Finding a point in the relative interior of a polyhedron

Coralia Cartis[*,‡] and Nicholas I. M. Gould[†,*,‡]

December 25, 2006

Abstract

A new initialization or 'Phase I' strategy for feasible interior point methods for linear programming is proposed that computes a point on the primal-dual central path associated with the linear program. Provided there exist primal-dual strictly feasible points — an all-pervasive assumption in interior point method theory that implies the existence of the central path — our initial method (Algorithm 1) is globally Q-linearly and asymptotically Q-quadratically convergent, with a provable worst-case iteration complexity bound. When this assumption is not met, the numerical behaviour of Algorithm 1 is highly disappointing, even when the problem is primal-dual feasible. This is due to the presence of implicit equalities, inequality constraints that hold as equalities at all the feasible points. Controlled perturbations of the inequality constraints of the primal-dual problems are introduced — geometrically equivalent to enlarging the primal-dual feasible region and then systematically contracting it back to its initial shape — in order for the perturbed problems to satisfy the assumption. Thus Algorithm 1 can successfully be employed to solve each of the perturbed problems. We show that, when there exist primal-dual strictly feasible points of the original problems, the resulting method, Algorithm 2, finds such a point in a finite number of changes to the perturbation parameters. When implicit equalities are present, but the original problem and its dual are feasible, Algorithm 2 asymptotically detects *all* the primal-dual implicit equalities and generates a point in the relative interior of the primal-dual feasible set. Algorithm 2 can also asymptotically detect primal-dual infeasibility. Successful numerical experience with Algorithm 2 on linear programs from NETLIB and CUTEr, both with and without any significant preprocessing of the problems, indicates that Algorithm 2 may be used as an *algorithmic preprocessor* for removing implicit equalities, with theoretical guarantees of convergence.

1 Introduction

1.1 Background and motivation

Finding a feasible point of a given polyhedron or detecting that the polyhedron is empty is a fundamental and ubiquitous problem in optimization. It is often referred to as the *linear feasibility problem* since algebraically it is equivalent to finding a solution to a given system of finitely many linear equalities and inequalities. A significant amount of research has been devoted to this topic (see [31, 40] and the references therein), in particular to constructing polynomial

[*]Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England, UK. Email: c.cartis@rl.ac.uk, n.i.m.gould@rl.ac.uk.

[†]Oxford University Computing Laboratory, Numerical Analysis Group, Wolfson Building, Parks Road, Oxford, OX1 3QD, England, UK. Email: nick.gould@comlab.ox.ac.uk.

[‡]This work was supported by the EPSRC grants GR/S34472 and GR/S42170.

time algorithms, with landmark methods such as that due to Khachiyan's [19] being developed in this context. The need to solve such problems occurs not only in numerous real-life applications, but also in the context of Linear (LP) and Nonlinear (NLP) programming. Reformulating an LP problem as a linear feasibility one has traditionally been regarded as a way for solving the former [31], though this view is now mostly considered computationally inefficient. Nevertheless, the relevance of linear feasibility problems as a useful initialization or 'Phase I' approach for LP — and also for NLP problems with (some) linear constraints — has prevailed, particularly when employing interior point-type methods to solve these problems. For the so-called *feasible* variants [15, 30, 36] of the latter algorithms, a starting point needs not only be feasible with respect to the constraints, but *strictly feasible*, satisfying *each* inequality constraint *strictly*. In the NLP context, where many problems involve a mixture of linear and nonlinear constraints, the desired outcome of such an initialization strategy is a reduction in problem dimension, by the satisfaction of the linear constraints of the problem (or a certificate that they are inconsistent) prior to the much harder computational task of satisfying the nonlinear constraints and solving the problem. Once a (strictly) feasible point for the linear constraints has been computed, algorithms for solving LP and NLP problems can subsequently keep these constraints (strictly) satisfied without much difficulty. The infeasibility detection component of an initialization strategy may also be employed during the course of an NLP algorithm for local infeasibility detection of the quadratic or linearized model of the problem (for example, in SQP or interior point methods).

In this paper, we develop and analyse a Phase I strategy suitable for initializing feasible *primal-dual* Interior Point Methods (IPMs) for LP, which can also be straightforwardly applied to finding a strictly feasible point of the polyhedron determined by the linear constraints of an NLP. Over the past fifteen years, primal-dual IPMs [36], with both feasible and infeasible variants (see, for example, [21] and [20], respectively), have emerged as the most successful IPMs for solving (especially large-scale) LP problems, while enjoying good theoretical convergence and complexity properties. See [13, 15, 36] for comprehensive reviews of the field of IPMs. In this context, one considers a linear program of the form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; c^T x \;\text{ subject to }\; Ax = b \;\text{ and }\; x \geq 0, \tag{P}$$

and its dual

$$\underset{y\in\mathbb{R}^m, s\in\mathbb{R}^n}{\text{maximize}} \;\; b^T y \;\text{ subject to }\; A^T y + s = c \;\text{ and }\; s \geq 0, \tag{D}$$

often denoted by (PD), as well as the *central path* associated with (P) and (D), which is the solution $v(\mu) := (x(\mu), y(\mu), s(\mu))$ to

$$Ax - b = 0 \tag{1.1a}$$
$$A^T y + s - c = 0 \tag{1.1b}$$
$$XSe = \mu e \tag{1.1c}$$
$$\text{and } x > 0, \;\; s > 0, \tag{1.1d}$$

as a function of the scalar parameter $\mu > 0$ [23, 36]; here e is the vector of ones and X and S are diagonal matrices whose diagonal entries are the components of x and s. The central path is well-defined provided that problems (P) and (D) have a primal-dual strictly feasible point and the matrix A has full row rank [36] (see the IPM conditions in Section 3). The system (1.1) represents the KKT optimality conditions of the following strictly convex problem

$$\underset{(x,y,s)}{\text{minimize}} \;\; c^\top x - b^\top y - \mu \sum_{i=1}^{n} \log x_i - \mu \sum_{i=1}^{n} \log s_i \;\; \text{subject to} \;\; Ax = b \;\; \text{and} \;\; A^\top y + s = c. \quad (1.2)$$

The presence of the log terms on all the primal-dual inequality constraints encourages the solution $v(\mu)$ of (1.2) to be away from the boundaries of the feasible set of (P) and (D), allowing it to approach it only in the limit, as $\mu \to 0$, when, indeed, $v(\mu)$ converges to a solution of (P) and (D) [23, 40]. Feasible primal-dual IPMs attempt to follow this or some other parametrization [15] of the central path, to a solution of the problems. Thus a natural approach for our 'Phase I' strategy is to aim to find *a* point on the central path, preferably with μ not too 'small' in order to discourage the ill-conditioning caused by the barrier terms of (1.2) at points close to the boundaries. Specifically, we fix μ and wish to find $v(\mu)$ satisfying (1.1), which is a primal-dual strictly feasible point.

1.2 Algorithm 1

Algorithm 1, described in Section 2, attempts to achieve the above-mentioned goal by applying damped a Newton method to the system (1.1), with a linesearch that ensures sufficient decrease in a measure Φ of the error in (1.1) (see (2.7)), while guaranteeing the iterates do not get too "close" to the boundaries of (P) and (D). Under the IPM conditions, we show that Algorithm 1 is globally Q-linearly convergent and that the iterates $v_k = (x_k, y_k, s_k)$ generated by the algorithm converge to $v(\mu)$, as $k \to \infty$ (see Theorem 3.5). Furthermore, Algorithm 1 is asymptotically Q-quadratically convergent and $v_k \to v(\mu)$ at least Q-superlinearly. If in Algorithm 1 we employ a particular choice of stepsize (given by (2.17) and (3.44)) then the stepsize becomes equal to one, and thus the iterates become strictly feasible for (P) and (D), after finitely many iterations; subsequently, our measure of the error $\Phi(v_k)$ converges Q-quadratically to zero, as do the iterates v_k to $v(\mu)$. In Section 3.2, we give an upper bound on the number of iterations required by Algorithm 1 to terminate with $\Phi(v_k)$ less than a given tolerance $\epsilon > 0$, by giving a precise bound on the number of iterations required for the quadratic convergence of the error Φ to zero to occur. Our complexity results are dependent upon the norm that is used to define Φ; for example, if we use the l_2 norm, then, besides $\log \epsilon$, our bounds depend explicitly only on $\Phi(v_0)$, μ and a pre-defined parameter θ of the stepsize (see Corollary 3.11). For other norms, such as l_1 and l_∞, a polynomial explicit dependence on n also occurs (see Corollary 3.11). We develop the entire theory for Algorithm 1, and not just its complexity, in terms of general norms on $\mathbb{R}^n$, so as to provide user-flexibility. The best complexity results for Algorithm 1 are obtained for the l_2 norm, while for convenience in our implementations, we employ the l_∞ norm.

1.3 Some disadvantages of the assumptions

In Section 5.1, our numerical experience with a Fortran 95 implementation of Algorithm 1 on NETLIB problems [27], without any preprocessing of the problems except to ensure that A has full row rank, confirms our theoretical results, with Example 1 illustrating the satisfactory behaviour of Algorithm 1 that we generally observe when the IPM conditions hold.

Example 2 and 3 in the same section portray, however, the highly disappointing performance of Algorithm 1 that we have also encountered on a significant number of problems in the NETLIB collection. There, Algorithm 1 reaches the maximum number of iterations allowed ($k = 1000$), without generating the required approximate solution of (1.1) for the given μ and tolerance ϵ. An approximately primal-dual feasible point is obtained in most cases, though possibly not to high accuracy, but not a strictly feasible one as some of the variables are very close to their bounds, while the corresponding dual ones are very large and increasing in magnitude, causing severe ill-conditioning. We rigorously show on the simple Example 3 that this behaviour is not due to a poor implementation of the algorithm, or to the choice of μ, but rather to the problem in question having primal-dual feasible, but not strictly feasible, points and thus, not satisfying the IPM conditions. Then, our general convergence and complexity results do not apply, and the central path (1.1) may not be well-defined.

When (P) and (D) are feasible, the absence of strictly feasible points for the feasible set of problems (P) and (D) indicates the presence of *implicit equalities* — inequality constraints that hold as equalities at all the feasible points. Geometrically, the implicit equalities of the primal-dual feasible set, together with its equality constraints, form its *affine hull* (see (6.69)), the "smallest" affine subspace containing this set. By assuming the existence of strictly feasible points, and also since the (nonredundant) implicit equalities are not known a priori, we may be overestimating the dimension of our feasible set; for example, we might think the feasible set is a cube in $\mathbb{R}^3$, when it is, in fact, a square in $\mathbb{R}^3$.

To assess the unfavorable implications of implicit equalities on the behaviour of Algorithm 1 and IPMs, recall that in the barrier terms of problem (1.2) — whose solution is the target point that Algorithm 1 attempts to compute — we included *all* the inequality constraints of problems (P) and (D). Since one does not know in advance the implicit equalities of (PD), by attempting to solve (1.1), one mistakingly includes all its inequalities constraints into the log barrier terms of (1.2), some of which thus blow up and are ultimately ill-defined.

Furthermore, the IPM conditions are the assumptions under which, the convergence and complexity analysis of, not only Algorithm 1, but of every (primal-dual) IPM we are aware of, with the notable exception of the approach in [41], is performed. Thus we are left with no known theoretical guarantees of convergence for these algorithms when applied to problems that are feasible, and thus have solutions, but that do not satisfy the IPM conditions.

There is no real need to emphasize that problems that do not satisfy these assumptions are frequently encountered in practice. For example, overspecifying an optimization model problem is a common occurrence as a practitioner setting up a model may simply wish to ensure that the constraints are satisfied, and is not concerned with detecting dependencies in the data and the constraints. The use of preprocessors for such purposes (see for example, [14]), before applying the algorithm in question to the problem, has been a standard approach among software developers, though its effectiveness is code-dependent and not guaranteed to be optimal. Several approaches for detecting implicit equalities in a polyhedral set of constraints have been proposed [6, 22, 28, 34], etc., the most relevant to our work being [32]. In the latter article, an auxiliary problem is constructed whose solution fully identifies the implicit equalities in (P) and is computed using (any) IPM. When applying this method in the context of the primal-dual problem, the dimension of the auxiliary reformulation doubles. Furthermore, we run into a vicious circle where we want to devise a Phase I for IPMs but need to apply an IPM to it. Thus we consider the approach in [32] to be unsuitable for our purposes.

1.4 Algorithm 2: overcoming the assumptions by controlled perturbations

In order to improve the numerical behaviour of Algorithm 1 when the IPM conditions are not satisfied, as well as its capacity to detect implicit equalities and infeasibility, we consider constructing in Section 5.2, a sequence of relaxations of our original problem, each of which satisfies the IPM conditions, and is thus amenable to Algorithm 1. The resulting method, Algorithm 2, has the following geometric interpretation. In its first iteration $k = 0$, we *enlarge* the primal-dual feasible set by perturbing the nonnegativity bounds of (P) and (D) by some quantities $-\lambda_0 \in \mathbb{R}^n$ and $-\gamma_0 \in \mathbb{R}^n$, respectively, where $(\lambda_0, \gamma_0) \geq 0$, which are chosen so as to ensure that the perturbed problem admits strictly feasible points (this condition can always be achieved without much difficulty if A has full row rank as we show in Section 5.2.1). We then apply Algorithm 1 to the perturbed problem and find (exactly) a point on its central path that we denote by $v_1 = (x_1, y_1, s_1)$. Subsequently, as described in Section 5.2.2, we *successively shrink* this enlarged polyhedron by reducing the perturbation parameters (λ_k, γ_k) such that the current iterate of Algorithm 2, $v_k = (x_k, y_k, s_k)$, $k \geq 1$, remains strictly feasible for the next (i. e., $(k+1)$st) shrunken polyhedron. Initializing Algorithm 1 at v_k, we again compute, exactly, a point on the central path of the $(k+1)$st perturbed feasible set. Algorithm 2 terminates when a strictly feasible point of (P) and (D) is found. In Section 6, the convergence of Algorithm 2 is analysed under the general assumption that the matrix A has full row rank, which is easy to satisfy.

1.5 Finite termination of Algorithm 2 when the IPM conditions hold

In Section 6.1, we show that if problems (P) and (D) admit strictly feasible points, then Algorithm 2 will find such a point in a finite number of changes to the perturbation parameters (λ_k, γ_k), or equivalently, of major iterations k (see Corollary 6.5, which also gives a bound on k). The perturbation parameters are equal to zero at termination. Furthermore, if the iterate at termination is "far" from the central path of (PD), one more run of Algorithm 1 starting from that point (i. e., one more iteration of Algorithm 2) will bring it onto the central path of (PD).

1.6 Asymptotic detection of all primal-dual implicit equalities and transformation of the problems

In Section 6.2.1, assuming that problems (P) and (D) are feasible, but not strictly feasible, we are able to fully characterize the set of implicit equalities of (P) and (D) in terms of the asymptotic behaviour of the (major) sequence of iterates $v_k = (x_k, y_k, s_k)$ generated by Algorithm 2. Corollary 6.12 summarizes these results. In particular, an index $i \in \{1, \ldots, n\}$ corresponds to a primal implicit equality (i. e., $x^i = 0$ for all primal feasible points) if and only if $x_k^i \to 0$, or equivalently, $s_k^i \to \infty$, as $k \to \infty$. Similarly, $j \in \{1, \ldots, n\}$ is the index of a dual implicit equality if and only if $x_k^j \to \infty$, or equivalently, $s_k^j \to 0$, as $k \to \infty$. Characterizations of the set of primal-dual inequality constraints that can be strictly satisfied are also given in Corollary 6.12. Furthermore, the perturbation parameters (λ_k, γ_k) are shown to converge to zero as $k \to \infty$. These characterizations asymptotically allow us to fully identify the set of primal-dual implicit equalities. We further employ this information in Section 6.2.1.1 to transform problems (P) and (D) into (6.100) and (6.101) as follows. The primal implicit equalities (i. e., $x^i = 0$) and the corresponding dual constraints may be *removed* from (P) and (D), thus reducing their dimensions. Furthermore, the dual implicit equalities (i. e., $s^j = 0$) determine the corresponding dual constraint $A_j^\top y + s^j = c^j$ to become $A_j^\top y = c^j$, and the corresponding primal variables to be *freed from their bounds*. Moreover, the finite components of any limit point of the iterates of Algorithm 2 provide a strictly feasible point of the transformed problems (6.100) and (6.101), which is on their central path. It is these transformed problems that a feasible IPM will now solve to optimality, starting from this strictly feasible point that we have computed using Algorithm 2.

1.7 Detecting infeasibility

Algorithm 2 is also able to detect asymptotically when (PD) is infeasible. Corollary 6.13 shows that if the limit point of the (convergent) sequence of perturbation parameters is not the zero vector, then (PD) is infeasible.

1.8 An inexact variant of Algorithm 2

In the construction of Algorithm 2 and its analysis, we have assumed that each perturbed subproblem is solved *exactly* by Algorithm 1. To bridge the gap with practical computations, in Section 7, we allow each subproblem k to be solved by Algorithm 1 to accuracy $\epsilon_k > 0$. The resulting method, Algorithm 2_ϵ, inherits the convergence properties of Algorithm 2, under very mild assumptions on the tolerances ϵ_k. For example, if the l_1, l_2 or l_∞ norm is employed in Φ, then sufficient conditions on (ϵ_k) are that they are bounded above by μ and away from it (see (7.10)); thus, they need not converge to zero. Then, if in addition, the IPM conditions are satisfied by (PD), Theorem 7.3 shows finite termination of Algorithm 2_ϵ in terms of major iterations k, which together with Corollary 3.11 that estimates the convergence of Algorithm 1, yields Corollary 7.4, proving finite termination of Algorithm 2_ϵ in terms of the total number of iterations. The remarks following Corollary 7.4 further detail some upper bounds on the number of total iterations. Furthermore, the characterizations of the set of implicit equalities of (PD) in terms of the asymptotic behaviour of the iterates, as well as the asymptotic detection of infeasibility, also hold for Algorithm 2_ϵ, as stated in Corollary 7.5 and the remarks that follow.

1.9 Numerical experience

We have implemented Algorithm 2_ϵ, which naturally contains Algorithm 1, as a Fortran 95 module WCP (Well-Centred Point) as part of Release 2.0 of the nonlinear optimization library GALAHAD [17]. The main details of the implementation are described in Section 8. We have tested the complete set of NETLIB LP problems and other LP test problems as distributed with CUTEr [16], without any preprocessing of the problems, except for ensuring that the matrix A has full row rank. The purpose of our tests is to persuade the reader that our algorithm achieves its objectives: when primal-dual strictly feasible points exist, it will find such a point efficiently; else, if the primal-dual problems are feasible, it will detect the primal-dual implicit equalities and compute a point in the relative interior of the feasible set, on the central path of some transformed problems, or provide a certificate of infeasibility. The results are given in Table 8.1, where we present the surprisingly large number of implicit primal and dual equalities that our algorithm finds, as well as other relevant information.

We also applied WCP to the NETLIB LP problems after they have been preprocessed using the CPLEX preprocessor [8]. The CPLEX preprocessor fails to find all, and indeed sometimes many, of the implicit equalities for 25 out of the 98 NETLIB problems, the results for the former being reported in Table 8.2. These experiments suggest that WCP may be employed as an algorithmic preprocessor for test problems, possibly after they have already been preprocessed by a common presolver, in order to ensure that all implicit equalities have been removed. The strong theoretical guarantees of convergence of Algorithm $2/2_\epsilon$ that is implemented in WCP make this software a rare occurence on the preprocessor "market".

1.10 Further relating of our work to existing literature

Numerous initialization strategies for LP have been proposed in the context of IPMs (see the introduction in [41] or Appendix A in [7] for a discussion on the advantages or otherwise of various approaches, as well as extensive bibliography). The most notable of these techniques — the current state-of-the-art in public and commercial software — are the so-called *Homogeneous Self-Dual* (HSD) reformulation [41] of an LP that is then solved using a feasible primal-dual IPM, and the *infeasible* IPMs that do not require a strictly feasible point for initialization (see for example, [20]). In the former approach, a primal-dual strictly feasible point on the central path is readily available for the reformulated problem, and the dimension of the problem does not increase significantly when reformulating because of self-duality. Furthermore, it provides an asymptotic criteria for infeasibility detection. It applies however, exclusively to the solution of LP problems, and not to computing a strictly feasible point of the feasible set of problems (P) and (D). Thus it cannot be used as a Phase I in the context of NLP problems, for example. The latter statement also applies to infeasible IPMs.

Despite highly insightful work [35] on infeasibility detection of (P) or (D) when employing infeasible IPMs, we believe that in computational practice, this issue is not fully resolved. Also, existing criteria that may be used for checking that infeasibility of (P) or (D) occurs when running an infeasible IPM (see for example, Theorem 9.7 and 9.8 in [36]) are unsatisfying. Furthermore, the entire convergence theory for infeasible IPMs relies on the IPM conditions being satisfied by (P) and (D), which implies that (P) and (D) are feasible and have solutions. No Phase I is usually employed to ensure these conditions are achieved by (some transformation of) the problems, though the problems are in general preprocessed. However, as we show in Section 8, due to its heuristical nature and lack of theoretical guarantees of convergence, preprocessing may fail to ensure the IPM conditions (see Table 8.2). The infeasible starting point strategy of Mehrotra [24] has been very successful when used in conjunction with infeasible primal-dual IPMs (see [24, 36]). It is however highly heuristical, with no theoretical guarantees of convergence, and its success is poorly understood.

1.11 The structure of the paper and notations

The structure of the paper is as follows. Section 2 details the construction of Algorithm 1, while Section 3 presents its convergence (Subsection 3.1) and complexity properties (Section 3.2). Section 4 succintly describes some possible, higher-order, extensions of Algorithm 1. Subsection 5.1 presents some satisfactory numerical experience with an implementation of Algorithm 1 (see Example 1), as well as some disappointing results (Example 2 and 3), when the problems do not have strictly feasible points. The latter dissatisfying behaviour of Algorithm 1 leads us to consider in Subsection 5.2 a modification of Algorithm 1 – Algorithm 2 – that employs successive relaxations of the inequality constraints of problems (P) and (D) in such a way so as to ensure that each of the perturbed problems satisfies the IPM conditions, and can thus be solved efficiently by Algorithm 1. Section 6 presents the convergence properties of Algorithm 2. In Subsection 6.1, we show that it takes a finite number of changes to the perturbation parameters for Algorithm 2 to generate a strictly feasible point of (PD), provided such points exist. Furthermore, if the point we obtain is "far" from the central path, one more iteration of Algorithm 2, starting from that point, will bring it onto the central path of (PD). In Subsection 6.2.1, we prove that when (PD) is feasible, but not strictly feasible, Algorithm 2 can find all primal-dual implicit equalities of (PD) asymptotically and generates a point in the relative interior of its feasible set. We employ this information to transform problems (P) and (D) into problems (6.100) and (6.101), for which the finite components of any limit point of the iterates provide a strictly feasible point, on the central path of the transformed problems. It is these problems, (6.100) and (6.101), that a feasible IPM will now solve to optimality, starting from their strictly feasible point that we have computed. Also, our algorithm can detect when (PD) is infeasible, asymptotically. In Algorithm 2, each perturbed subproblem is solved exactly using Algorithm 1. In Section 7, we relax this assumption and allow each subproblem to be solved only approximately. We then show that the resulting method, Algorithm 2_ϵ, inherits all the convergence properties of Algorithm 2, under a very mild assumption on the optimality tolerance that each subproblem is solved to. Section 8 details our numerical experiments with Algorithm 2_ϵ on the NETLIB LP problems and on other problems from the CUTEr test set, without and with preprocessing of the problems. We end our paper in Section 9, with conclusions.

Throughout, we employ the notation: vector components will be denoted by superscripts, and iteration numbers, by subscripts. Also, I is the $n \times n$ identity matrix and e, the n dimensional vector of all ones. The matrices X_k and S_k are diagonal, with the components of x_k and s_k as entries, respectively.

2 A basic Newton method

An obvious way to solve (1.1), for a fixed $\mu > 0$, is to apply a safeguarded Newton iteration. Specifically, given an iterate $v_k := (x_k, y_k, s_k)$ for which

$$x_k > 0 \;\text{ and }\; s_k > 0, \tag{2.1}$$

a Newton step $\dot{v}_k := (\dot{x}_k, \dot{y}_k, \dot{s}_k)$ satisfying

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{s}_k \end{pmatrix} = - \begin{pmatrix} Ax_k - b \\ A^T y_k + s_k - c \\ X_k S_k e - \mu e \end{pmatrix} \tag{2.2}$$

is computed. Due to (2.1), the matrix in (2.2) is nonsingular provided A has full row rank, and, for the purpose of our future theoretical analysis, we express the search direction $\dot{v}_k = (\dot{x}_k, \dot{y}_k, \dot{s}_k)$ explicitly

$$\dot{y}_k = -M_k^{-1}[r_k^p + AD_k^2 r_k^d - AS_k^{-1}(X_k s_k - \mu e)], \tag{2.3a}$$

$$\dot{s}_k = A^\top M_k^{-1} r_k^p - [I - A^\top M_k^{-1} AD_k^2] r_k^d - A^\top M_k^{-1} AS_k^{-1}(X_k s_k - \mu e), \tag{2.3b}$$

$$\dot{x}_k = -D_k^2 \dot{s}_k - S_k^{-1}(X_k s_k - \mu e), \tag{2.3c}$$

where $M_k := AD_k^2 A^\top$, $D_k := X_k^{1/2} S_k^{-1/2}$, $r_k^p := Ax_k - b$ and $r_k^d := A^\top y_k + s_k - c$.

The next iterate is then set to be

$$v_{k+1} := (x_{k+1}, y_{k+1}, s_{k+1}) = v_k(\alpha_k), \tag{2.4}$$

where

$$v_k(\alpha) := \begin{pmatrix} x_k(\alpha) \\ y_k(\alpha) \\ s_k(\alpha) \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \\ s_k \end{pmatrix} + \alpha \begin{pmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{s}_k \end{pmatrix} \tag{2.5}$$

and the stepsize $\alpha_k \in (0,1]$ is chosen both to ensure a significant reduction in a suitable merit function and to guarantee that

$$x_{k+1} > 0 \ \text{ and } \ s_{k+1} > 0. \tag{2.6}$$

We now consider how to achieve these goals.

2.1 A merit function

Letting $v := (x, y, s)$, an obvious merit function is

$$\Phi(v) := \|Xs - \mu e\| + r(v), \ \text{ where } \ r(v) := \|Ax - b\| + \|A^T y + s - c\|, \tag{2.7}$$

where $\|\cdot\|$ denotes any norm on $\mathbb{R}^n$ (independent of k). The function Φ only vanishes at the solution $v(\mu)$ to (1.1). For such a merit function, it follows immediately from (2.2) that for any $\alpha \in [0,1]$,

$$r(v_k(\alpha)) = (1-\alpha) r(v_k), \tag{2.8}$$

and

$$X_k(\alpha)s_k(\alpha) - \mu e = (1-\alpha)(X_k s_k - \mu e) + \alpha^2 \dot{X}_k \dot{s}_k. \tag{2.9}$$

Thus we have

$$\Phi(v_k(\alpha)) \leq (1-\alpha)\Phi(v_k) + \alpha^2 \|\dot{X}_k \dot{s}_k\|, \quad \alpha \in [0,1], \tag{2.10}$$

and

$$\Phi(v_k(\alpha)) \geq (1-\alpha)\Phi(v_k) - \alpha^2 \|\dot{X}_k \dot{s}_k\|, \quad \alpha \in [0,1]. \tag{2.11}$$

Notice that (2.10) implies that $\Phi(v_k(\alpha)) < \Phi(v_k)$ for all sufficiently small α provided that v_k is not the solution to (1.1).

2.2 Ensuring positivity of the iterates

Given (2.1), one way to ensure (2.6) is to require that

$$X_k s_k \geq \theta \mu e, \tag{2.12}$$

for some $\theta \in (0,1)$, and only allow $\alpha_k \in (0,1]$ for which

$$X_k(\alpha)s_k(\alpha) \geq \theta \mu e, \tag{2.13}$$

for all $0 \leq \alpha \leq \alpha_k$. Fortunately, (2.9) gives that

$$X_k(\alpha)s_k(\alpha) - \theta\mu e = (1-\alpha)(X_k s_k - \theta \mu e) + \alpha(1-\theta)\mu e + \alpha^2 \dot{X}_k \dot{s}_k. \tag{2.14}$$

Thus, for each $i \in \{1,\ldots,n\}$, (2.12) requires either $x_k^i s_k^i > \theta\mu$, in which case the dominant term in (2.14) is positive and independent of α, or $x_k^i s_k^i = \theta\mu$, in which case the dominant term is $\alpha(1-\theta)\mu > 0$. Hence, in either case, (2.13) holds for all sufficiently small $\alpha \geq 0$.

2.3 The choice of stepsize

At each v_k, having computed $\dot{v}_k$ from (2.2), we find the largest $\alpha_k^{\mathrm{U}} \in (0,1]$ for which (2.13) holds for all $\alpha \in [0, \alpha_k^{\mathrm{U}}]$, and let

$$\alpha_k^{\mathrm{Q}} := \underset{\alpha \in [0,\alpha_k^{\mathrm{U}}]}{\arg\min} \{(1-\alpha)\Phi(v_k) + \alpha^2 \|\dot{X}_k \dot{s}_k\|\}. \tag{2.15}$$

Then we pick any α_k for which

$$\Phi(v(\alpha_k)) \leq (1-\alpha_k^{\mathrm{Q}})\Phi(v_k) + (\alpha_k^{\mathrm{Q}})^2 \|\dot{X}_k \dot{s}_k\|. \tag{2.16}$$

Recalling (2.10), α_k may be chosen, for example, as the global minimizer of $\Phi(v_k(\alpha))$ in the interval $[0, \alpha_k^{\mathrm{U}}]$, or simply as α_k^{Q}.

The function on the right-hand side of (2.15) is a strictly convex quadratic in α. Thus α_k^{Q} is either the global minimizer of this quadratic function or the nonzero end point of the interval of minimization. In particular, we have

$$\alpha_k^{\mathrm{Q}} = \begin{cases} \frac{\Phi(v_k)}{2\|\dot{X}_k \dot{s}_k\|} := \alpha_k^{\mathrm{G}}, & \text{if } \alpha_k^{\mathrm{G}} \leq \alpha_k^{\mathrm{U}}, \\ \alpha_k^{\mathrm{U}}, & \text{otherwise.} \end{cases} \tag{2.17}$$

While $v_k \neq v(\mu)$, i. e., $\Phi(v_k) \neq 0$ and $\|\dot{X}_k \dot{s}_k\| \neq 0$, the relations (2.17), $\alpha_k^{\mathrm{G}} > 0$ and $\alpha_k^{\mathrm{U}} > 0$, provide the inequalities

$$0 < \alpha_k^{\mathrm{Q}} \leq \frac{\Phi(v_k)}{2\|\dot{X}_k \dot{s}_k\|}. \tag{2.18}$$

The first inequality in (2.18) and the last sentence of Section 2.2 imply that (2.16) allows positive steps α_k. Moreover, both inequalities in (2.18) provide that (2.16) is not satisfied by $\alpha_k = 0$. The proof that α_k does not become arbitrarily small as k increases, is deferred until Theorem 3.5.

2.4 Algorithm 1

As we have seen, it is easy to guarantee that condition (2.12) holds for each $k \geq 0$ provided it holds for $k = 0$. The latter will be ensured by a judicious choice of θ at $k = 0$.

Our algorithm can be summarized as follows.

Algorithm 1.

Assume a starting point $v_0 = (x_0, y_0, s_0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ with $x_0 > 0$ and $s_0 > 0$ is available. Choose $\theta \in (0,1)$ such that

$$\theta \leq \frac{1}{\mu_0} \min\{x_0^i s_0^i : i = \overline{1,n}\}, \tag{2.19}$$

and let $\epsilon > 0$ be a tolerance parameter.

On each iteration $k \geq 0$, DO:

Step 1. If $\Phi(v_k) \leq \epsilon$, STOP.

Step 2. Compute $\dot{v}_k$ from (2.2), and a stepsize α_k to satisfy (2.16).

Step 3. Set the new iterate v_{k+1} as in (2.4).

Step 4. Let $k := k+1$, and GO TO Step 1. $\diamond$

In the next section we investigate the global and asymptotic convergence properties of Algorithm 1, as well as its iteration complexity.

3 Convergence analysis for Algorithm 1

We assume that the following conditions are satisfied.

The IPM conditions.

A1). The matrix A has full row rank.

A2). There exists a *primal-dual strictly feasible* point $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ for (P) and (D), i. e.,

$$A\hat{x} = b, \quad A^\top \hat{y} + \hat{s} = c, \text{ and } (\hat{x}, \hat{s}) > 0. \tag{3.20}$$

The IPM conditions are equivalent to requiring the primal-dual feasible set of (P) and (D), and the primal-dual solution set, to be nonempty and bounded, respectively [7, 36].

The norm $\|\cdot\|$ on $\mathbb{R}^n$ that we employ throughout, is arbitrary provided there exist positive constants $\underline{p}$ and $\overline{p}$, independent of the iteration number k, such that

$$\underline{p}\|x\|_2 \leq \|x\| \leq \overline{p}\|x\|_2, \text{ for all } x \in \mathbb{R}^n. \tag{3.21}$$

We remark that the pair $(\underline{p}, \overline{p}) := (1, \sqrt{n})$ is optimal for the l_1 norm, and $(\underline{p}, \overline{p}) := (1/\sqrt{n}, 1)$, for the l_∞ case.

3.1 Global and asymptotic convergence

Firstly, we show that the sequence (x_k, s_k) is bounded, and thus it has limit points. We then prove that $(\Phi(v_k))$ converges globally to zero with Q-linear rate, as $k \to \infty$, which implies that $v_k \to v(\mu)$, our target point on the central path. Moreover, the asymptotic rates at which $(\Phi(v_k))$ and (v_k) converge are Q-quadratic and at least Q-superlinear, respectively.

The next lemma is useful.

Lemma 3.1 *(see Lemma 4.1 in [42], and [5, 38]) Let problems (P) and (D) satisfy the IPM conditions and let $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ be any primal-dual strictly feasible point. Suppose Algorithm 1 generates iterates $v_k = (x_k, y_k, s_k)$, $k \geq 0$. Consider the auxiliary sequences*

$$u_{k+1} = u_k + \alpha_k(x_k + \dot{x}_k - u_k), \tag{3.22a}$$

$$t_{k+1} = t_k + \alpha_k(y_k + \dot{y}_k - t_k), \tag{3.22b}$$

$$w_{k+1} = w_k + \alpha_k(s_k + \dot{s}_k - w_k), \quad k \geq 0, \tag{3.22c}$$

where α_k is the stepsize computed by our algorithm and $(u_0, t_0, w_0) := (\hat{x}, \hat{y}, \hat{s})$. Then

$$\begin{aligned} x_k - u_k &= \nu_k(x_0 - \hat{x}), \\ s_k - w_k &= \nu_k(s_0 - \hat{s}), \quad k \geq 0, \end{aligned} \tag{3.23}$$

where

$$\nu_k := \prod_{i=1}^{k-1}(1-\alpha_i) \in [0,1), \; k \geq 1, \;\; \textit{and} \;\; \nu_0 := 1. \tag{3.24}$$

Also,

$$Ax_k = b \;\; \textit{and} \;\; A^\top t_k + s_k = c, \quad k \geq 0. \tag{3.25}$$

Proof. The arguments can be found in the proof of Lemma 4.1 in [42]. □

Lemma 3.2 *Under the conditions of Lemma 3.1, the sequence (x_k, s_k), $k \geq 0$, is, componentwise, bounded both from above and away from zero.*

In particular, we have the bounds

$$C_1 := \frac{\mu\theta\underline{m}}{L} \leq x_k^i, \, s_k^i \leq \frac{L}{\underline{m}} := C_2, \;\; \textit{for all} \;\; k \geq 0 \;\; \textit{and} \;\; i \in \{1,\ldots,n\}, \tag{3.26}$$

where $\underline{m} := \min\{x_0^i, \hat{x}^i, s_0^i, \hat{s}^i \, : \, i = \overline{1,n}\}$ *and* $L := \hat{x}^\top \hat{s} + \sqrt{n}\underline{p}^{-1}\Phi(v_0) + n\mu + |\hat{x}^\top(s_0 - \hat{s})| + |\hat{s}^\top(x_0 - \hat{x})| + |(x_0 - \hat{x})^\top(s_0 - \hat{s})|$.

Proof. From (3.25) and (3.20), we deduce that $\hat{x} - u_k$ is in the null space of the matrix A, while $\hat{s} - w_k$ belongs to the range space of $A^\top$, for $k \geq 0$. Thus we have the orthogonality property

$$(\hat{x} - u_k)^\top(\hat{s} - w_k) = 0, \quad k \geq 0, \tag{3.27}$$

which, recalling (3.23), is equivalent to

$$[\hat{x} - x_k + \nu_k(x_0 - \hat{x})]^\top[\hat{s} - s_k + \nu_k(s_0 - \hat{s})] = 0, \quad k \geq 0. \tag{3.28}$$

By expanding (3.28), we further obtain

$$\begin{aligned} &x_k^\top[\hat{s} + \nu_k(s_0 - \hat{s})] + s_k^\top[\hat{x} + \nu_k(x_0 - \hat{x})] = \\ &\hat{x}^\top\hat{s} + x_k^\top s_k + \nu_k\hat{x}^\top(s_0 - \hat{s}) + \nu_k\hat{s}^\top(x_0 - \hat{x}) + \nu_k^2(x_0 - \hat{x})^\top(s_0 - \hat{s}), \; k \geq 0. \end{aligned} \tag{3.29}$$

Since $\hat{s} > 0$, $s_0 > 0$ and $\nu_k \in [0,1]$, we have

$$0 < \underline{s}^i := \min\{\hat{s}^i, s_0^i\} \leq \hat{s}^i + \nu_k(s_0^i - \hat{s}^i) \leq \max\{\hat{s}^i, s_0^i\}, \quad k \geq 0, \quad i \in \{1,\ldots,n\}, \tag{3.30}$$

and similarly,

$$0 < \underline{x}^i := \min\{\hat{x}^i, x_0^i\} \leq \hat{x}^i + \nu_k(x_0^i - \hat{x}^i) \leq \max\{\hat{x}^i, x_0^i\}, \quad k \geq 0, \quad i \in \{1,\ldots,n\}. \tag{3.31}$$

Now, letting $\underline{x} := (\underline{x}^i : i = \overline{1,n}) > 0$ and $\underline{s} := (\underline{s}^i : i = \overline{1,n}) > 0$ and recalling that $x_k > 0$ and $s_k > 0$, we obtain

$$x_k^\top\underline{s} + s_k^\top\underline{x} \leq x_k^\top[\hat{s} + \nu_k(s_0 - \hat{s})] + s_k^\top[\hat{x} + \nu_k(x_0 - \hat{x})], \quad k \geq 0. \tag{3.32}$$

The iterates v_k generated by Algorithm 1 have the property $\Phi(v_{k+1}) < \Phi(v_k)$, $k \geq 0$ (see Subsections 2.1 and 2.3). Thus we have $\Phi(v_k) \leq \Phi(v_0)$, $k \geq 0$, and

$$\|X_k s_k - \mu e\| \leq \Phi(v_0), \quad k \geq 0,$$

which further gives from (3.21),

$$\|X_k s_k - \mu e\|_2 \leq \frac{1}{\underline{p}} \Phi(v_0), \quad k \geq 0.$$

Since $x_k^\top s_k \leq \sqrt{n}\|X_k s_k\|_2$, we deduce

$$x_k^\top s_k \leq \frac{\sqrt{n}}{\underline{p}} \Phi(v_0) + n\mu, \quad k \geq 0. \tag{3.33}$$

Substituting (3.33) on the right-hand side of (3.29), together with $\nu_k \in [0, 1)$, and employing (3.32) on its left-hand side, we obtain

$$x_k^\top \underline{s} + s_k^\top \underline{x} \leq L, \quad k \geq 0, \tag{3.34}$$

where L is the constant defined in the statement of the lemma and is independent of k. Since the components of x_k, s_k, $\underline{x}$ and $\underline{s}$ are positive, we conclude from (3.34) that x_k and s_k are bounded above (independently of k), and the componentwise upper bound in (3.26) holds. This and (2.12) provide the lower bound in (3.26), and thus, x_k and s_k are also, componentwise, bounded away from zero. □

We remark that, if the starting point $v_0 = (x_0, y_0, s_0)$ of Algorithm 1 is strictly feasible for (PD), we may let $v_0 = \hat{v}$ in the above lemma and its proof.

The next lemma is technical.

Lemma 3.3 *Under the conditions of Lemma 3.1, we have the bounds*

$$\|\dot{x}_k\| \leq \kappa_1 \Phi(v_k) \ \textit{and}\ \|\dot{s}_k\| \leq \kappa_2 \Phi(v_k), \quad k \geq 0, \tag{3.35}$$

where κ_1 and κ_2 are positive constants, independent of k.

Proof. Consider the explicit expressions (2.3) of the directions $(\dot{x}_k, \dot{s}_k)$. The bounds (3.26) in Lemma 3.2 imply that the diagonal matrices D_k and S_k^{-1} are bounded above and below, independently of k. Thus all the matrices that occur in the expression (2.3b) are bounded above in norm, independently of k. In particular, to see that M_k^{-1} is bounded above in norm, we remark that $\|M_k^{-1}\|_2 = 1/\lambda_k^{\min}$, where $\lambda_k^{\min} > 0$ is the smallest eigenvalue of M_k and is characterized by

$$\lambda_k^{\min} = \min_{\|y\|_2=1} y^\top M_k y, \quad k \geq 0. \tag{3.36}$$

Let y_k^* be the solution of (3.36), $k \geq 0$. Assuming that $\left(\lambda_k^{\min}\right) = \left(\|M_k^{1/2} y_k^*\|_2\right)$ converges to zero (on a subsequence), as $k \to \infty$, we deduce from the expression of M_k that $D_k A^\top y_k^* \to 0$. Employing the fact that D_k is bounded below away from zero and A has full row rank, this further gives $y_k^* \to 0$ (on a subsequence), which contradicts $\|y_k^*\| = 1$, $k \geq 0$. Thus M_k^{-1} is bounded above in the l_2 norm, and thus, also in the generic norm $\|\cdot\|$, because of (3.21).

The definition of $\Phi(v_k)$ provides

$$\max\{\, \|r_k^p\|, \|r_k^d\|, \|X_k s_k - \mu e\|\} \leq \Phi(v_k), \quad k \geq 0,$$

and the second set of inequalities in (3.35) hold, for some positive constant κ_2 independent of k.

The bound on $\dot{x}_k$ now follows from that on $\dot{s}_k$, (2.3c), and the second sentence of the first paragraph of this proof. □

In what follows, we assume that there is a constant κ independent of k, such that

$$\|\dot{X}_k \dot{s}_k\| \leq \kappa \Phi(v_k)^2, \quad k \geq 0. \tag{3.37}$$

Lemma 3.3 shows that $\kappa := \kappa_1 \kappa_2$ suffices, although we will improve this bound later.

The next lemma shows that the step α_k^{U} determined by the inequalities (2.13) is bounded away from zero, which will then be employed in the subsequent theorem to show convergence of Algorithm 1.

Lemma 3.4 *Under the conditions of Lemma 3.1, let α_k^U be the largest value in $(0,1]$ such that the inequalities (2.13) are satisfied for all $\alpha \in [0, \alpha_k^U]$, where $\theta \in (0,1)$ is chosen such that the conditions (2.13) are achieved at the starting point v_0. Then α_k^U is bounded away from zero. In particular, we have the bound*

$$\alpha_k^U \geq \min\left\{1, \frac{\kappa_3}{[\Phi(v_k)]^2}\right\}, \quad k \geq 0, \tag{3.38}$$

where the constant $\kappa_3 := (1-\theta)\mu\underline{p}/\kappa$ is independent of k.

Proof. From (2.5) and the last n equations of the system (2.2), we deduce

$$X_k(\alpha)s_k(\alpha) = X_k s_k + \alpha(-X_k s_k + \mu e) + \alpha^2 \dot{X}_k \dot{s}_k, \quad k \geq 0. \tag{3.39}$$

Employing (2.12) and the inequality $\dot{X}_k \dot{s}_k \geq -\|\dot{X}_k \dot{s}_k\| e/\underline{p}$ into (3.39), we obtain the bound

$$X_k(\alpha)s_k(\alpha) \geq (1-\alpha)\theta\mu e + \alpha\mu e - \alpha^2 \|\dot{X}_k \dot{s}_k\| e/\underline{p}, \quad \alpha \in [0,1], \quad k \geq 0. \tag{3.40}$$

Substituting (3.37) into (3.40), we deduce

$$X_k(\alpha)s_k(\alpha) \geq (1-\alpha)\theta\mu e + \alpha\mu e - \alpha^2 \kappa [\Phi(v_k)]^2 e/\underline{p}, \quad \alpha \in [0,1], \quad k \geq 0. \tag{3.41}$$

Thus the inequalities (2.13) are satisfied provided that $\alpha \in [0,1]$ and the left-hand side of (3.41) is greater or equal to $\theta\mu e$, or equivalently,

$$\alpha \leq \min\left\{1, \frac{(1-\theta)\mu\underline{p}}{\kappa[\Phi(v_k)]^2}\right\}, \tag{3.42}$$

which further implies (3.38). The bound (3.38) provides that α_k^{U} is bounded away from zero since $\Phi(v_k) \leq \Phi(v_0)$, $k \geq 0$. □

The promised global convergence result for Algorithm 1 is given next. The second part of the theorem investigates the rate at which $\Phi(v_k)$ converges to zero asymptotically.

Theorem 3.5 *Let problems (P) and (D) satisfy the* IPM *conditions. Let $v_k = (x_k, y_k, s_k)$, $k \geq 0$, be the iterates generated by Algorithm 1 when applied to these problems. Then*

$$\Phi(v_k) \to 0, \quad \text{as } k \to \infty, \quad \text{at a global Q-linear rate,} \tag{3.43}$$

which implies that the sequence of iterates v_k converges to $v(\mu)$, our target point on the primal-dual central path.

Moreover,

$$\alpha_k^Q = 1 \ \textit{and} \ \frac{\Phi(v_{k+1})}{\Phi(v_k)^2} \leq \kappa, \ \textit{for all } k \geq k_0, \tag{3.44}$$

where κ is the constant occurring in (3.37) and some $k_0 \geq 0$, and hence, the values $\Phi(v_k)$, $k \geq 0$, converge to zero Q-quadratically. Furthermore,

$$\alpha_k \to 1, \ \textit{as} \ k \to \infty, \tag{3.45}$$

and thus, the asymptotic rate of convergence of (v_k) to $v(\mu)$ is at least Q-superlinear; if $\alpha_k = \alpha_k^Q$ for all k sufficiently large, the rate is Q-quadratic.

Proof. Relations (2.16) and (2.17) imply

$$\Phi(v_{k+1}) \leq \begin{cases} (1-\alpha_k^{\mathrm{G}})\Phi(v_k) + (\alpha_k^{\mathrm{G}})^2\|\dot{X}_k\dot{s}_k\|, & \text{if } \alpha_k^{\mathrm{G}} := \frac{1}{2}\Phi(v_k)/\|\dot{X}_k\dot{s}_k\| \leq \alpha_k^{\mathrm{U}}, \\ (1-\alpha_k^{\mathrm{U}})\Phi(v_k) + (\alpha_k^{\mathrm{U}})^2\|\dot{X}_k\dot{s}_k\|, & \text{otherwise.} \end{cases} \tag{3.46}$$

In the first case, when $\alpha_k^{\mathrm{G}} \leq \alpha_k^{\mathrm{U}}$, we substitute the expression of α_k^{G} into the corresponding inequality in (3.46) and obtain

$$\Phi(v_{k+1}) \leq (1 - \tfrac{1}{2}\alpha_k^{\mathrm{G}})\Phi(v_k). \tag{3.47}$$

From (3.37) and the expression of α_k^{G} in (3.46), we derive the following lower bound on α_k^{G}

$$\alpha_k^{\mathrm{G}} \geq \frac{1}{2\kappa\Phi(v_k)} \geq \frac{1}{2\kappa\Phi(v_0)} := \kappa_4, \quad k \geq 0. \tag{3.48}$$

The assumption that $\alpha_k^{\mathrm{G}} \leq \alpha_k^{\mathrm{U}}$ implies $\alpha_k^{\mathrm{G}} \geq \min\{1, \kappa_4\} := \kappa_5$. It follows from (3.47) that

$$\Phi(v_{k+1}) \leq (1 - \tfrac{1}{2}\kappa_5)\Phi(v_k), \ \text{ where } \ \kappa_5 \in (0, 1]. \tag{3.49}$$

Conversely, for the second case, i. e., $\alpha_k^{\mathrm{G}} > \alpha_k^{\mathrm{U}}$, this and the expression of α_k^{G} in (3.46) imply

$$\frac{\Phi(v_k)}{2\alpha_k^{\mathrm{U}}} \geq \|\dot{X}_k\dot{s}_k\|,$$

which we employ in the second inequality that involves $\Phi(v_{k+1})$ in (3.46), and obtain

$$\Phi(v_{k+1}) \leq (1-\alpha_k^{\mathrm{U}})\Phi(v_k) + (\alpha_k^{\mathrm{U}})^2\frac{\Phi(v_k)}{2\alpha_k^{\mathrm{U}}} = (1 - \tfrac{1}{2}\alpha_k^{\mathrm{U}})\Phi(v_k). \tag{3.50}$$

Now, (3.38) and $\Phi(v_k) \leq \Phi(v_0)$ provide a positive lower bound on α_k^{U}, i. e., $\alpha_k^{\mathrm{U}} \geq \kappa_6 := \min\{1, \kappa_3/[\Phi(v_0)]^2\}$, $k \geq 0$, where κ_3 is a constant, independent of k. It follows from (3.50)

$$\Phi(v_{k+1}) \leq (1 - \tfrac{1}{2}\kappa_6)\Phi(v_k), \ \text{ where } \ \kappa_6 \in (0, 1]. \tag{3.51}$$

We conclude from (3.46), (3.49) and (3.51) that $\Phi(v_k)$ is decreased by a fixed fraction in $(0, 1)$ on each iteration, which implies the global Q-linear rate of convergence of $\Phi(v_k)$ to zero, and (3.43).

The sequence (x_k, s_k) has limit points since it is bounded. Since $\|A^\top y_k + s_k - c\| \to 0$, for any limit point s_* of s_k, we have that $c - s_*$ belongs to the range space of $A^\top$. It follows from A having full row rank that there exists a unique $y_* \in \mathrm{I\!R}^m$ such that $c - s_* = A^\top y_*$. Moreover, employing again the full rank of A, we have that the corresponding subsequence of y_k converges to y_*. Thus y_k also has limit points. The limit (3.43) and the definition of Φ provide that all the limit points of v_k are solutions of the system (1.1), for the same value μ. Since there is only one such point on the central path, the iterates thus converge to a unique limit point.

To show (3.44), we employ the limit (3.43) in (3.38) and in the first inequality in (3.48), and deduce $\alpha_k^{\mathrm{G}} \geq \alpha_k^{\mathrm{U}} = 1$, for all k sufficiently large. Thus, according to (2.17), $\alpha_k^{\mathrm{Q}} = 1$, for all k sufficiently large. From (3.46), we have

$$\Phi(v_{k+1}) \leq \|\dot{X}_k \dot{s}_k\|, \text{ for all } k \text{ sufficiently large,} \tag{3.52}$$

which together with (3.37) completes the proof of (3.44).

Before proving the limit (3.45), note that (3.45) implies that the stepsize computed by Algorithm 1 does not become arbitrarily small. Furthermore, (3.45) and classical results for Newton's method provide that the rate of convergence of (v_k) to $v(\mu)$ is Q-superlinear. If α_k is chosen to be α^Q, at least for all k sufficiently large, then, (3.44) implies that (v_k) converges to $v(\mu)$ Q-quadratically [Theorem 5. 2. 1, [10]].

To establish (3.45), we deduce from (2.11)

$$(1-\alpha_k)\Phi(v_k) - \alpha_k^2\|\dot{X}_k \dot{s}_k\| \leq \Phi(v_{k+1}), \quad k \geq 0. \tag{3.53}$$

It follows from (3.37) and the second set of inequalities in (3.44)

$$(1-\alpha_k)\Phi(v_k) - \alpha_k^2\kappa[\Phi(v_k)]^2 \leq \kappa[\Phi(v_k)^2], \quad k \geq k_0, \tag{3.54}$$

or equivalently, as $\Phi(v_k) > 0$,

$$1-\alpha_k \leq (1+\alpha_k^2)\kappa[\Phi(v_k)]^2, \quad k \geq k_0. \tag{3.55}$$

Further, since $\alpha_k \in (0, 1]$, we obtain

$$0 \leq 1-\alpha_k \leq 2\kappa[\Phi(v_k)]^2, \quad k \geq k_0. \tag{3.56}$$

This, together with (3.43), implies (3.45) and completes the proof of the theorem. □

3.2 An iteration complexity bound for Algorithm 1

We now turn our attention to estimating how many iterations are required by Algorithm 1.

3.2.1 A general bound

Firstly, we give an upper bound on the number of iterations Algorithm 1 takes before $(\Phi(v_k))$ starts converging quadratically to zero. The following notation will be employed

$$q_1 := \frac{1}{2\kappa} \text{ and } q_2 := \sqrt{\frac{(1-\theta)\mu\underline{p}}{\kappa}}, \tag{3.57}$$

where κ is defined by (3.37), θ by (2.19), and $\underline{p}$ by (3.21).

Theorem 3.6 *In addition to the conditions of Theorem 3.5, let ϵ be a given tolerance such that*

$$\epsilon \geq \min\{q_1, q_2\} > 0, \tag{3.58}$$

where q_1 and q_2 are given in (3.57). Then we have

$$\Phi(v_k) \leq \epsilon, \tag{3.59}$$

for all k such that

$$k \geq k_0^\epsilon := \max\left\{0, \left\lceil 2\left(\Phi(v_0) - \epsilon\right) \cdot \max\left\{\frac{1}{q_1}, \frac{\Phi(v_0)}{q_2^2}\right\}\right\rceil\right\}. \tag{3.60}$$

(If $k_0^\epsilon = 0$, then Algorithm 1 terminates, and the second part of the theorem concerning quadratic convergence is irrelevant.) In particular, we have

$$\Phi(v_k) \leq \min\{q_1, q_2\}, \tag{3.61}$$

for all k such that

$$k \geq k_0 := \max\left\{0, \left\lceil 2\left(\Phi(v_0) - \min\{q_1, q_2\}\right) \cdot \max\left\{\frac{1}{q_1}, \frac{\Phi(v_0)}{q_2^2}\right\}\right\rceil\right\}. \tag{3.62}$$

Furthermore,

$$\alpha_k^G \geq \alpha_k^U = 1, \quad k \geq k_0, \tag{3.63}$$

and (3.44) holds, and thus $(\Phi(v_k))$ converges Q-quadratically to zero for $k \geq k_0$.

Proof. Assume that $\Phi(v_0) > \epsilon$, and thus $k_0^\epsilon > 0$. To prove (3.59) holds if (3.60) is satisfied, we argue exactly as in the proof of Theorem 3.5. Supposing that $\alpha_k^{\mathrm{G}} \leq \alpha_k^{\mathrm{U}}$, we substitute the first inequality in (3.48) into (3.47), and deduce

$$\Phi(v_{k+1}) \leq \Phi(v_k) - \frac{1}{2}q_1, \tag{3.64}$$

for each k such that $\Phi(v_k) > q_1$.

Similarly, if $\alpha_k^{\mathrm{G}} > \alpha_k^{\mathrm{U}}$, (3.50) and (3.38) provide

$$\Phi(v_{k+1}) \leq \Phi(v_k) - \frac{q_2^2}{2\Phi(v_0)}, \tag{3.65}$$

for each k such that $\Phi(v_k) > q_2$.

Thus, if $\Phi(v_0) > \epsilon$, and since $\epsilon \geq \min\{q_1, q_2\}$, we have

$$\Phi(v_{k+1}) \leq \Phi(v_k) - \frac{1}{2}\min\left\{q_1, \frac{q_2^2}{\Phi(v_0)}\right\}, \tag{3.66}$$

for all consecutive $k \geq 0$ such that $\Phi(v_k) > \epsilon$, which implies

$$\Phi(v_k) \leq \Phi(v_0) - \frac{k}{2}\min\left\{q_1, \frac{q_2^2}{\Phi(v_0)}\right\}, \tag{3.67}$$

for all $k \geq 0$ while $\Phi(v_k) > \epsilon$. It follows that (3.59) holds for all k satisfying (3.60). Letting $\epsilon := \min\{q_1, q_2\}$, we obtain (3.61) for all k achieving (3.62).

To see that (3.61) implies (3.63), recall the bound on α_k^{U} in (3.38), as well as the definition of α_k^{G} in (3.46) and the bound (3.37). The quadratic convergence of $(\Phi(v_k))$ follows from (3.63), as shown in the proof of (3.44) in Theorem 3.5. □

The next theorem evaluates the iteration complexity of Algorithm 1 for generating v_k with $\Phi(v_k)$ arbitrarily small.

Theorem 3.7 *If, in addition to the conditions of Theorem 3.5, the positive tolerance ϵ satisfies $\epsilon < \min\{q_1, q_2\}$, then Algorithm 1 will generate a point v_k such that $\Phi(v_k) \leq \epsilon$ for all k satisfying*

$$k \geq k_1 := k_0 + \left\lceil \frac{1}{\log 2} \log \left(\frac{\log\left(\epsilon(2q_1)^{-1}\right)}{\log r} \right) \right\rceil, \tag{3.68}$$

where k_0 is defined by (3.62) *and $r := \min\{1/2, \sqrt{(1-\theta)\mu\underline{p}\kappa}\}$.*

Proof. It follows from (3.44) and Theorem 3.6 that

$$\Phi(v_{k+1}) \leq \kappa \Phi(v_k)^2, \qquad \text{for all } \ k \geq k_0, \tag{3.69}$$

where k_0 is defined in (3.62). Thus we obtain recurrsively

$$\Phi(v_k) \leq \frac{1}{\kappa} (\kappa \Phi(v_{k_0}))^{2^{k-k_0}}, \quad k \geq k_0. \tag{3.70}$$

Since relations (3.61) and (3.62) imply $\Phi(v_{k_0}) \leq \min\{q_1, q_2\}$, (3.57) and (3.70) further give

$$\Phi(v_k) \leq \frac{1}{\kappa} \left(\min \left\{ \frac{1}{2}, \sqrt{(1-\theta)\mu\underline{p}\kappa} \right\} \right)^{2^{k-k_0}}, \quad k \geq k_0. \tag{3.71}$$

Thus $\Phi(v_k) \leq \epsilon$ provided

$$r^{2^{k-k_0}} \leq \epsilon\kappa, \tag{3.72}$$

where r is defined in the statement of the theorem. We remark that $\epsilon\kappa < 1$ since $\epsilon < \min\{q_1, q_2\}$. Passing to the logarithm in (3.72), we deduce

$$-2^{k-k_0} \log r \geq -\log(\epsilon\kappa), \tag{3.73}$$

or equivalently,

$$k - k_0 \geq \frac{1}{\log 2} \log \left(\frac{\log(\epsilon\kappa)}{\log r} \right), \tag{3.74}$$

which implies (3.68). □

3.2.2 A precise value for κ

Next, we deduce new values for the bound κ in (3.37) that are not only independent of k, as provided by Lemma 3.3, but depend explicitly on the problem data and on the parameters of the algorithm.

Lemma 3.8 *Let problems (PD) satisfy the* IPM *conditions and let Algorithm 1 be applied to these problems, where the l_2 norm is employed in the definition* (2.7) *of Φ. Let Φ_2 denote the resulting merit function of Algorithm 1. Then*

$$\|\dot{X}_k \dot{s}_k\|_2 \leq \frac{[\Phi_2(v_k)]^2}{\theta\mu} \left(1 + 2C_0 \max\left\{\text{res}^p, \text{res}^d\right\}\right)^2, \quad k \geq 0, \tag{3.75}$$

where C_0 is any finite componentwise upper bound on the sequences (x_k) and (s_k), i. e.,

$$\max_{k\geq 0} \max_{i=\overline{1,n}} \{x_k^i, \, s_k^i\} \leq C_0, \tag{3.76}$$

where

$$\text{res}^p := \begin{cases} \|x_0 - \hat{x}\|_2/\|r_0^p\|_2, & \textit{if } r_0^p = Ax_0 - b \neq 0, \\ 0, & \textit{otherwise}, \end{cases} \tag{3.77}$$

and

$$\text{res}^d := \begin{cases} \|s_0 - \hat{s}\|_2/\|r_0^d\|_2, & \textit{if } r_0^d = A^\top y_0 + s_0 - c \neq 0, \\ 0, & \textit{otherwise}, \end{cases} \tag{3.78}$$

where $(\hat{x}, \hat{s})$ corresponds to an arbitrary primal-dual strictly feasible point $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ of (PD).

Proof. Let $k \geq 0$. Multiplying the explicit expressions (2.3b) and (2.3c) of the search directions $\dot{s}_k$ and $\dot{x}_k$ by $D_k := X_k^{1/2} S_k^{-1/2}$ and D_k^{-1}, respectively, we deduce

$$D_k\dot{s}_k = D_k A^\top M_k^{-1} r_k^p - [I - D_k A^\top M_k^{-1} A D_k] D_k r_k^d - D_k A^\top M_k^{-1} A D_k X_k^{-1/2} S_k^{-1/2}(X_k s_k - \mu e), \tag{3.79a}$$

$$D_k^{-1}\dot{x}_k = -D_k\dot{s}_k - X_k^{-1/2} S_k^{-1/2}(X_k s_k - \mu e), \tag{3.79b}$$

where $M_k = AD_k^2 A^\top$, $D_k = X_k^{1/2} S_k^{-1/2}$, $r_k^p = Ax_k - b$ and $r_k^d = A^\top y_k + s_k - c$. Next, we express $D_k\dot{s}_k$ and $D_k^{-1}\dot{x}_k$ in terms of the orthogonal projection matrices P_k and $I - P_k$ onto the range space of $D_k A^\top$ and the null space of AD_k, respectively, where $P_k := D_k A^\top M_k^{-1} A D_k$. It follows from Lemma 3.1, in particular, from (3.25), that $r_k^p = A(x_k - u_k)$, $k \geq 0$, where (u_k) is defined recursively by (3.22a). It follows from (3.79)

$$D_k\dot{s}_k = P_k D_k^{-1}(x_k - u_k) - (I - P_k) D_k r_k^d - P_k X_k^{-1/2} S_k^{-1/2}(X_k s_k - \mu e), \tag{3.80a}$$

$$D_k^{-1}\dot{x}_k = -P_k D_k^{-1}(x_k - u_k) + (I - P_k) D_k r_k^d - (I - P_k) X_k^{-1/2} S_k^{-1/2}(X_k s_k - \mu e). \tag{3.80b}$$

Again from Lemma 3.1, $r_k^d = A^\top(y_k - t_k) + (s_k - w_k)$, where (w_k) and (t_k) are given in (3.22c) and (3.22b). Using orthogonal projection properties, we deduce that $(I - P_k)D_k r_k^d = (I - P_k)D_k(s_k - w_k)$. Furthermore, employing the fact that $\|P_k\|_2 \leq 1$ and $\|I - P_k\|_2 \leq 1$, we obtain

$$\max\left\{\|D_k\dot{s}_k\|_2, \|D_k^{-1}\dot{x}_k\|_2\right\} \leq \|D_k^{-1}(x_k - u_k)\|_2 + \|D_k(s_k - w_k)\|_2 + \|X_k^{-1/2} S_k^{-1/2}(X_k s_k - \mu e)\|_2. \tag{3.81}$$

It follows from the expression of D_k and $X_k s_k \geq \theta\mu e$,

$$\max\left\{\|D_k\dot{s}_k\|_2, \|D_k^{-1}\dot{x}_k\|_2\right\} \leq \frac{1}{\sqrt{\theta\mu}}\left[\|S_k(x_k - u_k)\|_2 + \|X_k(s_k - w_k)\|_2 + \|X_k s_k - \mu e\|_2\right]. \tag{3.82}$$

Recalling (3.23) from Lemma 3.1, we deduce

$$\max\left\{\|D_k\dot{s}_k\|_2, \|D_k^{-1}\dot{x}_k\|_2\right\} \leq \frac{1}{\sqrt{\theta\mu}}\left[\nu_k\|S_k\|_2 \cdot \|x_0 - \hat{x}\|_2 + \nu_k\|X_k\|_2 \cdot \|s_0 - \hat{s}\|_2 + \|X_k s_k - \mu e\|_2\right], \tag{3.83}$$

where ν_k is defined by (3.24). From (2.8), we obtain $r_k^p = \nu_k r_0^p$ and $r_k^d = \nu_k r_0^d$. Thus, if $r_0^p \neq 0$ and $r_0^d \neq 0$, then, employing $\max\{\|r_k^p\|_2, \|r_k^d\|_2, \|X_k s_k - \mu e\|_2\} \leq \Phi_2(v_k)$ and properties of the matrix norm provide, (3.83) further becomes

$$\max\left\{\|D_k\dot{s}_k\|_2, \|D_k^{-1}\dot{x}_k\|_2\right\} \leq \frac{\Phi_2(v_k)}{\sqrt{\theta\mu}}\left[1 + \max_{i=\overline{1,n}}\{x_k^i, s_k^i\}\left(\frac{\|x_0 - \hat{x}\|}{\|r_0^p\|} + \frac{\|s_0 - \hat{s}\|}{\|r_0^d\|}\right)\right]. \tag{3.84}$$

If $r_0^p = 0$, then x_0 is strictly feasible for (P) and we may let $x_0 = \hat{x}$ since $\hat{x}$ is an arbitrary strictly feasible primal point. Therefore the corresponding term on the right-hand side of (3.83) vanishes, and hence the definition (3.77) of res^p. We proceed similarly in the case when $r_0^d = 0$.

The bound (3.75) now follows from $\|\dot{X}_k \dot{s}_k\|_2 \leq \|D_k^{-1}\dot{x}_k\|_2 \cdot \|D_k \dot{s}_k\|_2$ and the componentwise boundedness of the sequences (x_k) and (s_k) (see the first part of Lemma 3.2). □

Next we develop the bound (3.75) to further clarify its dependence on problem data.

Corollary 3.9 *Assume the conditions of Lemma 3.8 hold, and let $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ be an arbitrary strictly feasible point of (PD).*

If the starting point $v_0 = (x_0, y_0, s_0)$ of Algorithm 1 satisfies

$$x_0 = \hat{x}, \quad y_0 = \hat{y}, \quad s_0 = \hat{s}, \tag{3.85}$$

(in other words, v_0 is strictly feasible for (PD)), then

$$\|\dot{X}_k \dot{s}_k\|_2 \leq \frac{1}{\theta\mu}[\Phi_2(v_k)]^2, \quad k \geq 0. \tag{3.86}$$

More generally, if $v_0 = (x_0, y_0, s_0)$ satisfies

$$x_0 \geq \hat{x} \text{ and } s_0 \geq \hat{s}, \tag{3.87}$$

then

$$\|\dot{X}_k \dot{s}_k\|_2 \leq \frac{[\Phi_2(v_k)]^2}{\theta\mu}\left(1 + 4\frac{n(\Phi_2(v_0)+\mu)}{\underline{m}} \max\left\{res^p,\, res^d\right\}\right)^2, \quad k \geq 0, \tag{3.88}$$

where $\underline{m} = \min\{\hat{x}^i, \hat{s}^i : i = \overline{1,n}\}$, and where res^p and res^d are defined by (3.77) *and* (3.78), *respectively.*

Proof. The bound (3.86) follows straightforwardly from (3.75), (3.77) and (3.78).

To show (3.88), recall the specific upper bound C_2 on the components of the sequences (x_k) and (s_k) given in (3.26). It follows from (3.87) that $\underline{m} = \min\{\hat{x}^i, \hat{s}^i : i = \overline{1,n}\}$. Moreover, the constant L that occurs in (3.26) now has the value $L = x_0^\top s_0 + \sqrt{n}\Phi_2(v_0) + n\mu$, where we also employed $\underline{p} = 1$ for the l_2 norm. The inequality (3.33) for $k = 0$ further provides

$$L \leq 2n[\Phi_2(v_0) + \mu].$$

The inequality (3.88) follows from (3.75), (3.26) and the above estimates of the value of L and $\underline{m}$. □

The next corollary allows for other norms besides l_2, to be employed in the merit function Φ.

Corollary 3.10 *Let problems (PD) satisfy the* IPM *conditions. Apply Algorithm 1 to these problems where any norm satisfying* (3.21) *may be employed in the definition* (2.7) *of* Φ.

If the starting point $v_0 = (x_0, y_0, s_0)$ of Algorithm 1 is strictly feasible for (PD), then

$$\|\dot{X}_k \dot{s}_k\| \leq \frac{\overline{p}}{\underline{p}^2} \cdot \frac{[\Phi(v_k)]^2}{\theta\mu}, \quad k \geq 0, \tag{3.89}$$

where $\overline{p}$ and $\underline{p}$ occur in (3.21).

More generally, if $v_0 = (x_0, y_0, s_0)$ satisfies (3.87), *where $(\hat{x}, \hat{s})$ corresponds to any primal-dual strictly feasible point $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ of (PD), then*

$$\|\dot{X}_k \dot{s}_k\| \leq \frac{\overline{p}}{\underline{p}^2} \cdot \frac{[\Phi(v_k)]^2}{\theta\mu}\left(1 + 4\frac{n(\Phi_2(v_0)+\mu)}{\underline{m}} \max\left\{res^p,\, res^d\right\}\right)^2, \quad k \geq 0, \tag{3.90}$$

where $\overline{p}$ *and* $\underline{p}$ *are defined within* (3.21)*,* Φ_2 *denotes* Φ *defined in terms of the* l_2 *norm, where* $\underline{m} = \min\{\hat{x}^i, \hat{s}^i : i = \overline{1,n}\}$*, and where* res^p *and* res^d *are given in* (3.77) *and* (3.78)*, respectively.*

Proof. Relation (3.21) provides

$$\|\dot{X}_k \dot{s}_k\| \leq \overline{p} \|\dot{X}_k \dot{s}_k\|_2, \text{ and } \Phi_2(v_k) \leq \frac{1}{\underline{p}} \Phi(v_k), \quad k \geq 0.$$

The bounds (3.89) and (3.90) now follow from Corollary 3.9. □

In the above corollary, the quantities on the right-hand side of (3.90) that depend on the l_2 norm are independent of k, and thus, they need not be expressed in terms of the general norm $\|\cdot\|$.

Similarly to Corollary 3.10, a variant of Lemma 3.8 can be given for the case when the general norm $\|\cdot\|$ is employed in defining Φ.

We remark that if (3.85) holds, the iterates v_k of Algorithm 1 remain primal-dual strictly feasible for $k \geq 0$, and Algorithm 1 attempts to find a well-centred point starting from a possibly badly centred one. The complexity of this case is important for the overall iteration complexity of Algorithm 2, to be studied in the next section.

When we start Algorithm 1 from an infeasible point with respect to the primal and/or dual equality constraints, the quantities res^p and res^d in (3.77) and (3.78) represent a measure of "the distance to strict feasibility" of the starting point v_0, and they are thus, a natural occurence in the iteration complexity bounds below. For theoretical purposes, we may set $(\hat{x}, \hat{s}) := (x(\mu), s(\mu))$ in the results above. In practice, however, since $(x(\mu), s(\mu))$ is unknown, a strictly feasible point, possibly badly centred, may be known and used in the above results to estimate the iteration complexity of Algorithm 1 (again, this case may be relevant for Algorithm 2 in the next section).

3.2.3 A specific iteration complexity bound for Algorithm 1

We now substitute the values of κ implied by the bounds (3.89) and (3.90) in Corollary 3.10 into the iteration complexity results at the beginning of this subsection, to deduce the following corollary.

Corollary 3.11 *Under the conditions and notations of Corollary 3.10, let*

$$q_0 := \begin{cases} 1, & \text{if } v_0 = (x_0, y_0, s_0) \text{ is strictly feasible for (PD)}, \\ 1 + 4n(\Phi_2(v_0) + \mu)\underline{m}^{-1} \max\{res^p, res^d\}, & \text{if } v_0 \text{ satisfies (3.87)}. \end{cases} \tag{3.91}$$

Let ϵ *be a positive tolerance. Then the following results hold*

i) If $\epsilon \geq \underline{p} q_0^{-1} \mu \min\{\theta/(2p_0 q_0), \sqrt{\theta(1-\theta)/p_0}\} := \epsilon_{min}$*, where* $p_0 := \overline{p}/\underline{p}$*, then Algorithm 1 generates an iterate* v_k *with* $\Phi(v_k) \leq \epsilon$*, for all* k *such that*

$$k \geq k_0^\epsilon := \max\left\{0, \left\lceil 2(\Phi(v_0) - \epsilon) \cdot \frac{p_0 q_0^2}{\theta \mu \underline{p}} \max\left\{2, \frac{\Phi(v_0)}{(1-\theta)\mu \underline{p}}\right\}\right\rceil\right\}. \tag{3.92}$$

ii) Let $k_0^q := k_0^\epsilon$ *for* $\epsilon = \epsilon_{min}$*. When* $\epsilon < \epsilon_{min}$*, then Algorithm 1 generates* v_k *such that* $\Phi(v_k) \leq \epsilon$ *for all* k *such that*

$$k \geq k_0^q + \left\lceil \frac{1}{\log 2} \log\left(\frac{\log(\epsilon \cdot p_0 q_0 (\theta \mu \underline{p})^{-1})}{\log r}\right)\right\rceil, \tag{3.93}$$

where $r := \min\{1/2, q_0 \sqrt{(1-\theta)p_0/\theta}\}$*.*

Let us consider in more detail the results in Corollary 3.11 when v_0 is primal-dual strictly feasible (and thus $q_0 = 1$). Then, in the case when Φ is defined using the l_2 norm, the above upper bounds on the number of iterations required by Algorithm 1 can actually be computed, as they depend only on $(\Phi(v_0), \theta, \mu)$ and ϵ (note that $q_0 = p_0 = \underline{p} = 1$ in (3.92) and (3.93)). The same is true when employing the l_1 or l_∞ norms, but the bounds also depend *explicitly* on the problem dimension n; this dependence is polynomial. For example, in the case of the l_1 norm, where the optimal choices in (3.21) are $\underline{p} = 1$ and $\overline{p} = \sqrt{n}$, the value of the bound (3.92) for $\epsilon = \epsilon_{min} = \mathcal{O}(\mu/\sqrt{n})$ implies that at most $2[\theta(1-\theta)\mu^2]^{-1}\sqrt{n}\Phi(v_0)\max\{2(1-\theta)\mu, \Phi(v_0)\}$ iterations of Algorithm 1 are required for the quadratic rate of convergence of $(\Phi(v_k))$ to occur. Similarly, when the l_∞ norm is employed in Φ, we have $\underline{p} = 1/\sqrt{n}$, $\overline{p} = 1$ and $\epsilon_{min} = \mathcal{O}(\mu/n)$. Then, the bound (3.92) implies that after at most $2[\theta(1-\theta)\mu^2]^{-1}n\Phi(v_0)\max\{2(1-\theta)\mu, \sqrt{n}\Phi(v_0)\}$ iterations of Algorithm 1, $(\Phi(v_k))$ converges quadratically. In our numerical experiments (see Section 8), where Φ is defined in terms of the l_∞ norm, this bound is highly pessimistic, and the quadratic convergence property almost always occurs within a very modest number of iterations. Independently of the choice of norm, the "asymptotic" bound (3.93) is of the form $\mathcal{O}(\log(\log(1/\epsilon)))$.

4 Higher-order Newton methods

It is common to use a predictor-corrector step to try to encourage faster convergence. Suppose that we have computed $\dot{v}_k$ to satisfy (2.2), and now we compute

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \ddot{x}_k \\ \ddot{y}_k \\ \ddot{s}_k \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ \dot{X}_k \dot{S}_k e \end{pmatrix}. \tag{4.1}$$

In this case, it is appropriate to replace the linesearch by a search along a quadratic arc of the form

$$v_k^{\mathrm{C}}(\alpha) \stackrel{\text{def}}{=} \begin{pmatrix} x_k^{\mathrm{C}}(\alpha) \\ y_k^{\mathrm{C}}(\alpha) \\ s_k^{\mathrm{C}}(\alpha) \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \\ s_k \end{pmatrix} + \alpha \begin{pmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{s}_k \end{pmatrix} + \alpha^2 \begin{pmatrix} \ddot{x}_k \\ \ddot{y}_k \\ \ddot{s}_k \end{pmatrix}. \tag{4.2}$$

This then ensures that, as before,

$$r(v_k^{\mathrm{C}}(\alpha)) = (1-\alpha) r(v_k), \tag{4.3}$$

and gives the appropriate variations

$$X_k^{\mathrm{C}}(\alpha) s_k^{\mathrm{C}}(\alpha) - \mu e = (1-\alpha)(X_k s_k - \mu e) + \alpha^3(\ddot{X}_k \dot{s}_k + \dot{X}_k \ddot{s}_k) + \alpha^4 \ddot{X}_k \ddot{s}_k, \tag{4.4}$$

and

$$\Phi(v_k^{\mathrm{C}}(\alpha)) \leq (1-\alpha)\Phi(v_k) + \alpha^3 \|\ddot{X}_k \dot{s}_k + \dot{X}_k \ddot{s}_k\| + \alpha^4 \|\ddot{X}_k \ddot{s}_k\|, \quad \alpha \in [0,1], \tag{4.5}$$

of (2.9)–(2.10). Furthermore, the corresponding lower bound to (3.53) is

$$\Phi(v_k^{\mathrm{C}}(\alpha)) \geq (1-\alpha)\Phi(v_k) - \alpha^3 \|\ddot{X}_k \dot{s}_k + \dot{X}_k \ddot{s}_k\| - \alpha^4 \|\ddot{X}_k \ddot{s}_k\|, \quad \alpha \in [0,1]. \tag{4.6}$$

It is then straightforward to derive convergence and complexity analyses similar to those of Algorithm 1.

5 On satisfying the assumptions

5.1 Some disadvantages of the assumptions

Our convergence analysis in Section 3 indicates that Algorithm 1 is reliable and fast, *provided the* IPM *conditions are satisfied by (PD)*. Our numerical experience with a Fortran 95 implementation of this algorithm on NETLIB problems [27] without preprocessing of the test problems except so as to ensure that the matrix A has full row rank (i. e., it satisfies assumption A1 of the IPM conditions) confirms our theoretical results. Example 1 illustrates the satisfactory behaviour of Algorithm 1 that we observe when the IPM conditions hold. In our implementation (and the tables of results below), we use the l_∞ norm in the definition (2.7) of the merit function "merit", and of the residuals "p-feas", "d-feas" and "com-slk" of the primal, dual and centrality equations of the system (1.1), respectively. The "time" column is the total CPU time in seconds.

Example 1. When applying our implementation of Algorithm 1 to problem AFIRO from the NETLIB test set, we obtain the following output.

```
Iter  p-feas   d-feas   com-slk  merit    stepsize  mu       time
   0  3.9E+01  6.1E+01  5.7E+03  5.8E+03    --      1.0E+02  0.00
   1  3.9E+01  6.0E+01  5.6E+03  5.8E+03  9.20E-03  1.0E+02  0.00
   2  3.8E+01  5.9E+01  5.5E+03  5.7E+03  1.79E-02  1.0E+02  0.00
   3  3.7E+01  5.8E+01  5.5E+03  5.6E+03  1.64E-02  1.0E+02  0.00
   4  3.6E+01  5.6E+01  5.2E+03  5.4E+03  4.20E-02  1.0E+02  0.00
   5  3.0E+01  4.8E+01  4.4E+03  4.5E+03  1.47E-01  1.0E+02  0.00
   6  2.0E+01  3.2E+01  2.7E+03  2.7E+03  3.36E-01  1.0E+02  0.00
   7  6.4E+00  1.0E+01  1.1E+03  1.2E+03  6.83E-01  1.0E+02  0.00
   8  9.9E-01  1.5E+00  1.6E+02  1.6E+02  8.45E-01  1.0E+02  0.00
   9  4.0E-03  6.2E-03  1.7E+00  1.7E+00  9.96E-01  1.0E+02  0.00

================ strictly feasible point found =================

  10  4.5E-14  0.0E+00  2.9E-04  2.9E-04  1.00E+00  1.0E+02  0.00
  11  5.7E-14  0.0E+00  1.2E-11  1.2E-11  1.00E+00  1.0E+02  0.00

=============== well-centred interior point found ===============
```

A strictly feasible point is found after 9 iterations, and a point satisfying (1.1) with $\mu = 100$, within 11 iterations of the method. The numerical results show that assumption A2 of the IPM conditions is verified by this problem. ◇

Frequently, however, we observed disappointing numerical performance of our implementation of Algorithm 1, as we exemplify next.

Example 2. Our implementation applied to the primal-dual feasible NETLIB problem 25FV47 generates the output below.

```
Iter  p-feas   d-feas   com-slk  merit    stepsize  mu        time
   0  5.7E+02  2.2E+03  4.0E+04  5.7E+04    --      2.2E-01   0.02
   1  5.7E+02  2.2E+03  4.0E+04  5.7E+04  5.37E-03  2.2E-01   0.07
   2  5.6E+02  2.2E+03  4.0E+04  5.7E+04  3.92E-03  2.2E-01   0.12
   3  5.6E+02  2.2E+03  3.9E+04  5.6E+04  8.67E-03  2.2E-01   0.16
...................................
 998  4.3E-07  1.3E-07  3.7E-03  3.7E-03  2.14E-03  2.2E-01  46.57
 999  4.3E-07  1.3E-07  3.7E-03  3.7E-03  2.14E-03  2.2E-01  46.61
1000  4.3E-07  1.3E-07  3.7E-03  3.7E-03  2.14E-03  2.2E-01  46.66
```

Thus progress is extremely slow and the stepsize does not increase to 1. Moreover, the algorithm terminates since it has reached the maximum number of iterations allowed ($k = 1000$), without generating the required approximate solution of (1.1) for $\mu = 0.22$ within the specified accuracy $\epsilon = 10^{-6}$. An approximate primal-dual feasible point is obtained, but not a strictly feasible one, as some of the dual slack variables are very close to their bounds. The primal variables corresponding to these dual small components increase rapidly in magnitude as the algorithm progresses, causing severe ill-conditioning. Such behaviour of our implementation occurs irrespectively of the choice of the target value μ. $\diamond$

We believe the inefficient behaviour of Algorithm 1 on problem 25FV47 is due to this example not satisfying assumption A2 of the IPM conditions. We will attempt to justify this claim in the remainder of this subsection, and convince the reader that the increasing ill-conditioning is not due to a poor implementation of Algorithm 1. Firstly, we explain from a theoretical point of view what happens when (3.20) does not hold.

Let $\mathcal{F}_P$, $\mathcal{F}_D$, $\mathcal{F}_{PD}$ denote the primal, dual and primal-dual feasible set of (PD), respectively; equivalently,

$$\begin{gathered}\mathcal{F}_P := \{x \in \mathbb{R}^n : \ Ax = b, \ x \geq 0\}, \\ \mathcal{F}_D := \{(y,s) \in \mathbb{R}^m \times \mathbb{R}^n : \ A^\top y + s = c, \ s \geq 0\}, \quad \mathcal{F}_{PD} := \mathcal{F}_P \times \mathcal{F}_D.\end{gathered} \tag{5.1}$$

Assume that $\mathcal{F}_{PD}$ is nonempty, and thus (PD) has solutions. Then (PD) does not admit strictly feasible points if and only if there exists $i \in \{1, \ldots, n\}$ such that $x_i = 0$, for all $x \in \mathcal{F}_P$, or $s_i = 0$ for all s such that $(y, s) \in \mathcal{F}_D$ for some y. In other words, an inequality constraint of (P) or (D) holds as an equality (is active) at all the feasible points of (P) or (D), respectively. We call such constraints *implicit equalities* of (P) or (D) [31].

In view of [7, 36], the presence of implicit equalities in $\mathcal{F}_P$ is equivalent to the set of optimal dual slacks s being unbounded. Similarly, $\mathcal{F}_D$ has implicit equalities if and only if the primal solution set is unbounded. We recall that in Example 2, some of the dual slack variables become very close to zero, while the corresponding primal components become very large, suggesting the presence of implicit equalities in the dual problem. More generally, as the system (1.1) has a solution provided the IPM conditions hold, it is perhaps not so suprising that Algorithm 1 seems unable to compute a well-centred point for Example 2. See also the explanation in the introduction for the unfavourable effect of implicit equalities that relates them to the problem (1.2).

Let us look more closely to the following simple example that is more amenable to analysis.

<u>Example 3.</u> Consider the problem of finding a feasible point of

$$x = 0 \ \text{ and } \ x \geq 0, \tag{5.2}$$

and the corresponding dual

$$y + s = 0 \ \text{ and } \ s \geq 0, \tag{5.3}$$

where $(x, y, s) \in \mathbb{R}^3$, and for which $x \geq 0$ is an implicit equality.

Our implementation of Algorithm 1 applied to (5.2) and (5.3) performs equally badly as it did for Example 2, despite the simplicity of the problem.

Iter	p-feas	d-feas	com-slk	merit	stepsize	mu	time
0	1.0E+00	1.0E+00	0.0E+00	2.0E+00	--	1.0E+00	0.00
1	5.0E-05	5.0E-05	1.0E-00	1.0E+00	1.00E-00	1.0E+00	0.00
2	2.5E-05	2.5E-05	7.5E-01	7.5E-01	5.00E-01	1.0E+00	0.00
3	1.6E-05	1.6E-05	6.1E-01	6.1E-01	3.75E-01	1.0E+00	0.00
......							
998	5.5E-10	2.4E-07	4.0E-03	4.0E-03	1.99E-03	1.0E+00	0.18
999	5.5E-10	2.4E-07	4.0E-03	4.0E-03	1.99E-03	1.0E+00	0.18
1000	5.5E-10	2.4E-07	4.0E-03	4.0E-03	1.98E-03	1.0E+00	0.18

In exact arithmetic, using the notation of Section 2 and 3, the Newton direction defined in (2.2) from any $(x_k, s_k) > 0$ satisfies

$$\dot{x}_k = -x_k \text{ and } \dot{s}_k = \frac{\mu}{x_k}, \quad k \geq 0. \tag{5.4}$$

Thus

$$\dot{x}_k \dot{s}_k = -\mu, \quad k \geq 0, \tag{5.5}$$

and $\kappa := \mu$ in (3.37) for this problem. Relation (5.5) also provides

$$x_k(\alpha)s_k(\alpha) - \theta\mu = (1-\alpha)(x_k s_k - \theta\mu) + \alpha(1-\theta)\mu - \alpha^2\mu,$$

which implies

$$1 - \theta \leq \alpha_k^{\text{U}} < 1, \quad k \geq 0. \tag{5.6}$$

It follows that a stepsize α_k of 1 cannot be taken by Algorithm 1 for any $k \geq 0$, when applied to this problem.

From (2.17) and (5.5), we have $\alpha_k^{\text{G}} = \Phi(v_k)/(2\mu)$. Thus if

$$\Phi(v_k) \geq \underline{\Phi} > 0, \quad k \geq 0, \tag{5.7}$$

then α_k^{G} is bounded below and away from zero. The same is true for α_k^{U} due to (5.6). Employing (5.5) again, we can now show, using similar arguments to those in the first part of the proof of Theorem 3.5, that

$$\Phi(v_{k+1}) \leq \frac{1}{2}(1-\theta)\Phi(v_k) \text{ or } \Phi(v_{k+1}) \leq \left(1 - \min\left\{1, \frac{\underline{\Phi}}{4\mu}\right\}\right)\Phi(v_k), \quad k \geq 0,$$

and therefore, $\Phi(v_k)$ converges Q-linearly to zero, which is a contradiction with our assumption (5.7). It follows that α_k^{G} cannot be bounded away from zero, and thus, $\Phi(v_k)$ and α_k^{G} converge to zero at least on a subsequence. The latter limit implies that the stepsize α_k of Algorithm 1 may become arbitrarily small on a subsequence $\mathcal{K}$ of iteration indices k, while the former gives that, on the same subsequence $k \in \mathcal{K}$, we have $x_k \to 0$ and $x_k s_k \to \mu$. It follows that $s_k \to \infty$, $k \to \infty$, $k \in \mathcal{K}$, which explains the occurence of increasing ill-conditioning in the numerics for problem (5.2). ◇

The simplicity of Example 3 allowed us to analyze it in exact arithmetic. For a general LP problem, however, we found it extremely difficult to ensure theoretically that whenever we have a variable that goes to zero while its dual becomes very large, we have an implicit equality. Moreover, from a numerical point of view, the shortening stepsizes and the ill-conditioning that occurs renders Algorithm 1 almost hopeless.

These issues motivate us to consider, in what follows, a sequence of relaxations of our original problem, each of which satisfies the IPM conditions, and is thus amenable to Algorithm 1. We present theoretical results that ensure that our modified algorithm, Algorithm 2, is convergent, and numerical experiments that show our approach is successful in practice, overcoming the numerical deficiencies highlighted in Examples 2 and 3 and in our remarks above. Algorithm 2 will also be able to find the implicit equalities of the original problems (PD), which can then be removed as detailed in Section 6.2.1.1.

When the IPM conditions do not hold, it may be due to (P) or/and (D) being infeasible (and thus, having no solutions). Such instances are not uncommon in practice, and Algorithm 2 attempts to detect this unfortunate case.

5.2 Overcoming the assumptions by controlled perturbations

For the remainder of this section, we assume that the matrix A has full row rank, which is assumption A1 of the IPM conditions. In practice, this is not a serious restriction, since simple preprocessing can usually identify and rectify rank deficiency (see Section 8).

Recall that the system of linear equalities and inequalities for which we want to find a point situated well within its relative interior is

$$\begin{cases} Ax = b, \\ A^T y + s = c, \\ x \geq 0, \ \ s \geq 0. \end{cases} \tag{5.8}$$

Consider the following relaxation of this system

$$\begin{cases} Ax = b, \\ A^T y + s = c, \\ x \geq -\lambda, \ \ s \geq -\gamma, \end{cases} \tag{5.9}$$

where $\lambda \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}^n$ are fixed parameters such that $\lambda \geq 0$ and $\gamma \geq 0$. We let $\Lambda := \text{diag}(\lambda)$ and $\Gamma := \text{diag}(\gamma)$. The following lemma holds.

Lemma 5.1 *For any $\lambda \geq 0$, $\gamma \geq 0$, and any scalar $\mu > 0$, the system*

$$\begin{cases} Ax = b, \\ A^T y + s = c, \\ (X + \Lambda)(S + \Gamma)e = \mu e, \\ x > -\lambda, \ \ s > -\gamma, \end{cases} \tag{5.10}$$

has a unique solution $(x(\delta, \mu), y(\delta, \mu), s(\delta, \mu))$, where $\delta := (\lambda, \gamma)$, provided that (5.9) admits strictly feasible points.

Proof. Letting

$$p := x + \lambda \ \text{ and } \ q := s + \gamma, \tag{5.11}$$

the system (5.10) is equivalent to

$$\begin{cases} Ap = b + A\lambda, \\ A^T y + q = c + \gamma, \\ PQe = \mu e, \\ p > 0, \ \ q > 0. \end{cases} \tag{5.12}$$

The result now follows from standard interior point theory (see for example, [36]; a unique solution to (5.12) exists conforming to the same result for which a unique solution of (1.1) exists). □

We remark that the condition that (5.9) admits strictly feasible points, together with our assumption that A has full row rank, are equivalent to the IPM conditions being satisfied by the pair of dual problems

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad (c+\gamma)^T(x+\lambda) \text{ subject to } Ax=b \text{ and } x \geq -\lambda, \tag{5.13}$$

and

$$\underset{y\in\mathbb{R}^m, s\in\mathbb{R}^n}{\text{maximize}} \quad (b+A\lambda)^T y \text{ subject to } A^T y + s = c \text{ and } s \geq -\gamma. \tag{5.14}$$

Thus the solutions of the system (5.10), as $\mu > 0$ varies, describe the central path of problems (5.13) and (5.14).

In contrast to our original problems (PD), it is now significantly easier to ensure that (5.9) (and equivalently, (5.13) and (5.14)) admits strictly feasible points, as we shall see next.

5.2.1 The choice of starting point and the initialization iteration of Algorithm 2

Suppose a point $\tilde{v}_0 = (\tilde{x}_0, \tilde{y}_0, \tilde{s}_0)$ is available that satisfies

$$A\tilde{x}_0 = b \text{ and } A^\top \tilde{y}_0 + \tilde{s}_0 = c, \tag{5.15}$$

where some components of x_0 and s_0 may be negative. Then various choices of $\lambda_0 \geq 0$ and $\gamma_0 \geq 0$ are possible such that $\tilde{v}_0$ is strictly feasible for (5.9) with $\delta = (\lambda, \gamma) := (\lambda_0, \gamma_0) := \delta_0$. For example, we may let $\lambda_0 = (\lambda_0^i : i = \overline{1,n})$,

$$\lambda_0^i := \begin{cases} 0, & \text{if } \tilde{x}_0^i > 0, \\ 1, & \text{if } \tilde{x}_0^i = 0, \\ -2\tilde{x}_0^i, & \text{if } \tilde{x}_0^i < 0, \end{cases} \tag{5.16}$$

and proceed similarly for γ_0. Thus the IPM conditions hold for (5.13) and (5.14) with $\delta := \delta_0$. The matrix A having full row rank ensures that (5.15) can be satisfied. Indeed, in our implementation of Algorithm 2, the only preprocessing that is done is to guarantee this condition. By factorization or otherwise, it is often computationally inexpensive to find a solution of (5.15).

Algorithm 2 may be initialized with $v_0 := \tilde{v}_0$ and with suitable perturbations $\delta_0 = (\lambda_0, \gamma_0)$, such as (5.16), that make $\tilde{v}_0$ strictly feasible for (5.9), an option which our code provides. It may also be convenient to start from any $v_0 = (x_0, y_0, s_0)$ having $(x_0 + \lambda_0, s_0 + \gamma_0) > 0$, that may not satisfy (5.15), as long as the initial perturbations $\delta_0 = (\lambda_0, \gamma_0)$ are chosen such that (5.9) admits strictly feasible points.

We also choose $\mu > 0$. Then, we apply Algorithm 1 to the system (5.10), or equivalently, to problems (5.13) and (5.14), with $\delta := \delta_0$, starting from v_0. In order words, we compute $v_1 := (x_1, y_1, s_1) := (x(\delta_0, \mu), y(\delta_0, \mu), s(\delta_0, \mu))$. Assuming in what follows that this calculation is performed exactly, v_1 uniquely satisfies the nonlinear system

$$\begin{cases} Ax_k = b, \\ A^T y_k + s_k = c, \\ (X_k + \Lambda_k)(S_k + \Gamma_k)e = \mu e, \\ x_k + \lambda_k > 0, \; s_k + \gamma_k > 0, \end{cases} \tag{5.17}$$

with $k := 1$, $\lambda_1 := \lambda_0$ and $\gamma_1 := \gamma_0$.

5.2.2 The kth iteration of Algorithm 2

For $k \geq 1$, let the current iterate $v_k = (x_k, y_k, s_k)$ be the exact solution of (5.17), where $\delta_k := (\lambda_k, \gamma_k) \geq 0$.

If $x_k > 0$ and $s_k > 0$, we have a strictly feasible point of (PD), and we either terminate or apply Algorithm 1 to (PD), starting from v_k, to make v_k well-centred.

Otherwise, there exists $i \in \{1, \ldots, n\}$ such that $x_k^i \leq 0$ or/and $s_k^i \leq 0$, and we define the new parameters $\lambda_{k+1} = (\lambda_{k+1}^i : i = \overline{1,n})$ and $\gamma_{k+1} = (\gamma_{k+1}^i : i = \overline{1,n})$,

$$\lambda_{k+1}^i := \begin{cases} 0, & \text{if } x_k^i > 0, \\ (1-\zeta)\lambda_k^i + \zeta(-x_k^i), & \text{if } x_k^i \leq 0, \end{cases} \tag{5.18}$$

and

$$\gamma_{k+1}^i := \begin{cases} 0, & \text{if } s_k^i > 0, \\ (1-\zeta)\gamma_k^i + \zeta(-s_k^i), & \text{if } s_k^i \leq 0, \end{cases} \tag{5.19}$$

where $\zeta \in (0,1)$ is a parameter chosen at the start of Algorithm 2. It follows that

$$0 \leq \lambda_{k+1} \leq \lambda_k \text{ and } 0 \leq \gamma_{k+1} \leq \gamma_k, \quad k \geq 1. \tag{5.20}$$

Recalling the strict inequalities in (5.17), we deduce from (5.18) and (5.19) that

$$x_k > -\lambda_{k+1} \text{ and } s_k > -\gamma_{k+1}, \quad k \geq 1, \tag{5.21}$$

and thus, v_k is strictly feasible for (5.9) with $\delta := \delta_{k+1} := (\lambda_{k+1}, \gamma_{k+1})$, and the IPM conditions are satisfied for problems (5.13) and (5.14). We then apply Algorithm 1 to (5.10) with $\delta := \delta_{k+1}$, starting from $v_k = (x_k, y_k, s_k)$. Since the first $m+n$ equations of the systems (5.10) are satisfied at v_k, only the last n equations need to be achieved by Algorithm 1. In exact arithmetic, we obtain a new point $v_{k+1} = (x_{k+1}, y_{k+1}, s_{k+1}) := (x(\delta_{k+1}, \mu), y(\delta_{k+1}, \mu), s(\delta_{k+1}, \mu))$ that uniquely satisfies (5.17) with $k := k+1$. We increase k by one unit, and repeat.

The relation (5.20) provides geometric intuition for the construction of Algorithm 2: in the initialization iteration, we *enlarge* our primal-dual feasible region and compute a well-centred point for the enlarged set. Then, we *successively shrink* this enlarged region and compute a well-centred point for each shrunken polyhedron using Algorithm 1. This computation is always initialized at the well-centred point of the previously perturbed polyhedron. The following statement now follows naturally: the resulting sequence v_k, $k \geq 1$, is strictly feasible for the first perturbed system (5.9) (i. e., when $\delta := \delta_1 = \delta_0$). Thus we have

$$x_k > -\lambda_k \geq -\lambda_1 \quad \text{and} \quad s_k > -\gamma_k \geq -\gamma_1, \quad k \geq 1. \tag{5.22}$$

As a further insight into the choice of perturbations, note that the strict inequalities in (5.17) and (5.21) provide

$$[x_k^i \leq 0 \Longrightarrow (\lambda_k^i, \lambda_{k+1}^i) > 0] \quad \text{and} \quad [s_k^i \leq 0 \Longrightarrow (\gamma_k^i, \gamma_{k+1}^i) > 0]. \tag{5.23}$$

We remark that since every v_k, $k \geq 1$, is strictly feasible for problems (5.13) and (5.14) with $\delta := \delta_{k+1}$ and it is the starting point of Algorithm 1 when applied to these perturbed problems, we only employ a *feasible* version of Algorithm 1 for $k \geq 1$. For $k = 0$, we may want to have the freedom to use the infeasible option available for Algorithm 1.

We have assumed that the system (5.10) is solved exactly by Algorithm 1 on each (major) iteration of Algorithm 2. This assumption is replaced in our implementation and in the analysis in Section 7 by solving (5.10) to high accuracy.

The algorithm we have described can be summarized as follows.

Algorithm 2.

Choose $\mu > 0$ and $\zeta \in (0, 1)$. Find perturbations $\delta_0 := (\lambda_0, \gamma_0) \geq 0$ such that (5.9) has strictly feasible points. Choose a starting point $v_0 = (x_0, y_0, s_0)$ with $(x_0 + \lambda_0, s_0 + \gamma_0) > 0$. Apply Algorithm 1 to (5.10) with $\delta := \delta_0$, starting from v_0, and obtain (exactly) its unique solution $v_1 = (x_1, y_1, s_1)$. Let $\delta_1 := \delta_0$.

On each iteration $k \geq 1$, DO:

Step 1. If $x^k > 0$ and $s^k > 0$, STOP.

Step 2. Update $\delta_{k+1} = (\lambda_{k+1}, \gamma_{k+1})$ according to (5.18) and (5.19).

Step 3. By applying Algorithm 1 to (5.10) with $\delta := \delta_{k+1}$, starting from v_k, compute its solution v_{k+1} (exactly).

Step 4. Let $k := k + 1$, and GO TO Step 1. ◇

We end this subsection with an illustration of the performance of our implementation of Algorithm 2 on Example 3 (page 24). As the table below shows, the problem (5.2) is solved to high accuracy in 7 total iterations (which includes the inner iterations of Algorithm 1). The implementation outputs a message to the user pointing out that there is one primal implicit equality.

```
Iter  p-feas   d-feas   com-slk  merit    step      mu       time
   0  1.0E+00  1.0E+00  1.4E-05  2.0E+00    --      1.0E+00  0.00
   1  4.3E-05  4.3E-05  1.0E-00  1.0E-00  1.00E-00  1.0E+00  0.00
   2  1.8E-05  1.8E-05  7.1E-01  7.1E-01  5.79E-01  1.0E+00  0.00
   3  8.5E-06  8.5E-06  5.2E-01  5.2E-01  5.31E-01  1.0E+00  0.00

================ point satisfying equations found ===============

   4  3.4E-06  3.4E-06  3.7E-01  3.7E-01  5.97E-01  1.0E+00  0.00
   5  1.9E-07  1.9E-07  1.9E-01  1.9E-01  9.45E-01  1.0E+00  0.00
   6  0.0E+00  2.9E-11  5.8E-03  5.8E-03  1.00E+00  1.0E+00  0.00
   7  0.0E+00  2.9E-11  2.0E-17  2.9E-11  1.00E-00  1.0E+00  0.00

=============== well-centred interior point found ==============
```

More numerical results will be presented in Section 8 of the paper. Next, we investigate the convergence properties of Algorithm 2 and its capacity to detect the implicit equalities of (PD).

6 On the convergence properties of Algorithm 2

Firstly, we prove a property of the sequence of iterates generated by Algorithm 2 that does not require the existence of strictly feasible or even feasible points of (PD). As in the previous subsection, an underlying assumption of the current section is that the matrix A has full row rank.

Lemma 6.1 *Let the matrix A have full row rank. The sequence of duality gaps $(x_k^\top s_k)$ of the iterates (x_k, y_k, s_k), $k \geq 1$, generated by Algorithm 2 when applied to the problems* (P) *and* (D), *is bounded from above, independently of k, and the following inequalities hold*

$$x_k^\top s_k \leq n\mu + \lambda_1^\top \gamma_1 := U_{dg}, \quad k \geq 1. \tag{6.1}$$

Proof. The nonlinear equations in (5.17) imply

$$(x_k + \lambda_k)^\top (s_k + \gamma_k) = n\mu, \quad k \geq 1, \tag{6.2}$$

or equivalently,

$$x_k^\top s_k = n\mu - x_k^\top \gamma_k - s_k^\top \lambda_k - \lambda_k^\top \gamma_k, \quad k \geq 1. \tag{6.3}$$

Since $s_k + \gamma_k > 0$ and $\lambda_k \geq 0$, for $k \geq 1$, we have

$$s_k^\top \lambda_k + \gamma_k^\top \lambda_k \geq 0, \quad k \geq 1. \tag{6.4}$$

Similarly, the inequalities $x_k + \lambda_k > 0$ and $\gamma_k \geq 0$ provide

$$x_k^\top \gamma_k + \lambda_k^\top \gamma_k \geq 0, \quad k \geq 1. \tag{6.5}$$

Summing up (6.4) and (6.5), we deduce

$$x_k^\top \gamma_k + s_k^\top \lambda_k \geq -2\lambda_k^\top \gamma_k, \quad k \geq 1. \tag{6.6}$$

It follows from (6.3) that

$$x_k^\top s_k \leq n\mu + \lambda_k^\top \gamma_k, \quad k \geq 1. \tag{6.7}$$

Since from (5.20) we have $\lambda_k^\top \gamma_k \leq \lambda_1^\top \gamma_1$, $k \geq 1$, we conclude that (6.1) holds. □

We remark that it is appropriate to refer to the scalar product $x_k^\top s_k$, $k \geq 1$, as the duality gap of $v_k = (x_k, y_k, s_k)$ since v_k is feasible with respect to the primal-dual equality constraints, conforming to (5.17), and thus $x_k^\top s_k = c^\top x_k - b^\top y_k$.

Next, we introduce some useful notation based on a property of the sequence of iterates of Algorithm 2 and on the structure of the primal-dual feasible set.

The construction of the relaxation parameters (5.18) and (5.19), and the inequalities in (5.17), imply that either $s_k^i \leq 0$, for all $k \geq 1$, or $s_k^i > 0$, at least for all k sufficiently large. In other words, we cannot have sign-alternating subsequences for s_k^i: there do not exist two (or more) subsequences $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $s_k^i \leq 0$, $k \in \mathcal{K}_1$ and $s_k^i > 0$, $k \in \mathcal{K}_2$; once there exists an iteration k_0 such that $s_{k_0}^i > 0$, then $s_k^i > 0$, for all $k \geq k_0$. Similarly, for the x_k components. Thus the following definitions are valid:

$$\mathcal{I}_x := \{i \in \{1, \ldots, n\} : \; x_k^i > 0 \text{ for sufficiently large } k\}, \tag{6.8}$$

and

$$\mathcal{I}_s := \{j \in \{1, \ldots, n\} : \; s_k^j > 0 \text{ for sufficiently large } k\}, \tag{6.9}$$

and their complements

$$\mathcal{N}_x := \{i \in \{1, \ldots, n\} : \ x_k^i \leq 0 \text{ for all } k \geq 1\} = \{1, \ldots, n\} \setminus \mathcal{I}_x, \tag{6.10}$$

and

$$\mathcal{N}_s := \{j \in \{1, \ldots, n\} : \ s_k^j \leq 0 \text{ for all } k \geq 1\} = \{1, \ldots, n\} \setminus \mathcal{I}_s, \tag{6.11}$$

where $\mathcal{N}_x$ or $\mathcal{N}_s$ may be empty (if both of them are, then Algorithm 2 terminates with $k = 1$). It follows from the above considerations that

$$\lambda_k^i = 0, \ i \in \mathcal{I}_x \ \text{ and } \ \gamma_k^j = 0, \ j \in \mathcal{I}_s, \ \text{ for all } k \text{ sufficiently large.} \tag{6.12}$$

The sets $\mathcal{I}_x$ and $\mathcal{I}_s$ may also be empty.

Let us also index the sets of implicit primal and dual equality constraints as

$$\mathcal{E}_p := \{i \in \{1, \ldots, n\} : x^i = 0 \ \text{ for all } \ x \in \mathcal{F}_P\}, \tag{6.13}$$

and

$$\mathcal{E}_d := \{j \in \{1, \ldots, n\} : s^j = 0 \ \text{ for all } \ (y, s) \in \mathcal{F}_D\}, \tag{6.14}$$

where $\mathcal{F}_{P,D}$ were defined by (5.1), and where $\mathcal{E}_p$ or $\mathcal{E}_d$ are empty whenever there are no primal or dual implicit equalities and no primal or dual feasible points, respectively. Let their complements be

$$\mathcal{I}_p := \{1, \ldots, n\} \setminus \mathcal{E}_p \ \text{ and } \ \mathcal{I}_d := \{1, \ldots, n\} \setminus \mathcal{E}_d. \tag{6.15}$$

The sets $\mathcal{I}_p$ and $\mathcal{I}_d$ are empty if and only if either $\mathcal{F}_P = \{x : x = 0\}$ and $\mathcal{F}_D = \{(y, s) : s = 0, \ A^\top y = c\}$, respectively, or $\mathcal{F}_{PD} = \emptyset$. The relation

$$\mathcal{E}_p \cap \mathcal{E}_d = \emptyset, \tag{6.16}$$

trivially holds when $\mathcal{F}_{PD}$ is empty, but it is also satisfied when $\mathcal{F}_{PD}$ is nonempty since then there always exists a primal-dual strictly complementary solution [31]. It follows from (6.15) and (6.16)

$$\mathcal{E}_p \subseteq \mathcal{I}_d \ \text{ and } \ \mathcal{E}_d \subseteq \mathcal{I}_p. \tag{6.17}$$

Since $\mathcal{F}_{PD}$ is a convex set, its relative interior rel$\mathcal{F}_{PD}$ is nonempty provided $\mathcal{F}_{PD}$ is nonempty [29], and we have

$$\text{rel}\mathcal{F}_{PD} = \{(x, y, s) \in \mathcal{F}_{PD} : x^l > 0, \ l \in \mathcal{I}_p, \ s^r > 0, \ r \in \mathcal{I}_d\}. \tag{6.18}$$

Lemma 6.2 *Let the matrix A have full row rank, the problems* (P) *and* (D) *be feasible, and Algorithm 2 be applied to these problems. Then the sequences of components (x_k^r), $r \in \mathcal{I}_d$, and (s_k^l), $l \in \mathcal{I}_p$, of the iterates, are bounded above and below, independently of k.*

In particular, we have the bounds

$$x_k^r \leq \frac{1}{\hat{m}}[n\mu + (\hat{x} + \lambda_1)^\top (\hat{s} + \gamma_1)], \quad r \in \mathcal{I}_d, \quad k \geq 1, \tag{6.19a}$$

$$s_k^l \leq \frac{1}{\hat{m}}[n\mu + (\hat{x} + \lambda_1)^\top (\hat{s} + \gamma_1)], \quad l \in \mathcal{I}_p, \quad k \geq 1, \tag{6.19b}$$

where $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ is any point in the relative interior (6.18) *of $\mathcal{F}_{PD}$ and $\hat{m} := \min\{\hat{x}^l, \hat{s}^r : l \in \mathcal{I}_p \ \text{ and } \ r \in \mathcal{I}_d\}$.*

Proof. The sequences (x_k) and (s_k) are bounded below due to (5.22).

Since $\mathcal{F}_{PD}$ is nonempty, (6.18) and the remark preceding it imply that there exists $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ satisfying

$$A\hat{x} = b,\ A^\top \hat{y} + \hat{s} = c,\ \hat{x}^l > 0,\ l \in \mathcal{I}_p,\ \hat{s}^r > 0,\ r \in \mathcal{I}_d,\ \hat{x}^i = 0,\ i \in \mathcal{E}_p,\ \hat{s}^j = 0,\ j \in \mathcal{E}_d. \tag{6.20}$$

From the first $m + n$ equations in (5.17) and in (6.20), we deduce that $(x_k - \hat{x})$ belongs to the null space of A and $(s_k - \hat{s})$ belongs to the range space of $A^\top$. Thus we have

$$(x_k - \hat{x})^\top (s_k - \hat{s}) = 0, \quad k \geq 1, \tag{6.21}$$

or equivalently,

$$x_k^\top \hat{s} + s_k^\top \hat{x} = x_k^\top s_k + \hat{x}^\top \hat{s}, \quad k \geq 1. \tag{6.22}$$

Recalling (6.20) and (6.15), relation (6.22) becomes

$$\sum_{r \in \mathcal{I}_d} x_k^r \hat{s}^r + \sum_{l \in \mathcal{I}_p} s_k^l \hat{x}^l = x_k^\top s_k + \hat{x}^\top \hat{s}, \quad k \geq 1. \tag{6.23}$$

Let $i \in \mathcal{I}_d$. Relation (6.23) is equivalent to

$$x_k^i \hat{s}^i = -\sum_{\substack{r \in \mathcal{I}_d \\ r \neq i}} x_k^r \hat{s}^r - \sum_{l \in \mathcal{I}_p} s_k^l \hat{x}^l + x_k^\top s_k + \hat{x}^\top \hat{s}, \quad k \geq 1. \tag{6.24}$$

It follows from (5.22), $(\lambda_1, \gamma_1) \geq 0$ and $(\hat{x}^l, \hat{s}^r) > 0$, $l \in \mathcal{I}_p$, $r \in \mathcal{I}_d$, that

$$-\sum_{\substack{r \in \mathcal{I}_d \\ r \neq i}} x_k^r \hat{s}^r - \sum_{l \in \mathcal{I}_p} s_k^l \hat{x}^l \leq \sum_{\substack{r \in \mathcal{I}_d \\ r \neq i}} \lambda_1^r \hat{s}^r + \sum_{l \in \mathcal{I}_p} \gamma_1^l \hat{x}^l \leq \lambda_1^\top \hat{s} + \gamma_1^\top \hat{x},\ \ k \geq 1, \tag{6.25}$$

where in the second inequality we have employed (6.20). The inequality (6.25) and the value of the bound U_{dg} on $(x_k^\top s_k)$ given in (6.1) imply that the right-hand side of (6.24) is bounded above by $n\mu + (\hat{x} + \lambda_1)^\top (\hat{s} + \gamma_1)$, for $k \geq 1$. The bound (6.19a) for $i \in \mathcal{I}_d$ now follows from $\hat{s}^i > 0$, (6.24) and (6.25). The bound (6.19b) follows similarly.

Note that the bounds (6.19) are immediate when $x_k^r \leq 0$ and $s_k^l \leq 0$, since the right-hand side of these bounds is positive. □

Employing the above lemma, we deduce the following theorem.

Theorem 6.3 *Let the matrix A have full row rank, the problems* (P) *and* (D) *be feasible, and Algorithm 2 be applied to these problems. Then $x_k^l > 0$, $l \in \mathcal{I}_p$, and $s_k^r > 0$, $r \in \mathcal{I}_d$, for all sufficiently large k. In other words,*

$$\mathcal{I}_p \subseteq \mathcal{I}_x \ \text{ and } \ \mathcal{I}_d \subseteq \mathcal{I}_s. \tag{6.26}$$

Proof. Since $\mathcal{I}_x$ and $\mathcal{N}_x$ form a partition of the index set $\{1, \ldots, n\}$, as do the sets $\mathcal{I}_s$ and $\mathcal{N}_s$, we deduce

$$\mathcal{I}_p = (\mathcal{I}_p \cap \mathcal{I}_x) \cup (\mathcal{I}_p \cap \mathcal{N}_x) \ \text{ and } \ \mathcal{I}_d = (\mathcal{I}_d \cap \mathcal{I}_s) \cup (\mathcal{I}_d \cap \mathcal{N}_s). \tag{6.27}$$

We will prove that

$$\mathcal{I}_p \cap \mathcal{N}_x = \emptyset \ \text{ and } \ \mathcal{I}_d \cap \mathcal{N}_s = \emptyset, \tag{6.28}$$

which together with (6.27), implies (6.26).

Let us assume there exists $i \in \mathcal{I}_p \cap \mathcal{N}_x$. The nonlinear equations of system (5.17) provide

$$x_k^i + \lambda_k^i = \frac{\mu}{s_k^i + \gamma_k^i}, \quad k \geq 1. \tag{6.29}$$

Employing Lemma 6.2 for $i \in \mathcal{I}_p$, as well as the dual strict inequalities in (5.17) and the second inequalities in (5.20), we deduce the bound

$$0 < s_k^i + \gamma_k^i < U^i + \gamma_1^i := 1/D^i, \quad k \geq 1, \tag{6.30}$$

where U^i is any finite upper bound on the sequence (s_k^i). Thus it follows from (6.29) that

$$x_k^i + \lambda_k^i \geq D^i \mu, \quad k \geq 1. \tag{6.31}$$

Since $i \in \mathcal{N}_x$, we now employ the definition of λ_{k+1}^i from (5.18) in the case $x_k^i \leq 0$, which can be written as

$$\lambda_{k+1}^i = \lambda_k^i - \zeta(x_k^i + \lambda_k^i), \tag{6.32}$$

and conclude

$$\lambda_{k+1}^i \leq \lambda_k^i - D^i \zeta \mu, \ \text{ for all } k \geq 1. \tag{6.33}$$

Summing up over $k \geq 1$, we deduce

$$\lambda_{k+1}^i \leq \lambda_1^i - k D^i \zeta \mu, \quad k \geq 1. \tag{6.34}$$

Furthermore, we obtain

$$\lambda_{k+1}^i \leq 0, \ \text{ for all } \ k \geq \frac{\lambda_1^i}{D^i \zeta \mu} := k_1, \tag{6.35}$$

where $\lambda_1^i > 0$ due to (5.23). Then it follows from the primal strict inequalities in (5.17) with $k := k+1$ that $x_{k+1}^i > 0$ for $k \geq k_1$. Recalling (6.10), we have now reached a contradiction with our assumption that $i \in \mathcal{N}_x$, and therefore, $\mathcal{I}_p \cap \mathcal{N}_x = \emptyset$.

Let now $j \in \mathcal{I}_d \cap \mathcal{N}_s$. By a similar argument, it can be shown that $\gamma_{k+1}^j \leq 0$ for all $k \geq \gamma_1^j/(P^j \zeta \mu) := k_2$, where P^j is a positive constant such that $1/P^j > x_k^j + \lambda_k^j > 0$ for all $k \geq 1$. Then, the dual strict inequalities in (5.17) with $k := k+1$ imply that $s_{k+1}^j > 0$ for $k \geq k_2$, and so $j \in \mathcal{I}_s$ which is a contradiction with the assumption that $j \in \mathcal{N}_s$ since $\mathcal{I}_s \cap \mathcal{N}_s = \emptyset$. □

Let $x^{\mathcal{I}_p} := (x^i : i \in \mathcal{I}_p)$ and $s^{\mathcal{I}_d} := (s^j : j \in \mathcal{I}_s)$, for any $x, s \in \mathbb{R}^n$. Theorem 6.3 and its proof can be restated as showing that it takes a finite number of major iterations k for any nonpositive component of $x_1^{\mathcal{I}_p}$ and $s_1^{\mathcal{I}_d}$ to become positive. This has an immediate implication for the case when (PD) has no implicit equalities, as we shall see next.

6.1 Finite termination of Algorithm 2 for strictly feasible (PD)

We will show that if our original problems (P) and (D) have strictly feasible points, then Algorithm 2 will find such a point in a finite number of major iterations k.

Note that the assumption A2 in the IPM conditions implies that $\mathcal{F}_{PD}$ has no implicit equalities, i. e., $\mathcal{E}_p = \mathcal{E}_d = \emptyset$. Thus,

$$\mathcal{I}_p = \mathcal{I}_d = \{1, \ldots, n\}, \tag{6.36}$$

where the index sets $\mathcal{I}_p$ and $\mathcal{I}_d$ are defined in (6.15). The next two corollaries follow immediately from Lemma 6.2 and Theorem 6.3, respectively.

Corollary 6.4 *Let problems* (P) *and* (D) *satisfy the* IPM *conditions, and apply Algorithm 2 to these problems. Then the iterates* (x_k) *and* (s_k) *are bounded above and below, independently of* k.

In particular, we have the bound

$$\max_{k\geq 1}\max_{i=\overline{1,n}}\{x_k^i, s_k^i\} \leq \frac{1}{\hat{m}}[n\mu + (\hat{x}+\lambda_1)^\top(\hat{s}+\gamma_1)], \tag{6.37}$$

where $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ *is any strictly feasible point of (PD) and* $\hat{m} := \min\{\hat{x}^i, \hat{s}^i : i = \overline{1,n}\}$.

Proof. Lemma 6.2 and (6.36) give the result. □

Corollary 6.5 *Let problems* (P) *and* (D) *satisfy the* IPM *conditions. When applied to these problems, Algorithm 2 will generate a strictly feasible point of (P) and (D) in a finite number of major iterations* k.

In particular, letting Q *denote any finite componentwise upper bound on the sequences* (x_k) *and* (s_k),

$$\max_{k\geq 1}\max_{i=\overline{1,n}}\{x_k^i, s_k^i\} \leq Q, \tag{6.38}$$

we have

$$x_k > 0 \ \ \text{and} \ \ s_k > 0, \ \ \text{for all} \ \ k \geq 1 + \left\lceil \frac{1}{\zeta\mu}\cdot \underset{\{i,j:\,(x_1^i, s_1^j)\leq 0\}}{\text{maximize}} \left\{\lambda_1^i(Q+\gamma_1^i), \gamma_1^j(Q+\lambda_1^j)\right\}\right\rceil. \tag{6.39}$$

The bound Q *in* (6.38) *may take the value given by the right-hand side of* (6.37).

Proof. Since the iterates (v_k) generated by Algorithm 2 satisfy the primal-dual equality constraints, it remains to show that after a finite number of iterations k, we have $x_k > 0$ and $s_k > 0$. This follows from Theorem 6.3, as (6.26) and (6.36) now give $\mathcal{I}_p = \mathcal{I}_d = \mathcal{I}_x = \mathcal{I}_s = \{1, \ldots, n\}$.

To deduce (6.39), note that relations (6.29)—(6.35) in the proof of Theorem 6.3 hold in fact, for any i such that $x_1^i \leq 0$ (which, due to (5.23), implies $\lambda_1^i > 0$). In particular, (6.29)—(6.31) hold for any $i \in \mathcal{I}_p = \{1, \ldots, n\}$. Furthermore, given $x_1^i \leq 0$, the updating rule (6.32) will be applied as long as $x_k^i \leq 0$. Thus relations (6.30) and (6.35) provide

$$i : x_1^i \leq 0 \quad \Longrightarrow \quad x_k^i > 0 \ \text{ for all } \ k \geq 1 + \left\lceil \frac{\lambda_1^i(U^i+\gamma_1^i)}{\zeta\mu}\right\rceil, \tag{6.40}$$

where U^i is any finite upper bound on the sequence (s_k^i).

Similarly, we derive

$$j : s_1^j \leq 0 \quad \Longrightarrow \quad s_k^j > 0 \ \text{ for all } \ k \geq 1 + \left\lceil \frac{\gamma_1^j(T^j+\lambda_1^j)}{\zeta\mu}\right\rceil, \tag{6.41}$$

where T^j is any finite upper bound on the sequence (x_k^j).

Relation (6.39) now follows from (6.38), (6.40) and (6.41). □

In the conditions of the above corollary, relation (6.12) further implies that there is *finite convergence* to zero of the relaxation parameters λ_k and γ_k in terms of major iterations k (alternatively, recall (6.34) in the proof of Theorem 6.3). This latter bound, (6.34), also gives that the larger the user-chosen value of μ, the less the number of outer iterations k.

In an attempt to ensure that the nonpositive components of x_1 and s_1 become "sufficiently" positive (i. e., "far" from constraint boundaries), one might be inclined to require that Algorithm 2 decreases the corresponding components of λ_1 and γ_1 below $-\rho$, where ρ is a "small" given tolerance. It is straightforward to see that Corollary 6.5 still holds in that case, provided the perturbed problems do not become infeasible. Determining an appropriate value for ρ in order to prevent infeasibility from occurring, is nontrivial and might prove to be a significant difficulty in implementations of Algorithm 2. Thus, as we already remarked in Subsection 5.2, when we are not satisfied with the proximity to the central path of the strictly feasible point of (PD) obtained when Algorithm 2 terminates, one further run of Algorithm 1 starting from that iterate can be performed, in order to bring it onto the central path of (PD).

6.2 Detecting the presence of implicit equalities in (PD) or infeasibility

Until now, we have shown that Algorithm 2 terminates in a finite number of major iterations, provided (P) and (D) admit strictly feasible points. However, if the latter condition is not satisfied, the termination criteria in Algorithm 2 cannot be met, and an (infinite) sequence of iterates v_k, $k \geq 1$, will be generated. For the remainder of this section, we analyse the cases when (PD), though feasible, has implicit equalities present among its inequality constraints, and when (PD) is infeasible.

Firstly, we remark that relations (5.20) imply that the sequences of perturbations (λ_k) and (γ_k) are convergent, as $k \to \infty$. Let

$$\lambda_* := \lim_{k\to\infty} \lambda_k \ \text{ and } \ \gamma_* := \lim_{k\to\infty} \gamma_k. \tag{6.42}$$

In the case when (PD) is infeasible or it has implicit equalities, we have

$$\mathcal{N}_x \cup \mathcal{N}_s \neq \emptyset,$$

where $\mathcal{N}_x$ and $\mathcal{N}_s$ were defined in (6.10) and (6.11), since otherwise we reach the contradiction that Algorithm 2 finds a strictly feasible point for (PD) in a finite number of iterations k. Therefore we have an infinite sequence of updates for the parameters λ_k^i, $i \in \mathcal{N}_x$, and γ_k^j, $j \in \mathcal{N}_s$, as the convex combinations defined in (5.18) and (5.19). The latter relations can be written equivalently

$$x_k^i := \frac{1}{\zeta}[-\lambda_{k+1}^i + (1-\zeta)\lambda_k^i],\ i \in \mathcal{N}_x, \quad \text{and} \quad s_k^j := \frac{1}{\zeta}[-\gamma_{k+1}^j + (1-\zeta)\gamma_k^j],\ j \in \mathcal{N}_s, \quad k \geq 1. \tag{6.43}$$

Thus the sequences of nonpositive components (x_k^i), $i \in \mathcal{N}_x$, and (s_k^j), $j \in \mathcal{N}_s$, are also convergent, and

$$\lim_{k\to\infty} x_k^i = -\lim_{k\to\infty} \lambda_k^i := -\lambda_*^i \leq 0,\ i \in \mathcal{N}_x, \ \text{ and } \lim_{k\to\infty} s_k^j = -\lim_{k\to\infty} \gamma_k^j := -\gamma_*^j \leq 0,\ j \in \mathcal{N}_s. \tag{6.44}$$

Relation (6.44) is equivalent to

$$i \in \mathcal{N}_x \quad \Longrightarrow \quad (x_k^i + \lambda_k^i) \to 0, \ \text{ as } \ k \to \infty, \tag{6.45}$$

and

$$j \in \mathcal{N}_s \quad \Longrightarrow \quad (s_k^j + \gamma_k^j) \to 0, \ \text{ as } \ k \to \infty. \tag{6.46}$$

The nonlinear equations in (5.17), i. e., $(x_k^i + \lambda_k^i)(s_k^i + \gamma_k^i) = \mu$, $i = \overline{1, n}$, provide the property

$$i \in \{1, \ldots, n\},\ k \to \infty:\ (x_k^i + \lambda_k^i) \to 0 \quad \Longleftrightarrow \quad s_k^i \to \infty, \tag{6.47}$$

and

$$j \in \{1, \ldots, n\},\ k \to \infty:\ (s_k^j + \gamma_k^j) \to 0 \quad \Longleftrightarrow \quad x_k^j \to \infty, \tag{6.48}$$

where for the forward implication in (6.47), we also used, from (5.20), that $\gamma_k \leq \gamma_1$, so that $s_k^i + \gamma_k^i \to \infty$ implies $s_k^i \to \infty$; for the reverse implication in (6.47), we also employed that, if (s_k^i) is unbounded above then $i \in \mathcal{I}_s$ and (6.12) holds. The proof of (6.48) is similar. Thus we conclude

$$i \in \mathcal{N}_x \quad \Longrightarrow \quad s_k^i \to \infty, \text{ as } k \to \infty, \tag{6.49}$$

and

$$j \in \mathcal{N}_s \quad \Longrightarrow \quad x_k^j \to \infty, \text{ as } k \to \infty. \tag{6.50}$$

6.2.1 Detecting all the implicit equalities in (PD)

Let (PD) be feasible, but not strictly feasible. Thus

$$\mathcal{E}_p \cup \mathcal{E}_d \neq \emptyset, \tag{6.51}$$

where $\mathcal{E}_p$ and $\mathcal{E}_d$ were defined in (6.13) and (6.14), respectively. The next corollary follows from properties proved at the beginning of the section.

Corollary 6.6 *Let the matrix A have full row rank, the problems (P) and (D) be feasible, and Algorithm 2 be applied to these problems. Then*

$$\mathcal{N}_x \subseteq \mathcal{E}_p \ \text{ and } \ \mathcal{N}_s \subseteq \mathcal{E}_d. \tag{6.52}$$

Furthermore,

$$i \in \{1, \ldots, n\},\ k \to \infty:\ x_k^i \to 0 \ \text{ or } \ s_k^i \to \infty \quad \Longrightarrow \quad i \in \mathcal{E}_p, \tag{6.53}$$

and

$$j \in \{1, \ldots, n\},\ k \to \infty:\ s_k^j \to 0 \ \text{ or } \ x_k^j \to \infty \quad \Longrightarrow \quad j \in \mathcal{E}_d. \tag{6.54}$$

Proof. The conditions of Theorem 6.3 are satisfied, and the set inclusions (6.52) are an equivalent formulation of (6.26).

Recalling that the sequences (x_k) and (s_k) are bounded below because of (5.22), an immediate consequence of Lemma 6.2 is that if $s_k^i \to \infty$, then $i \in \mathcal{E}_p$, and if $x_k^j \to \infty$, then $j \in \mathcal{E}_d$.

Let $x_k^i \to 0$, as $k \to \infty$. If $i \in \mathcal{N}_x$, then (6.52) provides $i \in \mathcal{E}_p$. Else, $i \in \mathcal{I}_x$ and (6.12) implies $\lambda_k^i = 0$, for all k sufficiently large. It follows from (6.47) that $s_k^i \to \infty$, which implies $i \in \mathcal{E}_p$. Similarly, $s_k^j \to 0$ implies $j \in \mathcal{E}_d$. □

We now show that in the limit, Algorithm 2 finds a feasible point of (PD). In particular, recalling that v_k, $k \geq 1$, satisfy the primal-dual equality constraints, we would like to prove that (x_k^i), $i \in \mathcal{N}_x$, and (s_k^j), $j \in \mathcal{N}_s$, converge to zero as $k \to \infty$, and not to a negative number as the relations (6.45) and (6.46) may allow. Moreover, furthering Corollary 6.6, we want to give a characterization of the primal and dual implicit equalities in terms of the behaviour of the iterates v_k, in order that we identify *all* the primal-dual implicit equalities. Towards this goal, we would like to answer the following questions: is each (s_k^i), $i \in \mathcal{E}_p$, and (x_k^j), $j \in \mathcal{E}_d$, unbounded above and moreover, divergent to $+\infty$ (i. e., each subsequence of each sequence tends to infinity)? and does each (x_k^i), $i \in \mathcal{E}_p$, and (s_k^j), $j \in \mathcal{E}_d$, converge to zero? The next theorem and corollary are further steps towards answering these questions.

The following property is useful:

$$x_k^i s_k^i = \mu, \text{ for all } k \text{ sufficiently large and each } i \in (\mathcal{E}_p \cap \mathcal{I}_x) \cup (\mathcal{E}_d \cap \mathcal{I}_s). \tag{6.55}$$

To see this, let $i \in \mathcal{E}_p \cap \mathcal{I}_x$. Then (6.12) provides $x_k^i + \lambda_k^i = x_k^i$ for all k sufficiently large. Furthermore, (6.17) and (6.26) imply $i \in \mathcal{I}_s$. Employing (6.12) again, we have $s_k^i + \gamma_k^i = s_k^i$ for all k sufficiently large. The relation (6.55) now follows from the ith nonlinear equation in (5.17), i. e., $(x_k^i + \lambda_k^i)(s_k^i + \gamma_k^i) = \mu$. The argument for $i \in \mathcal{E}_d \cap \mathcal{I}_s$ is similar.

Theorem 6.7 *Let the matrix A have full row rank, the problems (P) and (D) be feasible, and Algorithm 2 be applied to these problems. Then we have the properties*

$$i \in \mathcal{N}_x \quad \Longrightarrow \quad x_k^i \to 0, \ k \to \infty; \tag{6.56}$$

$$j \in \mathcal{N}_s \quad \Longrightarrow \quad s_k^j \to 0, \ k \to \infty. \tag{6.57}$$

Proof. Relation (6.23) in the proof of Lemma 6.2 can be rearranged as

$$x_k^\top s_k = \sum_{r \in \mathcal{I}_d} x_k^r \hat{s}^r + \sum_{l \in \mathcal{I}_p} s_k^l \hat{x}^l - \hat{x}^\top \hat{s}, \quad k \geq 1, \tag{6.58}$$

where $(\hat{x}, \hat{y}, \hat{s})$ belongs to the relative interior of the feasible set of (PD), and thus satisfies (6.20). Lemma 6.2 and $(\hat{x}^{\mathcal{I}_p}, \hat{s}^{\mathcal{I}_d}) > 0$ imply that the right-hand side of (6.58) is bounded above and below independently of k. Thus not only is the duality gap sequence $(x_k^\top s_k)$ bounded above as shown in Lemma 6.1, but, when (PD) is feasible, it is also bounded below by a constant, say L_{dg},

$$x_k^\top s_k \geq L_{dg}, \quad k \geq 1. \tag{6.59}$$

Relations (6.15) and (6.17) provide

$$\begin{aligned} x_k^\top s_k &= \sum_{i \in \mathcal{E}_p} x_k^i s_k^i + \sum_{j \in \mathcal{I}_p} x_k^j s_k^j, \\ &= \sum_{i \in \mathcal{E}_p} x_k^i s_k^i + \sum_{j \in \mathcal{E}_d} x_k^j s_k^j + \sum_{l \in \mathcal{I}_p \cap \mathcal{I}_d} x_k^l s_k^l, \quad k \geq 1. \end{aligned} \tag{6.60}$$

Lemma 6.2 implies that each term of the finite sum over $l \in \mathcal{I}_p \cap \mathcal{I}_d$ in (6.60) is bounded above and below. It follows from (6.59) that there exists a constant L_1, independent of k, such that

$$\sum_{i \in \mathcal{E}_p} x_k^i s_k^i + \sum_{j \in \mathcal{E}_d} x_k^j s_k^j \geq L_1, \quad k \geq 1. \tag{6.61}$$

Employing the second equalities in (6.10) and in (6.11), and (6.52), (6.61) can be expressed equivalently

$$\sum_{i \in \mathcal{N}_x} x_k^i s_k^i + \sum_{i \in \mathcal{E}_p \cap \mathcal{I}_x} x_k^i s_k^i + \sum_{j \in \mathcal{N}_s} x_k^j s_k^j + \sum_{j \in \mathcal{E}_d \cap \mathcal{I}_s} x_k^j s_k^j \geq L_1, \quad k \geq 1. \tag{6.62}$$

Relation (6.55) implies that the second and fourth sums on the left-hand side of expression (6.62) are bounded above and below, and we deduce that there exists a constant L_2, independent of k, such that

$$\sum_{i\in\mathcal{N}_x} x_k^i s_k^i + \sum_{j\in\mathcal{N}_s} x_k^j s_k^j \geq L_2, \quad k \geq 1. \tag{6.63}$$

It follows from (6.52), (6.17) and (6.26) that $\mathcal{N}_x \subseteq \mathcal{I}_s$ and $\mathcal{N}_s \subseteq \mathcal{I}_x$. Therefore $s_k^i > 0$, $i \in \mathcal{N}_x$, and $x_k^j > 0$, $j \in \mathcal{N}_s$, for all k sufficiently large. Furthermore, the definitions (6.10) and (6.11) imply $x_k^i \leq 0$, $i \in \mathcal{N}_x$, and $s_k^j \leq 0$, $j \in \mathcal{N}_s$, for all $k \geq 1$. Thus each term of the sums in (6.63) is nonpositive for all sufficiently large k, and we have

$$i \in \mathcal{N}_x: \qquad 0 \geq x_k^i s_k^i \geq L_2, \quad \text{for all } k \text{ sufficiently large,} \tag{6.64}$$

$$j \in \mathcal{N}_s: \qquad 0 \geq x_k^j s_k^j \geq L_2, \quad \text{for all } k \text{ sufficiently large.} \tag{6.65}$$

Relations (6.49) and (6.64) now imply that $x_k^i \to 0$, which is relation (6.56). Similarly, (6.50) and (6.65) provide (6.57). □

Corollary 6.8 *Let the matrix A have full row rank, the problems (P) and (D) be feasible, and Algorithm 2 be applied to these problems. Then we have the properties*

$$i \in \mathcal{E}_p,\ k \to \infty: s_k^i \to \infty \quad \Longrightarrow \quad x_k^i \to 0; \tag{6.66}$$

$$j \in \mathcal{E}_d,\ k \to \infty: x_k^j \to \infty \quad \Longrightarrow \quad s_k^j \to 0. \tag{6.67}$$

Proof. If $i \in \mathcal{E}_p \cap \mathcal{I}_x$ and $j \in \mathcal{E}_d \cap \mathcal{I}_s$, then (6.66) and (6.67) follow from (6.55). Else, when $i \in \mathcal{E}_p \cap \mathcal{N}_x = \mathcal{N}_x$ and $j \in \mathcal{E}_d \cap \mathcal{N}_s = \mathcal{N}_s$, where we have employed (6.52), then (6.66) and (6.67) follow from (6.56) and (6.57), respectively. The limits (6.49) and (6.50) give further insight. □

The next theorem addresses the question whether the component sequences (s_k^i) and (x_k^j) tend to infinity for *each* $i \in \mathcal{E}_p$ and $j \in \mathcal{E}_d$, respectively. Firstly, we need two auxiliary lemmas. We begin by citing a property of the affine hull of a polyhedron [31].

Let $\mathcal{P}$ be a nonempty polyhedron expressed in general matrix notation

$$\mathcal{P} := \{h : A'h = b',\ \tilde{A}h \leq \tilde{b}\}, \tag{6.68}$$

where $\tilde{b} \in \mathrm{I\!R}^r$, and let $\mathcal{E} \subseteq \{1,\ldots,r\}$ be the indices of the implicit equalities of $\mathcal{P}$. The polyhedron $\mathcal{P}$ is included in the affine subspace

$$\mathrm{aff}(\mathcal{P}) := \{h : A'h = b',\ \tilde{a}_i^\top h = \tilde{b}_i,\ i \in \mathcal{E}\}, \tag{6.69}$$

where $\tilde{a}_i$ denotes the ith row of $\tilde{A}$, which is called the *affine hull* of $\mathcal{P}$ [31]. Every affine subspace that contains $\mathcal{P}$ contains also the affine hull of $\mathcal{P}$ [31]. Furthermore, the implicit equalities of $\mathcal{P}$ play a crucial role in determining the dimension of the affine hull of $\mathcal{P}$, provided they are not redundant.

Lemma 6.9 *([31], page 100) Let the polyhedron $\mathcal{P}$ given in (6.68) be nonempty. Then the affine hull of $\mathcal{P}$ satisfies*

$$\mathit{aff}(\mathcal{P}) = \{h : A'h = b',\ \tilde{a}_i^\top h \leq \tilde{b}_i,\ i \in \mathcal{E}\}. \tag{6.70}$$

The next lemma investigates whether the set of implicit equalities of a polyhedron changes when some of its strict inequality constraints are removed.

Lemma 6.10 *Under the conditions of Lemma 6.9, consider the polyhedron*

$$\mathcal{P}' := \{h : \ A'h = b', \ \tilde{a}_i^\top h \leq \tilde{b}^i, \ i \in \mathcal{I}_0 \cup \mathcal{E}\},$$

where $\mathcal{I}_0 \subset \{1, \ldots, r\} \backslash \mathcal{E}$. *Then* $\mathcal{E}' = \mathcal{E}$, *where* $\mathcal{E}'$ *denotes the indices of the implicit equalities of* $\mathcal{P}'$.

Proof. Since $\mathcal{P} \subseteq \mathcal{P}'$, we have $\mathcal{E}' \subseteq \mathcal{E}$, and thus, $\text{aff}(\mathcal{P}) \subseteq \text{aff}(\mathcal{P}')$. It follows from (6.70) that $\mathcal{P}' \subseteq \text{aff}(\mathcal{P})$, which further implies $\text{aff}(\mathcal{P}') \subseteq \text{aff}(\mathcal{P})$, and so,

$$\text{aff}(\mathcal{P}') = \text{aff}(\mathcal{P}). \tag{6.71}$$

Let us assume there exists $i_0 \in \mathcal{E} \setminus \mathcal{E}'$. Then, (6.71) implies that the inequality constraint indexed by i_0 must be redundant and strict in $\mathcal{P}'$. In other words,

$$h \in \mathcal{P}' \quad \Longrightarrow \quad \tilde{a}_{i_0}^\top h < \tilde{b}_{i_0}.$$

It follows that the constraint i_0 is also strict and redundant in $\mathcal{P}$ since $\mathcal{P} \subseteq \mathcal{P}'$. This is a contradiction with $i_0 \in \mathcal{E}$. □

We are now ready to state and prove the promised theorem.

Theorem 6.11 *Let the matrix A have full row rank, the problems (P) and (D) be feasible, and Algorithm 2 be applied to these problems. Then*

$$i \in \mathcal{E}_p \quad \Longrightarrow \quad s_k^i \to \infty, \, k \to \infty; \tag{6.72}$$

$$j \in \mathcal{E}_d \quad \Longrightarrow \quad x_k^j \to \infty, \, k \to \infty. \tag{6.73}$$

Proof. Clearly, (6.72) and (6.73) hold for $i \in \mathcal{N}_x$ and $j \in \mathcal{N}_s$ because of (6.49) and (6.50), respectively. Let us assume that there exist indices $i \in \mathcal{E}_p \cap \mathcal{I}_x$ and $j \in \mathcal{E}_d \cap \mathcal{I}_s$ such that $s_k^i \nrightarrow \infty$ and $x_k^j \nrightarrow \infty$. Then, the sequence (s_k^i) has at least one subsequence that is bounded above; similarly, for (x_k^j). In what follows, we assume for simplicity that the entire sequence (s_k^i) is bounded above, and the same for (x_k^j). The reader can convince herself that this can be done without loss of generality, by studying the proof below.

Thus let

$$\mathcal{B}_x := \{i \in \mathcal{E}_p : \ (s_k^i) \text{ bounded above}\} \ \text{ and } \ \mathcal{B}_s := \{j \in \mathcal{E}_d : \ (x_k^j) \text{ bounded above}\}, \tag{6.74}$$

and assume that $\mathcal{B}_x \neq \emptyset$ and $\mathcal{B}_s \neq \emptyset$. If $\mathcal{B}_x$ or $\mathcal{B}_s$ is empty, the argument that follows simplifies significantly as there is nothing left to prove in the primal or the dual space, respectively. Then, only one of the two transformations of problems (P) and (D) detailed below is required, depending upon which of the two index sets is empty.

It follows from (6.49) and (6.50) that

$$\mathcal{B}_x \subseteq \mathcal{I}_x \ \text{ and } \ \mathcal{B}_s \subseteq \mathcal{I}_s. \tag{6.75}$$

Thus property (6.55) holds for each $i \in \mathcal{B}_x \cup \mathcal{B}_s$, which implies together with (6.17) and Lemma 6.2 that (x_k^i) and (s_k^i) are bounded above and below away from zero, and thus have limit points and any such limit points, say x_*^i and s_*^i, satisfy

$$x_*^i > 0 \ \text{ and } \ s_*^i > 0, \quad i \in \mathcal{B}_x \cup \mathcal{B}_s. \tag{6.76}$$

We will now consider two equivalent reformulations of problems (P) and (D), that we will then use to show that $\mathcal{B}_x$ and $\mathcal{B}_s$ are empty. Firstly, let us remove all the implicit equalities of (P) that correspond to unbounded dual slacks by removing the respective primal variables, i. e., x^i, $i \in \mathcal{E}_p \setminus \mathcal{B}_x$, and their columns of A, as well as the corresponding dual equality constraints and dual slacks. We obtain the equivalent form of (P) and (D), respectively,

$$\underset{(x^{\mathcal{B}_x}, x^{\mathcal{I}_p}) \in \mathbb{R}^p}{\text{minimize}} \quad \sum_{i \in \mathcal{B}_x \cup \mathcal{I}_p} c^i x^i \ \text{ subject to } \sum_{i \in \mathcal{B}_x \cup \mathcal{I}_p} A_i x^i = b, \ (x^{\mathcal{B}_x}, x^{\mathcal{I}_p}) \geq 0, \tag{6.77}$$

and

$$\underset{y \in \mathbb{R}^m}{\text{maximize}} \ \ b^\top y \ \text{ subject to } \ A_i^\top y \leq c_i, \ i \in \mathcal{B}_x \cup \mathcal{I}_p, \tag{6.78}$$

where $p := |\mathcal{B}_x| + |\mathcal{I}_p|$, $|\cdot|$ denotes the cardinality of the respective index set, A_i is the ith column of A, and where we keep in mind that the primal variables we removed are fixed at zero. Note that the problems remain dual to each other, and that any implicit equality in $\mathcal{F}_P$ and $\mathcal{F}_D$ that has not been removed remains implicit for (6.77) and (6.78), respectively. Let us argue this latter claim briefly. For any $i \in \mathcal{B}_x$, $x^i = 0$ remains implicit in (6.77) since otherwise, there exists a feasible point $\tilde{x} = (\tilde{x}^{\mathcal{B}_x}, \tilde{x}^{\mathcal{I}_p}) \in \mathbb{R}^p$ for (6.77) such that $\tilde{x}^i > 0$. Then, however, we can complete this point $\tilde{x}$ with zeros to obtain a point $x \in \mathbb{R}^n$ that will clearly satisfy $Ax = b$ and $x \geq 0$, and that has $x^i = \tilde{x}^i > 0$, which is a contradiction with $\mathcal{B}_x \subseteq \mathcal{E}_p$. Let u belong to the relative interior of $\mathcal{F}_P$, which implies that $u^{\mathcal{I}_p} > 0$. Furthermore, $u^{\mathcal{B}_x \cup \mathcal{I}_p}$ is feasible for (6.77). Thus no primal nonnegative bounds indexed in $\mathcal{I}_p$ can become implicit equalities in (6.77). For the dual, we have not removed any implicit equalities since $\mathcal{E}_d \subseteq \mathcal{I}_p$. To show that the dual constraints indexed in $\mathcal{E}_d$ are the implicit equalities of the feasible set of (6.78), we employ Lemma 6.10 with $A' := 0$, $b' := 0$, $\tilde{A} := A^\top$, $\tilde{b} = c$, $r := n$, $\mathcal{E} := \mathcal{E}_d$ and $\mathcal{I}_0 \cup \mathcal{E} := \mathcal{B}_x \cup \mathcal{I}_p$.

Next, we address the dual implicit equalities that correspond to unbounded primal variables. Since $\mathcal{E}_d$ is the index set of the implicit equalities of (6.78), we have that any $y \in \mathbb{R}^m$ satisfying $A_i^\top y \leq c^i$, $i \in \mathcal{B}_x \cup \mathcal{I}_p$, also achieves

$$A_i^\top y = c^i, \quad i \in \mathcal{E}_d \setminus \mathcal{B}_s. \tag{6.79}$$

Since $(\mathcal{B}_x, \mathcal{E}_d, \mathcal{I}_p \cap \mathcal{I}_d)$ form a partition of the index set $\mathcal{B}_x \cup \mathcal{I}_p$, it follows from (6.79) that (6.78) is equivalent to

$$\underset{y \in \mathbb{R}^m}{\text{maximize}} \ \ b^T y \ \text{ subject to } \ A_i^T y = c^i, \ i \in \mathcal{E}_d \setminus \mathcal{B}_s, \ \text{ and } \ A_j^T y \leq c^j, \ j \in \mathcal{B}_x \cup \mathcal{B}_s \cup (\mathcal{I}_d \cap \mathcal{I}_p). \tag{6.80}$$

The equivalent form (6.80) of the dual (6.78) implies that the primal variables corresponding to the dual implicit equalities indexed in $\mathcal{E}_d \setminus \mathcal{B}_s$ can be freed from their (nonnegative) lower bounds, and we deduce the dual to (6.80) to be

$$\begin{aligned} \underset{(x^{\mathcal{B}_x}, x^{\mathcal{I}_p}) \in \mathbb{R}^p}{\text{minimize}} \quad \textstyle\sum_{i \in \mathcal{B}_x \cup \mathcal{I}_p} c^i x^i \ \text{ subject to } \quad & \textstyle\sum_{i \in \mathcal{B}_x \cup \mathcal{I}_p} A_i x^i = b, \ (x^{\mathcal{B}_x}, x^{\mathcal{B}_s}, x^{\mathcal{I}_p \cap \mathcal{I}_d}) \geq 0, \\ & x^j \ \text{ free}, \ j \in \mathcal{E}_d \setminus \mathcal{B}_s, \end{aligned} \tag{6.81}$$

which is equivalent to (6.77). It is immediate that $\mathcal{B}_s$ is the index set of the implicit equalities of the feasible set of (6.80), since no constraints and variables have been removed from the feasible set of (6.78). To prove that the primal inequality constraints indexed in $\mathcal{B}_x$ are the implicit equalities of (6.81), employ Lemma 6.10 with $A' := (A_i : i \in \mathcal{B}_x \cup \mathcal{I}_p)$, $b' := b$, $\tilde{A} := -I$, $\tilde{b} := 0$, $\{1, \ldots, r\} := \mathcal{B}_x \cup \mathcal{I}_p$, $\mathcal{E} := \mathcal{B}_x$ and $\mathcal{I}_0 := \mathcal{B}_s \cup (\mathcal{I}_p \cap \mathcal{I}_d)$.

Now we return to the sequence of iterates generated by Algorithm 2 and to their relation to problems (6.80) and (6.81). From Lemma 6.2, (6.74) and (6.76), we deduce that the sequence (s_k^i) is bounded, for each $i \in \mathcal{B}_x \cup \mathcal{I}_p$. Thus $(s_k^i : i \in \mathcal{B}_x \cup \mathcal{I}_p)$ has limit points. Moreover, (6.67) in Corollary 6.8 provides $s_k^i \to 0$, $i \in \mathcal{E}_d \setminus \mathcal{B}_s$. Therefore letting $s_*^{\mathcal{B}_x \cup \mathcal{I}_p}$ be any such limit point, and $\mathcal{B} := \mathcal{B}_x \cup \mathcal{B}_s \cup (\mathcal{I}_p \cap \mathcal{I}_d)$, we have

$$s_*^{\mathcal{E}_d \setminus \mathcal{B}_s} = 0 \text{ and } s_*^{\mathcal{B}} \geq 0, \tag{6.82}$$

where to deduce the inequality, we employed (6.9), (6.76) and the set inclusions

$$\mathcal{I}_p \cap \mathcal{I}_d \subseteq \mathcal{I}_d \subseteq \mathcal{I}_s,$$

that follow from (6.26). Since A has full row rank, (y_k) is also bounded and thus, has limit points, say y_*. As the iterates (y_k, s_k), $k \geq 1$, satisfy $A^\top y_k + s_k = c$, it follows from (6.82) that y_* is feasible for (6.80), and we have

$$A_{\mathcal{E}_d \setminus \mathcal{B}_s}^\top y_* = c^{\mathcal{E}_d \setminus \mathcal{B}_s}, \quad A_{\mathcal{B}}^\top y_* \leq c^{\mathcal{B}}. \tag{6.83}$$

Similarly, the primal components (x_k^i), $i \in \mathcal{B}$, are bounded above and below. Let $x_*^{\mathcal{B}} \in \mathbb{R}^{|\mathcal{B}|}$ be any limit point of $(x_k^{\mathcal{B}})$. Then $x_*^{\mathcal{B}} \geq 0$, due to (6.76) and the set inclusions

$$\mathcal{I}_p \cap \mathcal{I}_d \subseteq \mathcal{I}_p \subseteq \mathcal{I}_x,$$

that follow from (6.26). We also have, however, that $x_k^j \to \infty$, $j \in \mathcal{E}_d \setminus \mathcal{B}_s$, and we need to do something less straightforward than in the dual case, in order to retrieve a feasible point of (6.81).

Since $\mathcal{B}_x \cup \mathcal{I}_p = \mathcal{B} \cup (\mathcal{E}_d \setminus \mathcal{B}_s)$, the equality constraints of (6.81) can be written equivalently as

$$\sum_{j \in \mathcal{E}_d \setminus \mathcal{B}_s} A_j x^j = b - \sum_{i \in \mathcal{B}} A_i x^i. \tag{6.84}$$

Letting $x^{\mathcal{B}} := x_*^{\mathcal{B}}$ in the right-hand side of (6.84), we obtain the system

$$\sum_{j \in \mathcal{E}_d \setminus \mathcal{B}_s} A_j x^j = b - \sum_{i \in \mathcal{B}} A_i x_*^i. \tag{6.85}$$

If the matrix $A_{\mathcal{E}_d \setminus \mathcal{B}_s}$ had full column rank, then the system (6.85) would have a (unique) *finite* solution $\tilde{x}^{\mathcal{E}_d \setminus \mathcal{B}_s}$, as the matrix and right-hand side of the system (6.85) are real and finite. However, $Ax_k = b$, $k \geq 1$, and $x_k^{\mathcal{B}} \to x_*^{\mathcal{B}} < \infty$ on a subsequence, imply

$$\sum_{j \in \mathcal{E}_d \setminus \mathcal{B}_s} A_j (\lim_{k \to \infty} x_k^j) = b - \sum_{i \in \mathcal{B}} A_i x_*^i, \tag{6.86}$$

where we have proved that $x_k^j \to \infty$, $k \to \infty$, for all $j \in \mathcal{E}_d \setminus \mathcal{B}_s$. Thus we have found a solution of (6.85) that is not finite, and reached a contradiction with our assumption that the matrix $A_{\mathcal{E}_d \setminus \mathcal{B}_s}$ can have full column rank. It follows that (6.85) has multiple (and finite) solutions in $\mathbb{R}^{|\mathcal{E}_d \setminus \mathcal{B}_s|}$. Compute any such finite solution, say $x_*^{\mathcal{E}_d \setminus \mathcal{B}_s}$, of (6.85). Then $(x_*^{\mathcal{E}_d \setminus \mathcal{B}_s}, x_*^{\mathcal{B}})$ is feasible for (6.81), and we have

$$A_{\mathcal{B}_x \cup \mathcal{I}_p} x_*^{\mathcal{B}_x \cup \mathcal{I}_p} = b, \quad x_*^{\mathcal{B}} \geq 0. \tag{6.87}$$

Since $\mathcal{B}_x$ and $\mathcal{B}_s$ are the indices of the implicit equalities of (6.81) and (6.80), respectively, it follows from (6.83) and (6.87) that

$$x_*^i = 0,\ i \in \mathcal{B}_x, \text{ and } s_*^j = c^j - A_j^\top y_* = 0,\ j \in \mathcal{B}_s. \tag{6.88}$$

This, however, contradicts (6.76).

Thus the sets $\mathcal{B}_x$ and $\mathcal{B}_s$ are empty. Now recall our remark in the first paragraph of the proof that the above analysis holds if we assume that (s_k^i) and (x_k^j) each have at least a bounded above subsequence, for some $i \in \mathcal{E}_p \cap \mathcal{I}_x$ and $j \in \mathcal{E}_d \cap \mathcal{I}_s$. Thus the index sets of such sequences would also be empty, and we conclude that $x_k^i \to 0$ and $s_k^i \to \infty$ when $i \in \mathcal{E}_p \cap \mathcal{I}_x$, and $x_k^j \to \infty$, $s_k^j \to 0$ when $j \in \mathcal{E}_d \cap \mathcal{I}_s$. □

The next corollary summarizes our results so far.

Corollary 6.12 *Let the matrix A have full row rank, the problems (P) and (D) be feasible, and Algorithm 2 be applied to these problems. Then*

$$\mathcal{E}_p = \{i \in \{1,\ldots,n\} :\ s_k^i \to \infty,\ k \to \infty\} = \{i \in \{1,\ldots,n\} :\ x_k^i \to 0,\ k \to \infty\}, \tag{6.89}$$

and

$$\mathcal{E}_d = \{j \in \{1,\ldots,n\} :\ x_k^j \to \infty,\ k \to \infty\} = \{j \in \{1,\ldots,n\} :\ s_k^j \to 0,\ k \to \infty\}. \tag{6.90}$$

Also, the complement index sets can be characterized as follows

$$\mathcal{I}_p = \{i \in \{1,\ldots,n\} :\ (s_k^i) \textit{ bounded above}\} = \{i \in \{1,\ldots,n\} :\ (x_k^i) \textit{ bounded away from zero}\}, \tag{6.91}$$

and

$$\mathcal{I}_d = \{j \in \{1,\ldots,n\} :\ (x_k^j) \textit{ bounded above}\} = \{j \in \{1,\ldots,n\} :\ (s_k^j) \textit{ bounded away from zero}\}. \tag{6.92}$$

Furthemore, the perturbation parameters δ_k satisfy

$$\delta_k = (\lambda_k, \gamma_k) \to 0, \quad \textit{as } k \to \infty. \tag{6.93}$$

Proof. Relations (6.89) and (6.90) follow from (6.53) and (6.54) in Corollary 6.6, from Corollary 6.8 and Theorem 6.11.

To show the first equality in (6.91) and in (6.92), employ Lemma 6.2, the first equality in (6.89) and in (6.90), and the definitions (6.15).

Let us now show the second equality in (6.91), the one in (6.92) following similarly. An equivalent formulation of (6.89) is

$$i \in \mathcal{I}_p \iff x_k^i \nrightarrow 0,\ k \to \infty \iff \text{there exists } (x_k^i),\ k \in \mathcal{K}, \text{ bounded away from zero.} \tag{6.94}$$

It remains to prove that for each $i \in \mathcal{I}_p$, the whole sequence (x_k^i) is bounded away from zero. Let $i \in \mathcal{I}_p$ be fixed and let us assume there exists a subsequence (x_k^i), $k \in \mathcal{K}_0$, such that

$$x_k^i \to 0, \text{ as } k \to \infty,\ k \in \mathcal{K}_0. \tag{6.95}$$

Recalling the first equality in (6.90) and that $(\mathcal{E}_d, \mathcal{I}_p \cap \mathcal{I}_d)$ forms a partition of $\mathcal{I}_p$, then, clearly, $i \notin \mathcal{E}_d$, and so $i \in \mathcal{I}_p \cap \mathcal{I}_d$. From (6.26), we have $\mathcal{I}_p \cap \mathcal{I}_d \subseteq \mathcal{I}_x \cap \mathcal{I}_s$. Thus (6.12) holds, which together with the ith nonlinear equation in (5.17), i. e., $(x_k^i + \lambda_k^i)(s_k^i + \gamma_k^i) = \mu$, imply

$$x_k^i s_k^i = \mu, \text{ for } k \text{ sufficiently large.}$$

This, and (6.95), provide $s_k^i \to \infty$, for $k \in \mathcal{K}_0$, $k \to \infty$, and thus, (s_k^i), $k \geq 1$, is unbounded above, which is a contradiction with $i \in \mathcal{I}_p$ and the first inequality in (6.91).

Finally, to prove (6.93), recall that we showed in (6.42) that (δ_k) is convergent. Moreover, recalling (6.12), it remains to show that $\lambda_k^i \to 0$, $i \in \mathcal{N}_x$, and $\gamma_k^j \to 0$, $j \in \mathcal{N}_s$. This now follows from (6.44), (6.56) and (6.57). □

The above corollary gives the promised characterization of the implicit equalities of (PD) in terms of the behaviour of the respective components of the sequence of iterates generated by Algorithm 2. Thus we have a practical way of identifying all of them. The next section discusses what to do with these constraints once we established they are implicit equalities, and how to obtain a feasible point of (PD) and a starting point that a feasible interior point method could then use to solve (PD) to optimality.

6.2.1.1 Removing the implicit equalities from (PD) and finding a point in the relative interior of the primal-dual feasible set

Assume now that we have identified the index sets $\mathcal{E}_p$ and $\mathcal{E}_d$, by employing Corollary 6.12. Provided $\mathcal{E}_p$ is nonempty, let us remove from (P) all its implicit equalities by deleting the respective primal variables x^i, $i \in \mathcal{E}_p$, and their columns of A, as well as the corresponding dual equality constraints and dual slacks. We obtain the equivalent form of (P) and (D)

$$\underset{x^{\mathcal{I}_p} \in \mathbb{R}^p}{\text{minimize}} \sum_{i \in \mathcal{I}_p} c^i x^i \text{ subject to } \sum_{i \in \mathcal{I}_p} A_i x^i = b, \; x^{\mathcal{I}_p} \geq 0, \tag{6.96}$$

and

$$\underset{(y, s^{\mathcal{I}_p}) \in \mathbb{R}^m \times \mathbb{R}^p}{\text{maximize}} \quad b^\top y \text{ subject to } A_i^\top y + s^i = c_i, \; i \in \mathcal{I}_p, \; s^{\mathcal{I}_p} \geq 0. \tag{6.97}$$

where $p := |\mathcal{I}_p|$, $|\cdot|$ denotes the cardinality of the respective index set, A_i is the ith column of A, and where we keep in mind that the primal variables we removed are fixed at zero. Problems (6.96) and (6.97) remain dual to each other.

If $\mathcal{E}_d = \emptyset$, then Lemma 6.2 implies that the sequence $(x_k^{\mathcal{I}_p}, s_k^{\mathcal{I}_p})$ is bounded and thus has limit points. The same is true for (y_k), since A has full row rank. Let $(x_*^{\mathcal{I}_p}, y_*, s_*^{\mathcal{I}_p})$ be any such limit point. Then (6.91) and (6.92) in Corollary 6.12 provide

$$(x_*^{\mathcal{I}_p}, s_*^{\mathcal{I}_p}) > 0. \tag{6.98}$$

Since (x_k, y_k, s_k), $k \geq 1$, are feasible with respect to the equality constraints of (PD), $(x_*^{\mathcal{I}_p}, y_*, s_*^{\mathcal{I}_p})$ is also feasible for the equality constraints of (6.96) and (6.97). Asymptotically, we have thus obtained a strictly feasible point of (6.96) and (6.97). Moreover, it is also well-centred for these problems (in the sense that it belongs to their central path) since (6.26), (5.18) and (5.19) imply $x_k^i s_k^i = \mu$, $i \in \mathcal{I}_p$, for all k sufficiently large, and therefore, in the limit, we have $x_*^i s_*^i = \mu$, $i \in \mathcal{I}_p$. It is now problems (6.96) and (6.97) that a (feasible) interior point method will solve to optimality, possibly starting from the well-centred strictly feasible point we have found.

When $\mathcal{E}_d \neq \emptyset$, due to (6.90), the primal components x_k^i, $i \in \mathcal{E}_d$, are unbounded above, and thus obtaining a primal-dual feasible point is less immediate in this case. Firstly, let us remove the dual implicit equalities $s^i = 0$, $i \in \mathcal{E}_d$, from problem (6.97). Note that since $\mathcal{E}_d \subseteq \mathcal{I}_p$, the indices of the implicit equalities of (D) are still present in (6.97). Furthermore, Lemma 6.10 can be employed to show that in fact, $\mathcal{E}_d$ is also the index set of the implicit equalities of (6.97). Thus for any $y \in \mathbb{R}^m$ such that $A_i^\top y \leq c^i$, $i \in \mathcal{I}_p$, we have

$$A_i^\top y = c^i, \quad i \in \mathcal{E}_d. \tag{6.99}$$

The dual problem (6.97) can be rewritten as

$$\underset{y \in \mathbb{R}^m}{\text{maximize}} \; b^T y \; \text{ subject to } \; A_i^T y = c^i,\, i \in \mathcal{E}_d, \; \text{ and } \; A_j^T y \leq c^j,\, j \in \mathcal{I}_d \cap \mathcal{I}_p. \tag{6.100}$$

The primal problem corresponding to (6.100) is

$$\underset{x^{\mathcal{I}_p} \in \mathbb{R}^p}{\text{minimize}} \; \sum_{i \in \mathcal{I}_p} c^i x^i \; \text{ subject to } \; \sum_{i \in \mathcal{I}_p} A_i x^i = b, \;\; x^j \geq 0,\, j \in \mathcal{I}_p \cap \mathcal{I}_d, \;\; x^i \text{ free},\, i \in \mathcal{E}_d. \tag{6.101}$$

Thus knowing the dual implicit equalities removes the nonnegativity constraints on the corresponding primal variables, which is always useful in computations. It is this problem, (6.101), and its dual (6.100), that a (feasible) interior point method will now solve to optimality, and for which, we will further deduce a strictly feasible point using the iterates generated by Algorithm 2.

Corollary 6.12 implies that $(s_k^{\mathcal{I}_d \cap \mathcal{I}_p})$ is bounded above and below away from zero, and $s_k^{\mathcal{E}_d} \to 0$. Thus every limit point of $(s_k^{\mathcal{I}_p})$, say $s_*^{\mathcal{I}_p}$, satisfies

$$s_*^{\mathcal{E}_d} = 0 \;\text{ and }\; s_*^{\mathcal{I}_d \cap \mathcal{I}_p} > 0. \tag{6.102}$$

Since A has full row rank, (y_k) is also bounded and has limit points. As the iterates (y_k, s_k) satisfy $A^\top y_k + s_k = c$, it follows from (6.102) that every limit point of (y_k) is strictly feasible for (6.100).

Similarly, (6.91) and (6.92) in Corollary 6.12 provide that the primal components $(x_k^{\mathcal{I}_d \cap \mathcal{I}_p})$ are bounded above and below away from zero. Thus any of its limit points, say $x_*^{\mathcal{I}_d \cap \mathcal{I}_p}$, satisfies

$$x_*^{\mathcal{I}_d \cap \mathcal{I}_p} > 0. \tag{6.103}$$

Due to (6.90), however, the components $x_k^{\mathcal{E}_d} \to \infty$. Since $\mathcal{I}_p = (\mathcal{I}_p \cap \mathcal{I}_d) \cup \mathcal{E}_d$, the equality constraints of (6.81) can be written equivalently as

$$\sum_{i \in \mathcal{E}_d} A_i x^i = b - \sum_{j \in \mathcal{I}_p \cap \mathcal{I}_d} A_j x^j. \tag{6.104}$$

Letting $x^{\mathcal{I}_p \cap \mathcal{I}_d} := x_*^{\mathcal{I}_p \cap \mathcal{I}_d}$ in the right-hand side of (6.104), we obtain the system

$$\sum_{i \in \mathcal{E}_d} A_i x^i = b - \sum_{j \in \mathcal{I}_p \cap \mathcal{I}_d} A_j x_*^j. \tag{6.105}$$

If the matrix $A_{\mathcal{E}_d}$ had full column rank, then the system (6.105) would have a (unique) *finite* solution $\tilde{x}^{\mathcal{E}_d}$, as the matrix and right-hand side of the system (6.105) are real and finite. However, $Ax_k = b$, $k \geq 1$, and $x_k^{\mathcal{I}_p \cap \mathcal{I}_d} \to x_*^{\mathcal{I}_p \cap \mathcal{I}_d} < \infty$ on a subsequence, imply

$$\sum_{i \in \mathcal{E}_d} A_i (\lim_{k \to \infty} x_k^i) = b - \sum_{j \in \mathcal{I}_p \cap \mathcal{I}_d} A_j x_*^j, \tag{6.106}$$

where we have proved that $x_k^i \to \infty$, $k \to \infty$, for all $i \in \mathcal{E}_d$. Thus we have found a solution of (6.105) that is not finite, and reached a contradiction with our assumption that the matrix $A_{\mathcal{E}_d}$ can have full column rank. It follows that (6.105) has multiple (and finite) solutions in $\mathrm{I\!R}^{|\mathcal{E}_d|}$. Compute any such finite solution, say $x_*^{\mathcal{E}_d}$, of (6.105), for instance, the minimal norm solution. Then $x_*^{\mathcal{I}_p} := (x_*^{\mathcal{E}_d}, x_*^{\mathcal{I}_p \cap \mathcal{I}_d})$ satisfies $A_{\mathcal{I}_p} x_*^{\mathcal{I}_p} = b$, and (6.103) further implies that $x_*^{\mathcal{I}_p}$ is strictly feasible for (6.101).

Also, relations (6.26), (5.18) and (5.19) imply that $x_k^i s_k^i = \mu$, $i \in \mathcal{I}_p \cap \mathcal{I}_d$, for all k sufficiently large. Thus, in the limit, we have

$$x_*^i s_*^i = \mu, \quad i \in \mathcal{I}_p \cap \mathcal{I}_d.$$

If $\mathcal{E}_p = \emptyset$ and $\mathcal{E}_d \neq \emptyset$, there is no need to perform the first transformation of problems (PD) into (6.96) and (6.97), we can do straightaway the change to (6.101) and (6.100).

6.2.2 Detecting when (PD) is infeasible

The remarks in the second paragraph at the beginning of the current subsection hold even when (PD) is infeasible, and the next corollary follows immediately from (6.93) in Corollary 6.12.

Corollary 6.13 *Let the matrix A have full row rank, and apply Algorithm 2 to problems* (P) *and* (D). *Recalling* (6.42), *if $(\lambda_*, \gamma_*) \neq 0$, then (PD) is infeasible.*

In the conditions of Corollary 6.13, it follows from (6.44) that if $\lambda_* = 0$, then $x_k^{\mathcal{N}_x} \to 0$, $k \to \infty$. We also have $x_k^i > 0$, $i \in \mathcal{I}_x$, for all k sufficiently large. Since $Ax_k = b$, $k \geq 1$, Algorithm 2 will then tend towards a primal feasible point, though some of the components of this point may be unbounded above. We can thus conclude that then, (P) is feasible and (D) is infeasible. Similarly if $\lambda_* \neq 0$ and $\gamma_* = 0$, then (P) is infeasible and (D), feasible.

We noticed in our numerical experiments that, when a problem is infeasible, it depends on the choice of starting point for Algorithm 2, as to which constraint has its corresponding relaxation parameter not converging to zero.

7 An inexact variant of Algorithm 2

Until now, we have assumed that, on each major iteration $k \geq 0$ of Algorithm 2, the system (5.10) with $\delta = (\lambda, \gamma) := \delta_{k+1}$ is solved *exactly* using Algorithm 1. In other words, the optimality tolerance ϵ in the termination criteria of Algorithm 1 is set to its asymptotic value of zero. To bridge the gap with practical computations, let us now allow for a *positive tolerance* ϵ_k in the termination criteria of Algorithm 1, for each $k \geq 0$. Thus we terminate each major iteration $k \geq 0$ when Algorithm 1 applied to (5.10) with $\delta := \delta_{k+1}$, has generated an iterate $v_{k,l} = (x_{k,l}, y_{k,l}, s_{k,l}) := v_{k+1}$ such that

$$\Phi_{k+1}(v_{k+1}) := \Phi_{k+1}(v_{k,l}) \leq \epsilon_k, \tag{7.1}$$

where we let $v_{k,0} := v_k$, $k \geq 0$, and Φ_{k+1} denote the merit function Φ defined in (2.7) that corresponds to problem (5.10) with $\delta := \delta_{k+1}$. We refer to this variant of Algorithm 2 as Algorithm 2_ϵ.

It is useful to require that, in addition to the conditions described in Section 5.2.1, for the initialization iteration $k = 0$ of Algorithm 2_ϵ, ϵ_0 is chosen such that Algorithm 1 iterates at least until the stepsize $\alpha_{0,r}$ equals one. Due to (3.44) in Theorem 3.5, we know this happens provided $\alpha_{0,r} := \alpha^{Q}_{0,r}$, for all $r \geq 0$. Furthermore, (3.62) in Theorem 3.6 and more precisely, (3.92) in Corollary 3.11, give an upper bound on the number of inner iterations r that Algorithm 1 requires to generate an iterate $v_{0,r}$ with this property. Then, we also have

$$Ax_{k,r} = b \ \text{ and } \ A^\top y_{k,r} + s_{k,r} = c, \ \text{ for all } r \geq 0 \text{ and } k \geq 1, \tag{7.2}$$

and

$$Ax_k = b \ \text{ and } \ A^\top y_k + s_k = c, \ \text{ for } k \geq 1. \tag{7.3}$$

It follows from (7.2) and (2.7) that

$$\Phi_{k+1}(v_{k,r}) = \|(X_{k,r} + \Lambda_{k+1})(S_{k,r} + \Gamma_{k+1})e - \mu e\|, \quad r \geq 0, \ k \geq 1. \tag{7.4}$$

The condition (7.1) implies

$$|(x^i_{k+1} + \lambda^i_{k+1})(s^i_{k+1} + \gamma^i_{k+1}) - \mu| \leq \frac{\epsilon_k}{p}, \ i = \overline{1,n}, \quad k \geq 0, \tag{7.5}$$

where $p := 1$ *when the* l_1, l_2 *or* l_∞ *is employed in the definition of* Φ_{k+1}, $k \geq 0$, and $p := \underline{p}$ when a general norm satisfying (3.21) is used in Φ_{k+1}, $k \geq 0$ ($\underline{p}$ is defined in (3.21)).

The issues to be addressed concern the values that the tolerances ϵ_k, $k \geq 0$, can take, if any, such that Algorithm 2_ϵ inherits the properties of Algorithm 2. There is plenty of freedom in the choice of (ϵ_k), as the following results show.

Let us assume in what follows that (ϵ_k) is bounded above, i. e.,

$$\epsilon_k \leq \overline{\epsilon}, \ k \geq 0, \ \text{ for some } \overline{\epsilon} > 0. \tag{7.6}$$

Firstly, please note that all the properties of Algorithm 2 in Subsection 5.2 carry through for Algorithm 2_ϵ. The next lemma is the equivalent of Lemma 6.1.

Lemma 7.1 *Let the matrix A have full row rank. Then, provided* (7.6) *holds, the sequence of duality gaps $(x_k^\top s_k)$ of the iterates (x_k, y_k, s_k), $k \geq 1$, generated by Algorithm 2_ϵ when applied to the problems* (P) *and* (D), *is bounded from above, independently of k. In particular,*

$$x_k^\top s_k \leq n\mu + n\frac{\overline{\epsilon}}{p} + \lambda_1^\top \gamma_1 := U^{\epsilon}_{dg}, \quad k \geq 1, \tag{7.7}$$

where p is defined in (7.5).

Proof. The proof follows exactly like that of Lemma 6.1, the only change being that (6.2) is now replaced by

$$(x_k + \lambda_k)^\top (s_k + \gamma_k) \leq n\left(\mu + \frac{\epsilon_k}{p}\right), \quad k \geq 1, \tag{7.8}$$

which follows from (7.5). □

Furthermore, the definitions (6.8), (6.9), (6.10) and (6.11) of the index sets $(\mathcal{N}_x, \mathcal{I}_x)$ and $(\mathcal{N}_s, \mathcal{I}_s)$ as partitions of $\{1, \ldots, n\}$ still hold for the iterates generated by Algorithm 2_ϵ. We have the following equivalent of Lemma 6.2.

Lemma 7.2 *Let problems* (P) *and* (D) *satisfy the* IPM *conditions, and apply Algorithm* 2_ϵ *to these problems. Then, provided* (7.6) *holds, the iterates* (x_k) *and* (s_k) *are bounded above and below, independently of* k.

In particular, we have the bound

$$\max_{k\geq 1} \max_{i=\overline{1,n}} \{x_k^i, s_k^i\} \leq \frac{1}{\hat{m}} \left[n\mu + n\frac{\overline{\epsilon}}{p} + (\hat{x}+\lambda_1)^\top (\hat{s}+\gamma_1) \right], \tag{7.9}$$

where $\hat{v} = (\hat{x}, \hat{y}, \hat{s})$ *is any strictly feasible point of (PD) and* $\hat{m} := \{\hat{x}^i, \hat{s}^i : i = \overline{1,n}\}$, *and where* p *is defined in* (7.5).

Proof. As (7.3) holds, the proof follows similarly to that of Lemma 6.2, where $\mathcal{I}_p$ and $\mathcal{I}_d$ are now equal to $\{1, \ldots, n\}$ since (PD) satisfies the IPM conditions. The only change that occurs is that the value of the bound on $(x_k^\top s_k)$ is now taken from (7.7). □

When (PD) has strictly feasible points, Algorithm 2_ϵ will find such a point in a finite number of major iterations k.

Theorem 7.3 *Let problems* (P) *and* (D) *satisfy the* IPM *conditions, and assume that* (7.6) *holds, where*

$$\overline{\epsilon} := p(\mu - \mu_0), \tag{7.10}$$

for any (fixed) $\mu_0 \in (0, \mu)$, *where* p *is defined in* (7.5). *Then Algorithm* 2_ϵ *will generate a strictly feasible point of (P) and (D) in a finite number of major iterations* k. *In particular, letting* Q^ϵ *denote any finite componentwise upper bound on the sequences* (x_k) *and* (s_k),

$$\max_{k\geq 1} \max_{i=\overline{1,n}} \{x_k^i, s_k^i\} \leq Q^\epsilon, \tag{7.11}$$

we have

$$x_k > 0 \text{ and } s_k > 0, \text{ for all } k \geq 1 + \left\lceil \frac{1}{\zeta\mu_0} \cdot \underset{\{i,j:\,(x_1^i, s_1^j)\leq 0\}}{\text{maximize}} \left\{ \lambda_1^i (Q^\epsilon + \gamma_1^i), \gamma_1^j (Q^\epsilon + \lambda_1^j) \right\} \right\rceil. \tag{7.12}$$

The bound Q^ϵ *in* (7.11) *may take the value given by the right-hand side of* (7.9).

Proof. The proof follows the same arguments as that of Theorem 6.3, where $\mathcal{I}_p = \mathcal{I}_d = \{1, \ldots, n\}$ since (PD) satisfies the IPM conditions. The only difference is that instead of employing the ith nonlinear equation in (5.17), we use (7.5), (7.6) and (7.10). □

The next corollary addresses the overall iteration complexity of Algorithm 2_ϵ.

Corollary 7.4 *In addition to the conditions of Theorem 7.3, let the starting point* v_0 *and the initial perturbations* $\delta_0 = (\lambda_0, \gamma_0)$ *of Algorithm* 2_ϵ *be chosen such that* v_0 *is strictly feasible for* (5.9) *with* $\delta := \delta_0$, *or such that* x_0 *and* s_0 *are larger, componentwise, than a strictly feasible point of* (5.9) *(a simple way to ensure these conditions is described in Section 5.2.1). Then Algorithm* 2_ϵ *will generate a strictly feasible point of (P) and (D) in a finite number of total iterations, where an upper bound on the number of inner iterations* l *performed by Algorithm 1 on each outer iteration* $k \geq 0$, *follows from Corollary 3.11.*

Proof. The assumptions of the corollary and (7.2) imply that the conditions of Corollary 3.11 are satisfied by Algorithm 1 when applied to each subproblem (5.9) with $\delta := \delta_k$, $k \geq 0$. Thus this corollary provides the property that it takes a finite number of minor iterations l for Algorithm 1 to generate an iterate $v_{k,l}$ satisfying (7.1) (l naturally depends on k). Theorem 7.3 concludes the proof. □

Let us briefly illustrate applying the complexity results of Corollary 3.11 in the context of Corollary 7.4. Let the starting point v_0 of Algorithm 2_ϵ be strictly feasible for (5.9) with $\delta := \delta_0$, and let the l_2 norm be employed in the definition of Φ_{k+1} for each $k \geq 0$. Then Φ_{k+1} has the expression (7.4) not only for $k \geq 1$, but also for $k = 0$. Let

$$\epsilon_k := \rho\mu, \;\; k \geq 0, \;\text{ for some } \rho \in (0, 1).$$

Thus conditions (7.6) and (7.10) hold, where $p = 1$ and $\mu_0 := (1 - \rho)\mu$, and the upper bound on the number of outer iterations k is given by (7.12) and (7.9). For each $k \geq 0$, in order to bound the inner iterations $l \geq 0$, Corollary 3.11 applies with $q_0 = p_0 = \underline{p} = 1$ since we employ Algorithm 1 with l_2 in Φ_{k+1} to solve problem (1.1) with $\delta := \delta_{k+1}$, starting from $v_{k,0} := v_k$, where v_k is strictly feasible for the $(k + 1)$th perturbed problem. For each $k \geq 0$, let us choose $\theta := \theta_k$ in the neighbourhood condition (2.13) of Algorithm 1 to satisfy $\theta_k \leq 4/5$. Then Corollary 3.11 provides that for each $k \geq 0$,

i) If $\rho \geq \theta_k/2$, then Algorithm 1 generates an iterate $v_{k,l}$ with $\Phi_{k+1}(v_{k,l}) \leq \epsilon_k$, for all l such that

$$l \geq l_0^\rho := \max\left\{0, \left\lceil \frac{2}{\theta_k\mu}(\Phi_{k+1}(v_k) - \epsilon_k)\max\left\{2, \frac{\Phi_{k+1}(v_k)}{(1-\theta_k)\mu}\right\}\right\rceil\right\}. \tag{7.13}$$

ii) Let $l_0 := l_0^\rho$ for $\rho = \theta_k/2$. When $\rho < \theta/2$, then Algorithm 1 generates $v_{k,l}$ such that $\Phi_{k+1}(v_{k,l}) \leq \epsilon_k$ for all l such that

$$l \geq l_0 + \left\lceil \frac{1}{\log 2}\log\left(\frac{\log(\rho/\theta_k)}{\log r}\right)\right\rceil, \tag{7.14}$$

where $r := \min\{1/2, \sqrt{(1-\theta_k)/\theta_k}\}$.

For overall complexity and practical computational purposes, it would be very useful to relate the values of $\Phi_{k+1}(v_k)$ and $\Phi_k(v_k)$ (recall that $\Phi_k(v_k) \leq \rho\mu$), and those of θ_{k+1} and θ_k. In this sense, we remark that (5.18) and (5.19) imply that

$$(x_k^i + \lambda_{k+1}^i)(s_k^i + \gamma_{k+1}^i) - \mu = (x_k^i + \lambda_k^i)(s_k^i + \gamma_k^i) - \mu, \;\; \forall k,\, i\colon (x_k^i, s_k^i) > 0 \text{ and } (x_{k-1}^i, s_{k-1}^i) > 0,$$

and

$$(x_k^i + \lambda_{k+1}^i)(s_k^i + \gamma_{k+1}^i) - \mu = (1-\zeta)[(x_k^i + \lambda_k^i)(s_k^i + \gamma_k^i) - \mu] - \zeta\mu, \;\; \forall k,\, i\colon x_k^i \leq 0 \text{ or } s_k^i \leq 0,$$

and

$$(x_k^i + \lambda_{k+1}^i)(s_k^i + \gamma_{k+1}^i) - \mu = (1-\zeta)^2[(x_k^i + \lambda_k^i)(s_k^i + \gamma_k^i) - \mu] - (2-\zeta)\zeta\mu, \;\; \forall k,\, i\colon (x_k^i, s_k^i) \leq 0.$$

Due to the generality of the updating rules for the perturbation parameters, however, it is difficult to estimate the above componentwise error in the centrality conditions in the cases when $x_k^i > 0$

and $x^i_{k-1} \le 0$, i. e., when $\lambda^i_{k+1} = 0$ and $\lambda^i_k > 0$, and similarly for s^i_k. Note that the last three displayed relations imply that the closer to zero ζ is, the closer together are the errors in the centrality equations, implying that once $\Phi_k(v_k) \le \rho\mu$ holds, it will take only a small number of inner iterations l to achieve $\Phi_{k+1}(v_{k+1}) \le \rho\mu$. Then, however, the number of outer iterations k increases since $k \sim 1/\zeta$ due to (7.12).

The asymptotic detection of the primal-dual implicit equalities is also possible when we apply Algorithm 2_ϵ to (PD).

Corollary 7.5 *Let the matrix A have full row rank, and the problems (P) and (D) be feasible. Assume that* (7.6) *and* (7.10) *hold. Then the characterizations* (6.89)—(6.92) *of the sets of implicit and strict inequalities of (PD) also hold for the iterates* $(v_k) = (x_k, y_k, s_k)$ *generated by Algorithm* 2_ϵ*.*

Proof. A careful investigation of the proofs in Section 6.2.1 shows that they continue to hold when the inexact condition (7.5), with the choices (7.6) and (7.10), is employed instead of the n nonlinear equations of (5.17). This is essentially due to the iterates v_k of Algorithm 2_ϵ satisfying the linear equations and strict inequalities in (5.17), to the updating schemes (5.18) and (5.19) being the same, and to the system (5.10) being well-defined and having a unique solution for all $\mu > 0$ and even, for any vector $w > 0$, if μe on the right-hand side of its nonlinear equations is replaced by w (see [30], page 224). □

The results in Corollary 6.13 also hold for the iterates (v_k) and the parameters (λ_k, γ_k) of Algorithm 2_ϵ, providing an asymptotic infeasibility detection criteria for (PD).

8 Numerical experience

We now briefly describe our initial numerical experience with the algorithms we developed and analysed in Sections 2–7. We have implemented Algorithm 2, more precisely Algorithm 2_ϵ, which naturally contains Algorithm 1 as a sub-procedure, as a Fortran 95 module `WCP` (`Well-Centred Point`) as part of the upcoming release 2.0 of the nonlinear optimization library GALAHAD [17].

`WCP` is actually designed to find a well-centred point for the general linear program

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; c^T x \;\; \text{subject to} \;\; c_l \le Ax \le c_u \;\; \text{and} \;\; x_l \le x \le x_u \tag{8.1}$$

and its dual. Here any or all of the components of the bounds c_l, c_u, x_l and x_u may be identical or infinite, thus allowing equality constraints and fixed or free variables. Although the details and bookkeeping are more complicated than for (P) and (D), the underlying method is identical, so here we simply describe the details in the context of (P) and (D).

The dominant cost per iteration of Algorithm 1 is, of course, the solution of the linear system (2.2). As is traditional for interior point methods, the variables $\dot{s}_k$ may be eliminated to give

$$\begin{pmatrix} X_k^{-1} S_k & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_k \\ -\dot{y}_k \end{pmatrix} = - \begin{pmatrix} c - A^T y_k - \mu X_k^{-1} e \\ A x_k - b \end{pmatrix} \tag{8.2}$$

and recovered via

$$\dot{s}_k = -X_k^{-1} S_k \dot{x}_k - s_k + \mu X_k^{-1} e.$$

The system (8.2) may be solved either using a symmetric, indefinite factorization of

$$\begin{pmatrix} X_k^{-1}S_k & A^T \\ A & 0 \end{pmatrix} \tag{8.3}$$

or that of the positive-definite Schur complement

$$AS_k^{-1}X_kA^T. \tag{8.4}$$

Factors of either are obtained using the GALAHAD module `SILS`, which is built on top of the HSL [18] package `MA27` [12]—`SILS` may be trivially replaced by the more powerful (but commercial) package `MA57` [11] if desired. In `WCP`, the default factorization is that of the Schur complement (8.4), but with a switch to that of (8.3) if A has any relatively dense columns or if (8.4) proves to be too ill-conditioned.

Although other strategies are available within `WCP`, the default is to fix the target μ to 1000 throughout the calculation. The relaxation weights λ_0 and γ_0 are set to 0.00001, and the initial point set so that the primal and dual variables x_0 and s_0 are 0.1 from their (perturbed) bounds—for the more general problem (8.1), the mid point between the lower and upper bounds is taken if the distance between the bounds is smaller than 0.2. Each major iteration in Step 3 of Algorithm 2 is terminated when the l_∞-norm of the residuals (5.17) is smaller than a prescribed stopping tolerance (in other words, we employ the condition (7.1)), here 0.000001, and the overall algorithm is terminated when the required centrality conditions (1.1) are violated (component-wise) by at most 0.000001 or if the violation fails to decrease by a factor 0.1 over 100 consecutive iterations.

No preprocessing of the problems is performed by default, aside from identifying and removing dependent linear equality constraints as required by Assumption A1). We do this by factorizing

$$\begin{pmatrix} \alpha I & A^T \\ A & 0 \end{pmatrix} = LBL^T$$

using `SILS`, and assessing rank deficiency from tiny eigenvalues of the one-by-one and two-by-two diagonal blocks that make up B—a value $\alpha = 0.01$ is used by default. We recognize that this is not as robust as, for example, a singular-value decomposition or a rank-revealing QR factorization, but fortunately, it has proved remarkably reliable in our tests.

All of our experiments were performed on a single processor of a 3.06 GHz Dell Precision 650 Workstation. The codes were compiled using full optimization with the Intel ifort compiler, and the computations were performed in double precision.

We consider the complete set of linear programming Netlib and other linear programming test problems as distributed with CUTEr [16]. Our intention here is simply to persuade the reader that our algorithm achieves its objectives. In particular, for those problems having strictly feasible points, we aim to find such a point efficiently, and for those whose feasible set is nonempty but not full-dimensional, we wish to identify the (number of) implicit (fixed or free) constraints/variables and compute a point in the relative interior of the primal-dual feasible set. We also want to identify when a problem is infeasible or unbounded.

We have made no attempt to tune the many algorithmic parameters to individual problems, nor to find a range that is globally better than the current defaults. This is work in progress and we anticipate improvements before the code is finally released.

In Table 8.1 we report the results of applying WCP, using default values, to our test set. For each problem, we give the number of variables, n, and constraints, m, the numbers of implicit constraints on the variables, x_{im} (the variables that are actually fixed), and on their duals, s_{im} (the variables that are actually free) the numbers of inequality constraints that are implicitly fixed, c_{im}, and the number that are actually free from one of their bounds, y_{im}. In addition, we record the total number of inner iterations (factorizations) performed, the exit status and the total CPU time taken in seconds.

Table 8.1: CUTEr LP test examples

name	n	m	x_{im}	s_{im}	c_{im}	y_{im}	its	status	time
25FV47	1571	821	0	4	0	0	36	ok	0.58
80BAU3B	9799	2262	63	8	47	8	70	ok	2.57
ADLITTLE	97	56	1	0	0	0	19	ok	0.01
AFIRO	32	27	0	0	0	0	14	ok	0.00
50AGG2	302	516	1	0	1	0	85	ok	0.48
AGG3	302	516	1	0	1	0	85	ok	0.49
AGG	163	488	42	0	11	0	151	ok	0.42
BANDM	472	305	20	0	0	0	91	ok	0.25
BCDOUT	5940	5414	12	3335	6	1	223	infeas	65.10
BEACONFD	262	173	78	2	0	0	327	ok	0.68
BLEND	83	74	0	0	0	0	17	ok	0.01
BNL1	1175	643	43	0	11	0	99	ok	0.64
BNL2	3489	2324	41	375	22	104	184	ok	7.64
BOEING1	384	351	4	0	5	0	64	ok	0.25
BOEING2	143	166	0	0	14	0	33	ok	0.05
BORE3D	315	233	107	0	10	0	313	ok	0.64
BRANDY	249	220	23	10	9	0	205	ok	0.45
CAPRI	353	271	0	14	0	0	26	ok	0.07
CYCLE	2857	1903	325	12	16	2	79	ok	2.56
CZPROB	3523	929	336	0	11	0	246	ok	2.23
D2Q06C	5167	2171	9	1	1	1	44	ok	3.95
D6CUBE	6184	415	0	0	0	0	32	ok	1.47
DEGEN2	534	444	68	0	166	0	326	ok	2.15
DEGEN3	1818	1503	257	0	458	0	377	ok	27.55
DEGENLPA	20	15	4	0	0	0	144	ok	0.02
DEGENLPB	20	15	4	0	0	0	146	ok	0.02
DFL001	12230	6071	5600	12	13	0	183	inacc	494.24
E226	282	223	19	2	10	5	79	ok	0.19
ETAMACRO	688	400	47	0	1	0	41	ok	0.20
EXTRASIM	2	1	0	1	0	0	2	ok	0.00
FFFFF800	854	524	19	0	0	0	63	ok	0.52
FINNIS	614	497	25	17	11	11	96	ok	0.36
FIT1D	1026	24	0	0	0	0	19	ok	0.09
FIT1P	1677	627	0	0	0	0	26	ok	0.25
FIT2D	10500	25	0	0	0	0	24	ok	2.28
FIT2P	13525	3000	0	0	0	0	24	ok	8.45
FORPLAN	421	161	118	0	6	0	58	ok	0.14
GANGES	1681	1309	100	0	0	0	92	ok	1.20
GFRD-PNC	1092	616	26	0	0	0	32	ok	0.11
GOFFIN	51	50	0	51	0	0	17	ok	0.02
GREENBEA	5405	2392	1039	28	9	0	521	inacc	25.38
GREENBEB	5405	2392	1040	31	9	0	531	inacc	48.27
GROW15	645	300	0	0	0	0	31	ok	0.14
GROW22	946	440	0	0	0	0	32	ok	0.21
GROW7	301	140	0	0	0	0	31	ok	0.07
ISRAEL	142	174	0	0	0	0	29	ok	0.11
KB2	41	43	0	0	0	0	20	ok	0.01
LINSPANH	97	33	18	0	1	0	23	ok	0.01
LOTFI	308	153	0	2	0	0	30	ok	0.04
MAKELA4	21	40	0	21	0	0	19	ok	0.00

Table 8.1: CUTEr LP test examples (continued)

name	n	m	x_{im}	s_{im}	c_{im}	y_{im}	its	status	time
MAROS-R7	9408	3136	0	0	0	0	15	ok	14.42
MAROS	1443	846	322	14	41	0	283	ok	3.39
MODSZK1	1620	687	1	2	0	0	45	ok	0.18
MPSBCD03	5940	5414	42	408	0	1	201	infeas	47.26
NESM	2923	662	0	0	0	0	53	ok	0.68
OET1	3	1002	0	3	0	0	89	ok	0.20
OET3	4	1002	0	4	0	0	84	ok	0.21
PEROLD	1376	625	28	88	0	0	64	ok	0.76
PILOT4	1000	410	17	88	0	0	48	ok	0.34
PILOT87	4883	2030	22	9	2	9	66	ok	56.23
PILOT-JA	1988	940	80	88	18	0	57	ok	1.30
PILOTNOV	2172	975	73	0	26	0	45	ok	2.10
PILOT	3652	1441	47	0	12	0	66	ok	9.51
PILOT-WE	2789	722	60	80	5	0	82	ok	0.86
PT	2	501	0	2	0	0	69	ok	0.07
QAP12	8856	3192	0	0	397	0	41	ok	127.68
QAP15	22275	6330	0	0	630	0	50	ok	1355.51
QAP8	1632	912	0	0	170	0	26	ok	2.23
QPBD_OUT	263	211	0	2	0	5	24	ok	0.06
READING2	6003	4000	0	2000	0	0	3	ok	0.14
RECIPELP	180	91	19	24	3	15	232	ok	0.14
S277-280	4	4	0	0	0	0	12	ok	0.00
SC105	103	105	0	0	0	0	14	ok	0.01
SC205	203	205	1	0	0	0	35	ok	0.04
SC50A	48	50	0	0	0	0	12	ok	0.01
SC50B	48	50	0	0	0	0	12	ok	0.00
SCAGR25	500	471	0	0	0	0	22	ok	0.06
SCAGR7	140	129	0	0	0	0	16	ok	0.01
SCFXM1	457	330	10	8	2	0	76	ok	0.23
SCFXM2	914	660	20	16	4	0	77	ok	0.48
SCFXM3	1371	990	30	24	6	0	72	ok	0.70
SCORPION	358	388	57	0	40	0	170	ok	0.35
SCRS8	1169	490	34	16	6	16	77	ok	0.30
SCSD1	760	77	0	0	0	0	6	ok	0.01
SCSD6	1350	147	0	0	0	0	6	ok	0.03
SCSD8	2750	397	0	0	0	0	8	ok	0.06
SCTAP1	480	300	0	0	0	0	14	ok	0.03
SCTAP2	1880	1090	0	0	0	0	13	ok	0.13
SCTAP3	2480	1480	0	0	0	0	14	ok	0.18
SEBA	1028	515	126	0	0	0	131	ok	0.42
SHARE1B	225	117	0	0	0	0	30	ok	0.03
SHARE2B	79	96	0	0	0	0	22	ok	0.02
SHELL	1775	536	0	0	1	0	80	ok	0.35
SHIP04L	2118	402	201	0	0	0	70	ok	0.38
SHIP04S	1458	402	89	0	0	0	58	ok	0.23
SHIP08L	4283	778	1126	0	0	0	206	ok	2.02
SHIP08S	2387	778	553	0	0	0	180	ok	1.19
SHIP12L	5427	1151	1066	0	0	0	202	ok	3.10
SHIP12S	2763	1151	360	0	0	0	142	ok	1.31
SIERRA	2036	1227	0	0	10	0	24	ok	0.25
SIPOW1M	2	2000	0	2	0	0	22	ok	0.10
SIPOW1	2	2000	0	2	0	0	23	ok	0.11
SIPOW2M	2	2000	0	2	0	0	26	ok	0.13
SIPOW2	2	2000	0	2	0	0	27	ok	0.12
SIPOW3	4	2000	0	4	0	0	87	ok	0.40
SIPOW4	4	2000	0	4	0	0	87	ok	0.43
SSEBLIN	194	72	0	0	0	0	32	ok	0.02
STAIR	467	356	0	8	0	0	36	ok	0.13
STANDATA	1075	359	13	0	19	0	87	ok	0.29
STANDGUB	1184	361	112	1	19	0	126	ok	0.43

Table 8.1: CUTEr LP test examples (continued)

name	n	m	x_{im}	s_{im}	c_{im}	y_{im}	its	status	time
STANDMPS	1075	467	13	0	19	0	67	ok	0.27
STOCFOR1	111	117	0	0	0	0	19	ok	0.02
STOCFOR2	2031	2157	0	0	0	0	21	ok	0.36
STOCFOR3	15695	16675	0	0	0	0	37	ok	6.13
TESTDECK	14	15	0	10	0	10	129	infeas	0.02
TFI2	3	101	0	3	0	0	27	ok	0.01
TRUSS	8806	1000	0	0	0	0	10	ok	0.36
TUFF	587	333	21	2	0	0	85	ok	0.38
VTP-BASE	203	198	67	1	71	0	267	ok	0.34
WOOD1P	2594	244	1094	0	0	0	315	ok	13.25
WOODW	8405	1098	3020	0	0	0	669	inacc	25.24

The exit status "inacc" simply means that the required accuracy was not achieved—this is an inevitable consequence of the ill-conditioning which arises for problems for which there is no interior. Of the problems that fail in this way, GREENBEA, GREENBEB and WOODW succeed if the convergence tolerance is raised to 0.00005. Unfortunately, DFL001 appears to be highly ill-conditioned, and thus far we have not been able to solve this problem (the simplex code MINOS [26] reports that the problem is feasible). Infeasible problems result in an exit status "infeas"; the infeasible BCDOUT and MPSBCD03 and dual infeasible TESTDECK end this way.

The most surprising aspect, for us, was the large number of problems which have no primal-dual strictly feasible points (whose feasible sets are not full-dimensional). Since most (but not all, see for example, [41]) convergence analyses for interior-point methods for linear programming is predicated on the IPM conditions, one can only conclude that these algorithms work in spite of the lack of these desirable conditions and not because of the convergence theory that is used to justify them! Of course commercial codes apply careful pre-solving techniques (e. g., [14]) and perhaps these remove implicit constraints either accidentally or by design (see for example, the last paragraph of this section and Table 8.2).

Of those (few) problems that have an interior, the number of iterations required is generally modest. For the others, often more iterations than we would like are required, but this is an artifact of needing a number of (increasingly ill-conditioned) minor iterations for convergence.

We had hoped that a strategy of letting a convenient initial point x_0 determine the relaxation weights λ_0 and γ_0, such as that outlined in Section 6.2, would be a good one. Unfortunately, this frequently led to large initial weights, and subsequently many major iterations were required.

We have also applied WCP to the NETLIB test problems [27] after they have been preprocessed using the CPLEX preprocessor [8]. The CPLEX preprocessor fails to find all, and indeed sometimes many, of the implicit equalities for 25 out of the 98 NETLIB problems, the results for the former being reported in Table 8.2. The numerics suggests that WCP may be employed as an algorithmic preprocessor for test problems, possibly after they have already been preprocessed by a common presolver, in order to ensure that all implicit equalities have been removed. The strong theoretical guarantees of convergence of Algorithm 2 that is implemented in WCP make this software a rare occurence among existing preprocessors.

Table 8.2: Preprocessed NETLIB LP test examples

name	n	m	x_{im}	s_{im}	c_{im}	y_{im}	its	status	time
80BAU3B	8510	1789	12	0	6	0	64	ok	1.99
BOEING1	418	287	0	0	2	0	55	ok	0.17

Table 8.2: Preprocessed NETLIB LP test examples (continued)

name	n	m	x_{im}	s_{im}	c_{im}	y_{im}	its	status	time
BOEING2	160	122	0	0	14	0	31	ok	0.03
D2Q06C	4561	1855	5	0	1	0	41	ok	2.87
DEGEN2	473	382	68	0	162	0	334	ok	2.02
DEGEN3	1721	1406	257	0	457	0	425	ok	27.82
DFL001	9052	3861	15	0	3	0	160	ok	276.41
FINNIS	363	296	2	4	0	8	47	ok	0.11
FORPLAN	421	161	118	0	6	0	58	ok	0.16
GREENBEA	3048	1015	0	2	0	0	163	ok	3.60
MAROS	809	524	15	0	2	0	94	ok	0.59
PEROLD	1074	503	3	0	0	0	56	ok	0.47
PILOT4	770	348	4	0	0	0	38	ok	0.17
PILOT-JA	1369	708	27	0	2	0	53	ok	0.89
PILOTNOV	1686	748	27	0	2	0	44	ok	0.78
PILOT	3066	1204	1	0	0	0	61	ok	8.76
PILOT-WE	2346	602	42	0	5	0	74	ok	0.67
RECIPELP	71	48	0	1	0	3	97	ok	0.02
SCRS8	799	174	0	6	0	14	31	ok	0.07
SHELL	1157	236	0	0	1	0	72	ok	0.19
SHIP08L	3099	470	3	0	0	0	27	ok	0.18
SHIP08S	1526	234	3	0	0	0	27	ok	0.09
SIERRA	1723	877	0	0	10	0	20	ok	0.16
WOOD1P	1716	170	286	0	0	0	226	ok	9.17
WOODW	4006	551	8	0	0	0	112	ok	22.44

9 Conclusions

We presented a 'Phase I' strategy — Algorithm 2, and an inexact variant Algorithm 2_ϵ, that contain Algorithm 1 as a sub-procedure — for feasible primal-dual interior point methods for linear programming. It computes a point in the relative interior of the primal-dual feasible set of (PD) or asymptotically detects that this set is empty. In the former case, if additionally, the IPM conditions are satisfied, Algorithm $2/2_\epsilon$ takes a finite number of outer iterations to generate such a point on the central path of (PD), while Algorithm 2_ϵ takes a finite number of total iterations. Else, when (PD) is feasible, but not strictly feasible, Algorithm $2/2_\epsilon$ is able to asymptotically detect all the primal-dual implicit equalities of the primal-dual feasible set from the behaviour of the iterates. We further use this information to transform problems (P) and (D) into equivalent ones which satisfy the IPM conditions, by eliminating the primal variables that we have identified as always being at their bounds and freeing those that correspond to dual implicit equalities. Moreover, the finite components of any limit point of the iterates provide a strictly feasible point of these transformed problems that belongs to their central path. It is to these reduced problems that a feasible interior point method may then be applied in order to solve them to optimality. Successful numerical experiments on LP problems from the NETLIB and CUTEr collections indicate that Algorithm $2/2_\epsilon$ can be employed as an algorithmic preprocessor for removing implicit equalities, with theoretical guarantees of convergence. We plan to further our numerical experience with Algorithm $2/2_\epsilon$ by comparing the efficiency of our initialization strategy when embedded into a feasible interior point code with existing infeasible interior point methods (whose convergence relies on the satisfaction of the IPM conditions).

An aspect of our results for Algorithm 1 that deserves further attention concerns casting the complexity bounds in Corollary 3.11 in a form that is common for such results in interior point methods.

From a computational point of view, we would also like to derive proven nonasymptotic criteria for detecting implicit equalities using Algorithm $2/2_\epsilon$. Tightening the existing flexibility in the choice of the perturbation parameters may be a way to tackle this issue. Such a result will further strengthen the reliability of our algorithmic preprocessor.

Inspired by some recent computational work in [4], we believe that our approach may offer a theoretically-reliable alternative for warmstarting interior point methods for linear programming and those nonlinear programs that contain (some) linear constraints. We plan to address this issue in the near future.

Acknowledgements. We would like to thank William Hager for providing us with the preprocessed NETLIB problems that we used for the tests reported in Table 8.2, and Raphael Hauser for useful discussions.

References

[1] E. D. Andersen. Certificates of primal or dual infeasibility in linear programming. *Comput. Optim. Appl.*, 20(2):171–183, 2001.

[2] E. D. Andersen and K. D. Andersen. The APOS linear programming solver: an implementation of the homogeneous algorithm. CORE Discussion paper 9337, CORE, Universite Catholique de Louvain, 1997.

[3] K. M. Anstreicher. On interior algorithms for linear programming with no regularity assumptions. *Oper. Res. Lett.*, 11(4):209–212, 1992.

[4] H. Benson and D. Shanno. An exact primal-dual penalty method approach to warmstarting interior point methods for linear programming. Working paper, to appear in Computational Optimization and Applications.

[5] S. C. Billups and M. C. Ferris. Convergence of an infeasible interior-point algorithm from arbitrary positive starting points. *SIAM J. Optim.*, 6(2):316–325, 1996.

[6] R. J. Caron. Redundancy in nonlinear programs. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, volume 5, pages 1–6. Kluwer Academic Publishers, Dordrecht, 2001.

[7] C. Cartis. On interior point methods for linear programming, 2004. PhD thesis, Department of Applied Mathematics and Theoretical Physics, University of Cambridge.

[8] CPLEX. See http://www.ilog.com/products/cplex/.

[9] V. DeMiguel, M. P. Friedlander, F. J. Nogales, and S. Scholtes. A two-sided relaxation scheme for mathematical programs with equilibrium constraints. *SIAM J. Optim.*, 16(2):587–609, 2005.

[10] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1996.

[11] I. S. Duff. MA57 - a code for the solution of sparse symmetric definite and indefinite systems. *ACM Transactions on Mathematical Software*, 30(2):118–144, 2004.

[12] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.

[13] R. M. Freund and S. Mizuno. Interior point methods: current status and future directions. In *High performance optimization*, volume 33 of *Appl. Optim.*, pages 441–466. Kluwer Acad. Publ., Dordrecht, 2000.

[14] J. Gondzio. Presolve analysis of linear programs prior to applying an interior point method. *INFORMS J. Comp.*, 9(1):73–91, 1994.

[15] C. C. Gonzaga. Path-following methods for linear programming. *SIAM Rev.*, 34(2), 1992.

[16] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394, 2003.

[17] N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software*, 29(4):353–372, 2003.

[18] HSL. A collection of Fortran codes for large-scale scientific computation, 2004. See http://hsl.rl.ac.uk/.

[19] L. G. Khachiyan. A polynomial algorithm in linear programming (Russian). *Dokl. Akad. Nauk SSSR*, 244(5):1093–1096, 1979.

[20] M. Kojima, N. Megiddo, and S. Mizuno. A primal-dual infeasible-interior-point algorithm for linear programming. *Math. Programming*, 61(3, Ser. A):263–280, 1993.

[21] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point algorithm for linear programming. In *Progress in mathematical programming (Pacific Grove, CA, 1987)*, pages 29–47. Springer, New York, 1989.

[22] J.-L. Lassez and M. J. Maher. On Fourier's algorithm for linear arithmetic constraints. *J. Automat. Reason.*, 9(3):373–379, 1992.

[23] N. Megiddo. Pathways to the optimal set in linear programming. In *Progress in mathematical programming (Pacific Grove, CA, 1987)*, pages 131–158. Springer, New York, 1989.

[24] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. Optim.*, 2(4):575–601, 1992.

[25] S. Mizuno and M. J. Todd. On two homogeneous self-dual approaches to linear programming and its extensions. *Math. Programming*, 89(3, Ser. A):517–534, 2001.

[26] B. A. Murtagh and M. A. Saunders. MINOS 5.5 USER'S GUIDE. Technical Report SOL83-20R, Department of Operations Research, Stanford University, Stanford, California 94305, USA, 1983. Revised 1998.

[27] NETLIB. The NETLIB collection of linear programming test problems. See http://www.netlib.org/lp/data/.

[28] P. Refalo. Approaches to incremental detection of implicit equalities with the revised simplex method. In *Principles of Declarative Programming*, volume 1490 of *Lecture Notes in Comp. Sci.*, pages 481–497. Springer, Heidelberg, 1998.

[29] R. T. Rockafellar. *Convex analysis.* Princeton University Press (reprint of the 1970 original), Princeton, 1997.

[30] C. Roos, T. Terlaky, and J.-P. Vial. *Theory and algorithms for linear optimization.* John Wiley & Sons Ltd., Chichester, 1997.

[31] A. Schrijver. *Theory of linear and integer programming.* John Wiley & Sons Ltd., Chichester, 1986.

[32] G. Sierksma and G. A. Tijssen. Degeneracy degrees of constraint collections. *Math. Methods Oper. Res.*, 57(3):437–448, 2003.

[33] G. Sonnevend. An "analytical centre" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In *System modelling and optimization (Budapest, 1985)*, volume 84 of *Lecture Notes in Control and Inform. Sci.*, pages 866–875. Springer, Berlin, 1986.

[34] J. Telgen. Identifying redundant constraints and implicit equalities in systems of linear constraints. *Management Sci.*, 29(10):1209–1222, 1983.

[35] M. J. Todd. Detecting infeasibility in infeasible-interior-point methods for optimization. In *Foundations of computational mathematics: Minneapolis, 2002*, volume 312 of *London Math. Soc. Lecture Note Ser.*, pages 157–192. Cambridge Univ. Press, Cambridge, 2004.

[36] S. J. Wright. *Primal-dual Interior-Point Methods.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1997.

[37] F. Wu, S. Wu, and Y. Ye. On quadratic convergence of the $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Ann. Oper. Res.*, 87:393–406, 1999.

[38] Q. Xu. On new linear programming algorithms: new sequential penalty function method and two point barrier function method, 1991. PhD thesis, Institute of Nuclear Technology, Tsinghua University, Beijing, China.

[39] X. Xu, P.-F. Hung, and Y. Ye. A simplified homogeneous and self-dual linear programming algorithm and its implementation. *Ann. Oper. Res.*, 62:151–171, 1996.

[40] Y. Ye. *Interior Point Algorithms:Theory and Analysis.* John Wiley and Sons, New York, 1997.

[41] Y. Ye, M. J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Math. Oper. Res.*, 19(1):53–67, 1994.

[42] Y. Zhang. On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM J. Optim.*, 4(1), 1994.