Hamburger Beiträge
zur Angewandten Mathematik

Predictor-Corrector Smoothing Methods for the Solution of Linear Programs

Stephan Engelke and Christian Kanzow

Reihe A
Preprint 153
March 2000

Hamburger Beiträge zur Angewandten Mathematik

Reihe A	Preprints
Reihe B	Berichte
Reihe C	Mathematische Modelle und Simulation
Reihe D	Elektrische Netzwerke und Bauelemente
Reihe E	Scientific Computing
Reihe F	Computational Fluid Dynamics and Data Analysis

PREDICTOR-CORRECTOR SMOOTHING METHODS FOR THE SOLUTION OF LINEAR PROGRAMS[1]

Stephan Engelke and Christian Kanzow

University of Hamburg
Department of Mathematics
Center for Optimization and Approximation
Bundesstrasse 55
20146 Hamburg
Germany
e-mail: engelke@math.uni-hamburg.de
kanzow@math.uni-hamburg.de

March 30, 2000

Abstract. We consider a class of predictor-corrector smoothing-type methods for the solution of linear programs. These methods follow a certain smoothing path in order to solve the linear program. This smoothing path is sometimes closely related to the central path used by interior-point methods. We establish global convergence for our class of methods. Moreover, one particular member of our class of algorithms is shown to be locally quadratically convergent under reasonable assumptions (which do not imply that the solution set consists of a single element). Finally, numerical results are reported for the whole netlib test suite.

Key Words. Linear programs, optimality conditions, path-following, smoothing method, global convergence, quadratic convergence.

[1]This research was supported by the DFG (Deutsche Forschungsgemeinschaft).

1 Introduction

Consider the linear program in primal form

$$\min c^T x \quad \text{s.t.} \quad Ax = b, \;\; x \geq 0, \tag{1}$$

where $A \in \mathbb{R}^{m \times n}, c \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$ are the given data. Throughout this manuscript, we assume without loss of generality that the matrix A has full rank.

The solution of a linear program like (1) is one of the most fundamental and still challenging problems in numerical mathematics. There is hardly any other mathematical problem that has to be solved in practice as often as a linear program. The development of robust and efficient methods for the solution of linear programs is therefore still a highly important research area.

One of the most frequently used solvers is Dantzig's simplex method or one of its variants, see, e.g., [1, 13]. It is typically applied to the primal linear program (1) or to the corresponding dual problem

$$\max b^T \lambda \quad \text{s.t.} \quad A^T \lambda \leq c,$$

which, in turn, can be rewritten as

$$\max b^T \lambda \quad \text{s.t.} \quad A^T \lambda + s = c, \;\; s \geq 0, \tag{2}$$

where s denotes a nonnegative slack variable. The basic idea of the simplex method is to enumerate the extreme points of the feasible sets from (1) or (2) in an intelligent way until an optimal solution has been found.

On the other hand, so-called interior-point methods try to solve a linear program by moving through the (relative) interior of the feasible sets. In this way, they get rid of the combinatorial nature of the simplex method, and it is possible to prove polynomial complexity results. The reader is referred to [32, 28, 35] for further details about interior-point methods.

Here, we only note that the currently most successful interior-point methods are based on the solution of the (primal-dual) optimality conditions

$$\begin{array}{rcl} A^T \lambda + s & = & c, \\ Ax & = & b, \\ x_i \geq 0, \;\; s_i \geq 0, \;\; x_i s_i & = & 0 \quad \forall i = 1, \ldots, n. \end{array} \tag{3}$$

Since these optimality conditions are precisely the KKT conditions of both the primal linear program (1) and the dual linear program (2), it follows that

$$(1) \text{ has a solution} \iff (2) \text{ has a solution} \iff (3) \text{ has a solution.}$$

This simple observation justifies to consider the system (3) in order to solve the underlying linear program (1).

The method to be presented here will also deal with the optimality conditions (3). The basic idea of our method is to reformulate the system (3) (consisting of equations and inequalities) as a nonlinear system of equations only, say

$$\Phi(x, \lambda, s) = 0. \tag{4}$$

However, it turns out that Φ is nonsmooth, so we introduce a smoothing parameter $\tau > 0$ and replace the nonsmooth system (4) by a sequence of smooth systems

$$\Phi_\tau(x, \lambda, s) = 0, \tag{5}$$

where Φ_τ is a suitable approximation of Φ. The precise definitions of Φ and Φ_τ are given in Section 2. We also note that this smoothing approach has some similarities with interior-point methods, see Section 2 and, for a more detailed discussion, the paper [15].

The idea to use such a smoothing-type method is not new. In fact, there are quite a few recent papers [2, 3, 5, 6, 8, 9, 10, 11, 17, 18, 19, 20, 22, 26, 30, 31, 33, 34] dealing with smoothing-type methods for the solution of linear complementarity problems (note that a linear program is just a special case of a linear complementarity problem). However, many times these papers use assumptions which are typically viewed as being too strong when specialized to linear programs, e.g., a certain matrix has to be of R_0-type or a sequence of certain matrices has to be uniformly nonsingular, implying that the solution set of the optimality conditions (3) is a singleton. In addition, numerical results for linear programs are usually not presented in these papers.

The only two exceptions we are currently aware of are the two references [4, 15]. In [4], Burke and Xu present a smoothing-type method for linear programs which, theoretically, use relatively strong assumptions, but for which the authors present quite reasonable numerical results. Unfortunately, reference [4] is a conference talk and is not available as a manuscript. Most of the theoretical background, however, can be found in [5].

The second exception mentioned above is reference [15]. There, the authors present a so-called Jacobian smoothing method (based on ideas from Chen, Qi, and Sun [12]) for the solution of linear programs. Still relatively strong assumptions have to be imposed in order to prove local fast convergence, however, the numerical results in [15] are not too bad although not completely satisfactory.

In this manuscript, we combine smoothing-type methods with the Jacobian smoothing idea. Basically, the smoothing-type steps are used in order to monitor global convergence, while the Jacobian smoothing step guarantees local fast convergence. We stress at this point that the assumptions used in our local convergence theory (which, to some extend, is based on the recent paper [31] by Tseng) do not imply (neither explicitly nor implicitly) that the solution set of (3) is a singleton.

The organization of this manuscript is as follows: In Section 2, we introduce the reformulations (4) and their smooth approximations (5), and state some preliminary results. In particular, we show that the smooth systems (5) have a (unique) solution. Our predictor-corrector smoothing-type method is then introduced and discussed in some detail in Section 3. Section 4 contains the global convergence analysis for this method. A locally quadratically convergent variant of this method is the subject of Section 5. Section 6 then presents extensive numerical results for the method from Section 5. We close this manuscript with some final remarks in Section 7.

Some words about our notation: All vector norms are Euclidean norms. For matrices, we take the corresponding spectral norm which, in particular, is consistent with the Euclidean vector norm in the sense that $\|Az\| \leq \|A\| \, \|z\|$ for all matrices A and all vectors z of appropriate dimensions. If z is an r-dimensional vector and $I \subseteq \{1, 2, \ldots, r\}$ any given subset, then r_I denotes the subvector consisting of the components r_i for $i \in I$. Similarly, if A denotes an $r \times s$-matrix and $I \subseteq \{1, \dots, r\}$, $J \subseteq \{1, \dots, s\}$ are two given subsets, A_{IJ} denotes the submatrix with elements a_{ij} for $i \in I$ and $j \in J$. The notations $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are used for the minimal and maximal eigenvalue of a symmetric matrix $A \in \mathbb{R}^{n \times n}$. By

$$\text{dist}_S(z) := \inf_{w \in S} \|w - z\|$$

we denote the Euclidean distance of a vector z to a nonempty set S. Furthermore, for any given vector z, we write z_+ for the projection of z on the nonnegative orthant, i.e., the components of z_+ are given by $\max\{0, z_i\}$ for all i. Finally, we recall that a vector $\hat{w} = (\hat{x}, \hat{\lambda}, \hat{s})$ is called a *strictly feasible point* for the optimality conditions (3) if it satisfies $A^T \hat{\lambda} + \hat{s} = c$, $A\hat{x} = b$ and $\hat{x}_i > 0$, $\hat{s}_i > 0$ for all $i = 1, \dots, n$.

2 Smoothing Functions and Smoothing Paths

The optimality conditions (3) can be reformulated as a nonlinear and usually nonsmooth system of equations by using so-called NCP-functions.

Definition 2.1 *A mapping $\varphi : \mathbb{R}^2 \to \mathbb{R}$ is called an* NCP-function *if it has the property that*

$$\varphi(a, b) = 0 \Longleftrightarrow a \geq 0,\ b \geq 0,\ ab = 0,$$

i.e., an NCP-function is zero exactly on the two nonnegative half-axes.

Let φ be an arbitrary NCP-function and define $\Phi : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ by

$$\Phi(w) := \Phi(x, \lambda, s) := \begin{pmatrix} A^T \lambda + s - c \\ Ax - b \\ \phi(x, s) \end{pmatrix}, \tag{6}$$

where

$$\phi(x, s) := (\varphi(x_1, s_1), \dots, \varphi(x_n, s_n))^T \in \mathbb{R}^n.$$

Then it follows immediately from the definition of an NCP-function that a vector $w^* = (x^*, \lambda^*, s^*)$ satisfies the optimality conditions (3) if and only if w^* is a solution of the system

$$\Phi(w) = 0. \tag{7}$$

We next state some examples for NCP-functions.

Example 2.2 *The following functions are NCP-functions:*

(a) $\varphi(a, b) := 2 \min\{a, b\}$.

(b) $\varphi(a, b) := a + b - \sqrt{a^2 + b^2}$.

(c) $\varphi(a, b) := \lambda \left(a + b - \sqrt{a^2 + b^2}\right) + (1 - \lambda) a_+ b_+$ *with* $\lambda \in (0, 1)$ *fixed.*

(d) $\varphi(a,b) := \lambda 2\min\{a,b\} + (1-\lambda)a_+b_+$ *with* $\lambda \in (0,1)$ *fixed.*

The functions from Example 2.2 (a) and (b) are called *minimum function* and *Fischer-Burmeister function*, respectively, see [25, 16] (the factor 2 in the minimum function is used here only for cosmetical reasons). The mapping from Example 2.2 (c) is named *penalized Fischer-Burmeister function* in [7]. All these functions are well-known tools for the solution of nonlinear complementarity problems. The fourth example seems to be new, however, it is constructed in a similar way from the minimum function as the penalized Fischer-Burmeister function from the Fischer-Burmeister function. Therefore, we call the function from Example 2.2 (d) the *penalized minimum function.*

Since φ and therefore also Φ are nonsmooth, it is not possible to apply the classical Newton method to the system (7). Hence the next idea is to approximate φ and Φ by some suitable smooth functions φ_τ and Φ_τ, respectively, depending on a certain smoothing parameter $\tau > 0$. The next example gives some ways to approximate the four nonsmooth NCP-functions from Example 2.2.

Example 2.3 *The following four functions are smooth approximations of their nonsmooth counterparts from Example 2.2:*

(a) $\varphi_\tau(a,b) := a + b - \sqrt{(a-b)^2 + 4\tau^2}$.

(b) $\varphi_\tau(a,b) := a + b - \sqrt{a^2 + b^2 + 2\tau^2}$.

(c) $\varphi_\tau(a,b) := \lambda\left(a + b - \sqrt{a^2 + b^2 + 2\tau^2}\right) + (1-\lambda)\sqrt{a_+^2 b_+^2 + 2\tau^2}$.

(d) $\varphi_\tau(a,b) := \lambda\left(a + b - \sqrt{(a-b)^2 + 4\tau^2}\right) + (1-\lambda)\sqrt{a_+^2 b_+^2 + 4\tau^2}$.

The function from Example 2.3 (a) is typically called the *Chen-Harker-Kanzow-Smale smoothing function*, see [8, 20, 29]. However, there exist several other suitable approximations of the minimum function, see, e.g., Chen and Mangasarian [11]. The mapping φ_τ from Example 2.3 (b) was first introduced in [20], whereas it seems that the two functions from Example 2.3 (c) and (d) were not used before.

The next result shows that the four smoothing functions φ_τ defined in Example 2.3 are indeed good approximations of their corresponding nonsmooth NCP-functions from Example 2.2.

Lemma 2.4 *For each of the four NCP-functions* φ *from Example 2.2, there is a constant* $c = c_\varphi > 0$ *(independent of* τ *and (a,b)) such that*

$$|\varphi(a,b) - \varphi_\tau(a,b)| \le c\tau$$

for all $(a,b) \in \mathbb{R}^2$ *and all* $\tau > 0$.

Proof. For the minimum function, the statement follows from [21]. As for the other three examples, the proof is similar and is therefore omitted here. However, we stress that the constant $c = c_\varphi$ (depending on the NCP-function φ) is known and equal to $\sqrt{2}$ for the

Fischer-Burmeister and the penalized Fischer-Burmeister function, and equal to 2 for the minimum and the penalized minimum function. □

Having defined suitable approximations φ_τ of the four nonsmooth NCP-functions φ from Example 2.2, we can define a smooth approximation of the corresponding Φ from (6) by

$$\Phi_\tau(w) := \Phi_\tau(x, \lambda, s) := \begin{pmatrix} A^T\lambda + s - c \\ Ax - b \\ \phi_\tau(x, s) \end{pmatrix},$$

where

$$\phi_\tau(x, s) := (\varphi_\tau(x_1, s_1), \dots, \varphi_\tau(x_n, s_n))^T \in \mathbb{R}^n.$$

The relation between Φ and Φ_τ is made more precise in our next result.

Lemma 2.5 *For each of the four possible mappings Φ, there is a constant $\kappa = \kappa_\Phi > 0$ (independent of τ and w) such that*

$$\|\Phi(w) - \Phi_\tau(w)\| \leq \kappa\tau$$

for all $w = (x, \lambda, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ and all $\tau > 0$.

Proof. The result follows immediately from Lemma 2.4 by setting $\kappa := \sqrt{n}c$ with $c = c_\varphi > 0$ being the constant from Lemma 2.4. □

We stress that the constant κ from Lemma 2.5 is known for any of the four possible mappings Φ. This is important since later, in our local analysis from Section 5, we have to assume that a certain parameter β is larger than κ.

Since the approximations Φ_τ of Φ are smooth, the Jacobian matrices $\Phi_\tau'(w)$ exist at any point w and for any smoothing parameter $\tau > 0$, and an elementary calculation shows that

$$\Phi_\tau'(w) = \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ D_{a,\tau} & 0 & D_{b,\tau} \end{pmatrix}, \tag{8}$$

where we used the abbreviations

$$\begin{aligned} D_{a,\tau} &:= \operatorname{diag}\left(\dots, \tfrac{\partial\varphi_\tau}{\partial a}(x_i, s_i), \dots\right), \\ D_{b,\tau} &:= \operatorname{diag}\left(\dots, \tfrac{\partial\varphi_\tau}{\partial b}(x_i, s_i), \dots\right). \end{aligned} \tag{9}$$

The next result states that these Jacobian matrices are always nonsingular (whereas $\Phi'(w)$ itself might be singular even at differentiable points).

Proposition 2.6 *Let Φ_τ denote any of the four possible mappings defined above. Then the Jacobian matrices $\Phi_\tau'(w)$ are nonsingular for every vector $w = (x, \lambda, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ and any smoothing parameter $\tau > 0$.*

Proof. The statement has been shown in [15] in case of the minimum and Fischer-Burmeister functions, and it can be verified in a similar way for the two penalized functions. (Recall that A has full rank, and note that the diagonal matrices $D_{a,\tau}, D_{b,\tau}$ defined in (9) are positive definite for each of the four possible functions φ_τ from Example 2.3.) $\square$

The following results hold for the penalized minimum function as well as for the penalized Fischer-Burmeister function. It is not true, in general, for the minimum and the Fischer-Burmeister functions themselves.

Proposition 2.7 *Let φ be the penalized minimum or penalized Fischer-Burmeister function, and consider the set*

$$\mathcal{L}_c := \{w = (x, \lambda, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \mid A^T\lambda + s = c,\ Ax = b,\ \|\Phi(w)\| \leq c\},$$

where $c \in \mathbb{R}$ is any given constant. Assume that there exists a strictly feasible point for the optimality conditions (3). Then the set $\mathcal{L}_c$ is compact.

Proof. We only consider the case where φ denotes the penalized minimum function since the proof for the penalized Fischer-Burmeister function is similar.

Assume that $\mathcal{L}_c$ is unbounded for some $c \in \mathbb{R}$. Then there is a sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\} \subseteq \mathcal{L}_c$ with $\{\|w^k\|\} \to +\infty$ for $k \to +\infty$. The definition of the penalized minimum function immediately implies that there is no index $i \in \{1, \ldots, n\}$ such that $x_i^k \to -\infty$ or $s_i^k \to -\infty$ on a subsequence, since otherwise we would have $\varphi(x_i^k, s_i^k) \to -\infty$ which, in turn, would imply $\|\Phi(w^k)\| \to +\infty$ on a subsequence in contrast to $\|\Phi(w^k)\| \leq c$ for all $k \in \mathbb{N}$. Hence all components of the two sequences $\{x^k\}$ and $\{s^k\}$ are bounded from below, i.e.,

$$x_i^k \geq \gamma \quad \text{and} \quad s_i^k \geq \gamma \quad \forall i = 1, \ldots, n,\ \ \forall k \in \mathbb{N}, \tag{10}$$

where $\gamma \in \mathbb{R}$ denotes a suitable (possibly negative) constant.

On the other hand, the sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ is unbounded by assumption. This implies that there is at least one component $i \in \{1, \ldots, n\}$ such that $x_i^k \to +\infty$ or $s_i^k \to +\infty$ on a subsequence since otherwise the two sequences $\{x^k\}$ and $\{s^k\}$ would be bounded which, in turn, would imply the boundedness of the sequence $\{\lambda^k\}$ as well because we have $A^T\lambda^k + s^k = c$ for all $k \in \mathbb{N}$ and because A is assumed to have full rank.

Now let $\hat{w} = (\hat{x}, \hat{\lambda}, \hat{s}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ be a strictly feasible point for (3) whose existence is guaranteed by our assumptions. Then, in particular, we have

$$A^T\hat{\lambda} + \hat{s} = c \quad \text{and} \quad A\hat{x} = b.$$

Since we also have

$$A^T\lambda^k + s^k = c \quad \text{and} \quad Ax^k = b$$

for all $k \in \mathbb{N}$, we get

$$A^T(\hat{\lambda} - \lambda^k) + (\hat{s} - s^k) = 0 \quad \text{and} \quad A(\hat{x} - x^k) = 0 \tag{11}$$

by subtracting these equations. Premultiplying the first equation in (11) with $(\hat{x} - x^k)^T$ and taking into account the second equation in (11) gives

$$(\hat{x} - x^k)^T (\hat{s} - s^k) = 0.$$

Reordering this equation, we obtain

$$\hat{s}^T x^k + \hat{x}^T s^k = (x^k)^T s^k + \hat{x}^T \hat{s} \tag{12}$$

for all $k \in \mathbb{N}$. Using (10) as well as $\hat{x} > 0$ and $\hat{s} > 0$ in view of the strict feasibility of the vector $\hat{w} = (\hat{x}, \hat{\lambda}, \hat{s})$, it follows from (12) and the fact that $x_i^k \to +\infty$ or $s_i^k \to +\infty$ on a subsequence for at least one index $i \in \{1, \dots, n\}$ that

$$(x^k)^T s^k \to +\infty.$$

Hence there exists a component $j \in \{1, \dots, n\}$ (independent of k) such that

$$x_j^k s_j^k \to +\infty$$

on a suitable subsequence. Using (10), this implies

$$\max\{0, x_j^k\} \max\{0, s_j^k\} \to +\infty$$

and therefore

$$\varphi(x_j^k, s_j^k) \to +\infty$$

on this subsequence. This, in turn, gives $\|\Phi(w^k)\| \to +\infty$, a contradiction to $w^k \in \mathcal{L}_c$ for all $k \in \mathbb{N}$. □

We next state a simple consequence of Proposition 2.7.

Corollary 2.8 *Let φ be the penalized minimum or penalized Fischer-Burmeister function, and consider the set*

$$\mathcal{L}_{c,\tau} := \{w = (x, \lambda, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \mid A^T\lambda + s = c,\ Ax = b,\ \|\Phi_\tau(w)\| \leq c\},$$

where $c \in \mathbb{R}$ and $\tau \geq 0$ are any given constants. Assume that there exists a strictly feasible point for the optimality conditions (3). Then the set $\mathcal{L}_{c,\tau}$ is bounded.

Proof. Let $w \in \mathcal{L}_{c,\tau}$ be given. Using Lemma 2.5 and the triangle inequality, it follows that

$$\|\Phi(w)\| \leq \|\Phi(w) - \Phi_\tau(w)\| + \|\Phi_\tau(w)\| \leq \kappa\tau + c.$$

Hence we have $w \in \mathcal{L}_{\bar{c}}$, where $\bar{c} := \kappa\tau + c$ and $\mathcal{L}_{\bar{c}}$ denotes the set defined in Proposition 2.7. This shows that the inclusion

$$\mathcal{L}_{c,\tau} \subseteq \mathcal{L}_{\bar{c}}$$

holds. Since the set on the right-hand side is compact by Proposition 2.7, the statement follows. □

The proof of the previous result actually shows that the following uniform boundedness result holds.

Corollary 2.9 *Let φ denote the penalized minimum or penalized Fischer-Burmeister function, and let $\mathcal{L}_{c,\tau}$ be the set defined in Corollary 2.8. Then the set*

$$\bigcup_{\tau \in [0,\bar{\tau}]} \mathcal{L}_{c,\tau}$$

is bounded for any $c \in \mathbb{R}$ and any $\bar{\tau} \geq 0$.

Proposition 2.7 and its corollaries will guarantee the existence of accumulation points when using one of the penalized functions.

In the remaining part of this section, we now investigate the question whether the smooth equation

$$\Phi_\tau(w) = 0 \tag{13}$$

has a solution for each $\tau > 0$. The answer is relatively simple if φ denotes either the minimum or the Fischer-Burmeister function; for these two functions, it was noted in [20] that (13) is satisfied if and only if the following conditions hold:

$$\begin{array}{rcl} A^T\lambda + s & = & c, \\ Ax & = & b, \\ x_i > 0, \;\; s_i > 0, \;\; x_i s_i & = & \tau^2 \quad \forall i = 1, \ldots, n. \end{array} \tag{14}$$

These conditions, however, are precisely the *central path conditions* used by basically all interior-point methods, and it is known that they have a unique solution for every $\tau > 0$ if there exists a strictly feasible vector $\hat{w} = (\hat{x}, \hat{\lambda}, \hat{s})$ for the optimality conditions (3), see, e.g., [32].

Hence we turn our attention to the two penalized functions. The next result states that the system (13) has a solution also in this case.

Proposition 2.10 *Let φ be the penalized minimum or penalized Fischer-Burmeister function. Assume that there is a strictly feasible vector for the optimality conditions (3). Then the system (13) has a solution for every $\tau > 0$.*

Proof. Let $\tau > 0$ be fixed, and consider the optimization problem

$$\min \Psi_\tau(w) \quad \text{s.t.} \quad A^T\lambda + s = c, \;\; Ax = b, \tag{15}$$

where $\Psi_\tau : \mathbb{R}^n \to \mathbb{R}$ denotes the function

$$\Psi_\tau(w) := \frac{1}{2}\|\Phi_\tau(w)\|^2.$$

Since A has full rank, the set $\mathcal{L}_{c,\tau}$ defined in Corollary 2.8 is nonempty (for sufficiently large $c > 0$) and compact. Hence the above optimization problem has a global minimum $w_\tau = (x_\tau, \lambda_\tau, s_\tau)$. We will show that w_τ is a solution of the nonlinear system (13).

To this end, note that (15) is a linearly constrained optimization problem. Hence there exist Lagrange multipliers $\eta \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^n$ such that (w_τ, η, μ) satisfies the KKT conditions of (15):

$$\nabla \Psi_\tau(w_\tau) + \begin{pmatrix} A^T\eta \\ A\mu \\ \mu \end{pmatrix} = 0.$$

Since $\nabla \Psi_\tau(w_\tau) = \Phi'_\tau(w_\tau)^T \Phi_\tau(w_\tau)$, we can use the expression (8) for the Jacobian in order to rewrite the above equation as

$$\begin{pmatrix} 0 & A^T & D_{a,\tau} \\ A & 0 & 0 \\ I & 0 & D_{b,\tau} \end{pmatrix} \begin{pmatrix} A^T\lambda_\tau + s_\tau - c \\ Ax_\tau - b \\ \phi_\tau(x_\tau, s_\tau) \end{pmatrix} + \begin{pmatrix} A^T\eta \\ A\mu \\ \mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

where the diagonal matrices $D_{a,\tau}$ and $D_{b,\tau}$ are defined in a way similar to (9). In addition, w_τ is feasible for (15). Consequently, the previous conditions can be simplified to

$$D_{a,\tau}\phi_\tau(x_\tau, s_\tau) = -A^T\eta, \;\; A\mu = 0, \;\; D_{b,\tau}\phi_\tau(x_\tau, s_\tau) + \mu = 0.$$

Multiplying the third equation with A, taking into account the second equation, and replacing $\phi_\tau(x_\tau, s_\tau)$ from the first equation gives

$$AD_{a,\tau}^{-1}D_{b,\tau}A^T\eta = 0.$$

Since A has full rank and $D_{a,\tau}, D_{b,\tau} \in \mathbb{R}^{n\times n}$ are positive definite diagonal matrices, it follows that $\eta = 0$. This, in turn, implies $\phi_\tau(x_\tau, s_\tau) = 0$. Using once again the fact that $w_\tau = (x_\tau, \lambda_\tau, s_\tau)$ satisfies the linear constraints in (15), we see that the vector w_τ is indeed a solution of the nonlinear system of equations (13). □

While Proposition 2.10 guarantees the existence of a solution of the system (13) also for the two penalized functions, it is currently not clear to the authors whether this solution is always unique. In view of Proposition 2.6, any such solution must be an isolated solution, however, this does not necessarily exclude multiple solutions. In any case, the existence of such a solution is used here only for motivational purposes since our smoothing-type method tries to follow the corresponding *smoothing path*

$$\tau \mapsto w_\tau,$$

where w_τ denotes a (unique) solution of (13). However, the convergence theory for our method still works even if this system may not have a solution (since, in any case, we follow this smoothing path only inexactly, i.e., we require only inexact solutions of the systems (13)).

3 Predictor-Corrector Smoothing Method

Let φ be any of the NCP-functions introduced in the previous section. The basic idea of our method is to solve the optimality conditions (3) by solving the equivalent nonlinear system of equations $\Phi(w) = 0$, where Φ is defined via (6). However, since Φ is nonsmooth and might also have singular Jacobian matrices, we replace the system $\Phi(w) = 0$ by the smooth equation $\Phi_\tau(w) = 0$ to which we apply Newton's method. Of course, we have to take care of the way we update the smoothing parameter τ. Using the notation

$$\Psi_\tau(w) := \frac{1}{2}\Phi_\tau(w)^T\Phi_\tau(w) = \frac{1}{2}\|\Phi_\tau(w)\|^2,$$

we first give a formal statement of our smoothing-type method. The method can be viewed as a simplification of a recently proposed continuation method by Chen and Chen [6] (in [6], the authors use some additional perturbations of the mapping Φ in order to improve their global convergence properties).

Algorithm 3.1 *(Predictor-Corrector Smoothing Method)*

(S.0) (Initialization)
Choose $w^0 := (x^0, \lambda^0, s^0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n, \tau_0 > 0, \beta \geq \|\Phi_{\tau_0}(w^0)\|/\tau_0, \rho, \sigma \in (0,1), \varepsilon \geq 0$*, and set* $k := 0$.

(S.1) (Termination Criterion)
If $\|\Phi(w^k)\| \leq \varepsilon$*: STOP.*

(S.2) (Predictor Step)
Compute a solution $\Delta w^k = (\Delta x^k, \Delta\lambda^k, \Delta s^k) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ *of the linear system*

$$\Phi'_{\tau_k}(w^k)\Delta w = -\Phi(w^k). \tag{16}$$

If $\|\Phi(w^k + \Delta w^k)\| = 0$*: STOP. Otherwise, if*

$$\|\Phi_{\tau_k}(w^k + \Delta w^k)\| > \beta\tau_k,$$

then set

$$\hat{w}^k := w^k, \ \ \hat{\tau}_k := \tau_k,$$

else compute $t_k = \rho^{\ell_k}$*, where* ℓ_k *is the nonnegative integer such that*

$$\begin{aligned} \|\Phi_{\rho^j\tau_k}(w^k + \Delta w^k)\| &\leq \beta\rho^j\tau_k \quad \forall j = 0,1,2,\ldots,\ell_k \ \textit{and} \\ \|\Phi_{\rho^{\ell_k+1}\tau_k}(w^k + \Delta w^k)\| &> \beta\rho^{\ell_k+1}\tau_k, \end{aligned}$$

and set $\hat{\tau}_k := t_k\tau_k$ *and*

$$\hat{w}^k := \begin{cases} w^k & \textit{if } \ell_k = 0, \\ w^k + \Delta w^k & \textit{otherwise.} \end{cases}$$

(S.3) (Corrector Step)
Compute a solution $\Delta\hat{w}^k = (\Delta\hat{x}^k, \Delta\hat{\lambda}^k, \Delta\hat{s}^k) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ *of the linear system*

$$\Phi'_{\hat{\tau}_k}(\hat{w}^k)\Delta\hat{w} = -\Phi_{\hat{\tau}_k}(\hat{w}^k). \tag{17}$$

Let $\hat{t}_k = \max\{\rho^\ell \mid \ell = 0, 1, 2, \dots\}$ *such that*

$$\Psi_{\hat{\tau}_k}(\hat{w}^k + \hat{t}_k \Delta \hat{w}^k) \le \Psi_{\hat{\tau}_k}(\hat{w}^k) + \hat{t}_k \sigma \nabla \Psi_{\hat{\tau}_k}(\hat{w}^k)^T \Delta \hat{w}^k, \tag{18}$$

and set

$$w^{k+1} := \hat{w}^k + \hat{t}_k \Delta \hat{w}^k.$$

Compute $\gamma_k = \max\{\rho^\ell \mid \ell = 0, 1, 2, \dots\}$ *such that*

$$\|\Phi_{(1-\gamma_k)\hat{\tau}_k}(w^{k+1})\| \le \beta(1-\gamma_k)\hat{\tau}_k, \tag{19}$$

and set $\tau_{k+1} := (1-\gamma_k)\hat{\tau}_k$.

(S.4) (Update)
Set $k \leftarrow k+1$*, and go to Step (S.1).*

Algorithm 3.1 solves two linear systems of equations at each iteration, one in the predictor step and one in the corrector step. However, the coefficient matrices of these two systems are identical in case the predictor step is not successful; in this case, we therefore have to calculate only one matrix factorization per iteration.

The global convergence properties of Algorithm 3.1 will be discussed in Section 4, whereas the local analysis is the subject of Section 5. Nevertheless, we would like to emphasize already at this point that the global properties mainly follow from the corrector step, whereas the local theory is entirely based on the predictor step. To this end, it is also interesting to note a major difference between the two linear systems (16) and (17): While the linear system (17) arising in the corrector step is just Newton's system applied to the nonlinear equation $\Phi_{\hat{\tau}_k}(w) = 0$, the right-hand side of the linear system (16) in the predictor step is unperturbed, i.e., does not depend on the current smoothing parameter. Hence the linear system (16) cannot be interpreted as a pure Newton equation. The resulting direction is sometimes called a Jacobian smoothing step, see, e.g., [12, 23, 15]. Such a Jacobian smoothing step may be viewed as the counterpart of the affine scaling step from the interior-point literature (see, e.g., [32]) for smoothing-type methods.

We next give some further explanations about the way Algorithm 3.1 works. The basic philosophy behind Algorithm 3.1 is quite simple: It tries to generate a sequence $\{w^k\}$ of iterates and a sequence $\{\tau_k\}$ of smoothing parameters satisfying the relation

$$\|\Phi_{\tau_k}(w^k)\| \le \beta \tau_k \tag{20}$$

for all $k \in \mathbb{N}$, see Proposition 3.4 below. In this way, we guarantee that each w^k belongs to a certain neighbourhood of the smoothing path. Moreover, we control the size of the neighbourhood by our parameter τ_k. A neighbourhood of the form (20) for the smoothing path was first used by Chen and Xiu [10] (for the minimum function), whereas other authors sometimes prefer to use slightly different neighbourhoods for the smoothing path, see, e.g., [3, 2, 5].

Having this philosophy in mind, the predictor step in Algorithm 3.1 does the following: After solving the Jacobian smoothing equation (16), it checks whether the full step $w^k + \Delta w^k$ still belongs to the neighbourhood (20) in the sense that

$$\|\Phi_{\tau_k}(w^k + \Delta w^k)\| \le \beta \tau_k. \tag{21}$$

If this is true, we set $\hat{w}^k := w^k + \Delta w^k$ and reduce the smoothing parameter τ_k as much as possible with the restriction that a condition like (20) holds at the intermediate iterate $\hat{w}^k$ and the new smoothing parameter $\hat{\tau}_k$. In Proposition 3.2 below we will see that the calculation of $\hat{\tau}_k$ is a finite procedure. On the other hand, if (21) does not hold, we basically skip the predictor step and set $\hat{w}^k := w^k, \hat{\tau}_k := \tau_k$. However, there is one exception which will become important in our global convergence analysis: Even if (21) is satisfied, we do not set $\hat{w}^k = w^k + \Delta w^k$ if $\ell_k = 0$ (so that $\hat{\tau}_k = \tau_k$).

The corrector step then computes a new Newton direction $\Delta\hat{w}^k$ in (17), followed by a line search in order to reduce the value of $\Psi_\tau(\cdot)$, so that $w^{k+1} = \hat{w}^k + \hat{t}_k \Delta\hat{w}^k$ becomes the new iterate, where, of course, $\hat{t}_k > 0$ denotes the steplength in the corrector step. In addition, we try to reduce the smoothing parameter in the corrector step in such a way that a condition like (20) still holds at iteration $k+1$.

In the remaining part of this section, we state some elementary properties of Algorithm 3.1 and show, in particular, that the method is well-defined. To this end, we first recall that the two linear systems (16) and (17) have a unique solution due to Proposition 2.6. We next show that the computation of $t_k > 0$ in the predictor step is a finite procedure.

Proposition 3.2 *The stepsize $t_k > 0$ in Step (S.2) of Algorithm 3.1 is uniquely defined.*

Proof. The proof is quite standard, see, e.g., [5], and is included here only for the sake of completeness: When we compute the stepsize $t_k > 0$ in Step (S.2) of Algorithm 3.1, we are in the situation that $\|\Phi(w^k + \Delta w^k)\| > 0$ and $\|\Phi_{\tau_k}(w^k + \Delta w^k)\| \leq \beta\tau_k$. Therefore, since Φ is continuous in both w and τ, there is a unique exponent ℓ_k such that $t_k = \rho^{\ell_k}$ has the properties given in Step (S.2). $\square$

On the other hand, the computation of the stepsize $\hat{t}_k$ in Step (S.3) of Algorithm 3.1 corresponds to a standard globalization of Newton's method for nonlinear systems of equations by Armijo's rule, and is known to be well-defined. Hence we can state the following result.

Proposition 3.3 *The stepsize $\hat{t}_k > 0$ in Step (S.3) of Algorithm 3.1 is uniquely defined.*

For later reference, we include a short discussion on the way we compute the stepsize $\hat{t}_k > 0$ in Step (S.3). To this end, we first note that the gradient $\nabla\Psi_{\hat{\tau}_k}(\hat{w}^k)$ is given by

$$\nabla\Psi_{\hat{\tau}_k}(\hat{w}^k) = \Phi'_{\hat{\tau}_k}(\hat{w}^k)^T \Phi_{\hat{\tau}_k}(\hat{w}^k).$$

Furthermore, we obtain from Proposition 2.6 and the linear system (17) that

$$\Delta\hat{w}^k = -\Phi'_{\hat{\tau}_k}(\hat{w}^k)^{-1} \Phi_{\hat{\tau}_k}(\hat{w}^k).$$

Hence Armijo's rule (18) can be rewritten in the form

$$\Psi_{\hat{\tau}_k}(\hat{w}^k + \hat{t}_k \Delta\hat{w}^k) \leq \Psi_{\hat{\tau}_k}(\hat{w}^k) - 2\hat{t}_k\sigma\Psi_{\hat{\tau}_k}(\hat{w}^k) = (1 - 2\hat{t}_k\sigma)\Psi_{\hat{\tau}_k}(\hat{w}^k). \tag{22}$$

In particular, it follows that

$$\|\Phi_{\hat{\tau}_k}(w^{k+1})\| < \|\Phi_{\hat{\tau}_k}(\hat{w}^k)\| \tag{23}$$

holds for all $k \in \mathbb{N}$.

The next result shows that all iterates w^k generated by Algorithm 3.1 belong to the neighbourhood

$$\mathcal{N}(\beta) := \{w \mid \|\Phi_\tau(w)\| \leq \beta\tau \text{ for some } \tau > 0\}$$

of the smoothing path introduced in the previous section.

Proposition 3.4 *The iterates w^k generated by Algorithm 3.1 satisfy $\|\Phi_{\tau_k}(w^k)\| \leq \beta\tau_k$ for all $k \in \mathbb{N}$.*

Proof. The proof is by induction on k. For $k = 0$, the choice of β in Step (S.0) guarantees that $\|\Phi_{\tau_0}(w^0)\| \leq \beta\tau_0$. So assume that $\|\Phi_{\tau_k}(w^k)\| \leq \beta\tau_k$ for some $k \geq 0$. Then the updating rules in Step (S.2) imply that $\|\Phi_{\hat{\tau}_k}(\hat{w}^k)\| \leq \beta\hat{\tau}_k$ (both if $\ell_k = 0$ and if $\ell_k > 0$). Consequently, we obtain from (23) that

$$\|\Phi_{\hat{\tau}_k}(w^{k+1})\| < \|\Phi_{\hat{\tau}_k}(\hat{w}^k)\| \leq \beta\hat{\tau}_k.$$

By continuity, the procedure to calculate $\gamma_k > 0$ in Step (S.3) is therefore finite and gives a new smoothing parameter τ_{k+1} satisfying $\|\Phi_{\tau_{k+1}}(w^{k+1})\| \leq \beta\tau_{k+1}$. □

Summarizing the above results, we can therefore say that Algorithm 3.1 is at least well-defined. — We close this section by stating another simple property of Algorithm 3.1.

Lemma 3.5 *Assume that the starting point $w^0 = (x^0, \lambda^0, s^0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ satisfies the linear equations $A^T\lambda^0 + s^0 = c$ and $Ax^0 = b$. Then all iterates $w^k = (x^k, \lambda^k, s^k)$ generated by Algorithm 3.1 satisfy these equations as well, i.e., we have $A^T\lambda^k + s^k = c$ and $Ax^k = b$ for all $k \in \mathbb{N}$.*

Proof. This is obvious. □

Note that it is quite easy to construct points $w^0 = (x^0, \lambda^0, s^0)$ satisfying the assumptions from Lemma 3.5 since our starting points are not assumed to have positive components x^0 and s^0 (in contrast to many interior-point methods, for example).

4 Global Convergence

Unless stated otherwise, we assume throughout this section that φ denotes any of the four NCP-functions introduced in Section 2. The aim of this section is to give a global convergence result for Algorithm 3.1. To this end, we first recall from Proposition 3.4 that any sequence $\{w^k\}$ generated by Algorithm 3.1 has the property that

$$\|\Phi_{\tau_k}(w^k)\| \leq \beta\tau_k$$

for all $k \in \mathbb{N}$. Therefore, if τ_k converges to zero, we obtain

$$\|\Phi_{\tau_k}(w^k)\| \to 0,$$

and this, in turn, shows that any accumulation point of the sequence $\{w^k\}$ is a solution of the optimality conditions (3), see Theorem 4.2 below. The basic question is therefore as follows: Under which assumptions does the sequence $\{\tau_k\}$ converge to zero?

Many related papers on smoothing-type methods (mainly in the context of complementarity problems) deal with the same question and prove $\{\tau_k\} \downarrow 0$ under somewhat restrictive assumptions which typically imply that the solution of the underlying problem is unique, see, e.g., [2, 3, 4, 5, 6, 10, 34] (the only three exceptions we are currently aware of are the papers [6, 18, 30]). Since the uniqueness of the solution is usually viewed as a too strong assumption in the context of linear programs, we want to state a global convergence result which also holds for solution sets that do not necessarily consist of a single vector. Our main step in this direction is contained in the following proposition.

Proposition 4.1 *Assume that the sequence $\{w^k\}$ generated by Algorithm 3.1 has at least one accumulation point. Then the sequence of smoothing parameters $\{\tau_k\}$ converges to* 0.

Proof. By construction, the sequence $\{\tau_k\}$ of smoothing parameters is monotonically decreasing. Since it is also bounded from below by zero, it follows that $\{\tau_k\}$ converges to some number $\bar{\tau} \geq 0$. If $\bar{\tau} = 0$, we are done.

Assume that $\bar{\tau} > 0$. Then the updating rules in Step (S.2) of Algorithm 3.1 imply that

$$\hat{\tau}_k = \tau_k \quad \text{and} \quad \hat{w}^k = w^k \tag{24}$$

for all $k \in \mathbb{N}$ sufficiently large. Furthermore, Step (S.3) of Algorithm 3.1 shows that we necessarily have

$$\gamma_k \downarrow 0. \tag{25}$$

Now let w^* be an accumulation point of the sequence $\{w^k\}$ whose existence is guaranteed in view of our assumption. Let $\{w^k\}_K$ be a corresponding subsequence converging to w^*. Since $\Phi'_{\bar{\tau}}(w^*)$ is nonsingular by Proposition 2.6, it follows from standard perturbation results (see, e.g., [14]) that there is a constant $\hat{c} > 0$ with

$$\|\Phi'_{\tau_k}(w^k)^{-1}\| = \|\Phi'_{\hat{\tau}_k}(\hat{w}^k)^{-1}\| \leq \hat{c} \tag{26}$$

for all $k \in K$ sufficiently large. We now consider the two cases $\liminf_{k\in K} \hat{t}_k > 0$ and $\liminf_{k\in K} \hat{t}_k = 0$ separately and derive a contradiction in each of these cases.

Case 1: $\liminf_{k\in K} \hat{t}_k > 0$.
Then there exists a positive constant $\bar{t} > 0$ such that $\hat{t}_k \geq \bar{t}$ for all $k \in K$. In view of (22), (24), and (18), this implies

$$\Psi_{\tau_k}(w^{k+1}) \leq (1 - 2\hat{t}_k\sigma)\Psi_{\tau_k}(w^k) \leq (1 - 2\bar{t}\sigma)\Psi_{\tau_k}(w^k).$$

Proposition 3.4 therefore gives

$$\begin{aligned} \|\Phi_{\tau_k}(w^{k+1})\| &\leq \sqrt{1 - 2\bar{t}\sigma}\|\Phi_{\tau_k}(w^k)\| \\ &\leq (1 - \bar{c})\|\Phi_{\tau_k}(w^k)\| \\ &\leq \beta(1 - \bar{c})\tau_k \end{aligned} \tag{27}$$

for some constant $\bar{c} \in (0,1)$. Since one can easily show that the inequality

$$\|\Phi_\tau(w) - \Phi_{\tau'}(w)\| \le \kappa|\tau - \tau'|$$

holds for all $\tau, \tau' > 0$ and all w, where $\kappa > 0$ denotes the constant from Lemma 2.5 (cf. [21]), we obtain from (27)

$$\begin{aligned}
\|\Phi_{(1-\gamma)\tau_k}(w^{k+1})\| &\le \|\Phi_{\tau_k}(w^{k+1})\| + \|\Phi_{(1-\gamma)\tau_k}(w^{k+1}) - \Phi_{\tau_k}(w^{k+1})\| \\
&\le \beta(1-\bar{c})\tau_k + \kappa\gamma\tau_k \\
&= (\beta - \bar{c}\beta + \kappa\gamma)\tau_k
\end{aligned}$$

for all $\gamma \in (0,1)$. Since an elementary calculation shows that the inequality

$$(\beta - \bar{c}\beta + \kappa\gamma)\tau_k \le \beta(1-\gamma)\tau_k$$

holds for all $\gamma > 0$ satisfying

$$\gamma \le \bar{c}\beta/(\kappa + \beta),$$

the computation of γ_k in Step (S.3) of Algorithm 3.1 yields

$$\gamma_k \ge \rho\bar{c}\beta/(\kappa+\beta) =: \bar{\gamma}$$

for all $k \in K$ sufficiently large, i.e., these γ_k are uniformly bounded away from zero by the positive constant $\bar{\gamma}$. This, however, is a contradiction to (25).

Case 2: $\liminf_{k\in K} \hat{t}_k = 0$.
Subsequencing if necessary, we then have $\lim_{k\in K} \hat{t}_k = 0$. Consequently, the Armijo condition (18) is not satisfied for $\alpha_k := \hat{t}_k/\rho$ and all $k \in K$ sufficiently large, i.e., we have

$$\Psi_{\hat{\tau}_k}(\hat{w}^k + \alpha_k\Delta\hat{w}^k) > \Psi_{\hat{\tau}_k}(\hat{w}^k) + \alpha_k\sigma\nabla\Psi_{\hat{\tau}_k}(\hat{w}^k)^T\Delta\hat{w}^k$$

or, equivalently,

$$\frac{\Psi_{\hat{\tau}_k}(\hat{w}^k + \alpha_k\Delta\hat{w}^k) - \Psi_{\hat{\tau}_k}(\hat{w}^k)}{\alpha_k} > \sigma\nabla\Psi_{\hat{\tau}_k}(\hat{w}^k)^T\Delta\hat{w}^k.$$

Using the Mean Value Theorem and (24), we obtain

$$\nabla\Psi_{\tau_k}(\xi^k)^T\Delta\hat{w}^k > \sigma\nabla\Psi_{\tau_k}(w^k)^T\Delta\hat{w}^k \tag{28}$$

for some vector ξ^k belonging to the line segment joining w^k and $w^k + \alpha_k\Delta\hat{w}^k$. From (26), (17), $\{\hat{\tau}_k\}_K = \{\tau_k\}_K \to \bar{\tau}$, and $\{\hat{w}^k\}_K = \{w^k\}_K \to w^*$, it follows that the sequence $\{\Delta\hat{w}^k\}_K$ converges to a vector Δw^* which solves the linear system

$$\Phi'_{\bar{\tau}}(w^*)\Delta\hat{w} = -\Phi_{\bar{\tau}}(w^*). \tag{29}$$

Taking the limit $k \to \infty$ on K and using $\{\alpha_k\}_K \to 0$, $\{\tau_k\}_K \to \bar{\tau}$, we then obtain from (28) and the continuity of $\nabla\Psi$ with respect to w and τ:

$$\nabla\Psi_{\bar{\tau}}(w^*)^T\Delta w^* \ge \sigma\nabla\Psi_{\bar{\tau}}(w^*)^T\Delta w^*.$$

Consequently, we have

$$\nabla\Psi_{\bar{\tau}}(w^*)^T \Delta w^* \geq 0.$$

On the other hand, it follows from $\nabla\Psi_{\hat{\tau}_k}(\hat{w}^k)^T \Delta\hat{w}^k = -\|\Phi_{\hat{\tau}_k}(\hat{w}^k)\|^2 \leq 0$ that

$$\nabla\Psi_{\bar{\tau}}(w^*)^T \Delta w^* \leq 0.$$

We therefore get

$$\nabla\Psi_{\bar{\tau}}(w^*)^T \Delta w^* = 0.$$

Since $\nabla\Psi_{\bar{\tau}}(w^*) = \Phi'_{\bar{\tau}}(w^*)^T \Phi_{\bar{\tau}}(w^*)$, we obtain

$$\Phi_{\bar{\tau}}(w^*) = 0 \tag{30}$$

from (29). On the other hand, it follows from (25) that γ_k/ρ does not satisfy the test (19) for all $k \in K$ sufficiently large, i.e., we have

$$\|\Phi_{(1-\gamma_k/\rho)\tau_k}(w^{k+1})\| > \beta\left(1 - \frac{\gamma_k}{\rho}\right)\tau_k.$$

Since $\gamma_k/\rho \to 0, \hat{t}_k \to 0$ and, therefore, $w^{k+1} \to w^*$ for $k \in K$, we obtain

$$\|\Phi_{\bar{\tau}}(w^*)\| \geq \beta\bar{\tau} > 0,$$

by taking the limit $k \to +\infty$ on K. This, however, is a contradiction to (30). □

Note that the assumption used in Proposition 4.1, namely the existence of at least one accumulation point, is weaker than the condition that the entire sequence $\{w^k\}$ remains bounded. Nevertheless, in Theorem 4.3 below we will give a sufficient condition for the boundedness of the whole sequence $\{w^k\}$.

Before, however, we state the main global convergence result for Algorithm 3.1. It is a direct consequence of Proposition 4.1.

Theorem 4.2 *Every accumulation point of a sequence $\{w^k\}$ generated by Algorithm 3.1 is a solution of the optimality conditions (3).*

Proof. In view of Proposition 3.4, we have

$$\|\Phi_{\tau_k}(w^k)\| \leq \beta\tau_k \tag{31}$$

for all $k \in \mathbb{N}$. Now let w^* be an accumulation point of $\{w^k\}$, and let $\{w^k\}_K$ be a subsequence converging to w^*. Then we have $\{\tau_k\} \to 0$ in view of Proposition 4.1. Therefore, taking the limit $k \to \infty$ for $k \in K$ and using the fact that Φ is a continuous function in both w and τ, we obtain from (31) that

$$\|\Phi(w^*)\| = \|\Phi_0(w^*)\| \leq 0.$$

Hence $\Phi(w^*) = 0$. The definition of Φ therefore implies that w^* is a solution of the optimality conditions (3). □

The next result gives a sufficient condition for the existence of accumulation points. Note that this result is stated here only for the penalized minimum and the penalized Fischer-Burmeister function, whereas it is not true, in general, for the minimum or Fischer-Burmeister functions themselves.

Theorem 4.3 *Assume there exists a strictly feasible point for the optimality conditions (3) and that the starting point* $w^0 = (x^0, \lambda^0, s^0)$ *satisfies the linear equations* $Ax^0 = b$ *and* $A^T\lambda^0 + s^0 = c$. *Then the sequence* $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ *generated by Algorithm 3.1 remains bounded.*

Proof. Using Lemma 3.5 and our assumption regarding the starting point, we have $A^T\lambda^k + s^k = c$ and $Ax^k = b$ for all $k \in \mathbb{N}$. Since $\|\Phi(w^k)\| \leq \beta\tau_k$ in view of Proposition 3.4 and $\tau_k \leq \tau_0$ for all $k \in \mathbb{N}$, the statement follows immediately from Corollary 2.9. $\square$

5 Quadratic Convergence

In this section, we investigate the local convergence properties of Algorithm 3.1. To this end, we assume throughout this section that the following assumptions hold.

Assumption 5.1 *(a)* φ *denotes the minimum function.*

(b) The starting point $w^0 = (x^0, \lambda^0, s^0)$ *satisfies* $A^T\lambda^0 + s^0 = c$ *and* $Ax^0 = b$.

(c) The parameter β *satisfies* $\beta \geq \|\Phi_{\tau_0}(w^0)\|/\tau_0$ *and* $\beta > \kappa$, *where* $\kappa > 0$ *denotes the (known) constant from Lemma 2.5.*

(d) The smoothing parameter satisfies $\lim_{k\to\infty} \tau_k = 0$.

(e) The sequence $\{w^k\}$ *converges to a vector* $w^* = (x^*, \lambda^*, s^*)$ *satisfying the strict complementarity condition* $x_i^* + s_i^* > 0$ *for all* $i = 1, \ldots, n$.

Assumption 5.1 (a) just says that φ is the minimum function. This is a restriction compared to our global convergence theory from the previous section, however, we were not able to extend the subsequent analysis to any of the NCP-functions stated in Section 2. Assumption 5.1 (b) requires the starting point $w^0 = (x^0, \lambda^0, s^0)$ to satisfy the linear equations $A^T\lambda^0 + s^0 = c$ and $Ax^0 = b$. We recall that this is a rather weak assumption since we do not assume, in addition, that x^0 and s^0 have positive components. Condition (c) assumes that the parameter β is sufficiently large, where the first condition in Assumption 5.1 (c) is identical with the initialization of the globally convergent Algorithm 3.1; the second condition is only used in order to establish local fast convergence, as will become clear from the analysis in this section. Assumption 5.1 (d) requires that τ_k converges to zero, and this is a reasonable assumption in view of our global convergence theory from Section 4. Finally, in (e) we assume two things, namely that the entire sequence $\{w^k\}$ converges to one point w^* (which, then, must be a solution of the optimality conditions (3) due to Theorem 4.2), and that the limit point w^* satisfies the strict complementarity condition. It is currently not clear to us whether this assumption holds automatically and is therefore superfluous. Note, however, that it is certainly satisfied if the optimality conditions (3) have a unique solution since then this solution satisfies the strict complementarity condition by the well-known Goldman-Tucker Theorem, see, e.g., [32].

We also note, however, that our subsequent analysis would remain true if we would replace Assumption 5.1 (e) by the requirement that

$$\liminf_{k\to\infty}(x_i^k + s_i^k) > 0,$$

so that every accumulation point satisfies the strict complementarity assumption, see also Tseng [31]. Since, in any case, part of our analysis is somewhat technical, we prefer to use the slightly stronger condition (e) throughout this section.

Let us denote the solution set of the optimality conditions (3) by $\mathcal{S}$:

$$\mathcal{S} := \{\bar{w} = (\bar{x}, \bar{\lambda}, \bar{s}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \,|\, \bar{w} \text{ solves (3)}\}.$$

In view of Assumption 5.1 (e) and Theorem 4.2, this set is nonempty. Furthermore, let $\mathcal{S}_P$ and $\mathcal{S}_D$ be the solution sets of the corresponding primal and dual linear programs:

$$\begin{aligned} \mathcal{S}_P &:= \{\bar{x} \,|\, \bar{x} \text{ solves (1)}\}, \\ \mathcal{S}_D &:= \{(\bar{\lambda}, \bar{s}) \,|\, (\bar{\lambda}, \bar{s}) \text{ solves (2)}\}. \end{aligned}$$

Note that

$$\mathcal{S} = \mathcal{S}_P \times \mathcal{S}_D,$$

i.e., $\mathcal{S}_P$ and $\mathcal{S}_D$ are the projections of $\mathcal{S}$ onto the x- and (λ, s)-spaces, respectively. Following standard notation from the linear programming literature, we further introduce the two index sets

$$\begin{aligned} \mathcal{B} &:= \{i \in \{1, \dots, n\} \,|\, \bar{x}_i > 0 \text{ for some } \bar{x} \in \mathcal{S}_P\}, \\ \mathcal{N} &:= \{i \in \{1, \dots, n\} \,|\, \bar{s}_i > 0 \text{ for some } (\bar{\lambda}, \bar{s}) \in \mathcal{S}_D\}. \end{aligned}$$

We first state some simple properties of these two index sets and relate it to the solution $w^* = (x^*, \lambda^*, s^*)$ from Assumption 5.1 (e).

Lemma 5.2 *The following statements hold:*

(a) We have $\bar{x}_{\mathcal{N}} = 0$ for all $\bar{x} \in \mathcal{S}_P$ and $\bar{s}_{\mathcal{B}} = 0$ for all $(\bar{\lambda}, \bar{s}) \in \mathcal{S}_D$.

(b) We have $\mathcal{B} = \{i \,|\, x_i^* > 0\}$ and $\mathcal{N} = \{i \,|\, s_i^* > 0\}$, where $w^* = (x^*, \lambda^*, s^*)$ denotes the vector from Assumption 5.1 (e).

Proof. (a) Let $\bar{x} \in \mathcal{S}_P$ and $i \in \mathcal{N}$ be arbitrarily given. Then there exists a solution $(\bar{\lambda}, \bar{s}) \in \mathcal{S}_D$ such that $\bar{s}_i > 0$. Since $(\bar{x}, \bar{\lambda}, \bar{s}) \in \mathcal{S}_P \times \mathcal{S}_D = \mathcal{S}$, it follows from the complementarity condition that $\bar{x}_i = 0$. This shows that $\bar{x}_{\mathcal{N}} = 0$ for all $\bar{x} \in \mathcal{S}_P$. In a similar way, one can verify the statement that $\bar{s}_{\mathcal{B}} = 0$ for all $(\bar{\lambda}, \bar{s}) \in \mathcal{S}_D$.

(b) Let $w^* = (x^*, \lambda^*, s^*)$ be the strictly complementary solution of (3) from Assumption 5.1 (e). Define the corresponding index sets

$$\mathcal{B}^* := \{i \,|\, x_i^* > 0\} \quad \text{and} \quad \mathcal{N}^* := \{i \,|\, s_i^* > 0\}.$$

Since x^* and (λ^*, s^*) are particular solutions of the linear programs (1) and (2), respectively, the definitions of the index sets $\mathcal{B}$ and $\mathcal{N}$ immediately give

$$\mathcal{B}^* \subseteq \mathcal{B} \quad \text{and} \quad \mathcal{N}^* \subseteq \mathcal{N}. \tag{32}$$

However, since $\mathcal{B}^* \cup \mathcal{N}^* = \{1, \dots, n\}$ by the strict complementarity assumption, this set has the same cardinality as the set $\mathcal{B} \cup \mathcal{N}$ (by the Goldman-Tucker Theorem). Hence equalities must hold in (32). $\square$

Note that Lemma 5.2 (b) holds for any strictly complementary solution $w^* = (x^*, \lambda^*, s^*)$ of the optimality conditions (3). — The following result states that the function $\|\Phi(\cdot)\|$ provides a local error bound for the distance of a vector w to the solution set $\mathcal{S}$ of the optimality conditions (3).

Lemma 5.3 *There is a constant $\gamma > 0$ with*

$$dist_{\mathcal{S}}(w^k) \leq \gamma \|\Phi(w^k)\|$$

for all $k \in \mathbb{N}$ sufficiently large.

Proof. We first note that Φ^{-1} is a polyhedral multifunction (as defined, e.g., in [27]). Therefore, it follows from Robinson [27, Proposition 1] that Φ^{-1} is locally upper Lipschitzian at the origin, i.e., there exist scalars $\varepsilon > 0$ and $\gamma > 0$ such that

$$\Phi^{-1}(z) \subseteq \Phi^{-1}(0) + \gamma \|z\| \mathcal{B}_1(0)$$

for all z with $\|z\| < \varepsilon$, where $\mathcal{B}_1(0)$ denotes the unit ball in $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$. Using $z = \Phi(w^k)$ and taking into account that $w^k \to w^*$ for some vector $w^* \in \mathcal{S}$ by Assumption 5.1 (e), it follows that

$$w^k \in \Phi^{-1}(0) + \gamma \|\Phi(w^k)\| \mathcal{B}_1(0)$$

for all $k \in \mathbb{N}$ sufficiently large. Since $\Phi^{-1}(0)$ is equal to the solution set $\mathcal{S}$ of the optimality conditions (3), this implies

$$\text{dist}_{\mathcal{S}}(w^k) \leq \gamma \|\Phi(w^k)\|$$

for all $k \in \mathbb{N}$ large enough. $\square$

We next show that the distance of the current iterate w^k to the solution set $\mathcal{S}$ is bounded from above by a constant times the current smoothing parameter τ_k.

Lemma 5.4 *We have*

$$dist_{\mathcal{S}}(w^k) = O(\tau_k)$$

for all $k \in \mathbb{N}$ large enough

Proof. Since the primal-dual solution set $\mathcal{S}$ is closed and nonempty, there exists, for each $k \in \mathbb{N}$, a vector $w^{*,k} \in \mathcal{S}$ such that

$$\|w^k - w^{*,k}\| = \text{dist}_{\mathcal{S}}(w^k).$$

In view of Lemma 5.3, there is a constant $\gamma > 0$ with

$$\|w^k - w^{*,k}\| \leq \gamma \|\Phi(w^k)\|$$

for all $k \in \mathbb{N}$ sufficiently large. Using Proposition 3.4 and Lemma 2.5, this gives

$$\begin{aligned}
\|w^k - w^{*,k}\| &\leq \gamma \|\Phi(w^k)\| \\
&\leq \gamma \left(\|\Phi_{\tau_k}(w^k)\| + \|\Phi(w^k) - \Phi_{\tau_k}(w^k)\| \right) \\
&\leq \gamma \left(\beta \tau_k + \kappa \tau_k \right) \\
&= \gamma(\beta + \kappa)\tau_k,
\end{aligned}$$

where $\kappa > 0$ denotes the constant from Lemma 2.5. This implies our statement. □

We next want to estimate the growth behaviour of the vector Δw^k, where, of course, Δw^k denotes the solution of the linear system (16) in the predictor step. Typically, one obtains simple estimates if the inverse of the Jacobian matrices $\Phi_{\tau_k}(w^k)^{-1}$ would remain bounded, and this is indeed the standard way in which most papers on smoothing-type methods prove local fast convergence results for their methods. However, according to [15] (see also [5]), the above Jacobian matrices converge to a singular matrix unless the optimality conditions (3) have a unique solution. Since we want to avoid the uniqueness assumption here, we cannot use standard perturbation results. Instead, we follow a recent paper by Tseng [31] and show by a somewhat technical analysis that the term $\|\Delta w^k\|$ can be estimated by a constant times the current smoothing parameter. This result will play a crucial role in order to establish local quadratic convergence of our method at the end of this section.

Lemma 5.5 *We have*

$$\|\Delta w^k\| = O(\tau_k)$$

for all $k \in \mathbb{N}$ sufficiently large.

Proof. Taking into account the structure of the Jacobian $\Phi'_{\tau_k}(w^k)$ (cf. (8)) and the definition of Φ, we get from the linear system (16) in the predictor step that

$$\begin{aligned}
A^T \Delta\lambda^k + \Delta s^k &= 0, & (33)\\
A \Delta x^k &= 0, & (34)\\
D^k_{a,\tau} \Delta x^k + D^k_{b,\tau} \Delta s^k &= -\phi(x^k, s^k), & (35)
\end{aligned}$$

where, similar to (9),

$$\begin{aligned}
D^k_{a,\tau} &:= \operatorname{diag}\left(\dots, \frac{\partial \varphi_{\tau_k}}{\partial a}(x_i^k, s_i^k), \dots \right) \in \mathbb{R}^{n \times n}, \\
D^k_{b,\tau} &:= \operatorname{diag}\left(\dots, \frac{\partial \varphi_{\tau_k}}{\partial b}(x_i^k, s_i^k), \dots \right) \in \mathbb{R}^{n \times n}, \\
\phi(x^k, s^k) &:= \left(\dots, \varphi(x_i^k, s_i^k), \dots \right)^T \in \mathbb{R}^n,
\end{aligned}$$

and the right-hand sides of (33) and (34) are zero because of Assumption 5.1 (b) and Lemma 3.5. For each $k \in \mathbb{N}$, let $w^{*,k}$ be a solution of the optimality conditions (3) such that

$$\mathrm{dist}_{\mathcal{S}}(w^k) = \|w^k - w^{*,k}\|. \tag{36}$$

Since $w^{*,k} = (x^{*,k}, \lambda^{*,k}, s^{*,k})$ satisfies (3), we have

$$\begin{aligned} A^T\lambda^{*,k} + s^{*,k} - c &= 0, && (37)\\ Ax^{*,k} - b &= 0, && (38)\\ \phi(x^{*,k}, s^{*,k}) &= 0. && (39) \end{aligned}$$

Furthermore, Lemma 3.5 implies that $w^k = (x^k, \lambda^k, s^k)$ satisfies the linear equations

$$\begin{aligned} A^T\lambda^k + s^k - c &= 0, && (40)\\ Ax^k - b &= 0. && (41) \end{aligned}$$

Adding (33), (37), (40) on the one hand, and (34), (38), (41) on the other hand, we obtain

$$\begin{aligned} A^T(\lambda^k + \Delta\lambda^k - \lambda^{*,k}) + (s^k + \Delta s^k - s^{*,k}) &= 0, && (42)\\ A(x^k + \Delta x^k - x^{*,k}) &= 0. && (43) \end{aligned}$$

Premultiplying (42) with $(x^k + \Delta x^k - x^{*,k})^T$ and taking into account (43) gives

$$(x^k + \Delta x^k - x^{*,k})^T(s^k + \Delta s^k - s^{*,k}) = 0. \tag{44}$$

Using (35), we further have

$$D^k_{a,\tau}(x^k + \Delta x^k - x^{*,k}) + D^k_{b,\tau}(s^k + \Delta s^k - s^{*,k}) = r^k \tag{45}$$

with

$$r^k := D^k_{a,\tau}(x^k - x^{*,k}) + D^k_{b,\tau}(s^k - s^{*,k}) - \phi(x^k, s^k). \tag{46}$$

Having derived these preliminary formulas, we now show in three steps that $\|\Delta x^k\| = O(\tau_k)$, $\|\Delta s^k\| = O(\tau_k)$, and $\|\Delta\lambda^k\| = O(\tau_k)$.

Step 1. In this step, we show that $\|\Delta x^k\| = O(\tau_k)$. Premultiplying (45) with $(x^k + \Delta x^k - x^{*,k})^T(D^k_{b,\tau})^{-1}$, using (44), and taking into account that the matrix product $D^k_{a,\tau}D^k_{b,\tau}$ is positive definite, we obtain

$$\begin{aligned} &\lambda_{\min}(D^k_{a,\tau}D^k_{b,\tau})\,\left\|(D^k_{b,\tau})^{-1}(x^k + \Delta x^k - x^{*,k})\right\|^2\\ \le\;& (x^k + \Delta x^k - x^{*,k})^T(D^k_{b,\tau})^{-1}D^k_{b,\tau}D^k_{a,\tau}(D^k_{b,\tau})^{-1}(x^k + \Delta x^k - x^{*,k})\\ =\;& (x^k + \Delta x^k - x^{*,k})^T D^k_{a,\tau}(D^k_{b,\tau})^{-1}(x^k + \Delta x^k - x^{*,k})\\ =\;& (x^k + \Delta x^k - x^{*,k})^T(D^k_{b,\tau})^{-1}r^k\\ \le\;& \left\|(D^k_{b,\tau})^{-1}(x^k + \Delta x^k - x^{*,k})\right\|\,\|r^k\| \end{aligned}$$

and therefore

$$\lambda_{\min}(D^k_{a,\tau}D^k_{b,\tau}) \left\|(D^k_{b,\tau})^{-1}(x^k + \Delta x^k - x^{*,k})\right\| \leq \|r^k\|. \tag{47}$$

Using the definition of r^k in (46) as well as the fact that

$$\bar{x}_{\mathcal{N}} = 0 \quad \forall \bar{x} \in \mathcal{S}_P$$

and

$$\bar{s}_{\mathcal{B}} = 0 \quad \forall (\bar{\lambda}, \bar{s}) \in \mathcal{S}_D$$

by Lemma 5.2 (a) together with the simple observation that

$$D^k_{a,\tau} + D^k_{b,\tau} = 2I$$

for all $k \in \mathbb{N}$ (recall that φ denotes the minimum function by Assumption 5.1 (a)), we obtain from Lemma 5.2 (b) and Assumption 5.1 (e) that

$$\begin{aligned}
r^k_{\mathcal{B}} &= (2I - D^k_{b,\tau})_{\mathcal{B}\mathcal{B}}(x^k_{\mathcal{B}} - x^{*,k}_{\mathcal{B}}) + (D^k_{b,\tau})_{\mathcal{B}\mathcal{B}}(s^k_{\mathcal{B}} - s^{*,k}_{\mathcal{B}}) - 2\min\{x^k_{\mathcal{B}}, s^k_{\mathcal{B}}\} \\
&= (2I - D^k_{b,\tau})_{\mathcal{B}\mathcal{B}}(x^k_{\mathcal{B}} - x^{*,k}_{\mathcal{B}}) + (D^k_{b,\tau})_{\mathcal{B}\mathcal{B}}s^k_{\mathcal{B}} - 2s^k_{\mathcal{B}} \\
&= (2I - D^k_{b,\tau})_{\mathcal{B}\mathcal{B}}\left(x^k_{\mathcal{B}} - x^{*,k}_{\mathcal{B}} - s^k_{\mathcal{B}} + s^{*,k}_{\mathcal{B}}\right) \\
&= (D^k_{a,\tau})_{\mathcal{B}\mathcal{B}}\left(x^k_{\mathcal{B}} - x^{*,k}_{\mathcal{B}} - s^k_{\mathcal{B}} + s^{*,k}_{\mathcal{B}}\right)
\end{aligned}$$

and

$$\begin{aligned}
r^k_{\mathcal{N}} &= (2I - D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}(x^k_{\mathcal{N}} - x^{*,k}_{\mathcal{N}}) + (D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}(s^k_{\mathcal{N}} - s^{*,k}_{\mathcal{N}}) - 2\min\{x^k_{\mathcal{N}}, s^k_{\mathcal{N}}\} \\
&= (2I - D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}x^k_{\mathcal{N}} + (D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}(s^k_{\mathcal{N}} - s^{*,k}_{\mathcal{N}}) - 2x^k_{\mathcal{N}} \\
&= (D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}\left(s^k_{\mathcal{N}} - s^{*,k}_{\mathcal{N}} - x^k_{\mathcal{N}} + x^{*,k}_{\mathcal{N}}\right).
\end{aligned}$$

Therefore, we have

$$\|r^k\| \leq \max\left\{\|(D^k_{a,\tau})_{\mathcal{B}\mathcal{B}}\|, \|(D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}\|\right\}\left(\|x^k - x^{*,k}\| + \|s^k - s^{*,k}\|\right). \tag{48}$$

Hence (47) implies

$$\begin{aligned}
&\lambda_{\min}(D^k_{a,\tau}D^k_{b,\tau})\|(D^k_{b,\tau})^{-1}(x^k + \Delta x^k - x^{*,k})\| \\
\leq\; &\max\{\|(D^k_{a,\tau})_{\mathcal{B}\mathcal{B}}\|, \|(D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}\|\}\left(\|x^k - x^{*,k}\| + \|s^k - s^{*,k}\|\right).
\end{aligned} \tag{49}$$

Since one can easily verify that the sequence

$$\left\{\frac{\max\{\|(D^k_{a,\tau})_{\mathcal{B}\mathcal{B}}\|, \|(D^k_{b,\tau})_{\mathcal{N}\mathcal{N}}\|\}}{\lambda_{\min}(D^k_{a,\tau}D^k_{b,\tau})}\right\} \tag{50}$$

remains bounded for $k \to \infty$, it follows from (49) and the observation that

$$\|D^k_{b,\tau}\| \leq 2$$

for all $k \in \mathbb{N}$ that

$$\begin{array}{rcl}
\frac{1}{2}\left(\|\Delta x^k\| - \|x^k - x^{*,k}\|\right) & \leq & \left|\|\Delta x^k\| - \|x^k - x^{*,k}\|\right|/2 \\
& \leq & \left|\|\Delta x^k\| - \|x^k - x^{*,k}\|\right|/\|D^k_{b,\tau}\| \\
& \leq & \|x^k + \Delta x^k - x^{*,k}\|/\|D^k_{b,\tau}\| \\
& \leq & \|(D^k_{b,\tau})^{-1}(x^k + \Delta x^k - x^{*,k})\| \\
& \leq & \alpha_1\left(\|x^k - x^{*,k}\| + \|s^k - s^{*,k}\|\right)
\end{array}$$

for some constant $\alpha_1 > 0$. Consequently, we obtain from (36) and Lemma 5.4 that

$$\|\Delta x^k\| \leq \alpha_2\|w^k - w^{*,k}\| = \alpha_2 \mathrm{dist}_{\mathcal{S}}(w^k) = O(\tau_k),$$

where $\alpha_2 > 0$ denotes a suitable constant.

Step 2. Here we show that $\|\Delta s^k\| = O(\tau_k)$. To this end, we premultiply (45) with $(D^k_{b,\tau})^{-1}$ to obtain

$$\begin{array}{rl}
& \|s^k + \Delta s^k - s^{*,k}\| \\
\leq & \|(D^k_{b,\tau})^{-1}r^k\| + \|(D^k_{a,\tau})(D^k_{b,\tau})^{-1}\left(x^k + \Delta x^k - x^{*,k}\right)\| \\
\leq & \|(D^k_{a,\tau})^{-1}(D^k_{b,\tau})^{-1}\|\,\|D^k_{a,\tau}\|\,\|r^k\| + \|D^k_{a,\tau}\|\,\|(D^k_{b,\tau})^{-1}\left(x^k + \Delta x^k - x^{*,k}\right)\| \\
\leq & 2\lambda_{\max}((D^k_{a,\tau})^{-1}(D^k_{b,\tau})^{-1}))\|r^k\| + 2\|(D^k_{b,\tau})^{-1}\left(x^k + \Delta x^k - x^{*,k}\right)\| \\
= & 2\dfrac{1}{\lambda_{\min}(D^k_{a,\tau}D^k_{b,\tau})}\|r^k\| + 2\|(D^k_{b,\tau})^{-1}\left(x^k + \Delta x^k - x^{*,k}\right)\|
\end{array}$$

since

$$\|D^k_{a,\tau}\| \leq 2$$

for all $k \in \mathbb{N}$. Therefore, using (48), (49) and the boundedness of the sequence in (50), we see that

$$\|s^k + \Delta s^k - s^{*,k}\| \leq \alpha_3\left(\|x^k - x^{*,k}\| + \|s^k - s^{*,k}\|\right)$$

for all $k \in \mathbb{N}$ sufficiently large and a constant $\alpha_3 > 0$. Using Lemma 5.4 and (36), this implies

$$\begin{array}{rcl}
\|\Delta s^k\| & \leq & \|s^k + \Delta s^k - s^{*,k}\| + \|s^k - s^{*,k}\| \\
& \leq & \alpha_4\|w^k - w^{*,k}\| \\
& = & \alpha_4 \mathrm{dist}_{\mathcal{S}}(w^k) \\
& = & O(\tau_k)
\end{array} \qquad (51)$$

for all $k \in \mathbb{N}$ sufficiently large and a suitable constant $\alpha_4 > 0$.

Step 3. Using (51), (33), and the full rank of the matrix A immediately implies that

$$\|\Delta\lambda^k\| = O(\tau_k),$$

which proves Step 3.

Putting together Steps 1–3, we finally obtain

$$\|\Delta w^k\| = O(\tau_k)$$

for all $k \in \mathbb{N}$ sufficiently large. □

The next result shows that, eventually, the full predictor direction Δw^k computed in (16) satisfies the neighbourhood condition

$$\|\Phi_{\tau_k}(w^k + \Delta w^k)\| \leq \beta\tau_k,$$

so that Step (S.2) of Algorithm 3.1 is eventually successful.

Lemma 5.6 *We have*

$$\|\Phi_{\tau_k}(w^k + \Delta w^k)\| \leq \beta\tau_k$$

for all $k \in \mathbb{N}$ *sufficiently large.*

Proof. Let us use the abbreviations

$$a^k_{\tau,i} := \frac{\partial\varphi_{\tau_k}}{\partial a}(x^k_i, s^k_i) \quad (i = 1, \dots, n)$$

and

$$b^k_{\tau,i} := \frac{\partial\varphi_{\tau_k}}{\partial b}(x^k_i, s^k_i) \quad (i = 1, \dots, n)$$

for the diagonal elements of the matrices $D^k_{a,\tau}$ and $D^k_{b,\tau}$, respectively. Since

$$a^k_{\tau,i} + b^k_{\tau,i} = 2 \quad \forall i = 1, \dots, n \ \forall k \in \mathbb{N},$$

we obtain from the linear system (16) and Assumption 5.1 (a) that

$$(2 - b^k_{\tau,i})\Delta x^k_i + b^k_{\tau,i}\Delta s^k_i = -2\min\{x^k_i, s^k_i\} \quad (i = 1, \dots, n), \tag{52}$$

cf. (8). Since the sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ converges to a strictly complementary solution $w^* = (x^*, \lambda^*, s^*)$ of the optimality conditions (3) by Assumption 5.1 (e), Lemma 5.2 (b) implies that there is a constant $\delta > 0$ such that

$$x^k_i \geq \delta \ \forall i \in \mathcal{B}, \quad s^k_i \geq \delta \ \forall i \in \mathcal{N} \tag{53}$$

and

$$|x^k_i| < \delta/2 \ \forall i \in \mathcal{N}, \quad |s^k_i| < \delta/2 \ \forall i \in \mathcal{B} \tag{54}$$

and for all $k \in \mathbb{N}$ large enough.

We now consider an arbitrary index $i \in \mathcal{B}$. Using (53) and (54), we get

$$x^k_i - s^k_i \geq \delta/2$$

for all $k \in \mathbb{N}$ sufficiently large. Furthermore, (52) implies

$$(2 - b_{\tau,i}^k)(\Delta x_i^k - \Delta s_i^k) = -2(s_i^k + \Delta s_i^k). \tag{55}$$

The definition of $b_{\tau,i}^k$ gives

$$\begin{aligned}
2 - b_{\tau,i}^k &= 2 - \frac{\partial \varphi_{\tau_k}}{\partial b}(x_i^k, s_i^k) \\
&= 2 - \left(1 + \frac{x_i^k - s_i^k}{\sqrt{(x_i^k - s_i^k)^2 + 4\tau_k^2}}\right) \\
&= 1 - \frac{x_i^k - s_i^k}{\sqrt{(x_i^k - s_i^k)^2 + 4\tau_k^2}} \\
&= \frac{\sqrt{(x_i^k - s_i^k)^2 + 4\tau_k^2} - (x_i^k - s_i^k)}{\sqrt{(x_i^k - s_i^k)^2 + 4\tau_k^2}} \\
&\leq \frac{2\tau_k}{x_i^k - s_i^k} \\
&\leq 4\tau_k/\delta.
\end{aligned}$$

Consequently, we obtain from (55) and Lemma 5.5 that

$$\begin{aligned}
2|s_i^k + \Delta s_i^k| &= (2 - b_{\tau,i}^k)|\Delta x_i^k - \Delta s_i^k| \\
&\leq 4\tau_k \left(\|\Delta x^k\| + \|\Delta s^k\|\right)/\delta \\
&= O(\tau_k^2).
\end{aligned}$$

Since

$$x_i^k + \Delta x_i^k \geq \delta/2$$

in view of (53) and Lemma 5.5, it follows that

$$|\varphi(x_i^k + \Delta x_i^k, s_i^k + \Delta s_i^k)| = 2|\min\{x_i^k + \Delta x_i^k, s_i^k + \Delta s_i^k\}| = 2|s_i^k + \Delta s_i^k| = O(\tau_k^2)$$

for all $i \in \mathcal{B}$.

In a similar way, we can also show that

$$|\varphi(x_i^k + \Delta x_i^k, s_i^k + \Delta s_i^k)| = 2|\min\{x_i^k + \Delta x_i^k, s_i^k + \Delta s_i^k\}| = 2|x_i^k + \Delta x_i^k| = O(\tau_k^2)$$

for all $i \in \mathcal{N}$. Consequently, we have

$$\|\Phi(w^k + \Delta w^k)\| = O(\tau_k^2)$$

because of Assumption 5.1 (b) and Lemma 3.5. Using Lemma 2.5 as well as the fact that $\beta > \kappa$ by Assumption 5.1 (c), we finally obtain

$$\begin{aligned}
\|\Phi_{\tau_k}(w^k + \Delta w^k)\| &\leq \|\Phi(w^k + \Delta w^k)\| + \|\Phi_{\tau_k}(w^k + \Delta w^k) - \Phi(w^k + \Delta w^k)\| \\
&\leq O(\tau_k^2) + \kappa\tau_k \\
&\leq \beta\tau_k
\end{aligned}$$

for all $k \in \mathbb{N}$ sufficiently large since $\tau_k \downarrow 0$ by Assumption 5.1 (d). $\square$

We next show that the updating of the smoothing parameter $\hat{\tau}_k = t_k \tau_k$ in the predictor step of Algorithm 3.1 is such that the factor t_k is in the order of the smoothing parameter τ_k itself.

Lemma 5.7 *We have*

$$t_k = O(\tau_k)$$

for all $k \in \mathbb{N}$ sufficiently large.

Proof. As shown in the proof of Lemma 5.6, we have

$$\|\Phi(w^k + \Delta w^k)\| = O(\tau_k^2).$$

Hence there is a constant $\alpha_5 > 0$ such that

$$\|\Phi(w^k + \Delta w^k)\| \leq \alpha_5 \tau_k^2$$

for all $k \in \mathbb{N}$ large enough. This implies

$$\begin{aligned} \|\Phi_{t\tau_k}(w^k + \Delta w^k)\| &\leq \|\Phi(w^k + \Delta w^k)\| + \|\Phi_{t\tau_k}(w^k + \Delta w^k) - \Phi(w^k + \Delta w^k)\| \\ &\leq \alpha_5 \tau_k^2 + \kappa t \tau_k \end{aligned}$$

for all $t > 0$, cf. Lemma 2.5. Now, an easy calculation shows that the inequality

$$\alpha_5 \tau_k^2 + \kappa t \tau_k \leq \beta t \tau_k$$

holds for all $t \geq \alpha_5 \tau_k / (\beta - \kappa)$ (recall that $\beta > \kappa$ by Assumption 5.1 (c)). Therefore, the definition of t_k in Step (S.2) of Algorithm 3.1 implies that

$$\rho^{\ell_k + 1} = \rho \rho^{\ell_k} = \rho t_k < \frac{\alpha_5}{\beta - \kappa} \tau_k.$$

Hence we have

$$t_k \leq \frac{\alpha_5}{\rho(\beta - \kappa)} \tau_k = O(\tau_k)$$

for all $k \in \mathbb{N}$ large enough. $\square$

We are now in the position to state our main local convergence result for Algorithm 3.1. It says that we eventually have local quadratic convergence of the smoothing parameter τ_k to zero.

Theorem 5.8 *We have*

$$\tau_{k+1} = O(\tau_k^2)$$

for all $k \in \mathbb{N}$ large enough, i.e., the sequence τ_k converges to zero at least Q-quadratically.

Proof. In view of Lemma 5.7 and the updating rules for the smoothing parameter in Algorithm 3.1, we have

$$\tau_{k+1} \leq \hat{\tau}_k = t_k \tau_k = O(\tau_k^2)$$

for all $k \in \mathbb{N}$ sufficiently large. □

We strongly feel that Theorem 5.8 is also true for some of the other NCP-functions mentioned in Section 2, or at least for the Fischer-Burmeister function, but we do not have a formal proof for this.

6 Numerical Results

We implemented Algorithm 3.1 in MATLAB by modifying the LIPSOL code from Zhang [36, 37]. LIPSOL is an interior-point solver for linear programs, written in MATLAB and calling a FORTRAN subroutine in order to solve certain linear systems using the sparse Cholesky method by Ng and Peyton [24]. Since the linear systems occuring in Algorithm 3.1 have essentially the same structure as those arising in primal-dual interior-point methods, it was possible to use the numerical linear algebra part from LIPSOL for our implementation of Algorithm 3.1.

We next discuss our choice of the starting point $w^0 = (x^0, \lambda^0, s^0)$. There are several possibilities for choosing w^0.

1. One could take the starting point provided by LIPSOL. However, this starting point is optimized for interior-point methods, and there is no reason why it should also be useful for our smoothing-type methods. In particular, $w^0 = (x^0, \lambda^0, s^0)$ does, in general, not satisfy the linear equations $A^T\lambda^0 + s^0 = c$ and $Ax^0 = b$, although this might be an important property for our method, see, in particular, Proposition 2.7 and Theorem 4.3.

2. In [15], the authors suggested to use a starting point $w^0 = (x^0, \lambda^0, s^0)$ whose components are computed as follows:

   (a) Solve $AA^Ty = b$ using a sparse Cholesky code in order to compute $y^0 \in \mathbb{R}^m$.

   (b) Set $x^0 := A^Ty^0$.

   (c) Define $\lambda^0 := 0$ and $s^0 := c$.

   Obviously, the resulting vector $w^0 = (x^0, \lambda^0, s^0)$ satisfies the linear equations $A^T\lambda^0 + s^0 = c$ and $Ax^0 = b$. However, the two subvectors x^0 and s^0 typically have many negative components.

3. The starting point $w^0 = (x^0, \lambda^0, s^0)$ we use in our implementation is calculated as follows:

   (a) Solve $AA^Ty = b$ using a sparse Cholesky code in order to compute $y^0 \in \mathbb{R}^m$.

   (b) Set $x^0 := A^Ty^0$.

(c) Solve $AA^T\lambda = Ac$ using a sparse Cholesky code in order to compute $\lambda^0 \in \mathbb{R}^m$.

(d) Set $s^0 := c - A^T\lambda^0$.

Again, the resulting vector $w^0 = (x^0, \lambda^0, s^0)$ satisfies the linear equations $A^T\lambda^0 + s^0 = c$ and $Ax^0 = b$. The difference between this starting vector and the one described in the previous procedure lies in the computation of the vector λ^0. The idea behind the strategy here is to have possibly many components of s^0 equal to zero (or at least small) by solving the equation $A^T\lambda = c$ in the least squares sense. The advantage of having zero (or small) components in s^0 is that the initial residual $\|\Phi(w^0)\|$ is smaller, so that this starting point w^0 is hopefully closer to the solution set of (3).

We stress, however, that the choice of the starting point is crucial and that the procedure given in 3. is not always better than the one stated in 2. Furthermore, since interior-point and smoothing-type methods use different starting points, it is difficult to compare both methods with each other.

Another crucial part of our implementation is the stopping criterion. Interior-point methods typically terminate the iteration if the (weighted) duality gap

$$\mu_k := (x^k)^T s^k / n \tag{56}$$

(or a certain residual vector whose size is controlled by the duality gap) is sufficiently small in the sense that

$$\mu_k < 10^{-8}.$$

This criterion cannot be used by our method since the x- and s-components of the iterates $w^k = (x^k, \lambda^k, s^k)$ are no longer guaranteed to stay positive. Hence we are urged to use a different criterion. The one implemented is

$$\tau_k < 10^{-4}, \tag{57}$$

where, of course, τ_k denotes our smoothing parameter. The motivation for this stopping rule is as follows: Using the number μ_k from (56), interior-point methods basically follow the central path conditions

$$\begin{aligned} A^T\lambda + s &= c, \\ Ax &= b, \\ x_i > 0,\, s_i > 0,\, x_i s_i &= \mu_k \quad \forall i = 1, \dots, n \end{aligned}$$

parameterized by μ_k. Comparing with (14), it therefore follows that μ_k plays (more or less) the role of τ_k^2, hence the criterion $\mu_k < 10^{-8}$ corresponds to $\tau_k^2 < 10^{-8}$ or, equivalently, to $\tau_k < 10^{-4}$. Moreover, in view of Proposition 3.4 and Lemma 2.5, the violation $\|\Phi(w^k)\|$ is also bounded from above by a constant times τ_k.

After some preliminary experiments with Algorithm 3.1, we observed that the predictor step gets accepted only a very few times. Moreover, the situation did not improved by allowing some kind of restricted line search in the predictor step; by this, we mean that if the full step Δw^k did not satisfy the neighbourhood criterion (21), we replaced Δw^k by, say,

$1/2\Delta w^k$ or even $1/4\Delta w^k$ and accepted this step in case (21) is satisfied with this shorter direction.

On the other hand, it turned out that this strategy seems to work more successful if we replace the right-hand side in (16) by $\Phi_{\tau_k}(w^k)$, so that the predictor step becomes a real smoothing step. Since this does not necessarily guarantee local fast convergence under Assumption 5.1, we finally decided to implement a three-step modification of Algorithm 3.1. This modification adds another step between the predictor step (S.2) and the corrector step (S.3), let us call it step (S.2a). Step (S.2a) is exactly the same as step (S.2) except that the right-hand side in (16) is replaced by $\Phi_{\tau_k}(w^k)$. In the following, we will call step (S.2) the *Jacobian smoothing predictor step*, and step (S.2a) the *smoothing predictor step*, while step (S.3) will still be called the *corrector step*. Obviously, this modified method has the same local and global convergence properties as Algorithm 3.1, but it seems to work better from a practical point of view.

On the other hand, the amount of work per iteration is higher for this three-step method than for Algorithm 3.1. In theory, it could happen that we have to solve up to three linear systems with different coefficient matrices, so that we have to factorize all these matrices, whereas in Algorithm 3.1 we have to factorize at most two matrices per iteration. However, if the Jacobian smoothing predictor step and/or the smoothing predictor step cannot be accepted, then all or at least two of the coefficient matrices are identical. In this case, we still solve three linear systems, but the amount of work is significantly smaller since we factorize only one or two matrices per iteration.

In practice, averaging over the test suite used for our numerical experience, it turns out that we have to factorize less than two matrices per iteration. We should mention, however, that this is still more than what is done by LIPSOL and many other implementations of interior-point methods which typically solve two linear systems per iteration having the same coefficient matrix so that only one factorization is needed.

Taking into account the previous modifications, we implemented Algorithm 3.1 using the penalized minimum function and the parameters

$$\tau_0 := \min\{\|\Phi(w^0)\|, 10^5\},\ \beta := \|\Phi_{\tau_0}(w^0)\|/\tau_0,\ \rho = 0.9,\ \sigma = 10^{-4}.$$

We run the program on a HP Hyperclass HP9000 V2250 machine using a PC-8200 processor with 240 MHz. The results for the complete `netlib` test suite are given in Table 1, whose columns have the following meanings:

problem:	Name of the test problem in the `netlib` collection
m:	number of rows (after preprocessing)
n:	number of columns (after preprocessing)
k:	number of iterations
J:	number of accepted Jacobian smoothing predictor steps
S:	number of accepted smoothing predictor steps
τ_f:	value of τ_k in final iterate
$\|\Phi(w^f)\|_\infty$:	value of $\|\Phi(w^k)\|_\infty$ in final iterate
primal obj.:	Value of primal objective function in final iterate.

In addition to the termination criterion (57), we stopped each test run after at most 150 iterations or if the steplength in the corrector step became too small.

Table 1: Numerical results for the three step method

problem	m	n	k	J	S	τ_f	$\|\Phi(w^f)\|_\infty$	primal obj.
25fv47	798	1854	20	4	15	4.2343e−05	2.9722e−03	5.5018457227e+03
80bau3b	2235	11516	37	5	22	8.7473e−05	1.3443e−02	9.8722419238e+05
adlittle	55	137	10	4	6	4.9199e−05	6.7157e−04	2.2549495986e+05
afiro	27	51	6	6	4	2.1099e−07	2.6817e−06	−4.6473928249e+02
agg	488	615	24	7	14	3.3230e−06	1.3124e−04	−3.5991766783e+07
agg2	516	758	21	8	17	6.0639e−08	2.6978e−06	−2.0239252356e+07
agg3	516	758	23	5	15	2.4600e−05	3.8846e−04	1.0312115935e+07
bandm	269	436	13	6	9	1.7564e−05	3.1625e−04	−1.5862801963e+02
beaconfd	148	270	17	4	12	1.9366e−05	3.3039e−04	3.3592485806e+04
blend	74	114	8	3	6	4.9968e−06	8.5378e−05	−3.0812150689e+01
bnl1	632	1576	21	5	16	9.0798e−05	2.1921e−03	1.9776272620e+03
bnl2	2268	4430	29	4	21	3.5118e−05	2.4411e−03	1.8112365402e+03
boeing1	347	722	21	4	17	2.9805e−05	5.5407e−04	−3.3521410246e+02
boeing2	140	279	20	5	14	8.0306e−05	2.6954e−03	−3.1501872383e+02
bore3d	199	300	26	5	12	1.2712e−06	2.8822e−05	1.3730803942e+03
brandy	149	259	15	8	10	1.2171e−05	3.1526e−04	1.5185098963e+03
capri	267	476	13	6	12	6.2009e−05	2.0095e−03	2.6900128345e+03
cycle	1801	3305	104	1	14	7.7353e−05	2.9079e−03	−5.1750493388e+00
czprob	737	3141	22	4	12	3.4584e−06	2.0349e−04	2.1851966989e+06
d2q06c	2171	5831	29	2	20	7.6625e−05	2.0615e−03	1.2278428380e+05
d6cube	404	6184	8	5	6	5.8390e−05	8.8236e−03	3.1549166040e+02
degen2	444	757	6	5	4	1.2860e−05	2.7119e−04	−1.4351780027e+03
degen3	1503	2604	9	5	6	1.3941e−05	3.1761e−04	−9.8729401032e+02
dfl001	6071	12230	—	—	—	—	—	—
e226	220	469	11	7	9	6.9197e−05	2.2793e−03	−1.8751926034e+01
etamacro	357	692	16	4	15	4.7998e−05	7.1797e−04	−7.5571524102e+02
fffff800	501	1005	33	6	23	3.7494e−05	9.5605e−04	5.5567957806e+05
finnis	492	1014	21	3	20	5.9699e−05	1.3796e−03	1.7279109279e+05
fit1d	24	1049	27	6	12	1.1214e−08	6.9938e−07	−9.1463780924e+03
fit1p	627	1677	—	—	—	—	—	—
fit2d	25	10524	13	6	6	8.0765e−06	1.5959e−03	−6.8464293130e+04
fit2p	3000	13525	—	—	—	—	—	—
forplan	135	463	34	3	17	1.2676e−05	4.2004e−04	−6.6421823582e+02
ganges	1137	1534	18	7	16	2.4600e−05	1.1170e−03	−1.0958573613e+05
gfrd-pnc	600	1144	30	5	11	3.0370e−05	7.2264e−04	6.9022359998e+06
greenbea	2318	5424	56	6	15	6.6930e−05	9.6818e−03	−7.2462429119e+07
greenbeb	2317	5415	42	4	26	1.0204e−05	1.1343e−03	−4.3022600830e+06
israel	174	316	29	5	15	1.8046e−06	4.9876e−05	−8.9664482215e+05
kb2	43	68	19	7	4	7.8250e−05	3.0236e−04	−1.7499001320e+03
lotfi	151	364	38	3	18	5.7195e−05	7.7849e−04	−2.5264485771e+01
maros	835	1921	57	5	25	2.2754e−05	1.8723e−03	−5.8063192480e+04
maros-r7	3136	9408	21	6	7	1.7667e−05	3.3476e−03	1.4971851665e+06

Table 1 (continued): Numerical results for the three step method

problem	m	n	k	J	S	τ_f	$\|\Phi(w^f)\|_\infty$	primal obj.
modszk1	686	1622	31	9	16	6.8418e−07	4.7751e−05	3.2061972882e+02
nesm	654	2922	32	3	25	8.0923e−05	4.8817e−03	1.4076036486e+07
perold	625	1530	58	3	24	5.3991e−05	2.4544e−03	−9.3806515651e+03
pilot	1441	4657	43	3	23	4.4615e−05	5.5343e−03	−5.5748462201e+02
pilotja	924	2044	98	1	27	9.6231e−05	3.1893e−03	−6.1120672246e+03
pilotwe	722	2930	34	1	28	6.9184e−05	2.4072e−03	−2.7201100292e+06
pilot4	402	1173	41	3	22	4.7978e−05	1.3641e−03	−2.5811349175e+03
pilot87	2030	6460	109	3	23	9.8681e−05	4.3860e−03	3.0175176854e+02
pilotnov	951	2242	47	4	16	2.8005e−05	2.4620e−03	−4.4972761883e+03
recipe	85	177	7	4	5	6.9831e−06	9.1921e−05	−2.6661600085e+02
sc105	105	163	9	4	7	2.8833e−05	9.1980e−05	−5.2202055953e+01
sc205	205	317	30	4	5	6.2525e−06	1.0522e−04	−5.2202061300e+01
sc50a	49	77	11	4	5	2.8432e−05	2.1298e−04	−6.4575051634e+01
sc50b	48	76	8	4	3	5.5517e−05	2.4751e−05	−6.9999991662e+01
scagr25	471	671	27	5	10	3.9984e−10	1.3256e−08	−1.4753433061e+07
scagr7	129	185	15	6	8	5.8858e−07	1.1136e−05	−2.3313898243e+06
scfxm1	322	592	15	5	12	9.1602e−05	3.7990e−03	1.8416759024e+04
scfxm2	644	1184	19	5	11	2.1761e−06	1.4029e−04	3.6660261565e+04
scfxm3	966	1776	19	7	14	6.8249e−07	2.8367e−05	5.4901254550e+04
scorpion	375	453	15	3	10	2.9324e−06	7.2669e−05	1.8781248227e+03
scrs8	485	1270	13	4	11	8.5648e−05	3.9037e−03	9.0429695222e+02
scsd1	77	760	6	4	5	5.7306e−11	8.1876e−10	8.6666666743e+00
scsd6	147	1350	7	5	4	1.7731e−05	2.2098e−04	5.0500000072e+01
scsd8	397	2750	4	4	4	5.6900e−05	3.5013e−03	9.0498687573e+02
sctap1	300	660	13	5	5	4.6773e−05	2.2754e−03	1.4122500000e+03
sctap2	1090	2500	11	6	7	4.9132e−07	1.6708e−05	1.7248071429e+03
sctap3	1480	3340	9	5	6	3.7794e−06	2.8174e−04	1.4240000000e+03
seba	515	1036	—	—	—	—	—	—
share1b	112	248	33	6	22	9.4543e−05	1.2943e−03	−7.6589318582e+04
share2b	96	162	10	2	8	7.6299e−07	1.0478e−05	−4.1573224076e+02
shell	496	1487	19	7	15	3.9179e−09	9.6377e−08	1.2088253460e+09
ship04l	356	2162	17	5	7	1.3961e−06	1.1316e−04	1.7933245380e+06
ship04s	268	1414	14	6	5	1.8627e−08	9.2378e−07	1.7987147004e+06
ship08l	688	4339	20	8	6	2.7510e−08	3.2405e−06	1.9090552115e+06
ship08s	416	2171	18	7	5	6.3046e−06	4.6767e−04	1.9200982106e+06
ship12l	838	5329	16	5	9	5.5636e−05	5.1779e−03	1.4701879193e+06
ship12s	466	2293	14	5	7	2.3603e−06	1.6594e−04	1.4892361344e+06
sierra	1222	2715	18	9	9	4.4889e−07	2.0652e−05	1.5394362288e+07
stair	356	538	12	6	10	1.9625e−05	4.3062e−04	−2.5126806092e+02
standata	359	1258	13	6	8	1.9143e−06	8.0021e−05	1.2576993389e+03
standgub	361	1366	7	5	5	2.2919e−08	5.6933e−07	1.2576994071e+03
standmps	467	1258	13	5	7	1.8569e−08	6.3709e−07	1.4060175000e+03

Table 1 (continued): Numerical results for the three step method

problem	m	n	k	J	S	τ_f	$\|\Phi(w^f)\|_\infty$	primal obj.
stocfor1	109	157	18	2	6	6.2146e−06	8.9620e−05	−4.1131976482e+04
stocfor2	2157	3045	32	5	10	4.5268e−05	3.0008e−03	−3.9024408555e+04
stocfor3	16675	23541	63	4	18	1.1011e−05	6.4996e−04	−3.9976783964e+04
stocfor3old	16675	23541	43	3	21	7.8870e−05	1.9968e−02	−3.9976739946e+04
truss	1000	8806	15	7	6	2.2041e−07	1.5358e−05	4.5881584718e+05
tuff	292	617	22	5	17	9.9787e−05	3.8650e−03	2.9214732232e−01
vtpbase	194	325	13	7	10	2.3190e−07	8.2976e−06	1.2983146246e+05
wood1p	244	2595	10	5	7	4.3562e−05	3.4471e−03	1.4429024110e+00
woodw	1098	8418	15	2	12	6.2276e−05	6.9536e−03	1.3044743048e+00

Table 2: Numerical results for the `grow*` examples (authors' initial point)

problem	m	n	k	J	S	τ_f	$\|\Phi(w^f)\|_\infty$	primal obj.
grow7	140	301	130	3	5	9.9175e−07	1.9719e−05	−4.7787811822e+07
grow15	300	645	201	3	6	4.1590e−07	8.0793e−06	−1.0687094132e+08
grow22	440	946	295	3	5	1.0204e−09	1.4146e−08	−1.6083433648e+08

Table 3: Numerical results for the `grow*` examples (LIPSOL's initial point)

problem	m	n	k	J	S	τ_f	$\|\Phi(w^f)\|_\infty$	primal obj.
grow7	140	301	33	4	11	3.8723e−10	5.1334e−09	−4.7787811815e+07
grow15	300	645	29	5	12	6.6616e−10	1.1677e−08	−1.0687094129e+08
grow22	440	946	24	4	12	5.0103e−05	9.5985e−04	−1.6083433738e+08

Table 4: Local behaviour of `grow22` (authors' initial point)

k	τ_k	$\|\Phi(w^k)\|_\infty$	primal obj.
291	1.0544e+00	2.5478e+00	−1.6083406624e+08
292	1.0544e+00	1.3278e+00	−1.6083431197e+08
293	5.6036e−01	1.3278e+00	−1.6083431197e+08
294	5.3514e−04	4.6024e−06	−1.6083433648e+08
295	1.0204e−09	1.4146e−08	−1.6083433648e+08

The results given in Table 1 are quite good although we have four failures on the test problems `dfl001`, `fit1p`, `fit2p` and `seba` (note that LIPSOL also produces a failure on dfl001, at least on our machine). In particular, the results are significantly better than those presented in [15] for a pure Jacobian smoothing method.

Looking at the results in Table 1, we see that most test problems can be solved in less than 30 iterations. Furthermore, there are typically only a few number of Jacobian smoothing predictor steps which were accepted by our algorithm, whereas the number of accepted smoothing predictor steps is usually higher. Moreover, we note that the method works particularly well whenever the number of accepted predictor steps (either Jacobian smoothing or smoothing) is relatively high.

Compared with interior-point methods (although a direct comparison is not really possible), our smoothing-type method seems to be less robust. However, the results given here should be viewed as a kind of starting point for using smoothing methods for linear programs and related problems.

Finally, we mention that we have excluded the three test problems grow7, grow15 and grow22 from Table 1. The corresponding results for these problems are given in Table 2. This table indicates that our method behaves pretty bad on these examples, and we currently do not have an explanation for this. However, it is interesting to note that the results are getting significantly better if we use the starting point provided by LIPSOL. Table 3 contains the corresponding results.

Finally, in Table 4 we present some data from the last few iterations for problem grow22 (using our starting point). It clearly shows that our method is locally quadratically convergent for this example, although the number of iterations is very high in order to get into the region of fast convergence.

7 Concluding Remarks

In this manuscript, we investigated the theoretical and numerical properties of a predictor-corrector smoothing-type method for the solution of linear programs. Compared with other smoothing methods previously suggested mainly for the solution of complementarity problems, the theoretical properties are quite strong, and numerical results for such a method applied to linear programs are not available with the only exception of the talk [4] and the authors' previous paper [15].

On the other hand, the somewhat related class of primal-dual interior-point methods seems to work slightly better, but we feel that there is still enough room for several improvements (both theoretical and numerical) in the field of smoothing-type methods. For example, we are currently working on a modification of Algorithm 3.1 which should accept the predictor steps more frequently, and we therefore hope to report on some numerical improvements in the not too far future.

References

[1] D. Bertsimas and J.N. Tsitsiklis: *Introduction to Linear Optimization.* Athena Scientific, Belmont, MA, 1997.

[2] J.V. Burke and S. Xu: *The global linear convergence of a noninterior path-following algorithm for linear complementarity problems.* Mathematics of Operations Research 23, 1998, pp. 719–734.

[3] J.V. Burke and S. Xu: *A non-interior predictor-corrector path-following method for LCP.* In: M. Fukushima and L. Qi (eds.): *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods.* Kluwer Academic Publishers, 1999.

[4] J.V. Burke and S. Xu: *Preliminary numerical experience with non-interior path following methods for LCP.* Talk presented at the *International Conference on Nonlinear Programming and Variational Inequalities*, Hongkong, December 1998.

[5] J.V. Burke and S. Xu: *A non-interior-predictor-corrector path following algorithm for the monotone linear complementarity problem.* Mathematical Programming 87, 2000, pp. 113–130.

[6] B. Chen and X. Chen: *A global and local superlinear continuation-smoothing method for P_0 and R_0 NCP or monotone NCP.* SIAM Journal on Optimization 9, 1999, pp. 624–645.

[7] B. Chen, X. Chen and C. Kanzow: *A penalized Fischer-Burmeister NCP-function.* Mathematical Programming, to appear.

[8] B. Chen and P.T. Harker: *A non-interior-point continuation method for linear complementarity problems.* SIAM Journal on Matrix Analysis and Applications 14, 1993, pp. 1168–1190.

[9] B. Chen and P.T. Harker: *Smooth approximations to nonlinear complementarity problems.* SIAM Journal on Optimization 7, 1997, pp. 403–420.

[10] B. Chen and N. Xiu: *A global linear and local quadratic noninterior continuation method for nonlinear complementarity problems based on Chen-Mangasarian smoothing functions.* SIAM Journal on Optimization 9, 1999, pp. 605–623.

[11] C. Chen and O.L. Mangasarian: *A class of smoothing functions for nonlinear and mixed complementarity problems.* Computational Optimization and Applications 5, 1996, pp. 97–138.

[12] X. Chen, L. Qi and D. Sun: *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities.* Mathematics of Computation 67, 1998, pp. 519–540.

[13] G.B. Dantzig: *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963.

[14] J.E. Dennis and R.B. Schnabel: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, NJ, 1983 (reprinted by SIAM, Philadelphia, PA, 1996).

[15] S. Engelke and C. Kanzow: *On the solution of linear programs by Jacobian smoothing methods.* Annals of Operations Research, to appear.

[16] A. Fischer: *A special Newton-type optimization method.* Optimization 24, 1992, pp. 269–284.

[17] S.A. Gabriel and J.J. Moré: *Smoothing of mixed complementarity problems.* In: M.C. Ferris and J.-S. Pang (eds.): *Complementarity and Variational Problems: State of the Art.* SIAM, Philadelphia, PA, 1997, pp. 105–116.

[18] K. Hotta and A. Yoshise: *Global convergence of a class of non-interior point algorithms using Chen-Harker-Kanzow-Smale functions for nonlinear complementarity problems.* Mathematical Programming 86, 1999, pp. 105–133.

[19] H. Jiang: *Smoothed Fischer-Burmeister equation methods for the complementarity problem.* Technical Report, Department of Mathematics, University of Melbourne, Melbourne, Australia, June 1997.

[20] C. Kanzow: *Some noninterior continuation methods for linear complementarity problems.* SIAM Journal on Matrix Analysis and Applications 17, 1996, pp. 851–868.

[21] C. Kanzow: *A new approach to continuation methods for complementarity problems with uniform P-functions.* Operations Research Letters 20, 1997, pp. 85–92.

[22] C. Kanzow and H. Jiang: *A continuation method for (strongly) monotone variational inequalities.* Mathematical Programming 81, 1998, pp. 103–125.

[23] C. Kanzow and H. Pieper: *Jacobian smoothing methods for nonlinear complementarity problems.* SIAM Journal on Optimization 9, 1999, pp. 342–373.

[24] E. Ng and B.W. Peyton: *Block sparse Cholesky algorithm on advanced uniprocessor computers.* SIAM Journal on Scientific Computing 14, 1993, pp. 1034–1056.

[25] J.-S. Pang: *Newton's method for B-differentiable equations.* Mathematics of Operations Research 15, 1990, pp. 311–341.

[26] L. Qi, D. Sun and G. Zhou: *A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities.* Mathematical Programming 87, 2000, pp. 1–35.

[27] S.M. Robinson: *Some continuity properties of polyhedral multifunctions.* Mathematical Programming Study 14, 1981, pp. 206–214.

[28] C. Roos, T. Terlaky and J.-Ph. Vial: *Theory and Algorithms for Linear Optimization: An Interior Point Approach.* John Wiley & Sons, Chichester, 1997.

[29] S. Smale: *Algorithms for solving equations.* In *Proceedings of the International Congress of Mathematicians.* AMS, Providence, 1987, pp. 172–195.

[30] P. Tseng: *Analysis of a non-interior continuation method based on Chen-Mangasarian smoothing functions for complementarity problems.* In: M. Fukushima and L. Qi (eds.): *Reformuation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods.* Kluwer Academic Publishers, 1998, pp. 381–404.

[31] P. Tseng: *Error bounds and superlinear convergence analysis of some Newton-type methods in optimization.* In: G. Di Pillo and F. Giannessi (eds.): *Nonlinear Optimization and Related Topics.* Kluwer Academic Publishers, to appear.

[32] S.J. Wright: *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, PA, 1997.

[33] S. Xu: *The global linear convergence and complexity of a non-interior path-following algorithm for monotone LCP based on Chen-Harker-Kanzow-Smale smoothing functions.* Technical Report, Department of Mathematics, University of Washington, Seattle, WA, February 1997.

[34] S. Xu and J.V. Burke: *A polynomial time interior-point path-following algorithm for LCP based on Chen-Harker-Kanzow smoothing techniques.* Mathematical Programming 86, 1999, pp. 91–103.

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

[36] Y. Zhang: *Solving large-scale linear programs by interior-point methods under the MATLAB environment.* Optimization Methods and Software 10, 1998, pp. 1–31.

[37] Y. Zhang: *User's guide to LIPSOL: Linear programming interior point solver v0.4.* Optimization Methods and Software 11 & 12, 1999, pp. 385–396.