimization solver.[6] We can see the times for the CVXsolver increase dramatically. The relative duality gap from CVX is *not* a provable gap as the lower bound is obtained using the dual optimal value minus the posted accuracy of the solve from CVX; we do not have accurate primal feasibility or dual feasibility from CVX. Thus the relative gap is essentially the posted accuracy from CVX.[7] We do find a nearest feasible point to find the upper bound. In summary, we see that the dramatic difference in time and the improved accuracy with the guaranteed lower bound that verifies optimality.

We include large problems in Table 8.2, page 175. Each set has a constant number of elements n. Other than outliers, the times are very reasonable.

Hidden embedding dimension

The embedding dimension d is *hidden* in our model as we only use the distances between the N points. This is why we do not see d in Figure 8.3. The size of the **DNN** model is $N+1$ irrespective of d. The *hardness* of the problem is often hiding in the rank of the *optimal solution* Y of the **DNN** relaxation, i.e., if the rank is one, then we have solved the original NP-hard problem. However, if the rank of the optimal Y is large, then the heuristics for the upper and lower bounds may not be enough to find an optimal solution for the original NP-hard Problem 8.2.1. However, the rank of the objective matrix D that comes from d did not influence this at all. In the various tests that we have done we did not see any changes in the solution times or the ability to obtain a near zero duality gap for wide varying values of d.

Figure 8.4 in 174 shows cpu time for our **sADMM** method when we fix the problem size $N = nk$ and vary the embedding dimension d. Random problems are generated with $(n, k) \in$

[6] We used a Dell R840, 4 Intel Xeon Gold 6240, 2.6GHz, 72 core 384 GB PC, with MATLAB version 2025b.

[7] The negative relative gaps are due to round-off errors during the solver's internal iterations. We emphasize that these numbers are within machine precision, meaning the solutions are mathematically valid and optimal.

dim/sets/size			Time (s)		rel. duality gap	
d	k	N	ADMM	CVXsolver	ADMM	CVXsolver
2	10	120	0.33	11.55	2.6e-15	6.1e-14
2	10	140	2.14	23.84	1.7e-14	5.5e-14
2	10	160	0.70	21.85	8.9e-15	9.9e-14
2	10	180	0.94	42.89	9.8e-16	6.2e-14
2	11	132	0.78	17.80	2.3e-15	5.3e-14
2	11	154	0.73	36.39	1.3e-15	4.2e-14
2	11	176	0.93	40.11	1.0e-14	7.7e-14
2	11	198	1.15	87.65	1.1e-14	4.4e-14
2	12	144	0.64	29.10	2.2e-14	4.0e-14
2	12	168	0.76	58.98	-3.1e-15	4.4e-14
2	12	192	1.05	101.87	2.8e-14	6.5e-14
2	12	216	1.97	108.40	1.5e-14	3.9e-14
3	10	120	0.35	11.41	1.3e-15	5.9e-14
3	10	140	0.56	21.94	-1.2e-14	6.7e-14
3	10	160	0.58	25.57	-7.3e-15	4.1e-14
3	10	180	0.98	49.75	4.2e-15	6.4e-14
3	11	132	0.46	18.02	4.0e-15	4.7e-14
3	11	154	0.55	27.24	1.2e-14	4.9e-14
3	11	176	1.01	40.16	3.4e-14	5.0e-14
3	11	198	0.93	80.77	1.4e-14	4.8e-14
3	12	144	1.12	26.59	7.4e-15	4.8e-14
3	12	168	0.71	59.18	-5.1e-15	4.6e-14
3	12	192	1.05	84.04	1.7e-14	4.1e-14
3	12	216	1.27	99.45	3.1e-14	4.2e-14

Table 8.1: Comparing ADMM versus CVX with Mosek Optimization Solver

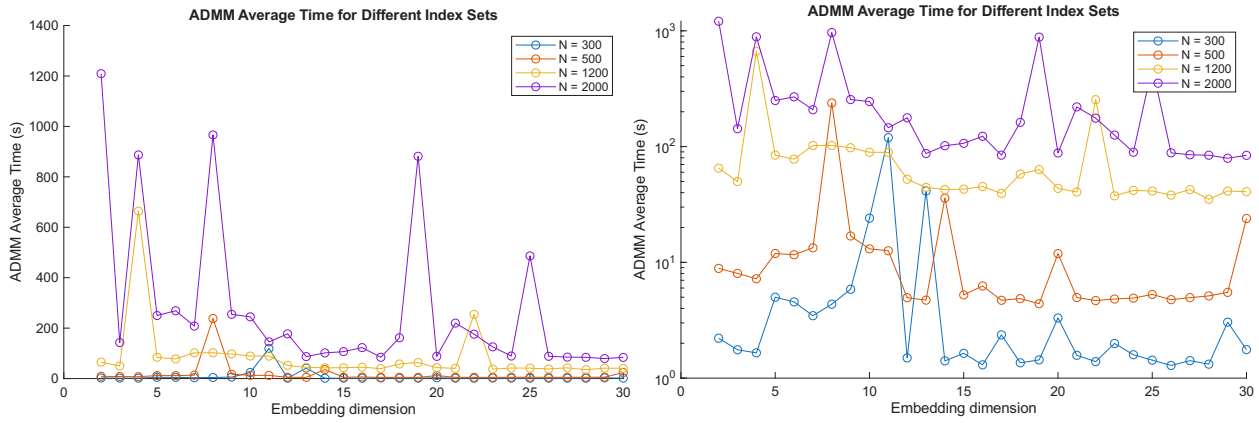

Figure 8.4: **sADMM**: Embedding dimension d versus cpu time in various sizes

$\{10, 40\} \times \{30, 50\}$, and $d \in \{2, \ldots, 30\}$. The cpu time is dependent only on problem size and not on the embedding dimension.

dim/sets/size			Time (s)	rel.duality gap
d	k	N	ADMM	ADMM
8	30	1200	104.81	-1.9e-15
8	30	1230	67.08	-3.2e-14
8	31	1240	94.90	-1.3e-14
8	31	1271	81.93	2.5e-14
8	32	1280	75.29	3.1e-14
8	32	1312	2025.42	1.8e-13
9	30	1200	4586.51	1.2e-13
9	30	1230	63.91	2.6e-14
9	31	1240	93.96	3.3e-14
9	31	1271	71.31	3.2e-14
9	32	1280	92.89	3.6e-14
9	32	1312	86.67	-2.2e-13

Table 8.2: Large problems with **sADMM** on biglinux server

8.5 Multiple optimal solutions and duality gaps

We now show that *multiple optimal* solutions for the original hard problem can lead to a duality gap between the optimal value of the original NP-hard problem and the lower bound found from the **DNN** relaxation.

8.5.1 Criteria for duality gaps

To find duality gaps for **SDP** relaxations, we want to find optimal points for the relaxation that are not vertices, i.e., they are outside of the polyhedral set formed from the convex hull of the lifted vertices. The key for this is having multiple optimal solutions for the original problem. The following Lemma 8.5.1 and Corollary 8.5.2 provides a construction for obtaining a positive gap between the optimal values of a *general* hard problem with multiple optimal solutions and its **DNN** relaxation.

Lemma 8.5.1. *Let $\{x_i\}_{i=1}^n \subset \mathbb{R}^n_+$ be a linearly independent set. Define the lifted vertices and barycenter, respectively,*

$$\left\{X_i = x_i x_i^T\right\}_{i=1}^n \subset \mathbb{S}^n, \quad \hat{X} := \frac{1}{n}\sum_{i=1}^n X_i.$$

Then $\hat{x} := \sum_i x_i > 0$, and

$$\hat{X} \in \mathbb{S}^n_{++}.$$

Proof. That $\hat{x} > 0$ follows by contradiction. Now note that $X_i \succeq 0, \forall i$, and so $\hat{X} \succeq 0$ as well. Now suppose that $0 = \hat{X}v$, for some $0 \neq v \in \mathbb{R}^n$. Then

$$0 = v^T\hat{X}v = v^T\sum_i X_i v \implies 0 = v^T X_i v, \forall i \implies (v^T x_i)^2 = 0, \forall i \implies v = 0,$$

by the linear independence assumption; thus contradicting $v \neq 0$ and yielding $\hat{X} > 0$. □

Corollary 8.5.2. *Suppose that the hypotheses of Lemma 8.5.1 hold. Let the points $x_i, i = 1, \ldots, n$ be given (multiple) optima for a given hard minimization problem, i.e.,*

$$(P) \quad p^* = \min \left\{ x^T Q x \,:\, x \in \{0,1\}^n \right\} = x_i^T Q x_i,\ i = 1, \ldots, n.$$

Moreover, suppose that there exists a feasible y that is not optimal for (P),

$$y \in \{0,1\}^n, y^T Q y > p^*.$$

*Then the **SDP** relaxation has feasible points $Y = yy^T, Z$ such that*

$$\operatorname{trace} YQ > p^* > \operatorname{trace} ZQ,$$

i.e., Z yields a duality gap.

Proof. From Lemma 8.5.1 we have that the barycenter $\hat{X}$ of the $\{x_i\}$ satisfies $\hat{X} \in \mathbb{S}^n_{++}$. Note that $\operatorname{trace} YQ = y^T Q y > p^* = \operatorname{trace} \hat{X} Q$. Therefore, $\operatorname{trace}(\hat{X} - Y)Q < 0$, and for $\epsilon > 0$,

$$\operatorname{trace}(\hat{X} + \epsilon(\hat{X} - Y))Q = p^* + \epsilon \operatorname{trace}(\hat{X} - Y)Q < p^*.$$

Moreover, the line segment $[Y, \hat{X} + \epsilon(\hat{X} - Y)]$ is feasible for the **SDP** relaxation for small enough $\epsilon > 0$ by $\hat{X} \in \mathbb{S}^n_{++}$. Therefore, we set $Z_\epsilon = \hat{X} + \epsilon(\hat{X} - Y), 0 < \epsilon << 1$ and obtain a duality gap for any such $Z = Z_\epsilon$. □

We can extend this theory to problems with general linear constraints $Ax = b$ by using **FR**. We now specifically extend it to our **BCQP** in (8.8). After **FR**, we need $nk+1-k$ linearly independent optimal points. This can be obtained when we choose $k >> n$. Recall the matrix K in (8.10) used for facial reduction and the facially reduced **DNN** relaxation in (8.16).

Corollary 8.5.3. *We consider the **BCQP** in (8.8) with optimal value p^*, and the **DNN** relaxation in (8.16). Let*

$$\left\{ y_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix} \right\}_{i=1}^{nk+1-k} \subset \mathbb{R}^{nk+1}_+$$

*be a linearly independent set that are optimal for **BCQP** and with $\sum_i y_i > 0$. Define the lifted vertices and barycenter, respectively,*

$$\left\{ Y_i = y_i y_i^T \right\}_i, \forall i, \quad \hat{Y} := \frac{1}{nk+1-k} \sum_{i=1}^{nk+1-k} Y_i.$$

*Moreover, suppose that there exists a feasible $\bar{x}$ for **BCQP** that is not optimal. Then*

$$\hat{Y} = V \hat{R} V^T \succeq 0,\ \hat{Y} > 0,\ \hat{R} > 0.$$

And there exists $Z = V R_Z V^T, R_Z > 0$ with optimal value $\operatorname{trace} DZ < p^*$, yielding a duality gap.

Proof. First note that incident vectors are feasible for the linear constraints and this guarantees that we have enough feasible points to guarantee that the diagonal entries and the off-diagonal blocks of $\hat{Y}$ are strictly positive. All lifted feasible points of the relaxation are in the minimal face

and have a corresponding matrix $R \in \mathbb{S}^{nk+1-k}$ for the facial reduction $Y = VRV^T$. Since $R > 0$ after the **FR**, we can apply the same proof as in Corollary 8.5.2. In addition, note that the linear constraints, the arrow constraint and gangster constraints, remain satisfied in the line formed from any two feasible points. □

8.5.2 Wheel of wheels examples with a duality gap

We illustrate the above theory with some specific problems with special structure that have multiple optimal solutions for the original **NP**-hard problem. We see that a duality gap can exist between the optimal value of the original **NP**-hard problem and the optimal value of the **DNN** relaxation.

Example 8.5.4 (Wheel of wheels)**.** *We next present another input data distribution for which the duality gap between the optimal value of the **BCQP** formulation and the Lagrangian dual value is non-trivial. The issue is again the non-uniqueness of the optimal solutions and the **sADMM** algorithm fails to break ties among them.*

The data distributions compose of a wheel of wheels, i.e., a wheel with an odd number of sets each of which is a wheel. Hence we call it an odd wheel. Given problem size parameters (k, n, d), define

- $\theta_k := \frac{2\pi}{k}$.
- *a set of k centroids encoded by a matrix $C \in \mathbb{R}^{k\times 2}$ such that*

$$C(i,:) = \begin{bmatrix}\cos(i\theta_k - \theta_k) & \sin(i\theta_k - \theta_k)\end{bmatrix}, i = 1, ..., k.$$

- *the radius of each cluster $r_k := \frac{\sqrt{\cos(\theta_k - 1)^2 + \sin\theta_k^2}}{4}$.*
- *the set of input points encoded by a matrix $P := (C \otimes e_k) + r_k(e_k \otimes C) \in \mathbb{R}^{k^2\times 2}$.*

When k is odd, there exists more than one optimal solution. A simple example with $k = 3 = n$ follows in Figure 8.5, page 178. We use the corresponding nine points in the configuration matrix ordered $1 - 9$ counter-clockwise in the triangles ordered counter-clockwise.

$$P = \begin{bmatrix} 1.7536 & 0.0137 \\ 0.6195 & 0.6362 \\ 0.6239 & -0.6643 \\ 0.2590 & 0.8609 \\ -0.8839 & 1.5449 \\ -0.8692 & 0.2201 \\ 0.2629 & -0.8740 \\ -0.8937 & -0.2100 \\ -0.8721 & -1.5275 \end{bmatrix}$$

The distances are ordered by choosing the points in lexicographic order:

$$(1,1), (1,2), \ldots (3,3).$$

The unique minimum distance is 11.1607 *obtained from the points* $2, 3, 2$ *and with the primal optimal* $x^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}^T$. *The optimal value from CVX to* 9 *decimals precision is* 10.8246, *thus verifying an empirical duality gap of* .3 *to* 9 *decimals precision. The motivation for this counter-example is to have near optimal solutions. One could shrink triangle two to make point* 4 *equidistant to points* $7, 8$ *and move point* 2 *closer to point* 3 *and thus have a tie optimal solution.*[8] *Note that the maximum distance is* 56.0227 *obtained from points* $(1, 2, 3)$.

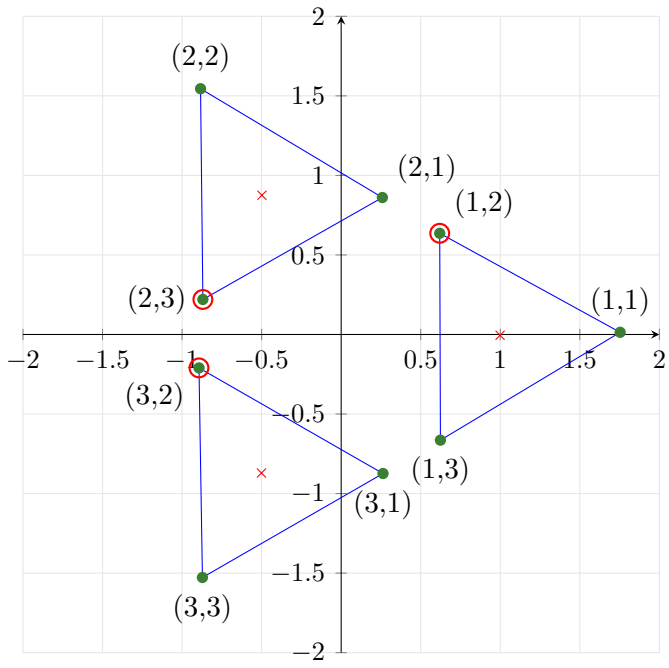

Figure 8.5: Duality gap for wheel of wheels: k=3=n

However, when k *is even, only one optimal solution clearly exists and the duality gap becomes trivial. An example with* $k = 6 = n$ *follows in Figure 8.6, page 179.*

[8] The wheel graph was used successfully to obtain duality gaps for the second lifting of the max-cut problem, see [13].

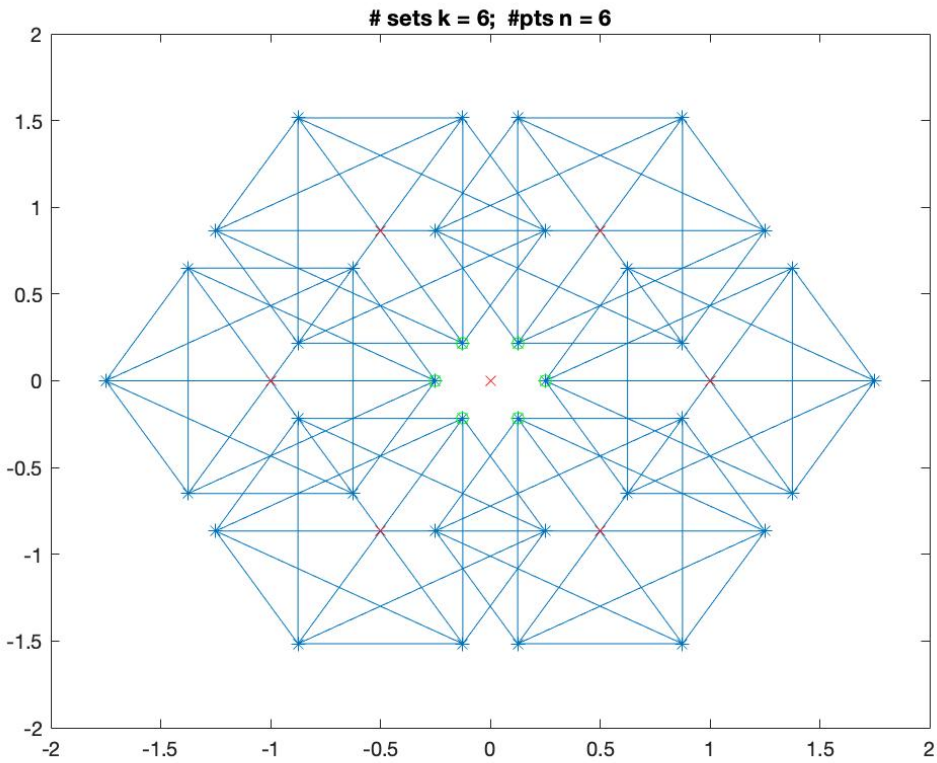

Figure 8.6: No duality gap for wheel of wheels: k=6=n

Chapter 9

Conclusions

In this thesis we studied various preprocessing strategies for challenging optimization problems where the difficulty arose from structural pathologies, e.g., matrix ill-conditioning, Hadamard ill-posedness, and failures of constraint qualifications.

Condition numbers

In Chapter 3, we studied ω, a nonclassical matrix condition number defined as the ratio of the arithmetic and geometric means of the eigenvalues of a positive definite matrix $M = A^T A$. We showed that ω offers several advantages over κ, the classical condition number defined as the ratio of the largest to smallest eigenvalues, as the latter primarily reflects a *worst-case* scenario.

Moreover, the inverse invariance $\kappa(M) = \kappa(M^{-1})$ (which does not hold for ω) can be misleading. In particular, the conditioning of a linear system satisfies $\text{cond}(A) \neq \text{cond}(A^\dagger)$, and it is typically the latter that should be reduced. Indeed, for linear systems $Ax = b$, we illustrated that it is $\omega_{\text{inv}} \cong \sqrt{\omega((A^T A)^{-1})}$ that should be minimized. At present, this observation remains primarily of theoretical interest, as effectively exploiting ω_{inv} without explicitly computing $(A^T A)^{-1}$ remains an open problem.

We exploited the differentiability and analytical simplicity of trace and determinant in $\omega(A)$ to derive optimal parameters for improving condition numbers via low-rank updates arising in nonsmooth Newton methods. Numerical experiments showed that the resulting ω-optimal preconditioners improve the performance of iterative methods.

Compared to the classical κ-condition number, ω is more closely correlated with practical performance measures such as iteration counts and computational time for iterative solvers applied to positive definite systems. This is consistent with established results showing that eigenvalue clustering is crucial for iterative methods, indicating that incorporating information from the full spectrum—rather than only the extremal eigenvalues—is beneficial. Our numerical results further demonstrated that $\omega(A)$ provides a significantly more accurate estimate of the true conditioning of a linear system, namely the sensitivity of the solution x to perturbations in the data A and b.

In future work, an interesting direction is to further exploit the measure $\sqrt{\omega((A^T A)^{-1})}$ by developing practical approximation techniques; see, e.g., [179]. While the denominator remains of comparable complexity, estimating the trace of an inverse presents a significant computational challenge. More broadly, condition numbers play a fundamental role in complexity analysis of

optimization methods, for instance in the convergence of conjugate gradient type algorithms. An important direction is to go beyond worst-case analysis and develop more representative average-case measures based on ω.

Diagonal preconditioners

In Chapter 4, we studied optimal diagonal preconditioning through the lens of these two condition numbers, κ and ω. On the theoretical side, we introduced an affine-based pseudoconvex reformulation of the κ-optimal preconditioning problem, yielding simple optimality conditions and enabling efficient optimization over an n-dimensional vector. Moreover, since the κ-condition number is positively homogeneous of degree zero, the associated minimization problem is Hadamard ill-posed. To address this, we introduced a reformulation that removes this homogeneity, leading to improved computational stability in practice. Building on this formulation, we developed a highly efficient subgradient method with convergence guarantees that significantly outperforms existing SDP-based approaches in both scalability and accuracy. For example, Table 4.2 shows that Algorithm 4.2 (resp. Algorithm 4.3) is, on average, 77.48 (resp. 356.22) times faster than [149] when minimizing κ on relatively small SuiteSparse matrices. Our methods also substantially outperform [71] in terms of runtime. In addition to these speedups, our subgradient methods consistently achieve significantly greater reductions in κ than both [149] and [71], often reducing it by more than half. Finally, our algorithms scale to large problem instances with hundreds of thousands of variables, solving them within minutes, whereas existing methods [71, 149] struggle to handle such instances within a reasonable time.

In parallel, we provided explicit and unified characterizations of ω-optimal diagonal and block-diagonal preconditioners, showing that many classical schemes such as Jacobi scaling, row/column normalization, and matrix balancing arise naturally as ω-optimal solutions or stationary points of the ω-optimal preconditioning problem. These results offer a new perspective on widely used preconditioning techniques and, to the best of our knowledge, constitute the first comprehensive comparison of κ- and ω-based optimality conditions.

Our numerical results further reveal a clear and practically important message: while κ-optimal preconditioners reduce the worst-case condition number more aggressively, ω-optimal preconditioners are substantially cheaper to compute and more strongly correlated with the performance of iterative methods such as PCG and LSQR. Moreover, applying the ω-optimal diagonal scaling to linear systems that are already κ-optimally preconditioned leads to further significant improvements in PCG's performance.

Overall, our findings suggest that ω-based preconditioning provides a computationally efficient and practically superior alternative to κ-based approaches for large-scale problems, and highlight the importance of moving beyond worst-case conditioning when designing preconditioners.

Local and Global Minimizers of the Smooth Stress Function

In Chapter 5, we addressed the nonconvex optimization problem arising from the exact recovery of points from a given **EDM**. Our investigation advanced the understanding of when the smooth stress function, as it is known in the MDS literature, admits local nonglobal minimizers in **EDM** problems. We established that, for this quartic objective function in $P \in \mathbb{R}^{n\times d}$, all second-order stationary points are global minimizers when $n \leqslant d+1$. For $n > d+1$, we identified **lngm**s through

numerical methods and used Kantorovich's theorem to provide rigorous analytical proofs of their existence. Moreover, based on the special patterns observed in these **lngm**s, we constructed the analytical Example 5.5.1. Our methodology relied on two reduction techniques based on translation and rotation invariance. These reductions were necessary for applying Kantorovich's theorem.

The findings of this research resolved a long-standing open question regarding the existence of **lngm**s in the context of MDS. These findings also highlight the importance of second-order methods for minimizing the smooth stress function.

For future work, one direction is to further characterize the properties of **lngm**s, for example with respect to embedding dimensions and ranks. Another direction is to study conditions for the existence of **lngm** when $\bar{D}$ has inexact or missing entries, as this setting is closer to the so-called **EDM** completion problem; see, e.g., [114, 119]. The present work serves as a first step in this direction, as we assumed that $\bar{D}$ is complete and is a true **EDM**.

Single Element Error Correction in an EDM

In Chapter 6, we studied a structured error-correction problem for **EDM**s, where the **EDM** D has known embedding dimension d and exactly one distance entry is corrupted. We presented three divide-and-conquer strategies and three different types of facial reduction. Our approaches accurately identified and corrected a single corrupted distance in an **EDM**.

The numerical tests confirmed that our methods solve very large instances quickly and to high precision. In particular, for random instances with $n = 100{,}000$, the best of the three methods took approximately 100 seconds to reach near machine precision. This performance confirmed our $O(n)$ complexity analysis for the best-performing algorithm among the three tested methods. For example, when $n = 100{,}000$ and $d = 3$, the point matrix P has 3×10^5 variables, while the associated dense Gram matrix has approximately 5×10^9 independent entries. Attempting to solve such problems using **SDP** formulations with interior-point or first-order methods would be impractical and, even if possible, would be unlikely to achieve comparable accuracy. In contrast, our methods achieved near machine precision.

We also provided a characterization of when a perturbation of a single distance entry results in an **EDM** whose embedding dimension remains unchanged. This is equivalent to maintaining the difficult constant-rank constraint. Moreover, we characterized when the **NEDM** problem solves our problem. This occurs if and only if the original data element satisfies $(D_0)_{ij} = 0$ and the perturbation satisfies $\epsilon < 0$, which is a highly degenerate trivial case since the corrupted location (i, j) is then identified. This observation emphasizes the difficulty of extending our exact recovery results to the case of multiple corrupted entries.

Although our main analysis focused on the case of a single corrupted distance, the algorithm can be successfully extended to multiple corrupted entries provided that they lie outside the selected blocks. The framework can also be adapted to chordal graphs by choosing overlapping cliques to form the principal submatrices.

Projection onto Spectrahedra

We presented and analyzed a semismooth Newton method for the best approximation problem, namely the projection problem onto spectrahedra. We showed that nondegeneracy is essential for

the semismooth Newton method to perform well. In the absence of a regularity condition, the dual optimal set becomes unbounded, which explains the poor numerical performance. Moreover, we showed that the failure of strict feasibility leads to degeneracy and ill-conditioning of the Jacobian at optimality. Our numerical results illustrated the importance of strict feasibility. In particular, we studied degeneracy phenomena for the elliptope and vontope.

Although we focused on **SDP**, many modern relaxations of hard problems involve the doubly nonnegative cone, **DNN**, defined by $\mathbf{DNN} = \mathbb{S}^n_+ \cap \mathbb{R}^{n\times n}_+$. In particular, splitting methods can efficiently exploit this intersection of two cones, and facial reduction often provides a natural and efficient splitting; see, e.g., [80,108,136]. The results obtained for the semismooth Newton method applied to the **BAP** suggest that similar ideas may extend to splitting methods for feasible sets of the form $\mathcal{L} \cap \mathbf{DNN}$.

Simplified Wasserstein Barycenter Problem

In Chapter 8, we presented a strategy for solving a class of **NP**-hard binary quadratic problems. Our approach formulated a **DNN** relaxation and applied **FR** to obtain a natural splitting for a symmetric alternating direction method of multipliers (**sADMM**), combined with an intermediate multiplier update and strong upper and lower bounding techniques. In particular, the structure of both the primal and dual solutions was exploited in the update steps of **sADMM**. We applied this framework to the **NP**-hard simplified Wasserstein barycenter problem.

Surprisingly, for the random instances we generated, the gap between the upper and lower bounds was zero to machine precision. Thus, we were able to provably solve the original **NP**-hard optimization problem, in the sense that any alternative optimal solution would attain the same optimal value to machine precision. This coincided with the rank-one property $\text{rank}(Y^*) = 1$ for the optimal solution obtained from the **DNN** relaxation. We observed that the embedding dimension d is hidden in the **DNN** relaxation. However, for specially constructed input data with nearly multiple optimal solutions, the algorithm had difficulty breaking ties. This resulted in nontrivial gaps between the lower and upper bounds and coincided with $\text{rank}(Y^*) > 1$, indicating that the original Wasserstein problem was not solved to optimality.

For future research, an important direction is to better understand the theoretical reasons for the positive duality gaps and to identify broader classes of instances where such gaps occur. One natural question is whether the absence of gaps corresponds to large normal cones at boundary points of the feasible set. Another direction is to study the effect of small perturbations on instances with duality gaps, and in particular whether such perturbations can close the gaps.

Finally, we are collecting data on airports in North America by state and province from ourairports.com/continents/NA/. We plan to apply our approach to the problem of finding the best hub in each state (or province) in order to identify a best hub location for the entire country.

References

[1] M. Agueh and G. Carlier, *Barycenters in the wasserstein space*, SIAM Journal on Mathematical Analysis, 43 (2011), pp. 904–924. 155

[2] S. Al-Homidan and H. Wolkowicz, *Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming*, Linear Algebra Appl., 406 (2005), pp. 109–141. 83, 108

[3] J. Alencar, C. Lavor, and L. Liberti, *Realizing Euclidean distance matrices by sphere intersection*, Discrete Appl. Math., 256 (2019), pp. 5–10. 78

[4] A. Alfakih, *On the uniqueness of Euclidean distance matrix completions: the case of points in general position*, Linear Algebra Appl., 397 (2005), pp. 265–277. 10

[5] ——, *Euclidean distance matrices and their applications in rigidity theory*, Springer, Cham, 2018. 9, 11, 16, 78, 105, 106, 107, 122, 123

[6] ——, *On yielding and jointly yielding entries of Euclidean distance matrices*, Linear Algebra Appl., 556 (2018), pp. 144–161. 10, 104, 123

[7] A. Alfakih and H. Wolkowicz, *Two theorems on Euclidean distance matrices and Gale transform*, Linear Algebra Appl., 340 (2002), pp. 149–154. 106

[8] A. Y. Alfakih, M. F. Anjos, V. Piccialli, and H. Wolkowicz, *Euclidean distance matrices, semidefinite programming and sensor network localization*, Port. Math., 68 (2011), pp. 53–102. 105

[9] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1999), pp. 13–30. Computational optimization—a tribute to Olvi Mangasarian, Part I. 16, 80

[10] F. Alizadeh, J.-P. Haeberly, and M. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Program., 77 (1997), pp. 111–128. 147

[11] J. Altschuler and E. Boix-Adserà, *Wasserstein barycenters are NP-hard to compute*, SIAM J. Math. Data Sci., 4 (2022), pp. 179–203. 155, 157

[12] J. M. Altschuler and E. Boix-Adsera, *Wasserstein barycenters can be computed in polynomial time in fixed dimension*, Journal of Machine Learning Research, 22 (2021), pp. 1–19. 155

[13] M. F. Anjos and H. Wolkowicz, *Strengthened semidefinite relaxations via a second lifting for the Max-Cut problem*, Discrete Appl. Math., 119 (2002), pp. 79–106. Foundations of heuristics in combinatorial optimization. 178

[14] A. Auslender and M. Teboulle, *Asymptotic cones and functions in optimization and variational inequalities*, Springer Monographs in Mathematics, Springer-Verlag, New York, 2003. 133

[15] A. D. Báez Sánchez and C. Lavor, *On the estimation of unknown distances for a class of Euclidean distance matrix completion problems with interval data*, Linear Algebra Appl., 592 (2020), pp. 287–305. 78

[16] M. Bakonyi and C. Johnson, *The Euclidean distance matrix completion problem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 646–654. 105, 115

[17] A. S. Bandeira, N. Boumal, and V. Voroninski, *On the low-rank approach for semidefinite programs arising in synchronization and community detection*, in 29th Annual Conference on Learning Theory, V. Feldman, A. Rakhlin, and O. Shamir, eds., vol. 49 of Proceedings of Machine Learning Research, Columbia University, New York, New York, USA, 23–26 Jun 2016, PMLR, pp. 361–382. 79

[18] W. Barrett, C. Johnson, and M. Lundquist, *Determinantal formulae for matrix completions associated with chordal graphs*, Linear Algebra Appl., 121 (1989), pp. 265–289. Linear algebra and applications (Valencia, 1987). 105

[19] A. Barvinok, *A remark on the rank of positive semidefinite matrices subject to affine constraints*, Discrete Comput. Geom., 25 (2001), pp. 23–31. 132

[20] H. Bauschke and V. Koch, *Projection methods: Swiss army knives for solving feasibility and best approximation problems with halfspaces*, in Infinite products of operators and their applications, vol. 636 of Contemp. Math., Amer. Math. Soc., Providence, RI, 2015, pp. 1–40. 132

[21] A. Ben-Israel and T. Greville, *Generalized Inverses: Theory and Applications*, Wiley-Interscience, 1974. 135

[22] M. Benzi, *Preconditioning techniques for large linear systems: a survey*, J. Comput. Phys., 182 (2002), pp. 418–477. 45

[23] L. Bergamaschi, *A survey of low-rank updates of preconditioners for sequences of symmetric linear systems*, Algorithms, 13 (2020). 20

[24] L. Bergamaschi, R. Bru, and A. Martínez, *Low-rank update of preconditioners for the inexact Newton method with SPD Jacobian*, Mathematical and Computer Modelling, 54 (2011), pp. 1863–1873. Mathematical models of addictive behaviour, medicine & engineering. 20

[25] A. Berman, *Cones, Matrices and Mathematical Programming*, Springer-Verlag, Berlin, New York, 1973. 108

[26] R. Bixby, *Solving real-world linear programs: a decade and more of progress*, Oper. Res., 50 (2002), pp. 3–15. 50th anniversary issue of Operations Research. 13

[27] R. Borsdorf and N. J. Higham, *A preconditioned Newton algorithm for the nearest correlation matrix*, IMA J. Numer. Anal., 30 (2010), pp. 94–107. 132

[28] J. Borwein and H. Wolkowicz, *Regularizing the abstract convex program*, J. Math. Anal. Appl., 83 (1981), pp. 495–530. 136

[29] ——, *A simple constraint qualification in infinite-dimensional programming*, Math. Programming, 35 (1986), pp. 83–96. 139

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learning, 3 (2011), pp. 1–122. 171

[31] S. Burer and R. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program., 95 (2003), pp. 329–357. Computational semidefinite and second order cone programming: the state of the art. 79

[32] F. Burkowski, H. Im, and H. Wolkowicz, *A Peaceman-Rachford splitting method for the protein side-chain positioning problem*, INFORMS J. Comput., 37 (2025), pp. 962–976. 151, 155, 156, 160, 163, 166, 167, 169, 170

[33] R. J. Caron and N. Gould, *Finding a positive semidefinite interval for a parametric matrix*, Linear Algebra Appl., 76 (1986), pp. 19–29. 123

[34] M. Carter, H. Jin, M. Saunders, and Y. Ye, *SpaseLoc: an adaptive subproblem algorithm for scalable wireless sensor network localization*, SIAM J. Optim., 17 (2006), pp. 1102–1128. 105

[35] Y. Censor, , W. Moursi, T. Weames, and H. Wolkowicz, *Regularized nonsmooth Newton algorithms for best approximation with applications*, tech. rep., University of Waterloo, Waterloo, Ontario, 2022 under revision. 37 pages, research report, under revision. 20, 29, 38, 132, 139, 149

[36] Y. Censor, *Weak and strong superiorization: between feasibility-seeking and minimization*, An. Ştiinţ. Univ. "Ovidius" Constanţa Ser. Mat., 23 (2015), pp. 41–54. 132

[37] A. Charnes, *Optimality and degeneracy in linear programming*, Econometrica, 20 (1952), pp. 160–170. 13

[38] K. Chen, *Matrix preconditioning techniques and applications*, vol. 19 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005. 43

[39] X. Chen, R. S. Womersley, and J. J. Ye, *Minimizing the condition number of a Gram matrix*, SIAM J. Optim., 21 (2011), pp. 127–148. 29

[40] Y. Chen and X. Ye, *Projection onto a simplex*, 2011. 168

[41] G. Ciaramella and M. J. Gander, *Iterative methods and preconditioners for systems of linear equations*, vol. 19 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, [2022] ©2022. 45

[42] S. Cipolla and J. Gondzio, *Proximal Stabilized Interior Point Methods and Low − Frequency − Update Preconditioning Techniques*, J. Optim. Theory Appl., 197 (2023), pp. 1061–1103. 29

[43] F. Clarke, *Optimization and nonsmooth analysis*, vol. 5 of Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 1990. 142

[44] V. Cohen-Addad, C. Fan, E. Lee, and A. de Mesmay, *Fitting metrics and ultrametrics with minimum disagreements*, SIAM Journal on Computing, 54 (2025), pp. 92–133. 105

[45] M. Cuturi and A. Doucet, *Fast computation of wasserstein barycenters*, in Proceedings of the 31st International Conference on Machine Learning, E. P. Xing and T. Jebara, eds., vol. 32 of Proceedings of Machine Learning Research, Beijing, China, 22–24 Jun 2014, PMLR, pp. 685–693. 155

[46] J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*, Meboo Publishing, USA, 2005. 107

[47] W. C. Davidon, *Optimally conditioned optimization algorithms without line searches*, Mathematical Programming, 9 (1975), pp. 1–30. 20

[48] T. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Transactions on Mathematical Software (TOMS), 38 (2011), pp. 1–25. 69

[49] M. de Carli Silva and L. Tunçel, *Vertices of spectrahedra arising from the elliptope, the theta body, and their relatives*, SIAM J. Optim., 25 (2015), pp. 295–316. 132

[50] J. Demmel, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math., 51 (1987), pp. 251–289. 25

[51] ——, *The probability that a numerical analysis problem is difficult*, Math. Comp., 50 (1988), pp. 449–480. 19

[52] J. E. Dennis, Jr. and H. Wolkowicz, *Sizing and least-change secant methods*, SIAM J. Numer. Anal., 30 (1993), pp. 1291–1314. 20, 21, 23, 29, 43, 59, 60

[53] J. Dennis Jr. and R. Schnabel, *Least change secant updates for quasi-Newton methods*, SIAM Review, 21 (1979), pp. 443–459. 30

[54] J. Dennis Jr. and R. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, vol. 16 of Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Corrected reprint of the 1983 original. 95, 99

[55] L. Ding and M. Udell, *A strict complementarity approach to error bound and sensitivity of solution of conic programs*, Optim. Lett., 17 (2023), pp. 1551–1574. 131

[56] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, *Sensor network localization, Euclidean distance matrix completions, and graph realization*, Optim. Eng., 11 (2010), pp. 45–66. 105

[57] X. Doan and H. Wolkowicz, *Numerical computations and the ω-condition number*, Tech. Rep. CORR 2011-03, University of Waterloo, Waterloo, Ontario, 2011. www.math.uwaterloo.ca/~hwolkowi/henry/reports/ONE.pdf. 59, 65, 66

[58] X. V. Doan, S. Kruk, and H. Wolkowicz, *A robust algorithm for semidefinite programming*, Optim. Methods Softw., 27 (2012), pp. 667–693. 66

[59] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, *Euclidean distance matrices: Essential theory, algorithms, and applications*, IEEE Signal Processing Magazine, 32 (2015), p. 12–30. 105

[60] E. Dolan and J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213. 39

[61] D. Drusvyatskiy, N. Krislock, Y.-L. C. Voronin, and H. Wolkowicz, *Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier*, SIAM J. Optim., 27 (2017), pp. 2301–2331. 105, 106, 111, 112, 116

[62] D. Drusvyatskiy, G. Li, and H. Wolkowicz, *A note on alternating projections for ill-posed semidefinite feasibility problems*, Math. Program., 162 (2017), pp. 537–548. 131, 132

[63] D. Drusvyatskiy and H. Wolkowicz, *The many faces of degeneracy in conic optimization*, Foundations and Trends® in Optimization, 3 (2017), pp. 77–170. 15, 16, 131, 162

[64] M. Dür, B. Jargalsaikhan, and G. Still, *Genericity results in linear conic programming—a tour d'horizon*, Math. Oper. Res., 42 (2017), pp. 77–94. 131

[65] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrica, 1 (1936), pp. 211–218. 108, 139

[66] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, vol. 2, Springer, 2003. 139, 140

[67] A. Franceschini, M. Ferronato, C. Janna, V. A. Paludetto Magri, et al., *Recent advancements in preconditioning techniques for large size linear systems suited for high performance computing*, Dolomites Research Notes on Approximation, 11 (2018), pp. 11–22. 43

[68] S. Friedland, *Convex spectral functions*, Linear and Multilinear Algebra, 9 (1980/81), pp. 299–316. 2, 8

[69] D. Gale, *Neighboring vertices on a convex polyhedron*, in Linear inequalities and related system, vol. 38 of Annals of Mathematics Studies, Princeton University Press, Princeton, NJ, 1956, pp. 255–263. 106, 115

[70] W. Gao, Y.-C. Chu, Y. Ye, and M. Udell, *Gradient methods with online scaling*, tech. rep., 2024. arXiv, 2411.01803. 43, 46

[71] W. Gao, Z. Qu, M. Udell, and Y. Ye, *Scalable approximate optimal diagonal preconditioning*, tech. rep., 2023. arXiv, 2312.15594. 24, 43, 44, 45, 46, 69, 70, 181

[72] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Programming, 12 (1977), pp. 136–138. 142, 145

[73] D. Ge, H. Wang, Z. Xiong, and Y. Ye, *Interior-point methods strike back: Solving the Wasserstein barycenter problem*, in 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, vol. 33, 2019, pp. 1–12. 155, 159

[74] J. A. George and J. W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981. 43

[75] M. Goemans and D. Williamson, *.878-approximation algorithms for MAX CUT and MAX 2SAT*, in ACM Symposium on Theory of Computing (STOC), 1994. 156

[76] G. Golub and C. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, fourth ed., 2013. 43

[77] J. Gondzio, S. Pougkakiotis, and J. Pearson, *General-purpose preconditioning for regularized interior point methods*, Comput. Optim. Appl., 83 (2022), pp. 727–757. 29

[78] P. Goulart, Y. Nakatsukasa, and N. Rontsis, *Accuracy of approximate projection to the semidefinite cone*, Linear Algebra and its Applications, 594 (2020), pp. 177–192. 132

[79] N. Gould and J. Scott, *The state-of-the-art of preconditioners for sparse linear least-square problems*, ACM Trans. Math. Software, 43 (2017), pp. Art. 36, 35. 43

[80] N. Graham, H. Hu, H. Im, X. Li, and H. Wolkowicz, *A restricted dual Peaceman-Rachford splitting method for a strengthened DNN relaxation for QAP*, INFORMS J. Comput., 34 (2022), pp. 2125–2143. 148, 183

[81] ——, *A restricted dual Peaceman-Rachford splitting method for a strengthened DNN relaxation for QAP*, INFORMS J. Comput., 34 (2022), pp. 2125–2143. 169

[82] A. Greenbaum, *Iterative methods for solving linear systems*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. 27

[83] C. Greif and J. M. Varah, *Minimizing the condition number for small rank modifications*, SIAM J. Matrix Anal. Appl., 29 (2006/07), pp. 82–97. 29

[84] R. Grimes and J. Lewis, *Condition number estimation for sparse matrices*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 384–388. 24

[85] B. Grünbaum, *Convex polytopes*, vol. 221 of Graduate Texts in Mathematics, Springer-Verlag, New York, second ed., 2003. Prepared and with a preface by Volker Kaibel, Victor Klee and Günter M. Ziegler. 115

[86] S. Guo, H.-D. Qi, and L. Zhang, *Perturbation analysis of the euclidean distance matrix optimization problem and its numerical implications*, Comput. Optim. Appl., 86 (2023), pp. 1193–1227. 105

[87] W. W. Hager, *Condition estimates*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 311–316. 19, 20, 24

[88] D. Han and X. Yuan, *Local linear convergence of the alternating direction method of multipliers for quadratic programs*, SIAM J. Numer. Anal., 51 (2013), pp. 3446–3457. 169

[89] B. He, H. Liu, Z. Wang, and X. Yuan, *A strictly contractive Peaceman-Rachford splitting method for convex programming*, SIAM J. Optim., 24 (2014), pp. 1011–1040. 169

[90] D. Henrion and J. Malick, *Projection methods for conic feasibility problems: applications to polynomial sum-of-squares decompositions*, Optimization Methods and Software, 26 (2011), pp. 23–46. 132

[91] D. Higham, *Condition numbers and their condition numbers*, Linear Algebra Appl., 214 (1995), pp. 193–213. 19, 25

[92] N. Higham, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29 (1987), pp. 575–596. 24

[93] N. Higham and F. Tisseur, *A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1185–1201 (electronic). 24

[94] N. J. Higham, *Computing the nearest correlation matrix—a problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343. 132

[95] N. J. Higham and N. Strabić, *Bounds for the distance to the nearest correlation matrix*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1088–1102. 132

[96] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*, Grundlehren Text Editions, Springer-Verlag, Berlin, 2001. Abridged version of ıt Convex analysis and minimization algorithms. I [Springer, Berlin, 1993; MR1261420 (95m:90001)] and ıt II [ibid.; MR1295240 (95m:90002)]. 136

[97] H. Hu, H. Im, J. Lin, N. Lütkenhaus, and H. Wolkowicz, *QKD key rate with inequalities and nuclear norm*, tech. rep., University of Waterloo, Waterloo, Ontario, 2023. 25 pages, research report. 170

[98] H. Hu, X. Li, H. Im, and H. Wolkowicz, *A semismooth Newton-type method for the nearest doubly stochastic matrix problem*, Math. Oper. Res., 49 (2024), pp. 729–751. 29

[99] Y. Hu, J. Li, and C. K. W. Yu, *Convergence rates of subgradient methods for quasi-convex optimization problems*, Comput. Optim. Appl., 77 (2020), pp. 183–212. 56, 57

[100] H. Im, *Implicit Loss of Surjectivity and Facial Reduction: Theory and Applications*, PhD thesis, University of Waterloo, 2023. 13, 131

[101] ——, *Implicit redundancy and degeneracy in conic program*, tech. rep., arXiv, 2403.04171, 2024. arXiv, 2403.04171. 132

[102] H. Im and H. Wolkowicz, *A strengthened Barvinok-Pataki bound on SDP rank*, Oper. Res. Lett., 49 (2021), pp. 837–841. 132

[103] H. Im and H. Wolkowicz, *Revisiting degeneracy, strict feasibility, stability, in linear programming*, European J. Oper. Res., 310 (2023), pp. 495–510. 111, 112, 131, 132

[104] C. G. J. Jacobi, *Ueber eine neue auflösungsart der bei der methode der kleinsten quadrate vorkommenden lineären gleichungen*, Astronomische Nachrichten, 22 (1845), p. 297–306. 59

[105] C. Johnson, *Matrix completion problems: A survey*, in Matrix Theory and Applications 40, Proc. Sympos. Appl. Math., C. R. Johnson, ed., AMS, Providence, RI, 1990, pp. 171–198. 105

[106] C. Johnson and P. Tarazaga, *Connections between the real positive semidefinite and distance matrix completion problems*, Linear Algebra Appl., 223/224 (1995), pp. 375–391. Special issue honoring Miroslav Fiedler and Vlastimil Pták. 105, 115

[107] W. L. Jung, D. Torregrosa-Belén, and H. Wolkowicz, *The ω-condition number: applications to preconditioning and low rank generalized Jacobian updating*, Comput. Optim. Appl., 91 (2025), pp. 235–282. 59, 60, 65

[108] W. L. Jung and H. Wolkowicz, *Exact solutions for the NP-hard Wasserstein barycenter problem using a doubly nonnegative relaxation and a splitting method*, Springer Optimization and its Applications, Springer, Waterloo, Ontario, 2024. 21 pages, to appear. 183

[109] K. Kiwiel, *Convergence and efficiency of subgradient methods for quasiconvex minimization*, Mathematical programming, 90 (2001), pp. 1–25. 8, 9, 51, 52, 53, 56

[110] P. A. Knight, *The Sinkhorn–Knopp algorithm: convergence and applications*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 261–275. 59, 63, 65

[111] P. A. Knight, D. Ruiz, and B. Uçar, *A symmetry preserving algorithm for matrix scaling*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 931–955. 63

[112] N. Krislock, V. Piccialli, and H. Wolkowicz, *Robust semidefinite programming approaches for sensor network localization with anchors*, Tech. Rep. CORR 2006-12, University of Waterloo, Waterloo, Ontario, 2006. URL:orion.uwaterloo.ca/~hwolkowi/henry/reports/ABSTRACTS.html#sensorKPW. 105

[113] N. Krislock and H. Wolkowicz, *Explicit sensor network localization using semidefinite representations and facial reductions*, SIAM J. Optim., 20 (2010), pp. 2679–2708. 16, 17, 105, 111, 115

[114] ——, *Euclidean distance matrices and applications*, in Handbook on semidefinite, conic and polynomial optimization, vol. 166 of Internat. Ser. Oper. Res. Management Sci., Springer, New York, 2012, pp. 879–914. 78, 107, 182

[115] M. Laurent, *A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems*, in Topics in Semidefinite and Interior-Point Methods, The Fields Institute for Research in Mathematical Sciences Communications Services, vol 18, AMS, Providence, RI, 1998, pp. 51–76. 105

[116] J. Lee, *Smooth manifolds*, in Introduction to smooth manifolds, Springer, 2003, pp. 1–29. 57

[117] A. Lewis, *Convex analysis on the Hermitian matrices*, SIAM J. Optim., 6 (1996), pp. 164–177. 2, 8

[118] X. Li, T. Pong, H. Sun, and H. Wolkowicz, *A strictly contractive Peaceman-Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem*, Comput. Optim. Appl., 78 (2021), pp. 853–891. 156, 167, 169

[119] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, *Euclidean distance geometry and applications*, SIAM Rev., 56 (2014), pp. 3–69. 78, 182

[120] ——, *Euclidean distance geometry and applications*, SIAM Review, 56 (2014), pp. 3–69. 105

[121] M. Liu and G. Pataki, *Exact duals and short certificates of infeasibility and weak infeasibility in conic linear programming*, Mathematical Programming, 167 (2018), pp. 435–480. 13

[122] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 272–284 (electronic). 132, 139

[123] J. Malick and H. S. Sendov, *Clarke generalized Jacobian of the projection onto the cone of positive semidefinite matrices*, Set-Valued Anal., 14 (2006), pp. 273–293. 2, 8, 132, 140, 141

[124] S. Malone, P. Tarazaga, and M. Trosset, *Better initial configurations for metric multidimensional scaling*, Comput. Statist. Data Anal., 41 (2002), pp. 143–156. 79

[125] S. Malone and M. Trosset, *A study of the critical points of the SSTRESS criterion in metric multidimensional scaling*, Tech. Rep. 00-06, Department of Computational & Applied Mathematics, Rice University, Houston, TX, USA, 2000. 79

[126] O. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, NY, 1969. 3, 36

[127] R. Mansour, *New Algorithmic Structures for Feasibility-Seeking and for Best Approximation Problems and their Convergence Analyses*, ProQuest LLC, Ann Arbor, MI, 2023. Thesis (Ph.D.)–University of Haifa (Israel). 132

[128] R. Mathar and M. Trosset, *On the existence of nonglobal minimizers of the stress criterion for metric multidimensional scaling*, in Proceedings of the Statistical Computing Section, 1997, pp. 158–162. 79

[129] N. Megiddo, *A note on degeneracy in linear programming*, Math. Programming, 35 (1986), pp. 365–367. 13

[130] C. Micchelli, P. Smith, J. Swetits, and J. Ward, *Constrained l_p approximation*, Journal of Constructive Approximation, 1 (1985), pp. 93–102. 139

[131] J. Moreau, *Décomposition orthogonale d'un espace hilbertien selon deux cones mutuellement polaires*, C.R. Acad.Sci Paris, 255 (1962), pp. 238–240. 108

[132] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., *Distance geometry*, Springer, New York, 2013. Theory, methods, and applications. 78

[133] S. Natva and S. Nannuru, *Denoising and completion of Euclidean distance matrix from multiple observations*, IEEE, 2024 National Conference on Communications (NCC), (2024), pp. pp. 1–6. 105

[134] V.-B. Nguyen and T. Ngan Nguyen, *Positive semidefinite interval of matrix pencil and its applications to the generalized trust region subproblems*, Linear Algebra and its Applications, 680 (2024), pp. 371–390. 123

[135] H. Ochiai, Y. Sekiguchi, and H. Waki, *Analytic formulas for alternating projection sequences for the positive semidefinite cone and an application to convergence analysis*, J. Math. Anal. Appl., 544 (2025), pp. Paper No. 129070, 30. 132

[136] D. Oliveira, H. Wolkowicz, and Y. Xu, *ADMM for the SDP relaxation of the QAP*, Math. Program. Comput., 10 (2018), pp. 631–658. 156, 170, 183

[137] S. S. Oren and D. G. Luenberger, *Self-scaling variable metric (ssvm) algorithms: Part i: Criteria and sufficient conditions for scaling a class of algorithms*, Management Science, 20 (1974), pp. 845–862. 20

[138] V. Panaretos and Y. Zemel, *Statistical aspects of wasserstein distances*, Annual Review of Statistics and Its Application, 6 (2019), pp. 405–431. 155

[139] J.-S. Pang, D. Sun, and J. Sun, *Semismooth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems*, Math. Oper. Res., 28 (2003), pp. 39–63. 140

[140] P. Pardalos and S. Vavasis, *Quadratic programming with one negative eigenvalue is NP-hard*, J. Global Optim., 1 (1991), pp. 15–22. 159

[141] R. Parhizkar, *Euclidean Distance Matrices Properties, Algorithms and Applications*, PhD thesis, ISC, Lausanne, 2013. 79

[142] N. Parikh and S. Boyd, *Proximal algorithms*, Foundations and Trends® in Optimization, 1 (2013), pp. 123–231. 2, 7

[143] G. Pataki, *On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues*, Math. Oper. Res., 23 (1998), pp. 339–358. 132

[144] G. Pataki, *Geometry of Semidefinite Programming*, in Handbook OF Semidefinite Programming: Theory, Algorithms, and Applications, H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., Kluwer Academic Publishers, Boston, MA, 2000. 13, 108, 148, 149

[145] L. Pattersson, S. Sremac, F. Wang, and H. Wolkowicz, *Noisy Euclidean distance matrix completion with a single missing node*, J. Global Optim., 75 (2019), pp. 973–1002. 105

[146] J. Pearson and J. Pestana, *Preconditioners for Krylov subspace methods: an overview*, GAMM-Mitt., 43 (2020), pp. e202000015, 35. 19, 20

[147] G. Pini and G. Gambolati, *Is a simple diagonal scaling the best preconditioner for conjugate gradients on supercomputers?*, Advances in Water Resources, 13 (1990), pp. 147–153. 43

[148] H. Qi and D. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM journal on matrix analysis and applications, 28 (2006), pp. 360–385. 29, 148

[149] Z. Qu, W. Gao, O. Hinder, Y. Ye, and Z. Zhou, *Optimal diagonal preconditioning*, Operations Research, 73 (2024), pp. 1479–1495. ix, xiv, 43, 44, 45, 46, 69, 70, 71, 72, 73, 74, 181

[150] M. V. Ramana, L. Tunçel, and H. Wolkowicz, *Strong duality for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 641–662. 143

[151] R. Rockafellar, *Convex analysis*, Princeton Mathematical Series, No. 28, Princeton University Press, Princeton, N.J., 1970. 167

[152] Y. Saad, *Iterative methods for linear systems of equations: a brief historical journey*, 754 ([2020] ©2020), pp. 197–215. 59

[153] J. B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard*, in Proceedings of the 17th Allerton Conference in Communications, Control and Computing, 1979, pp. 480–489. 78

[154] M. Scetbon, C. Ma, W. Gong, and E. Meeds, *Gradient multi-normalization for stateless and scalable LLM training*, arXiv preprint arXiv:2502.06742, (2025). 59, 63

[155] K. Schaecke, *Essay on: The Kronecker product*, Master's thesis, University of Waterloo, 2004. 162

[156] R. Schnabel, *Analysing and improving quasi-Newton methods for unconstrained optimization*, PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1977. Also available as TR-77-320. 30

[157] I. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert"*, Ann. Math., 36 (1935), pp. 724–732. 10, 80

[158] A. Singer and Y. Shkolnisky, *Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming*, SIAM J. Imaging Sci., 4 (2011), pp. 543–572. 105

[159] M. Song, D. S. Gonçalves, W. L. Jung, C. Lavor, A. Mucherino, and H. Wolkowicz, *On the local and global minimizers of the smooth stress function in Euclidean distance matrix problems*, Linear Algebra Appl., 727 (2025), pp. 234–267. 105

[160] G. W. Soules, *The rate of convergence of Sinkhorn balancing*, in Proceedings of the First Conference of the International Linear Algebra Society (Provo, UT, 1989), vol. 150, 1991, pp. 3–40. 59, 63

[161] S. Sremac, *Error bounds and singularity degree in semidefinite programming*, PhD thesis, University of Waterloo, 2019. 13

[162] S. Sremac, H. Woerdeman, and H. Wolkowicz, *Error bounds and singularity degree in semidefinite programming*, SIAM J. Optim., 31 (2021), pp. 812–836. 13, 131

[163] G. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, NY, 1973. 19

[164] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, vol. 1993, Springer, 1980. 19

[165] G. Strang, *Linear Algebra and Its Applications 4th ed.*, 2012. 19

[166] J. Sturm, *Error bounds for linear matrix inequalities*, SIAM J. Optim., 10 (2000), pp. 1228–1248 (electronic). 13, 131

[167] D. Sun, *The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications*, Math. Oper. Res., 31 (2006), pp. 761–776. 140

[168] D. Sun and J. Sun, *Semismooth matrix-valued functions*, Mathematics of Operations Research, 27 (2002), pp. 150–169. 140

[169] K. Toh, M. Todd, and R. Tütüncü, *SDPT3—a MATLAB software package for semidefinite programming, version 1.3*, Optim. Methods Softw., 11/12 (1999), pp. 545–581. Interior point methods. 152

[170] L. Tunçel and H. Wolkowicz, *Strengthened existence and uniqueness conditions for search directions in semidefinite programming*, Linear Algebra Appl., 400 (2005), pp. 31–60. 46

[171] ——, *Strong duality and minimal representations for cone optimization*, Comput. Optim. Appl., 53 (2012), pp. 619–648. 132

[172] N. van der Aa, H. ter Morsche, and R. Mattheij, *Computation of eigenvalue and eigenvector derivatives for a general complex-valued eigensystem*, Electron. J. Linear Algebra, 16 (2007), pp. 300–314. 48

[173] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969/1970), pp. 14–23. 43

[174] A. Žilinskas and A. Podlipskytundefined, *On multimodality of the sstress criterion for metric multidimensional scaling*, Informatica, 14 (2003), p. 121–130. 79

[175] I. Waldspurger and A. Waters, *Rank optimality for the burer–monteiro factorization*, SIAM Journal on Optimization, 30 (2020), pp. 2577–2602. 79

[176] A. Wathen, *Preconditioning*, Acta Numer., 24 (2015), pp. 329–376. 45

[177] H. Wolkowicz and Q. Zhao, *Semidefinite programming relaxations for the graph partitioning problem*, Discrete Appl. Math., 96/97 (1999), pp. 461–479. The satisfiability problem (Certosa di Pontignano, 1996); Boolean functions. 156, 162

[178] R. Wong and T. Lee, *Matrix completion with noisy entries and outliers*, J. Mach. Learn. Res., 18 (2017), pp. 1–25. 105

[179] L. Wu, J. Laeuchli, V. Kalantzis, A. Stathopoulos, and E. Gallopoulos, *Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse*, Journal of Computational Physics, 326 (2016), pp. 828–844. 180

[180] J. Ye, P. Wu, J. Z. Wang, and J. Li, *Fast discrete distribution clustering using Wasserstein barycenter with sparse support*, IEEE Transactions on Signal Processing, 65 (2017), pp. 2317–2332. 155

[181] Y.-L. Yu, *The proximity operator*, in Semantic Scholar, 2014. 2

[182] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite programming relaxations for the quadratic assignment problem*, J. Comb. Optim., 2 (1998), pp. 71–109. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Toronto, ON, 1996). 148, 156, 162

Index