Numerical Stability in Linear Programming and Semidefinite Programming

by

Hua Wei

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

Waterloo, Ontario, Canada, 2006

I hereby declare that I am the sole author of this 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.

Abstract

We study numerical stability for interior-point methods applied to Linear Programming, LP, and Semidefinite Programming, SDP. We analyze the difficulties inherent in current methods and present robust algorithms.

We start with the error bound analysis of the search directions for the normal equation approach for LP. Our error analysis explains the surprising fact that the ill-conditioning is not a significant problem for the normal equation system. We also explain why most of the popular LP solvers have a default stop tolerance of only 10^{-8} when the machine precision on a 32-bit computer is approximately 10^{-16}.

We then propose a simple alternative approach for the normal equation based interior-point method. This approach has better numerical stability than the normal equation based method. Although, our approach is not competitive in terms of CPU time for the NETLIB problem set, we do obtain higher accuracy. In addition, we obtain significantly smaller CPU times compared to the normal equation based direct solver, when we solve well-conditioned, huge, and sparse problems by using our iterative based linear solver. Additional techniques discussed are: crossover; purification step; and no backtracking.

Finally, we present an algorithm to construct SDP problem instances with prescribed strict complementarity gaps. We then introduce two *measures of strict complementarity gaps*. We empirically show that: (i) these measures can be evaluated accurately; (ii) the size of the strict complementarity gaps correlate well with the number of iteration for the SDPT3 solver, as well as with the local asymptotic convergence rate; and (iii) large strict complementarity gaps, coupled with the failure of Slater's condition, correlate well with loss of accuracy in the solutions. In addition, the numerical tests show that there is *no* correlation between the strict complementarity gaps and the geometrical measure used in [31], or with Renegar's condition number.

Acknowledgments

I would like to express my deep thanks to my supervisor, Professor Henry Wolkowicz. Without his continues guidance and support, I could not finish this thesis. I would also like to thank the committee members, Professor Miguel Anjos, Professor Chek Beng Chua, Professor Levent Tunçel, and Professor Yin Zhang, for their detailed comments and careful reading of the draft.

Thanks to the professors, colleagues, and friends in the Department of Combinatorics and Optimization at the University of Waterloo.

Thank Ontario Graduate Scholarship Program, NSERC, and Bell Canada for their financial support during my PhD study.

Thanks my parents, my brother for their love and continues encouragement. Although they were not in Canada when I was writing the thesis, I can always feel their support.

Last, I owe great thanks to my wife, Feng Zou, for her love, encouragement, and being my company for countless good or bad days. This thesis is dedicated to my daughter Laura, who just turned one year old when I finished the first draft.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview and Outline of Thesis

The main goal of this thesis is to investigate the numerical stability for Linear Programming, LP, and Semidefinite Programming, SDP.

We first investigate the long puzzling fact that most of the practical, popular, interior point LP solvers can attain solutions with 10^{-8} accuracy, even when the condition number of the underlying linear system can be as large as 10^{30}. The standard condition number based error analysis, which predicts the worst case accuracy of the solution to a linear system by the condition number, is overly pessimistic in this case, e.g. Stewart and Sun [91, p120]:

> If a matrix has a condition number of 10^k and its elements are perturbed in their t-th digits, then the elements of its inverse will be perturbed in their $(t-k)$-th digits.

Since most popular 32-bit PCs have a machine precision of about 10^{-16}, we see almost no accuracy in the inverse of a matrix when the condition number is larger than 10^{16}. Although, we generally do not form the inverse of a matrix explicitly when solving a linear system, ill-conditioning still explains well the worst case forward error. Solving for the search direction for LP problems involves highly ill-conditioned linear systems. We show that for certain LP starting point, this ill-conditioning do cause serious error (see Example 4.1 (p68)). However, in practice, we observe much better accuracy than the condition numbers suggest. In this

thesis we investigate this phenomena and demonstrate that it is a result of the LP algorithm special structure.

Based on our error analysis, we propose a simple modification to the popular *normal equation* LP solver. This new method demonstrates better numerical stability. It is more efficient when the LP problem has a certain special structure. We also discuss the technique of using a pure Newton's method at the final stage of the interior point method to get quadratic convergence. Purify step, which identifies those variables that converge to zero and eliminates them to get a smaller system, is discussed. Due to the stability of the new system, we investigate the interior point method without backtracking steps, i.e., once we have the search direction, we go all the way to the boundary.[1]

For interior point algorithms in SDP, the same ill-conditioning as in LP is observed. However, we do not have the same surprising numerical stability when solving for the search direction. Although, most of the interior point algorithms for SDP are extensions of LP algorithms, it is observed that the SDP algorithms have many important differences. For example SDP needs a constraint qualification to guarantee strong duality. Moreover, unlike the LP case, SDP may not have a strictly complementary primal-dual optimal solution. The strict complementarity condition plays a crucial role in much of the SDP theory. For example, we need strict complementarity to ensure that the central path converges to the analytic center of the optimal face, see [46, 64]. Also, many of the local superlinear and quadratic convergence results for interior point methods depend on the strict complementarity assumption, e.g., [84, 50, 4, 64, 59]. In this thesis, we derive a procedure to generate a class of problems for which we can control the size of the strict complementarity gap. These problems provide *hard instances* for testing SDP algorithms. We also develop measures to estimate the size of the strict complementarity gap.[2]

1.2 Historical Perspective

Modern operation research starts with Danzig's simplex method for LP [18]. The simplex method moves from one vertex to an adjacent vertex of the feasible set and tries to improve

[1] This part of the thesis is based on the report [41].

[2] This part of the thesis is based on the report [106].

the objective value at each step. It is effective in solving most practical problems; and it generally requires at most $2m$ to $3m$ iterations, where m is the number of constraints of the LP in standard form, see [77, pp391]. It is shown by Borgwardt and Huhn [12], and Smale [89], that the expected average number of iterations for the simplex method is polynomial. The more recent smoothed analysis by Spielman and Teng [90] reveals that the smoothed complexity of the simplex method is polynomial in: the input size and standard deviation of Gaussian perturbations.

However, there is no worst case polynomial complexity bound for any type of simplex method so far. By the inherent *combinatorial* property of simplex methods, worst case scenarios may be constructed to go through every vertex of the feasible region; and thus the running time becomes exponential. It was shown by Klee and Minty [56] that under a standard pivoting rule, the worst case scenario does happen.

The lack of a polynomial complexity bound for the simplex method motivated people to find a polynomial time algorithm. Khachian [54, 55], using the ellipsoid method of Shor [88] and Yudin and Nemirovskii [120], was the first to give a polynomial algorithm for LP. However, contrary to the theoretical polynomial-time convergence property, which suggests it should be a fast algorithm, the ellipsoid method performs poorly in practice compared to the simplex method. It usually achieves the worst case theoretical bound for the number of iterations.

More recently, Karmarkar's seminal paper [53] in 1984 gave a polynomial time algorithm for LP; and, it was announced as more efficient than the simplex method. Contrary to the inherent *combinatorial* property of the simplex method, Karmarkar's algorithm is more like an algorithm working on a nonlinear optimization problem. It evolves through a series of strictly feasible points (interior points), and converges to an optimal solution. That is why it and its successor variants are called interior point methods.

Karmarkar's paper attracted many researchers into this area. Vanderbei, Meketon, and Freedman [102] and Barnes [8] proposed a natural simplification of Karmarkar's algorithm, called the affine scaling method. It turned out that as early as 1967, Dikin [26] had a very similar proposal.

It was shown by Gill Murray, Saunders, Tomlin, and M. Wright [36] that there was an equivalence between Karmarkar's primal potential based interior point method and the

classical logarithmic barrier method applied to LP. However, the logarithmic barrier method, which was popularized by Fiacco and McCormick [28] long back in the sixties, lost favour due to the inherent ill-conditioning of the underlying Newton system. However, the huge condition numbers of the Newton system in current versions of interior point methods have not stopped its successful implementation. The lost interest in logarithmic barrier methods has been reignited by the efficiency of interior point methods for LP.

Many researchers have questioned why interior point LP solvers have such numerical robustness. Error analysis for interior point methods has been studied in the literature. S. Wright [115, 112] did a thorough error analysis on the augmented system for LP. He showed that the ill-conditioning of the augmented system does not cause major problems for the search direction for non-degenerate problems. Forsgren, Gill, and Shinnerl [29] performed a similar analysis in the context of logarithmic barrier methods for nonlinear problems. M. Wright [111] worked on the ill-conditioning of the condensed system (equivalent to the normal system in LP) for nonlinear programming problems. Her work assumed positive definiteness of the Hessian of the Lagrange function, an assumption that does not hold in the LP case. The most closely related work to ours is that done in S. Wright [116]. He did the analysis for the normal equation approach for LP based on a class of particular modified Cholesky solvers. This class of modified Cholesky solvers are adapted for many of the practical solvers. He explained why we usually see convergence to a relative accuracy of 10^{-8} with certain numerical estimation on the size of computed search directions.

Besides the global polynomial-time convergence rate analysis, there are has been a lot of researches done on the local asymptotic convergence rate of the interior point method. They show that interior point method can have a quadratic convergence rate. See for example Tapia and Zhang [94], Ye, Güler, Tapia and Zhang [119], and Tunçel [97].

The work of Nesterov and Nemirovski [73, 74] generalized the logarithmic barrier based interior point methods and the complexity analysis to general convex programming problems. A special application is SDP. Independently, Alizadeh extended interior point methods from linear programing to semidefinite programming [1, 2, 3].

Since SDP has polynomial time algorithms and it is more general than LP, many applications are developed based on SDP. Lovász introduced one of the most interesting and exciting applications in combinatorial optimization in his paper about the *theta function* [63]. (See

also [58] for more references and details.) The now classical Goemans and Williamson paper [38, 37] provided a significant improvement for a polynomial time approximation bound for the max-cut problem. This generated more attention and applications. For a more complete review see [108].

However, SDP generally has less desirable numerical properties than LP. Several papers addressed the numerical problems of SDP, e.g. [4, 61, 62, 70, 93, 96]. It is harder to get high accuracy solution for SDP than for LP using the current popular algorithms. Unlike the LP case, ill-conditioning causes major problems in SDP. In general, the so-called AHO direction [5], has better numerical accuracy in the final stages of their interior point method in SDP than the HRVW/KSH/M [48, 60, 71] and NT [75, 76] search directions.

Kruk, Muramatsu, Rendl, Vanderbei, and Wolkowicz [62] used a Gauss-Newton type method and show that they can get high accuracy solutions for SDP. But since the dimension of the Gauss-Newton system is large, $n(n+1)/2$, solving such a system is expensive when n is large. Sturm [93] proposed an implementation of the NT direction to overcome some of the numerical difficulties. Instead of keeping the X and Z variables, the implementation factors these variables using a product of a stable U-factor and a well conditioned matrix. Over the iterations, the algorithm updates the stable U-factor and the well conditioned matrix. His implementation then achieves relative high accuracy for the NT direction for some of the SDPLIB problem set, [11].

Chapter 2

Fundamentals of Linear Programming

2.1 Basic Theorems of Linear Programming

We consider the Linear Programming (LP) problem and its dual program in the following form:

$$\begin{array}{lllll} & p^* := & \min & c^T x & \\ \text{(LP)} & & \text{s.t.} & Ax = b & \\ & & & x \geq 0 & \end{array} \qquad \begin{array}{lllll} & d^* := & \max & b^T y & \\ \text{(DLP)} & & \text{s.t.} & A^T y + z = c & \\ & & & z \geq 0, & \end{array} \tag{2.1}$$

where A is a full row rank matrix in $\mathbb{R}^{m\times n}$, c is in $\mathbb{R}^n$, and b is in $\mathbb{R}^m$. The variable x in the primal (LP) is thus in $\mathbb{R}^n$ and the variables y and z in the dual (DLP) are in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.

The following is the well known weak duality relation for LP.

Theorem 2.1 (Weak Duality) *Let $\bar{x}$ and $(\bar{y}, \bar{s})$ be a feasible solution for (LP) and (DLP) respectively, then the primal objective value is greater than or equal to the dual objective value, that is*

$$c^T\bar{x} \geq b^T\bar{y}, \quad \textit{and} \quad c^T\bar{x} - b^T\bar{y} = \bar{x}^T\bar{s}\ .$$

Proof.

$$c^T\bar{x} = (A^T\bar{y} + \bar{s})^T\bar{x} = \bar{y}^T A^T x + \bar{s}^T\bar{x} = \bar{y}^T b + \bar{x}^T\bar{s}\ .$$

Because $\bar{x} \geq 0$ and $\bar{s}^T \geq 0$, we have $c^T\bar{x} \geq b^T\bar{y}$. ■

Strong duality holds for LP as well. See for example [113, Theorem 2.1,p25].

Theorem 2.2 (Strong Duality) *1. Suppose that (LP) and (DLP) are feasible. Then optimal solutions for (LP) and (DLP) exist, and their optimal values are equal.*

2. If either problem (LP) or (DLP) has an optimal solution, then so does the other, and the objective values for both are equal.

The well-known primal-dual optimality conditions (primal feasibility, dual feasibility, and complementary slackness) follow from the weak and strong duality properties. In the following theorem, we use X and Z to denote $n \times n$ diagonal matrices whose diagonals are x and z, respectively. The vector e is the vector of all ones.

Theorem 2.3 *The primal-dual variables (x, y, z), with $x, z \geq 0$, are optimal for the primal-dual pair of LPs if and only if*

$$F(x, y, z) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ ZXe \end{bmatrix} = 0. \tag{2.2}$$

Another important property of LP is the existence of a strict complementarity optimal solution pair, i.e. the Goldman-Tucker Theorem [40]. We define two index sets denoted by $\mathcal{B}$ and $\mathcal{N}$.

$$\mathcal{B} := \{i \in \{1, 2, \ldots, n\} : x_i^* > 0 \text{ for some optimum } x^* \text{ to problem (LP) }\}; \tag{2.3}$$

$$\mathcal{N} := \{i \in \{1, 2, \ldots, n\} : z_i^* > 0 \text{ for some dual optimum } z^* \text{ to problem (DLP)}\}. \tag{2.4}$$

Theorem 2.4 (Goldman-Tucker) *If an LP has an optimal solution, then there must exist a strict complementary pair of optimal solutions x^* and z^* such that $x^* + z^* > 0$. In other words, the two index sets $\mathcal{B}$ and $\mathcal{N}$ are a partition of the indices $\{1, 2, \ldots, n\}$. That is $\mathcal{B} \cap \mathcal{N} = \emptyset$ and $\mathcal{B} \cup \mathcal{N} = \{1, 2, \ldots, n\}$.*

2.2 Central Path

We define a pair of families of non-linear programming problems, parameterized by $\mu > 0$:

$$\begin{array}{lll} & \min & c^T x - \mu \sum_{i=1}^n \ln x_i \\ (\mathrm{LP}_\mu) & \text{s.t.} & Ax = b \\ & & (x > 0) \end{array} \qquad \begin{array}{lll} & \max & b^T y + \mu \sum_{i=1}^n \ln z_i \\ (\mathrm{DLP}_\mu) & \text{s.t.} & A^T y + z = c \\ & & (z > 0) \end{array}$$

The functions $-\sum_{i=1}^n \ln x_i$ and $\sum_{i=1}^n \ln z_i$ are called the barrier functions (for the primal and dual, respectively). These barrier functions force the inequality constraints to hold. So, the inequality constraints are implicit here.

Theorem 2.5 *Suppose the primal (LP) and the dual (DLP) problems both have strictly feasible solutions, then (LP_μ) and (DLP_μ) have a unique optimal solution pair $x(\mu)$, $\big(y(\mu), z(\mu)\big)$, for each $\mu > 0$.*

Proof. Let $\bar{x}$ and $\bar{z}$ be fixed strictly feasible solutions to (LP) and (DLP), respectively. Then, there exists $\bar{y}$ such that $A^T\bar{y} + \bar{z} = c$ and

$$\bar{z}^T x - c^T x = (c - A^T\bar{y})^T x - c^T x = -\bar{y}^T b, \quad \text{a constant.}$$

Therefore, the objective function of (LP_μ) can be rewritten as $f(x) := \bar{z}^T x - \mu \sum_{i=1}^n \ln x_i$. The function f is strictly convex and goes to infinity if any of the entries of x go to 0 or infinity. Thus the set $\{x : f(x) \le f(\bar{x})\}$ is bounded and closed. We are minimizing a strictly convex function over a compact set. Thus the minimizer for LP_μ exists and is unique.

Similarly, we can prove that the solution for the dual (DLP_μ) is also unique. ■

Theorem 2.6 *Suppose the primal (LP) and the dual (DLP) problems both have strictly feasible solutions. Then for a fixed $\mu > 0$, the unique solutions $x(\mu)$, $\big(y(\mu),\ z(\mu)\big)$ of (LP_μ) and (DLP_μ) make up the unique solution to the following system:*

$$\begin{aligned} Ax &= b, \quad x > 0\,, \\ A^T y + z &= c, \quad z > 0\,, \\ Xz &= \mu e\,. \end{aligned} \tag{2.5}$$

Proof. We use the Karush-Kuhn-Tucker (KKT) conditions to prove the theorem. For the parameterized primal problem (LP_μ), the Lagrangian function and its derivatives are:

$$\begin{aligned} L(x,\lambda) &:= (c^T x - \mu \sum_{i=1}^{n} \ln x_i) - (Ax-b)^T\lambda \ , \\ \nabla_x L(x,\lambda) &= c - \mu X^{-1} e - A^T\lambda \ , \\ \nabla^2_{xx} L(x,\lambda) &= X^{-2} \ . \end{aligned}$$

The Hessian of the Lagrangian is positive definite. So, the KKT conditions, $\nabla_x L(x,\lambda) = 0$, are both sufficient and necessary in this case. Let $z := \mu X^{-1} e > 0,\ y := \lambda$. Then $Xz = \mu e$. Moreover, $\nabla_x L(x,\lambda) = 0$ is equivalent to $A^T y + z = c$. Also, because x is a feasible solution to the problem (LP_μ), we must have $Ax = b$ and $x > 0$. Thus system (2.5) is a restatement of the KKT conditions of problem (LP_μ). So, a solution of system (2.5) is equivalent to the optimal solution of (LP_μ). Theorem 2.5 shows that (LP_μ) has a unique solution. Thus, this also proves that the solution of system (2.5) is unique.

The proof for the dual (DLP_μ) part is similar. ■

If a feasible solution pair $(x,(y,z))$ satisfies system (2.5) for some $\mu > 0$, then we say that they are on the central path.

As μ goes to 0, $x(\mu)^T z(\mu)$, which is μn, also goes to 0. So if $x(\mu)$ and $z(\mu)$ converge, then $x(\mu)$ and $z(\mu)$ must converge to a solution of the system (2.2), which is an optimal solution pair to the primal (LP) and dual (DLP) problem. McLinden [67] proved the following theorem for the monotone linear complementarity problem, which includes linear programming.

Theorem 2.7 *Let $(x(\mu),(y(\mu),z(\mu)))$ be on the central path. Then $(x(\mu),(y(\mu),z(\mu)))$ converges to an optimal solution pair for primal (LP) and dual (DLP) problem.*

Ye [118, Theorem 2.17, p72] shows that the central path converges to a pair of strict complementary solutions, which are the analytic center of the primal and dual optimal face, respectively.

So, if we can find a feasible pair for (LP_μ) and (DLP_μ), and decrease μ at each iteration, we will obtain an optimal solution. This is the basic idea behind the path-following methods.

Since it is expensive to get an exact optimal solution for (LP_μ) and (DLP_μ), we usually find an approximate solution near the optimal solution of the central path, and then decrease μ and go to the next iteration. Usually a neighbourhood of the central path is defined to theoretically guarantee good progress of algorithms. Before we give several examples of neighbourhoods of the central path, we first give the notation for the feasible region $\mathcal{F}$ and strictly feasible region $\mathcal{F}_+$ as follows:

$$\mathcal{F}(P) := \{x : x \text{ is feasible for primal problem (LP)}\},$$

$$\mathcal{F}(D) := \{z : z \text{ is feasible for dual problem (DLP)}\},$$

$$\mathcal{F}_+(P) := \{x > 0 : x \in \mathcal{F}(P)\}, \text{ and } \mathcal{F}_+(D) := \{z > 0 : z \in \mathcal{F}(D)\}.$$

The following are some examples of the neighbourhoods of the central path.

Example 1: $\mathcal{N}_2(\beta) := \{(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D) : \|Xs - \mu e\|_2 \le \beta\mu\}$.

Example 2: $\mathcal{N}_\infty(\beta) := \{(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D) : \|Xs - \mu e\|_\infty \le \beta\mu\}$.

Example 3: $\mathcal{N}_\infty^-(\beta) := \{(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D) : \|Xs - \mu e\|_\infty^- \le \beta\mu\}$.

Here, for $v \in \mathbb{R}^n$, $\|v\|_\infty^- := -\min\{0, \min_j\{v_j\}\}$.
Clearly, for $v \in \mathbb{R}^n$, $\|v\|_2 \ge \|v\|_\infty \ge \|v\|_\infty^-$. So, for every $\beta \ge 0$, we have

$$\mathcal{N}_2(\beta) \subseteq \mathcal{N}_\infty(\beta) \subseteq \mathcal{N}_\infty^-(\beta).$$

2.3 Algorithms

A natural way to solve a nonlinear system like (2.2) and (2.5) is to use Newton's method. However, due to the non-negativity constraints in the optimality conditions (2.2), it is generally impossible to guarantee that Newton's method converges correctly to the nonnegative solution. However, when μ is sufficiently large, the central path neighbourhood ($\mathcal{N}_2(\beta), \mathcal{N}_\infty(\beta)$, or $\mathcal{N}_\infty^+(\beta)$) is much larger compared with the one when μ is small. Thus when μ is sufficiently large, the effect of the non-negativity constraints of x and z is negligible and Newton's method can be directly applied in this case. Thus the path-following method starts with a big μ value and solves (2.5) approximately. It then decreases the value of μ at each iteration.

We list an algorithmic framework below. There are many variants of interior point methods for LP. Almost all of them share this similar algorithmic framework. We define

$$F_{\sigma\mu}(x, y, z) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ Xz - \sigma\mu e \end{bmatrix}. \tag{2.6}$$

The Jacobian of $F_{\sigma\mu}$ is

$$F'_{\sigma\mu} = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ X & 0 & Z \end{bmatrix}.$$

Algorithm 1 Interior Point Method Framework for LP

Require: x and z both positive; $\epsilon > 0$ desired tolerance

1: **while** $x^T z > \epsilon$ or $\|Ax - b\| + \|A^T y + z - c\| \geq \epsilon$ **do**

2: solve $F'_{\sigma\mu}(x, y, z) \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = -F_{\sigma\mu}(x, y, z)$, where $\sigma \in [0, 1]$ and $\mu = x^T s/n$;

3: choose $\alpha > 0$, such that $(x^+, z^+) := (x, z) + \alpha(dx, dz) > 0$;

4: let $x := x^+, z := z^+, y := y + \alpha dy$;

5: **end while**

6: **return** solution (x, y, z).

Many algorithms differ in the choice of the parameter σ and the step length α. For example, if we set the parameter σ to 1, then we call the search direction the "centering direction". The Newton search direction then aims toward a solution on the central path with the fixed value μ. However, if we set the parameter σ to 0, then we call the search direction the "affine scaling direction". The search direction then aims toward the optimal solution of the original LP.

One of the most successful heuristics in practice is Mehrotra's predictor-corrector approach [68]. It has two steps: the predictor step and the corrector step. In the predictor step, it first sets $\sigma = 0$ and finds the affine scaling direction dx, dy, dz in step 2 of the above algorithm. Then it finds a maximal step over this search direction such that $x + \alpha dx$ and

$z + \alpha dz$ are both nonnegative. It then evaluates the progress for the affine scaling direction by calculating the centering value

$$\sigma = [(x + \alpha dx)^T (z + \alpha dz)/x^T z]^3. \tag{2.7}$$

In the corrector step, it substitutes the right-hand side of the linear equation in step 2 Algorithm 1 with $[0,\ 0,\ \sigma\mu e - dx \circ dz]^T$ and solves for the search direction, where σ comes from (2.7), the dx and dz come from the affine scaling direction, and $\circ$ means the Hadamard product (entry-wise product). The final search direction is the sum of the predictor direction and corrector direction.

The predictor step tries to predict how far the search direction can go if we aim at the optimal solution. The quantity σ is a natural indicator of the predictor step's progress. If the predictor step goes well, then we can aim to a smaller $\sigma\mu$ on the central path. If the predictor step does not have a large step α, then our σ is larger and the step is more like a centering step. The corrector step then uses the information from the predictor step, the σ, to decide how much weight to put in the centering direction. Also, the $dx \circ dz$ in the corrector step is a second order approximation of the linearization. We can see that if there are dx and dz such that $(x + dx) \circ (z + dz) = \sigma\mu$, then we have $Xdz + Zdx = -XZe + \sigma\mu - dx \circ dz$.

The two-step procedure is efficient in implementations. The extra corrector direction with the new right-hand side can be quickly obtained using the LU factorization from the predictor step.

Chapter 3

Numerical Stability in Linear Programming

3.1 Introduction

Ill-conditioning has an interesting history and a growing influence in optimization. For example, logarithmic barrier methods for minimization were proposed in the 1950s and popularized in the 1960s, see e.g. [35, 28, 109, 110]. These methods lost favour because, at each iteration, they need to solve a linear system (the Newton equation) that becomes increasingly ill-conditioned as the iterates approach an optimum. Current interior point methods are based on a logarithmic barrier approach. The optimality conditions that arise from minimizing the log-barrier function (in particular, the complementary slackness part) are typically modified to avoid the ill-conditioning, see e.g. [28]. However, the popular interior point methods, e.g. those that solve the so-called normal equations or the augmented equations, result in another level of ill-conditioning. When solving the Newton equation, block elimination is introduced to take advantage of the sparse structure. This results in a Jacobian that is singular at the optimum, i.e. ill-conditioning arises as the iterates approach an optimum. However, in practice, most of the LP codes behave surprisingly well, even with huge condition numbers. This raises many questions concerning the error analysis.

In this chapter, we study error bounds of the search directions in the normal equation approach for LP. We show that, although the condensed central block after the block

eliminations, with matrix $AXZ^{-1}A^T$, may not be ill-conditioned for non-degenerate LPs, the Jacobian of the complete system is still ill-conditioned. Its condition number diverges to infinity when the x and z variables approaches the optimal solution. We then study the accuracy of the solutions of the complete ill-conditioned system. We derive the error bounds for the search directions under certain degeneracy and certain non-degeneracy assumptions.

Our work differs from previous works in the sense that we only assume a general backward stable linear solver and we give a complete error analysis for all cases: non-degenerate, degenerate, centering direction, and affine scaling direction. We also give numerical examples to show that all of our derived bounds are tight. One of the most influential paper by M. Wright [111] analyzes a similar condensed system in nonlinear-programming. However, her work assumes that the Hessian of the Lagrange function is positive definite, as a result it can not be applied to the LP case. Our work for the non-degenerate case is similar to her work. S. Wright [115] investigates the error for the augmented system. His another work [116] analyzes the error in the normal equation system for a class of modified Cholesky factorizations with certain empirical estimates on the size of the computed search direction dy. He also explains why most of the popular LP solvers' default stop tolerance is 10^{-8}.

We assume we are working on a popular 32-bit computer with machine precision approximately 10^{-16}. We use m to denote the number of constraints in the standard equality form.

We obtain the following results on the search directions.

1. The best error bound is obtained for the non-degenerate case. The maximum step length computed using the computed search direction has only unit error relative to the step length computed from the exact search direction. Therefore, the normal equation (NEQ) based interior point method can get a solution with accuracy of about 10^{-16}.

2. For the degenerate case with $\text{rank}\,(A_{\mathcal{B}}) < m$:

   (a) when σ is small, $(O(\mu))$, the search direction is close to the affine scaling direction. Then we obtain a good error bound for the search direction. The NEQ based interior point method can get a solution with accuracy of 10^{-8}.

(b) when σ is large, the search direction is close to the centering direction. This results in the worst error bound for the search direction. It may not yield a correct step length.

3. For the degenerate case with $\text{rank}\,(A_{\mathcal{B}}) = m$:

   the magnitude of the error bound lies between that of the non-degenerate case (Item 1) and the affine scaling direction in the degenerate case (Item 2a). However, depending on the σ parameter, the step length might be inaccurate. If σ is small, the error on the step length is no worse than the case in Item 2a. It σ is large, the error on the step length can be large.

Since most practical codes use the predictor-corrector heuristic, and the predictor-corrector heuristic usually gives a small σ value at the final stage of interior point method, the above error bounds explains well why in practice, most of the solvers can get solutions with 10^{-8} accuracy, even for the degenerate case. This explains well why 10^{-8} is the standard tolerance for most solvers.

3.1.1 Preliminaries

We consider the linear program in standard form, (2.1) (p6). The optimality conditions are given in (2.2). For interior point methods, we use the perturbed optimality conditions

$$F_{\sigma\mu}(x, y, z) = 0, \tag{3.1}$$

with $x, z > 0$, where $F_{\sigma\mu}$ is defined in (2.6). After linearization, we have the Newton equation

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -ZXe + \sigma\mu e \end{bmatrix}, \tag{3.2}$$

where $0 \le \sigma \le 1$ is the centering parameter, and r_p and r_d are the primal and dual residual vectors, respectively,

$$r_p := Ax - b, \qquad r_d := A^T y + z - c. \tag{3.3}$$

Instead of solving the above linear system (3.2) directly, the normal equation approach uses certain block eliminations to exploit the sparsity (see Section 4.2.2). After the block eliminations, we get the following linear system.

$$\begin{bmatrix} 0 & A^T & I_n \\ 0 & AZ^{-1}XA^T & 0 \\ I_n & -Z^{-1}XA^T & 0 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p + A(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e) \\ Z^{-1}Xr_d - x + \sigma\mu Z^{-1}e \end{bmatrix}. \tag{3.4}$$

We solve for dy first, and then back-solve for dx and dz. This way, we are solving a smaller, positive definite, system of size m. However, the block elimination brings back instability (ill-conditioning). It is shown in [41] as well as in Proposition 4.2 (p71) that the condition number of the matrix in (3.4) goes to infinity as x and z approach an optimum, even for non-degenerate problems. It is also shown in Example 4.1 (p68) that if the residuals r_p and r_d are relatively large, then the roundoff errors in the calculation of the search directions can be catastrophic. Thus, this verifies that large condition numbers for the linear system can result in inaccurate solutions.

Notation

We use $\mathbf{u}$ to denote unit roundoff, see e.g. [49, p42–44], i.e. for any real number x in the range of a floating-point number system and any two representable numbers y and z in that floating-point system, $\mathbf{u}$ is the smallest positive number such that

$$\mathrm{fl}(x) = x(1+\delta) \text{ and } \mathrm{fl}(y \text{ op } z) = (y \text{ op } z)(1+\delta), \qquad |\delta| \le \mathbf{u}, \tag{3.5}$$

where $\mathrm{fl}(\cdot)$ denotes the floating point representation of a number and op denotes an arithmetic operation (i.e., $+, -, \times, /, \sqrt{\cdot}$). With binary IEEE arithmetic, $\mathbf{u} \simeq 6 \times 10^{-8}$ in single precision and $\mathbf{u} \simeq 1.1 \times 10^{-16}$ in double precision.

We also use the order notation $O(\cdot)$ in a slightly unconventional way (following S. Wright [115]). When x and y are two numbers depending on a parameter μ, we write $x = O(y)$ if there exists a constant C (not too large and independent of μ) such that $|x| \le C|y|$. We write $x = \Theta(y)$ if $x = O(y)$ and $y = O(x)$. For matrix A, we write $A = O(y)$ if $\|A\| = O(y)$. Such notation ($O(\cdot)$ and $\Theta(\cdot)$) will greatly simplify the analysis and presentation. However, when some of the constant C in the $O(\cdot)$ notation becomes too large, many of the results

may not be true any more. Also, "there are too many unknown factors and mathematically imprecise rules of thumb to permit a rigorous theorem. ([111])" Thus, we make the following assumptions. We also give numerical examples to verify our results.

We let $\mathcal{B}, \mathcal{N}$ represent a partition of the indices as defined in (2.3) and (2.4).

Assumptions

Throughout the chapter we use some or all of the following assumptions about the floating point operations.

Assumption 3.1 *1. For real matrices A, B, C, with dimensions not too large, and with elements that are in the range of floating-pointing number system, we have*

$$fl(A) = A + E_1 \quad \textit{and} \quad fl(B \text{ op } C) = B \text{ op } C + E_2,$$

where the op *denotes an matrix operation (i.e., $+, -, \times$), $\|E_1\| = O(\mathbf{u})\|A\|$ and $\|E_2\| = O(\mathbf{u})\|B \text{ op } C\|$. In this chapter, we use the simplified notation*

$$fl(B \text{ op } C) = B \text{ op } C + O(\delta),$$

where $O(\delta)$ denotes the perturbation matrix E_2 that satisfies $\|E_2\| = O(\delta)$.

2. All the input data A, b, and c of the LP problem are floating point representable. i.e.

$$fl(A) = A, \quad fl(b) = b, \quad fl(c) = c.$$

All the intermediate computed variables x, y, z, and μ are also floating point representable. i.e

$$fl(x) = x, \quad fl(y) = y, \quad fl(z) = z, \quad \textit{and} \quad fl(\mu) = \mu.$$

We make the assumption in Assumptions 3.1 item 2 because when we consider the numerical stability of a search direction, we usually consider a particular iteration of the interior point method with data $A, b, c,\ x, y, z,$ and μ. This data is stored in the computer and thus is floating point representable. Another consideration of this assumption is to make the analysis easier to read. Having a unit relative round off error on the data will not have any difference on our results.

For most results we use the following assumption on the order of the data and the iterates. Let $\mathcal{B}$ and $\mathcal{N}$ be the partition of the indices according to the Goldman-Tucker Theorem (Theorem 2.4).

Assumption 3.2 *1. The data A is not too large, i.e. $A = \Theta(1)$. The matrix A has full row rank and the smallest nonzero singular values of A and $A_{\mathcal{B}}$ are both $\Theta(1)$.*

2. The parameter μ is sufficiently small. The sequence of iterates (x, y, z) generated by the interior point algorithm satisfies the following properties:

$$x_i = \Theta(1) \ \ (i \in \mathcal{B}), \qquad z_i = \Theta(1) \ \ (i \in \mathcal{N}), \tag{3.6}$$

$$x_i = \Theta(\mu) \ \ (i \in \mathcal{N}), \qquad z_i = \Theta(\mu) \ \ (i \in \mathcal{B}). \tag{3.7}$$

(This assumption means x, z are in some neighbourhood of the central path, see e.g. [113].)

3. In addition, the residuals defined in (3.3) are $O(\mu)$; that are,

$$r_p = O(\mu), \qquad r_d = O(\mu). \tag{3.8}$$

Our assumption that μ is sufficiently small in Item 2 means that the μ value is small enough so that we can clearly see the difference between the quantities $x_{\mathcal{B}}$ ($\Theta(1)$) and $x_{\mathcal{N}}$ ($\Theta(\mu)$). Notice that the size of $x_{\mathcal{B}}$ ($\Theta(1)$) depends on the input data A, b, c. In practice, if μ is less than 10^{-3} then it usually can be treated as small enough for most of the problems.

Our analysis in the non-degeneracy section requires the following assumption.

Assumption 3.3 *The problem is non-degenerate. More specifically, we require*

$$|\mathcal{B}| = m \quad \textit{and} \quad (A_{\mathcal{B}} A_{\mathcal{B}}^T)^{-1} = \Theta(1).$$

In particular, this implies that the condition number of $A_{\mathcal{B}} A_{\mathcal{B}}^T$ is not too large. (Here $A_{\mathcal{B}}$ denotes a submatrix of A whose columns are specified by the index set $\mathcal{B}$.)

3.2 Properties of the Normal Equation System

In this section, we present a few properties of the normal equation system. The theorems illustrate the structural information on the matrix $AXZ^{-1}A^T$. We also give the roundoff error on the right-hand side of the normal equation. The properties in this section hold for the normal equation system in general, regardless of degeneracy.

3.2.1 Roundoff Error in the Right-Hand Side

Lemma 3.4 *Suppose that Assumption 3.2 (items 1,2) holds. Then the floating point representations of the residuals in* (3.3) *satisfy*

$$fl(r_p) - r_p = O(\mathbf{u}), \qquad fl(r_d) - r_d = O(\mathbf{u}).$$

Proof.

$$\begin{aligned}
\mathrm{fl}(r_p) &= \mathrm{fl}(Ax - b) \\
&= \mathrm{fl}(Ax) - \mathrm{fl}(b) + O(\mathbf{u}) && \text{(by Assumption 3.2 (items 1,2))} \\
&= Ax + O(\mathbf{u}) - b + O(\mathbf{u}) && \text{(since } Ax \text{ is } O(1)) \\
&= r_p + O(\mathbf{u}).
\end{aligned}$$

$$\begin{aligned}
\mathrm{fl}(r_d) &= \mathrm{fl}(A^T y + z - c) \\
&= \mathrm{fl}(A^T y) + \mathrm{fl}(z) - \mathrm{fl}(c) + O(\mathbf{u}) && \text{(since } A^T y + z \text{ is } O(1)) \\
&= A^T y + O(\mathbf{u}) + z - c + O(\mathbf{u}) && \text{(since } A^T y \text{ is } O(1)) \\
&= r_d + O(\mathbf{u}).
\end{aligned}$$

■

Lemma 3.5 *Assume that the scalars $\beta = \Theta(\mu)$ and $\theta = \Theta(1)$. Then*

$$fl(1/\beta) = 1/\beta + O(\mathbf{u}/\mu), \qquad fl(1/\theta) = 1/\theta + O(\mathbf{u}).$$

Proof. This follows from a direct application of (3.5). ■

Theorem 3.6 *Suppose Assumption 3.2 holds. Then the floating point roundoff error in the right-hand side in the middle block of the normal equation system is $O(\mathbf{u}/\mu)$, more specifically,*

$$\begin{aligned} &fl(-r_p + A(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)) = \\ &\qquad -r_p + A(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e) + \{A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u})\}. \end{aligned}$$

Proof. If the index $i \in \mathcal{B}$, then

$$\begin{aligned} \mathrm{fl}(-z_i^{-1}x_i(r_d)_i) &= \mathrm{fl}(-z_i^{-1}x_i)\mathrm{fl}((r_d)_i) + O(\mathbf{u}), \quad (\text{since } z_i^{-1}x_i(r_d)_i \text{ is } O(1)\ ) \\ &= [\mathrm{fl}(-z_i^{-1})\mathrm{fl}(x_i) + O(\mathbf{u}/\mu)]((r_d)_i + O(\mathbf{u})) + O(\mathbf{u}), \quad (\text{since } z_i^{-1}x_i \text{ is } \Theta(\tfrac{1}{\mu})) \\ &= [(\underline{-z_i^{-1}} + O(\mathbf{u}/\mu))\underline{x_i} + O(\mathbf{u}/\mu)]((r_d)_i + \underline{O(\mathbf{u})}) + O(\mathbf{u}) \\ &= -z_i^{-1}x_i(r_d)_i + O(\mathbf{u}/\mu), \end{aligned} \tag{3.9}$$

where the error term $O(\mathbf{u}/\mu)$ in the last step comes from the $z_i^{-1}x_iO(\mathbf{u})$ term as underlined. Other error terms are much smaller than $O(\mathbf{u}/\mu)$ and thus can be folded into this error term.

If index $i \in \mathcal{N}$, then

$$\begin{aligned} \mathrm{fl}(-z_i^{-1}x_i(r_d)_i) &= \mathrm{fl}(-z_i^{-1}x_i)\mathrm{fl}((r_d)_i) + O(\mu^2\mathbf{u}), \ \ (\text{since } z_i^{-1}x_i(r_d)_i \text{ is } O(\mu^2)\ ) \\ &= [\mathrm{fl}(-z_i^{-1})\mathrm{fl}(x_i) + O(\mu\mathbf{u})]((r_d)_i + O(\mathbf{u})) + O(\mu^2\mathbf{u}), \quad (\text{since } z_i^{-1}x_i \text{ is } \Theta(\mu)) \\ &= [(\underline{-z_i^{-1}} + O(\mathbf{u}))\underline{x_i} + O(\mu\mathbf{u})]((r_d)_i + \underline{O(\mathbf{u})}) + O(\mu^2\mathbf{u}) \\ &= -z_i^{-1}x_i(r_d)_i + O(\mu\mathbf{u}), \end{aligned} \tag{3.10}$$

where the $O(\mu\mathbf{u})$ term in the last step comes from $z_i^{-1}x_iO(\mathbf{u})$ as underlined. For the $\sigma\mu Z^{-1}e$ part, if $i \in \mathcal{B}$, we have

$$\begin{aligned} \mathrm{fl}((\sigma\mu Z^{-1}e)_i) &= \mathrm{fl}(\sigma\mu {z_i}^{-1}) \\ &= \mathrm{fl}(\sigma\mu)\mathrm{fl}({z_i}^{-1}) + O(\mathbf{u}) \qquad\qquad (\text{since } \sigma\mu {z_i}^{-1} \text{ is } \Theta(1)) \\ &= \sigma\mu[{z_i}^{-1} + O(\mathbf{u}/\mu)] + O(\mathbf{u}) \\ &= (\sigma\mu Z^{-1}e)_i + O(\mathbf{u}). \end{aligned} \tag{3.11}$$

If $i \in \mathcal{N}$, we have

$$\begin{aligned}
\mathrm{fl}((\sigma\mu Z^{-1}e)_i) &= \mathrm{fl}(\sigma\mu {z_i}^{-1}) \\
&= \mathrm{fl}(\sigma\mu)\mathrm{fl}({z_i}^{-1}) + O(\mu\mathbf{u}) && \text{(since } \sigma\mu {z_i}^{-1} \text{ is } \Theta(\sigma\mu)) \\
&= \sigma\mu[{z_i}^{-1} + O(\mathbf{u})] + O(\mu\mathbf{u}) \\
&= (\sigma\mu Z^{-1}e)_i + O(\mu\mathbf{u}). && (3.12)
\end{aligned}$$

Thus, if $i \in \mathcal{B}$, we get

$$\begin{aligned}
&\mathrm{fl}((-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_i) \\
=&\mathrm{fl}((-Z^{-1}Xr_d + x)_i) - \mathrm{fl}((\sigma\mu Z^{-1}e)_i) + O(\mathbf{u}) && \text{(since both of the terms are } O(1)) \\
=&\mathrm{fl}((-Z^{-1}Xr_d)_i) + \mathrm{fl}(x_i) - \mathrm{fl}((\sigma\mu Z^{-1}e)_i) + O(\mathbf{u}) \\
=&(-Z^{-1}Xr_d)_i + x_i - (\sigma\mu Z^{-1}e)_i + O(\mathbf{u}/\mu). && \text{(using (3.9) and (3.11))} \quad (3.13)
\end{aligned}$$

Similarly, if $i \in \mathcal{N}$, we get

$$\begin{aligned}
&\mathrm{fl}((-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_i) \\
=&\mathrm{fl}((-Z^{-1}Xr_d + x)_i) - \mathrm{fl}((\sigma\mu Z^{-1}e)_i) + O(\mu\mathbf{u}) && \text{(since both of the terms are } O(\mu) \text{)} \\
=&\mathrm{fl}((-Z^{-1}Xr_d)_i) + \mathrm{fl}(x_i) - \mathrm{fl}((\sigma\mu Z^{-1}e)_i) + O(\mu\mathbf{u}) \\
=&(-Z^{-1}Xr_d)_i + x_i - (\sigma\mu Z^{-1}e)_i + O(\mu\mathbf{u}). && \text{(using (3.10) and (3.12))} \quad (3.14)
\end{aligned}$$

So the right-hand side error is bounded by the following

$$\begin{aligned}
&\mathrm{fl}(-r_p + A(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)) \\
=&\mathrm{fl}(-r_p) + \mathrm{fl}(A_{\mathcal{B}}(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_{\mathcal{B}}) + \mathrm{fl}(A_{\mathcal{N}}(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_{\mathcal{N}}) + O(\mathbf{u}) \\
=&-r_p + O(\mathbf{u}) + \mathrm{fl}(A_{\mathcal{B}})\mathrm{fl}((-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_{\mathcal{B}})) + O(\mathbf{u}) \\
&+ \mathrm{fl}(A_{\mathcal{N}})\mathrm{fl}((-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_{\mathcal{N}}) + O(\mu\mathbf{u}) + O(\mathbf{u}) \\
=&-r_p + A_{\mathcal{B}}[(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_{\mathcal{B}} + O(\mathbf{u}/\mu)] \\
&+ A_{\mathcal{N}}[(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)_{\mathcal{N}} + O(\mu\mathbf{u})] + O(\mathbf{u}) \\
=&-r_p + A(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e) + \{A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u})\}. \quad (3.15)
\end{aligned}$$

■

The right-hand side error can be divided into three parts. The first part $A_{\mathcal{B}}O(\mathbf{u}/\mu)$ is large and is in the range of $A_{\mathcal{B}}$; the second part $A_{\mathcal{N}}O(\mu\mathbf{u})$ is small and is located in the range of $A_{\mathcal{N}}$; the third part is a random error in the right-hand side with size $O(\mathbf{u})$.

3.2.2 The Structure of $AXZ^{-1}A^T$ and fl$(AXZ^{-1}A^T)$

Before we analyze the structure of $AXZ^{-1}A^T$, we present some related theorems.

Theorem 3.7 *Let $B \in \mathbb{C}^{m\times n}$ have singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ and let $C = AB$ have singular values $\tau_1 \geq \tau_2 \geq \ldots \geq \tau_n$. Then*

$$\tau_i \leq \sigma_i \|A\|_2, \qquad i = 1, \cdots, n.$$

(This is [91, Theorem I.4.5, p34].)

Theorem 3.8 (Weyl's Theorem) *Let A be a Hermitian matrix with eigenvalues*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,$$

and $\tilde{A} = A + E$ denote a Hermitian perturbation of A with eigenvalues

$$\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n.$$

Then

$$\max_i \{|\tilde{\lambda}_i - \lambda_i|\} \leq \|E\|_2 .$$

(This is [91, Corollary IV.4.10, p203].)

Theorem 3.9 *Let M denote a real symmetric matrix, and define the perturbed matrix $\tilde{M}$ as $M + E$, where E is symmetric. Consider an orthogonal matrix $[X_1\ X_2]$ where X_1 has l columns, such that* range(X_1) *is a simple invariant subspace of M, with*

$$\begin{bmatrix} X_1^T \\ X_2^T \end{bmatrix} M \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix} \quad \textit{and} \quad \begin{bmatrix} X_1^T \\ X_2^T \end{bmatrix} E \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix}.$$

Let $d_1 =$ sep$(L_1, L_2) - \|E_{11}\| - \|E_{22}\|$ and $v = \|E_{12}\|/d_1$, where sep$(L_1, L_2) = \min_{i,j} |\lambda_i(L_1) - \lambda_j(L_2)|$, *with $\lambda_k(\cdot)$ denoting the kth eigenvalue of its argument. If $d_1 > 0$ and $v < 1/2$, then*

1. *there are orthonormal bases* $\widetilde{X}_1$ *and* $\widetilde{X}_2$ *for simple invariant subspaces of the perturbed matrix* $\tilde{M}$ *satisfying* $\|X_1 - \widetilde{X}_1\| \leq 2v$ *and* $\|X_2 - \widetilde{X}_2\| \leq 2v$*;*

2. *for* $i = 1, \ldots, l$*, there is an eigenvalue* $\tilde{\omega}$ *of* $\tilde{M}$ *satisfying* $|\tilde{\omega} - \breve{\omega}_i| \leq 3\|E_{12}\|v$*, where* $\{\breve{\omega}_i\}$ *are the eigenvalues of* $X_1^T \tilde{M} X_1$*.*

(This is [111, Theorem 3.1]. It is a specialized version of [91, Theorem V.2.7, p236].)

For the complete definition of simple invariant subspaces, see [91, Definition V.1.2, p221]. Briefly, in Theorem 3.9, we say range(X_1) is a simple invariant subspace of M if range(MX_1) $\subset$ range(X_1) and the diagonal blocks L_1 and L_2 do not have any eigenvalues in common.

The following theorem is based on the work of M. Wright [111]. In that paper, she showed a similar result but for a matrix $AXZ^{-1}A^T + \Theta(1)$. This is also partially mentioned in [116, (5.10)]. The result illustrates the splitting of the eigenvalues of $AXZ^{-1}A^T$ into two parts of size $\Theta(1/\mu)$ and $\Theta(\mu)$.

Theorem 3.10 *Suppose that Assumption 3.2 (item 1, 2) holds. Let* $\hat{m}$ *denote the rank of* $A_{\mathcal{B}}$*;* $\lambda_1 \geq \cdots \geq \lambda_m$ *denote the (ordered) eigenvalues of* $AXZ^{-1}A^T$*; and* $[U_L \; U_S]$ *be an orthogonal matrix where the columns of* U_S *span the null space of* $A_{\mathcal{B}}^T$*.*
Then

1. *The* $\hat{m}$ *largest eigenvalues of* $AXZ^{-1}A^T$ *are* $\Theta(1/\mu)$*.*

2. *If* $\hat{m} < m$*, then each eigenvalue* $\lambda_{\hat{m}+k}$*,* $k = 1, \cdots, n - \hat{m}$*, differs at most by* $O(\mu)$ *from some eigenvalue of* $A_{\mathcal{B}} X_{\mathcal{B}} Z_{\mathcal{B}}^{-1} A_{\mathcal{B}}^T$ *and, in addition, it is* $\Theta(\mu)$*.*

3. $AXZ^{-1}A^T$ *has simple invariant subspaces close to those defined by* U_L *and* U_S *in the sense that there exist matrices* $\widetilde{U}_L$ *and* $\widetilde{U}_S$ *whose columns form orthonormal bases for simple invariant subspaces of* $AXZ^{-1}A^T$ *such that*

$$\|\widetilde{U}_L - U_L\| = O(\mu^2) \quad \textit{and} \quad \|\widetilde{U}_S - U_S\| = O(\mu^2).$$

Proof. We first observe that $X_{\mathcal{B}} Z_{\mathcal{B}}^{-1}$ is $\Theta(1/\mu)$ by (3.6) in Assumption 3.2 (p18). In addition, the assumption implies that $A_{\mathcal{B}}$ is $\Theta(1)$, which in turn yields

$$\|A_{\mathcal{B}} X_{\mathcal{B}} Z_{\mathcal{B}}^{-1} A_{\mathcal{B}}^T\| \leq \|A_B\|^2 \|X_{\mathcal{B}} Z_{\mathcal{B}}^{-1}\| = O(1/\mu).$$

So $\lambda_1(A_{\mathcal{B}}X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A_{\mathcal{B}}^T) = O(1/\mu)$. Let the matrix $X_{\mathcal{B}}^{1/2}Z_{\mathcal{B}}^{-1/2}A_{\mathcal{B}}^T$ be the matrix B in Theorem 3.7 and $X_{\mathcal{B}}^{-1/2}Z_{\mathcal{B}}^{1/2}$ be the matrix A in Theorem 3.7. Then, we can use Theorem 3.7 and Assumption 3.2 Item 1 (p18) to see that the $\hat{m}$ largest singular values of $X_{\mathcal{B}}^{1/2}Z_{\mathcal{B}}^{-1/2}A_{\mathcal{B}}^T$ are $\Theta(1/\sqrt{\mu})$. Thus $\lambda_{\hat{m}}(A_{\mathcal{B}}X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A_{\mathcal{B}}^T) = \Theta(1/\mu)$. Then part 1 follows by applying Theorem 3.8 in conjunction with the above bounds. (notice that $A_{\mathcal{N}}X_{\mathcal{N}}Z_{\mathcal{N}}^{-1}A_{\mathcal{N}}^T = O(\mu)$.)

The eigenvalue perturbation result Theorem 3.8, in conjunction with the above bounds shows that the eigenvalue $\lambda_{\hat{m}+k},\ k = 1, \cdots, n-\hat{m}$ differs at most by $O(\mu)$ from some eigenvalue of $A_{\mathcal{B}}X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A_{\mathcal{B}}^T$. Thus $\lambda_{\hat{m}+k}$ is $O(\mu)$. To show that $\lambda_{\hat{m}+k}$ is $\Theta(\mu)$, we need to show that $\lambda_m \geq C\mu$, for some constant C. Notice that with the assumption that A is full row rank, we get that $A^Ty \neq 0$ if $y \neq 0$, and that

$$\min_{\|y\|=1} \frac{y^TAXZ^{-1}A^Ty}{(y^TA)(A^Ty)} \geq \lambda_{\min}(XZ^{-1}) \geq C_1\mu,$$

where C_1 is some constant coefficient by our Assumption 3.2 (item 2). We now have

$$\lambda_{\min}(AXZ^{-1}A^T) = \min_{\|y\|=1} y^TAXZ^{-1}A^Ty \geq \lambda_{\min}(XZ^{-1}) \min_{\|y\|=1}(y^TAA^Ty) \geq C\mu,$$

where C is the smallest singular value of AA^T times C_1. Here we use Assumption 3.2 (item 1).

Part 3 is obtained by using Theorem 3.9 and the fact that $AXZ^{-1}A^T$ can be thought of as a perturbation of the matrix $A_{\mathcal{B}}X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A_{\mathcal{B}}^T$ by $A_{\mathcal{N}}X_{\mathcal{N}}Z_{\mathcal{N}}^{-1}A_{\mathcal{N}}^T$. ■

The error in $\mathrm{fl}(AXZ^{-1}A^T)$ can be bounded using the following. Since $A_{\mathcal{B}}X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A_{\mathcal{B}}^T$ is $O(1/\mu)$, we get

$$\begin{aligned}
\mathrm{fl}(AXZ^{-1}A^T) =& \mathrm{fl}(A_{\mathcal{B}}X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A_{\mathcal{B}}^T) + \mathrm{fl}(A_{\mathcal{N}}X_{\mathcal{N}}Z_{\mathcal{N}}^{-1}A_{\mathcal{N}}^T) + O(\mathbf{u}/\mu) \\
=& \mathrm{fl}(A_{\mathcal{B}})\mathrm{fl}(X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A^T) + \mathrm{fl}(A_{\mathcal{N}})\mathrm{fl}(X_{\mathcal{N}}Z_{\mathcal{N}}^{-1}A_{\mathcal{N}}^T) + O(\mathbf{u}/\mu) \\
=& A_{\mathcal{B}}(X_{\mathcal{B}}Z_{\mathcal{B}}^{-1}A_{\mathcal{B}}^T + O(\mathbf{u}/\mu)) \\
& + A_{\mathcal{N}}(X_{\mathcal{N}}Z_{\mathcal{N}}^{-1}A_{\mathcal{N}}^T + O(\mu\mathbf{u})) + O(\mathbf{u}/\mu) \\
=& AXZ^{-1}A^T + \{A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u}/\mu)\}. \qquad (3.16)
\end{aligned}$$

If we use the above error bound on $\mathrm{fl}(AXZ^{-1}A^T)$ and maintain $\mu \geq 10\sqrt{\mathbf{u}}$, we can extend the structure information in Theorem 3.10 to the matrix $\mathrm{fl}(AXZ^{-1}A^T)$.

Corollary 3.11 *Suppose that Assumption 3.2 (item 1, 2) holds and assume that* $\mu \geq 10\sqrt{\mathbf{u}}$. *Let* $\hat{m}$ *denote the rank of* $A_{\mathcal{B}}$ *and* $\{\hat{\lambda}_k\}$ *denote the eigenvalues of* $AXZ^{-1}A^T$ *such that* $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_m$. *Furthermore, let* $[U_L\ U_S]$ *be an orthogonal matrix, where the columns of* U_S *span the null space of* $A_{\mathcal{B}}^T$.
Then

1. *The* $\hat{m}$ *largest eigenvalues of fl*$(AXZ^{-1}A^T)$ *are* $\Theta(1/\mu)$.
2. *If* $\hat{m} < m$, *then every remaining eigenvalue* $\hat{\lambda}_{\hat{m}+k},\ k = 1, \cdots, n - \hat{m}$, *is* $\Theta(\mu)$.
3. *fl*$(AXZ^{-1}A^T)$ *has simple invariant subspaces close to those defined by* U_L *and* U_S *in the sense that there exist matrices* $\widehat{U}_L$ *and* $\widehat{U}_S$ *whose columns form orthonormal bases for simple invariant subspaces of fl*$(AXZ^{-1}A^T)$ *such that*

$$\|\widehat{U}_L - U_L\| = O(\mu^2) \quad \textit{and} \quad \|\widehat{U}_S - U_S\| = O(\mu^2).$$

Proof. Notice that when $\mu \geq 10\sqrt{\mathbf{u}}$, we have that $\mathbf{u}/\mu \leq \mu/100$. Thus by (3.16), $\mathrm{fl}(AXZ^{-1}A^T)$ is an $O(\mu/100)$ perturbation of $AXZ^{-1}A^T$. Using Theorem 3.8 and a similar proof to part 3 in Theorem 3.10 yields the results. ■

For the case that $\mathrm{rank}\,(A_{\mathcal{B}}) = m$, we get a stronger result that does not need the $\mu > 10\sqrt{\mathbf{u}}$ assumption.

Corollary 3.12 *Suppose that Assumption 3.2 (item 1, 2) holds and assume that* $\mathrm{rank}\,(A_{\mathcal{B}}) = m$ *and* $(A_{\mathcal{B}}A_{\mathcal{B}})^{-1} = \Theta(1)$. *Then The eigenvalues of fl*$(AXZ^{-1}A^T)$ *are* $\Theta(1/\mu)$, *i.e. fl*$(AXZ^{-1}A^T)$ *remains well-conditioned.*

Proof. By (3.16), we can see $\mathrm{fl}(AXZ^{-1}A^T)$ is a $O(\mathbf{u}/\mu)$ perturbation of $AXZ^{-1}A^T$. Thus by Theorem 3.8 we can derive the results. ■

The significance of Theorem 3.10 and Corollary 3.11 lies in that we obtain a block decomposition of $AXZ^{-1}A^T$ as follows.

$$AXZ^{-1}A^T = [\widetilde{U}_L\ \widetilde{U}_S] \begin{bmatrix} \Sigma_L & 0 \\ 0 & \Sigma_S \end{bmatrix} \begin{bmatrix} \widetilde{U}_L^T \\ \widetilde{U}_S^T \end{bmatrix}, \tag{3.17}$$

where Σ_L is a $\hat{m} \times \hat{m}$ submatrix (may not be diagonal), whose eigenvalues are the first $\hat{m}$ largest eigenvalues of $AXZ^{-1}A^T$; and Σ_S is a $(n-\hat{m}) \times (n-\hat{m})$ submatrix, whose eigenvalues are the remaining small eigenvalues of $AXZ^{-1}A^T$. Thus we have

$$\Sigma_L = \Theta(1/\mu),\ \Sigma_L^{-1} = \Theta(\mu), \quad \text{and} \quad \Sigma_S = \Theta(\mu),\ \Sigma_S^{-1} = \Theta(1/\mu). \tag{3.18}$$

Part 3 of Theorem 3.10 implies that

$$A_{\mathcal{B}}^T \widetilde{U}_S = O(\mu^2). \tag{3.19}$$

Similar results exist for $\mathrm{fl}(AXZ^{-1}A^T)$, i.e. we have

$$\mathrm{fl}(AXZ^{-1}A^T) = [\widehat{U}_L\ \widehat{U}_S] \begin{bmatrix} \widehat{\Sigma}_L & 0 \\ 0 & \widehat{\Sigma}_S \end{bmatrix} \begin{bmatrix} \widehat{U}_L^T \\ \widehat{U}_S^T \end{bmatrix}, \tag{3.20}$$

where $\widehat{\Sigma}_L$ is a $\hat{m} \times \hat{m}$ submatrix (may not be diagonal), whose eigenvalues are the first $\hat{m}$ largest eigenvalues of $\mathrm{fl}(AXZ^{-1}A^T)$; and $\widehat{\Sigma}_S$ is a $(n-\hat{m}) \times (n-\hat{m})$ submatrix, whose eigenvalues are the remaining small eigenvalues. Thus we have

$$\widehat{\Sigma}_L = \Theta(1/\mu),\ \widehat{\Sigma}_L^{-1} = \Theta(\mu), \quad \text{and} \quad \widehat{\Sigma}_S = \Theta(\mu),\ \widehat{\Sigma}_S^{-1} = \Theta(1/\mu). \tag{3.21}$$

Part 3 of Corollary 3.11 implies that

$$A_{\mathcal{B}}^T \widehat{U}_S = O(\mu^2). \tag{3.22}$$

Corollary 3.12 gives stronger result on the structure information without the assumption of $\mu > 10\sqrt{\mathbf{u}}$. This corollary can be applied to the non-degenerate case and the degenerate case with $\mathrm{rank}\,(A_{\mathcal{B}}) = m$ and $|\mathcal{B}| > m$, where we are able to prove our results without the assumption $\mu > 10\sqrt{\mathbf{u}}$.

3.3 Non-Degenerate Case

3.3.1 Estimating the Magnitudes of dx, dy, dz

Theorem 3.13 *Suppose that Assumptions 3.2 and 3.3 hold. Let (dx, dy, dz) be the* **exact** *solution of the original system* (3.2) *(equivalently the* **exact** *solution of* (3.4)*). Then*

$$(dx, dy, dz) = O(\mu).$$

Proof. See [115]. We provide an alternative proof here using our structure analysis of $AXZ^{-1}A^T$. Notice that the right-hand side of the second block of (3.4) is $O(1)$. Then by using (3.17) (p25) and the non-degeneracy assumption (Assumption 3.3), we have

$$\begin{aligned} dy &= (AXZ^{-1}A^T)^{-1}O(1) \\ &= [\widetilde{U}_L \ \widetilde{U}_S] \begin{bmatrix} \Sigma_L^{-1} & 0 \\ 0 & \Sigma_S^{-1} \end{bmatrix} \begin{bmatrix} \widetilde{U}_L^T \\ \widetilde{U}_S^T \end{bmatrix} O(1) \\ &= \widetilde{U}_L \Sigma_L^{-1} \widetilde{U}_L^T O(1) = O(\mu). \end{aligned}$$

We then can see that $dz = O(\mu)$ follows from $dz = -A^T dy - r_d$; and also that $dx_{\mathcal{N}} = O(\mu)$ follows from $Z_{\mathcal{N}} dx_{\mathcal{N}} + X_{\mathcal{N}} dz_{\mathcal{N}} = (-ZXe + \sigma\mu e)_{\mathcal{N}}$. Then we have $dx_{\mathcal{B}} = O(\mu)$ from $A_{\mathcal{B}} dx_{\mathcal{B}} + A_{\mathcal{N}} dx_{\mathcal{N}} = -r_p$ and the non-degeneracy assumption (i.e., $A_{\mathcal{B}}$ is invertible and well-conditioned).

■

3.3.2 Error in fl(dy)

We state a slightly modified version of [101] and [49, p133].

Lemma 3.14 *Let*

$$Mx = b, \quad \textit{and} \quad (M + \Delta M)\tilde{x} = b + \Delta b.$$

Assume $M + \Delta M$ is nonsingular. Then

$$\tilde{x} - x = (M + \Delta M)^{-1}(\Delta b - \Delta M x).$$

Proof. Notice that $(M + \Delta M)(\tilde{x} - x) = \Delta b - \Delta M x$. ■

Theorem 3.15 *Suppose Assumption 3.2 and 3.3 hold. Let dy be the exact solution of the middle block of* (3.4) *(p16). Let fl(dy) be the computed solution by any backward stable linear equation solver. Then*

$$fl(dy) - dy = O(\mathbf{u}).$$

Proof. Assume $\text{fl}(dy)$ is a solution which comes from a backward stable linear system. This means

$$\text{fl}(AXZ^{-1}A^T)\text{fl}(dy) = \text{fl}(-r_p + A(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e)) + O(\mathbf{u}). \tag{3.23}$$

The $O(\mathbf{u})$ term can be folded into the argument of the $\text{fl}(\cdot)$ operator because the argument is $O(1)$. Now, using Lemma 3.14, the error bound for the right-hand side (Theorem 3.6 (p20)), and Corollary 3.12, we have

$$\begin{aligned}
&\text{fl}(dy) - dy \\
=&\text{fl}(AXZ^{-1}A^T)^{-1}[O(\mathbf{u}/\mu) - (\text{fl}(AXZ^{-1}A^T) - AXZ^{-1}A^T)dy] \\
=&\widehat{U}_L\widehat{\Sigma}_L^{-1}\widehat{U}_L^T[O(\mathbf{u}/\mu) - (A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u}/\mu))dy] \qquad \text{(by (3.16), (3.20))} \\
=&O(\mathbf{u}),
\end{aligned}$$

where the last step follows from $\widehat{\Sigma}^{-1} = \Theta(\mu)$. ■

Notice that $dy = O(\mu)$. In addition, when $\mu > \mathbf{u}$, Theorem 3.15 means that $\text{fl}(dy)$ is also $O(\mu)$.

3.3.3 Error in $\text{fl}(dx)$

Theorem 3.16 *Suppose that Assumptions 3.2 and 3.3 hold. Let dx be the exact solution obtained from the back-substitution using dy and the third equation of (3.4). Let $fl(dx)$ be the floating point computed solution from the back-substitution with $fl(dy)$ and the third equation of (3.4). If $fl(dy)$ has the error bound in Theorem 3.15, then*

$$fl(dx_i) = dx_i + O(\mathbf{u}) \ \ (i \in \mathcal{B}), \qquad fl(dx_i) = dx_i + O(\mu\mathbf{u}) \ \ (i \in \mathcal{N}).$$

Proof. Notice that the equation for solving dy is:

$$AZ^{-1}XA^Tdy = -r_p + A(-Z^{-1}Xr_d + x - \sigma\mu Z^{-1}e). \tag{3.24}$$

The $Ax - r_p$ term in the right-hand side of (3.24) is equal to b. Thus,

$$AZ^{-1}XA^Tdy = b + A(-Z^{-1}Xr_d - \sigma\mu Z^{-1}e). \tag{3.25}$$

We split this term according to the partition of indices, $\mathcal{B}, \mathcal{N}$, i.e.

$$A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} A_{\mathcal{B}}^T dy + A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} A_{\mathcal{N}}^T dy = b - A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} r_{d\mathcal{B}} - A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} r_{d\mathcal{N}} - \sigma\mu A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} e - \sigma\mu A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} e.$$

Now, move the parts associated with $\mathcal{B}$ to one side.

$$\begin{aligned} A_{\mathcal{B}}(Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} A_{\mathcal{B}}^T dy + Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} r_{d\mathcal{B}} + \sigma\mu Z_{\mathcal{B}}^{-1} e) \\ = b - A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} r_{d\mathcal{N}} - \sigma\mu A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} e - A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} A_{\mathcal{N}}^T dy. \end{aligned} \quad (3.26)$$

Similar to (3.25), our computed solution $\mathrm{fl}(dy)$, from a backward stable linear solver, satisfies the following equality

$$\mathrm{fl}(AXZ^{-1}A^T)\mathrm{fl}(dy) = \mathrm{fl}(b + A(-Z^{-1}Xr_d - \sigma\mu Z^{-1}e) + O(\mathbf{u}).$$

We now follow the same procedure from (3.25) to (3.26). We first do the split according to the partition with indices $\mathcal{B}, \mathcal{N}$. The $O(\cdot)$ item is added to represent the roundoff error in the floating point operation.

$$\begin{aligned} &[\mathrm{fl}(A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} A_{\mathcal{B}}^T) + \mathrm{fl}(A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} A_{\mathcal{N}}^T) + O(\mathbf{u}/\mu)]\mathrm{fl}(dy) \\ &\quad = b - \mathrm{fl}(A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} r_{d\mathcal{B}}) - \mathrm{fl}(A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} r_{d\mathcal{N}}) - \mathrm{fl}(\sigma\mu A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} e) - \mathrm{fl}(\sigma\mu A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} e) + O(\mathbf{u}). \end{aligned} \quad (3.27)$$

Now, move the parts associated with $\mathcal{B}$ to one side and combine all the error terms. (Notice that $O(\mathbf{u}/\mu)\mathrm{fl}(dy) = O(\mathbf{u})$.) We get

$$\begin{aligned} &\mathrm{fl}(A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} A_{\mathcal{B}}^T)\mathrm{fl}(dy) + \mathrm{fl}(A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} r_{d\mathcal{B}}) + \mathrm{fl}(\sigma\mu A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} e) \\ &\qquad = b - \mathrm{fl}(A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} r_{d\mathcal{N}}) - \mathrm{fl}(\sigma\mu A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} e) - \mathrm{fl}(A_{\mathcal{N}} Z_{\mathcal{N}}^{-1} X_{\mathcal{N}} A_{\mathcal{N}}^T)\mathrm{fl}(dy) + O(\mathbf{u}). \end{aligned} \quad (3.28)$$

By factoring out $A_{\mathcal{B}}$, we rewrite the left-hand side.

$$\begin{aligned} &\mathrm{fl}(A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} A_{\mathcal{B}}^T)\mathrm{fl}(dy) + \mathrm{fl}(A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} r_{d\mathcal{B}}) + \mathrm{fl}(\sigma\mu A_{\mathcal{B}} Z_{\mathcal{B}}^{-1} e) \\ &= [A_{\mathcal{B}}\mathrm{fl}(Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} A_{\mathcal{B}}^T) + O(\mathbf{u}/\mu)]\mathrm{fl}(dy) + [A_{\mathcal{B}}\mathrm{fl}(Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} r_{d\mathcal{B}}) + O(\mathbf{u})] + [A_{\mathcal{B}}\mathrm{fl}(\sigma\mu Z_{\mathcal{B}}^{-1} e) + O(\mathbf{u})]. \end{aligned} \quad (3.29)$$

We can see from the above equation that all the error terms are $O(\mathbf{u})$ (as $O(\mathbf{u}/\mu)\text{fl}(dy) = O(\mathbf{u})$). So, we can rewrite (3.28) as

$$\begin{aligned} &A_{\mathcal{B}}[\text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T)\text{fl}(dy) + \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}}) + \text{fl}(\sigma\mu Z_{\mathcal{B}}^{-1}e)] \\ &\quad = b - \text{fl}(A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}r_{d\mathcal{N}}) - \text{fl}(\sigma\mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e) - \text{fl}(A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}A_{\mathcal{N}}^T)\text{fl}(dy) + O(\mathbf{u}). \quad (3.30) \end{aligned}$$

Now if we take the difference of (3.26) and (3.30), we have

$$\begin{aligned} & A_{\mathcal{B}}[\text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T\text{fl}(dy)) + \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}}) + \text{fl}(\sigma\mu Z_{\mathcal{B}}^{-1}e) \\ &\quad - (Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T dy + Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}} + \sigma\mu Z_{\mathcal{B}}^{-1}e)] \\ = \;& [A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}r_{d\mathcal{N}} + \sigma\mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e + A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}A_{\mathcal{N}}^T dy \quad (3.31) \\ & -\text{fl}(A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}r_{d\mathcal{N}}) - \text{fl}(\sigma\mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e) - \text{fl}(A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}A_{\mathcal{N}}^T)\text{fl}(dy) + O(\mathbf{u})]. \end{aligned}$$

Since each item of $A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}r_{d\mathcal{N}}$, $\sigma\mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e$, $A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}A_{\mathcal{N}}^T dy$ in the right-hand side of (3.31) is $O(1)$ and the right-hand side is the sum of the roundoff errors of these terms, we conclude that the right-hand side is at most $O(\mathbf{u})$. Thus the above equation (3.31) can be written as

$$\begin{aligned} & A_{\mathcal{B}}[\text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T\text{fl}(dy)) + \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}}) + \text{fl}(\sigma\mu Z_{\mathcal{B}}^{-1}e) \\ &\quad - (Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T dy + Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}} + \sigma\mu Z_{\mathcal{B}}^{-1}e)] \\ &= O(\mathbf{u}). \quad (3.32) \end{aligned}$$

By the non-degeneracy assumption (Assumption 3.3) that $A_{\mathcal{B}}$ is non-singular and well conditioned, we have that

$$\begin{aligned} & \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T\text{fl}(dy)) + \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}}) + \text{fl}(\sigma\mu Z_{\mathcal{B}}^{-1}e) \\ &\quad - (Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T dy + Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}} + \sigma\mu Z_{\mathcal{B}}^{-1}e) \\ &= A_{\mathcal{B}}^{-1}O(\mathbf{u}) = O(\mathbf{u}). \quad (3.33) \end{aligned}$$

Moreover, using Assumption 3.1, Item 2 (p17), that $\text{fl}(x_{\mathcal{B}}) = x_{\mathcal{B}}$, we see that

$$\begin{aligned} \text{fl}(dx_{\mathcal{B}}) &= \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T)\text{fl}(dy) + \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}}) + \text{fl}(\sigma\mu Z_{\mathcal{B}}^{-1}e) - \text{fl}(x_{\mathcal{B}}) + O(\mathbf{u}) \\ &= Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}A_{\mathcal{B}}^T dy + Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}r_{d\mathcal{B}} + \sigma\mu Z_{\mathcal{B}}^{-1}e - x_{\mathcal{B}} + O(\mathbf{u}), \quad \text{by (3.33)} \\ &= dx_{\mathcal{B}} + O(\mathbf{u}). \end{aligned}$$

If index $i \in \mathcal{N}$, we have

$$\begin{aligned}
\text{fl}(dx_i) =& \text{fl}(z_i^{-1}x_i(A^T\text{fl}(dy))_i) + \text{fl}(z_i^{-1}x_i(r_d)_i) - x_i + \text{fl}(\sigma\mu z_i^{-1}) + O(\mu\mathbf{u}) \\
=& [(z_i^{-1}x_i + O(\mu\mathbf{u})][(A^T dy)_i + O(\mathbf{u})] + [z_i^{-1}x_i + O(\mu\mathbf{u})][(r_d)_i + O(\mathbf{u})] \\
& - x_i + [\sigma\mu(z_i^{-1} + O(\mathbf{u})) + O(\mu\mathbf{u})] + O(\mu\mathbf{u}) \\
=& z_i^{-1}x_i(A^T dy)_i + O(\mu\mathbf{u}) + z_i^{-1}x_i(r_d)_i + O(\mu\mathbf{u}) - x_i + \sigma\mu z_i^{-1} + O(\mu\mathbf{u}) \\
=& dx_i + O(\mu\mathbf{u}). \qquad (3.34)
\end{aligned}$$

■

3.3.4 Error in fl(dz)

We use two equations to back-solve for dz. One is with $A^T dy + dz = -r_d$, the first equation of (3.4) or (3.2). The other one is with $Zdx + Xdz = -ZX + \sigma\mu e$, the third equation of (3.2). The error bounds on fl(dz) using these two approaches are the same.

Theorem 3.17 *Suppose Assumptions 3.2 and 3.3 hold. Let dz be the exact solution obtained from a back-solve with dx using $dz = X^{-1}[-ZX + \sigma\mu e - Zdx]$, the third equation of (3.2). Let $fl(dz) = fl(X^{-1}[-ZX + \sigma\mu e - Zfl(dx)])$ be the floating pointing computed solution of dz, where $fl(dx)$ has the error bound in Theorem 3.16. Then*

$$fl(dz_i) = dz_i + O(\mu\mathbf{u}) \ \ (i \in \mathcal{B}), \qquad fl(dz_i) = dz_i + O(\mathbf{u}) \ \ (i \in \mathcal{N}).$$

Proof. The proof follows directly from the proof of the augmented system in [115]. (It also follows from a standard error analysis argument on each arithmetic operation.) ■

Theorem 3.18 *Suppose Assumptions 3.2 and 3.3 hold. Let dz be the exact solution obtained from a back-solve with dy using $dz = -A^T dy - r_d$, the first equation of (3.4). Let $fl(dz) = fl(-r_d - A^T fl(dy))$ be the floating point computed solution of dz, where $fl(dy)$ has the error bound in Theorem 3.15. Then*

$$fl(dz_i) = dz_i + O(\mu\mathbf{u}) \ \ (i \in \mathcal{B}), \qquad fl(dz_i) = dz_i + O(\mathbf{u}) \ \ (i \in \mathcal{N}).$$

Proof. By using the fact $dy = O(\mu)$, we have

$$\begin{aligned}
\text{fl}(dz) =& \text{fl}(-r_d - A^T dy) \\
=& -\text{fl}(r_d) - [\text{fl}(A^T)\text{fl}(dy) + O(\mu\mathbf{u})] + O(\mu\mathbf{u}) \\
=& -(r_d + O(\mathbf{u})) - [A^T(dy + O(\mathbf{u})) + O(\mu\mathbf{u})] + O(\mu\mathbf{u}) \\
=& -r_d - A^T dy + O(\mathbf{u}) \qquad (O(\mu\mathbf{u}) \text{ folded into } O(\mathbf{u})) \\
=& dz + O(\mathbf{u}). \qquad (3.35)
\end{aligned}$$

We now show the bound for index $i \in \mathcal{B}$. By using the second equation in (3.4) we get

$$AZ^{-1}XA^T dy = -r_p - AZ^{-1}Xr_d + Ax - \sigma\mu AZ^{-1}e. \tag{3.36}$$

Equating the $Ax - r_p$ term to b and moving $-AZ^{-1}Xr_d$ to the left-hand side, we have

$$AZ^{-1}X(r_d + A^T dy) = b - \sigma\mu AZ^{-1}e.$$

We split the left-hand side according to the partition of indices, $\mathcal{B}, \mathcal{N}$, i.e.,

$$A_{\mathcal{B}}Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}(r_d + A^T dy)_{\mathcal{B}} + A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}X_{\mathcal{N}}(r_d + A^T dy)_{\mathcal{N}} = b - \sigma\mu AZ^{-1}e. \tag{3.37}$$

Rearranging, we get

$$A_{\mathcal{B}}Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}(r_d + A^T dy)_{\mathcal{B}} = b - \sigma\mu AZ^{-1}e - A_{\mathcal{N}}X_{\mathcal{N}}Z_{\mathcal{N}}^{-1}(r_d + A^T dy)_{\mathcal{N}}. \tag{3.38}$$

For the floating point computation, we have similar equations. Notice that for a backward stable system, the floating point computed solution of $\text{fl}(dy)$ satisfies the following equation (similar to (3.36)).

$$\text{fl}(AZ^{-1}XA^T)\text{fl}(dy) = \text{fl}(b - AZ^{-1}Xr_d - \sigma\mu AZ^{-1}e) + O(\mathbf{u})$$

This implies

$$[\text{fl}(AZ^{-1}X)A^T(1 + O(\mathbf{u}))]\text{fl}(dy) = \text{fl}(b - \sigma\mu AZ^{-1}e) - \text{fl}(AZ^{-1}X)\text{fl}(r_d) + O(\mathbf{u}).$$

Rearranging again, we get

$$\text{fl}(AZ^{-1}X)(A^T\text{fl}(dy) + \text{fl}(r_d)) = \text{fl}(b - \sigma\mu AZ^{-1}e) + O(\mathbf{u}).$$

Now, split the indices according to the partition of $\mathcal{B}$ and $\mathcal{N}$.

$$\begin{aligned}\mathrm{fl}((AZ^{-1}X)_{\mathcal{B}})(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{B}}+\mathrm{fl}((AZ^{-1}X)_{\mathcal{N}})(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{N}}+O(\mathbf{u})\\ =\mathrm{fl}(b-\sigma\mu AZ^{-1}e)+O(\mathbf{u}).\end{aligned}$$

Rearrange:

$$\begin{aligned}&\mathrm{fl}((AZ^{-1}X)_{\mathcal{B}})(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{B}}\\ &\qquad=\mathrm{fl}(b-\sigma\mu AZ^{-1}e)-\mathrm{fl}((AZ^{-1}X)_{\mathcal{N}})(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{N}}+O(\mathbf{u}).\end{aligned}\tag{3.39}$$

Now using the definition of $\mathrm{fl}(\cdot)$, we can see that

$$\begin{aligned}\mathrm{fl}((AZ^{-1}X)_{\mathcal{B}})&=\mathrm{fl}(A_{\mathcal{B}})\mathrm{fl}((Z^{-1}X)_{\mathcal{B}})+O(\mathbf{u}/\mu)\\ &=\mathrm{fl}(A_{\mathcal{B}})((Z^{-1}+O(\mathbf{u}/\mu))X+O(\mathbf{u}/\mu))+O(\mathbf{u}/\mu)\\ &=A_{\mathcal{B}}Z^{-1}X+O(\mathbf{u}/\mu).\end{aligned}$$

Then, we substitute this error estimate into (3.39) and obtain

$$\begin{aligned}&[(AZ^{-1}X)_{\mathcal{B}}+O(\mathbf{u}/\mu)](A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{B}}\\ &\qquad=\mathrm{fl}(b-\sigma\mu AZ^{-1}e)-\mathrm{fl}((AZ^{-1}X)_{\mathcal{N}})(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{N}}+O(\mathbf{u}).\end{aligned}$$

Since the term $(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{B}}$ is $O(\mu)$, the error term $O(\mathbf{u}/\mu)(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{B}}$ is $O(\mathbf{u})$. The above equation implies that

$$\begin{aligned}&(AZ^{-1}X)_{\mathcal{B}}(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{B}}\\ &\qquad=\mathrm{fl}(b-\sigma\mu AZ^{-1}e)-\mathrm{fl}((AZ^{-1}X)_{\mathcal{N}})(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{N}}+O(\mathbf{u}).\end{aligned}\tag{3.40}$$

Now, by taking the difference of (3.40) and (3.38), we have

$$\begin{aligned}&(AZ^{-1}X)_{\mathcal{B}}(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{B}}-(A^Tdy+r_d)_{\mathcal{B}})\\ &\qquad=\mathrm{fl}(b-\sigma\mu AZ^{-1}e)-\mathrm{fl}((AZ^{-1}X)_{\mathcal{N}})(A^T\mathrm{fl}(dy)+\mathrm{fl}(r_d))_{\mathcal{N}}+O(\mathbf{u})\\ &\qquad\qquad-[b-\sigma\mu AZ^{-1}e-(AZ^{-1}X)_{\mathcal{N}}(A^Tdy+r_d)_{\mathcal{N}}].\end{aligned}\tag{3.41}$$

Since each term of the right-hand side of (3.41) is $O(1)$, after the cancellation, the right-hand side is $O(\mathbf{u})$. Thus

$$\begin{aligned}(A^T \mathrm{fl}(dy) + \mathrm{fl}(r_d))_{\mathcal{B}} - (A^T dy + r_d)_{\mathcal{B}}) &= (AZ^{-1}X)_{\mathcal{B}}^{-1} O(\mathbf{u}) \\ &= O(\mu\mathbf{u}), \end{aligned} \tag{3.42}$$

which is

$$\mathrm{fl}(dz_{\mathcal{B}}) - dz_{\mathcal{B}} = O(\mu\mathbf{u}).$$

■

3.3.5 The Maximal Step Length α

The following theorem [115, Theorem 4.1] shows that interior point methods progress well (i.e. the maximal step length is approximately 1 when μ is sufficiently small.) The theorem also shows that the maximal step length calculated from $\mathrm{fl}(dx)$ and $\mathrm{fl}(dz)$ only has an error of $O(\mathbf{u})$ compared to the exact one calculated from exact dx and dz.

Theorem 3.19 *Suppose that Assumption 3.2 holds. Let (dx, dy, dz) be the exact solution of (3.2) (equivalently, (3.4)), and let $(\widehat{dx}, \widehat{dy}, \widehat{dz})$ be an approximation to this step. Suppose that the centering parameter σ in (3.2) lies in the range $[0, 1/2]$ and that the following conditions hold:*

$$(dx, dz) = O(\mu), \tag{3.43}$$

$$(dx_{\mathcal{B}}, dz_{\mathcal{N}}) - (\widehat{dx_{\mathcal{B}}}, \widehat{dz_{\mathcal{N}}}) = O(\mathbf{u}), \tag{3.44}$$

$$(dx_{\mathcal{N}}, dz_{\mathcal{B}}) - (\widehat{dx_{\mathcal{N}}}, \widehat{dz_{\mathcal{B}}}) = O(\mu\mathbf{u}). \tag{3.45}$$

Let α^* denote the largest number in $[0, 1]$ such that

$$(x + \alpha dx, z + \alpha dz) \geq 0 \quad \textit{for all } \alpha \in [0,\ \alpha^*]; \tag{3.46}$$

$$(x + \alpha dx)^T (z + \alpha dz) \textit{ is decreasing for all } \alpha \in [0,\ \alpha^*]. \tag{3.47}$$

Suppose $\hat{\alpha}^*$ is obtained by replacing (dx, dz) with $(\widehat{dx}, \widehat{dz})$ in (3.46) and (3.47). Then for all μ sufficiently small, we have

$$\begin{aligned} 1 - \alpha^* &= O(\mu), && (3.48) \\ \hat{\alpha}^* = \alpha^* + O(\mathbf{u}) &= 1 - O(\mu) + O(\mathbf{u}), && (3.49) \\ (x + \hat{\alpha}^* \widehat{dx})^T (z + \hat{\alpha}^* \widehat{dz})/n &= \sigma O(\mu) + O(\mu(\mu + \mathbf{u})). && (3.50) \end{aligned}$$

S. Wright [115] uses the above theorem to show that the augmented system in LP, under a non-degeneracy assumption, can have close to 1 step lengths at the final stage of interior point methods. Thus, the roundoff error is not a problem for the augmented system. Our error bounds on fl(dx), fl(dz) are the same as those from (3.43) to (3.45). Thus, this theorem can be applied to our analysis without modification. We also expect the normal equation system to have a close to 1 step length at the final stage of interior point methods for non-degenerate (specified by Assumption 3.3) problems where Assumption 3.2 holds. This can happen even when the condition number for the left-hand side of the normal equation system, (3.4), can go to infinity, see [41]. The step length $\hat{\alpha}^*$ computed using fl(dx) and fl(dz) has an error of $O(\mathbf{u})$ compared to the exact α.

3.3.6 Numerical Example for The Non-Degenerate Case

The following example illustrates that our error estimates are tight on the computed search direction.

Example 3.20 *The data A and an optimal solution x^*, y^*, and z^* of the LP problem are given below:*

$$A = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 2 & 2 & 4 & 1 \end{bmatrix}, \quad x^* = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad y^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad z^* = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}. \tag{3.51}$$

The data b, c is defined by $Ax^* = b$ and $A^T y^* + z^* = c$. And the partition of the indices is

$\mathcal{B} = \{1, 2\}$, *and* $\mathcal{N} = \{3, 4\}$. *We let the initial* x, y, *and* z *be*

$$x = \begin{bmatrix} 1.0002568 \\ 0.99981378 \\ 3.9374932e{-}4 \\ 1.634266e{-}4 \end{bmatrix}, \quad y = \begin{bmatrix} 1.00005026 \\ 1.16595e-4 \end{bmatrix}, \quad z = \begin{bmatrix} 1.9454628e{-}4 \\ 1.398727e{-}4 \\ 1.0001686 \\ 1.0001916 \end{bmatrix}.$$

We check the duality gap and the residuals

$$\mu = 2.2292914e{-}004, \quad r_p = \begin{bmatrix} 1.0442986e{-}003 \\ 1.8795839e{-}003 \end{bmatrix}, \quad r_d = \begin{bmatrix} 4.7799628e{-}004 \\ 3.7306273e{-}004 \\ 7.3550000e{-}004 \\ 3.0819500e{-}004 \end{bmatrix}.$$

This data satisfies Assumption 3.2 and $A_{\mathcal{B}}$ *satisfies Assumption 3.3.*

We use double precision to solve for dy, dx, dz *and assume this is the accurate solution. We then simulate the* $fl(\cdot)$ *operation by keeping the* $-\log(\mathbf{u})$ *most significant digits through a roundoff computation after each arithmetic operation. So, it can be thought of as having an error of size* $\mathbf{u}$. *In Table 3.1, we list the error for* $fl(dx)$, $fl(dy)$, *and* $fl(dz)$ *at different* $\mathbf{u}$ *values. We see that the error bound is consistent with Theorems 3.15, 3.16, and 3.18 outlined in this section.*

3.4 The Degenerate Case with $\operatorname{rank}(A_{\mathcal{B}}) < m$

For degenerate problems, our error bounds on $\text{fl}(dx)$, $\text{fl}(dy)$, and $\text{fl}(dz)$ in the previous section can fail. First, it is generally not true that $dy = O(\mu)$ for the degenerate case. Second, the proof of the error bounds for $\text{fl}(dx)$ and $\text{fl}(dz)$ uses the property that $A_{\mathcal{B}}$ is invertible. This is not true in the degenerate case.

But in practice, surprisingly, degeneracy seldom causes serious problems. We explain this in the following discussion. In this section, we assume that the rank of $A_{\mathcal{B}}$ is less than m and $\mu > 10\sqrt{\mathbf{u}}$.

We first state a lemma on the bound of the magnitude of dx, dy, dz from [116].

	$\mathbf{u} = 1e{-}7$	$\mathbf{u} = 1e{-}8$	$\mathbf{u} = 1e{-}9$	$\mathbf{u} = 1e{-}10$	$\mathbf{u} = 1e{-}11$	$\mathbf{u} = 1e{-}12$
$\lvert dy - \mathrm{fl}(dy)\rvert_i$:	$3.6e{-}9$	$3.7e{-}9$	$3.7e{-}9$	$2.8e{-}10$	$2.0e{-}11$	$6.3e{-}16$
($\lVert dy\rVert$=1.3e−4)	$1.2e{-}10$	$1.5e{-}11$	$8.8e{-}13$	$1.9e{-}14$	$1.1e{-}14$	$5.8e{-}16$
	$8.7e{-}7$	$3.3e{-}8$	$7.3e{-}9$	$3.3e{-}10$	$7.1e{-}11$	$1.2e{-}12$
$\lvert dx - \mathrm{fl}(dx)\rvert_i$:	$2.0e{-}6$	$1.8e{-}7$	$4.3e{-}9$	$6.5e{-}10$	$1.5e{-}10$	$5.7e{-}12$
$\left(\begin{smallmatrix}\lVert dx_{\mathcal{B}}\rVert=3.2e-4\\ \lVert dx_{\mathcal{N}}\rVert=4.3e-4\end{smallmatrix}\right)$	$1.8e{-}10$	$5.6e{-}12$	$2.6e{-}12$	$2.0e{-}13$	$1.8e{-}14$	$7.6e{-}17$
	$8.7e{-}12$	$1.3e{-}12$	$6.7e{-}13$	$3.4e{-}14$	$3.6e{-}15$	$4.3e{-}16$
	$2.1e{-}10$	$7.5e{-}12$	$1.5e{-}12$	$3.8e{-}14$	$1.2e{-}14$	$5.4e{-}16$
$\lvert dz - \mathrm{fl}(dz)\rvert_i$:	$2.8e{-}10$	$3.0e{-}11$	$1.8e{-}12$	$3.9e{-}14$	$2.1e{-}14$	$1.2e{-}15$
$\left(\begin{smallmatrix}\lVert dz_{\mathcal{B}}\rVert=2.4e-4\\ \lVert dz_{\mathcal{N}}\rVert=2.6e-4\end{smallmatrix}\right)$	$4.9e{-}7$	$7.5e{-}9$	$7.4e{-}9$	$5.6e{-}10$	$4.0e{-}11$	$1.1e{-}15$
	$2.0e{-}7$	$5.0e{-}9$	$5.0e{-}9$	$2.0e{-}14$	$1.0e{-}14$	$4.8e{-}16$
$\lvert \alpha - \mathrm{fl}(\alpha)\rvert$: (α=1.0)	$2.2e{-}6$	$2.4e{-}7$	$9.0e{-}9$	$2.7e{-}11$	$1.3e{-}10$	$7.1e{-}12$

Table 3.1: The error in $\mathrm{fl}(dx), \mathrm{fl}(dy)$, $\mathrm{fl}(dz)$, and $\mathrm{fl}(\alpha)$ for different $\mathbf{u}$ for the data in Example 3.20, where $\mathrm{fl}(\alpha)$ is the largest number (≤ 1) such that $(x + \mathrm{fl}(\alpha)\mathrm{fl}(x), z + \mathrm{fl}(\alpha)\mathrm{fl}(z)) \geq 0$, and $\sigma = 0$ in (3.2) (p15). Here $\mathcal{B} = \{1, 2\}$ and $\mathcal{N} = \{3, 4\}$.

Lemma 3.21 *Suppose Assumption 3.2 holds. Let dx, dy, and dz be the solution of*

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -ZXe + w \end{bmatrix}, \tag{3.52}$$

where $w = O(\mu^2)$. Then

$$(dx, dy, dz) = O(\mu).$$

Proof. See [116, sect. 5.1]. ■

However, the estimates for the magnitudes are different for the case of a centering direction, as shown in the following lemma.

Lemma 3.22 *Suppose Assumption 3.2 holds. Let dx, dy, and dz be the solution of*

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -ZXe + \mu e \end{bmatrix}. \tag{3.53}$$

Then

$$dy = O(1),$$

$$dx_{\mathcal{B}} = O(1), \quad dx_{\mathcal{N}} = O(\mu), \text{ and } dz_{\mathcal{B}} = O(\mu), \quad dz_{\mathcal{N}} = O(1).$$

Proof. The direction (dx, dy, dz) can be split into an affine scaling component (dx^{aff}, dy^{aff}, dz^{aff}) (satisfying (3.53) without the μe component in the right-hand side) and a component $(dx^{\mu}, dy^{\mu}, dz^{\mu})$ that satisfies

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx^{\mu} \\ dy^{\mu} \\ dz^{\mu} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e \end{bmatrix}. \tag{3.54}$$

It is shown in [116, sect. 5.1] that

$$(dx^{\text{aff}}, dy^{\text{aff}}, dz^{\text{aff}}) = O(\mu). \tag{3.55}$$

We notice that dy^μ also satisfies the following equation by a block elimination on (3.54)

$$AXZ^{-1}A^T dy^\mu = -\mu AZ^{-1}e.$$

Using the structure information of $AXZ^{-1}A^T$ in (3.17) (p25), we have

$$\begin{aligned} dy^\mu &= -(AXZ^{-1}A^T)^{-1}\mu AZ^{-1}e \\ &= -[\widetilde{U}_L\ \widetilde{U}_S]\begin{bmatrix}\Sigma_L^{-1} & 0\\ 0 & \Sigma_S^{-1}\end{bmatrix}\begin{bmatrix}\widetilde{U}_L^T\\ \widetilde{U}_S^T\end{bmatrix}(\mu A_{\mathcal{B}}Z_{\mathcal{B}}^{-1}e + \mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e) \\ &= -\widetilde{U}_L\Sigma_L^{-1}\widetilde{U}_L^T(\mu A_{\mathcal{B}}Z_{\mathcal{B}}^{-1}e + \mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e) - \underline{\widetilde{U}_S\Sigma_S^{-1}\widetilde{U}_S^T}(\mu A_{\mathcal{B}}Z_{\mathcal{B}}^{-1}e + \underline{\mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e}). \end{aligned} \tag{3.56}$$

From (3.18), (3.19) (p26) and Assumption 3.2 (p18), we can see that the underlined part in (3.56) is the dominant part with size $O(1)$. So $dy^\mu = O(1)$. Since $dy = dy^{\text{aff}} + dy^\mu$, we see that $dy = O(1)$.

Since $dy = O(1)$, we see that $dz = O(1)$ from $dz = -A^T dy - r_d$. Notice that from (3.56), we have

$$\begin{aligned} A_{\mathcal{B}}^T dy^\mu &= -\underline{A_{\mathcal{B}}^T\widetilde{U}_L\Sigma_L^{-1}\widetilde{U}_L^T}(\underline{\mu A_{\mathcal{B}}Z_{\mathcal{B}}^{-1}e} + \mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e) - A_{\mathcal{B}}^T\widetilde{U}_S\Sigma_S^{-1}\widetilde{U}_S^T(\mu A_{\mathcal{B}}Z_{\mathcal{B}}^{-1}e + \mu A_{\mathcal{N}}Z_{\mathcal{N}}^{-1}e) \\ &= O(\mu), \end{aligned} \tag{3.57}$$

where we used (3.18) (p26), (3.19) (p26) and Assumption 3.2. The dominating part is underlined. Thus using (3.55), (3.57), we have

$$\begin{aligned} dz_{\mathcal{B}} &= -A_{\mathcal{B}}^T dy_{\mathcal{B}} - (r_d)_{\mathcal{B}} \\ &= -A_{\mathcal{B}}^T(dy_{\mathcal{B}}^{\text{aff}} + dy_{\mathcal{B}}^\mu) - (r_d)_{\mathcal{B}} \\ &= O(\mu). \end{aligned}$$

To prove the bound on dx, we use the third equation of (3.53) and have

$$dx = -Z^{-1}Xdz - x + \mu Z^{-1}e.$$

Using the bounds of $dz_{\mathcal{N}}$ and $dz_{\mathcal{B}}$, and the size of x_i and z_i in Assumption 3.2, we see that $dx_{\mathcal{B}} = O(1)$ and $dx_{\mathcal{N}} = O(\mu)$. ■

We remark that the bounds in the above theorem are tight as illustrated by the data in Table 3.3 (p51).

We can use the same technique (using the structure information of $AXZ^{-1}A^T$) to prove that in Lemma 3.21, the component $dy - dy^{\text{aff}}$ is magnitude $O(\mu)$, and thus prove dy is $O(\mu)$ in Lemma 3.21 in conjunction with the $O(\mu)$ bound for the affine scaling direction. This gives an alternative proof for the bound on dy in Lemma 3.21.

Due to the different estimates of the size of dy, we have different error bounds for these two linear systems. We call the direction defined in Lemma 3.21 the "semi-affine" direction; and we call the direction defined in Lemma 3.22 the "centering" direction. In the following sections we find the error bounds for both directions.

3.4.1 The Semi-Affine Direction (3.52)

Error in fl(dy) for The Semi-Affine Direction

Theorem 3.23 *Suppose Assumption 3.2 holds,* $\text{rank}\,(A_{\mathcal{B}}) < m$, *and* $\mu > 10\sqrt{\mathbf{u}}$. *Let dy be the exact solution of*

$$AXZ^{-1}A^T dy = -r_p + A(-Z^{-1}Xr_d + x - w), \tag{3.58}$$

where $w = O(\mu^2)$. *Let fl(dy) be the computed solution of* (3.58) *using a backward stable linear equation solver. Then*

$$fl(dy) - dy = O(\mathbf{u}/\mu).$$

Proof. Since fl(dy) comes from a backward stable solver, we have

$$\text{fl}(AXZ^{-1}A^T)\text{fl}(dy) = \text{fl}(-r_p + A(-Z^{-1}Xr_d + x - w)) + O(\mathbf{u}).$$

The $O(\mathbf{u})$ term can be folded into the fl($\cdot$) on the right-hand side because the argument in the fl($\cdot$) is $O(1)$. So, we have

$$\text{fl}(AXZ^{-1}A^T)\text{fl}(dy) = \text{fl}(-r_p + A(-Z^{-1}Xr_d + x - w)). \tag{3.59}$$

Using similar analysis as in Theorem 3.6 (p20), we can bound the right-hand side in (3.59) as follows.

$$\text{fl}(-r_p+A(-Z^{-1}Xr_d+x-w)) = -r_p+A(-Z^{-1}Xr_d+x-w)+[A_{\mathcal{B}}O(\mathbf{u}/\mu)+A_{\mathcal{N}}O(\mu\mathbf{u})+O(\mathbf{u})]. \tag{3.60}$$

Notice that (3.58) in the lemma is obtained from (3.52) by a block elimination. Now, using Lemma 3.14, (3.20) (p26), (3.60), and (3.16) (p24), we have

$$\begin{aligned}\mathrm{fl}(dy) - dy = {} & [\mathrm{fl}(AXZ^{-1}A^T)]^{-1} \qquad (3.61)\\ & \{A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u}) - [\mathrm{fl}(AXZ^{-1}A^T) - AXZ^{-1}A^T]dy\}\\ = {} & [\widehat{U}_L\ \widehat{U}_S]\begin{bmatrix}\widehat{\Sigma}_L^{-1} & 0\\ 0 & \underline{\widehat{\Sigma}_S^{-1}}\end{bmatrix}\begin{bmatrix}\widehat{U}_L^T\\ \widehat{U}_S^T\end{bmatrix}\\ & \{A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + \underline{O(\mathbf{u})} - [A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + \underline{O(\mathbf{u}/\mu)}]dy\}. \qquad (3.62)\end{aligned}$$

Since $\widehat{\Sigma}_L^{-1} = \Theta(\mu)$, $\widehat{\Sigma}_S^{-1} = \Theta(1/\mu)$, $dy = O(\mu)$ (Lemma 3.21), and $\widehat{U}_S^T A_{\mathcal{B}} = O(\mu^2)$, we observe that the dominant error, the underlined part, is $O(\mathbf{u}/\mu)$. ■

Error in fl(dx)

Theorem 3.24 *Suppose Assumption 3.2 holds,* $\mathrm{rank}\,(A_{\mathcal{B}}) < m$, *and* $\mu > 10\sqrt{\mathbf{u}}$. *Let* dx *be the exact solution back-solved from* dy *by*

$$dx = Z^{-1}XA^T dy + ZXr_d - x + w. \qquad (3.63)$$

Let fl(dx) be the floating point computed solution back-solved from fl(dy) by the same equation. If fl(dy) has the error bound in Theorem 3.23, then

$$fl(dx_i) = dx_i + O(\mathbf{u}/\mu)\ \ (i \in \mathcal{B}), \qquad fl(dx_i) = dx_i + O(\mathbf{u})\ \ (i \in \mathcal{N}).$$

Proof. Similar to the result in (3.34) (p31), we can derive the bound on $\mathrm{fl}(dx_{\mathcal{N}})$ by using the error bound on $\mathrm{fl}(dy)$ from Theorem 3.23. If index $i \in \mathcal{N}$, we have

$$\begin{aligned}& \mathrm{fl}(dx_i)\\ =& \mathrm{fl}(z_i^{-1}x_i(A^T\mathrm{fl}(dy))_i) + \mathrm{fl}(z_i^{-1}x_i(r_d)_i) - x_i + \mathrm{fl}(w) + O(\mu\mathbf{u})\\ =& [(z_i^{-1}x_i + O(\mu\mathbf{u})][\underline{(A^T dy)_i + O(\mathbf{u}/\mu)}] + [z_i^{-1}x_i + O(\mu\mathbf{u})][(r_d)_i + O(\mathbf{u})]\\ & - x_i + w + O(\mu\mathbf{u})\\ =& z_i^{-1}x_i(A^T dy)_i + O(\mathbf{u}) + z_i^{-1}x_i(r_d)_i + O(\mu\mathbf{u}) - x_i + w + O(\mu\mathbf{u})\\ =& dx_i + O(\mathbf{u}). \qquad (3.64)\end{aligned}$$

The underlined part is the main difference from (3.34) (p31).

If index $i \in \mathcal{B}$, using (3.62), we have

$$\begin{aligned}
&\|A_{\mathcal{B}}^T(\mathrm{fl}(dy) - dy)\| \\
=&\left\| A_{\mathcal{B}}^T[\widehat{U}_L\ \widehat{U}_S] \begin{bmatrix} \widehat{\Sigma}_L^{-1} & 0 \\ 0 & \widehat{\Sigma}_S^{-1} \end{bmatrix} \begin{bmatrix} \widehat{U}_L^T \\ \widehat{U}_S^T \end{bmatrix} \right. \\
&\left. \{\underline{A_{\mathcal{B}} O(\mathbf{u}/\mu)} + A_{\mathcal{N}} O(\mu\mathbf{u}) + O(\mathbf{u}) - [A_{\mathcal{B}} O(\mathbf{u}/\mu) + A_{\mathcal{N}} O(\mu\mathbf{u}) + O(\mathbf{u}/\mu)]dy\} \right\|. \qquad (3.65)
\end{aligned}$$

Again, using the property that $A_{\mathcal{B}}^T \widehat{U}_S = O(\mu^2)$, $\widehat{\Sigma}_L^{-1} = \Theta(\mu)$, $\widehat{\Sigma}_S^{-1} = \Theta(1/\mu)$, $dy = O(\mu)$, we see the underlined parts dominate, which gives

$$A_{\mathcal{B}}^T(\mathrm{fl}(dy) - dy) = O(\mathbf{u}). \qquad (3.66)$$

So, similarly to (3.64), we have

$$\begin{aligned}
&\mathrm{fl}(dx_{\mathcal{B}}) \\
=&\mathrm{fl}(Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} (A_{\mathcal{B}}^T \mathrm{fl}(dy))) + \mathrm{fl}(Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} (r_d)_{\mathcal{B}}) - x_{\mathcal{B}} + \mathrm{fl}(w) + O(\mathbf{u}) \\
=&[(Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} + O(\mathbf{u}/\mu)][\underline{(A^T dy)_{\mathcal{B}} + O(\mathbf{u})}] + [Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} + O(\mathbf{u}/\mu)][(r_d)_{\mathcal{B}} + O(\mathbf{u})] \\
&- x_{\mathcal{B}} + w + O(\mathbf{u}) \\
=&Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} (A^T dy)_{\mathcal{B}} + O(\mathbf{u}/\mu) + Z_{\mathcal{B}}^{-1} X_{\mathcal{B}} (r_d)_{\mathcal{B}} + O(\mathbf{u}/\mu) - x_{\mathcal{B}} + w + O(\mathbf{u}) \\
=&dx_{\mathcal{B}} + O(\mathbf{u}/\mu). \qquad (3.67)
\end{aligned}$$

■

Error in fl(dz)

Theorem 3.25 *Suppose Assumption 3.2 (p18) holds,* $\mathrm{rank}\,(A_{\mathcal{B}}) < m$, *and* $\mu > 10\sqrt{\mathbf{u}}$. *Let* dz *be the exact solution back-solved from* dy *by* $A^T dy + dz = -r_d$, *the first equation of (3.52). Let* $fl(dz) = fl(-r_d - A^T fl(dy))$ *be the floating point computed solution of* dz *from* $fl(dy)$, *and suppose that* $fl(dy)$ *has the error bound in Theorem 3.23. Then*

$$fl(dz_i) = dz_i + O(\mathbf{u}) \ \ (i \in \mathcal{B}), \qquad fl(dz_i) = dz_i + O(\mathbf{u}/\mu) \ \ (i \in \mathcal{N}).$$

Proof. By using the property that $dy = O(\mu)$ and Theorem 3.23, we have

$$\begin{aligned}
\text{fl}(dz) =& \text{fl}(-r_d - A^T dy) \\
=& -\text{fl}(r_d) - [\text{fl}(A^T)\text{fl}(dy) + O(\mu\mathbf{u})] + O(\mu\mathbf{u}) \\
=& -(r_d + O(\mathbf{u})) - [A^T(dy + O(\mathbf{u}/\mu))] + O(\mu\mathbf{u}) \\
=& -r_d - A^T dy + O(\mathbf{u}/\mu) \\
=& dz + O(\mathbf{u}/\mu).
\end{aligned}$$

We now show the bound for index $i \in \mathcal{B}$. By using the bound in (3.66), we have

$$\begin{aligned}
\text{fl}(dz_{\mathcal{B}}) =& \text{fl}(-(r_d)_{\mathcal{B}} - A_{\mathcal{B}}^T dy) \\
=& -\text{fl}(r_d)_{\mathcal{B}} - [\text{fl}(A_{\mathcal{B}}^T)\text{fl}(dy) + O(\mu\mathbf{u})] + O(\mu\mathbf{u}) \\
=& -(r_d)_{\mathcal{B}} + O(\mathbf{u})) - [A_{\mathcal{B}}^T dy + O(\mathbf{u})] + O(\mu\mathbf{u}) \\
=& -(r_d)_{\mathcal{B}} - A_{\mathcal{B}}^T dy + O(\mathbf{u}) \\
=& dz_{\mathcal{B}} + O(\mathbf{u}). \qquad (3.68)
\end{aligned}$$

■

3.4.2 The Centering Direction

Error in fl(dy) for the centering direction

Theorem 3.26 *Suppose Assumption 3.2 holds,* $\text{rank}\,(A_{\mathcal{B}}) < m$, *and* $\mu > 10\sqrt{\mathbf{u}}$. *Let* dy *be the exact solution of the middle block of* (3.4) *(p16) with* $\sigma = 1$. *Let* $fl(dy)$ *be the computed solution by any backward stable linear equation solver. Then*

$$fl(dy) - dy = O(\mathbf{u}/\mu^2).$$

Proof. Since fl(dy) comes from a backward stable linear system, we have

$$\text{fl}(AXZ^{-1}A^T)\text{fl}(dy) = \text{fl}(-r_p + A(-Z^{-1}Xr_d + x - \mu Z^{-1}e)) + O(\mathbf{u}). \qquad (3.69)$$

The $O(\mathbf{u})$ term can be folded into the fl$(\cdot)$ on the right-hand side because the argument in the fl$(\cdot)$ is $O(1)$. So, we have

$$\text{fl}(AXZ^{-1}A^T)\text{fl}(dy) = \text{fl}(-r_p + A(-Z^{-1}Xr_d + x - \mu Z^{-1}e)). \tag{3.70}$$

Now, using Lemma 3.14, we have

$$\begin{aligned}\text{fl}(dy) - dy =&[\text{fl}(AXZ^{-1}A^T)]^{-1} \qquad (3.71)\\ &\{A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u}) - [\text{fl}(AXZ^{-1}A^T) - AXZ^{-1}A^T]dy\}\\ =&[\widehat{U}_L\ \widehat{U}_S]\begin{bmatrix}\widehat{\Sigma}_L^{-1} & 0\\ 0 & \underline{\widehat{\Sigma}_S^{-1}}\end{bmatrix}\begin{bmatrix}\widehat{U}_L^T\\ \widehat{U}_S^T\end{bmatrix}\\ &\{A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u}) - [A_{\mathcal{B}}O(\mathbf{u}/\mu) + A_{\mathcal{N}}O(\mu\mathbf{u}) + \underline{O(\mathbf{u}/\mu)}]dy\}. \qquad (3.72)\end{aligned}$$

Since $\widehat{\Sigma}_L^{-1} = \Theta(\mu)$, $\widehat{\Sigma}_S^{-1} = \Theta(1/\mu)$, $dy = O(1)$, and $\widehat{U}_S^T A_{\mathcal{B}} = O(\mu^2)$, we observe that the dominant errors are the underlined parts, which are $O(\mathbf{u}/\mu^2)$. ■

Error in fl(dx)

Theorem 3.27 *Suppose Assumption 3.2 holds,* $\text{rank}\,(A_{\mathcal{B}}) < m$, *and* $\mu > 10\sqrt{\mathbf{u}}$. *Let* dx *be the exact solution back-solved from* dy *by*

$$dx = Z^{-1}XA^T dy + ZXr_d - x + \mu Z^{-1}e. \tag{3.73}$$

Let fl(dx) be the floating point computed solution back-solved from fl(dy) by the same equation. If fl(dy) has the error bound in Theorem 3.26, then

$$fl(dx_i) = dx_i + O(\mathbf{u}/\mu)\ \ (i \in \mathcal{B}), \qquad fl(dx_i) = dx_i + O(\mathbf{u}/\mu)\ \ (i \in \mathcal{N}).$$

Proof. Similar to the result in (3.34) (p31), we can derive the bound on fl$(dx_{\mathcal{N}})$ by using

the error bound on fl(dy) from Theorem 3.26. If index $i \in \mathcal{N}$, we have

$$\begin{aligned}
&\text{fl}(dx_i)\\
=&\text{fl}(z_i^{-1}x_i(A^T\text{fl}(dy))_i) + \text{fl}(z_i^{-1}x_i(r_d)_i) - x_i + \text{fl}(\mu z_i^{-1}) + O(\mathbf{u})\\
=&[(z_i^{-1}x_i + O(\mu\mathbf{u})]\underline{[(A^Tdy)_i + O(\mathbf{u}/\mu^2)]} + [z_i^{-1}x_i + O(\mu\mathbf{u})][(r_d)_i + O(\mathbf{u})]\\
&- x_i + [\mu(z_i^{-1} + O(\mathbf{u})) + O(\mu\mathbf{u})] + O(\mu\mathbf{u})\\
=&z_i^{-1}x_i(A^Tdy)_i + O(\mathbf{u}/\mu) + z_i^{-1}x_i(r_d)_i + O(\mu\mathbf{u}) - x_i + \mu z_i^{-1} + O(\mu\mathbf{u})\\
=&dx_i + O(\mathbf{u}/\mu). \qquad (3.74)
\end{aligned}$$

The underlined part in the above equation is the main difference from (3.34) (p31).

If index $i \in \mathcal{B}$, using (3.72), we have

$$\begin{aligned}
&\|A_{\mathcal{B}}^T(\text{fl}(dy) - dy)\|\\
=&\left\| A_{\mathcal{B}}^T[\widehat{U}_L\ \widehat{U}_S]\begin{bmatrix}\widehat{\Sigma}_L^{-1} & 0\\ 0 & \widehat{\Sigma}_S^{-1}\end{bmatrix}\begin{bmatrix}\widehat{U}_L^T\\ \widehat{U}_S^T\end{bmatrix}\right.\\
&\left.\{\underline{A_{\mathcal{B}}O(\mathbf{u}/\mu)} + A_{\mathcal{N}}O(\mu\mathbf{u}) + O(\mathbf{u}) - [\underline{A_{\mathcal{B}}O(\mathbf{u}/\mu)} + A_{\mathcal{N}}O(\mu\mathbf{u}) + \underline{O(\mathbf{u}/\mu)}]dy\}\right\|. \qquad (3.75)
\end{aligned}$$

Again, using the property that $A_{\mathcal{B}}^T\widehat{U}_S = O(\mu^2)$, $\widehat{\Sigma}_L^{-1} = \Theta(\mu)$, $\widehat{\Sigma}_S^{-1} = \Theta(1/\mu)$, $dy = O(1)$, we see the underlined parts as well as the term $A_{\mathcal{B}}^T\widehat{U}_S\widehat{\Sigma}_s^{-1}\widehat{U}_S^TO(\mathbf{u}/\mu)dy$ dominate, which gives

$$A_{\mathcal{B}}^T(\text{fl}(dy) - dy) = O(\mathbf{u}). \qquad (3.76)$$

So, using (3.76), we have

$$\begin{aligned}
\text{fl}(dx_{\mathcal{B}}) =&\text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}(A_{\mathcal{B}}^T\text{fl}(dy))) + \text{fl}(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}(r_d)_{\mathcal{B}}) - x_{\mathcal{B}} + \text{fl}(\mu z_{\mathcal{B}}^{-1}) + O(\mathbf{u})\\
=&[(Z_{\mathcal{B}}^{-1}X_{\mathcal{B}} + O(\mathbf{u}/\mu)]\underline{[(A^Tdy)_{\mathcal{B}} + O(\mathbf{u})]} + [Z_{\mathcal{B}}^{-1}X_{\mathcal{B}} + O(\mathbf{u}/\mu)][(r_d)_{\mathcal{B}} + O(\mathbf{u})]\\
&- x_{\mathcal{B}} + [\mu(z_{\mathcal{B}}^{-1} + O(\mathbf{u}/\mu)) + O(\mathbf{u})] + O(\mathbf{u})\\
=&Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}(A^Tdy)_{\mathcal{B}} + O(\mathbf{u}/\mu) + Z_{\mathcal{B}}^{-1}X_{\mathcal{B}}(r_d)_{\mathcal{B}} + O(\mathbf{u}/\mu) - x_{\mathcal{B}} + \mu z_{\mathcal{B}}^{-1} + O(\mathbf{u})\\
=&dx_{\mathcal{B}} + O(\mathbf{u}/\mu). \qquad (3.77)
\end{aligned}$$

■

Error in fl(dz)

Theorem 3.28 *Suppose Assumption 3.2 (p18) holds,* $\text{rank}\,(A_{\mathcal{B}}) < m$, *and* $\mu > 10\sqrt{\mathbf{u}}$. *Let* dz *be the exact solution back-solved from* dy *by* $A^T dy + dz = -r_d$. *Let* $fl(dz) = fl(-r_d - A^T fl(dy))$ *be the floating point computed solution of* dz *back-solved from* $fl(dy)$, *and* $fl(dy)$ *has the error bound in Theorem 3.26. Then*

$$fl(dz_i) = dz_i + O(\mathbf{u}) \;\; (i \in \mathcal{B}), \qquad fl(dz_i) = dz_i + O(\mathbf{u}/\mu^2) \;\; (i \in \mathcal{N}).$$

Proof. By using the property that $dy = O(1)$ and the bound on fl(dy) (Theorem 3.26), we have

$$\begin{aligned}
\text{fl}(dz) =& \text{fl}(-r_d - A^T \text{fl}(dy)) \\
=& -(r_d + O(\mathbf{u})) - [A^T(dy + O(\mathbf{u}/\mu^2))] + O(\mathbf{u}) \\
=& -r_d - A^T dy + O(\mathbf{u}/\mu^2) \\
=& dz + O(\mathbf{u}/\mu^2).
\end{aligned}$$

We now show the bound for index $i \in \mathcal{B}$. By using the bound in (3.76), we have

$$\begin{aligned}
\text{fl}(dz_{\mathcal{B}}) =& \text{fl}(-(r_d)_{\mathcal{B}} - A_{\mathcal{B}}^T dy) \\
=& -\text{fl}(r_d)_{\mathcal{B}} - [\text{fl}(A_{\mathcal{B}}^T)\text{fl}(dy) + O(\mathbf{u})] + O(\mathbf{u}) \\
=& -(r_d)_{\mathcal{B}} + O(\mathbf{u})) - [A_{\mathcal{B}}^T dy + O(\mathbf{u})] + O(\mathbf{u}) \\
=& -(r_d)_{\mathcal{B}} - A_{\mathcal{B}}^T dy + O(\mathbf{u}) \\
=& dz_{\mathcal{B}} + O(\mathbf{u}).
\end{aligned}$$

■

Remarks: these two sets of error bounds for the semi-affine and the centering direction are interesting in the sense that just the change of the parameter σ yields a big change in the error estimates. Our numerical results in Tables 3.2 (p50), 3.3 (p51) show that these error bounds are tight.

In summary, we observe that the error bounds for dx, dy, dz in the degenerate case are worse than the error bounds for dx, dy, dz in the non-degenerate case. However, this may not pose a big problem in computations.

3.4.3 The Maximal Step Length α

Most of the search directions in practice are a combination of the semi-affine direction and the centering direction. We consider a convex combination of these two directions with $(1-\sigma)$ weight on the semi-affine direction and σ weight on the centering direction. Such a convex combination satisfies the linear system (3.78). If we assume that the error bounds for the semi-affine and the centering direction in the previous section hold, then the error bounds on their convex combination satisfy the bounds (3.79)–(3.84). The following theorem shows the error bound for the maximal step length.

Theorem 3.29 *Suppose that Assumption 3.2 holds. Let (dx, dy, dz) be the exact solution of the following linear system*

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -ZXe + \sigma\mu e + (1-\sigma)w \end{bmatrix}, \tag{3.78}$$

where $w = O(\mu^2)$ and $\sigma \in [0,1]$. Let $(\widehat{dx}, \widehat{dy}, \widehat{dz})$ be an approximation to this step and let the following conditions hold:

$$\begin{aligned}
(dx_{\mathcal{B}}, dz_{\mathcal{N}}) &= (1-\sigma)O(\mu) + \sigma O(1), && (3.79)\\
(dx_{\mathcal{N}}, dz_{\mathcal{B}}) &= O(\mu), && (3.80)\\
dx_{\mathcal{B}} - \widehat{dx_{\mathcal{B}}} &= O(\mathbf{u}/\mu), && (3.81)\\
dx_{\mathcal{N}} - \widehat{dx_{\mathcal{N}}} &= (1-\sigma)O(\mathbf{u}) + \sigma O(\mathbf{u}/\mu), && (3.82)\\
dz_{\mathcal{B}} - \widehat{dz_{\mathcal{B}}} &= O(\mathbf{u}), && (3.83)\\
dz_{\mathcal{N}} - \widehat{dz_{\mathcal{N}}} &= (1-\sigma)O(\mathbf{u}/\mu) + \sigma O(\mathbf{u}/\mu^2). && (3.84)
\end{aligned}$$

Suppose that the centering parameter σ is small enough such that

$$-dx_i/x_i < 1, \quad \textit{and} \quad -dz_j/z_j < 1 \quad \forall i \in \mathcal{B} \textit{ and } \forall j \in \mathcal{N}, \tag{3.85}$$

and

$$-\widehat{dx_i}/x_i < 1 \quad \textit{and} \quad -\widehat{dz_j}/z_j < 1 \quad \forall i \in \mathcal{B} \textit{ and } \forall j \in \mathcal{N}. \tag{3.86}$$

Let α^* denote the largest number in $[0,1]$ such that

$$(x + \alpha dx, z + \alpha dz) \geq 0 \quad \textit{for all } \alpha \in [0,\ \alpha^*]. \tag{3.87}$$

And, suppose $\hat{\alpha}^*$ is obtained by replacing (dx, dz) with $(\widehat{dx}, \widehat{dz})$ in (3.87). Then for all μ and σ sufficiently small, we have

$$\alpha^* = 1 - (1-\sigma)O(\mu) - \sigma O(1), \tag{3.88}$$

$$\hat{\alpha}^* = \alpha^* + (1-\sigma)O(\mathbf{u}/\mu) + \sigma(\mathbf{u}/\mu^2). \tag{3.89}$$

Proof. (We follow a similar approach as the one for Theorem 3.19.) Our assumptions (3.85) (and (3.86)) show that the values $dx_{\mathcal{N}}$, $dz_{\mathcal{B}}$ (and $\widehat{dx_{\mathcal{N}}}$, $\widehat{dz_{\mathcal{B}}}$) determine whether or not α^* (and $\hat{\alpha}^*$), is less than 1. Hence, α^* satisfies

$$\frac{1}{\alpha^*} = \max(1, \max_{i\in\mathcal{B}} -\frac{dz_i}{z_i}, \max_{i\in\mathcal{N}} -\frac{dx_i}{x_i}). \tag{3.90}$$

From the last row of (3.78), we have $z_i dx_i + x_i dz_i = -z_i x_i + \sigma\mu + (1-\sigma)w_i$. Since $z_i x_i = \Theta(\mu)$ and $w_i = O(\mu^2)$, we have

$$-\frac{dx_i}{x_i} = 1 + \frac{dz_i}{z_i} - \sigma\frac{\mu}{x_i z_i} - (1-\sigma)\frac{w_i}{x_i z_i} < 1 + \frac{dz_i}{z_i} + (1-\sigma)O(\mu).$$

For $i \in \mathcal{N}$, we have from (3.79) and (3.6) that $dz_i/z_i = (1-\sigma)O(\mu) + \sigma O(1)$. Thus

$$\max_{i\in\mathcal{N}} -\frac{dx_i}{x_i} \leq 1 + (1-\sigma)O(\mu) + \sigma O(1).$$

Similarly, we have

$$\max_{i\in\mathcal{B}} -\frac{dz_i}{z_i} \leq 1 + (1-\sigma)O(\mu) + \sigma O(1).$$

So if σ is small enough, we have

$$1/\alpha^* \leq \max(1, 1 + (1-\sigma)O(\mu) + \sigma O(1)) \implies \alpha^* = 1 - (1-\sigma)O(\mu) - \sigma O(1).$$

For the quantity $\hat{\alpha}^*$, we have from (3.82) that

$$\frac{\widehat{dx_i}}{x_i} - \frac{dx_i}{x_i} = \frac{(1-\sigma)O(\mathbf{u}) + \sigma O(\mathbf{u}/\mu)}{\Theta(\mu)} = (1-\sigma)O(\mathbf{u}/\mu) + \sigma O(\mathbf{u}/\mu^2), \quad (i \in \mathcal{N}).$$

Similarly, we have from (3.83) that

$$\frac{\widehat{dz_i}}{z_i} - \frac{dz_i}{z_i} = \frac{O(\mathbf{u})}{\Theta(\mu)} = O(\mathbf{u}/\mu), \quad (i \in \mathcal{B}).$$

Therefore, from (3.90), we have $\hat{\alpha}^* = \alpha^* + (1-\sigma)O(\mathbf{u}/\mu) + \sigma(\mathbf{u}/\mu^2)$. ■

We remark that the above error bound on $\hat{\alpha}^*$ requires small values for σ. For example, in the case of the centering direction, $\sigma = 1$, we can obtain an inaccurate maximal step length $\hat{\alpha}^*$, as illustrated by the value of $|\alpha - \mathrm{fl}(\alpha)|$ in Table 3.3 (p51). However, if σ is small, then the above theorem states that the algorithm makes good progress.

3.4.4 Numerical Example

In this subsection, we use the same matrix A, as in Example 3.20, to illustrate the error bounds for the degenerate case. The data A and optimal solution x^*, y^*, and z^* of the LP problem is given below:

$$A = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 2 & 2 & 4 & 1 \end{bmatrix}, \quad x^* = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad y^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad z^* = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \tag{3.91}$$

The data b, c is defined by $Ax^* = b$ and $A^T y^* + z^* = c$. And, the partition of the indices is $\mathcal{B} = \{1, 3\}$, and $\mathcal{N} = \{2, 4\}$. Notice that $\mathrm{rank}\,(A_{\mathcal{B}}) = 1$. We let the initial x, y, and z be

$$x = \begin{bmatrix} 1.0004568 \\ 1.1713298e{-}4 \\ 1.0001432 \\ 1.634266e{-}4 \end{bmatrix}, \quad y = \begin{bmatrix} 1.00005026 \\ 1.16595e{-}4 \end{bmatrix}, \quad z = \begin{bmatrix} 1.9454628e{-}4 \\ 9.9961681e{-}1 \\ 1.398727e{-}4 \\ 1.0001916 \end{bmatrix}.$$

We check the duality gap and the residuals

$$\mu = 1.1641818e{-}004, \quad r_p = \begin{bmatrix} 7.4320000e{-}004 \\ 1.7370086e{-}003 \end{bmatrix}, \quad r_d = \begin{bmatrix} 4.7799628e{-}004 \\ -1.5000000e{-}004 \\ 7.0448398e{-}004 \\ 3.0819500e{-}004 \end{bmatrix}.$$

	$\mathbf{u} = 1e{-}7$	$\mathbf{u} = 1e{-}8$	$\mathbf{u} = 1e{-}9$	$\mathbf{u} = 1e{-}10$	$\mathbf{u} = 1e{-}11$	$\mathbf{u} = 1e{-}12$
$\lvert dy - \mathrm{fl}(dy)\rvert_i$:	$5.5e{-}4$	$3.8e{-}4$	$4.1e{-}4$	$1.3e{-}5$	$1.2e{-}6$	$2.9e{-}7$
($\lVert dy\rVert$=$4.1e{-}4$)	$2.8e{-}4$	$1.9e{-}4$	$2.0e{-}4$	$6.3e{-}6$	$6.0e{-}7$	$1.4e{-}7$
	$1.1e{-}3$	$1.9e{-}5$	$2.5e{-}5$	$1.3e{-}6$	$4.4e{-}8$	$1.6e{-}12$
$\lvert dx - \mathrm{fl}(dx)\rvert_i$:	$6.5e{-}8$	$4.4e{-}8$	$4.8e{-}8$	$1.5e{-}9$	$1.4e{-}10$	$3.4e{-}11$
($\lVert dx_{\mathcal{B}}\rVert$=$3.3e{-}4$, $\lVert dx_{\mathcal{N}}\rVert$=$1.2e{-}4$)	$5.5e{-}4$	$9.5e{-}6$	$1.2e{-}5$	$6.3e{-}7$	$2.2e{-}8$	$1.4e{-}11$
	$4.5e{-}9$	$3.1e{-}9$	$3.3e{-}9$	$1.0e{-}10$	$9.7e{-}12$	$2.3e{-}12$
	$2.2e{-}7$	$3.6e{-}9$	$4.9e{-}9$	$2.5e{-}10$	$8.5e{-}12$	$8.2e{-}16$
$\lvert dz - \mathrm{fl}(dz)\rvert_i$:	$5.5e{-}4$	$3.8e{-}4$	$4.1e{-}4$	$1.3e{-}5$	$1.2e{-}6$	$2.9e{-}7$
($\lVert dz_{\mathcal{B}}\rVert$=$2.4e{-}4$, $\lVert dz_{\mathcal{N}}\rVert$=$3.7e{-}4$)	$7.6e{-}8$	$1.3e{-}9$	$1.7e{-}9$	$8.7e{-}11$	$3.0e{-}12$	$2.8e{-}15$
	$2.8e{-}4$	$1.9e{-}4$	$2.0e{-}4$	$6.3e{-}6$	$6.0e{-}7$	$1.4e{-}7$
$\lvert\alpha - \mathrm{fl}(\alpha)\rvert$: (α=1.0)	$5.5e{-}4$	$2.6e{-}5$	$4.1e{-}4$	$1.3e{-}5$	$1.2e{-}6$	$2.9e{-}7$

Table 3.2: The affine scaling direction ($\sigma = 0$). Error in $\mathrm{fl}(dx), \mathrm{fl}(dy)$, $\mathrm{fl}(dz)$, and $\mathrm{fl}(\alpha)$ on different $\mathbf{u}$ for the data in Section 3.4.4, where $\mathrm{fl}(\alpha)$ is the largest number (≤ 1) such that $(x + \alpha\mathrm{fl}(x), z + \mathrm{fl}(\alpha)\mathrm{fl}(z)) \geq 0$. Here $\mathcal{B} = \{1, 3\}$ and $\mathcal{N} = \{2, 4\}$.

This data satisfies Assumption 3.2.

We use double precision to solve for dy, dx, dz and assume this is the accurate solution. We then simulate the $\mathrm{fl}(\cdot)$ operation by keeping the $-\log(\mathbf{u})$ most significant digits through a roundoff computation after each arithmetic operation. So, it can be thought of as having an error of size $\mathbf{u}$.

We list the error in the affine scaling direction, at different $\mathbf{u}$ values, in Table 3.2 (p50). We see that this is consistent with the error bounds in Theorems 3.23, 3.24, and 3.25.

In Table 3.3, we list the errors for the centering direction at different $\mathbf{u}$ value. We see that these errors are consistent with the theorems (Theorem 3.26, 3.27, and 3.28) outline in this section.

	$\mathbf{u} = 1e{-}7$	$\mathbf{u} = 1e{-}8$	$\mathbf{u} = 1e{-}9$	$\mathbf{u} = 1e{-}10$	$\mathbf{u} = 1e{-}11$	$\mathbf{u} = 1e{-}12$
$\lvert dy - \mathrm{fl}(dy)\rvert_i$:	$1.5e{+}0$	$1.3e{+}0$	$4.9e{+}0$	$1.5e{-}2$	$1.5e{-}2$	$7.5e{-}4$
($\lVert dy\rVert$=1.6e+0)	$7.3e{-}1$	$6.3e{-}1$	$2.5e{+}0$	$7.6e{-}3$	$7.3e{-}3$	$3.7e{-}4$
	$1.1e{-}3$	$4.9e{-}6$	$3.6e{-}5$	$2.3e{-}6$	$2.7e{-}7$	$1.8e{-}8$
$\lvert dx - \mathrm{fl}(dx)\rvert_i$:	$1.7e{-}4$	$1.5e{-}4$	$5.8e{-}4$	$1.8e{-}6$	$1.7e{-}6$	$8.7e{-}8$
$\left(\begin{smallmatrix}\lVert dx_{\mathcal{B}}\rVert=2.8e-1\\ \lVert dx_{\mathcal{N}}\rVert=1.9e-4\end{smallmatrix}\right)$	$5.4e{-}4$	$4.8e{-}5$	$1.9e{-}5$	$1.1e{-}5$	$9.1e{-}7$	$5.0e{-}8$
	$1.2e{-}5$	$1.0e{-}5$	$4.0e{-}5$	$1.2e{-}7$	$1.2e{-}7$	$6.1e{-}9$
	$2.2e{-}7$	$9.5e{-}10$	$7.0e{-}9$	$4.5e{-}10$	$5.3e{-}11$	$3.4e{-}12$
$\lvert dz - \mathrm{fl}(dz)\rvert_i$:	$1.5e{+}0$	$1.3e{+}0$	$4.9e{+}0$	$1.5e{-}2$	$1.5e{-}2$	$7.5e{-}4$
$\left(\begin{smallmatrix}\lVert dz_{\mathcal{B}}\rVert=1.3e-4\\ \lVert dz_{\mathcal{N}}\rVert=1.6e+0\end{smallmatrix}\right)$	$7.5e{-}8$	$6.7e{-}9$	$2.7e{-}9$	$1.5e{-}9$	$1.3e{-}10$	$6.9e{-}12$
	$7.3e{-}1$	$6.3e{-}1$	$2.5e{+}0$	$7.6e{-}3$	$7.3e{-}3$	$3.7e{-}4$
$\lvert \alpha - \mathrm{fl}(\alpha)\rvert$: (α=0.7)	$3.1e{-}1$	$3.1e{-}1$	$4.1e{-}1$	$7.2e{-}3$	$7.0e{-}3$	$3.6e{-}4$

Table 3.3: The centering direction $\sigma = 1$ in (3.2) (p15). The error in $\mathrm{fl}(dx), \mathrm{fl}(dy)$, $\mathrm{fl}(dz)$, and $\mathrm{fl}(\alpha)$ on different $\mathbf{u}$ for the data in Section 3.4.4, where $\mathrm{fl}(\alpha)$ is the largest number (≤ 1) such that $(x + \mathrm{fl}(\alpha)\mathrm{fl}(x), z + \mathrm{fl}(\alpha)\mathrm{fl}(z)) \geq 0$. Here $\mathcal{B} = \{1, 3\}$ and $\mathcal{N} = \{2, 4\}$.

3.5 The Degenerate Case with $|\mathcal{B}| > m$ and $\text{rank}\,(A_{\mathcal{B}}) = m$

When $|\mathcal{B}| > m$ and $\text{rank}\,(A_{\mathcal{B}}) = m$, we have stronger error bounds for the search directions than the degenerate case but weaker error bounds than the non-degenerate case.

We first give the estimates on the magnitude of dx, dy, dz. Similar to the degenerate case with $\text{rank}\,(A_{\mathcal{B}}) < m$ (Section 3.4), the estimates depend on the parameter σ. In the case of the semi-affine direction (defined in (3.52) (p38)), Lemma 3.21 still holds; and, we have

$$(dx, dy, dz) = O(\mu).$$

In the case of the centering direction ($\sigma = 1$), we have

$$(dx_{\mathcal{N}}, dy, dz) = O(\mu) \quad \text{and} \quad dx_{\mathcal{B}} = O(1).$$

For the proof of the magnitude of $dx_{\mathcal{N}}$, dy, and dz, we can apply the proof in Theorem 3.13 (p26) without any modification, since the non-degeneracy assumption is not used for those bounds. For the proof of $dx_{\mathcal{B}} = O(1)$, we use $Z_{\mathcal{B}}dx_{\mathcal{B}} + X_{\mathcal{B}}dz_{\mathcal{B}} = -(XZe)_{\mathcal{B}} + \sigma\mu e$. Since the right-hand side, $X_{\mathcal{B}}dz_{\mathcal{B}}$, and $Z_{\mathcal{B}}$ are all $O(\mu)$, it can be seen that $dx_{\mathcal{B}}$ is $O(1)$.

We note that the $O(1)$ bound on $dx_{\mathcal{B}}$ is tight as illustrated by the following example. Let $A = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$, $b = [0,\ 1]^T$, $c = [0,\ 0,\ 0,\ 1]^T$, $\mathcal{B} = \{1, 2, 3\}$, and $\mathcal{N} = \{4\}$. Let $x = [1.0001,\ 1.0001,\ 1.0001,\ 0.0001]^T$, $y = [0.0001,\ 0.0001]^T$, $z = [0.0001,\ 0.0001,\ 0.0001,\ 1.0001]^T$, and $\sigma = 1$. It can be verified that the assumptions are satisfied. A computation gives that $dx = [1,\ -1.999e{-}4,\ 1,\ -1e{-}8]^T$, i.e., $dx_{\mathcal{B}} = O(1)$.

Second, the error bounds for $\text{fl}(dx), \text{fl}(dy)$, and $\text{fl}(dz)$ can be obtained by reusing much of the previous analysis. The proof for the error bound of $\text{fl}(dy)$ (Theorem 3.15) in the non-degenerate case still applies. Thus we have

$$\text{fl}(dy) - dy = O(\mathbf{u}). \tag{3.92}$$

For the error on dx, we can apply the analysis in (3.34) (p31) to $dx_{\mathcal{N}}$ without modification. Thus

$$\text{fl}(dx_{\mathcal{N}}) - dx_{\mathcal{N}} = O(\mu\mathbf{u}). \tag{3.93}$$

For the error on $\mathrm{fl}(dx_{\mathcal{B}})$, we first observe from (3.92) that we have

$$\|A_{\mathcal{B}}^T[\mathrm{fl}(dy) - dy]\| = O(\mathbf{u}). \tag{3.94}$$

This error bound is the same as the one in (3.66) (p42). We then can use the same analysis as in (3.67) (p42) to show that

$$\mathrm{fl}(dx_{\mathcal{B}}) - dx_{\mathcal{B}} = O(\mathbf{u}/\mu). \tag{3.95}$$

For the error on $\mathrm{fl}(dz)$, we see that (3.35) (p32) in Theorem 3.18 is still valid. Thus

$$\mathrm{fl}(dz_{\mathcal{N}}) - dz_{\mathcal{N}} = O(\mathbf{u}). \tag{3.96}$$

Since we have the bound (3.94), we then can use the same analysis as in (3.68) (p43) to get the bound

$$\mathrm{fl}(dz_{\mathcal{B}}) - d_{\mathcal{B}} = O(\mathbf{u}). \tag{3.97}$$

We remark that in the analysis of this section, we do not need the assumption that $\mu \geq 10\sqrt{\mathbf{u}}$ as in Section 3.4.

3.5.1 The Maximal Step Length α

We still consider a search direction which is a convex combination of the centering direction and the semi-affine direction, with weights of σ and $(1-\sigma)$, respectively. Such a convex combination of the search directions satisfies equation (3.98). The magnitude and error bounds on the convex combination satisfy the bounds (3.99)–(3.103).

Theorem 3.30 *Suppose that Assumption 3.2 holds. Let (dx, dy, dz) be the exact solution of the following linear system*

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -ZXe + \sigma\mu e + (1-\sigma)w \end{bmatrix}, \tag{3.98}$$

where $w = O(\mu^2)$ *and* $\sigma \in [0,1]$. *Let* $(\widehat{dx}, \widehat{dy}, \widehat{dz})$ *be an approximation to this step and assume that the following conditions hold:*

$$\begin{aligned} dx_{\mathcal{B}} &= (1-\sigma)O(\mu) + \sigma O(1), && (3.99)\\ (dx_{\mathcal{N}}, dz) &= O(\mu), && (3.100)\\ dx_{\mathcal{B}} - \widehat{dx_{\mathcal{B}}} &= O(\mathbf{u}/\mu), && (3.101)\\ dx_{\mathcal{N}} - \widehat{dx_{\mathcal{N}}} &= O(\mu\mathbf{u}), && (3.102)\\ dz - \widehat{dz} &= O(\mathbf{u}). && (3.103) \end{aligned}$$

Suppose that the centering parameter σ *is small enough such that*

$$-dx_i/x_i < 1, \quad \textit{and} \quad -\widehat{dx}_i/x_i < 1 \quad \forall i \in \mathcal{B}. \tag{3.104}$$

Let α^* *denote the largest number in* $[0,1]$ *such that*

$$(x + \alpha dx, z + \alpha dz) \geq 0 \quad \textit{for all } \alpha \in [0,\ \alpha^*]. \tag{3.105}$$

And, suppose $\hat{\alpha}^*$ *is obtained by replacing* (dx, dz) *with* $(\widehat{dx}, \widehat{dz})$ *in* (3.87). *Then, for all* μ *and* σ *sufficiently small, we have*

$$\begin{aligned} \alpha^* &= 1 - (1-\sigma)O(\mu) - \sigma O(1),\\ \hat{\alpha}^* &= \alpha^* + O(\mathbf{u}/\mu). \end{aligned}$$

Proof. (We follow a similar approach as in Theorem 3.19.) Our assumption of (3.104) ensures that whether or not α^* (and $\hat{\alpha}^*$) is less than 1 is determined by $dx_{\mathcal{N}}$ and $dz_{\mathcal{B}}$ ($\widehat{dx_{\mathcal{N}}}$ and $\widehat{dz_{\mathcal{B}}}$). Hence, α^* satisfies

$$\frac{1}{\alpha^*} = \max(1, \max_{i\in\mathcal{B}} -\frac{dz_i}{z_i}, \max_{i\in\mathcal{N}} -\frac{dx_i}{x_i}). \tag{3.106}$$

From the last row of (3.98), we have $z_i dx_i + x_i dz_i = -z_i x_i + \sigma\mu + (1-\sigma)w_i$. Since $z_i x_i = \Theta(\mu)$ and $w_i = O(\mu^2)$, we have

$$-\frac{dx_i}{x_i} = 1 + \frac{dz_i}{z_i} - \sigma\frac{\mu}{x_i z_i} - (1-\sigma)\frac{w_i}{x_i z_i} < 1 + \frac{dz_i}{z_i} + (1-\sigma)O(\mu).$$

For $i \in \mathcal{N}$, we have from (3.100) and (3.6) that $dz_i/z_i = O(\mu)$. Thus

$$\max_{i\in\mathcal{N}} -\frac{dx_i}{x_i} \leq 1 + O(\mu) + (1-\sigma)O(\mu) = 1 + (2-\sigma)O(\mu).$$

Similarly, we have

$$\max_{i\in\mathcal{B}} -\frac{dz_i}{z_i} \leq 1 + (1-\sigma)O(\mu) + \sigma O(1).$$

So if σ is small enough, we have

$$1/\alpha^* \leq \max(1, 1+(1-\sigma)O(\mu)+\sigma O(1), 1+(2-\sigma)O(\mu)) \implies \alpha^* = 1-(1-\sigma)O(\mu)-\sigma O(1).$$

For the quantity $\hat{\alpha}^*$, we have from (3.101) that

$$\frac{\widehat{dx_i}}{x_i} - \frac{dx_i}{x_i} = \frac{O(\mu\mathbf{u})}{\Theta(\mu)} = O(\mathbf{u}), \quad (i \in \mathcal{B}).$$

Similarly, we have from (3.103) that

$$\frac{\widehat{dz_i}}{z_i} - \frac{dz_i}{z_i} = \frac{O(\mathbf{u})}{\Theta(\mu)} = O(\mathbf{u}/\mu), \quad (i \in \mathcal{N}).$$

Therefore, from (3.106), we have $\hat{\alpha}^* = \alpha^* + O(\mathbf{u}/\mu)$. ■

We remark that the above error bound on $\hat{\alpha}^*$ requires small σ values. However, comparing this result to the one in the degenerate case with rank $(A) < m$ (Theorem 3.29), we see that this result is less dependent on σ in the sense that the final error bound on $\hat{\alpha}^*$ is not depend on σ.

3.5.2 Numerical Example

In this subsection, we use a similar matrix A to Example 3.20 to illustrate our error bound in the degenerate case. The data A and optimal solution x^*, y^*, and z^* of the LP problem is given below:

$$A = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 2 & 2 & 2 & 1 \end{bmatrix}, \quad x^* = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad y^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad z^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

The data b, c is defined by $Ax^* = b$ and $A^T y^* + z^* = c$. And, the partition of the indices is $\mathcal{B} = \{1, 2, 3\}$, and $\mathcal{N} = \{4\}$. Notice that $\text{rank}\,(A_{\mathcal{B}}) = 2$. We let the initial x, y, and z be

$$x = \begin{bmatrix} 1.0004568 \\ 0.99951378 \\ 1.0001432 \\ 1.634266e{-}4 \end{bmatrix}, \quad y = \begin{bmatrix} 1.00005026 \\ 1.16595e{-}4 \end{bmatrix}, \quad z = \begin{bmatrix} 1.9454628e{-}4 \\ 1.1987273e{-}4 \\ 1.398727e{-}4 \\ 1.0001916 \end{bmatrix}.$$

We check the duality gap and the residuals

$$\mu = 1.1641818e{-}004, \quad r_p = \begin{bmatrix} 7.4320000e{-}004 \\ 3.9098550e{-}004 \end{bmatrix}, \quad r_d = \begin{bmatrix} 9.3033628e{-}004 \\ 3.5306273e{-}004 \\ 1.3782627e{-}003 \\ 5.0819500e{-}004 \end{bmatrix}.$$

These data satisfy Assumption 3.2.

We use double precision to solve for dy, dx, dz and assume this is the accurate solution. We then simulate the $\text{fl}(\cdot)$ operation by keeping the $-\log(\mathbf{u})$ most significant digits through a roundoff computation after each arithmetic operation. So, it can be thought of as having an error of size $\mathbf{u}$.

We list the error of the search directions in Table 3.4 at different $\mathbf{u}$ value. We can see that these error bounds are consistent with (3.92), (3.93), (3.95), (3.96), and (3.97).

3.6 Numerical Examples on NETLIB Problems

Notice that in the proof of the error bound for the degenerate case (Theorem 3.23, 3.24, 3.25), we assume that $\mu > 10\sqrt{\mathbf{u}}$. This assumption means for a 32-bit computer, where $\mathbf{u} \simeq 10^{-16}$, we usually can progress well up to 10^{-8} accuracy. This is often observed in practice for many popular codes and that may explain why most codes' default stop tolerance is 10^{-8}.

To verify the claim that any backward stable linear solver can get up to 10^{-8} accuracy without much difficulty, we modified LIPSOL [122] to compute the NETLIB problems. Our modification is only changing the linear solver in LIPSOL to the standard backslash linear solver in Matlab. LIPSOL uses some special technique (setting a very small diagonal pivot

	$\mathbf{u} = 1e{-}7$	$\mathbf{u} = 1e{-}8$	$\mathbf{u} = 1e{-}9$	$\mathbf{u} = 1e{-}10$	$\mathbf{u} = 1e{-}11$	$\mathbf{u} = 1e{-}12$
$\lvert dy - \mathrm{fl}(dy)\rvert_i$:	$1.2e{-}7$	$1.8e{-}8$	$1.2e{-}9$	$1.4e{-}10$	$3.4e{-}13$	$4.2e{-}17$
($\lVert dy\rVert$=5.2e−4)	$9.2e{-}8$	$6.4e{-}9$	$5.8e{-}10$	$5.0e{-}11$	$2.3e{-}12$	$9.6e{-}16$
	$3.1e{-}3$	$2.1e{-}4$	$2.0e{-}5$	$1.7e{-}6$	$7.7e{-}8$	$9.0e{-}12$
$\lvert dx - \mathrm{fl}(dx)\rvert_i$:	$1.5e{-}3$	$1.1e{-}4$	$9.7e{-}6$	$8.3e{-}7$	$3.9e{-}8$	$7.7e{-}12$
$\left(\begin{smallmatrix}\lVert dx_{\mathcal{B}}\rVert=4.5e-4\\ \lVert dx_{\mathcal{N}}\rVert=1.6e-4\end{smallmatrix}\right)$	$1.5e{-}3$	$1.1e{-}4$	$9.8e{-}6$	$8.3e{-}7$	$3.8e{-}8$	$1.6e{-}11$
	$7.6e{-}11$	$4.5e{-}12$	$5.3e{-}13$	$2.9e{-}14$	$1.1e{-}15$	$1.1e{-}16$
	$6.0e{-}7$	$4.1e{-}8$	$3.8e{-}9$	$3.2e{-}10$	$1.5e{-}11$	$1.8e{-}15$
$\lvert dz - \mathrm{fl}(dz)\rvert_i$:	$1.8e{-}7$	$1.3e{-}8$	$1.2e{-}9$	$10.0e{-}11$	$4.7e{-}12$	$1.9e{-}15$
$\left(\begin{smallmatrix}\lVert dz_{\mathcal{B}}\rVert=2.7e-4\\ \lVert dz_{\mathcal{N}}\rVert=3.9e-4\end{smallmatrix}\right)$	$2.1e{-}7$	$1.5e{-}8$	$1.4e{-}9$	$1.2e{-}10$	$5.4e{-}12$	$1.7e{-}15$
	$2.9e{-}7$	$1.1e{-}8$	$5.6e{-}9$	$5.0e{-}11$	$2.3e{-}12$	$8.4e{-}16$
$\lvert\alpha - \mathrm{fl}(\alpha)\rvert$: (α=1.0)	$2.7e{-}3$	$1.1e{-}4$	$9.7e{-}6$	$8.3e{-}7$	$3.9e{-}8$	$2.0e{-}11$

Table 3.4: Error in $\mathrm{fl}(dx), \mathrm{fl}(dy)$, $\mathrm{fl}(dz)$, and $\mathrm{fl}(\alpha)$ at different $\mathbf{u}$ for the data in Section 3.5.2, where $\mathrm{fl}(\alpha)$ is the largest number (≤ 1) such that $(x + \mathrm{fl}(\alpha)\mathrm{fl}(x), z + \mathrm{fl}(\alpha)\mathrm{fl}(z)) \geq 0$. Here $\mathcal{B} = \{1, 2, 3\}$ and $\mathcal{N} = \{4\}$ and $\sigma = 0$.

	LIPSOL	Modified LIPSOL
bore3d	1e-11	8e-1
d2q06c	6e-10	2e-5
degen2	6e-9	5e-7
degen3	5e-9	2e-6
dfl001	1e-7	4e+1
greenbea	1e-7	3e-6
scorpion	1e-13	3e+4
ship08l	3e-13	4e-7
ship08s	3e-13	3e-7
ship12s	1e-11	1e-6
sierra	9e-11	1e-7

Table 3.5: NETLIB problems that Modified LIPSOL can not get desired accuracy of 10^{-8}. The numbers are the accuracies LIPSOL and Modified LIPSOL can get. The Modified LIPSOL only changes the linear solver to the standard backslash linear solver in Matlab.

to large number) in the Cholesky factorization to handle the potential breakdown due to the highly ill-conditioned matrix. Thus, by changing the linear solver in LIPSOL to a standard one, we should be able to see that most of the problems can still converge to 10^{-8} without much difficulty as long as the linear solver does not break down. We ran through all the NETLIB problems, except "QAP8, QAP12, QAP15, STOCFOR3, TRUSS", which require certain generators. Our modified LIPSOL solved almost all the 93 problems in NETLIB to the desired accuracy of 10^{-8} except the problems list in Table 3.5. The problems *bore3d, dfl001, scorpion* have large error mainly due to the break down of the linear solver. For example, the coefficient matrix of *bore3d* is not full row rank and may cause the linear solver to break down.

3.7 Summary

We summarize our results here. The bounds on the magnitude of (dx, dy, dz) are listed in the following table.

	non-deg.	deg. with rank $(A_{\mathcal{B}}) < m$		deg. with rank $(A_{\mathcal{B}}) = m$
		semi-affine	centering	
$\|dy\|$:	$O(\mu)$	$O(\mu)$	$O(1)$	$O(\mu)$
$\|dx_{\mathcal{B}}\|$:	$O(\mu)$	$O(\mu)$	$O(1)$	$O(\mu)$(semi-affine) | $O(1)$
$\|dx_{\mathcal{N}}\|$:	$O(\mu)$	$O(\mu)$	$O(\mu)$	$O(\mu)$
$\|dz_{\mathcal{B}}\|$:	$O(\mu)$	$O(\mu)$	$O(\mu)$	$O(\mu)$
$\|dz_{\mathcal{N}}\|$:	$O(\mu)$	$O(\mu)$	$O(1)$	$O(\mu)$
$\|\alpha^*\|$:	$1 - O(\mu)$	$1 - (1-\sigma)O(\mu) - \sigma O(1)$		$1 - (1-\sigma)O(\mu) - \sigma O(1)$

where the σ in the table is the weight on the centering direction if we consider a convex combination of the centering direction and semi-affine direction (see (3.78) (p47)).

The error bounds on $(\text{fl}(dx), \text{fl}(dy), \text{fl}(dz))$ and $\text{fl}(\alpha^*)$ are summarized in the following table. Our numerical examples illustrate that both the bounds on the magnitudes in the

	non-deg.	deg. with rank $(A_{\mathcal{B}}) < m$		deg. with rank $(A_{\mathcal{B}}) = m$
		semi-affine	centering	
$\|\text{fl}(dy) - dy\|$:	$O(\mathbf{u})$	$O(\mathbf{u}/\mu)$	$O(\mathbf{u}/\mu^2)$	$O(\mathbf{u})$
$\|\text{fl}(dx_{\mathcal{B}}) - dx_{\mathcal{B}}\|$:	$O(\mathbf{u})$	$O(\mathbf{u}/\mu)$	$O(\mathbf{u}/\mu)$	$O(\mathbf{u}/\mu)$
$\|\text{fl}(dx_{\mathcal{N}}) - dx_{\mathcal{N}}\|$:	$O(\mu\mathbf{u})$	$O(\mathbf{u})$	$O(\mathbf{u}/\mu)$	$O(\mu\mathbf{u})$
$\|\text{fl}(dz_{\mathcal{B}}) - dz_{\mathcal{B}}\|$:	$O(\mu\mathbf{u})$	$O(\mathbf{u})$	$O(\mathbf{u})$	$O(\mathbf{u})$
$\|\text{fl}(dz_{\mathcal{N}}) - dz_{\mathcal{N}}\|$:	$O(\mathbf{u})$	$O(\mathbf{u}/\mu)$	$O(\mathbf{u}/\mu^2)$	$O(\mathbf{u})$
$\|\text{fl}(\alpha^*) - \alpha^*\|$:	$O(\mathbf{u})$	$(1-\sigma)O(\mathbf{u}/\mu) + \sigma O(\mathbf{u}/\mu^2)$		$O(\mathbf{u}/\mu)$

Table 3.6: Summary of our error analysis.

first table and the error bounds on $(\text{fl}(dx), \text{fl}(dy), \text{fl}(dz))$ and $\text{fl}(\alpha^*)$ in the second table (Table 3.6) are tight.

For comparison purpose, we also consider the well-understood condition number analysis.

Let x be the solution of $Mx = b$ and let $\hat{x}$ be the solution of $M\hat{x} = b + \Delta b$. Then

$$\|\hat{x} - x\| \leq \|M^{-1}\|\|\Delta b\| \quad \text{and} \quad \|b\| = \|Mx\| \leq \|M\|\|x\|.$$

Thus the difference between $\hat{x}$ and x can be bounded as follows.

$$\|\hat{x} - x\| \leq \|M\|\|M^{-1}\|\|x\|\frac{\|\Delta b\|}{\|b\|} = \text{cond}\,(M)\|x\|\frac{\|\Delta b\|}{\|b\|}, \tag{3.107}$$

where $\text{cond}\,(M)$ denotes the condition number of matrix M.

We provide a similar error bounds table based on the analysis of the condition number. For simplicity, we do not consider the error in the matrix $AXZ^{-1}A^T$. The condition number estimate for the matrix $AXZ^{-1}A^T$ comes from our structure information on $AXZ^{-1}A^T$ (Section 3.2.2 (p22)). Since the right-hand side error is $O(\mathbf{u}/\mu)$ and the right-hand side is $\Theta(1)$, we can estimate the error bound on $\text{fl}(dy)$ using (3.107). The error bounds estimates that predicted by the condition number analysis are listed in the following table. In the table, we obtain the error bounds on $\text{fl}(dx)$ by using a standard entry-wise error analysis as the one used in (3.34) (p31), where no special technique is used. We obtain the error bounds on $\text{fl}(dz)$ by applying a standard entry-wise error analysis on $\text{fl}(dz) = \text{fl}(-r_d - A^T\text{fl}(dy))$.

	non-deg.	deg. with $\text{rank}\,(A_{\mathcal{B}}) < m$		deg. with $\text{rank}\,(A_{\mathcal{B}}) = m$
		semi-affine	centering	
$\text{cond}\,(AXZ^{-1}A^T)$	$\Theta(1)$	$\Theta(1/\mu^2)$		$\Theta(1)$
$\|\text{fl}(dy) - dy\|$:	$O(\mathbf{u})$	$O(\mathbf{u}/\mu^2)$	$O(\mathbf{u}/\mu^3)$	$O(\mathbf{u})$
$\|\text{fl}(dx_{\mathcal{B}}) - dx_{\mathcal{B}}\|$:	$O(\mathbf{u}/\mu)$	$O(\mathbf{u}/\mu^3)$	$O(\mathbf{u}/\mu^4)$	$O(\mathbf{u}/\mu)$
$\|\text{fl}(dx_{\mathcal{N}}) - dx_{\mathcal{N}}\|$:	$O(\mu\mathbf{u})$	$O(\mathbf{u}/\mu)$	$O(\mathbf{u}/\mu^2)$	$O(\mu\mathbf{u})$
$\|\text{fl}(dz) - dz\|$:	$O(\mathbf{u})$	$O(\mathbf{u}/\mu^2)$	$O(\mathbf{u}/\mu^3)$	$O(\mathbf{u})$

This table shows much worse error bounds than our error bounds in Table 3.6. Our improvement is especially significant in the degenerate case with $\text{rank}\,(A_{\mathcal{B}}) < m$.

In conclusion, our error bound analysis shows that the NEQ approach obtains relative accurate solutions for the non-degenerate case. For part of the search directions ($dx_{\mathcal{B}}$ and $dz_{\mathcal{N}}$), the accuracy is the best we can get since it only has an $O(\mathbf{u})$ relative error. For the degenerate case with $\text{rank}\,(A_{\mathcal{B}}) < m$, the accuracy of the search direction depends on the value of the centering parameter σ. Smaller σ values give better accuracy. The error bounds in this case require the assumption that $\mu > 10\sqrt{\mathbf{u}}$. For the degenerate case with

rank $(A_{\mathcal{B}}) = m$ and $|\mathcal{B}| > m$, the error bounds are no worse than the previous degenerate case. We do not need the $\mu > 10\sqrt{\mathbf{u}}$ assumption in this case to obtain these error bounds. In general, our error analysis explains well why most of the practical codes have a default stop tolerance of 10^{-8}. It also explains why NEQ based codes can generally progress well up to 10^{-8} without significant numerical problems as long as the data satisfies our assumption, despite the huge condition number of the underlying linear system.

Chapter 4

A Simple Stable LP Algorithm

4.1 Introduction

The purpose of this chapter is to study a *simple* alternative primal-dual development for Linear Programming (LP) based on an (inexact) Newton's method with preconditioned conjugate gradients (PCG). We do not form the usual *normal equations* (NEQ) system. No special techniques need to be introduced to avoid ill-conditioning or loss of sparsity.

We assume the coefficient matrix A is full rank and the set of strictly feasible points defined as

$$\mathcal{F}^+ = \{(x, y, z) : Ax = b, A^T y + z = c, x > 0, z > 0\}$$

is not empty.

Throughout this chapter we will use the following notation. Given a vector $x \in \mathbb{R}^n$, the matrix $X \in \mathbb{R}^{n \times n}$, or equivalently Diag (x), denotes the diagonal matrix with the vector x on the diagonal. The matrix I denotes the identity matrix, also with the corresponding correct dimension. Unless stated otherwise, $\|.\|$ denotes the Euclidean norm.

4.1.1 Background and Motivation

Solution methods for Linear Programming (LP) have evolved dramatically following the introduction of interior point methods. Currently the most popular methods are the elegant

primal-dual path-following methods. These methods are based on log-barrier functions applied to the non-negativity constraints. For example, we can start with the dual log-barrier problem, with parameter $\mu > 0$,

$$\text{(Dlogbarrier)} \qquad \begin{array}{rll} d^*_\mu := & \max & b^T y + \mu \sum_{j=1}^n \log z_j \\ & \text{s.t.} & A^T y + z = c \\ & & z > 0. \end{array} \tag{4.1}$$

The stationary point of the Lagrangian for (4.1) (x plays the role of the vector of Lagrange multipliers for the equality constraints) yields the optimality conditions

$$\begin{bmatrix} A^T y + z - c \\ Ax - b \\ X - \mu Z^{-1} \end{bmatrix} = 0, \quad x, z > 0. \tag{4.2}$$

For each $\mu > 0$, the solution of these optimality conditions is unique. The set of these solutions forms the so-called *central path* that leads to the optimum of (LP), as μ tends to 0. However, it is well-known that the Jacobian of these optimality conditions grows ill-conditioned as the log-barrier parameter μ approaches 0. This ill-conditioning (as observed for general nonlinear programs in the classical [28]) can be avoided by changing the third row of the optimality conditions to the more familiar form of the complementary slackness conditions, $ZXe - \mu e = 0$. One then applies a damped Newton's method to solve this system while maintaining positivity of x, z and reducing μ to 0. Equivalently, this can be viewed as *path following* of the central path.

It is inefficient to solve the resulting linearized system as it stands. But it has special structure that can be exploited. Block eliminations yield a positive definite system (called the normal equations, NEQ) of size m, with matrix ADA^T, where D is diagonal; see Section 4.2.2. Alternatively, a larger *augmented system* or *quasi-definite system*, of size $n \times n$ can be used, e.g. [114], [103, Chap. 19]. However, the ill-conditioning returns for these systems, i.e. we first get rid of the ill-conditioning by changing the log-barrier optimality conditions; we then bring it back with the back-solves after the block eliminations; see Section 4.2.2. Another potential difficulty for NEQ system is the possible loss of sparsity after forming ADA^T, e.g. in the presence of dense columns in A.

However, there are advantages when considering the two reduced systems. The size of the normal equations system is m compared to the size $m + 2n$ of the original linearized system. And efficient factorization schemes can be applied. The augmented system is larger but there are gains in exploiting sparsity when applying factorization schemes. Moreover, the ill-conditioning for both systems has been carefully studied. Our analysis in Chapter 3 shows that the normal equation approach can usually get to accuracy of 10^{-8} without much difficulty, under the mild assumption that the data is in general well behaved. However, if we want higher accuracy, we may have problems for the degenerate case with $\text{rank}\,(A_{\mathcal{B}}) < m$. For further results on the ill-conditioning of the augmented system, see e.g. [112, 115] and the books [103, 113]. For a discussion on the growth in the condition number after the back-solve, see Remark 4.3.

The major work (per iteration) is the formation and factorization of the reduced system. However, factorization schemes can fail for huge problems and/or problems where the reduced system is not sparse. If A is sparse, then one could apply conjugate-gradient type methods and avoid the matrix multiplications, e.g. one could use $A(D(A^T v))$ for the matrix vector multiplications for the ADA^T system. However, classical iterative techniques for large sparse linear systems have not been generally used. One difficulty is that the normal equations can become extremely ill-conditioned in certain degenerate case. Iterative schemes need efficient preconditioners to be competitive. This can be the case for problems with special structure, see e.g. [51]. For other iterative approaches see e.g. [45, 22, 65, 7, 78].

Although the reduced normal equations approach has benefits as mentioned above, the ill conditioning that arises is still a potential numerical problem for obtaining high accuracy solutions. In this chapter we look at a modified approach for these interior point methods. We use a simple preprocessing technique to eliminate the primal and dual feasibility equations. Under non-degeneracy assumptions, the result is a nonsingular bilinear equation that does not necessarily become ill-conditioned. We work on this equation with an inexact Newton approach and use a preconditioned conjugate gradient type method to (approximately) solve the linearized system for the search direction. One can still use efficient Cholesky techniques in the preconditioning process, e.g. partial Cholesky factorizations that preserve sparsity (or partial QR factorizations). The advantage is that these techniques are applied to a system that does not necessarily get ill-conditioned and sparsity can be directly exploited without us-

ing special techniques. As in the case mentioned above, the approach is particularly efficient when the structure of the problem can be exploited to construct efficient preconditioners. (This is the case for certain classes of Semidefinite Programming (SDP) problems, see [107].) We also use crossover and purification techniques to speed up the convergence. In particular, the robustness of the linear system allows us to apply the so-called Tapia indicators [27] to correctly detect those variables that are zero at the solution. In addition, a crossover technique can be applied at the final stage of interior point method to take advantage of the full quadratic convergence of the pure Newton step.

4.2 Block Eliminations

4.2.1 Linearization

Note that the function F in (2.2) (p7) maps from $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ to $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$. Let $\mu > 0$ and let us consider the perturbed optimality conditions

$$F_\mu(x, y, z) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ ZXe - \mu e \end{bmatrix} = \begin{bmatrix} r_d \\ r_p \\ r_c \end{bmatrix} = 0, \tag{4.3}$$

thus defining the (resp. dual, primal) residual vectors r_d, r_p and perturbed complementary slackness r_c. The Newton equation (the linearization) for the Newton direction $ds = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}$ is

$$F'_\mu(x, y, z)ds = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} ds = -F_\mu(x, y, z). \tag{4.4}$$

Damped Newton steps

$$x \leftarrow x + \alpha_p dx, \quad y \leftarrow y + \alpha_d dy, \quad z \leftarrow z + \alpha_d dz,$$

are taken that *backtrack* from the non-negativity boundary to maintain the positivity/interiority, $x > 0, z > 0$.

Suppose that $F_\mu(x, y, z) = 0$ in (4.3). Then (4.3) imply

$$\mu = \frac{1}{n}\mu e^T e = \frac{1}{n} e^T ZXe = \frac{1}{n} z^T x = \frac{1}{n}(\text{ duality gap}),$$

i.e. the barrier parameter μ is a good measure of the duality gap. However, most practical interior point methods are infeasible methods, i.e. they do not start with primal-dual feasible solutions and stop with nonzero residuals. Similarly, if feasibility is obtained, roundoff error can result in nonzero residuals r_d, r_p in the next iteration. Therefore, in both cases,

$$\begin{aligned} n\mu &= z^T x \\ &= (c - A^T y + r_d)^T x \\ &= \left(c^T x - y^T A x + r_d^T x\right) \\ &= \left(c^T x - y^T (b + r_p) + r_d^T x\right) \\ &= \left(c^T x - b^T y - r_p^T y + r_d^T x\right) \\ &= (c + r_d)^T x - (b + r_p)^T y, \end{aligned} \tag{4.5}$$

i.e. $n\mu$ measures the duality gap of a perturbed LP. (See e.g. the survey article on error bounds [80].)

4.2.2 Reduction to the Normal Equations

The Newton equation (4.4) is solved at each iteration of a primal-dual interior point (p-d i-p) algorithm. This is the major work involved in these path-following algorithms. Solving (4.4) directly is too expensive. There are several manipulations that can be done that result in a much smaller system. We can consider this in terms of block elimination steps.

First Step in Block Elimination for Normal Equations

The customary first step in the literature is to eliminate dz using the first row of equations. (Note the linearity and coefficient I for z in the first row of (4.3).) Equivalently, apply elementary row operations to matrix $F'_\mu(x, y, z)$, or find a matrix P_Z such that the multiplication of $P_Z F'_\mu(x, y, z)$ results in a matrix with the corresponding columns of dz being formed by the identity matrix and zero matrices. This is,

$$\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -X & 0 & I \end{bmatrix} \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & -XA^T & 0 \end{bmatrix}, \tag{4.6}$$

with right-hand side

$$-\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -X & 0 & I \end{bmatrix} \begin{bmatrix} A^T y + z - c \\ Ax - b \\ ZXe - \mu e \end{bmatrix} = -\begin{bmatrix} r_d \\ r_p \\ -Xr_d + ZXe - \mu e \end{bmatrix}. \tag{4.7}$$

We let

$$P_Z = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -X & 0 & I \end{bmatrix}, \quad K = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & -XA^T & 0 \end{bmatrix}. \tag{4.8}$$

Second Step in Block Elimination for Normal Equations

The so-called normal equations are obtained by further eliminating dx. (Note the **non**linearity in x in the third row of (4.3).) Following a similar procedure as above, we define the matrices F_n, P_n with

$$\begin{aligned} F_n := P_n K \quad &:= \quad \begin{bmatrix} I & 0 & 0 \\ 0 & I & -AZ^{-1} \\ 0 & 0 & Z^{-1} \end{bmatrix} \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & -XA^T & 0 \end{bmatrix} \\ &= \quad \begin{bmatrix} 0 & A^T & I_n \\ 0 & \boxed{AZ^{-1}XA^T} & 0 \\ I_n & -Z^{-1}XA^T & 0 \end{bmatrix}. \end{aligned} \tag{4.9}$$

The right-hand side becomes

$$-P_n P_Z \begin{bmatrix} A^T y + z - c \\ Ax - b \\ ZXe - \mu e \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p + A(-Z^{-1}Xr_d + x - \mu Z^{-1}e) \\ Z^{-1}Xr_d - x + \mu Z^{-1}e \end{bmatrix}. \tag{4.10}$$

The algorithm for finding the Newton search direction using the normal equations is now evident from (4.9), i.e. we move the third column before column one and interchange the

second and third rows.

$$\begin{bmatrix} I_n & 0 & A^T \\ 0 & I_n & -Z^{-1}XA^T \\ 0 & 0 & \boxed{AZ^{-1}XA^T} \end{bmatrix} \begin{bmatrix} dz \\ dx \\ dy \end{bmatrix} = \begin{bmatrix} -r_d \\ Z^{-1}Xr_d - x + \mu Z^{-1}e \\ -r_p + A(-Z^{-1}Xr_d + x - \mu Z^{-1}e) \end{bmatrix}. \tag{4.11}$$

Thus we first solve for dy. We then back-solve for dx and finally back-solve for dz. This block upper-triangular system has the disadvantage of being ill-conditioned when evaluated at points close to the optimum. This will be shown in the next section. The condition number for the system is found from the condition number of the matrix F_n and not just the matrix $AZ^{-1}XA^T$. (Though, as mentioned above, the latter can have a uniformly bounded condition number under some standard neighbourhood assumptions plus the non-degeneracy assumption, see e.g. [44], or under the degeneracy assumption with $\text{rank}\,(A_{\mathcal{B}}) = m$, see Corollary 3.12 (p25).)

4.2.3 Roundoff Difficulties for NEQ; Examples

Roundoff difficulties are demonstrated clearly in Chapter 3. It is shown that the worst case roundoff error happens when we use NEQ to solve for the "centering" direction in Section 3.4.2. Here we show another simple example to demonstrate the catastrophic consequence of ill-conditioning when Assumption 3.2 (p18) is not satisfied.

Non-degenerate but with Large Residual

Though a problem is non-degenerate, problems can arise if the the current primal-dual point has a large residual error relative to the duality gap.

Example 4.1 *Here the residuals are not the same order as μ. We see that we get catastrophic roundoff error. Consider the simple data*

$$A = \begin{bmatrix} 1 & 1 \end{bmatrix}, \; c = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \; b = 1.$$

The optimal primal-dual variables are

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; y = -1, \; z = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

We use 6 *decimals accuracy in the arithmetic and start with the following points (infeasible) obtained after several iterations:*

$$x = \begin{bmatrix} 9.183012 \times 10^{-1} \\ 1.356397 \times 10^{-8} \end{bmatrix}, \; z = \begin{bmatrix} 2.193642 \times 10^{-8} \\ 1.836603 \end{bmatrix}, \; y = -1.163398 \; .$$

The residuals (relatively large) and duality gap measure are:

$$\|r_b\| = 0.081699, \quad \|r_d\| = 0.36537, \quad \mu = x^T z/n = 2.2528 \times 10^{-8}.$$

Though μ is small, we still have large residuals for both primal and dual feasibility. Therefore, $2\mu = n\mu$ is not a true measure of the duality gap as we do not have a feasible primal-dual pair. The two search directions, $\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}$, that are found using first the full matrix F'_μ and second the system F_n (solving dy first and then back-solving dx, dz) are, respectively,

$$\begin{bmatrix} 8.16989 \times 10^{-2} \\ -1.35442 \times 10^{-8} \\ 1.63400 \times 10^{-1} \\ -2.14348 \times 10^{-8} \\ 1.63400 \times 10^{-1} \end{bmatrix}, \quad \begin{bmatrix} -6.06210 \times 10 \\ -1.35441 \times 10^{-8} \\ 1.63400 \times 10^{-1} \\ 0 \\ 1.63400 \times 10^{-1} \end{bmatrix}.$$

Though the error in dy is small, since $m = 1$, the error after the back-substitution for the first component of dx is large, with no decimals accuracy. The resulting search direction results in no improvements in the residuals or the duality gap. Using the accurate direction from F'_μ results in good improvement and convergence.

In practice, the residuals generally decrease at the same rate as μ. (For example, this is assumed in the discussion in [114].) But, as our tests in Section 4.4 below show, the residuals and roundoff do cause a problem for NEQ when μ gets small, generally less than 10^{-8}.

4.2.4 Simple/Stable Reduction

There are other choices for the above second step in Section 4.2.2, e.g. the one resulting in the augmented system [113] or equivalently the one used in the software package [104, LOQO] that results in the quasi-definite system.

In our approach we present a different type of second elimination step. We assume that we have the special structure $A = \begin{bmatrix} S & E \end{bmatrix}$ (perhaps obtained by permuting rows and columns), where S is $m \times m$, nonsingular, and it is inexpensive to solve the corresponding linear system $Su = d$. For example, the best choice is $S = I$ obtained when adding slack variables.

We partition the diagonal matrix Z, X using the vectors $z = \begin{bmatrix} z_m \\ z_v \end{bmatrix}, \quad x = \begin{bmatrix} x_m \\ x_v \end{bmatrix}$ with lengths $m, v = n - m$. And, we denote an initial primal vector $\hat{x} = \begin{bmatrix} \hat{x}_m \\ \hat{x}_v \end{bmatrix}$. If possible, this vector is chosen primal feasible, e.g. $\hat{x} = \begin{bmatrix} \hat{x}_m \\ \hat{x}_v \end{bmatrix} = \begin{bmatrix} S^{-1}b \\ 0 \end{bmatrix}$, in the case that $S^{-1}b \geq 0$. With K given in (4.8), we define the matrices F_s, P_s with

$$\begin{aligned} F_s : \;=\; P_s K &= \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & S^{-1} & 0 & 0 \\ 0 & -Z_m S^{-1} & I_m & 0 \\ 0 & 0 & 0 & I_v \end{bmatrix} \begin{bmatrix} 0 & 0 & A^T & I_n \\ S & E & 0 & 0 \\ Z_m & 0 & -X_m S^T & 0 \\ 0 & Z_v & -X_v E^T & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & A^T & I_n \\ I_m & S^{-1}E & 0 & 0 \\ 0 & \boxed{\begin{matrix} -Z_m S^{-1} E & -X_m S^T \\ Z_v & -X_v E^T \end{matrix}} & 0 & 0 \\ 0 & & & 0 \end{bmatrix}. \end{aligned} \tag{4.12}$$

The right-hand side becomes

$$\begin{aligned} -P_s P_Z \begin{bmatrix} A^T y + z - c \\ Ax - b \\ ZXe - \mu e \end{bmatrix} &= -P_s \begin{bmatrix} r_d \\ r_b \\ -X_m (r_d)_m + Z_m X_m e - \mu e \\ -X_v (r_d)_v + Z_v X_v e - \mu e \end{bmatrix} \\ &= \begin{bmatrix} -r_d \\ -S^{-1} r_p \\ Z_m S^{-1} r_p + X_m (r_d)_m - Z_m X_m e + \mu e \\ X_v (r_d)_v - Z_v X_v e + \mu e \end{bmatrix}. \end{aligned} \tag{4.13}$$

Our algorithm uses the last two rows to solve for dx_v, dy. We then use the second row to back-solve for dx_m and then the first row to back-solve for dz. The matrix S^{-1} is never evaluated if an iterative linear solver is used, but rather the required operation is performed using a system solve. Therefore, we require this operation to be both efficient and stable. Moreover, if we started with exact dual feasibility and we find the step-length $\alpha > 0$ that maintains positivity for x, z, then we can update $y \leftarrow y + \alpha dy$ first, and then set $z = c - A^T y$; thus we maintain exact dual feasibility (up to the accuracy of the matrix multiplication and vector subtraction). There is no reason to evaluate and carry the residual to the next iteration. This works for the normal equations back-solve as well. But, if we start with exact feasibility for the primal as well, we can also update $x_v \leftarrow x_v + \alpha dx_v$ and then solve $Sx_m = b - Ex_v$. Thus we guarantee stable primal feasibility as well (up to the accuracy in evaluating E, the matrix vector multiplications and additions, and the system solve for x_m). This is discussed further at the end of Section 4.2.6.

The matrix derived in (4.12) is generally better conditioned than the one from the normal equations system (4.9) in the sense that, under non-degeneracy assumptions, the condition number is bounded at the solution. We do not change a well-posed problem into an ill-posed one. The result proved in Proposition 4.2 shows the advantages of using this *Stable Reduction*.

4.2.5 Condition Number Analysis

Proposition 4.2 *Let F_n and F_s be the matrices defined in (4.9) and (4.12). Then, the condition number of F_n diverges to infinity if $x(\mu)_i/z(\mu)_i$ diverges to infinity, for some i, as μ converges to 0. The condition number of F_s is uniformly bounded if there exists a unique primal-dual solution of problems (LP) and (DLP) in* (2.1).

Proof. Note that

$$F_n^T F_n = \begin{bmatrix} I_n & -Z^{-1}XA^T & 0 \\ -AXZ^{-1} & (AA^T + (AZ^{-1}XA^T)^2 + AZ^{-2}X^2A^T) & A \\ 0 & A^T & I_n \end{bmatrix}. \tag{4.14}$$

We now see, using interlacing of eigenvalues, that this matrix becomes increasingly ill-conditioned. Let $D = Z^{-1}X$. Then the nonzero eigenvalue of $D_{ii}^2 A_{:,i}(A_{:,i})^T$ diverges to

infinity, as μ converges to 0. Therefore the largest eigenvalue of the matrix in the middle block $AD^2A^T = \sum_{i=1}^n D_{ii}^2 A_{:,i}(A_{:,i})^T$ must diverge to infinity, i.e. the largest eigenvalue of $F_n^T F_n$ diverges to infinity. Since the smallest eigenvalue cannot exceed 1, this implies that the condition number of $F_n^T F_n$ diverges to infinity, as $\mu \to 0$ and $x(\mu)_i/z(\mu)_i$ diverges to infinity, for some i. On the other hand, the condition number of F_s is uniformly bounded. This follows from the fact that F_s converges to the Jacobian matrix in (4.22), which, as shown in Theorem 4.5 below, is nonsingular at the solution. ■

Remark 4.3 *We can observe that the condition number of the matrix $F_n^T F_n$ is greater than the largest eigenvalue of the block $AZ^{-2}X^2A^T$; equivalently, $\frac{1}{\mathrm{cond}(F_n^T F_n)}$ is smaller than the reciprocal of this largest eigenvalue. With the assumption that x and z stay in a certain neighbourhood of the central path, we know that $\min_i(z_i/x_i)$ is $O(\mu)$. Thus the reciprocal of the condition number of F_n is $O(\mu)$.*

4.2.6 The Stable Linearization

The stable reduction step above corresponds to our linearization approach. Recall the primal LP

$$\begin{array}{rrll} & p^* = & \min & c^T x \\ (LP) & & \text{s.t.} & Ax = b \\ & & & x \geq 0. \end{array} \tag{4.15}$$

An essential preprocessing step is to find a (hopefully sparse) representation of the null space of A as the range of a matrix N, i.e.

$$Ax = b \quad \text{if and only if} \quad x = \hat{x} + Nv, \text{ for some } v \in \mathbb{R}^{n-m}.$$

For our method to be efficient, we would like both matrices A, N to be sparse. More precisely, since we use an iterative method, we need both matrix vector multiplications Ax, Nv to be inexpensive. If the original problem is in symmetric form, i.e. if the constraint is of the type

$$Ex \leq b, \quad E \in \mathbb{R}^{m \times (n-m)},$$

(Applications for this form abound, e.g. the diet problem and minimum cost production problem. See e.g. [103, Chap. 16][104].) then we can add slack variables and get $A = \begin{bmatrix} I_m & E \end{bmatrix}, N = \begin{bmatrix} -E \\ I_{n-m} \end{bmatrix}$. More generally, in this paper we assume that

$$A = \begin{bmatrix} S & E \end{bmatrix}, \quad N = \begin{bmatrix} -S^{-1}E \\ I_{n-m} \end{bmatrix}, \tag{4.16}$$

where E is sparse and the linear system $Sv = d$ is nonsingular, well-conditioned and inexpensive to solve. (For example, S is block diagonal or triangular. Surprisingly, this structure holds for many of the NETLIB test set problems, i.e. except for a small, of order 4, square block, S is upper triangular and sparse.)

We can now substitute for both z, x and eliminate the first two (linear) blocks of equations in the optimality conditions (4.3). We obtain the following **single** block of equations for optimality. By abuse of notation, we keep the symbol F for the nonlinear operator. The meaning is clear from the context.

Theorem 4.4 *Suppose that $A\hat{x} = b$ and the range of N equals the nullspace of A. Then the primal-dual variables x, y, z, with $x = \hat{x} + Nv \geq 0$, $z = c - A^T y \geq 0$, are optimal for (LP), (DLP) if and only if they satisfy the single bilinear optimality equation*

$$F(v, y) := \text{Diag}\,(c - A^T y)\text{Diag}\,(\hat{x} + Nv)e = 0. \tag{4.17}$$

■

This leads to the single perturbed optimality conditions that we use for our primal-dual method,

$$F_\mu(v, y) := \text{Diag}\,(c - A^T y)\text{Diag}\,(\hat{x} + Nv)e - \mu e = 0. \tag{4.18}$$

This is a nonlinear (bilinear) system. The linearization (or Newton equation) for the search direction $ds := \begin{bmatrix} dv \\ dy \end{bmatrix}$ is

$$-F_\mu(v, y) \;\; = \;\; F'_\mu(v, y)ds. \tag{4.19}$$

The Jacobian matrix

$$F'_\mu(v,y) = \Big[\text{Diag}\,(c - A^T y)N \quad - \text{Diag}\,(\hat{x} + Nv)A^T\Big] \tag{4.20}$$

and, therefore, system (4.19) becomes

$$-F_\mu(v,y) \;=\; \text{Diag}\,(c - A^T y)Ndv - \text{Diag}\,(\hat{x} + Nv)A^T dy. \tag{4.21}$$

We note that this is a linear system of size $n \times n$. Algorithms that use reduced linearized systems of this size, exist, e.g. [103, Chap. 19] discusses the *quasi-definite* system of size $n \times n$.

Under standard assumptions, the above system has a unique solution at each (v, y) point corresponding to a strictly feasible point. In addition, we now show non-singularity of the Jacobian matrix at optimality, i.e. it does not necessarily get ill-conditioned as μ approaches 0.

Theorem 4.5 *Consider the primal-dual pair (LP),(DLP). Suppose that A is onto (full rank), the range of N is the null space of A, N is full column rank, and (x, y, z) is the* unique *primal-dual optimal solution. Then the matrix of the linear system*

$$\begin{aligned} -F_\mu &= F'_\mu ds \\ &= ZNdv - XA^T dy \end{aligned} \tag{4.22}$$

(F'_μ is Jacobian of F_μ) is nonsingular.

Proof. Suppose that $F'_\mu(v,y)ds = 0$. We need to show that $ds = (dv, dy) = 0$.

Let $\mathcal{B}$ and $\mathcal{N}$ denote the set of indices j such that $x_j = \hat{x}_j + (Nv)_j > 0$ and set of indices i such that $z_i = c_i - (A^T y)_i > 0$, respectively. Under the non-degeneracy (uniqueness) and full rank assumptions, we get $\mathcal{B} \bigcup \mathcal{N} = \{1, ...n\}$, $\mathcal{B} \bigcap \mathcal{N} = \emptyset$, and the cardinalities $|\mathcal{B}| = m$, $|\mathcal{N}| = n - m$. Moreover, the submatrix $A_\mathcal{B}$, formed from the columns of A with indices in $\mathcal{B}$, is nonsingular.

By our assumption and (4.21), we get that

$$\big(F'_\mu(v,y)ds\big)_k = (c - A^T y)_k (Ndv)_k - (\hat{x} + Nv)_k (A^T dy)_k = 0, \quad \forall k.$$

From the definitions of $\mathcal{B}, \mathcal{N}$, this implies that

$$(A^T dy)_j = 0, \forall j \in \mathcal{B}, \qquad (Ndv)_i = 0, \forall i \in \mathcal{N}. \tag{4.23}$$

The left part of (4.23) implies $A_{\mathcal{B}}^T dy = 0$, i.e. we obtain $dy = 0$.

It remains to show that $dv = 0$. From the definition of N we have $AN = 0$. Therefore, using the right part of (4.23) implies

$$\begin{aligned} 0 &= \begin{bmatrix} A_{\mathcal{B}} & A_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} (Ndv)_{\mathcal{B}} \\ (Ndv)_{\mathcal{N}} \end{bmatrix} \\ &= A_{\mathcal{B}}(Ndv)_{\mathcal{B}} + A_{\mathcal{N}}(Ndv)_{\mathcal{N}} \\ &= A_{\mathcal{B}}(Ndv)_{\mathcal{B}}. \end{aligned}$$

By the right part of (4.23) and the non-singularity of $A_{\mathcal{B}}$, we get

$$Ndv = 0.$$

Now, full rank of N implies $dv = 0$.

(An alternative proof follows using (4.12). We can see (after permutations if needed) that both K, P_s are nonsingular matrices.) ■

We use equation (4.18) and the linearization (4.21) to develop our PCG based primal-dual algorithm. This algorithm is presented and described in the next section.

4.3 Primal-Dual Algorithm

The algorithm we use follows the primal-dual interior point framework, see e.g. Algorithm 1 (p11). That is, we use Newton's method applied to the perturbed system of optimality conditions with damped step lengths for maintaining non-negativity (not necessarily positivity) constraints. The search direction is found using a preconditioned conjugate gradient type method, LSQR, due to Paige and Saunders [79]. These are applied to the last two rows of (4.12),(4.13). This contrasts with popular approaches that find the search directions by using direct factorization methods on the normal equations system. In addition, we use a crossover step, i.e. we use affine scaling (the perturbation parameter $\mu = 0$) and we do not backtrack to preserve positivity of z, x once we have found (or estimate) the region of quadratic convergence of Newton's method. Therefore, the algorithm mixes interior and exterior ideas. We also include the identification of zero values for the primal variable x and

eliminate the corresponding indices; thus reducing the dimension of the original problem. We call this a *purification step*.

The procedures are explained in more detail in the following sections.

4.3.1 Preconditioning Techniques

Recall that $Z := Z(y) = \mathrm{Diag}\,(c - A^T y)$, $X := X(v) = \mathrm{Diag}\,(\hat{x} + Nv)$ and the Jacobian of F_μ (equation (4.20)) is

$$J := F'_\mu(v, y) = \begin{bmatrix} ZN & -XA^T \end{bmatrix}. \tag{4.24}$$

Since we are interested in using a conjugate gradient type method for solving the linear system (4.22), we need efficient preconditioners. For a preconditioner we mean a *simple* nonsingular matrix M such that JM^{-1} is well conditioned. To solve system (4.22), we can solve the better conditioned systems $JM^{-1}\Delta q = -F_\mu$ and $Mds = dq$. It is clear that the best condition for JM^{-1} is obtained when the matrix M is the inverse of J. We look for a matrix M such that $M^T M$ approximates $J^T J$.

We use the package LSQR [79], that implicitly solves the normal equations $J^T Jds = -J^T F'_\mu$. Two possible choices for the preconditioning matrix M are: the square root of the diagonal of $J^T J$; and the partial Cholesky factorization of the diagonal blocks of $J^T J$. In the following we describe these approaches. Since our system is non-symmetric, other choices would be, e.g. quasi-minimal residual (QMR) algorithms [33, 34]. However, preconditioning for these algorithms is more difficult, see e.g. [9, 10].

Optimal Diagonal Column Preconditioning

We begin with the simplest of the preconditioners. For any given square matrix K let us denote $\omega(K) = \frac{\mathrm{trace}\,(K)/n}{\det(K)^{1/n}}$. Let $M = \arg\min \omega((JD)^T(JD))$ over all positive diagonal matrices D. In [24, Prop. 2.1(v)] it was shown that $M_{ii} = 1/\|J_{:i}\|$, the i-th column norm. This matrix has been identified as a successful preconditioner (see [43, Sect. 10.5], [100]) since ω is a measure of the condition number, in the sense that it is bounded above and below by a constant times the standard condition number (ratio of largest and smallest singular values).

Partial (Block) Cholesky Preconditioning

From (4.24) we obtain that

$$J^T J = \begin{bmatrix} N^T Z^2 N & -N^T Z X A^T \\ -AXZN & AX^2A^T \end{bmatrix}.$$

Suppose that z, x lies near the central path, i.e. $ZX \cong \mu I$ (approximately equal). Then the off diagonal terms of $J^T J$ are approximately 0, since $AN = 0$, by definition of N. In this case, block (partial) Cholesky preconditioning is extremely powerful.

We now look at finding a partial Cholesky factorization of $J^T J$ by finding the factorizations of the two diagonal blocks. We can actually do this using the Q-less QR factorization, i.e. suppose that $Q_Z R_Z = ZN$, $Q_X R_X = XA^T$ represents the QR factorizations with both R_Z, R_X square matrices (using the Q-less efficient form, where both Q_Z, Q_R are not found explicitly). Then

$$R_Z^T R_Z = N^T Z^2 N, \quad R_X^T R_X = AX^2 A^T. \tag{4.25}$$

We can now choose the approximate factorization

$$J^T J \cong M^T M, \quad M = \begin{bmatrix} R_Z & 0 \\ 0 & R_X \end{bmatrix}.$$

We should also mention that to calculate this preconditioner is expensive. The expense is comparable to the Cholesky factorization of the normal equation $AZ^{-1}XA^T$, i.e. $O(m^3)$. Therefore, we tested both a complete and an incomplete (denoted ILU) Cholesky preconditioner for the diagonal blocks.

4.3.2 Crossover Criteria

Let us assume that the Jacobian matrix of the function F defining the optimality conditions is nonsingular at the solution. Then, the problem has unique primal and dual solutions, Let us call it s^*. Therefore, from the standard theory for Newton's method, there is a neighbourhood of the solution s^* of quadratic convergence and, once in this neighbourhood, we can safely apply affine scaling with step lengths of one without backtracking to maintain positive definiteness.

To estimate the guaranteed convergence area of the optimal solution, we need to use a theorem due to Kantorovich [52]. We use the form in [23, Theorem 5.3.1]. We let $\mathcal{N}(x, r)$ denote the neighbourhood of x with radius r and $\mathrm{Lip}_\gamma(\mathcal{N}(x, r))$ denotes Lipschitz continuity with constant γ in the neighbourhood.

Theorem 4.6 (Kantorovich) *Let $r > 0$, $s_0 \in \mathbb{R}^n$, $F : \mathbb{R}^n \to R^n$, and assume that F is continuously differentiable in $\mathcal{N}(s_0, r)$. Assume for a vector norm and the induced operator norm that $J \in Lip_\gamma(\mathcal{N}(s_0, r))$ with $J(s_0)$ nonsingular, and that there exist constants β, $\eta \geq 0$ such that*

$$\|J(s_0)^{-1}\| \leq \beta, \quad \|J(s_0)^{-1}F(s_0)\| \leq \eta.$$

Define $\alpha = \beta\gamma\eta$. If $\alpha \leq \frac{1}{2}$ and $r \geq r_0 := (1 - \sqrt{1 - 2\alpha}\,)/(\beta\gamma)$, then the sequence $\{s_k\}$ produced by

$$s_{k+1} = s_k - J(s_k)^{-1}F(s_k), \quad k = 0, 1, \ldots,$$

is well defined and converges to s_*, a unique zero of F in the closure of $\mathcal{N}(s_0, r_0)$. If $\alpha < \frac{1}{2}$, then s_* is the unique zero of F in $\mathcal{N}(s_0, r_1)$, where $r_1 := \min[r, (1 + \sqrt{1 - 2\alpha}\,)/(\beta\gamma)]$ and

$$\|s_k - s_*\| \leq (2\alpha)^{2^k}\frac{\eta}{\alpha}, \quad k = 0, 1, \ldots,$$

■

We follow the notation in Dennis and Schnabel's book [23] and find the Lipschitz constant used to determine the region of quadratic convergence.

Lemma 4.7 *The Jacobian*

$$F'(v, y) := \begin{bmatrix} \mathrm{Diag}\,(c - A^T y)N & \mathrm{Diag}\,(\hat{x} + Nv)A^T \end{bmatrix}$$

is Lipschitz continuous with constant

$$\gamma = \sigma_{\max}(F') \leq \sqrt{2}\|A\|\|N\|, \tag{4.26}$$

where $\sigma_{\max}(F')$ is the largest singular value of the linear transformation $F' : \mathbb{R}^n \to \mathbb{R}^{n\times n}$.

Proof. For each $s = (v, y) \in \mathbb{R}^n$ we get the matrix $F'(s) \in \mathbb{R}^{n\times n}$. This mapping is denoted by the linear transformation $F' : \mathbb{R}^n \to \mathbb{R}^{n\times n}$. The largest singular value of the matrix representation is denoted $\sigma_{\max} := \sigma_{\max}(F')$. This satisfies $\|F'(s) - F'(\bar{s})\| = \|F'(s - \bar{s})\| \leq \sigma_{\max}\|s - \bar{s}\|$, i.e. by setting $s = 0$ and $\bar{s}$ to be the singular vector corresponding to the largest singular value, we conclude $\gamma = \sigma_{\max}$.

Now let $ds = \begin{bmatrix} dv \\ dy \end{bmatrix}$. Since

$$\begin{aligned}
\|F'(s) - F'(\bar{s})\| &= \max \frac{\|(F'(s) - F'(\bar{s}))ds\|}{\|ds\|} \\
&= \max \frac{\|\text{Diag}\,(A^T(y - \bar{y}))Ndv - \text{Diag}\,(A^T dy)N(v - \bar{v})\|}{\|ds\|} \\
&\leq \max \frac{\|A^T(y - \bar{y})\|\|Ndv\| + \|A^T dy\|\|N(v - \bar{v})\|}{\|ds\|} \\
&\leq \|A\|\|N\|\|y - \bar{y}\| + \|A\|\|N\|\|v - \bar{v}\| \\
&\leq \sqrt{2}\|A\|\|N\|\|s - \bar{s}\|.
\end{aligned}$$

Therefore a Lipschitz constant is

$$\gamma = \sqrt{2}\|A\|\|N\|.$$

■

Observe that the Lipschitz constant depends on the representation matrix N that we consider. In particular, N can be chosen so that its columns are orthonormal and $\|Ndv\| = \|dv\|$ and $\|N(v - \bar{v})\| = \|v - \bar{v}\|$. In this case, the Lipschitz constant $\gamma \leq \sqrt{2}\|A\|$.

We can evaluate the largest singular value $\sigma_{\max}$ in the above Theorem 4.6 as follows. Consider the linear transformation $\mathcal{L} : \mathbb{R}^n \mapsto \mathbb{R}^{n^2}$ defined by

$$\mathcal{L}\begin{bmatrix} v \\ y \end{bmatrix} := \text{vec}\,([\text{Diag}\,(A^T y)N \ \text{Diag}\,(Nv)A^T]),$$

where $\text{vec}\,(M)$ denotes the vector formed column-wise from the matrix M. The inverse of

vec is denoted Mat . Let $w \in \mathbb{R}^{n^2}$. The inner-product

$$\begin{aligned} < \mathcal{L} \begin{bmatrix} v \\ y \end{bmatrix}, w > &= < \mathrm{vec}\,([\mathrm{Diag}\,(A^T y) N \ \mathrm{Diag}\,(Nv) A^T]), w > \\ &= < \begin{bmatrix} v \\ y \end{bmatrix}, \begin{bmatrix} N^T \mathrm{diag}\,(A^T W_2^T) \\ A\,\mathrm{diag}\,(N W_1^T) \end{bmatrix} > \end{aligned}$$

where W_1 is the first $n - m$ columns of $\mathrm{Mat}\,(w)$ and W_2 is the remaining m columns of $\mathrm{Mat}\,(w)$. Therefore, the adjoint operator of $\mathcal{L}$ is

$$\mathcal{L}^*(w) = \begin{bmatrix} N^T \mathrm{diag}\,(A^T W_2^T) \\ A\,\mathrm{diag}\,(N W_1^T) \end{bmatrix}.$$

We can use a few iterations of the power method to efficiently approximate the largest eigenvalue of $\mathcal{L}^*\mathcal{L}$ (which is the equivalent to the square of the largest singular value of $\mathcal{L}$). This can be done without forming the matrix representation of $\mathcal{L}$.

We also need to estimate β, the bound on the norm of the inverse of the Jacobian matrix at the current $s = (v, y)$, i.e.

$$\beta \geq \|[ZN \ - XA^T]^{-1}\| = 1/\sigma_{\min}([ZN \ - XA^T]). \tag{4.27}$$

Finally, to estimate η, we note that

$$\|J^{-1} F_0(v, y)\| = \|[ZN \ - XA^T]^{-1}(-ZXe)\| \leq \eta. \tag{4.28}$$

The vector $[ZN \ - XA^T]^{-1}(-ZXe)$ is the affine scaling direction and is available within the predictor-corrector approach that we use.

We now have the following heuristic for our crossover technique.

Theorem 4.8 *With the notation in Theorem 4.6 and $s_0 = (v_0, y_0)$, suppose that we have estimated the three constants γ, β, η in* (4.26),(4.27),(4.28). *And, suppose that*

$$\alpha = \gamma\beta\eta < \frac{1}{2}.$$

Then the sequence s_k generated by

$$s_{k+1} = s_k - J(s_k)^{-1} F(s_k)$$

converges to s^*, the unique zero of F in the neighbourhood $\mathcal{N}(s_0, r_1)$.

Remark 4.9 *Theorem 4.8 guarantees convergence of the affine scaling direction without backtracking. But it does not guarantee convergence to a nonnegative solution. Nonetheless, all our numerical tests were successful.*

4.3.3 Purify Step

Purifying here refers to detecting the variables that are zero at optimality. This is equivalent to identifying active constraints, e.g. [15, 16, 17]. We use the Tapia indicators [27] to detect the x variables going to zero. (See also [69, 6].) This is more difficult than the crossover step, as variables can increase and decrease while converging to 0, see e.g. [42].

Once we identified a x variable going to zero, we can remove the corresponding columns in A and shrink the data. For example, assume $\tilde{\mathcal{N}}$ is the index set of x variables that has been detected to goto zero in the current iteration. $\tilde{\mathcal{B}}$ is the rest indices. At the next iteration, we will make the input to the infeasible interior point method to be $A_{\tilde{\mathcal{B}}}, b, c_{\tilde{\mathcal{B}}}$ with initial point of $x_{\tilde{\mathcal{B}}}, y, z_{\tilde{\mathcal{B}}}$. Since we drop those x variables going to zero, the infeasibility at next iteration is small. The infeasible interior point method is easy to correct such infeasibility. To keep the $[S\ E]$ structure of our data matrix A, we only limit our choice of $\tilde{\mathcal{N}}$ corresponding to the E columns.

4.4 Numerical Tests

Our numerical tests use the well known NETLIB library as well as randomly generated data.

Our randomly generated problems use data A, b, c, with a known optimal basis in A and optimal values x, y, and z. For the infeasible code tests, we used the same starting point strategy given in LIPSOL [122]. For the feasible code tests we applied a Newton step from the optimal point with a large positive μ, in order to maintain feasibility of the starting point. In addition, we ensure that the Jacobian of the optimality conditions at the optimum is nonsingular and its condition number is not *large*, since we want to illustrate how the stable system takes advantage of well-conditioned problems. The iteration is stopped when the relative duality gap (including the relative infeasibility) is less than 10^{-12}. The computations were done in MATLAB 6.5 on a Pentium 3 733MHz running Windows 2000 with 256MB RAM.

data	m	n	nnz(E)	cond($A_{\mathcal{B}}$)	cond(J)	NEQ		Stable direct	
						D_time	its	D_Time	its
1	100	200	1233	51295	32584	0.03	$*$	0.06	6
2	200	400	2526	354937	268805	0.09	6	0.49	6
3	200	400	4358	63955	185503	0.10	$*$	0.58	6
4	400	800	5121	14261771	2864905	0.61	$*$	3.66	6
5	400	800	8939	459727	256269	0.64	6	4.43	6
6	800	1600	10332	11311945	5730600	5.02	6	26.43	6
7	800	1600	18135	4751747	1608389	5.11	$*$	33.10	6

Table 4.1: nnz(E) - number of nonzeros in E; cond($\cdot$) - condition number; $J = (ZN - XA^T)$ at optimum, see (4.24); D_time - avg. time per iteration for search direction, in sec.; its - iteration number of interior point methods. * denotes NEQ stalls at relative gap 10^{-11}.

We use both a direct and iterative method for finding the search direction. The direct solver uses the "LU($\cdot$)" function in MATLAB to find an LU factorization of the matrix. It then uses the MATLAB \ (backslash) command applied to the LU factorization in both the predictor and corrector step to solve the linear system. (We note that using "LU($\cdot$)" is generally slower than using \ (backslash) directly on the linear system.) The iterative approach uses LSQR [79] with different preconditioners. Both the direct and iterative based method share the exact same interior point framework except for the method used for computing the search direction and the inclusion of crossover and purify steps.

The normal equation, NEQ, approach uses the "CHOL($\cdot$)" function in MATLAB to find a Cholesky factorization of $AZ^{-1}XA^T$. It then uses the Cholesky factor with the MATLAB \ (backslash) in both the predictor and corrector step. (We note that using "CHOL($\cdot$)" is generally three times slower than using \ (backslash) directly on NEQ.) The NEQ approach can solve many of the random generated problems to the required accuracy. However, if we set the stop tolerance to 10^{-15}, we do encounter quite a few examples where NEQ stalls with relative gap approximately 10^{-11}, while the stable system has no problem reaching the desired accuracy.

The tests in Tables 4.1, 4.2, and 4.3 are done without the crossover and purification tech-

data set	LSQR with ILU				LSQR with Diag			
	D_Time	its	L_its	Pre_time	D_Time	its	L_its	Pre_time
1	0.15	6	37	0.06	0.41	6	556	0.01
2	3.42	6	343	0.28	2.24	6	1569	0.00
3	2.11	6	164	0.32	3.18	6	1595	0.00
4	NA	Stalling	NA	NA	13.37	6	4576	0.01
5	NA	Stalling	NA	NA	21.58	6	4207	0.01
6	NA	Stalling	NA	NA	90.24	6	9239	0.02
7	NA	Stalling	NA	NA	128.67	6	8254	0.02

Table 4.2: Same data sets as in Table 4.1; two different preconditioners (diagonal and incomplete Cholesky with drop tolerance 0.001); D_time - average time for search direction; its - iteration number of interior point methods. L_its - average number LSQR iterations per major iteration; Pre_time - average time for preconditioner; Stalling - LSQR cannot converge due to poor preconditioning.

data set	LSQR with block Chol. Precond.			
	D_Time	its	L_its	Pre_time
1	0.09	6	4	0.07
2	0.57	6	5	0.48
3	0.68	6	5	0.58
4	5.55	6	6	5.16
5	6.87	6	6	6.45
6	43.28	6	5	41.85
7	54.80	6	5	53.35

Table 4.3: Same data sets as in Table 4.1; LSQR with Block Cholesky preconditioner; Notation is the same as Table 4.2.

nique. The stable method with the direct solver and also with the diagonal preconditioner consistently obtain high accuracy optimal solutions. The stable method is not competitive in terms of time compared to the NEQ approach for this test set. One possible reason is that the condition numbers of J, the Jacobian at the optimum, and of the basis matrix $A_{\mathcal{B}}$, are still too large so that the iterative method is not effective. We provide another set of numerical tests based on well conditioned $A_{\mathcal{B}}$ in the following subsection.

We also performed many tests with the crossover. Using our crossover criteria in Theorem 4.8 with the inexpensive bound for γ, we can usually detect the guaranteed convergence region at $\mu = 10^{-6}$ or with the relative gap tolerance at 10^{-4} or 10^{-5}. We also encounter a few examples where the crossover begins as early as $\mu = 10^{-4}$ and some examples that the crossover begins as late as $\mu = 10^{-8}$. Once the crossover criteria is detected, we use a pure Newton step, i.e. we use the affine scaling direction with step length 1 without limiting x and z to be positive. It usually takes only one iteration to achieve the required accuracy 10^{-12}. This is not a surprise considering the quadratic convergence rate of Newton's method. This behaviour has some similarity with the least squares projection method discussed by Ye [117] and Vavasis and Ye [105]. They proved that an exact optimal solution on the optimal face can be found by solving a least squares problem when the iterations are in the final stage of the interior point method.

If we compare to the method without a crossover, then we conclude that the crossover technique gives an average 1 iteration saving to achieve the desired accuracy. We also encountered several instances where NEQ did not converge after we detected the crossover; while our stable method had no difficulty. We should mention that NEQ is not suitable for crossover since the Jacobian becomes singular. Moreover, a catastrophic error can occur if a z element becomes zero.

We also tested the purification technique. It showed a benefit for the stable direction when n was large compared to m, since we only identify nonbasic variables. (However, the size of NEQ $AXZ^{-1}A^T$ is still $m \times m$. So there is no benefit there.) The time saving on solving the linear system for the stable direction is cubic in the percentage of variables eliminated, e.g. if half the variables are eliminated, then the time is reduced to $(\frac{1}{2})^3 = \frac{1}{8}$ the original time. The purification technique starts to identify nonbasic variables as early as 6-7 iterations before convergence. It usually identifies most of the nonbasic variables from two

to four iterations before convergence. For all our random generated tests, the purification technique successfully identified all the nonbasic variables before the last two iterations.

We should also mention the computation costs. For the crossover, we need to evaluate the smallest singular value of a sparse $n \times n$ matrix to find β, and then solve an $n \times n$ linear system to find the value η. The cost of finding the smallest singular value is similar to that of solving a system of the same size. Solving this linear system is inexpensive since the *matrix* is the same as the one for the search direction.

In the above tests we restricted ourselves to non-degenerate problems. See Figure 4.1 for a comparison on a typical degenerate problem. Note that NEQ had such difficulties on more than half of our degenerate test problems. This is consistent with our analysis in Chapter 3, in which we suggest that NEQ have problems after 10^{-8} for degenerate problems with $\text{rank}\,(A_{\mathcal{B}}) < m$.

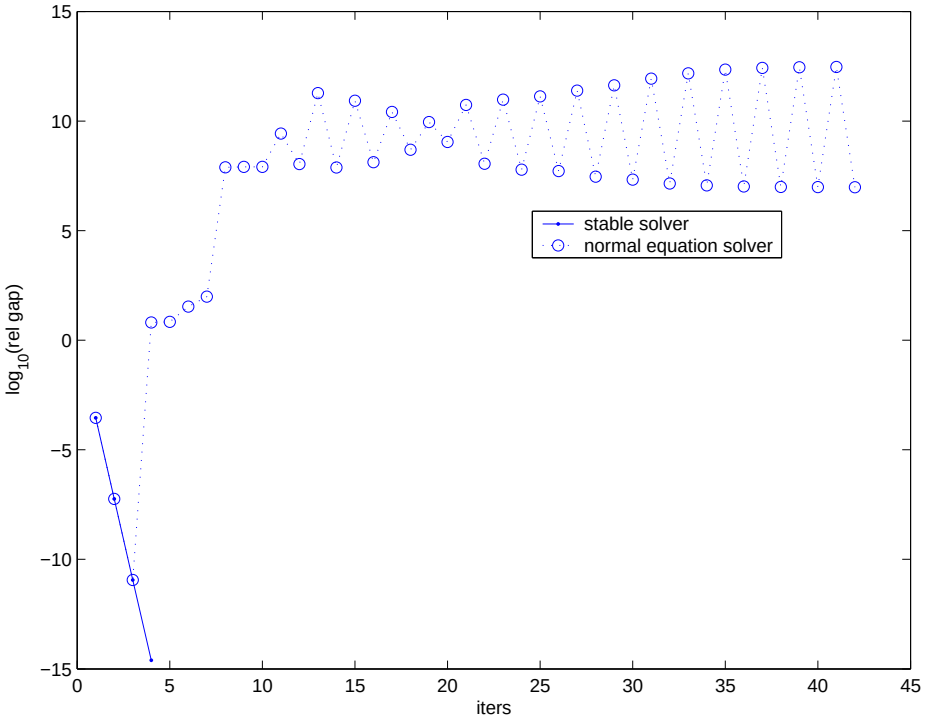

Figure 4.1: Iterations for Degenerate Problem

4.4.1 Well Conditioned $A_{\mathcal{B}}$

Our previous test examples in Tables 4.1,4.2,4.3 are all sparse with 10 to 20 nonzeros per row. In this section we generate *sparser* problems with about 3-4 nonzeros per row in E but we still maintain non singularity of the Jacobian at the optimum. We first fix the indices of a basis $\mathcal{B}$; we choose half of the column indices j so that they satisfy $1 \leq j \leq m$ and the other half satisfies $m+1 \leq j \leq n$. We then add a random diagonal matrix to $A_{\mathcal{B}}$ to obtain a well-conditioned basis matrix and generate two random (sufficiently) positive vectors $x_{\mathcal{B}}$ and $z_{\mathcal{N}}$. We set the optimal $x^* = \begin{bmatrix} x_{\mathcal{B}} \\ x_{\mathcal{N}} \end{bmatrix}$ with $x_{\mathcal{N}} = 0$; and the optimal $z^* = \begin{bmatrix} z_{\mathcal{B}} \\ z_{\mathcal{N}} \end{bmatrix}$, with $z_{\mathcal{B}} = 0$. The data b, c are determined from $b := Ax^*, \quad c := A^T y^* + z^*, \quad y^* \in \mathbb{R}^m$ arbitrary.

We now compare the performance of three different solvers for the search direction, i.e. NEQ solver, direct linear solver on the stable system, and LSQR on the stable system. In this section, we restrict ourselves to the diagonal preconditioner when we use the LSQR solver. (The computations in this section were done on a Sun-Fire-480R running SunOS 5.8.)

The problems in Table 4.4 all have the same dimensions and two full dense columns, while the total number of nonzeros increases. The loss in sparsity has essentially no effect on NEQ, since the ADA^T matrix is dense due to the two dense columns. But we can see the negative effect that the loss of sparsity has on the stable direct solver, since the density in the system (4.20) increases. However, we see that for these problem instances, using LSQR with the stable system can be up to twenty times faster than the NEQ solver.

Our second test set in Table 4.5 shows how size affects the three different solvers. The time for the NEQ solver is proportional to m^3. The stable direct solver is about twice that of NEQ. LSQR is the best among these 3 solvers on these instances. The computational advantage of LSQR becomes more apparent as the dimension grows.

Our third test set in Table 4.6 shows how the number of dense columns affects the different solvers. Having at least one dense column affects the direct solvers the most. LSQR spends more time when the number of dense columns increase, but this is due to the increased number of nonzeros.

We also use the well-known Matlab based linear programming solver LIPSOL to solve our test problems, see Table 4.7. Our tests use LIPSOL's default settings except that the

data sets				NEQ		Stable Direct		LSQR		
Name	cond($A_{\mathcal{B}}$)	cond(J)	nnz(E)	D_Time	its	D_Time	its	D_Time	its	L_its
nnz2	19	13558	4490	3.75	7	5.89	7	0.19	7	81
nnz4	21	19540	6481	3.68	7	7.38	7	0.27	7	106
nnz8	28	10170	10456	3.68	7	11.91	7	0.42	7	132
nnz16	76	11064	18346	3.69	7	15.50	7	0.92	7	210
nnz32	201	11778	33883	3.75	9	18.43	9	2.29	8	339

Table 4.4: *Sparsity vs Solvers:* cond(·) - (rounded) condition number; D_time - average time for search direction; its - number of iterations; L_its - average number LSQR iterations per major iteration; All data sets have the same dimension, 1000×2000, and have 2 dense columns.

data sets				NEQ		Stable Direct		LSQR	
name	size	cond($A_{\mathcal{B}}$)	cond(J)	D_Time	its	D_Time	its	D_Time	its
sz1	400×800	20	2962	0.29	7	0.42	7	0.07	7
sz2	400×1600	15	2986	0.29	7	0.42	7	0.11	7
sz3	400×3200	13	2358	0.30	7	0.43	7	0.19	7
sz4	800×1600	19	12344	1.91	7	3.05	7	0.13	7
sz5	800×3200	15	15476	1.92	7	3.00	7	0.27	7
sz6	1600×3200	20	53244	16.77	7	51.52	7	0.41	7
sz7	1600×6400	16	56812	16.70	7	51.75	7	0.65	8
sz8	3200×6400	19	218664	240.50	7	573.55	7	0.84	7
sz9	6400×12800	24	8.9×10^5					2.20	6
sz10	12800×25600	22	2.4×10^5					4.67	6

Table 4.5: *How problem dimension affects different solvers.* cond(·) - (rounded) condition number; D_time - average time for search direction; its - number of iterations. All the data sets have 2 dense columns. The sparsity for the data sets are similar. Without the 2 dense columns, they have about 3 nonzeros per row.

data sets				NEQ		Stable Direct		LSQR	
name	dense cols.	cond($A_{\mathcal{B}}$)	cond(J)	D_Time	its	D_Time	its	D_Time	its
den0	0	18	45	0.60	6	1.31	6	0.14	6
den1	1	19	13341	3.64	7	6.15	7	0.17	7
den2	2	19	18417	3.62	7	6.03	7	0.20	7
den3	3	19	19178	3.65	7	6.08	7	0.23	7
den4	4	18	18513	3.65	7	6.06	7	0.30	7

Table 4.6: *How number of dense columns affect different solvers.* cond(·) - (rounded) condition number; D_time - average time for search direction; its - number of iterations. All the data sets are the same dimension, 1000×2000. The sparsity for the data sets are similar. Without the dense columns, they all have about 3 nonzeros per row.

stop tolerance is set to 10^{-12}. Note that LIPSOL has a special routine to deal with dense columns by default. LIPSOL uses an infeasible code, so we can see that the numbers of iterations of the interior point method are in a different range from our tests in Tables 4.4, 4.5, 4.6, which are usually in the range of 6-8. It can be observed that LIPSOL in general performs better than the NEQ code we have written. Considering that LIPSOL has some special code to deal with factorization, while our code of direct method just uses the LU(or Chol) factorization from Matlab, it is not unusual to see the better performance of LIPSOL.

But comparing to the iterative method, we should mention that when the problem size becomes large, the iterative method has an obvious advantage over the direct factorization method. This can be seen clearly from the solving time of problems sz8, sz9, sz10 in Table 4.7 and the corresponding time of "LSQR" in Table 4.5. When the problem size doubles, the solution time for LIPSOL increases roughly by a factor of 8-10, while the solution time for our iterative method only roughly doubles. This is also true for fully sparse problems as mentioned in the Caption part of Table 4.7.

The iterative solver LSQR does not spend the same amount of time at different stages of an interior point method. To illustrate this, we take the data set in Table 4.4. For each problem we draw the number of LSQR iterations at each iteration, see Figure 4.2.

data sets	lipsol	
name	D_Time	its
nnz2	0.08	12
nnz4	0.50	14
nnz8	1.69	14
nnz16	2.72	14
nnz32	3.94	13
sz1	0.16	11
sz2	0.15	13
sz3	0.15	14
sz4	0.05	12
sz5	0.03	14
sz6	0.22	15
sz7	0.06	15
sz8	1.55	14
sz9	12.80	15
sz10	126.47	15
den0	0.06	10
den1	0.06	12
den2	0.08	13
den3	0.09	13
den4	0.10	12

Table 4.7: *LIPSOL results* D_time - average time for search direction; its - number of iterations. (We also tested problems sz8,sz9,sz10 with the two dense columns replaced by two sparse columns, only 6 nonzeros in these new columns. (D_time, iterations) on LIPSOL for these three fully sparse problems are: $(0.41, 11)$, $(2.81, 11)$, $(43.36, 11)$.)

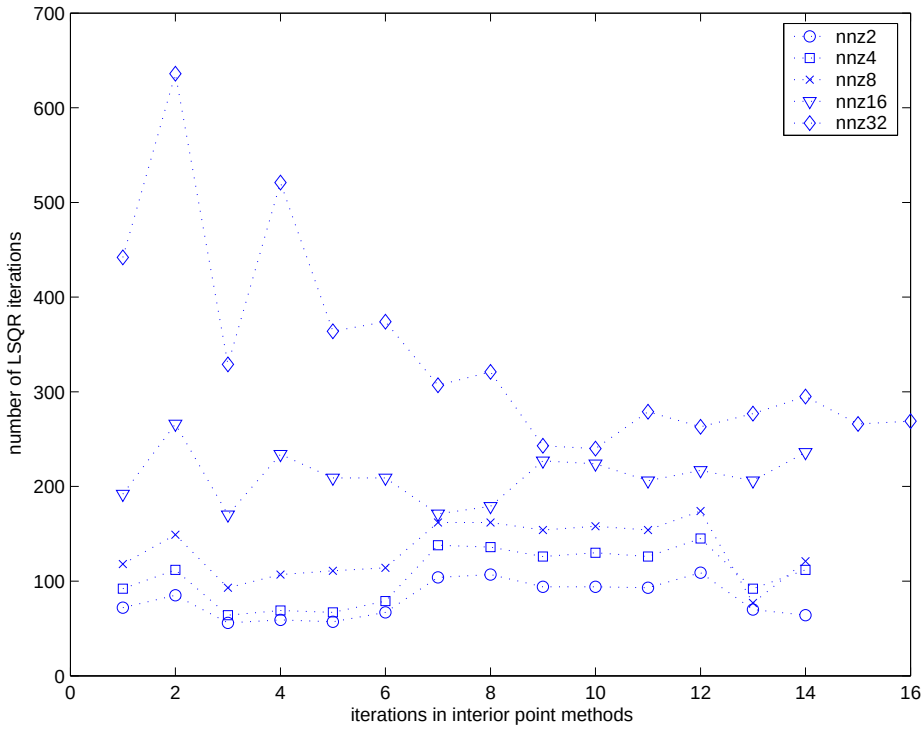

Figure 4.2: Illustration for LSQR iterations at different stage of interior point methods for the data set in Table 4.4. Each major iteration in interior point method is divided into a predictor step and a corrector step.

4.4.2 NETLIB Set - Ill-conditioned Problems

The NETLIB LPdata set is made up of highly degenerate problems which result in singular Jacobians. Nevertheless, we applied our method to these problems. The ill-conditioning of the linear systems negatively affects the performance of the algorithm when using iterative methods. A direct factorization method with our stable system is better suited for the NETLIB set.

For general LP problems, we want to find an S which is sparse and easy to invert in the $[S\ E]$ structure. An upper triangular matrix is a good choice. The heuristic we use is to go through the columns of the matrix A and find those columns that only have one nonzero entry. We then permute the columns and rows so that these nonzero entries are

on the diagonal of S. (In the case of multiple choices in one row, we picked the one with the largest magnitude.) We remove the corresponding rows and columns, and then repeat the procedure on the remaining submatrix. If this procedure is successful, we end up with an upper triangular matrix S. However, sometimes, we may have a submatrix $\hat{A}$ of A such that no column has one nonzero entry. Usually, such a submatrix $\hat{A}$ is much smaller in size. We use an LU factorization on this small submatrix and find an upper triangular part $\hat{U}$ in the U part of the LU factorization by using the above procedure. The S is then determined by incorporating those columns of $\hat{U}$ after an appropriate permutation. This procedure also results in a useful LU factorization for S. In our tables, we denote the row dimension of the $\hat{A}$ as *no-tri-size of* S. For NETLIB problems, surprisingly, most of them have a zero no-tri-size of S as shown in Tables 4.9–4.11. It is worth noting that some of the NETLIB problems may not have full row rank or the LU factorization on the submatrix $\hat{A}$ may not give an upper triangular U. Thus we may not be able to identify the upper triangular matrix $\hat{U}$. In Tables 4.9–4.11, these problems are marked with a " *" in the column of no-tri-size of S. For these problems, our solver may not give a correct answer. (This issue can be resolved by preprocessing to eliminate all redundant rows and by a better LU factorization. This is beyond the scope of this chapter.) Among these singular problems, "bore3d" and "standgub" have a complete zero row; thus we can easily identify the linearly dependent row in the matrix A and remove it. Our answers for these two problems are accurate.

To make a fair comparison on the errors, we changed the error term in LIPSOL to be the same as ours, which is defined as

$$\text{error} := \frac{|c^T x - b^T y|}{1 + |c^T x|} + \frac{\|r_p\|}{1 + \|b\|} + \frac{\|r_d\|}{1 + \|c\|}. \tag{4.29}$$

Note that LIPSOL can solve all the NETLIB problems to 10^{-8}. In addition, we added the preprocessing step that LIPSOL is using to our code.

We observed improved robustness when using our stable direct factorization method. For example, when the stop tolerance is set to 10^{-12}, LIPSOL could not solve the subset of NETLIB problems in Table 4.8 and, incorrectly, finds that several problems are infeasible. Table 4.8 lists the highest accuracy that LIPSOL can get. (LIPSOL does solve problems *fit1p, fit2p, seba* when the stop tolerance is set to 10^{-8} and does solve problems *bnl2, dfl001, greenbea* with tolerance 10^{-8} and its own error term.) This illustrates the numerical diffi-

NETLIB problems	Accuracy
bnl2	infeasible
cycle	9.19×10^{-11}
dfl001	infeasible
etamacro	7.66×10^{-11}
fit1p	infeasible
fit2p	infeasible
greenbea	infeasible
grow15	4.35×10^{-10}
grow22	9.24×10^{-10}
grow7	2.62×10^{-10}
kb2	3.75×10^{-12}
pilot87	1.21×10^{-8}
seba	infeasible

Table 4.8: LIPSOL failures with desired tolerance 10^{-12}; highest accuracy attained by LIPSOL.

culties that arise for NEQ based methods when the requested accuracy is more than 10^{-8}. Our stable direct factorization method not only achieved the desired accuracy (except for *capri* with $1.2e{-}12$, *pilot.ja* with $3.7e{-}12$, *pilot* with $6.7e{-}12$) but also exhibited quadratic convergence during the final few iterations on these problems. For complete results on the NETLIB problem, see Tables 4.9–4.11. (Further numerical tests appear in the forthcoming [83] and in the recent Masters thesis [82]. In [82, 83], a different transformation on the NETLIB problems is used to obtain the $[I\ E]$ structure. The numerical tests on the NETLIB problems in [82, 83] show that the ill-conditioning negatively affects the performance of the stable algorithm. However, it was also observed that much more accurate solutions were obtained by using the stable linearization approach compared to NEQ.)

problems	LIPSOL			Stable Direct			
Name	D_time	its	error	D_time	its	error	no-tri-size of S
25fv47	0.05	25	1.21e-14	0.94	24	8.7e-15	2
80bau3b	0.14	41	4.38e-14	2.84	49	5.5e-13	0
adlittle	0.01	12	4.13e-14	0.01	12	3.7e-16	2
afiro	0.01	8	3.70e-15	0.00	8	3.5e-16	0
agg	0.03	19	1.06e-13	0.10	19	4.5e-13	0
agg2	0.03	17	1.28e-13	0.19	17	1.4e-15	0
agg3	0.03	17	2.38e-15	0.18	16	1.4e-13	0
bandm	0.01	20	1.77e-14	0.05	17	2.3e-15	0
beaconfd	0.01	13	3.64e-14	0.04	13	3.0e-15	0
blend	0.01	12	8.32e-13	0.01	12	3.4e-15	0
bnl1	0.02	28	2.32e-14	0.37	27	3.0e-14	8
bnl2	0.08	7	2.40e+01	2.01	51	7.3e-13	0
boeing1	0.03	22	1.46e-13	0.14	23	4.7e-15	0
boeing2	0.01	20	1.46e-14	0.03	17	7.9e-13	0
bore3d	0.01	18	9.62e-14	0.03	18	3.3e-14	4*
brandy	0.01	17	8.37e-15	0.04	15	4.2e-13	52
capri	0.02	19	2.76e-13	0.06	20	1.2e-12	0
cycle	0.12	36	9.19e-11	1.98	29	2.5e-13	4
czprob	0.03	36	7.91e-14	1.06	34	7.1e-13	0
d2q06c	0.18	33	1.92e-14	6.21	30	2.1e-13	132*
d6cube	0.11	25	1.23e-15	3.54	14	4.8e-14	404*
degen2	0.03	14	3.62e-13	0.14	13	2.4e-15	97*
degen3	0.25	29	1.22e-13	2.02	17	3.8e-13	159*
dfl001	19.63	17	2.28e+00	46.65	52	1.0e+01	4275*
e226	0.01	22	1.05e-13	0.06	21	3.7e-13	0
etamacro	0.02	45	7.66e-11	0.11	37	7.3e-13	16
fffff800	0.03	27	9.21e-14	0.21	25	4.1e-14	0
finnis	0.02	30	7.40e-13	0.08	27	8.6e-13	0
fit1d	0.04	24	4.18e-13	0.50	18	9.2e-15	0
fit1p	0.30	17	1.75e-05	0.25	16	9.2e-14	0
fit2d	0.43	26	7.05e-13	80.99	23	8.4e-15	0
fit2p	0.68	22	2.35e-07	5.76	23	5.1e-14	0
forplan	0.02	23	1.98e-13	0.09	28	7.9e-13	0

Table 4.9: NETLIB set with LIPSOL and Stable Direct method. D_time - avg. time per iteration for search direction, in sec.; its - iteration number of interior point methods.

problems	LIPSOL			Stable Direct			
Name	D_time	its	error	D_time	its	error	no-tri-size of S
ganges	0.04	19	5.14e-14	0.28	20	9.6e-13	12
gfrd-pnc	0.02	20	3.53e-14	0.1	20	9.9e-15	0
greenbea	0.24	32	6.01e-04	5.68	45	4.6e-13	2
greenbeb	0.15	38	2.01e-13	5.49	37	6.1e-14	2
grow15	0.03	31	4.35e-10	0.86	12	2.4e-13	0
grow22	0.04	25	9.24e-10	2.27	14	4.3e-14	0
grow7	0.02	37	2.62e-10	0.16	12	2.2e-15	0
israel	0.02	23	5.06e-13	0.04	23	9.6e-14	0
kb2	0.01	34	3.75e-12	0.01	16	1.1e-14	0
lotfi	0.01	19	1.51e-15	0.05	17	9.5e-13	0
maros-r7	2.03	15	1.43e-15	14.97	15	1.3e-15	0
maros	0.05	33	5.24e-13	0.59	31	1.1e-13	4
modszk1	0.02	25	3.23e-13	0.22	68	9.8e-13	0
nesm	0.06	35	1.45e-13	2.77	32	7.3e-13	0
perold	0.04	32	5.66e-13	0.71	37	6.4e-13	0
pilot.ja	0.30	33	2.63e-13	1.34	35	3.7e-12	0
pilot	0.07	35	7.72e-13	13.69	42	6.7e-12	0
pilot.we	0.04	36	7.61e-13	0.95	40	4.5e-15	0
pilot4	0.03	31	1.80e-13	0.3	31	1.5e-13	0
pilot87	0.80	99	1.21e-08	27.58	42	2.8e-15	0
pilotnov	0.06	20	1.73e-13	1.86	24	1.3e-13	0
recipe	0.01	11	1.32e-13	0.01	11	6.1e-15	0
sc105	0.01	11	4.42e-16	0.01	10	6.0e-16	0
sc205	0.01	11	2.26e-13	0.02	10	7.2e-13	0
sc50a	0.01	10	3.34e-15	0.01	10	5.3e-16	0
sc50b	0.01	8	1.35e-15	0.01	8	6.1e-16	0
scagr25	0.01	17	7.46e-15	0.04	16	3.0e-15	0
scagr7	0.01	13	2.50e-13	0.01	13	7.5e-16	0
scfxm1	0.01	18	1.79e-13	0.06	18	2.0e-15	8
scfxm2	0.02	21	4.24e-14	0.13	20	3.3e-15	16
scfxm3	0.03	21	1.21e-14	0.19	20	3.5e-15	24
scorpion	0.01	15	1.99e-13	NA	NA	NA	132*
scrs8	0.02	26	7.17e-13	0.1	25	6.2e-13	0
scsd1	0.01	10	6.40e-13	0.12	11	3.3e-14	0

Table 4.10: NETLIB set with LIPSOL and Stable Direct method continued

problems	LIPSOL			Stable Direct			
Name	D_time	its	error	D_time	its	error	no-tri-size of S
scsd6	0.02	15	7.31e-15	0.42	15	6.1e-15	0
scsd8	0.03	12	1.07e-14	2.64	13	2.2e-15	0
sctap1	0.01	17	5.67e-13	0.05	18	2.6e-14	0
sctap2	0.03	19	7.33e-13	0.27	16	1.9e-15	0
sctap3	0.04	18	1.46e-13	0.36	21	1.9e-15	0
seba	0.10	23	8.39e-07	0.1	17	7.4e-15	0
share1b	0.01	21	1.92e-13	0.03	24	5.5e-15	66
share2b	0.01	14	5.69e-15	0.01	12	1.2e-14	0
shell	0.02	20	1.61e-15	0.04	12	1.2e-15	494*
ship04l	0.02	13	1.88e-13	0.24	13	1.9e-15	0
ship04s	0.02	14	2.76e-13	0.14	13	1.7e-15	0
ship08l	0.04	16	3.34e-15	0.49	16	2.4e-15	0
ship08s	0.02	14	2.47e-13	0.2	15	2.0e-15	0
ship12l	0.05	17	9.98e-13	0.62	17	1.0e-14	0
ship12s	0.02	19	3.94e-15	0.21	16	3.7e-15	0
sierra	0.06	17	1.50e-13	0.17	12	5.5e-15	515*
stair	0.02	15	2.93e-13	0.1	14	4.8e-13	0
standata	0.02	17	1.62e-14	0.13	17	4.5e-15	0
standgub	0.02	17	5.15e-13	0.06	17	4.0e-15	1*
standmps	0.02	24	9.87e-14	0.19	23	1.7e-14	0
stocfor1	0.01	16	6.84e-13	0.01	19	3.9e-14	0
stocfor2	0.05	22	1.19e-13	0.32	22	1.8e-13	0
tuff	0.02	23	2.83e-16	0.13	20	1.4e-13	0
vtp.base	0.01	23	5.76e-13	0.03	27	3.5e-13	0
wood1p	0.15	21	4.37e-13	0.76	13	6.4e-14	241*
woodw	0.11	30	6.13e-13	41.59	30	9.6e-14	0

Table 4.11: NETLIB set with LIPSOL and Stable Direct method continued

4.4.3 No Backtracking

We now present some interesting numerical results under the condition that the interior point method takes a complete step to the boundary without the customary backtracking that guarantees sufficient positivity of the variables x, z. We present the results from the three algorithms: (i) NEQ with backtracking; (ii) stable system with backtracking; (iii) stable system with no backtracking. Since the NEQ approach is undefined at the boundary, we cannot include a fourth comparison. No backtracking does not create problems for our stable system, since we do not need the inverse of X or Z.

See Figure 4.3 for a comparison between NEQ with backtracking and the stable direction with and without backtracking. In this example, the relative gap stop tolerance for NEQ is set to 10^{-12}, which is the highest accuracy NEQ can get for this problem. However, the relative gap stop tolerances for both of the stable system approaches are set to 10^{-14}. For the first 4 iterations the three approaches are almost indistinguishable, since the backtrack (we backtrack with .9998) is such a small step. However, once the duality gap is small, no backtracking means we are close to taking a complete Newton step so we get a large improvement with the no-backtracking strategy. We reach the desired tolerance in 6 iterations compared to 8 for the stable direction with backtracking. The difference with using backtracking for the stable direction is typical; while stalling for NEQ occurs for about half our tests.

For many tests, we see that the number of iterations are reduced and the last step behaves just as if the crossover was implemented, i.e. we jump to the stopping tolerance of 14 decimals. This is probably due to the fact that a full step to the boundary is close to a full Newton step, i.e. this is comparable to implementing the crossover technique. On average, the stable direct method without backtracking results in a 1-2 reduction in the number of iterations.

4.5 Summary

We have studied a simple, robust alternative to solving LPs. The advantages of our approach are: the resulting linear system does not necessarily get ill-conditioned as we approach the optimum; this allows for the application of preconditioned iterative methods, a crossover

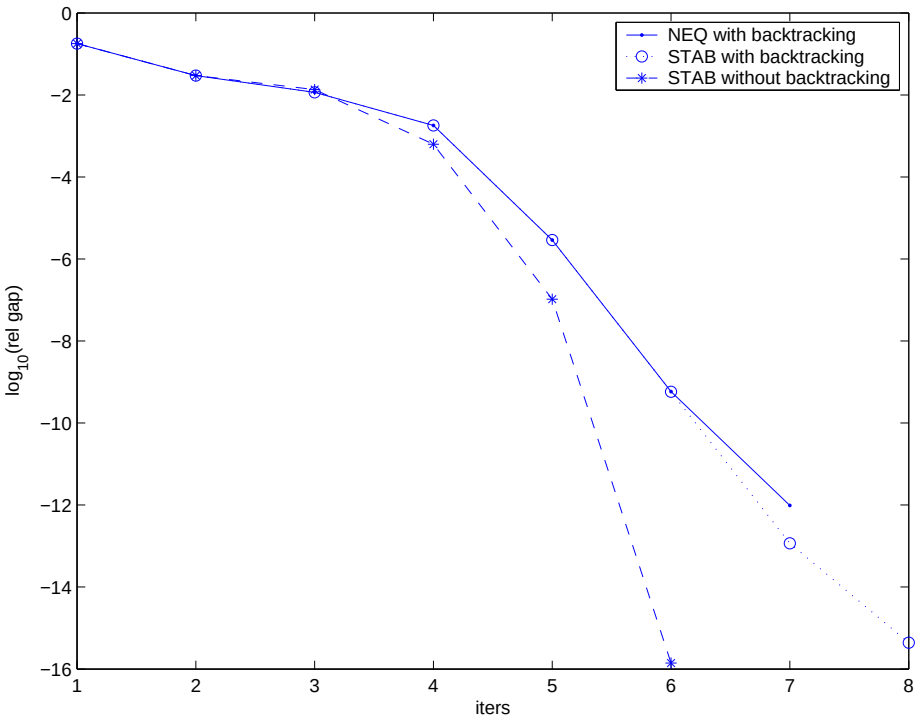

Figure 4.3: Iterations for Different Backtracking Strategies. The data is from row 2 in Table 4.1.

technique to affine scaling without backtracking, dynamic purification, and no backtracking from the boundary (taking the complete step to the boundary is advantageous); high accuracy solutions are obtained; and exact primal-dual feasibility is maintained throughout the iterations, if we start feasible.

Since our reduced linear system is larger than the usual normal equations approach, NEQ, our method is not competitive for the highly ill-conditioned NETLIB test set, with respect to CPU time, though we can obtain higher accuracy solutions.

In conclusion, we believe that the stable approach has some advantages, compared with the NEQ approach, for some applications where the nondegeneracy assumptions are satisfied or where higher accuracy solutions are needed. Our numerical tests show that we can take direct advantage of sparsity for large sparse well-conditioned problems. The NEQ approach has its advantages, the main one being the smaller size and the positive definiteness of the

linear system to solve (before the backsubstitutions).

Chapter 5

Fundamentals of Semidefinite Programming

5.1 Introduction to Semidefinite Programming

Similarly to the LP case, the primal and dual Semidefinite Programming (SDP) problem we consider is

$$\begin{array}{lllll} & \min & \operatorname{trace} CX & & \max \quad b^T y \\ \text{(PSDP)} & \text{s.t.} & \mathcal{A}(X) = b & \text{(DSDP)} & \text{s.t.} \quad \mathcal{A}^*(y) + Z = C \\ & & X \succeq 0 & & Z \succeq 0, \end{array} \tag{5.1}$$

where $C, X, Z \in \mathcal{S}^n$, $\mathcal{S}^n$ denotes the space of $n \times n$ real symmetric matrices, $y, b \in \mathbb{R}^m$, and $\succeq$ ($\succ$) denotes positive semidefiniteness (resp. positive definiteness). The linear operator $\mathcal{A} : \mathcal{S}^n \rightarrow \mathbb{R}^m$ is an onto linear transformation and $\mathcal{A}^*$ is the adjoint transformation.

SDP is a generalization of LP. SDP looks just like an LP and in fact, if we require X and Z to be diagonal matrices, then (PSDP) and (DSDP) are equivalent to standard LP problems. Many of the properties from LP follow through. For instance, weak duality holds, i.e., for any primal feasible solution $\bar{X}$ and any dual feasible solution $\bar{y}$ and $\bar{Z}$, we always have $\operatorname{trace} C\bar{X} \geq b^T \bar{y}$.

However, some important properties in SDP differ from those in LP. For example, just as in general convex programming, strong duality can fail. There may exist a nonzero duality

gap, or the optimal objective value of (PSDP) or (DSDP) may not be attained. The duality gap is assured to be zero if a constraint qualification, e.g., Slater's condition (strict feasibility) holds, see e.g. [87, 85]. Measure of strict feasibility, also called *distance to infeasibility*, have been used in complexity analysis, e.g., [86, 30, 31, 32]. The optimality conditions are

Theorem 5.1 *Suppose that Slater's condition holds for (PSDP) and (DSDP). The primal-dual variables (X, y, Z), with $X, Z \succeq 0$, are optimal for (PSDP) and (DSDP) if and only if*

$$F(X, y, Z) := \begin{bmatrix} \mathcal{A}^* y + Z - C \\ \mathcal{A}X - b \\ XZ \end{bmatrix} = 0. \tag{5.2}$$

Another important property is strict complementarity. In LP, there always exists a pair of strict complementary solutions as described by the well-known Goldman-Tucker result [40], see also Section 2.1. In SDP, strict complementary solutions do not necessary exist. Similar to the lack of strict feasibility, the lack of strict complementarity can result in both theoretical and numerical difficulties. For example, many of the local superlinear and quadratic convergence results for interior point methods depend on the strict complementarity assumption, e.g. [84, 50, 4, 64, 59]. Also, the convergence of the central path to the analytic center of the optimal face relies on strict complementarity, see [46]. However, it has been proved that strict complementarity holds generically, see [4] and [81].

5.2 Central Path

Similar to the LP case, we can add a barrier function to the objective function and thus we define a pair of families of perturbed barrier problems, parameterized by $\mu > 0$.

$$\begin{aligned} (SDP_\mu) \quad & \min \operatorname{trace}(CX) - \mu \ln \det X \\ & \mathcal{A}X = b \\ & (X \succ 0) \end{aligned}$$

$$\begin{aligned}(DSDP_\mu) \quad & \max b^T y + \mu \ln\det Z \\ & \mathcal{A}^* y + Z = C \\ & (Z \succ 0)\,.\end{aligned}$$

Here the $\ln\det(\cdot)$ operator takes the determinant of the matrix and then takes the natural logarithm of the determinant. Using some matrix calculus, for example, see the online Matrix Reference Manual [14], we see that (assume X is symmetric positive definite)

$$\det{}'X = \det X \cdot X^{-1},$$

and

$$-\ln\det{}'X = -X^{-1}.$$

The second derivative is

$$-\ln\det{}''X = -(X^{-1})' = X^{-1}(\cdot)X^{-1}.$$

Notice that $X^{-1}(\cdot)X^{-1}$ is positive definite because for any matrix $U \neq 0$, we have

$$\mathrm{trace}\,(X^{-1}(U)X^{-1}U) = \mathrm{trace}\,(X^{-1/2}UX^{-1/2})^2 > 0.$$

Thus the barrier function $-\ln\det(\cdot)$ is strictly convex. So, we have that the KKT conditions for the (SDP_μ) and $(DSDP_\mu)$ are both sufficient and necessary. The KKT conditions for both the (SDP_μ) and $(DSDP_\mu)$ are equivalent to (after an appropriate multiplication of Z or X to the third equation)

$$\begin{aligned}\mathcal{A}X &= b, \quad X \succ 0\,, \\ \mathcal{A}^* y + Z &= C, \quad Z \succ 0\,, \\ XZ &= \mu I\,.\end{aligned} \tag{5.3}$$

The existence and uniqueness of a solution for system (5.3) is stated in the following theorem.

Theorem 5.2 *Suppose both (SDP) and (DSDP) have strictly feasible solutions. Then for a fixed $\mu > 0$, there is a unique solution $X(\mu)$ of (SDP_μ) and unique solution $y(\mu)$, $Z(\mu)$ of $(DSDP_\mu)$. This solution $X(\mu)$, $y(\mu)$, $Z(\mu)$ is also the unique solution to system* (5.3).

The proof is similar to the LP case as shown in Theorem 2.5. For a complete proof, please see [72].

We call the set of solution $(X(\mu), y(\mu), Z(\mu))$ to system (5.3), the central path of SDP. However, we have not proved that the set of solution are analytic in μ. The *Implicit function theorem* can be used to prove the analyticity. (We used the version in Dieudonné [25, Theorem 10.2.4].)

Theorem 5.3 (Implicit function theorem) *Let $f : \mathbb{R}^{n+m} \mapsto \mathbb{R}^m$ be an analytic function of $w \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$ such that:*

1. *There exist $\bar{w} \in \mathbb{R}^n$ and $\bar{z} \in \mathbb{R}^m$ such that $f(\bar{w}, \bar{z}) = 0$.*
2. *The Jacobian of f with respect to z is nonsingular at $(\bar{w}, \bar{z})$.*

Then there exist open sets $S_{\bar{w}} \subset \mathbb{R}^n$ and $S_{\bar{z}} \subset \mathbb{R}^m$ containing $\bar{w}$ and $\bar{z}$ respectively, and an analytic function $\phi : S_{\bar{w}} \mapsto S_{\bar{z}}$ such that $\bar{z} = \phi(\bar{w})$ and $f(w, \phi(w)) = 0$ for all $w \in S_{\bar{w}}$. Moreover

$$\bigtriangledown \phi(w) = - \bigtriangledown_z f(w, \phi(w))^{-1} \bigtriangledown_w f(w, \phi(w)).$$

Remark 5.4 *To prove the analyticity of the central path, De Klerk[19] defines the function f as follows.*

$$f(X, y, Z, \mu) := \begin{bmatrix} \mathcal{A}^* y + Z - C \\ \mathcal{A}X - b \\ XZ - \mu I \end{bmatrix}. \tag{5.4}$$

He proceeds to argue that the symmetric requirement in the above equation is redundant on the central path because Z is symmetric by the symmetry of C and $\mathcal{A}^* y$ and X is symmetric by $XZ = \mu I$. Thus the function f (5.4) can be viewed as a function mapping $\mathbb{R}^{n^2} \times \mathbb{R}^m \times \mathbb{R}^{n^2} \times \mathbb{R} \mapsto \mathbb{R}^{n^2} \times \mathbb{R}^m \times \mathbb{R}^{n^2}$. He then argues that the Jacobian of f with respect to (X, y, Z) is nonsingular. Using the implicit function theorem, he shows that $(X(\mu), y(\mu), Z(\mu)$ is analytic.

Monteiro and Todd [72] use a slight different approach to show the analyticity of the central path. Instead of working on the function $f(X, y, Z, \mu)$ in (5.4), they define an equivalent

function

$$g(X, y, Z, \mu) := \begin{bmatrix} \mathcal{A}^* y + Z - C \\ \mathcal{A}X - b \\ Z - \mu X^{-1} \end{bmatrix}. \tag{5.5}$$

This function g maps from $S^n \times \mathbb{R}^m \times S^n \times \mathbb{R}$ to $S^n \times \mathbb{R}^m \times S^n$. They then proceed to show that the Jacobian of g with respect to (X, y, Z) is nonsingular. Using the implicit function theorem, they show the analyticity of the central path.

Unlike the LP case, central path in SDP does not necessarily converge to the analytical center of the optimal faces (see [64] and [46] for the definition of the analytical center of the optimal faces). Halická, De Klerk, and Roos [46] show that the central path converges to some optimal solution in the limit. However, without the strict complementarity condition, the central path may converge to some point which is not the analytical center of the optimal face. With the strict complementarity condition, Luo, Sturm, and Zhang [64] show that the convergence of the central path to the analytical center of the optimal face.

5.3 Algorithm

The framework of interior point methods for SDP is mostly similar to the LP case (See Algorithm 1 (p11)).

Following the approach for LP, we perturb the optimality conditions by adding a barrier parameter μ:

$$F_\mu(X, y, Z) := \begin{bmatrix} \mathcal{A}^* y + Z - C \\ \mathcal{A}X - b \\ XZ - \mu I \end{bmatrix} = 0. \tag{5.6}$$

Currently, the popular primal-dual interior point path following algorithms use a damped Newton's method to approximately solve this system of equations with $(X, Z) \succ 0$. This is done in conjunction with decreasing μ to 0. The linearization is

$$F'_\mu(X, y, Z) \begin{bmatrix} dX \\ dy \\ dZ \end{bmatrix} = \begin{bmatrix} 0 & \mathcal{A}^* & I \\ \mathcal{A} & 0 & 0 \\ (\cdot)Z & 0 & X(\cdot) \end{bmatrix} \begin{bmatrix} dX \\ dy \\ dZ \end{bmatrix} = -F_\mu(X, y, Z). \tag{5.7}$$

However, since the operator $F'_\mu(X, y, Z)$ maps from $\mathcal{S}^n \times \mathbb{R}^m \times \mathcal{S}^n$ to $\mathcal{S}^n \times \mathbb{R}^m \times \mathcal{M}^n$ and F'_μ is an overdetermined system, where $\mathcal{M}^n$ is the space of $n \times n$ matrices, we can not directly use Newton's method here.

One natural change to the over-determined system is make the third block of (5.6) symmetric by changing the equation $ZX - \mu I$ to

$$(XZ + ZX)/2 - \mu I.$$

The linearization now gives the following equations for the search direction in addition to the linearization of the feasibility equation:

$$(dX\,Z + Z\,dX + X\,dZ + dZ\,X)/2 = \mu I - (XZ + ZX)/2.$$

This direction is the AHO direction in Alizadeh, Haeberly, Overton [5]. However, the AHO direction is not well defined for every pair of primal-dual interior-points. For sufficient conditions that guarantee existence and uniqueness of the AHO direction, see [98] and the references therein.

Another popular search direction is derived by allowing dX to be non-symmetric first and solve (5.7) anyway. Once we have the dX, which might not be symmetric, we then symmetrize it. The un-symmetrized dX satisfies the equation

$$dX\,Z + X\,dZ = \mu I - XZ,$$

or

$$dX + X\,dZ\,Z^{-1} = \mu I Z^{-1} - X.$$

The symmetrization process is thus equivalent to the following system

$$dX + (X\,dZ\,Z^{-1} + Z^{-1}\,dZ\,X)/2 = \mu I Z^{-1} - X.$$

We call this direction the HRVW/KSH/M direction. It was discovered by Helmberg, Rendl, Vanderbei, and Wolkowicz [48], Kojima, Shindoh, and Hara [60], and Monteiro [71].

The third popular search direction is the NT direction named in Nesterov and Todd [75, 76]. The main motivation is to obtain primal-dual symmetry. In another words, we

want to find a linear transformation T such that $X = T^2(Z)$ and $Z^{-1} = T^2(X^{-1})$. Such a linear transformation T^2 is uniquely determined:

$$T^2 = Z^{-1/2}(Z^{1/2}XZ^{1/2})^{1/2}Z^{-1/2}(\cdot)Z^{-1/2}(Z^{1/2}XZ^{1/2})^{1/2}Z^{-1/2}.$$

Thus the last equation $XZ - \mu I$ is changed to

$$(T^{-1}(X)T(Z) + T(Z)T^{-1}(X))/2 - \mu I = 0.$$

The linearization gives

$$dX + T^2(dZ) = \mu Z^{-1} - X.$$

or

$$T^{-2}(dX) + dZ = \mu X^{-1} - S.$$

These two equations are equivalent because $X = T^2(Z)$.

Another more direct approach to tackle the over-determined system (5.6) is using the Gauss-Newton method directly on the system. This was proposed by Kruk et al. [61, 62]. It is shown more accurate solution can be obtained. De Klerk, Peng, Roos and Terlaky [20] showed polynomial convergence for the scaled version of the Gauss-Newton method.

Zhang [121] gave a unified approach to the above three search directions (AHO, HRVW/KSH/M, and NT). Todd [95] studied about twenty search directions and their theoretical and computational proprieties. For other discussions of search directions see e.g. [108].

5.4 Numerical Stability Issue in Semidefinite Programming

SDP algorithms in general obtain lower accuracy in practice than LP algorithms. One issue that SDP has but LP does not have is the cancellation error. For two quantity α and β, if the magnitude of an operation is much smaller than α and β, i.e., $(\alpha +/- \beta)/(|\alpha| + |\beta|) << 1$, a large cancellation error can occur. For example, on a machine with only 4 digits precision, the computation $\mathrm{fl}(\mathrm{fl}(1.2342678) - \mathrm{fl}(1.2331234)) = \mathrm{fl}(1.234 - 1.233) = .001$ only has 1 digit accuracy. Sturm [93] observed that SDP problems have large cancellation errors as X, Z

approach the optimum. This is because the quantity XZ approaches 0 while some elements in X, Z are not small.

Another error comes from the computation of X^{-1} or Z^{-1}. Since X and Z become ill-conditioned as they approach the optimum, the computation of X^{-1} and Z^{-1} becomes less and less reliable.

Based on these observations, Sturm proposed the U-factor approach, which comes from the idea in Higham [49, Lemma 8.6]. Instead of keeping the X and Z variables, the implementation factored the variable X and Z using a product of stable U-factors (a special triangular matrix) and a well conditioned matrix. Over the iterations, the algorithm updated the stable U-factors and the well conditioned matrix. His implementation then achieved relative high accuracy for the NT direction for some of the problems in SDPLIB [11].

Another issue in SDP that can cause numerical instability comes from strict complementarity and Slater's condition. In the following Chapter, we present results on this issues.

Chapter 6

Hard Instances in Semidefinite Programming

6.1 Introduction

In this chapter we present an algorithm for generating *hard instances* of SDP, i.e. by hard we mean problems where strict complementarity fails. We use this set of hard problems to study the correlation between the loss of strict complementarity and the number of iterations needed to obtain optimality to a desired accuracy by interior point algorithms. We compare and contrast our results to recent work by Freund, Ordóñez, and Toh [31], who found that the number of iterations needed by practical interior point methods correlated well with the their aggregated geometrical measure as well as with Renegar's condition number.

We consider the SDP in the form of (5.1) (p99). The set of optimal primal (resp. dual) solutions is denoted $\mathcal{P}^*$ (resp. $\mathcal{D}^*$).

The SDP model has important applications, elegant theory, and efficient solution techniques, see [108]. Moderate sized problems can be solved to near optimality using primal-dual interior point (p-d i-p) methods. These methods are based on Newton's method with path following, i.e. the (Newton) search direction is found using a linearization of the (perturbed, symmetrized) optimality conditions. The iterates follow the central path, i.e. primal-dual feasible solutions with $ZX - \mu I = 0,\ \mu > 0$. On the central path, X and Z are mutually orthogonally diagonalizable, $X = QD_XQ^T, Z = QD_ZQ^T$; and their corresponding vectors of

eigenvalues, $\lambda_X = \operatorname{diag}(D_X), \lambda_Z = \operatorname{diag}(D_Z)$, satisfy

$$\lambda_X \circ \lambda_Z = \mu e, \tag{6.1}$$

where $\circ$ denotes the *Hadamard* or elementwise product of the vectors, and $\operatorname{diag}(W)$ is the vector formed from the diagonal of W. The optimum dual pair of SDP is attained in the limit as $\mu \downarrow 0$; strict complementarity is indicated at $\mu = 0$ if $X + Z \succ 0$, i.e. strict positive definiteness. Therefore, as in linear programming, either $(\lambda_X)_i \downarrow 0, (\lambda_Z)_i \to \mathrm{O}(1)$ holds, or $(\lambda_Z)_i \downarrow 0, (\lambda_X)_i \to \mathrm{O}(1)$ holds. However, examples exist where the optimal X, Z have a nontrivial nullspace vector in common, i.e. strict complementarity fails. From (6.1), this means there exists i with both $(\lambda_Z)_i \downarrow 0, (\lambda_X)_i \downarrow 0$ but $(\lambda_Z)_i(\lambda_X)_i \cong \mu$, i.e. the value of each eigenvalue is order $\sqrt{\mu}$. For example, if the p-d i-p algorithm stops with a near optimal solution with duality gap $\mu = \operatorname{trace} ZX/n = \mathrm{O}(10^{-12})$, then we can expect the value of both eigenvalues to be as large as $\sqrt{\mu} = 10^{-6}$. In addition, the Jacobian of the optimality conditions at an optimum is singular, raising the question of slowed convergence. (See Remark 6.7.) These problems result in *hard instances* of SDP. P-d i-p methods typically run into difficulties such as slow (linear rate) convergence and low accuracy of the optimum.

6.1.1 Outlines

In this chapter we outline a procedure for generating hard instances of SDP. We then introduce two *measures of hardness*. We empirically show that: (i) these measures can be evaluated accurately; (ii) the size of the strict complementarity gaps correlate well with the number of iteration for the SDPT3 [99] solver, as well as with the local asymptotic convergence rate; and (iii) larger strict complementarity gaps coupled with the failure of Slater's condition correlate with loss of accuracy in the solutions. In addition, the numerical tests show that there is *no* correlation between the strict complementarity gaps and the geometrical measure used in [31], or with Renegar's condition number.

We include tests on the SDPLIB problem set. Here we only found weak correlations due to lack of accuracy in the optimal solutions.

The procedure for generating hard problems has been submitted to the Decision Tree for Optimization Software, URL: plato.la.asu.edu/guide.html. See also SDPLIB e.g. [11],

URL: www.nmt.edu/~sdplib/. The MATLAB programs are available with URL: orion.math.uwaterloo.ca:80/~hwolkowi/henry/software/readme.html

6.2 Generating Hard SDP Instances

In this section we show how to generate the *hard* SDP instances; i.e. the problems where strict complementarity fails.

Definition 6.1 *A primal-dual pair of optimal solutions* $(\bar{X}, \bar{Z}) \in \mathcal{P}^* \times \mathcal{D}^*$ *is called a* maximal complementary solution pair *to the problems (PSDP) and (DSDP), if the pair maximizes the sum* $\operatorname{rank}(X) + \operatorname{rank}(Z)$ *over all primal-dual optimal solution pairs* (X, Z).

A primal-dual pair of optimal solutions $(\bar{X}, \bar{S})$ is maximal complementary if and only if

$$\mathcal{R}(\hat{X}) \subseteq \mathcal{R}(\bar{X}),\ \forall \hat{X} \in \mathcal{P}^*, \qquad \mathcal{R}(\hat{S}) \subseteq (\bar{S}),\ \forall \hat{S} \in \mathcal{D}^*, \tag{6.2}$$

where $\mathcal{R}$ denotes range space. This follows from the fact that

$$\hat{X}\bar{S} = \bar{X}\hat{S} = \hat{X}\hat{S} = 0,\ \forall \hat{X} \in \mathcal{P}^*, \forall \hat{S} \in \mathcal{D}^*,$$

i.e. all optimal solution pairs are mutually orthogonally diagonalizable.

Definition 6.2 *The* strict complementarity gap *is defined as* $g = n - \operatorname{rank}(\bar{X}) - \operatorname{rank}(\bar{Z})$, *where* $(\bar{X}, \bar{Z})$ *is a maximal complementary solution pair.*

Note that g is equal to the minimum of the number of zero eigenvalues of $X + Z$, where the minimum is taken over all optimal solution pairs (X, Z).

For more details and proofs of these characterizations see [21], [39] and the references therein.

We assume the linear operator $\mathcal{A} : \mathcal{S}^n \mapsto \mathbb{R}^m$ in our SDP problem (PSDP) in matrix form is

$$\mathcal{A}(X) := [\operatorname{trace}(A_1X),\ \operatorname{trace}(A_2X), \ldots,\ \operatorname{trace}(A_mX)]^T,$$

where $A_1, A_2, \ldots, A_m \in \mathcal{S}^n$ are linearly independent. The adjoint of $\mathcal{A}$ is $\mathcal{A}^* : \mathbb{R}^m \mapsto \mathcal{S}^n$.

$$\mathcal{A}^*(y) := \sum_{i=1}^{m} A_i y_i.$$

Algorithm 6.3 *Constructing* Hard SDP *Instances with gap g*

1. Given: positive integers $r > 0$ and $m > 1$ are the rank of an optimum X and the number of constraints, respectively.

2. Let $Q = [Q_P|Q_N|Q_D]$ be an orthogonal matrix, where the dimensions of Q_P, Q_N, Q_D are $n \times r$, $n \times g$, $n \times (n-r-g)$, respectively, and $r > 0$. Construct positive semidefinite matrices X and Z as follows:

$$X := Q_P D_X Q_P^T, \qquad Z := Q_D D_Z Q_D^T,$$

where D_X and D_Z are diagonal positive definite.

3. Define

$$A_1 = [Q_P|Q_N|Q_D] \begin{bmatrix} 0 & 0 & Y_2^T \\ 0 & Y_1 & Y_3^T \\ Y_2 & Y_3 & Y_4 \end{bmatrix} [Q_P|Q_N|Q_D]^T, \tag{6.3}$$

where Y_1, Y_2, Y_3, and Y_4 are block matrices of appropriate dimensions, $Y_1 \succ 0$, and $Q_D Y_2 \neq 0$.

4. Choose $A_i \in \mathcal{S}^n, i = 2, \ldots, m$, such that $\{A_1 Q_P, A_2 Q_P, \ldots, A_m Q_P\}$ is a linearly independent set. (Note that $A_1 Q_P = Q_D Y_2 \neq 0$.)

5. Set

$$b := \mathcal{A}(X), \qquad C := \mathcal{A}^*(y) + Z, \quad \text{with} \quad y \in \mathbb{R}^m \text{ randomly generated.}$$

Theorem 6.4 *The data* $(\mathcal{A}, b, C)$ *constructed in Algorithm 6.3 gives a* hard SDP *instance with a strict complementarity gap* g.

Proof. Suppose that X, y, Z are constructed by the algorithm. Step 2 guarantees that X, Z are positive semidefinite and $ZX = 0$ (complementary slackness holds). Step 5 guarantees that X, y, Z are primal-dual feasible. Therefore, our construction implies that $X, (y, Z)$ are a primal-dual optimal pair.

Choose any $\bar{X}, \bar{Z} \in \mathcal{P}^* \times \mathcal{D}^*$ with $\mathcal{R}(X) \subseteq \mathcal{R}(\bar{X})$ and $\mathcal{R}(Z) \subseteq \mathcal{R}(\bar{Z})$. We now show that $\mathcal{R}(X) = \mathcal{R}(\bar{X})$ and $\mathcal{R}(Z) = \mathcal{R}(\bar{Z})$, i.e. by (6.2) X, Z are a maximal complementary pair.

Since $\bar{X}$ and Z must also be an optimal pair, i.e $\bar{X}Z = 0$, we get that $\mathcal{R}(\bar{X}) \subseteq \mathcal{R}(Z)^{\perp} = \mathcal{R}([Q_P|Q_N])$. So, we can write

$$\bar{X} = [Q_P|Q_N] \begin{bmatrix} D_{P,\bar{X}} & W_{\bar{X}}^T \\ W_{\bar{X}} & D_{N,\bar{X}} \end{bmatrix} [Q_P|Q_N]^T,$$

where, in particular, $D_{N,\bar{X}} \succeq 0$. Let

$$\Delta X = \bar{X} - X = [Q_P|Q_N] \begin{bmatrix} D_{P,\bar{X}} - D_X & W_{\bar{X}}^T \\ W_{\bar{X}} & D_{N,\bar{X}} \end{bmatrix} [Q_P|Q_N]^T.$$

Since

$$\operatorname{trace}(A_1 \Delta X) = \operatorname{trace}(A_1 \bar{X} - A_1 X) = 0,$$

We have

$$\operatorname{trace}\left(\begin{bmatrix} Q_P|Q_N \end{bmatrix}^T A_1 \begin{bmatrix} Q_P|Q_N \end{bmatrix} \begin{bmatrix} D_{P,\bar{X}} - D_X & W_{\bar{X}}^T \\ W_{\bar{X}} & D_{N,\bar{X}} \end{bmatrix} \right) = 0$$

From the structure of A_1, we see that

$$[Q_P|Q_N]^T A_1 [Q_P|Q_N] = \begin{bmatrix} 0 & 0 \\ 0 & Y_1 \end{bmatrix}.$$

So

$$0 = \operatorname{trace}(A_1 \Delta X) = \operatorname{trace}\left(\begin{bmatrix} 0 & 0 \\ 0 & Y_1 \end{bmatrix} \begin{bmatrix} D_{P,\bar{X}} - D_X & W_{\bar{X}}^T \\ W_{\bar{X}} & D_{N,\bar{X}} \end{bmatrix} \right) = \operatorname{trace}\left(Y_1 D_{N,\bar{X}} \right).$$

By $Y_1 \succ 0$ and $D_{N,\bar{X}} \succeq 0$, we have that $D_{N,\bar{X}} = 0$. Since $\bar{X}$ is positive semidefinite, we have $W_{\bar{X}} = 0$ and $\mathcal{R}(\bar{X}) = \mathcal{R}(Q_P) = \mathcal{R}(X)$.

Similarly, we see that $\mathcal{R}(\bar{Z}) \subseteq \mathcal{R}(Q_N, Q_D)$ from $X\bar{Z} = 0$. Let $\Delta Z = \bar{Z} - Z$ and $\Delta y = \bar{y} - y$, where $\mathcal{A}^*(\bar{y}) + \bar{Z} = C$. Then we have $\mathcal{A}^*(\Delta y) = -\Delta Z$. Since Q_P is a subspace of the the null space of both $\bar{Z}$ and Z, we have $-\Delta Z Q_P = 0$, i.e. $\mathcal{A}^*(\Delta y) Q_P = 0$. We write it in matrix form,

$$\sum_{i=1}^{m} A_i Q_P \Delta y_i = 0.$$

Since $\{A_iQ_P\}$ are nonzero and linearly independent, we see that $\Delta y_i = 0$ for all i. Thus, $\bar{Z} = Z$.

Therefore X, Z is a maximal complementary pair. Since, by construction, $\operatorname{rank}(X) + \operatorname{rank}(Z) = n - g$, we have shown that the SDP is a hard instance with strict complementarity gap g. ■

To avoid conflicts between the loss of strict complementarity and the loss of strict feasibility, we can use the following additional condition.

Corollary 6.5 *Suppose that the data $(\mathcal{A}, b, C)$ is constructed using Algorithm 6.3 with the additional condition that A_2 satisfies*

$$[Q_P|Q_N]^T A_2 [Q_P|Q_N] \succ 0. \tag{6.4}$$

Then Slater's condition holds for the dual program (DSDP).

Proof. Suppose that X, y, Z are as constructed by the algorithm. Then $Z = C - \mathcal{A}^*(y) = Q_D D_Z Q_D^T \succeq 0$. From [13, Theorem 7.1], we get that Slater's condition fails for (DSDP) if and only if

$$\exists R \succeq 0 \text{ with } R \neq 0, RZ = 0, \nabla_y \operatorname{trace} R(\mathcal{A}^* y - C) = \mathcal{A}(R) = 0.$$

Now $RZ = 0$ implies that $R = [Q_P|Q_N] D_R [Q_P|Q_N]^T$, for some symmetric D_R of appropriate size. Therefore, $\mathcal{A}(R) = 0$ implies that

$$0 = \operatorname{trace} A_2 R = \operatorname{trace} A_2 [Q_P|Q_N] D_R [Q_P|Q_N]^T = \operatorname{trace} ([Q_P|Q_N]^T A_2 [Q_P|Q_N]) D_R.$$

The assumption (6.4) now implies that $D_R = 0$ and so also $R = 0$. Therefore, Slater's condition holds. ■

6.3 Measures for Strict Complementarity Gaps

In [31], the authors indicate the following difficulties in measuring the existence and size of the strict complementarity gap.

> "Furthermore, in interior point methods for either linear or semidefinite programming, we terminate the algorithm with a primal-dual solution that is almost optimal but not actually optimal. Hence there are genuine conceptual difficulties in trying to quantify and compute the extent of near-non-strict-complementarity for an SDP instance."

The measure, κ in (6.10), is proposed in [31]. However, this measure does not distinguish between a small or large strict complementarity gap g. But, as our numerical tests in Section 6.4 indicate, large values of g are well correlated with large iterations numbers. This motivates the introduction of our following two new measures.

6.3.1 Strict Complementarity Gap Measures g_t and g_s

Measure g_t

For barrier parameter $\mu > 0$, $\mu \downarrow 0$, and corresponding feasible pairs $X = X_\mu, Z = Z_\mu$ on the central path, let the orthogonal eigenvalue decomposition be $X = Q\Lambda_X Q^T$ and $Z = Q\Lambda_Z Q^T$. Consider the eigenvalue ratios $w_i^d := \Lambda_{Zi}/\Lambda_{Xi}$. Then

$$XZ = Q\Lambda_X Q^T Q\Lambda_Z Q^T = \Lambda_X\Lambda_Z = \mu I, \qquad w_i^d = \frac{\mu}{(\Lambda_X)_i^2}.$$

Suppose that $X \to \bar{X}$, $Z \to \bar{Z}$. We then expect the following behaviour.

$$w_i^d \to \begin{cases} \infty & \text{if } \Lambda_{\bar{X}i} + \Lambda_{\bar{Z}i} > 0 \text{ (no gap) and } \Lambda_{Xi} \to 0 \\ 0 & \text{if } \Lambda_{\bar{X}i} + \Lambda_{\bar{Z}i} > 0 \text{ (no gap) and } \Lambda_{Zi} \to 0 \\ O(1) & \text{if } \Lambda_{\bar{X}i} + \Lambda_{\bar{Z}i} = 0 \text{ (a gap).} \end{cases}$$

Empirical evidence suggests that the sequence $\left\{w_i^d\right\}$ converges when there is a strict complementarity gap. The measure we define exploits this behaviour. In practice, we use the vector of eigenvalues

$$w^d = \frac{1}{2}\lambda\left(X^{-1}Z + ZX^{-1}\right). \tag{6.5}$$

(Note that the eigenvalues of $X^{-1}Z$ interlace the eigenvalues of $\frac{1}{2}\left(X^{-1}Z + ZX^{-1}\right)$, e.g. [66].) For given tolerances T_u and T_l, we estimate the strict complementarity gap using the cardinality

$$g_t := |\{w_i^d : T_l < w_i^d < T_u\}|. \tag{6.6}$$

Measure g_s

The second measure exploits the idea from [31]. We let X and Z to be a solution pair on the central path corresponding to $\mu > 0$. The eigenvalue decompositions of X and Z are $X = Q\Lambda_X Q^T$ and $Z = Q\Lambda_Z Q^T$. The measure uses the numerical rank (e.g. [92, 47]) of $X + Z$. We compute

$$w^s := \frac{1}{2\sqrt{\mu}}\lambda\left(X + Z\right), \tag{6.7}$$

where $\mu = \operatorname{trace} ZX/n$. Given a tolerance $T > 0$, we estimate the strict complementarity gap g using the cardinality

$$g_s := |\{w_j^s : w_j^s \leq T\}|. \tag{6.8}$$

Remark 6.6 *Note that on the central path, X, Z are mutually diagonalizable. Therefore, the eigenvalues of the sum $X + Z$ is the same as the sum of the eigenvalues. However, this is not necessarily true off the central path, see the recent paper [57]. In this remarkable paper the author solves a classical problem about the eigenvalues of sums of Hermitian operators, connecting it to the Schubert calculus for the homology of Grassmannians and the moduli of vector bundles.*

However, we should point out that this measure g_s may incorrectly include some indices which do not belong to the strict complementarity gap when the solution estimates X, Z are not accurate enough. Consider the following results from a randomly generated problem instance with strict complementarity gap 1. The first 7 eigenvalues from the solution estimates X and Z (obtained using SDPT3) are

$$\lambda_X = \begin{bmatrix} 7.3 \times 10^{-7} \\ 1.8 \times 10^{-6} \\ 2.0 \times 10^{-6} \\ 1.4 \times 10^{-5} \\ 6.2 \\ 8.6 \times 10^{3} \\ 9.2 \times 10^{3} \end{bmatrix}, \quad \lambda_Z = \begin{bmatrix} 65 \\ 59 \\ 54 \\ 2.4 \\ 2.1 \times 10^{-5} \\ 4.7 \times 10^{-9} \\ 4.5 \times 10^{-9} \end{bmatrix}. \tag{6.9}$$

Note that

$$(\lambda_X)_4 << (\lambda_X)_5 << (\lambda_X)_6, \qquad (\lambda_Z)_6 << (\lambda_Z)_5 << (\lambda_Z)_4,$$

i.e. the fifth elements are relatively small/large compared to the next/previous larger/smaller elements. This indicates that there is a strict complementarity gap $g = 1$. However, the sum of these two eigenvalues fails to correctly estimate the size of the gap,

$$\lambda_X + \lambda_Z = \begin{bmatrix} 65 \\ 59 \\ 54 \\ 2.4 \\ 6.2 \\ 8.6 \times 10^3 \\ 9.2 \times 10^3 \end{bmatrix}.$$

Higher accuracy in the approximate optimal solutions X, Z often corrects this issue, see the numerics in Section 6.4.

6.3.2 Measure κ

The last measure we introduce is κ used in [31]. For a given tolerance T, define the following index set $T^s := \{j : w_j^s \leq T\}$, where w^s is defined in (6.7). Then

$$\kappa := -\sum_{j \in T^s} \ln(w_j^s)/|T^s|. \tag{6.10}$$

When strict complementarity holds (resp. fails), we expect to see a relatively large (resp. small) κ.

6.4 Numerics

We now compare the various measures on randomly generated instances with guaranteed strict complementarity gaps as well as on problems from the SDPLIB test set, [11].

6.4.1 Randomly Generated Instances

We use Algorithm 6.3 to generate the hard instances. To implement the algorithm, we generate a random orthogonal matrix Q and random diagonal D_X and D_Z. The elements of D_X and D_Z are uniformly distributed in the range of [0.1, 100.1] to ensure positivity of D_X and D_Z. The optimal solution of this hard SDP instance is then determined from step 2 in Algorithm 6.3. For the special matrix A_1, we construct Y_1 according to:

1. generate the random symmetric matrix Y_1 with uniformly distributed elements in $[-10000, 10000]$;
2. add rI to the above matrix Y_1, where r is a random number in $[0, 20000]$;
3. if Y_1 is *not* sufficiently positive definite, repeat the process from step 1.

All the elements of the random matrices Y_2, Y_3, and Y_4, are uniformly distributed in the interval $[-10000, 10000]$. If necessary, we symmetrize the matrices. If $Q_D Y_2$ is close to a zero matrix, we repeat the process for Y_2. Our special matrix A_1 is then constructed from Step 3 in Algorithm 6.3. Once we have such a special matrix, we generate random symmetric uniformly distributed matrices A_j. If one of the $A_j Q_p$ is not properly linearly independent, then we add a new A_j to the list. To guarantee that Slater's condition holds, we apply the condition in Corollary 6.5.

We present the average of results from 100 groups of tests. Each group consists of SDP instances with 26 different gap values. We set the following parameters: $m = 10, n = 30, \text{gap} = 0, \ldots, 25$. The rank for the dual optimal solution is fixed at 4. The name of the instance shows how large the gap is, e.g. gap5. The accuracy of solutions is given by the err term:

$$\text{err} := \max\left\{ \frac{\|\mathcal{A}(X) - b\| + |\min(\text{eig}(X), 0)|}{1 + \|b\|_\infty}, \frac{\|\mathcal{A}^*(y) + Z - C\| + |\min(\text{eig}(Z), 0)|}{1 + \|C\|_\infty}, \frac{|C \cdot X - b^T y|}{1 + |b^T y|} \right\} \quad (6.11)$$

When computing g_t (6.6), we set the tolerances T_u, T_l dynamically. More precisely, we sort the w_i^d (6.5) in ascending order. We then use the ratios $\bar{w}_i^d := w_i^d / w_{i+1}^d$, to measure how

fast the w_i^d are changing. If there is only one small (< 0.02) $\bar{w}_i^d$, we assume that there is no gap. Otherwise, we find the two smallest valued $\bar{w}_i^d$ and set the two indices to be j and k ($j < k$). Then $T_l := w_j + \epsilon$ and $T_u := w_k + \epsilon$. In practice, once we have found the indices j and k, the estimated gap g_t is returned by using the value of $k - j$.

When computing g_s in (6.8), we set the tolerance $T = \max\{100, \min_i(w_i^s)\}$, where w^s is defined in (6.7). The tolerance T for the measure κ is the same as the one used for g_s. This is the same tolerance as that used in [31].

6.4.2 Plots for Randomly Generated Instances

To illustrate the relationships among the various measures we consider three groups of figures. To illustrate the influence of accuracy in the solutions, each group consists of three figures with decreasing stop tolerances 10^{-8}, 10^{-10}, and 10^{-12}, respectively.

The x-axis of each figure represents the strict complementarity gap ranging from 0 to 24. The y-axes, from left to right, represent, respectively:

iteration numbers,
negative log (base 10) of errors (6.11),
measure g_t,
measure g_s,
measure κ,
local convergence rate (discussed in Item 5 (p119)).

- The first three Figures 6.1, 6.2, 6.3, are average results from 100 instances. We apply Corollary 6.5 to guarantee that Slater's condition holds for the dual.
- The next three figures 6.4, 6.5, 6.6, show the behaviour of a typical single instance without applying Corollary 6.5. We see in Table 6.1 that Slater's condition generally holds for all the primal but generally fails for the dual of problems gap0–gap21 as the quantity D_p is very large. (See also the discussion in Section 6.4.3 (p123) on the computation and meaning of the quantities D_p.)
- The last three Figures 6.7, 6.8, 6.9 consider the average behaviour on 100 instances. Again, we do not apply Corollary 6.5.

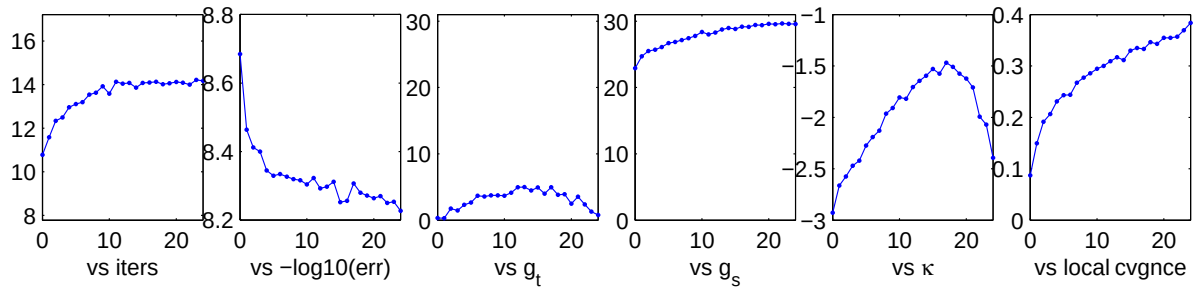

Figure 6.1: Slater's holds; stop tolerance 10^{-8}; strict complementarity gaps from 0 to 24 **versus average of**: iterations, $-\log_{10}\text{err}$, g_t, g_s, κ, local convergence; 100 instances.

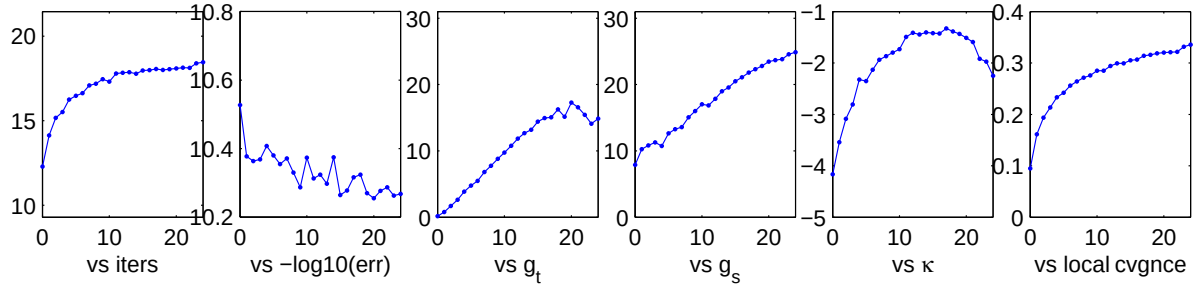

Figure 6.2: Slater's holds; stop tolerance 10^{-10}; strict complementarity gaps from 0 to 24 **versus average of**: iterations, $-\log_{10}\text{err}$, g_t, g_s, κ, local convergence; 100 instances.

Observations from the nine figures 6.1 to 6.9:

1. There is a *strong correlation* between the iteration number to achieve the desired stopping tolerance and the size of the strict complementarity gap. In the case that the Slater's condition holds (Figure 6.1–6.3), we see the larger the strict complementarity gap, the more steps SDPT3 needs to get higher accuracy. In the case that the Slater's condition generally fails (Figure 6.7–6.9), we observe that such a correlation is even stronger, i.e. we almost have a straight line between iterations numbers and gaps when the gap is less than 21. Although it is possible that such a correlation may due to other unknown facts, we believe such controlled environment to generating random problems on different strict complementarity gaps is the best we can do empirically so far.

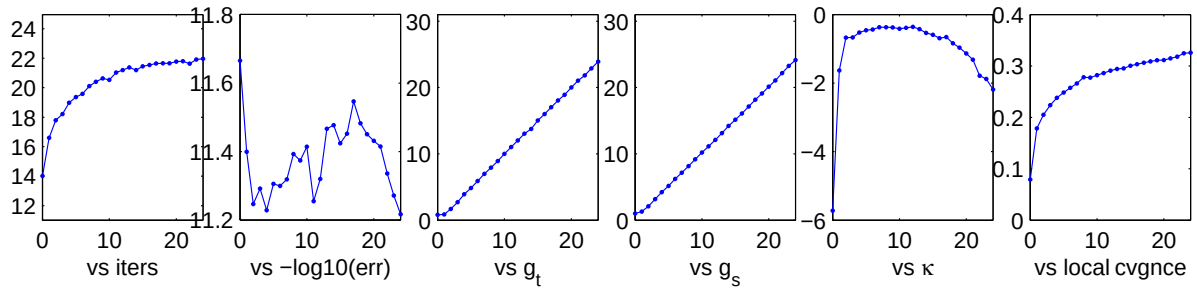

Figure 6.3: Slater's holds; stop tolerance 10^{-12}; strict complementarity gaps from 0 to 24 **versus average of**: iterations, $-\log_{10}\text{err}$, g_t, g_s, κ, local convergence; 100 instances.

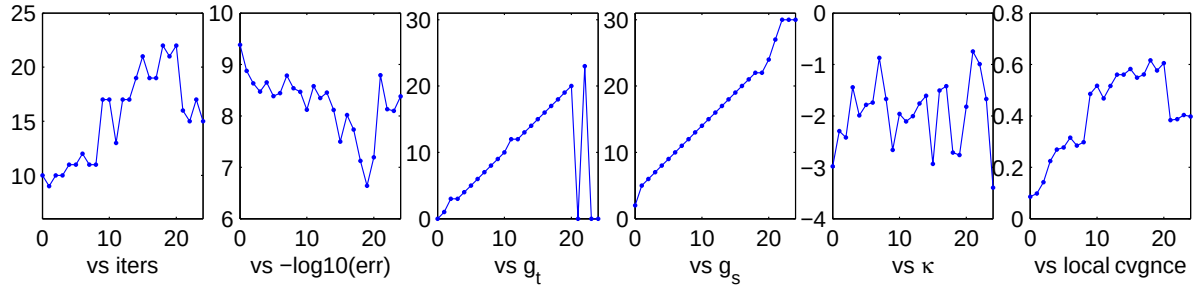

Figure 6.4: Slater's fails for gap0–gap21; stop tolerance 10^{-8}; strict complementarity gaps from 0 to 24 **versus**: iterations, $-\log_{10}\text{err}$, g_t, g_s, κ, local convergence; single instance.

2. It can be seen from Figure 6.1–6.3 that the accuracy is not a problem for strict complementarity gaps in general. They almost all achieved desired accuracy except that in Figure 6.3 we have slightly larger than desired error. However, the failure of Slater's condition coupled with the large strict complementarity gaps cause significant error for SDPT3 as shown in Figure 6.4–6.6 and Figure 6.7–6.9.

3. The measures g_t, g_s both improve dramatically as the accuracy increases in Figures 6.1, 6.2, 6.3. We see this same phenomenon in the other two groups of figures.

4. The measure κ also improves with smaller stopping tolerances.

5. Local Asymptotic Convergence Rate vs Strict Complementarity Gap: in the literature, e.g. [84] [50] [4] [64] [59], local superlinear or quadratic convergence results depend on

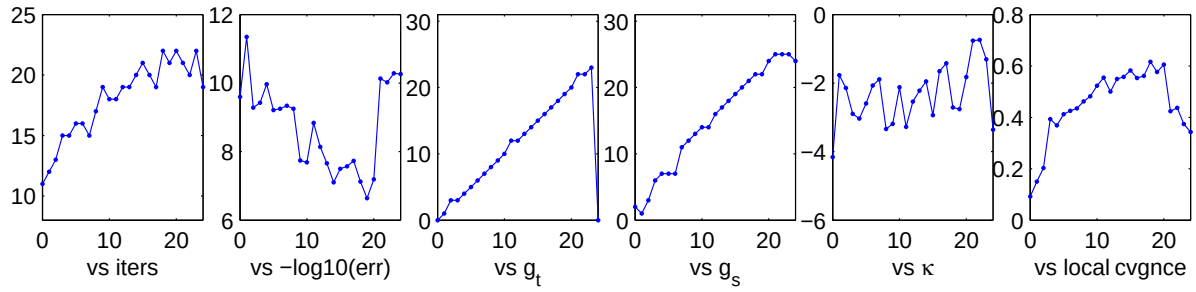

Figure 6.5: Slater's fails for gap0–gap21; stop tolerance 10^{-10}; strict complementarity gaps from 0 to 24 **versus**: iterations, $-\log_{10}\text{err}$, g_t, g_s, κ, local convergence; single instance.

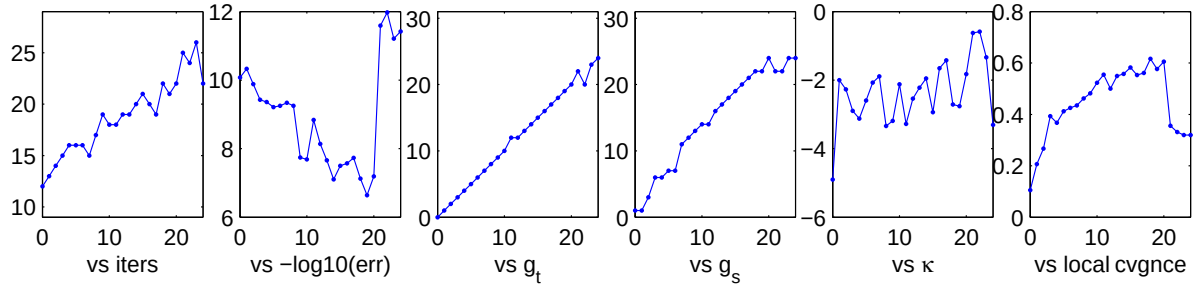

Figure 6.6: Slater's fails for gap0–gap21; stop tolerance 10^{-12}; strict complementarity gaps from 0 to 24 **versus**: iterations, $-\log_{10}\text{err}$, g_t, g_s, κ, local convergence; single instance.

the assumption of strict complementarity. Thus it is intuitive to expect this in practice as well. Our numerical results confirm this conjecture. The convergence rate is defined by the ratio of the relative duality gap at successive iterations. We list the geometrical mean of the convergence rate for the last five iterations. This is illustrated in the rightmost picture in the figures. It is evident from Figure 6.1–6.3 that the larger the strict complementarity gaps, the slower the convergence rate is.

6. Slater's condition's effect: by a comparison between Figure 6.1–6.3 and Figure 6.7–6.9, we can see that the failure of Slater's condition can

   (a) strengthen the correlation between the iteration numbers and strict complementarity gaps;

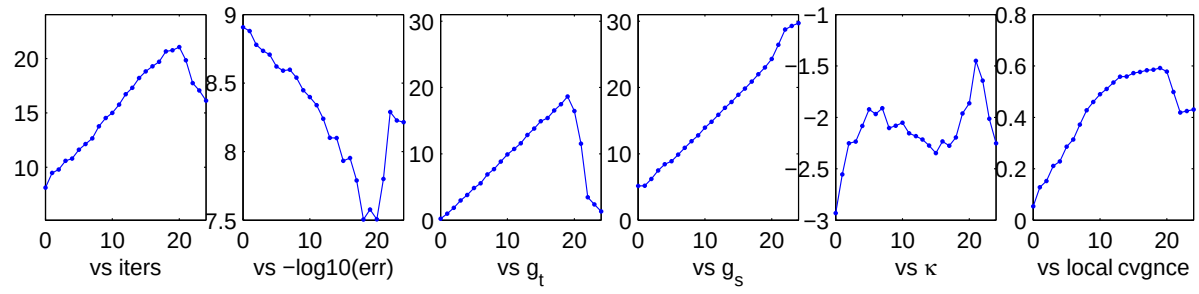

Figure 6.7: Slater's generally fails; stop tolerance 10^{-8}; strict complementarity gaps from 0 to 24 **versus average of**: iterations, error, g_t, g_s, κ, local convergence; 100 instances.

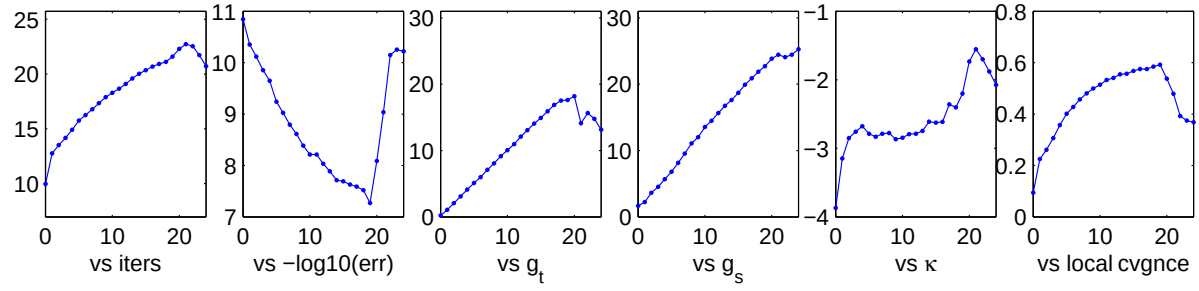

Figure 6.8: Slater's generally fails; stop tolerance 10^{-10}; strict complementarity gaps from 0 to 24 **versus average of**: iterations, error, g_t, g_s, κ, local convergence; 100 instances.

(b) increase the errors when strict complementarity gaps increase;

(c) make the computation of strict complementarity gaps measures (g_t and g_s) more accurate;

(d) slow the local convergence rate when strict complementarity gaps increase.

Remark 6.7 *The slow convergence rates can be partially explained by the singularity of the Jacobian, which occurs in the presence of a strict complementarity gap.*

Suppose that strict complementarity fails for the optimum pair estimate X, Z. Then we

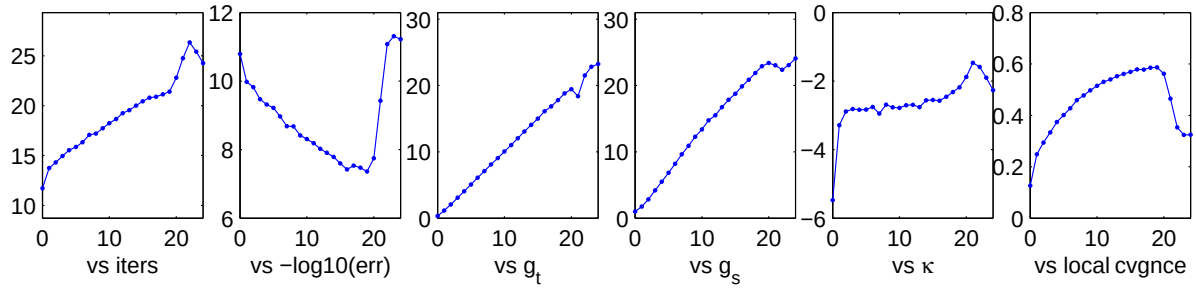

Figure 6.9: Slater's generally fails; stop tolerance 10^{-12}; strict complementarity gaps from 0 to 24 **versus average of**: iterations, error, g_t, g_s, κ, local convergence; 100 instances.

can assume the joint diagonalization structure

$$X = Q \begin{bmatrix} D_X & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} Q^T, \qquad Z = Q \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & D_Z \end{bmatrix} Q^T$$

for some orthogonal matrix Q and positive definite diagonal matrices D_X, D_Z. Then we can rewrite the Jacobian of the SDP *optimality conditions as*

$$\begin{bmatrix} 0 & \bar{\mathcal{A}}^* & I \\ \bar{\mathcal{A}} & 0 & 0 \\ \begin{bmatrix} D_X & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & 0 & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & D_Z \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta\bar{X} \\ \Delta\bar{y} \\ \Delta\bar{Z} \end{bmatrix} = 0,$$

where $\Delta\bar{X} = Q^T \Delta X Q$,$\Delta\bar{Z} = Q^T \Delta Z Q$ and the symmetric matrices A_i defining the linear transformation $\mathcal{A}$ are changed to $Q^T A_i Q$ for $\bar{\mathcal{A}}$. If we assume that both $\Delta\bar{X}, \Delta\bar{Z}$ are diagonal, then this reduces the problem to an ordinary square system and the resulting Jacobian is singular due to the zero row. The diagonal assumption does not change the feasibility of the first and third blocks of equations. We can then modify the off-diagonal part of $\Delta\bar{Z}$ to guarantee the feasibility of the second block of equations.

The singularity of the Jacobian means that we should expect loss of both quadratic and superlinear convergence for Newton type methods.

6.4.3 Geometrical Measure vs Large Strict Complementarity Gaps

In [31], the authors use SDPT3 and the SDPLIB test set. They show that the aggregate geometrical measure g^m, i.e. the geometric mean of the four geometric measures D_p, g_p, D_d, g_d, in Table 6.1, is (generally) well correlated with the iteration number. They also show that the correlation holds for Renegar's condition number, see also Table 6.2. Briefly, the geometrical measure D_p is the maximum norm over all ϵ-optimal solutions; the measure g_p will be smaller if the feasible region of the primal SDP contains a point X whose norm is not too large and whose distance from the boundary of the Semidefinite cone is not too small. The meanings for measures D_d and g_d are almost identical except that they are applied to the dual SDP. In [31], the authors state that

> For primal and dual feasible conic problem, the objective function level sets of the primal problem are unbounded ($D_p = \infty$) if and only if the dual problem contains no slack vector in the interior of the dual cone ($g_d = \infty$).

For more details on the geometrical measure and Renegar's condition number and their computation, please see [31] and the references therein.

The values for these measures for the SDP instance in Figures 6.7, 6.8, 6.9 are given in Tables 6.1, 6.2. We use the same code used in [31] to compute the geometrical measure g^m and Renegar's condition number. [1]

As pointed out in [31], the strict complementarity gap might not be theoretically related to the geometrical measures or Renegar's condition number. In fact, our numerical computations on our generated instances confirm this, see Tables 6.1, 6.2. The geometric measures and Renegar's condition measure show no correlation with the size of the strict complementarity gap. Our numerics suggest that there is no strictly feasible point for most of the duals (when gap ≤ 21) for the generated hard instances, since the g_d, and D_P measures are large or infinity in Table 6.1. Since the distances to dual infeasibility are small, Renegar's condition numbers in Table 6.2 are also large, regardless of the change in the strict complementarity gaps.

[1] **Acknowledgment:** The authors thank Professor Ordóñez, University of Southern California, for providing the software for the measure evaluations.

[2] In [31] it is shown that $g_d = \infty \iff \rho_D(d) = 0$. However, due to inaccuracy from SDPT3, we get inconsistencies here.

Problem	D_p	g_p	D_d	g_d	g^m
gap0	Inf	1.9e+02	1.5e+02	Inf	Inf
gap1	1.2e+12	1.5e+02	2.4e+02	Inf	Inf
gap2	2.9e+08	1.6e+02	2.3e+02	Inf	Inf
gap3	3.1e+08	1.4e+02	1.7e+02	Inf	Inf
gap4	1.1e+11	1.8e+02	1.9e+02	MAXIT	N/A
gap5	1.6e+08	1.1e+02	2.4e+02	Inf	Inf
gap6	4.5e+08	9.9e+01	2.8e+02	Inf	Inf
gap7	1.9e+08	1.6e+02	1.1e+02	Inf	Inf
gap8	1.4e+09	2.1e+02	1.3e+02	Inf	Inf
gap9	1.4e+09	1.8e+02	1.8e+02	Inf	Inf
gap10	2.1e+09	1.3e+02	4.0e+02	6.2e+04	4.7e+00
gap11	1.0e+09	1.7e+02	1.4e+02	Inf	Inf
gap12	Inf	1.3e+02	3.2e+02	Inf	Inf
gap13	Inf	1.6e+02	2.6e+01	Inf	Inf
gap14	Inf	1.6e+02	Nacc	Inf	N/A
gap15	Inf	1.7e+02	6.9e+01	Inf	Inf
gap16	3.0e+10	2.5e+02	2.2e+02	Inf	Inf
gap17	Inf	2.6e+02	2.1e+02	Inf	Inf
gap18	Inf	1.5e+02	2.6e+02	Inf	Inf
gap19	1.2e+10	1.1e+02	2.6e+02	Inf	Inf
gap20	6.3e+10	1.8e+02	2.3e+02	Inf	Inf
gap21	1.2e+10	2.2e+02	1.4e+02	Inf	Inf
gap22	2.7e+02	1.2e+02	2.3e+02	MAXIT	N/A
gap23	1.8e+02	2.5e+02	1.4e+02	Nacc	N/A
gap24	3.6e+01	2.5e+02	1.1e+02	MAXIT	N/A

Table 6.1: Notation from [31]: (D_p, g_p) - primal geometrical measure; (D_d, g_d) - dual geometrical measure; (g^m) - aggregate geometrical measure, i.e. geometrical mean of D_p, g_p, D_d, and g_d. MAXIT - max iteration limit reached; Nacc - no accurate/meaningful solution.

Problem	$\rho_P(d)$	$\rho_D(d)$	$\|d\|_l$	$\|d\|_u$	$C(d)_l$	$C(d)_u$
gap0	2.8e+04	7.4e-04	1.1e+09	1.1e+09	1.5e+12	1.5e+12
gap1	3.1e+04	9.9e-04	2.2e+09	2.2e+09	2.2e+12	2.2e+12
gap2	2.9e+04	1.3e-03	2.5e+09	2.5e+09	2.0e+12	2.0e+12
gap3	3.2e+04	2.5e-04	7.3e+08	7.3e+08	2.9e+12	2.9e+12
gap4	3.4e+04	1.1e-03	8.0e+08	8.0e+08	7.3e+11	7.3e+11
gap5	2.9e+04	2.4e-03	8.1e+08	8.1e+08	3.4e+11	3.4e+11
gap6	3.0e+04	1.9e-04	1.0e+09	1.0e+09	5.3e+12	5.3e+12
gap7	3.0e+04	1.4e-03	4.3e+09	4.3e+09	3.0e+12	3.0e+12
gap8	3.1e+04	2.4e-04	1.1e+09	1.1e+09	4.6e+12	4.6e+12
gap9	2.7e+04	2.6e-03	3.2e+09	3.2e+09	1.3e+12	1.3e+12
gap10	3.1e+04	4.2e-03	8.5e+08	8.5e+08	2.0e+11	2.0e+11
gap11	3.2e+04	2.6e-04	4.3e+09	4.3e+09	1.7e+13	1.7e+13
gap12	2.8e+04	6.7e-03	1.9e+09	1.9e+09	2.9e+11	2.9e+11
gap13	2.5e+04	1.1e-03	6.9e+08	6.9e+08	6.1e+11	6.1e+11
gap14	2.4e+04	6.4e-03	9.8e+08	9.8e+08	1.5e+11	1.5e+11
gap15	2.5e+04	2.8e-04	2.1e+09	2.1e+09	7.2e+12	7.2e+12
gap16	2.4e+04	3.1e-03	5.0e+09	5.0e+09	1.6e+12	1.6e+12
gap17	2.4e+04	2.4e-04	7.1e+08	7.1e+08	3.0e+12	3.0e+12
gap18	2.1e+04	3.0e-04	7.1e+08	7.1e+08	2.3e+12	2.3e+12
gap19	2.5e+04	5.1e-03	1.9e+09	1.9e+09	3.7e+11	3.7e+11
gap20	2.0e+04	4.2e-03	1.4e+09	1.4e+09	3.3e+11	3.3e+11
gap21	1.6e+04	1.1e-03	4.1e+09	4.1e+09	3.7e+12	3.7e+12
gap22	2.3e+04	4.0e-03	7.0e+08	7.0e+08	1.7e+11	1.7e+11
gap23	1.5e+04	1.9e-03	4.5e+09	4.5e+09	2.3e+12	2.3e+12
gap24	1.5e+04	8.0e-03 [2]	4.4e+09	4.4e+09	5.4e+11	5.4e+11

Table 6.2: Renegar's condition number on SDPs with strict complementarity gaps. Notation from [31]: $(\rho_P(d))$ - distance to primal infeasibility; $(\rho_D(d))$ - distance to dual infeasibility; $(\|d\|_l, \|d\|_u)$ - lower and upper bounds of the norm of the data; $(C(d)_l, C(d)_u)$ - lower and upper bounds on Renegar's condition number, $C(d) = \frac{\|d\|}{\min\{\rho_P(d), \rho_D(d)\}}$.

6.4.4 SDPLIB Instances

Our results in Section 6.4 show that, generally, measure g_t can accurately measure the gap g, though it can give large errors when the solution estimates are not accurate enough. The measure g_s is more consistent in measuring the strict complementarity gap, g. The measure κ is also sensitive to the accuracy of the solution.

We applied these measures g_t, g_s, and κ to the SDPLIB [11] problem set. Though we used 10^{-10} as the stop tolerance in SDPT3, it was rarely attained. For some of the problems, there were big discrepancies between the two measures g_t and g_s. There was also no significant correlation between the iteration numbers and the three measures:

$$\text{corr}\,(g_t, \text{its}\,) = -0.01,\ \text{corr}\,(g_s, \text{its}\,) = -0.067,\ \text{and corr}\,(\kappa, \text{its}\,) = 0.2856.$$

However, if we only consider those SDP instances (47 such instances), where the error obtained was less than 10^{-7}, then we see a significant increase in the correlations between the measures and the iteration numbers:

$$\text{corr}\,(g_t, \text{its}\,) = 0.1472,\ \text{corr}\,(g_s, \text{its}\,) = 0.4509,\ \text{and corr}\,(\kappa, \text{its}\,) = 0.4371.$$

Their plots are shown in Figure 6.10.

6.5 Summary

We have presented an algorithm for generating hard SDP instances, i.e. problem instances where we can control the strict complementarity gap, g. We then tested several measures on randomly generated instances. The tests confirm the intuitive expectation: *The number of iterations for interior point methods are closely related to the size of the strict complementarity gaps.* In addition, we tested three measures g_t, g_s, and κ on the generated hard SDP instances. These measures g_t, g_s generally provide accurate measurement of the strict complementarity gaps; with the measure g_s being more consistent. All three measures are negatively affected by inaccurate solution estimates.

Our numerics show that the failure of Slater's condition coupled with large strict complementarity gap give the hardest problem for SDPT3. For these problems, SDPT3 general

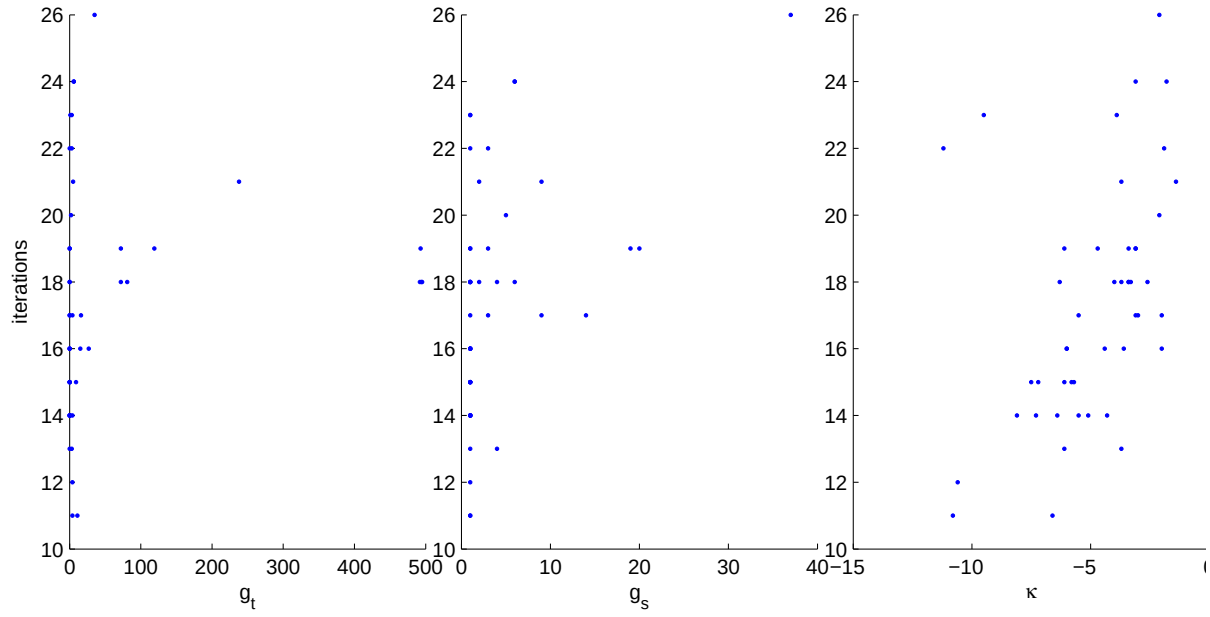

Figure 6.10: Scatter plots of g_t, g_s, κ versus # iterations for SDPLIB instances with attained tolerance $< 10^{-7}$.

takes more iterations and can not get desired accuracy. Also, the local convergence rate is slow. However, the fail of Slater's condition gives a better estimates for those measures (g_t, g_s, κ) for the strict complementarity gap compared to the case that the Slater's condition holds.

We also tested the aggregated geometrical measure and Renegar's condition number on the generated hard SDP instances; and, we did not find any correlation between them and the size of the strict complementarity gaps. It appears that these geometric measures are more closely related to the distance to infeasibility, i.e. strict feasibility. Our randomly generated hard SDP instances have consistently large aggregate geometrical measure and large Reneger condition number, despite having different strict complementarity gap values.

Finally, we used the SDPLIB test set but had trouble coming to any concrete conclusions since the approximate solutions we found were not accurate enough. We hope to obtain improved solutions and redo these tests in the future.

Chapter 7

Conclusions

7.1 Contributions

We list below the main contributions of the thesis.

1. We give an analysis on the error bounds for the NEQ based search directions. Our numerical examples show that all of these bounds are tight. These error bounds explain well why we usually do not see significant numerical instability despite the extremely large ill-conditioning for the underlying linear system. The error analysis suggests that for certain degenerate cases, the NEQ based approach will not have major problem up to 10^{-8}. This explains why most of the popular solvers set the default stop tolerance to 10^{-8}.

2. Based on the error analysis for NEQ, we propose an alternative approach. We show in theory that the underlying linear system for our approach is non-singular under a non-degeneracy assumption. Our numerical tests show that this approach has better numerical stability for both the NETLIB problems (even for degenerate problems) and our randomly generated problems. However, due to the larger linear system, we are not competitive in terms of CPU time for the NETLIB problems compared to NEQ. If the data sets have some nice properties, i.e. well-conditioned, sparse, and large scale, our proposed approach beats popular NEQ based approaches by a large margin. We also analyze some special techniques which can be incorporated into our approach. They

are purification, crossover, and no backtracking.

3. We present an algorithm to construct SDP instances with prescribed strict complementarity gap. We use this algorithm to construct a group of hard instances. We test a few measures on the strict complementarity gaps and find out that high accuracy solution can give correct measures. We also find out that the failure of the Slater's condition coupled with large strict complementarity gap yields the hardest problems. SDPT3 needs significantly more iterations for these hard problems and gets less accurate solutions. We empirically test the relation between the strict complementarity gap and the geometrical measure and Renegar's condition number; no relation is found.

7.2 Future Research Directions

1. To extend our LP error analysis to the SDP. This appears to be more challenging due to the complicated structure of SDP.

2. We know from our error analysis that the eigenvalues of the matrix $AXZ^{-1}A^T$ in LP split into two parts. Each part of the eigenvalues is relatively close. It is well know that the convergence rate of an iterative linear solver depends on the clustering of the eigenvalues. Can we take advantage of this property and design a fast iterative solver for the linear system of $AXZ^{-1}A^T$? Also, Our error analysis shows the size of the error on the matrix $AXZ^{-1}A^T$ and the correspond right-hand side. These error sizes can be a good suggestion of the stop tolerance for iterative linear solver. Moreover, our error analysis shows that a centering direction usually has large errors. Since iterative linear solver may not be as accurate as a direct linear solver, this may suggest that when using iterative linear solver, we should avoid using a centering direction.

3. For the new simple stable approach, can we find any better preconditioners cheaply? This is a challenge problem. Our iterative solver only works faster on well-conditioned system due to the lack of good preconditioner. We may want to exploit the special structure of the linear system.

4. Our algorithm for hard instances for SDP requires a special matrix A_1 such that

trace $(A_1 X) = 0$. Notice that the right-hand side has to be 0. Is it possible to design an algorithm which does not require that one of the right-hand side is zero? This can be useful for simulating certain classes of problems. For example, the SDP relaxation for max-cut problems has right-hand side all 1s.

5. The hard instances chapter gives many interesting numerical results. For example, we show that strict complementarity gaps are correlated with iteration numbers as well as local convergence rate. We also show that the failure of Slater's condition has interesting effects on the iteration numbers, the numerical stability, local convergence rate, and the accuracy of strict complementarity gaps measures g_t, g_s, and κ. It will be interesting to have a theoretical explanation on these phenomena.

Bibliography

[1] F. ALIZADEH. *Combinatorial optimization with interior point methods and semidefinite matrices*. PhD thesis, University of Minnesota, 1991.

[2] F. ALIZADEH. Optimization over positive semi-definite cone; interior-point methods and combinatorial applications. In P.M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 1–25. North–Holland, 1992.

[3] F. ALIZADEH. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[4] F. ALIZADEH, J.-P. A. HAEBERLY, and M. L. OVERTON. Primal–dual interior–point methods for semidefinite programming : Convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 8:746–768, 1998.

[5] F. ALIZADEH, J-P.A. HAEBERLY, and M.L. OVERTON. A new primal-dual interior-point method for semidefinite programming. In J.G. Lewis, editor, *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, pages 113–117. SIAM, 1994.

[6] E.D. ANDERSEN and Y. YE. Combining interior-point and pivoting algorithms for linear programming. *Management Science*, 42:1719–1731, 1996.

[7] K.M. ANSTREICHER. Linear programming in $O((n^3/\ln n)L)$ operations. *SIAM Journal on Optimization*, 9(4):803–812 (electronic), 1999. Dedicated to John E. Dennis, Jr., on his 60th birthday.

[8] E. R. BARNES. A variation on Karmarkar's algorithm for solving linear programming problems. *Mathematical Programming*, 36:174–182, 1986.

[9] M. BENZI, C.D. MEYER, and M. TŮMA. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM Journal on Scientific Computing*, 17(5):1135–1149, 1996.

[10] M. BENZI and M. TŮMA. A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 19(3):968–994, 1998.

[11] B. BORCHERS. SDPLIB 1.2, a library of semidefinite programming test problems. *Optimization Methods and Software*, 11(1):683–690, 1999. Interior point methods.

[12] K.H. BORGWARDT and P. HUHN. A lower bound on the average number of pivotsteps for solving linear programs. Valid for all variants of the simplex-algorithm. *Mathematical Methods of Operations Research*, 49(2):175–210, 1999.

[13] J.M. BORWEIN and H. WOLKOWICZ. Regularizing the abstract convex program. *Journal of Mathematical Analysis and Applications*, 83(2):495–530, 1981.

[14] M. BROOKES. The matrix reference manual, 2005. [online] `http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html`.

[15] J.V. BURKE. On the identification of active constraints. II. The nonconvex case. *SIAM Journal on Numerical Analysis*, 27(4):1081–1103, 1990.

[16] J.V. BURKE and J.J. MORÉ. On the identification of active constraints. *SIAM Journal on Numerical Analysis*, 25(5):1197–1211, 1988.

[17] J.V. BURKE and J.J. MORÉ. Exposing constraints. *SIAM Journal on Optimization*, 4(3):573–595, 1994.

[18] G.B. DANTZIG. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963, 1963.

[19] E. de KLERK. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications.* Applied Optimization Series. Kluwer Academic, Boston, MA, 2002.

[20] E. de KLERK, J. PENG, C. ROOS, and T. TERLAKY. A scaled gauss–newton primal-dual search direction for semidefinite optimization. *SIAM Journal on Optimization*, 11(4):870–888, 2001.

[21] E. de KLERK, C. ROOS, and T. TERLAKY. Initialization in semidefinite programming via a self-dual skew-symmetric embedding. *Operations Research Letters*, 20(5):213–221, 1997.

[22] R. DE LEONE and O.L. MANGASARIAN. Serial and parallel solution of large scale linear programs by augmented Lagrangian successive overrelaxation. In *Optimization, parallel processing and applications (Oberwolfach, 1987 and Karlsruhe, 1987)*, volume 304 of *Lecture Notes in Econom. and Math. Systems*, pages 103–124. Springer, Berlin, 1988.

[23] J.E. DENNIS Jr. and R.B. SCHNABEL. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Corrected reprint of the 1983 original.

[24] J.E. DENNIS Jr. and H. WOLKOWICZ. Sizing and least-change secant methods. *SIAM Journal on Numerical Analysis*, 30(5):1291–1314, 1993.

[25] J. DIEUDONNÉ. *Foundations of Modern Analysis*. Academic Press, New York, 1960.

[26] I. I. DIKIN. Iterative solution of problems of linear and quadratic programming. *Doklady Akademii Nauk SSSR*, 174:747–748, 1967. Translated in : *Soviet Mathematics Doklady* 8:674–675, 1967.

[27] A.S. EL-BAKRY, R.A. TAPIA, and Y. ZHANG. A study of indicators for identifying zero variables in interior-point methods. *SIAM Review*, 36(1):45–72, 1994.

[28] A.V. FIACCO and G.P. McCORMICK. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second (first 1968) edition, 1990.

[29] A. L. FORSGREN and P. E. GILL. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8:1132–1152, 1998.

[30] R.M. FREUND. Complexity of convex optimization using geometry-based measures and a reference point. *Mathematical Programming*, 99(2, Ser. A):197–221, 2004.

[31] R.M. FREUND, F. ORDÓÑEZ, and K.C. TOH. Behavioral measures and their correlation with IPM iteration counts on semi-definite programming problems. USC-ISE working paper #2005-02, MIT, 2005. url: http://www-rcf.usc.edu/˜fordon/.

[32] R.M. FREUND and J.R. VERA. Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm. *SIAM Journal on Optimization*, 10(1):155–176 (electronic), 1999.

[33] R.W. FREUND, M.H. GUTKNECHT, and N.M. NACHTIGAL. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM Journal on Scientific Computing*, 14:137–158, 1993.

[34] R.W. FREUND and F. JARRE. A QMR-based interior-point algorithm for solving linear programs. *Mathematical Programming, Series B*, 76:183–210, 1996.

[35] K.R. FRISCH. The logarithmic potential method of convex programming. Technical report, Institute of Economics, Oslo University, Oslo, Norway, 1955.

[36] P. E. GILL, W. MURRAY, M. A. SAUNDERS, J. A. TOMLIN, and M. H. WRIGHT. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Mathematical Programming*, 36:183–209, 1986.

[37] M.X. GOEMANS and D.P. WILLIAMSON. .878-approximation algorithms for MAX CUT and MAX 2SAT. In *ACM Symposium on Theory of Computing (STOC)*, 1994.

[38] M.X. GOEMANS and D.P. WILLIAMSON. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association of Computing Machinery*, 42(6):1115–1145, 1995.

[39] D. GOLDFARB and K. SCHEINBERG. Interior point trajectories in semidefinite programming. *SIAM Journal on Optimization*, 8(4):871–886, 1998.

[40] A.J. GOLDMAN and A.W. TUCKER. Theory of linear programming. In *Linear inequalities and related systems*, pages 53–97. Princeton University Press, Princeton, N.J., 1956. Annals of Mathematics Studies, no. 38.

[41] M. GONZALEZ-LIMA, H. WEI, and H. WOLKOWICZ. A stable iterative method for linear programming. Technical Report CORR 2004-26, University of Waterloo, Waterloo, Ontario, 2004.

[42] N.I.M. GOULD, D. ORBAN, A. SARTENAER, and Ph.L. TOINT. Componentwise fast convergence in the solution of full-rank systems of nonlinear equations. Tr/pa/00/56, CERFACS, Toulouse Cedex 1, France, 2001.

[43] A. GREENBAUM. *Iterative methods for solving linear systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[44] O. GÜLER, D. DEN HERTOG, C. ROOS, T. TERLAKY, and T. TSUCHIYA. Degeneracy in interior point methods for linear programming: a survey. *Annals of Operations Research*, 46/47(1-4):107–138, 1993. Degeneracy in optimization problems.

[45] W.W. HAGER. The dual active set algorithm and the iterative solution of linear programs. In *Novel approaches to hard discrete optimization (Waterloo, ON, 2001)*, volume 37 of *Fields Inst. Commun.*, pages 97–109. Amer. Math. Soc., Providence, RI, 2003.

[46] M. HALICKÁ, E. de KLERK, and C. ROOS. On the convergence of the central path in semidefinite optimization. *SIAM Journal on Optimization*, 12(4):1090–1099 (electronic), 2002.

[47] P.C. HANSEN and P.Y. YALAMOV. Computing symmetric rank-revealing decompositions via triangular factorization. *SIAM Journal on Matrix Analysis and Applications*, 23(2):443–458 (electronic), 2001.

[48] C. HELMBERG, F. RENDL, R.J. VANDERBEI, and H. WOLKOWICZ. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2):342–361, 1996.

[49] N.J. HIGHAM. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 1995.

[50] J. JI, F.A. POTRA, and R. SHENG. On the local convergence of a predictor-corrector method for semidefinite programming. *SIAM Journal on Optimization*, 10(1):195–210 (electronic), 1999.

[51] J.J. JÚDICE, J. PATRICIO, L.F. PORTUGAL, M.G.C. RESENDE, and G. VEIGA. A study of preconditioners for network interior point methods. *Computational Optimization and Applications*, 24(1):5–35, 2003.

[52] L.V. KANTOROVICH. Functional analysis and applied mathematics. *Uspekhi Mat. Nauk.*, 3:89–185, 1948. Transl. by C. Benster as N.B.S. Rept. 1509, Washington D.C., 1952.

[53] N. K. KARMARKAR. A new polynomial–time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[54] L. G. KHACHIYAN. A polynomial algorithm for linear progrmming. *Soviet Math. Dokl.*, 20:191–194, 1979.

[55] L. G. KHACHIYAN. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Math. Phys.*, 20:53–72, 1980.

[56] V. KLEE and G.J. MINTY. How good is the simplex algorithm. In O. Shisha, editor, *Inequalities - III*. Academic Press Inc., New York and London, 1972.

[57] A.A. KLYACHKO. Stable bundles, representation theory and Hermitian operators. *Selecta Math. (N.S.)*, 4(3):419–445, 1998.

[58] D.E. KNUTH. The sandwich theorem. *Electronic J. Combinatorics*, 1:48pp, 1994.

[59] M. KOJIMA, M. SHIDA, and S. SHINDOH. Local convergence of predictor–corrector infeasible–interior–point algorithms for SDPs and SDLCPs. *Mathematical Programming*, 80:129–160, 1998.

[60] M. KOJIMA, S. SHINDOH, and S. HARA. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7(1):86–125, 1997.

[61] S. KRUK. *High Accuracy Algorithms for the Solutions of Semidefinite Linear Programs*. PhD thesis, University of Waterloo, 2001.

[62] S. KRUK, M. MURAMATSU, F. RENDL, R.J. VANDERBEI, and H. WOLKOWICZ. The Gauss-Newton direction in linear and semidefinite programming. *Optimization Methods and Software*, 15(1):1–27, 2001.

[63] L. LOVÁSZ. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(1):1–7, 1979.

[64] Z-Q. LUO, J. F. STURM, and S. ZHANG. Superlinear convergence of a symmetric primal-dual path-following algorithm for semidefinite programming. *SIAM Journal on Optimization*, 8:59–81, 1998.

[65] O.L. MANGASARIAN. Iterative solution of linear programs. *SIAM Journal on Numerical Analysis*, 18(4):606–614, 1981.

[66] A.W. MARSHALL and I. OLKIN. *Inequalities: Theory of Majorization and its Applications*. Academic Press, New York, NY, 1979.

[67] L. McLINDEN. The analogue of Moreau's proximation theorem, with applications to the nonlinear complementarity problem. *Pacific Journal of Mathematics*, 88:101–161, 1980.

[68] S. MEHROTRA. On the implementation of a primal–dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[69] S. MEHROTRA and Y. YE. Finding an interior point in the optimal face of linear programs. *Mathematical Programming*, 62(3, Ser. A):497–515, 1993.

[70] H. D. MITTELMANN. An independent benchmarking of SDP and SOCP solvers. *Mathematical Programming*, 95(2, Ser. B):407–430, 2003. Computational semidefinite and second order cone programming: the state of the art.

[71] R.D.C. MONTEIRO. Primal-dual path-following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7(3):663–678, 1997.

[72] R.D.C. MONTEIRO and M.J. TODD. Path-following methods. In *Handbook of Semidefinite Programming*, pages 267–306. Kluwer Acad. Publ., Boston, MA, 2000.

[73] Y.E. NESTEROV and A.S. NEMIROVSKI. Self–concordant functions and polynomial–time methods in convex programming. Book–Preprint, Central Economic and Mathematical Institute, USSR Academy of Science, Moscow, USSR, 1989. Published in Nesterov and Nemirovsky [74].

[74] Y.E. NESTEROV and A.S. NEMIROVSKI. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM Publications. SIAM, Philadelphia, USA, 1994.

[75] Y.E. NESTEROV and M.J. TODD. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1):1–42, 1997.

[76] Y.E. NESTEROV and M.J. TODD. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8:324–364, 1998.

[77] J. NOCEDAL and S.J. WRIGHT. *Numerical optimization*. Springer-Verlag, New York, 1999.

[78] A.R.L. OLIVEIRA and D.C. SORENSEN. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and its Applications*, 394:1–24, 2005.

[79] C.C. PAIGE and M.A. SAUNDERS. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.

[80] J. PANG. Error bounds in mathematical programming. *Mathematical Programming*, 79(1-3, Ser. B):299–332, 1997. Lectures on mathematical programming (ismp97) (Lausanne, 1997).

[81] G. PATAKI and L. TUNÇEL. On the generic properties of convex optimization problems in conic form. *Mathematical Programming*, 89(Ser. A):449–457, 2001.

[82] S. PEREZ-GARCIA. Alternative iterative primal-dual interior-point algorithms for linear programming. Master's thesis, Simon Bolivar University, Center for Statistics and Mathematical Software (CESMa), Venezuela, 2003.

[83] S. PEREZ-GARCIA and M. GONZALEZ-LIMA. On a non-inverse approach for solving the linear systems arising in primal-dual interior point methods for linear programming. Technical Report 2004-01, Simon Bolivar University, Center for Statistical and Mathematical Software, Caracas, Venezuela, 2004.

[84] F.A. POTRA and R. SHENG. Superlinear convergence of a predictor-corrector method for semidefinite programming without shrinking central path neighborhood. *Bull. Math. Soc. Sci. Math. Roumanie (N.S.)*, 43(91)(2):107–124, 2000.

[85] M.V. RAMANA, L. TUNÇEL, and H. WOLKOWICZ. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7(3):641–662, 1997.

[86] J. RENEGAR. Linear programming, complexity theory and elementary functional analysis. *Mathematical Programming*, 70(3, Ser. A):279–351, 1995.

[87] A. SHAPIRO. Duality and optimality conditions. In *HANDBOOK OF SEMIDEFINITE PROGRAMMING: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Boston, MA, 2000.

[88] N. SHOR. Utilization of the operation of space dilatation in the minimization of convex functions. *Kibernetika*, 1:6–12, 1970. (In Russian). Translated in : *Cybernetics, 6,* 7-15.

[89] S. SMALE. On the average number of steps of the simplex method of linear programming. *Mathematical Programming*, 27(3):241–262, 1983.

[90] D. A. SPIELMAN and S.-H. TENG. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463 (electronic), 2004.

[91] G. W. STEWART and J.-G. SUN. *Matrix Perturbation Theory*. Academic Press, Boston, 1990.

[92] G.W. STEWART. Updating a rank-revealing ULV decomposition. *SIAM Journal on Matrix Analysis and Applications*, 14(2):494–499, 1993.

[93] J.F. STURM. Avoiding numerical cancellation in the interior point method for solving semidefinite programs. Technical Report 2001-27, Tilburg University, The Netherlands, 2001.

[94] R. A. Tapia and Y. Zhang. On the quadratic convergence of the singular Newton's method. *SIAG/OPT Views–and–News, A forum for the SIAM Activity Group on Optimization*, 1:6–8, 1992.

[95] M.J. TODD. A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optimization Methods and Software*, 11&12:1–46, 1999.

[96] M.J. TODD, K.C. TOH, and R.H. TÜTÜNCÜ. On the Nesterov-Todd direction in semidefinite programming. *SIAM Journal on Optimization*, 8(3):769–796, 1998.

[97] L. Tunçel. On the convergence of primal–dual interior point methods with wide neighborhoods. *Computational Optimization and Applications*, 4:139–158, 1995.

[98] L. TUNÇEL and H. WOLKOWICZ. Strengthened existence and uniqueness conditions for search directions in semidefinite programming. *Linear Algebra and its Applications*, 400:31–60, 2005.

[99] R.H. TÜTÜNCÜ, K.C. TOH, and M.J. TODD. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2, Ser. B):189–217, 2003. Computational semidefinite and second order cone programming: the state of the art.

[100] A. VAN der SLUIS. Condition numbers and equilibration of matrices. *Numerische Mathematik*, 14:14–23, 1969/1970.

[101] A. VAN der SLUIS. Stability of solutions of linear algebraic systems. *Numerische Mathematik*, 14:246–251, 1969/1970.

[102] R. J. VANDERBEI, M. S. MEKETON, and B. A. FREEDMAN. A modification of Karmarkar's linear programming algorithm. *Algorithmica*, 1(4):395–407, 1986.

[103] R.J. VANDERBEI. *Linear Programming: Foundations and Extensions*. Kluwer Acad. Publ., Dordrecht, 1998.

[104] R.J. VANDERBEI. LOQO: an interior point code for quadratic programming. *Optimization Methods and Software*, 11/12(1-4):451–484, 1999. Interior point methods.

[105] S.A. VAVASIS and Y. YE. A primal-dual interior point method whose running time depends only on the constraint matrix. *Mathematical Programming*, 74(1, Ser. A):79–120, 1996.

[106] H. WEI and H. WOLKOWICZ. Generating and solving hard instances in semidefinite programming. Technical Report CORR 2006-01, University of Waterloo, Waterloo, Ontario, 2005. in progress.

[107] H. WOLKOWICZ. Solving semidefinite programs using preconditioned conjugate gradients. *Optimization Methods and Software*, 19(6):653–672, 2004.

[108] H. WOLKOWICZ, R. SAIGAL, and L. VANDENBERGHE, editors. *HANDBOOK OF SEMIDEFINITE PROGRAMMING: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Boston, MA, 2000. xxvi+654 pages.

[109] M.H. WRIGHT. *Numerical methods for nonlinearly constrained optimization*. PhD thesis, Department of Computer Science, Stanford University, 1976.

[110] M.H. WRIGHT. The interior-point revolution in constrained optimization. In *High performance algorithms and software in nonlinear optimization (Ischia, 1997)*, volume 24 of *Appl. Optim.*, pages 359–381. Kluwer Acad. Publ., Dordrecht, 1998.

[111] M.H. WRIGHT. Ill-conditioning and computational error in interior methods for nonlinear programming. *SIAM Journal on Optimization*, 9(1):84–111 (electronic), 1999.

[112] S.J. WRIGHT. Stability of linear equations solvers in interior-point methods. *SIAM Journal on Matrix Analysis and Applications*, 16(4):1287–1307, 1995.

[113] S.J. WRIGHT. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, 1996.

[114] S.J. WRIGHT. Modifying SQP for degenerate problems. Technical report, Argonne National Laboratory, 1997.

[115] S.J. WRIGHT. Stability of augmented system factorizations in interior-point methods. *SIAM Journal on Matrix Analysis and Applications*, 18(1):191–222, 1997.

[116] S.J. WRIGHT. Modified Cholesky factorizations in interior-point algorithms for linear programming. *SIAM Journal on Optimization*, 9(4):1159–1191 (electronic), 1999. Dedicated to John E. Dennis, Jr., on his 60th birthday.

[117] Y. YE. *Interior point algorithms*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1997. Theory and analysis, A Wiley-Interscience Publication.

[118] Y. YE. *Interior Point Algorithms: Theory and Analysis*. Wiley-Interscience series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1997.

[119] Y. Ye, O. Güler, R. A. Tapia, and Y. Zhang. A quadratically convergent $O(\sqrt{n}L)$–iteration algorithm for linear programming. *Mathematical Programming*, 59:151–162, 1993.

[120] D. YUDIN and A. NEMIROVSKII. Informational complexity and efficient methods for the solution of convex extremal problems. *Èkon.i Mat. Metody*, 12:357–369, 1976. (In Russian). Translated in : *Matekon 13(2)* 3-25.

[121] Y. ZHANG. On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming. *SIAM Journal on Optimization*, 8:365–386, 1998.

[122] Y. ZHANG. User's guide to LIPSOL: linear-programming interior point solvers V0.4. *Optimization Methods and Software*, 11/12(1-4):385–396, 1999. Interior point methods.