# Convergence of a Class of Inexact Interior-Point Algorithms for Linear Programs*

Roland W. Freund
Bell Laboratories
Lucent Technologies
700 Mountain Avenue, Room 2C–420
Murray Hill, New Jersey 07974–0636, U.S.A.

Florian Jarre
Institut für Angewandte Mathematik und Statistik
Universität Würzburg, Am Hubland
D–97074 Würzburg, Federal Republic of Germany

Shinji Mizuno
The Institute of Statistical Mathematics
Department of Prediction and Control
4-6-7 Minami-Azabu, Minato-Ku, Tokyo 106, Japan

**Abstract.** We present a convergence analysis for a class of inexact infeasible-interior-point methods for solving linear programs. The main feature of inexact methods is that the linear systems defining the search direction at each interior-point iteration need not be solved to high accuracy. More precisely, we allow that these linear systems are only solved to a moderate accuracy in the *residual*, but no assumptions are made on the accuracy of the search direction in the search space. In particular, our analysis does not require that feasibility is maintained even if the initial iterate happened to be a feasible solution of the linear program. We also present a few numerical examples to illustrate the effect of using inexact search directions on the number of interior-point iterations.

*AMS 1991 subject classification.* Primary: 90C05, Secondary: 65K05, 90C06.

*Key words.* Linear program, infeasible-interior-point method, inexact search direction, linear system, residual, convergence.

## 1. Introduction

Since the publication [9] of Karmarkar's original interior-point algorithm for linear programs, numerous variants of the method have been developed; see, e.g., Kojima, Mizuno, and Yoshise [11],

*Numerical Analysis Manuscript No. 97–3–11, Bell Laboratories, Murray Hill, New Jersey, October 1997. Available on WWW at {`http://cm.bell-labs.com/cs/doc/97`}.

Megiddo [14], Monteiro and Adler [18], and Tanabe [21]. Especially interesting is the algorithm by Kojima, Megiddo, and Mizuno [10], since it is also practically efficient; see, e.g., [12].

### 1.1 Inexact Interior-Point Methods

The motivation for the study in this paper is to provide a theoretical justification for recent modifications of the algorithm [10] in which the search directions are computed only inexactly at each interior-point iteration. One feature of such inexact interior-point methods is that they allow the use of iterative techniques to approximately solve the linear systems defining the search directions. For example, in [3], we employ a preconditioned QMR algorithm for the iterative solution of these linear systems.

First, we describe a modification of the algorithm in [10] that is adapted to inexact computations of the search directions, and then we show its global convergence. Our goal is to design an algorithm such that the number of iterations to solve a given linear program under "reasonably" inexact computations is not much higher than for the algorithm in [12]. We are aware that the algorithm in [12] has been improved in the meantime; see, e.g., [13] and many other more recent publications. However, we are not aware of any more efficient algorithm that is readily extended to inexact computations.

In some applications, an initial strictly dual-feasible solution of the linear program is known. For such problems, in [19], an implementation of a primal-infeasible dual-feasible interior-point method is presented and results of numerical experiments are reported. Our approach originates from [3], and it is distinctly different from [19]. In particular, the method of our paper also converges even if there is no strictly dual-feasible solution.

### 1.2 Outline

The outline of this article is as follows. In Section 2, we formulate the linear program in standard form and state our assumptions. In Section 3, we describe the class of inexact interior-point methods studied in this paper. In Section 4, we discuss issues in defining appropriate inexact search directions. In particular, we explain why we assume a certain accuracy requirement, and how we can achieve it. In Section 5, we establish global convergence. In Section 6, we present a few numerical examples to illustrate the effect of using inexact search directions on the number of interior-point iterations. In Section 7, we make some concluding remarks.

### 1.3 Notation

Throughout the paper, we use the following notation. Lower-case letters are used for vectors, and upper-case letters are used for matrices. For a vector denoted by a lower-case letter, the upper-case version means the diagonal matrix whose diagonal elements are the components of the vector. For example,

$$X = \operatorname{diag}(x_1, x_2, \ldots, x_n)$$

is the diagonal matrix associated with the vector $x = [\,x_1 \quad x_2 \quad \cdots \quad x_n\,]^T \in \mathbb{R}^n$. By $e$ we denote the vector $e = [\,1 \quad 1 \quad \cdots \quad 1\,]^T \in \mathbb{R}^n$. The vector norm $\|x\| = \sqrt{x^T x}$ is always the Euclidean norm, and $\|M\| = \max_{\|x\|=1} \|Mx\|$ denotes the corresponding Euclidean matrix norm. The notation $x > 0$ (respectively, $x \geq 0$) means that all components of the vector $x$ are positive (respectively, nonnegative). Instead of $x \geq 0$, we also write $x \in \mathbb{R}^n_+$. The same symbol, 0, will be used to denote the number zero, the zero vector, and the zero matrix. When 0 denotes the zero vector or the zero matrix, then its length or dimension, respectively, will always be clear from the context. The capital letter $I$ is used to denote the (square) identity matrix; its size will always be apparent from the context as well.

# 2. The Problem

## 2.1 Formulation and Optimality Conditions

We consider linear programs in the standard form

$$\text{(1)} \qquad \underset{x\in\mathbb{R}^n}{\text{minimize}} \; c^T x \qquad \text{subject to} \quad Ax = b \quad \text{and} \quad x \geq 0.$$

We assume that $A$ is a given real $m \times n$ matrix, and $b$ and $c$ are given real vectors of length $m$ and $n$, respectively. The dual problem to (1) can be written as

$$\text{(2)} \qquad \underset{s\in\mathbb{R}^n,\; y\in\mathbb{R}^m}{\text{maximize}} \; b^T y \qquad \text{subject to} \quad A^T y + s = c \quad \text{and} \quad s \geq 0.$$

The Karush-Kuhn-Tucker optimality conditions for the primal-dual problems (1) and (2) can be stated as follows:

$$\begin{aligned} &\text{(3)} & Ax &= b, \\ &\text{(4)} & A^T y + s &= c, \\ &\text{(5)} & Xs &= 0, \\ &\text{(6)} & x,\, s &\geq 0. \end{aligned}$$

Equations (3) and (4) are referred to as primal and dual feasibility, respectively, and the nonlinear equation (5) is called the complementarity condition.

## 2.2 Assumptions

In the sequel, we always assume that $A$ has full row rank. Theoretically, this assumption could be enforced by computing a singular value decomposition of $A$, but given a sparse constraint matrix $A$, this is not done in practice. Instead, a given linear program is typically preprocessed before an interior-point method for its solution is applied; see, e.g., [1, 8]. For example, simple heuristics are used to detect fixed variables, empty rows and columns, redundancies, and certain implicit dependencies. These heuristics are often sufficient to reduce the linear program not only in size but also to a form that the matrix $A$ has linearly independent rows, even if the constraint matrix of (1) in its initial form has not.

The convergence analysis of our inexact interior-point method depends on the smallest singular value, $\sigma_{\min}(A)$, of $A$. The smaller $\sigma_{\min}(A)$, the higher the accuracy that is required at each step of the method. We now briefly discuss the size of $\sigma_{\min}(A)$, assuming that all rows of $A$ are scaled to Euclidean norm 1. We then have $1 \leq \sigma_{\max}(A) \leq \sqrt{m}$, where $\sigma_{\max}(A)$ is the largest singular value of $A$. When $\sigma_{\min}(A)$ is very small, the linear program (1) is not well-conditioned. However, assuming that $\sigma_{\min}(A)$ is large (say greater than 0.1) does not imply that the problem (1) is well-conditioned.

In fact, also for $0 \ll \sigma_{\min}(A) \leq 1$, the linear program may be degenerate, and then the pair of primal-dual problems has infinite condition number; see the appendix in[5]. In particular, prior to solving a given linear program (1), the problem could always be transformed into an equivalent linear program with a constraint matrix $A'$ such that $\sigma_{\min}(A') = 1$. Indeed, since $A$ is assumed to have full rank, the matrix $A^T$ admits a QR factorization of the form

$$A^T = QR = Q_1 R_1, \quad Q = [\, Q_1 \quad Q_2 \,], \quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \tag{7}$$

where $Q$ is an orthogonal $n \times n$ matrix with $Q_1$ denoting its first $m$ columns, and $R_1$ is a nonsingular $m \times m$ upper triangular matrix. By means of (7), the equality constraints in (1) can be rewritten as follows:

$$Ax = R_1^T Q_1^T x = b \quad \Longleftrightarrow \quad Q_1^T x = R_1^{-T} b. \tag{8}$$

The reformulation $Q_1^T x = R_1^{-T} b$ of the equality constraints is obviously well-conditioned with $\sigma_{\min}(Q_1^T) = 1$. Thus, theoretically, one might first compute the QR factorization (7) as well as $R_1^{-T} b$ (possibly with higher accuracy than the standard machine precision), and then solve the transformed linear program. Unfortunately, the reformulation (8) typically destroys the sparsity of the constraint matrix, as $Q_1^T$ may have considerably more nonzero entries than $A$, and therefore, this reformulation is not used in practice.

We do, however, assume in the sequel that a positive lower bound,

$$0 < \sigma \leq \sigma_{\min}(A), \tag{9}$$

for the smallest singular value of the original constraint matrix $A$ is available.

# 3. A Class of Inexact Interior-Point Algorithms

The algorithms considered in this paper generate a sequence of iterates that simultaneously converge to feasibility and optimality. To guarantee convergence, we require that all iterates lie in a certain set $\mathcal{N}$.

## 3.1 An Infinity-Norm-Neighborhood of the Perturbed Path

The set $\mathcal{N}$ depends on some initial point $(x^0, y^0, s^0)$ with $x^0 > 0$ and $s^0 > 0$.

**Definition.** *Let $x^0$, $s^0 \in \mathbb{R}^n$, $y^0 \in \mathbb{R}^m$ and two real numbers $\nu_p > 0$ and $\nu_d > 0$ be given such that $x^0 \geq \nu_p e$ and $s^0 \geq \nu_d e$. Set*

$$\bar{b} := Ax^0 - b, \quad \bar{c} := A^T y^0 + s^0 - c, \quad \bar{\mu} := \frac{(x^0)^T s^0}{n}. \tag{10}$$

*For a real number $\beta > 1$, a neighborhood $\mathcal{N}(\beta) = \mathcal{N}_{A,b,c,x^0,y^0,s^0}(\beta)$ is defined by the set of equalities* (11) *and the two sets of inequalities* (12) *and* (13) *below. A point $(x, y, s) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ belongs to $\mathcal{N}(\beta)$ if there exist*

$$\textit{vectors} \quad \tilde{c} \in \mathbb{R}^n, \quad \tilde{b} \in \mathbb{R}^m, \quad \textit{and numbers} \quad t_1, t_2, t_3 \in (0, 1]$$

*such that*

$$\text{(11)} \qquad Ax = b + t_1(\bar{b} + \tilde{b}), \quad A^T y + s = c + t_2(\bar{c} + \tilde{c}), \quad x^T s = t_3 n \bar{\mu},$$

$$\text{(12)} \qquad \|\tilde{b}\| \;\le\; \tfrac{1}{2}\nu_p \sigma, \quad \|\tilde{c}\| \;\le\; \tfrac{1}{2}\nu_d,$$

$$\text{(13)} \qquad \textit{and} \quad \frac{e}{\beta} \le \frac{Xs}{t_3\bar{\mu}}, \quad \frac{t_3}{\beta} \;\le\; t_1 \;\le\; t_3\beta, \quad \frac{t_3}{\beta} \;\le\; t_2 \;\le\; t_3\beta.$$

A closely-related neighborhood was proposed in [10]. The new element in the above definition is the perturbation vectors $\tilde{b}$, $\tilde{c}$ in (11) and (12); in [10], the vectors $\tilde{b}$ and $\tilde{c}$ are zero.

Unless some good initial estimate $(x^0, y^0, s^0)$ for the solution of problems (1) and (2) is available, we will think of $\nu_p$ and $\nu_d$ as large numbers, $\nu_p\ \nu_d \gg 0$, and of $x^0 \ge \nu_p e$, $s^0 \ge \nu_d e$ as large vectors. The set $\mathcal{N}$ is a "wide" infinity-norm-neighborhood of the "perturbed path"

$$\text{(14)} \qquad Ax \;=\; b + t\bar{b},$$

$$\text{(15)} \qquad A^T y + s \;=\; c + t\bar{c},$$

$$\text{(16)} \qquad Xs \;=\; t\bar{\mu}e, \quad x,\, s > 0.$$

It is well known that there exists a unique solution $(x, y, s)$ of (14)–(16) for each $t \in (0, 1]$, provided (3), (4), and (6) have a solution and $A$ has full rank; see, e.g., [11, 14]. Above, "infinity-norm" refers to the neighborhood with respect to the nonlinear equation $Xs = t_3\bar{\mu}e$, while the deviation from the linear equalities is measured in the Euclidean norm. The requirement on the size of $\tilde{b}$ in (12) could be weakened slightly by requiring $\|R_1^{-T}\tilde{b}\| \le \nu_p/2$ where $R_1$ is as in (7) and (8). However, our goal is to formulate a method that does not require the QR factorization (7). To emphasize this principle, the neighborhood is defined in terms of the lower bound (9), $\sigma$, on the smallest singular value of $A$.

### 3.2 The Newton Step

The class of interior-point algorithms considered in this paper can be viewed as variants of damped Newton's method applied to the system of equations

$$\text{(17)} \qquad Ax \;=\; b + \lambda_1\bar{b},$$

$$\text{(18)} \qquad A^T y + s \;=\; c + \lambda_2\bar{c},$$

$$\text{(19)} \qquad Xs \;=\; \lambda_3\bar{\mu}e.$$

Here, $\lambda_1$, $\lambda_2$, and $\lambda_3$ are nonnegative numbers. They are adjusted at each iteration of the interior-point method such that they converge to zero in the course of the algorithm. The step lengths in the damped Newton method are suitably chosen to guarantee that $(x, y, s) \in \mathcal{N}(\beta)$ for some $\beta > 1$.

Applying Newton's method to (17)–(19) yields the system of linear equations

$$\text{(20)} \qquad A\Delta x \;=\; b + \lambda_1\bar{b} - Ax,$$

$$\text{(21)} \qquad A^T\Delta y + \Delta s \;=\; c + \lambda_2\bar{c} - A^T y - s,$$

$$\text{(22)} \qquad S\Delta x + X\Delta s \;=\; \lambda_3\bar{\mu}e - Xs,$$

for the Newton correction $(\Delta x, \Delta y, \Delta s)$. We now denote by

$$r := b + \lambda_1 \bar{b} - Ax, \quad p := c + \lambda_2 \bar{c} - A^T y - s, \quad \text{and} \quad q := \lambda_3 \bar{\mu} e - Xs \tag{23}$$

the right-hand sides in (20)–(22). Then, after a reordering, the linear system (20)–(22) can be written in the form

$$\begin{bmatrix} X & S & 0 \\ I & 0 & A^T \\ 0 & A & 0 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} q \\ p \\ r \end{bmatrix}, \tag{24}$$

with a $3 \times 3$-block matrix. In our algorithm, the systems (24) are not solved exactly but only to a given accuracy in the primal and dual residual,

$$\|A\Delta x - r\| \leq \varepsilon_x, \qquad \|A^T \Delta y + \Delta s - p\| \leq \varepsilon_s, \tag{25}$$

$$\text{and} \quad X\Delta s + S\Delta x = q. \tag{26}$$

As we will explain in Section 4, it is meaningful and easily possible to solve the equation (26) exactly (up to machine precision), even when allowing possibly larger errors $\varepsilon_x$ and $\varepsilon_s$ for (20) and (21). In Section 4, we will also define appropriate values for $\varepsilon_x$ and $\varepsilon_s$.

### 3.3 An Infeasible-Interior-Point Algorithm

Let $(x, y, s) = (x^k, y^k, s^k)$ with $x^k, s^k > 0$ be the iterate at step $k$ of the interior-point method, and let $(\Delta x, \Delta y, \Delta s)$ be the solution of the linear system (24). Then, an improved iterate $(x^+, y^+, s^+) = (x^{k+1}, y^{k+1}, s^{k+1})$ at the next step, $k+1$, is obtained via

$$x^+ = x + \alpha_x \Delta x, \quad y^+ = y + \alpha_s \Delta y, \quad s^+ = s + \alpha_s \Delta s,$$

where the step lengths $\alpha_x$ and $\alpha_s$ are chosen such that $0 < \alpha_x, \alpha_s \leq \alpha_{\max} < 1$ and $x^+$, $s^+$ satisfy $x^+ > 0$ and $s^+ > 0$. Here, $\alpha_{\max}$ is a parameter that can be chosen very close to 1.

The class of algorithms considered in this paper is designed to support the iterative solution of the linear systems (24) (as in [3]). The concept of an algorithm using an iterative solver is based on the following principles.

- No predictor-corrector strategy is applied. If the solution of the linear systems is dominated by the iteration itself and not by the computation of the preconditioner, then the corrector step is nearly as expensive as the predictor step. This is in contrast to the standard methods based on direct solvers that use the same sparse factorization of some reduced version of the coefficient matrix of (24) for the predictor and the corrector step. However, if computing the preconditioner dominates the iterative solve, then employing a predictor-corrector strategy may well prove to be worthwhile.

- *Relatively short steps* (i.e., not 99.995% of the way to the boundary, as in [13]) are taken. This is done for two reasons. First, if the search direction is computed to low relative accuracy, a long step is not meaningful, and secondly, very long steps might worsen the condition number of the linear systems.

The stopping criterion in the algorithm below is based on the error measure

$$\delta(x,y,s) := \frac{\|Ax-b\|}{\max\{\,1,\ \|b\|\,\}} + \frac{\|A^Ty+s-c\|}{\max\{\,1,\ \|c\|\,\}} + \frac{x^Ts}{\max\{\,1,\ \|b\|\cdot\|c\|\,\}}. \tag{27}$$

The algorithm uses a number of parameters. An informal description of these parameters is given first.

- $t_1^k,\ t_2^k,\ t_3^k\ \in (0,1]$ along with $\bar{b},\ \bar{c},\ \bar{\mu}$, measure the primal and dual infeasibility, and the duality gap.

- $\lambda_1^k,\ \lambda_2^k,\ \lambda_3^k$ denote "target values" to which $t_1^k,\ t_2^k,\ t_3^k$ should ideally be reduced by the correction $(\Delta x,\ \Delta y,\ \Delta s)$ at step $k$.

**Algorithm 1** (Inexact infeasible-interior-point algorithm)

INPUT: *Parameters* $\epsilon,\ \nu_p,\ \nu_d > 0,\ \beta > 1,\ \alpha_{\max} < 1,$ *and an initial point* $(x^0, y^0, s^0)$ *with* $x^0 \geq \nu_p e,\ s^0 \geq \nu_d e.$

*Set* $\bar{b} = Ax^0 - b,\ \bar{c} = A^Ty^0 + s^0 - c,\ \bar{\mu} = (x^0)^Ts^0/n,$ *and* $t_1^1 = t_2^1 = t_3^1 = 1.$

*For* $k = 0, 1, \ldots,$ *do* :

1) *Compute* $\delta(x^k, y^k, s^k)$.
   *If* $\delta(x^k, y^k, s^k) \leq \epsilon$, *then stop.*
2) *Choose* $\lambda_1^k,\ \lambda_2^k \in [0, 1)$ *and* $\lambda_3^k \in (0, t_3)$.
3) *Compute* $(\Delta x, \Delta y, \Delta s)$ *by solving the linear system* (24) (*with right-hand side* (23) *and* $x = x^k,\ y = y^k,\ s = s^k$) *up to an error satisfying* (25) *with* $\varepsilon_x = \varepsilon_x^k$ *and* $\varepsilon_s = \varepsilon_s^k$.
4) *Choose* $\alpha_x^k,\ \alpha_s^k \in (0, \alpha_{\max}]$.
5) *Set*

$$\begin{aligned} x^{k+1} &= x^k + \alpha_x^k \Delta x, \\ y^{k+1} &= y^k + \alpha_s^k \Delta y, \\ s^{k+1} &= s^k + \alpha_s^k \Delta s. \end{aligned}$$

6) *Update* $t_1^k,\ t_2^k,\ t_3^k$ *to*

$$\begin{aligned} t_1^{k+1} &= (1-\alpha_x^k)t_1^k + \alpha_x^k \lambda_1^k, \\ t_2^{k+1} &= (1-\alpha_s^k)t_2^k + \alpha_s^k \lambda_2^k, \\ t_3^{k+1} &= \frac{1}{\bar{\mu}\, n}\,(x^{k+1})^T s^{k+1}. \end{aligned}$$

Next, we specify the choice of the parameters $\lambda_1^k,\ \lambda_2^k,\ \lambda_3^k,\ \varepsilon_x^k,\ \varepsilon_s^k$ and the step lengths $\alpha_x^k,\ \alpha_s^k$ in the above algorithm. This choice guarantees that all iterates remain in the neighborhood $\mathcal{N}(\beta)$; see Theorem 1.

### 3.4 Specifications of the Algorithm

The quantities $\alpha_x^k$, $\alpha_s^k$, and $\lambda_3^k$ depend on three additional input parameters $\chi_1$, $\chi_2$, and $\chi_3$ of the algorithm with

$$\frac{2}{1+\beta} < \chi_1 < \chi_2 < \chi_3 < 1.$$

Here, $\chi_1$ controls the "centering parameter" $\lambda_3^k$, and $\chi_2$, $\chi_3$ are used as safeguards when determining $\alpha_x^k$ and $\alpha_s^k$.

We now specify Steps 2), 3), and 4) of Algorithm 1.

**Step 2)**

We set

$$\lambda_1^k = \begin{cases} \min\{t_1^k, \lambda_3^k\} & \text{if } t_1^k < \frac{2}{1+\beta} t_3^k, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{and} \qquad \lambda_2^k = \begin{cases} \min\{t_2^k, \lambda_3^k\} & \text{if } t_2^k < \frac{2}{1+\beta} t_3^k, \\ 0 & \text{otherwise.} \end{cases}$$

For later use, we note that

$$(28) \qquad t_1^k - \lambda_1^k \geq 0 \quad \text{and} \quad t_2^k - \lambda_2^k \geq 0 \quad \text{for all} \quad k.$$

The number $\lambda_3^k$ is defined as follows. A number $\gamma_k$ is chosen such that

$$(29) \qquad \min\left\{ \chi_1, \frac{\delta(x^k, y^k, s^k)}{\delta(x^0, y^0, s^0)} \right\} \leq \gamma_k \leq \chi_1,$$

and $\lambda_3^k := \gamma_k t_3$. The possibility of a small choice of $\gamma_k$ for small $\delta(x^k, y^k, s^k)$ is to allow for superlinear convergence.

**Step 3)**

The accuracy $\varepsilon_x^k$, $\varepsilon_s^k$ for solving (24) is determined as follows. First, choose two anticipated values $\alpha_x^{\text{ant}} \leq \alpha_{\max}$ and $\alpha_s^{\text{ant}} \leq \alpha_{\max}$ for the actual step lengths $\alpha_x^k$ and $\alpha_s^k$ in Step 5). Then compute the corresponding anticipated values

$$(30) \qquad t_1^+ := \left(1 - \alpha_x^{\text{ant}}\right) t_1^k + \alpha_x^{\text{ant}} \lambda_1^k \quad \text{and} \quad t_2^+ := \left(1 - \alpha_s^{\text{ant}}\right) t_2^k + \alpha_s^{\text{ant}} \lambda_2^k.$$

The vector $\Delta x$ for solving $A(x + \Delta x) = b$ is then required to satisfy

$$(31) \qquad \left\| b + \lambda_1 \bar{b} - A(x^k + \Delta x) \right\| \leq \varepsilon_x^k := \frac{\nu_p \sigma}{2} \left\{ t_1^+ (t_1^k - \lambda_1^k) + \lambda_1^k (1 - t_1^k) \right\},$$

and likewise, $\Delta s, \Delta y$ are required to satisfy

$$(32) \qquad \left\| c + \lambda_2 \bar{c} - A^T(y^k + \Delta y) - (s^k + \Delta s) \right\| \leq \varepsilon_s^k := \frac{\nu_d}{2} \left\{ t_2^+ (t_2^k - \lambda_2^k) + \lambda_2^k (1 - t_2^k) \right\}.$$

In general, the values of $\alpha_x^{\text{ant}}$ and $\alpha_s^{\text{ant}}$ will depend on $k$. They may be set to $\alpha_x^{\text{ant}} = \alpha_s^{\text{ant}} = \alpha_{\max}$ or to $\alpha_x^{\text{ant}} = \alpha_s^{\text{ant}} = \max\{\frac{1}{2}, 1 - t_3^k\}$, for example. Next, in the specification of Step 4), we mention a scheme that determines possibly smaller values of $\alpha_x^{\text{ant}}$ and $\alpha_s^{\text{ant}}$ iteratively when solving for $\Delta x$ and $\Delta s$.

**Step 4)**

To determine the step lengths $\alpha_x^k$, $\alpha_s^k$ at iteration step $k$, we first compute $\bar{\alpha}^k$ as the maximum $\hat{\alpha} \le 1$ such that the point $(x^k + \alpha\Delta x, y^k + \alpha\Delta y, s^k + \alpha\Delta s)$ satisfies (13) and

$$\left(x^k + \alpha\Delta x\right)^T \left(s^k + \alpha\Delta s\right) \;\le\; (1 - \alpha(1-\chi_2))(x^k)^T s^k \tag{33}$$

for all $0 \le \alpha \le \hat{\alpha}$. As we will see below, the definition of $\varepsilon_x^k$ and $\varepsilon_s^k$ in (31) and (32) guarantees that (12) holds as well, and thus, $(x^k + \bar{\alpha}^k\Delta x, y^k + \bar{\alpha}^k\Delta y, s^k + \bar{\alpha}^k\Delta s) \in \mathcal{N}(\beta)$. Furthermore, we will show that $\bar{\alpha}^k$ is always positive and bounded away from zero. After $\bar{\alpha}^k$ has been determined, we choose

$$\alpha_x^k \le \alpha_x^{\text{ant}} \qquad \text{and} \qquad \alpha_s^k \le \alpha_s^{\text{ant}} \tag{34}$$

such that the conditions

$$\left(x^k + \alpha_x^k\Delta x\right)^T \left(s^k + \alpha_s^k\Delta s\right) \;\le\; (1 - \bar{\alpha}^k(1-\chi_3))(x^k)^T s^k \tag{35}$$

and $(x^k + \alpha_x^k\Delta x, y^k + \alpha_s^k\Delta y,\ s^k + \alpha_s^k\Delta s) \in \mathcal{N}(\beta)$ are satisfied. The rules (33) and (35) were proposed in [10], and there, a cheap way of computing $\bar{\alpha}^k$ is also discussed.

**Remark 1:** In [3], an adaptive choice of the step lengths $\alpha_x^k$, $\alpha_s^k$ was proposed using the following strategy that is based on two further parameters $\chi_4 \ge 1$ and $\chi_5 \in (0,1)$ with suggested default values $\chi_4 = 2$ and $\chi_5 = 1/2$. First, one determines

$$\bar{\alpha}_x^k := \max\{\, \alpha \le 1 \mid x^k + \alpha\Delta x \ge 0 \,\} \quad \text{and} \quad \bar{\alpha}_s^k := \max\{\, \alpha \le 1 \mid s^k + \alpha\Delta s \ge 0 \,\}, \tag{36}$$

and chooses $\rho_k$ such that

$$\chi_5 \;\le\; \rho_k \;\le\; \max\left\{ \min\{\, 1 - \gamma_k^{\chi_4},\ 1 - (1-\bar{\alpha}_x^k)^{\chi_4},\ 1 - (1-\bar{\alpha}_s^k)^{\chi_4} \,\},\ \chi_5 \right\}. \tag{37}$$

Then one sets

$$\alpha_x^k := \rho_k \bar{\alpha}_x^k \quad \text{and} \quad \alpha_s^k := \rho_k \bar{\alpha}_s^k.$$

The motivation for this strategy is that $\gamma_k$ being small implies a large reduction in complementarity, and $\bar{\alpha}_x^k$ and $\bar{\alpha}_s^k$ being close to 1 implies a large reduction of infeasibility by the full Newton step. If both is the case, a large $\rho_k$ seems profitable even though it may bring the iterate close to the boundary of the feasible set. Therefore, $\alpha_x^k$, $\alpha_s^k$ may be a practically efficient choice. When $\chi_3 < 1$ is close to 1, the constraint (35) on the step lengths is fairly weak, and the above choice of $\alpha_x^k$, $\alpha_s^k$ may satisfy (35) for most iterations $k$. If (35) is not satisfied or if the result does not lie in $\mathcal{N}(\beta)$, then step lengths that do satisfy these criteria may be computed as in [10].

The above scheme also allows to find values for $\alpha_x^{\text{ant}}$ and $\alpha_s^{\text{ant}}$ in Step 3) that possibly result in a less strict accuracy requirement (31), (32), (30) for the search directions $(\Delta x, \Delta y, \Delta s)$, compared to the choice $\alpha_x^{\text{ant}} = \alpha_s^{\text{ant}} = \alpha_{\max}$. Based on the above rules (36), and (37), we may determine values for $\alpha_x^{\text{ant}}$ and $\alpha_s^{\text{ant}}$ for a given approximation or guess $(\Delta x, \Delta y, \Delta s)$ for the solution of (24). With these values of $\alpha_x^{\text{ant}}$ and $\alpha_s^{\text{ant}}$, one then evaluates $\varepsilon_x^k$ and $\varepsilon_s^k$ in (31) and (32), and checks whether the current approximation $(\Delta x, \Delta y, \Delta s)$ satisfies the accuracy requirements (31) and (32). If yes, $(\Delta x, \Delta y, \Delta s)$ is accepted as a sufficiently accurate solution of (24). If not, one continues the iterative method used for the inexact solution of (24), until the stopping criteria (31) and (32) (with appropriately updated values of $\alpha_x^{\mathrm{ant}}$ and $\alpha_s^{\mathrm{ant}}$) are met.

**Remark 2:** The main computational effort in Algorithm 1 lies in the solution of the systems (24). Before solving these systems, we first employ the stable reduction described in Section 4.1 below to reduce the $3 \times 3$-block system to a smaller $2 \times 2$-block system. The smaller systems can be solved by either a direct method (see, e.g., [2, 22]), or by an iterative method, such as the one proposed in [3].

### 3.5 The Main Result

The main result, which is stated as Theorem 1 in Section 5 below, is that all iterates generated by Algorithm 1 remain in the neighborhood $\mathcal{N}(\beta)$. Furthermore, the iterates either converge to the set of optimal solutions if this set is nonempty, or it is detected that no optimal solution exists whose norm is less than a given bound.

The result that the iterates remain in $\mathcal{N}(\beta)$ is a consequence of the following two features of Algorithm 1. (i) The first two equality constraints in (11) are trivially maintained by a suitable update of $\tilde{b}$ and $\tilde{c}$, while the third equality constraint in (11) is trivially satisfied due the choice of $t_3^k$ in Algorithm 1. (ii) The norm constraints (12) on $\tilde{b}$ and $\tilde{c}$ are guaranteed by a proper choice of $\varepsilon_x$ and $\varepsilon_s$, and the inequality constraints (13) on $Xs$ and on the $t_i$'s are maintained by selecting suitable parameters $\lambda_1$, $\lambda_2$, $\lambda_3$ and step lengths $\alpha_x$, $\alpha_s$.

## 4. Inexact Search Directions

In this section, we discuss how the accuracy requirement in Step 3) of Algorithm 1 can be satisfied, and why we have chosen this type of accuracy requirement.

### 4.1 A Stable Reduction of the $3 \times 3$-Block System

To show how a search direction satisfying an accuracy requirement of the form (25) can be obtained in practice, we first review the stable reduction (introduced in [3]) of the $3 \times 3$-block system (24) to a linear system with the $2 \times 2$-block matrix

$$K := \begin{bmatrix} -SX^{-1} & A^T \\ A & 0 \end{bmatrix}. \tag{38}$$

The basis of the stable reduction is a partition[1] of the vectors $x$ and $s$ into two parts $x_1$, $x_2$ and $s_1$, $s_2$, respectively, such that $x_1 \geq s_1$ and $x_2 < s_2$. Moreover, for the ease of notation, we assume that $x_1$ and $s_1$ are the leading entries of $x$ and $s$, i.e.,

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{and} \quad s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}. \tag{39}$$

[1] We stress that the choice of this partition may be changed, for example, to further reduce the primal residual at the cost of the dual residual. In preliminary numerical experiments, it was our observation that a partitioning according to $\nu_d x_1 \geq \nu_p s_1$ and $\nu_d x_2 < \nu_p s_2$ might be slightly more efficient.

In this subsection, we do not refer to specific components of the vectors $x$ or $s$; instead $x_1$ and $s_2$ always denote vectors containing all "large" components of $x$ and $s$, respectively. We partition the constraint matrix $A$ conforming with (39), and write

$$A = [\, A_1 \quad A_2 \,]. \tag{40}$$

Similarly, (39) induces the partition

$$p = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{41}$$

of the residual vectors $p$ and $q$ defined in (23). As usual, we denote by $X_1$, $X_2$, $S_1$, and $S_2$ the diagonal matrices associated with the vectors $x_1$, $x_2$, $s_1$, and $s_2$, respectively.

Instead of the original system (24), we propose to solve the equivalent reduced $2 \times 2$-block system

$$\begin{bmatrix} -SX^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} u \\ \Delta y \end{bmatrix} = \begin{bmatrix} \tilde{r} \\ \tilde{p} \end{bmatrix}, \tag{42}$$

where

$$\tilde{r} := \begin{bmatrix} p_1 - X_1^{-1} q_1 \\ p_2 \end{bmatrix} \quad \text{and} \quad \tilde{p} := r - A_2 S_2^{-1} q_2. \tag{43}$$

Furthermore, the new variable $u$ is connected with $\Delta x_1$ and $\Delta s_2$ via the relation

$$u = \begin{bmatrix} \Delta x_1 \\ -X_2 S_2^{-1} \Delta s_2 \end{bmatrix}. \tag{44}$$

Using (39)–(41), (43), and (44), together with

$$\begin{aligned} \Delta x_2 &= S_2^{-1} \left( q_2 - X_2 \Delta s_2 \right), \\ \Delta s_1 &= X_1^{-1} \left( q_1 - S_1 \Delta x_1 \right), \end{aligned} \tag{45}$$

one readily verifies that the reduced $2 \times 2$-block system (42) is indeed equivalent to the original system (24).

We consider interior-point methods that solve the reduced system (42) (iteratively) to a low relative accuracy, and thus obtain only an approximate solution. Given such an approximate solution $u$, $\Delta y$ of (42), the quantity $\Delta x_1$ is computed by using the relation (44). Note that computing $\Delta s_2$ from (44) would involve the inverse of the "small" matrix $X_2$. This can be avoided by computing $\Delta s_2$ via

$$\Delta s_2 = \lambda_2 \bar{c}_2 + c_2 - A_2^T (y + \Delta y) - s_2,$$

and then obtaining $\Delta x_2$ and $\Delta s_1$ from (45). This guarantees that equation (26) is satisfied up to machine precision. In the above computations, only the inverses of the "large" matrices $X_1$ and $S_2$ are used, but not $X_2^{-1}$ or $S_1^{-1}$. Furthermore, we remark that the partition of $x$ into $x_1$ and $x_2$ has nothing to do with guessing active indices. The partition is merely done to improve the stability for solving the current linear system, it does not involve any additional computational cost, and there is no loss if the partition changes at each iteration.

## 4.2 Motivation of Inexactness Requirements

Next, we further motivate our requirement in Step 3) of Algorithm 1 on the accuracy of the computation of the search directions in (25). We implicitly assume that we could compute the search direction to arbitrarily high precision, but that we are not willing to invest more computational effort for more accuracy than necessary.

To guarantee convergence of the interior-point method even for inexact search directions, we need a careful analysis of the errors introduced by the inexactness of the search direction.

Assume for the moment that we set $\lambda_1 = \lambda_2 = 0$ in (24). If $(\Delta x, \Delta s, \Delta y)$ solve the linear systems (24) exactly, then, by the linearity of $r$, the primal residual $r^+$ at $x^+ = x + \alpha_x \Delta x$ satisfies

$$\|r^+\| = \|b - Ax^+\| = \|b - A(x + \alpha_x \Delta x)\| = \|(1 - \alpha_x)(b - Ax)\| = (1 - \alpha_x)\|r\|.$$

Thus, the norm of the primal residual $r$ is reduced by the factor $1 - \alpha_x$ in the case of exact solution of (24). If we allow for some tolerance $\varepsilon$ in the approximate solution of (24) by requiring that $\Delta x$ is merely computed to a certain relative accuracy in the residual,

$$\|b - Ax - A\Delta x\| \ \le \ \varepsilon(1 - \alpha_x)\|b - Ax\| \tag{46}$$

instead of $A\Delta x = b - Ax$. Then, by straightforward computations, we have

$$\|r^+_{inexact}\| = \|b - A(x + \alpha_x \Delta x)\| \le (1 + \varepsilon\alpha_x)(1 - \alpha_x)\|r\|.$$

The result $r^+_{inexact}$ obtained from inexact computations is almost as good as the result obtained from exact computations. Above inequality also shows that the size of $\alpha_x$ determines the reduction of $r$, and that it is important to define a search direction with large $\alpha_x$. Of course, analogous considerations as above for $\alpha_x$, $r$, and $r^+$ also hold for $\alpha_s$, $p$, and $p^+$.

In the equation $Xs = \mu e$ corresponding to $q$ in (19), the quantity $\mu > 0$ is used to keep the iterate away from the boundary, $x \ge 0$, $s \ge 0$, and to allow for long steps $\alpha_x$, $\alpha_x$ that do not leave the positive orthant. We assume that the stable reduction (45) above is used so that

$$x_i \Delta s_i + s_i \Delta x_i = q_i \equiv \lambda_3 \bar{\mu} - x_i s_i, \quad i = 1, 2, \ldots, n, \tag{47}$$

holds true.

Finally, we remark that criterion (46) is not satisfactory in this form for the following reason. An iterative approach for solving the linear systems requires that we allow for a low relative accuracy even if the initial residual happened to be zero, i.e., if $Ax^0 = b$, and criterion (46) does not allow any increase in the residual. We therefore do not use criterion (46) but a more appropriate modification of (46), which is discussed next.

## 4.3 Analysis of the Perturbed Feasible Sets

We consider the perturbed constraints,

$$Ax \ = \ b + t_1(\bar{b} + \tilde{b}), \tag{48}$$

$$A^T y + s \ = \ c + t_2(\bar{c} + \tilde{c}), \tag{49}$$

$$x,\ s \ \ge \ 0, \tag{50}$$

that are part of the relations defining $\mathcal{N}(\beta)$. In the following proposition, we give a condition on the perturbation vectors $\tilde{b}$ and $\tilde{c}$ that guarantees the feasibility of (48)–(50). We remark that in many practical examples the feasible sets of (1) or (2) have no interior points, and when $\bar{b}=0$ or $\bar{c}=0$, the system (48)–(50) may be infeasible even if $\|\tilde{b}\|$ or $\|\tilde{c}\|$ is very small.

The result in Proposition 1 is crucial for the convergence analysis in the next section. Furthermore, we remark that the lower bound (9) on the smallest singular value of $A$ is used only in the proof of this result.

**Proposition 1** *Assume that the problems* (1) *and* (2) *have an optimal solution. If* $x^0 \geq \nu_p e$ *and* $s^0 \geq \nu_d e$, *then the constraints* (48)–(50) *have a feasible solution for all perturbation vectors* $\tilde{b}$, $\tilde{c}$ *with*

$$\|\tilde{b}\| \leq \tfrac{1}{2}\nu_p\sigma, \quad \|\tilde{c}\| \leq \tfrac{1}{2}\nu_d, \tag{51}$$

*and for all* $0 \leq t_1,\, t_2 \leq 1$. *More precisely, there exists a feasible solution* $(\tilde{x},\tilde{y},\tilde{s})$ *of* (48)–(50) *with*

$$\tilde{x} \geq \tfrac{1}{2}\nu_p t_1 e \quad \textit{and} \quad \tilde{s} \geq \tfrac{1}{2}\nu_d t_2 e. \tag{52}$$

*Moreover, if* $t_1 = t_2 = 1$, *then* $\tilde{x}$, $\tilde{s}$ *can be chosen to satisfy*

$$\tilde{x} \leq \tfrac{3}{2}x^0 \quad \textit{and} \quad \tilde{s} \leq \tfrac{3}{2}s^0. \tag{53}$$

**Proof:** Let the point $(\bar{x},\bar{y},\bar{s})$ be a feasible solution of (1) and (2), i.e., it satisfies the constraints (3), (4), and (6). We need to construct a feasible solution $(\tilde{x},\tilde{y},\tilde{s})$ of (48)–(50). To this end, we first set $h := A^+\tilde{b}$, where $A^+$ denotes the pseudoinverse of $A$ (see, e.g., [20]). By construction, the vector $h$ satisfies $Ah=\tilde{b}$. Moreover, since $\|A^+\| = 1/\sigma_{\min}(A) \leq 1/\sigma$ and by the first inequality in (51), we have

$$\|h\|_\infty \leq \|h\| \leq \|A^+\| \cdot \|\tilde{b}\| \leq \tfrac{1}{2}\nu_p. \tag{54}$$

We now define $(\tilde{x},\tilde{y},\tilde{s})$ as follows:

$$\tilde{x} := (1-t_1)\bar{x} + t_1 x^0 + t_1 h, \tag{55}$$

$$\tilde{y} := (1-t_2)\bar{y} + t_2 y^0,$$

$$\tilde{s} := (1-t_2)\bar{s} + t_2 s^0 + t_2\tilde{c}. \tag{56}$$

Using (3), (4), the definitions of $\bar{b}$ and $\bar{c}$ in (10), and the fact that $Ah=\tilde{b}$, one readily verifies that the point $(\tilde{x},\tilde{y},\tilde{s})$ satisfies (48) and (49).

It remains to show (52) and (53). Note that (54) implies $h \geq -\nu_p e/2$. Together with $\bar{x}\geq 0$, $1-t_1 \geq 0$, $t_1 \geq 0$, $x^0 \geq \nu_p e$, it follows from (55) that

$$\tilde{x} \geq t_1 x^0 + t_1 h \geq \nu_p t_1 e - \tfrac{1}{2}\nu_p t_1 e = \tfrac{1}{2}\nu_p t_1 e,$$

which is just the first inequality in (52). By the second inequality in (51), we have $\|\tilde{c}\|_\infty \leq \|\tilde{c}\| \leq \frac{1}{2}\nu_d$, and thus $\tilde{c} \geq -\frac{1}{2}\nu_d e$. Together with (56), this implies the second inequality in (52). Finally, (53) follows from (55) (with $t_1 = 1$), (56) ($t_2 = 1$), and

$$h \leq \tfrac{1}{2}\nu_p e \leq \tfrac{1}{2}x^0, \quad \tilde{c} \leq \tfrac{1}{2}\nu_d e \leq \tfrac{1}{2}s^0.$$

□

Here, we would like to stress our observation of Section 2, that theoretically, prior to applying the interior-point method, one can transform problem (1) to an equivalent problem such that $\sigma = 1$, and practically, when this transformation is not appropriate, we may still anticipate that $1/\sigma$ is moderate even if (1) happened to be poorly conditioned. In particular, degeneracy or near-degeneracy of (1) do not depend on $\sigma$.

### 4.4 Required Accuracy

In the course of Algorithm 1, the vectors $\bar{b} = Ax^0 - b$ and $\bar{c} = A^T y^0 + s^0 - c$ are reduced to $t_1^k \bar{b}$ and $t_2^k \bar{c}$, respectively. Furthermore, by the inexactness of the search direction, "new" perturbation vectors $\tilde{b}^k$ and $\tilde{c}^k$ are introduced. More precisely, we have

$$Ax^k = b + t_1^k(\bar{b} + \tilde{b}^k) \quad \text{and} \quad A^T y^k + s^k = c + t_2^k(\bar{c} + \tilde{c}^k),$$

where $\tilde{b}^k$ and $\tilde{c}^k$ are defined as follows:

$$\tilde{b}^k := \frac{1}{t_1^k}\left(Ax^k - b\right) - \bar{b} \quad \text{and} \quad \tilde{c}^k := \frac{1}{t_2^k}\left(A^T y^k + s^k - c\right) - \bar{c}. \tag{57}$$

We now show that the vectors (57) remain bounded, provided that in Step 3) of Algorithm 1 the search directions are computed to the accuracies $\varepsilon_x = \varepsilon_x^k$ and $\varepsilon_s = \varepsilon_s^k$ defined in (31) and (32), respectively.

**Proposition 2** *For all iterates computed by Algorithm* 1, *the associated perturbations vectors* $\tilde{b}^k$ *and* $\tilde{c}^k$ *defined in* (57) *satisfy*

$$\|\tilde{b}^k\| \le \tfrac{1}{2}\sigma(1 - t_1^k)\nu_p \quad \textit{and} \quad \|\tilde{c}^k\| \le \tfrac{1}{2}(1 - t_2^k)\nu_d. \tag{58}$$

**Proof:** The proof is by induction on $k$. For $k = 0$, we have $t_1^0 = t_2^0 = 1$, $Ax^0 = b + \bar{b}$, and $A^T y^0 + s^0 = c + \bar{c}$. Hence, $\tilde{b}^0 = 0$ and $\tilde{c}^0 = 0$, and thus (58) is trivially satisfied for $k = 0$.

Now assume that (58) holds for some $k \ge 0$. We show that then (58) is also satisfied for $k + 1$. We will only verify the first relation in (58); the second relation is treated analogously. Recall that

$$x^{k+1} = x + \alpha\Delta x \quad \text{and} \quad t_1^{k+1} = (1 - \alpha)t_1 + \alpha\lambda_1, \tag{59}$$

where we omitted indices in $x = x^k$, $\alpha = \alpha_x^k$, $t_1 = t_1^k$, and $\lambda_1 = \lambda_1^k$. Using (57) (with $k$ replaced by $k + 1$) and (59), we obtain

$$\begin{aligned} t_1^{k+1}\tilde{b}^{k+1} &= Ax^{k+1} - b - t_1^{k+1}\bar{b} \\ &= A(x + \alpha\Delta x) - b - ((1 - \alpha)t_1 + \alpha\lambda_1)\bar{b} \\ &= \alpha(A(x + \Delta x) - b - \lambda_1\bar{b}) + (1 - \alpha)(Ax - b - t_1\bar{b}). \end{aligned} \tag{60}$$

By the first relations in (57) and (58), respectively, we have

$$\|Ax - b - t_1\bar{b}\| = t_1\|\tilde{b}^k\| \le \tfrac{1}{2}\sigma(1 - t_1)t_1\nu_p. \tag{61}$$

Taking norms in (60) and using (31) and (61), it follows that

$$t_1^{k+1}\|\tilde{b}^{k+1}\| \le \alpha\varepsilon_x + \tfrac{1}{2}(1 - \alpha)\sigma(1 - t_1)t_1\nu_p. \tag{62}$$

Next, recall from (28) and (34) that $\lambda_1 \leq t_1$ and $\alpha \leq \alpha_x^{\text{ant}}$. Thus, together with the definition of $t_1^+$ in (30), it follows that

$$t_1^+ = t_1 + \alpha_x^{\text{ant}}(\lambda_1 - t_1) \leq t_1 + \alpha(\lambda_1 - t_1) = t_1^{k+1}.$$

Using the definition of $\varepsilon_x$ in (31), we further obtain

$$\varepsilon_x \;\leq\; \frac{\nu_p \sigma}{2}\left(t_1^{k+1}(t_1 - \lambda_1) + \lambda_1(1 - t_1)\right). \tag{63}$$

Inserting (63) into (62) and using $t_1^{k+1} = (1-\alpha)t_1 + \alpha\lambda_1$, we get

$$\begin{aligned}
\|\tilde{b}^{k+1}\| &\leq \frac{\nu_p \sigma}{2} \frac{1}{t_1^{k+1}} \left(\alpha t_1^{k+1}(t_1 - \lambda_1) + (1 - t_1)\left((1-\alpha)t_1 + \alpha\lambda_1\right)\right) \\
&= \frac{\nu_p \sigma}{2}\left(\alpha(t_1 - \lambda_1) + (1 - t_1)\right) = \tfrac{1}{2}\sigma(1 - t_1^{k+1})\nu_p.
\end{aligned}$$

This is just the desired first relation in (58) (with $k$ replaced by $k+1$). $\square$

**Remark 3:** When $\lambda_1^k = 0$, the upper bound (63) for $\varepsilon_x^k$ reduces to

$$\varepsilon_x^k \;\leq\; \tfrac{1}{2}\sigma\nu_p\left(1 - \alpha_x^k\right)(t_1^k)^2.$$

To relate this to some statement about relative accuracy, we point out that the norm of the right-hand side $r$ in (23) is bounded by $2t_1^k\|\bar{b}\|$ (if $\sigma \leq 1$). Note that $\nu_p$ and $\|\bar{b}\|$ are known and $\|\bar{b}\|/\nu_p$ can easily be bounded for a suitable choice of $x^0$. If $\|\bar{b}\|$ is zero or very small compared to $\nu_p$, the concept (46) of solving for a certain relative accuracy is more restrictive than (31), and if $\|\bar{b}\|$ is large, say $\|\bar{b}\| \approx \nu_p$, then $\|r\| \approx t_1^k\nu_p$ and condition (31) roughly means that

$$\varepsilon_x^k \;\approx\; \tfrac{1}{2}\sigma\left(1 - \alpha_x^k\right)t_1^k\|r\|.$$

This is in the spirit of (46), except for the additional factor $t_1^k \leq 1$. (Likewise for $p$.) The conditions (31) and (32) implying that for small $t_1^k$, $t_2^k$ and very small $\lambda_1^k$, $\lambda_2^k$ a high relative accuracy is required, coincide with the desire of a higher relative accuracy in the final iterations of the interior-point method to obtain superlinear convergence.

## 5. Convergence Analysis

In this section, we present a proof of convergence of the inexact interior-point Algorithm 1. Our main result is the following theorem.

**Theorem 1** *Let* $\left\{(x^k, y^k, s^k)\right\}_{k\geq 0}$ *denote the sequence of iterates generated by Algorithm* 1. *Then:*

a) $(x^k, y^k, s^k) \in \mathcal{N}(\beta)$ *for all* $k \geq 0$.

b) *If the sequence* $(x^k, s^k)$ *is unbounded, then the linear program* (1) *has no solution. More precisely, if* $\nu_d\|x^k\|_1 + \nu_p\|s^k\|_1 > M_1$ *for some* $k \geq 0$ *and* $M_1 \geq ((9/2) + 2\beta)\,(x^0)^T s^0$, *then there exists no optimal solution* $(x^{opt}, y^{opt}, s^{opt})$ *of the primal and dual linear programs* (1) *and* (2) *with*

$$\|x^{opt}\|_1 + \|s^{opt}\|_1 \;\leq\; \frac{2M_1 - (9 + 4\beta)\,(x^0)^T s^0}{6\beta^2 \max\left\{\|x^0\|_\infty,\; \|s^0\|_\infty\right\}}.$$

c) *If the iterates $(x^k, s^k)$ remain bounded, then the stopping criterion in Step* 1) *of Algorithm* 1 *will be satisfied after a finite number of iterations.*

The proof of Theorem 1 makes use of the following three technical propositions.

**Proposition 3** *If the primal and dual linear programs* (1) *and* (2) *have an optimal solution, then the $x$- and $s$-components of the neighborhood $\mathcal{N}(\beta)$ are bounded, i.e., there exists a constant $M_1$ such that*

$$\nu_d\|x\|_1 + \nu_p\|s\|_1 \le M_1 \quad \text{for all} \quad (x, y, s) \in \mathcal{N}(\beta). \tag{64}$$

*The constant $M_1$ in* (64) *can be chosen as*

$$M_1 = (\tfrac{9}{2} + 2\beta)(x^0)^T s^0 + 3\beta^2 \left(\|x^{opt}\|_1 + \|s^{opt}\|_1\right) \max\{\|x^0\|_\infty, \|s^0\|_\infty\}, \tag{65}$$

*where $(x^{opt}, y^{opt}, s^{opt})$ is any optimal solution of* (1) *and* (2).

**Proof:** This proposition was used in slightly different form in [15, 16]. Since the form stated here is more general than in [15, 16], we give a complete proof.

Let $(x, y, s) \in \mathcal{N}(\beta)$, with associated values $t_1$ and $t_2$, and $\tilde{b}$ and $\tilde{c}$, be arbitrary, but fixed. By Proposition 1, there exists a point $(\tilde{x}, \tilde{y}, \tilde{s})$ such that

$$A\tilde{x} = b + \bar{b} + \tilde{b}, \quad A^T\tilde{y} + \tilde{s} = c + \bar{c} + \tilde{c}, \quad \tfrac{1}{2}\nu_p e \le \tilde{x} \le \tfrac{3}{2}x^0, \quad \tfrac{1}{2}\nu_d e \le \tilde{s} \le \tfrac{3}{2}s^0. \tag{66}$$

Moreover, let $(x^{\rm opt}, y^{\rm opt}, s^{\rm opt})$ be an optimal solution of (1) and (2), and set

$$x(t) = t\tilde{x} + (1-t)x^{\rm opt}, \quad y(t) = t\tilde{y} + (1-t)y^{\rm opt}, \quad \text{and} \quad s(t) = t\tilde{s} + (1-t)s^{\rm opt}. \tag{67}$$

Recall from (1) and (2) that $Ax^{\rm opt} = b$ and $A^T y^{\rm opt} + s^{\rm opt} = c$. Using (67) and the first two relations in (11) and (66), respectively, we get $A(x(t_1) - x) = 0$ and $A^T(y(t_2) - y) + s(t_2) - s = 0$. These two equations imply that

$$(x(t_1) - x)^T(s(t_2) - s) = -(x(t_1) - x)^T A^T (y(t_2) - y) = -(A(x(t_1) - x))(y(t_2) - y) = 0.$$

We thus have

$$\begin{aligned}(x(t_1))^T s(t_2) + x^T s &= x^T s(t_2) + s^T x(t_1)\\ &= t_2 x^T\tilde{s} + (1-t_2)x^T s^{\rm opt} + t_1 s^T\tilde{x} + (1-t_1)s^T x^{\rm opt}\\ &\ge t_2 x^T\tilde{s} + t_1 s^T\tilde{x}\\ &\ge \tfrac{1}{2}\left(t_2\nu_d\|x\|_1 + t_1\nu_p\|s\|_1\right) \ge \tfrac{1}{2}\min\{t_1, t_2\}\left(\nu_d\|x\|_1 + \nu_p\|s\|_1\right).\end{aligned} \tag{68}$$

Here, the second-to-last inequality follows from the lower bounds for $\tilde{x}$ and $\tilde{s}$ in (66). Using (67), (11), (13), the upper bounds for $\tilde{x}$ and $\tilde{s}$ in (66), the fact that, by (5), $(x^{\rm opt})^T s^{\rm opt} = 0$, and the fact that $(x^0)^T s^0 = n\bar{\mu}$, we get

$$\begin{aligned}(x(t_1))^T s(t_2) + x^T s &= t_1 t_2\tilde{x}^T\tilde{s} + (1-t_1)t_2(x^{\rm opt})^T\tilde{s} + t_1(1-t_2)(s^{\rm opt})^T\tilde{x} + x^T s\\ &\le \min\{t_1, t_2\}\left(\tilde{x}^T\tilde{s} + \beta^2\left((x^{\rm opt})^T\tilde{s} + (s^{\rm opt})^T\tilde{x}\right) + n\beta\bar{\mu}\right)\\ &\le \min\{t_1, t_2\}\left(\left(\tfrac{9}{4} + \beta\right)(x^0)^T s^0 + \tfrac{3}{2}\beta^2\left((x^{\rm opt})^T s^0 + (s^{\rm opt})^T x^0\right)\right).\end{aligned} \tag{69}$$

By combining (68) and (69) and dividing by $\frac{1}{2}\min\{t_1, t_2\}$, we obtain

$$\nu_d\|x\|_1 + \nu_p\|s\|_1 \le \left(\tfrac{9}{2} + 2\beta\right)(x^0)^T s^0 + 3\beta^2\left((x^{\mathrm{opt}})^T s^0 + (s^{\mathrm{opt}})^T x^0\right). \tag{70}$$

This concludes the proof of (64).

Finally, note that

$$(x^{\mathrm{opt}})^T s^0 + (s^{\mathrm{opt}})^T x^0 \le (\|x^{\mathrm{opt}}\|_1 + \|s^{\mathrm{opt}}\|_1)\max\{\|x^0\|_\infty, \|s^0\|_\infty\}.$$

Inserting this inequality into the upper bound in (70), we obtain the constant $M_1$ given by (65). $\square$

**Proposition 4** *For all $\beta > 1$, $M_2 < \infty$, and $t \in (0,1]$, there is a $\xi > 0$ such that*

$$\xi e \le x, s \le \frac{1}{\xi}e \quad \textit{for all} \quad (x,y,s) \in \mathcal{N}(\beta) \quad \textit{with} \quad \frac{x^T s}{n} \ge t \quad \textit{and} \quad \nu_d\|x\| + \nu_p\|s\| < M_2.$$

**Proof:** The proof is straightforward by using the definition of $\mathcal{N}(\beta)$. $\square$

**Proposition 5** *Let $t \in (0,1]$, $M_2 < \infty$, and $\beta > 1$ be given. Then there is an $M_3 < \infty$ such that for any point $(x^k, y^k, s^k)$ generated by Algorithm 1 with $(x^k, y^k, s^k) \in \mathcal{N}(\beta)$, $(x^k)^T s^k/n \ge t$, and $\nu_d\|x^k\| + \nu_p\|s^k\| < M_2$, the bound $\|\Delta x\|, \|\Delta s\| \le M_3$ holds. Here, $\Delta x$ and $\Delta s$ are determined from (24) and (23) for a given accuracy specified by (31), (32).*

**Proof:** Note that $\|A^T y^k + s^k - c\|$ is bounded for all iterates. From boundedness of $\|s^k\|$ thus follows boundedness of $\|y^k\|$. Note that $\varepsilon_x$ and $\varepsilon_s$ are bounded as well, and $\Delta x$, $\Delta s$ are the exact solution of a system (24) with $r$ and $q$ perturbed by a (bounded) quantity of size $\varepsilon_x$ and $\varepsilon_s$. Thus, we may assume that the right-hand sides (23) are bounded, and from Proposition 4 and our full-rank assumption on $A$ follows that the inverses of the matrices in (24) are uniformly bounded. (The inverse exists for any $x > 0$ and $s > 0$, and is a continuous function of $x$ and $s$!) This concludes the proof. $\square$

**Proof of Theorem 1:** The proof is based on ideas in Kojima et. al. [10].

First, we prove part a). By Proposition 2, the vectors $\tilde{b}$ and $\tilde{c}$ associated to $x^k$ and $s^k$ satisfy (58), and hence (12). Relation (13) is guaranteed by the update in Step 4) of Algorithm 1, and hence the iterates remain in $\mathcal{N}(\beta)$.

Part b) is an immediate consequence of Proposition 3.

Finally, we prove part c). From now on, it is assumed that the sequence $(x^k, s^k)$ is bounded, say $\nu_d\|x^k\| + \nu_p\|s^k\| \le M_2$ for some $M_2 < \infty$. We show that then Algorithm 1 finds an approximate optimal solution after a finite number of iterations.

Suppose that $t_3^k \to 0$, and $(x^k, y^k, s^k) \in \mathcal{N}(\beta)$. Then, $t_1^k \to 0$ and $t_2^k \to 0$ by (13), and by the bounds (12), also $\|Ax^k - b\| \to 0$ and $\|A^T y^k + s^k - c\| \to 0$. By (11), we also have $(x^k)^T s^k = t_3^k n\bar{\mu} \to 0$. Therefore, in view of (27), the stopping criterion of Algorithm 1 is satisfied after a finite number of steps.

Conversely, as long as Algorithm 1 is running, we have $\delta(x^k, y^k, s^k) > \epsilon$. Hence there is some $\xi_2 > 0$ such that $t_3^k = (x^k)^T s^k/n > \xi_2$ for all $k$ (for which Algorithm 1 does not halt). By Proposition 4, there is some $\xi_3 > 0$ such that $x^k \ge \xi_3 e$ and $s^k \ge \xi_3 e$.

Also, by (29), $\gamma_k$ is bounded away from zero, without loss of generality, $\gamma_k \geq \xi_4 > 0$ for all $k$. Further, Proposition 5 is applicable, and there is some $M_3 < \infty$ such that $\|\Delta x^k\|$, $\|\Delta s^k\| \leq M_3$ for all $k$. (Without loss of generality, we assume $M_3 \geq 1$.) Finally, in view of (29), we may assume that $\gamma^k (x^k)^T s^k / n \geq \hat{\epsilon}$ for some $\hat{\epsilon} > 0$ independent of $k$.

To conclude the proof of part c), we show that $\bar{\alpha}^k$ is bounded away from zero, i.e., that there is some $\xi_5 > 0$ such that

$$\bar{\alpha}^k \geq \xi_5. \tag{71}$$

Recall that $\bar{\alpha}^k$ is defined via conditions (33) and (13). Note that both, $\alpha_x^k$ and $\alpha_s^k$ can be chosen equal to $\bar{\alpha}^k$. Thus (71) guarantees that a step length as required by (33) can indeed be found, and (35) ensures that $t_3^k = (x^k)^T s^k / (n\bar{\mu})$ converges to zero. Hence, showing (71) indeed completes the proof of convergence.

We consider some iteration $k$, and show the existence of a value $\xi_5 > 0$, independent of $k$, such that the bounds in (13) and (33) are satisfied for some $\bar{\alpha}^k \geq \xi_5$.

**Bound on $Xs/(t_3\bar{\mu})$**

For completeness, we write the proof from [10] in our notation.

Let $x(\alpha) = x^k + \alpha \Delta x$, and $s(\alpha) = s^k + \alpha \Delta s$. Clearly, since $x_i^k$, $s_i^k \geq \xi_3$ and $\|\Delta x\|$, $\|\Delta s\| \leq M_3$ (independently of $k$), there is some $\alpha_1 > 0$ such that $x(\alpha)$, $s(\alpha) > 0$ for $0 \leq \alpha \leq \alpha_1$. We consider

$$x_i(\alpha) s_i(\alpha) - \frac{x(\alpha)^T s(\alpha)}{n\beta}. \tag{72}$$

This number is positive if, and only if, the first relation of (13) is satisfied. From (47) it follows that

$$x_i(\alpha) s_i(\alpha) = (1-\alpha) x_i^k s_i^k + \alpha \lambda_3 \bar{\mu} + \alpha^2 \Delta x_i \Delta s_i \geq (1-\alpha) \frac{(x^k)^T s^k}{n\beta} + \alpha \lambda_3 \bar{\mu} + \alpha^2 \Delta x_i \Delta s_i,$$

with

$$\lambda_3 = \gamma_k t_3^k = \gamma_k \frac{(x^k)^T s^k}{n\bar{\mu}}.$$

We also have

$$\frac{x(\alpha)^T s(\alpha)}{n\beta} = (1-\alpha) \frac{(x^k)^T s^k}{n\beta} + \alpha \frac{\lambda_3 \bar{\mu}}{\beta} + \alpha^2 \frac{(\Delta x)^T \Delta s}{n\beta}.$$

Thus we can bound (72) from below by

$$\left(1 - \frac{1}{\beta}\right) \gamma_k \frac{(x^k)^T s^k}{n} \alpha + \left(\Delta x_i \Delta s_i - \frac{(\Delta x)^T \Delta s}{n\beta}\right) \alpha^2.$$

Note that $|\Delta x_i \Delta s_i| \leq M_3^2$ and $|(\Delta x)^T \Delta s / n| \leq M_3^2$. It follows that for all

$$0 \ \leq \ \alpha \ \leq \ \alpha_2 := \frac{\hat{\epsilon}}{2M_3^2} \left(1 - \frac{1}{\beta}\right)$$

the relation (72) holds provided that it holds for $\alpha = 0$.

**Bounds linking $t_1$ and $t_3$**

First we show that there is some $\alpha_3 > 0$ (independent of $k$) such that $t_1(\alpha)\beta - t_3(\alpha) \geq 0$ for $0 < \alpha \leq \alpha_3$ assuming that $t_1\beta - t_3 \geq 0$ for $t_1 = t_1(0)$ and $t_3 = t_3(0)$. Here, $t_1(\alpha)$ and $t_3(\alpha)$ are defined in the obvious way:

$$t_1(\alpha) = (1-\alpha)t_1^k + \alpha\lambda_1^k \quad \text{and} \quad t_3(\alpha) = \frac{x(\alpha)^T s(\alpha)}{n\bar{\mu}}.$$

Note that

$$t_1(\alpha)\beta - t_3(\alpha) = (\beta t_1 - t_3)(1-\alpha) + \alpha(\lambda_1\beta - \lambda_3) - \alpha^2\frac{(\Delta x)^T\Delta s}{n\bar{\mu}}.$$

Recall that, in Algorithm 1, either $t_1 \geq 2t_3/(1+\beta)$ and $\lambda_1 = 0$, or $t_3/\beta \leq t_1 < 2t_3/(1+\beta)$ and $\lambda_1 = \min\{t_1, \lambda_3\}$. We refer to the first instance as "case i)" and to the latter as "case ii)". In case i), it follows that

$$\beta t_1 - t_3 \;\geq\; \left(\frac{2\beta}{1+\beta} - 1\right) t_3 \;\geq\; \frac{\beta-1}{\beta+1}\xi_2,$$

and $\lambda_1\beta - \lambda_3 \geq -1$. Straightforward calculations yield that $t_1(\alpha)\beta - t_3(\alpha) \geq 0$ for

$$0 \leq \alpha \leq \frac{1}{2\eta_2}\left(\sqrt{(\eta_1+1)^2 + 4\eta_1\eta_2} - (\eta_1+1)\right), \quad \text{with} \quad \eta_1 = \frac{\beta-1}{\beta+1}\xi_2, \quad \eta_2 = \left|\frac{(\Delta x)^T\Delta s}{n\bar{\mu}}\right| \leq \frac{M_3^2}{\bar{\mu}}.$$

In case ii), it follows from $\beta t_1 \geq t_3 > \xi_2 > 0$ that

$$\lambda_1\beta - \lambda_3 \;\geq\; \min\{\,\beta t_1 - \chi_1 t_3,\; (\beta-1)\lambda_3\,\} \;\geq\; \xi_6$$

for some $\xi_6 > 0$ independent of $k$. Since $\beta t_1 - t_3 \geq 0$, we further conclude that $t_1(\alpha)\beta - t_3(\alpha) \geq 0$ for $0 \leq \alpha \leq \xi_6\bar{\mu}/M_3^2$. The minimum of these bounds yields $\alpha_3$.

Next we show that $t_3(\alpha)\beta - t_1(\alpha) \geq 0$ for $0 \leq \alpha \leq \alpha_4$ and some $\alpha_4 > 0$. This expression is written as

$$(\beta t_3 - t_1)(1-\alpha) + \alpha(\lambda_3\beta - \lambda_1) + \alpha^2\beta\frac{(\Delta x)^T\Delta s}{n\bar{\mu}}.$$

In case i), the quantity $\lambda_3\beta - \lambda_1$ is bounded away from zero, and in case ii), $\beta t_3 - t_1$ is bounded away from zero, so that the desired result follows again in a straightforward manner.

Finally, the bounds relating $t_2$ and $t_3$ are analogous to the ones between $t_1$ and $t_3$.

We close by noting that the rules (33), (35) are identical to [10], so that the analysis there fully applies to our algorithm, and guarantees, that the step lengths satisfying (33), (35) are bounded away from zero as long as the algorithm does not stop. □

## 6. Numerical Examples

Theorem 1 shows that Algorithm 1 is convergent, even though inexact search directions are used. However, no statement is made about the number of iterations compared to the case of exact search directions. In this section, we report a few numerical examples to illustrate the effects of the choice of $\lambda_1$ and $\lambda_2$ in Step 2) of Algorithm 1, and to demonstrate that the number of interior-point iterations does not increase significantly when inexact search directions are used.

We have implemented Algorithm 1 in MATLAB. The linear systems (42) defining the search directions are solved by means of an iterative method, the symmetric QMR algorithm with preconditioning. We remark that the linear systems (42) are symmetric indefinite, and that symmetric QMR is an iterative procedure tailored to the solution of such systems; see [3, 6]. As preconditioner $M$, we use an incomplete $L\Delta L^T$-factorization of the coefficient matrix $K$ defined in (38). More precisely, $L$ is a lower triangular matrix, $\Delta$ is a block-diagonal matrix with $1 \times 1$ and $2 \times 2$ diagonal blocks, and $P$ is a permutation matrix such that $L\Delta L^T$ approximates $PKP^T$. We allow at most 50 percent fill-in in the triangular matrix $L$, compared to the number of nonzeros in the original matrix $K$. The remaining elements are discarded, so that $L\Delta L^T$ is only an approximate factorization of $PKP^T$. Our implementation is rather slow since MATLAB does not efficiently handle the loops defining the preconditioner. We are therefore limited to present just a few small examples. A more efficient Fortran implementation of this method is currently in preparation [4]. We stress however, that in spite of the long MATLAB running times our program has the following features:

- It does not suffer from uncontrolled fill-in when forming the factorization;
- It is not limited to only special classes of linear programs.

Our preliminary numerical examples were done with problems without bounds and ranges from the NETLIB test set [7]. We solved the problems *afiro*, *adlittle*, *agg*, *blend*, *israel*, *lotfi*, *sc105*, *sc205*, *sc50a*, *sc50b*, *scagr7*, *share1b*, *share2b*, and *stocfor1*. All problems were solved by Algorithm 1 using inexact search directions satisfying (31), (32). The tolerance for the stopping criterion was set to $\epsilon = 10^{-8}$. When replacing the inexact search direction by the exact search direction, the number of iterations remained the same except for two problems where it increased by one. Furthermore, Algorithm 1 with (31), (32) showed a fast local convergence, reducing the error measure $\delta$ defined in (27) by a factor of $10^{-1}$ up to $10^{-6}$ in the final iteration.

For the above examples, other state-of-the-art implementations are up to three times faster than our implementation measured in terms of number of iterations. This is because all state-of-the-art implementations use longer step lengths along the search direction and require one additional back solve at each iteration to compute the Mehrotra corrector.

In Table 1, we list the results obtained for problem *israel* for the following parameter settings in Algorithm 1.

- Column 1) reports the results obtained with tolerances (31), (32).
- Column 2) reports the results for the same algorithm using exact search directions.
- Column 3) reports the results for obtained with tolerances (31), (32) and $\lambda_1 = \lambda_2 \equiv 0$.
- Column 4) reports the results for obtained using exact search directions and $\lambda_1 = \lambda_2 \equiv 0$.
- Column 5) reports the results when $\lambda_1 = \lambda_2 \equiv \lambda_3$, and inexact search directions with tolerances (31), (32) are used.
- In the rows we report:
    1) The final accuracy;
    2) The overall number of QMR-iterations used;
    3) The number of interior-point iterations.

Table 1: Results for *israel*

| | Version 1) | Version 2) | Version 3) | Version 4) | Version 5) |
|---|---|---|---|---|---|
| $\delta$ | 7.9E-9 | 1.7E-9 | 2.3E-9 | 1.6E-9 | 3.6E-9 |
| # qmr | 493 | – | 604 | – | 622 |
| # ipm | 47 | 47 | 51 | 50 | 54 |

We observe the following. The choice of $\lambda_1$ and $\lambda_2$ in Step 2) of Algorithm 1 slightly reduces the number of iterations compared to the choice $\lambda_1 = \lambda_2 \equiv 0$. This is true even when exact search directions are used. By Propositions 1 and 2, the effect of the perturbations $\bar{b}$ and $\bar{c}$ is that the interior of the perturbed set at some iteration $k$ contains a ball of radius at least $\frac{1}{2}\nu_p t_1^k$ in primal space and $\frac{1}{2}\nu_d t_2^k$ in dual space, while the interior of the unperturbed set may be empty. Thus, the Newton step to find the center of the perturbed set may be longer than in the case when $\bar{b} = 0$ and $\bar{c} = 0$. A similar behavior as above was observed for problem *agg*, where the algorithm took 44 iterations for the case $\lambda_1 = \lambda_2 \equiv 0$ and 43 iterations—and a slightly smaller value $\delta$—when using the choice of $\lambda_1$ and $\lambda_2$ from Step 2) of Algorithm 1.

For all other problems the algorithm converged slightly faster when $\lambda_1 = \lambda_2 \equiv 0$. This does not contradict our theory; in fact, if the interior of (1) and (2) is sufficiently large, there is no benefit of choosing $\lambda_1 > 0$ or $\lambda_2 > 0$.

## 7. Concluding Remarks

In this paper, we proved global convergence of the infeasible-interior-point method stated as Algorithm 1. This method allows that the linear systems at each iteration are solved only to moderate accuracy in the residual. Here, the accuracy requirement depends on an estimate $\sigma$ for the smallest singular value $\sigma_{\min}(A)$ of the constraint matrix $A$. The numbers $\sigma$ or $\sigma_{\min}(A)$ are not used elsewhere in the algorithm.

Algorithm 1 is very similar to the algorithm in the paper [10], which, however does not provide an analysis for the case where the search directions are not computed exactly. For very large linear programs, direct methods for solving the linear systems are often prohibitively expensive, and then iterative methods are often the only viable alternative. The iterative solutions, of course, contain a non-negligible error. In [3], an efficient iterative method, namely a variant of the QMR algorithm [6] tailored to indefinite symmetric systems, is discussed along with different strategies for preconditioning the linear systems arising in Algorithm 1. The assumptions made for Algorithm 1 originated from and were motivated by the method in [3].

We chose the method in [10] as a reference, since it can be adapted to inexact computations without substantial loss in its performance, and since it is known to be efficient also for numerical implementations; see, e.g., [12]. Other efficient interior-point methods, for example the methods based on a self-dual formulation of the linear program, do not allow such a simple extension to inexact computations. Apart from the specifications of the inexactness, the only modification of the algorithm in [10] that was made for Algorithm 1 is a slightly different choice of the parameters $\lambda_1$ and $\lambda_2$. These parameters are set to zero in [10]. They are also set to zero in Step 2) of Algorithm 1, except for the case that the corresponding residual $r$ or $p$ is already very small. In the latter case we do not expect much loss by not further reducing the residual at the current iteration.

In [17], another inexact infeasible-interior-point method is analyzed. The main difference lies in the choice of the parameters $\lambda_1$ and $\lambda_2$. The method in [17] defines positive values for $\lambda_1$ and $\lambda_2$ via certain conditions that prohibit $\lambda_1$ and $\lambda_2$ from being close to zero. Moreover, the algorithm in [17] uses the value $\sigma$, or more precisely a certain norm that can be bounded in terms of $\sigma$, not only for the accuracy of the search direction, but also for determining a certain parameter $\theta$ and the step length. Therefore, the method in [17] does not lend itself to an actual practical numerical implementation. However, the theoretical results for the method in [17], are somewhat stronger; linear convergence could be established, and for a special choice of parameters, also a proof of polynomiality is given.

## References


[1] Andersen, E. D., K. D. Andersen (1995). Presolving in linear programming. *Math. Programming* **71** 221–245.

[2] Fourer, R., S. Mehrotra (1993). Solving symmetric indefinite systems in an interior-point method for linear programming. *Math. Programming* **62** 15–39.

[3] Freund, R. W., F. Jarre (1996). A QMR-based interior-point algorithm for solving linear programs. *Math. Programming* **76** 183–210.

[4] ——, *Numerical experiences with a QMR-based interior-point algorithm for linear programs.* Numerical Analysis Manuscript, Bell Laboratories, Murray Hill, New Jersey, in preparation.

[5] Freund, R. W., F. Jarre, S. Mizuno (1996). *Convergence of a Class of Inexact Interior-Point Algorithms for Linear Programs.* Numerical Analysis Manuscript 96–16, Bell Laboratories, Murray Hill, New Jersey. (Available on WWW at `http://cm.bell-labs.com/cs/doc/96`)

[6] Freund, R. W., N. M. Nachtigal (1991). QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.* **60** 315–339.

[7] Gay, D. M. (1985). Electronic mail distribution of linear programming test problems. *Mathematical Programming Society COAL Newsletter* **13** 10–12.

[8] Gondzio, J., T. Terlaky (1994). *A computational view of interior-point methods for linear programming.* Report 94-73, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft.

[9] Karmarkar, N. K. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica* **4** 373–395.

[10] Kojima, M., N. Megiddo, S. Mizuno (1993). A primal-dual infeasible-interior-point algorithm for linear programming. *Math. Programming* **61** 263–280.

[11] Kojima, M., S. Mizuno, A. Yoshise (1989). A primal-dual interior point algorithm for linear programming. N. Megiddo, ed., *Progress in Mathematical Programming, Interior-Point and Related Methods*, Springer-Verlag, New York, 29–47.

[12] Lustig, I. J., R. E. Marsten, D. F. Shanno (1991). Computational experience with a primal-dual interior point method for linear programming. *Linear Algebra Appl.* **152** 191–222.

[13] ——, (1992). On implementing Mehrotra's predictor-corrector interior-point method for linear programming. *SIAM J. Optim.* **2** 435–449.

[14] Megiddo, N. (1989). Pathways to the optimal set in linear programming. N. Megiddo, ed., *Progress in Mathematical Programming, Interior-Point and Related Methods*, Springer-Verlag, New York, 131–158.

[15] Mizuno, S. (1994). *Polynomiality of infeasible-interior-point algorithms for linear programming. Math. Programming* **67** 109–119.

[16] Mizuno, S., F. Jarre (1996). An infeasible-interior-point algorithm using projections onto a convex set. *Ann. Oper. Res.* **62** 59–80.

[17] ——, (1996). *Global and polynomial-time Convergence of an infeasible-interior-point algorithm using inexact computation.* Research Memorandum 605, The Institute of Statistical Mathematics, Tokyo.

[18] Monteiro, R. D. C., I. Adler (1989). Interior path following primal-dual algorithms, Part I: Linear programming. *Math. Programming* **44** 27–41.

[19] Portugal, L. F., M. G. C. Resende, G. Veiga, J. J. Judice (1994). *A truncated primal-infeasible dual-feasible interior point network flow method.* Technical Report, AT&T Bell Laboratories, Murray Hill, NJ. Revised version, 1996.

[20] Stoer J., R. Bulirsch (1993). *Introduction to Numerical Analysis, Second Edition*. Springer-Verlag, New York.

[21] Tanabe, K. (1988). Centered Newton method for mathematical programming. M. Iri, K. Yajima, eds., *System Modelling and Optimization*, Springer-Verlag, New York, 197–206.

[22] Vanderbei, R. J., T. J. Carpenter (1993). Symmetric indefinite systems for interior point methods. *Math. Programming* **58** 1–32.