Hamburger Beiträge
zur Angewandten Mathematik

On the Solution of Linear Programs by Jacobian Smoothing Methods

Stephan Engelke and Christian Kanzow

Reihe A
Preprint 149
December 1999

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

ON THE SOLUTION OF LINEAR PROGRAMS BY JACOBIAN SMOOTHING METHODS[1]

Stephan Engelke and Christian Kanzow

University of Hamburg
Institute of Applied Mathematics
Bundesstrasse 55
20146 Hamburg
Germany
e-mail: engelke@math.uni-hamburg.de
kanzow@math.uni-hamburg.de

November 30, 1999 (revised December 1, 1999)

Abstract. We introduce a class of algorithms for the solution of linear programs. This class is motivated by some recent methods suggested for the solution of complementarity problems. It reformulates the optimality conditions of a linear program as a nonlinear system of equations and applies a Newton-type method to this system of equations. We investigate the global and local convergence properties and present some numerical results. The algorithms introduced here are somewhat related to the class of primal-dual interior-point methods. Although, at this stage of our research, the theoretical results and the numerical performance of our method are not as good as for interior-point methods, our approach seems to have some advantages which will also be discussed in detail.

Key Words. Linear programs, Newton-type method, smoothing method, interior-point method, global convergence, quadratic convergence.

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

1 Introduction

Consider the linear program in standard form

$$\min\ c^T x \quad \text{subject to} \quad Ax = b,\ x \geq 0, \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ has full rank, $c \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$. As usual, we will call (1) the primal problem. The corresponding dual problem is given by

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

where $\lambda \in \mathbb{R}^m$ denotes the dual variable, and $s \in \mathbb{R}^n$ is a nonnegative slack variable. Both the primal and the dual linear programs have the same optimality conditions, namely

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

Consequently, the primal problem (1) has an optimal solution $x^* \in \mathbb{R}^n$ if and only if the dual problem (2) has an optimal solution. Moreover, any of these two conditions is equivalent to the solvability of the optimality conditions (3). Hence solving the optimality conditions (3) is completely equivalent to solving the original linear program (1). This well-known observation is the basis of our approach.

In this approach, we use some recent ideas from the field of complementarity problems (see, e.g., [7] for an algorithmic survey) in order to reformulate the optimality conditions (3) as a nonlinear system of equations

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

(The precise definition of Φ will be given in Section 2.) We then try to solve this system by a Newton-type method in order to get a solution of the linear program (1). However, since Φ is nonsmooth in general, we cannot use the classical Newton method for the solution of (4). Furthermore, the Jacobian matrices $\Phi'(x, \lambda, s)$ turn out to be singular even at some differentiable points (x, λ, s). Consequently, it is also not advisable to apply a nonsmooth Newton method (see [13, 15, 14]) to (4) since these nonsmooth Newton methods coincide with the classical one at all continuously differentiable points and, therefore, would also have to deal with singular Jacobian matrices.

On the other hand, a closer look at the structure of these singular Jacobian matrices shows that one can avoid the singularity by an arbitrarily small perturbation of certain matrix entries. This motivates the use of some perturbed nonsmooth Newton methods, see, e.g., Fischer [9] as well as Yamashita and Fukushima [19] for two examples in the context of complementarity problems. A slightly different and quite elegant form of a perturbed nonsmooth Newton method was recently introduced by Chen, Qi, and Sun [5], see also Kanzow and Pieper [12]. In fact, the method to be discussed in this manuscript is precisely the method from [5], but specialized to linear programs.

The details of this method are given in Section 2. The global and local convergence properties are presented in Sections 3 and 4, respectively. We stress that our analysis is quite similar to the one from [5], but that one has to be a bit more careful here. In fact, the assumptions used in [5] in order to establish global convergence are not satisfied for linear programs; moreover, in the local convergence part, we can exploit special properties of linear programs to get a complete characterization of the situation where the optimality conditions (3) have a unique solution. Section 5 then contains some numerical results for our method applied to the netlib test problem collection. Finally, we compare our method with the class of primal-dual interior-point methods in Section 6.

The notation used in this manuscript is rather standard: The Euclidean vector norm and its associated matrix norm are denoted by $\|\cdot\|$, whereas $\|\cdot\|_F$ stands for the Frobenius norm of matrices. Furthermore, in order to simplify our notation, we write (x, λ, s) instead of the more correct $(x^T, \lambda^T, s^T)^T$, where $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$, and $s \in \mathbb{R}^n$ are given vectors.

2 Jacobian Smoothing Method

The aim of this section is to give a detailed description of our method for the solution of the linear program (1) via its optimality conditions (3). To this end, the notion of an NCP-function turns out to be very helpful.

Definition 2.1 *A function* $\varphi : \mathbb{R}^2 \to \mathbb{R}$ *is called an* NCP-function *if*

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

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

$$\Phi(x, \lambda, s) := \begin{pmatrix} A^T\lambda + s - c \\ Ax - b \\ \phi(x, s) \end{pmatrix}, \tag{5}$$

where

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

Then the following equivalence is obvious:

$$(x^*, \lambda^*, s^*) \text{ solves (3)} \Longleftrightarrow (x^*, \lambda^*, s^*) \text{ solves } \Phi(x, \lambda, s) = 0.$$

The solution of the optimality conditions (3) can therefore be reduced to the solution of a nonlinear system of equations. This system of equations depends heavily on the choice of the NCP-function φ.

Two important examples of an NCP-function are the *minimum function*

$$\varphi(a, b) := 2\min\{a, b\} \tag{6}$$

(the factor 2 is used here only for cosmetical reasons) and the *Fischer-Burmeister function* [8]

$$\varphi(a, b) := a + b - \sqrt{a^2 + b^2}. \tag{7}$$

Both functions are nonsmooth, but can easily be approximated by smooth functions. For example, let φ denote the minimum function. This function can be rewritten in the form

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

Although the expression on the right-hand side looks more complicated, it clearly indicates that the minimum function can be approximated by the so-called *Chen-Harker-Kanzow-Smale smoothing function* [4, 10, 16]

$$\varphi_\tau(a,b) := a+b-\sqrt{(a-b)^2+4\tau^2}, \tag{8}$$

where $\tau \geq 0$ denotes the smoothing parameter. Note that we have $\varphi = \varphi_\tau$ for $\tau = 0$, and that φ_τ is continuously differentiable for any $\tau > 0$. Similarly, we can approximate the Fischer-Burmeister function (7) by

$$\varphi_\tau(a,b) := a+b-\sqrt{a^2+b^2+2\tau^2}, \tag{9}$$

see [10]. Throughout this manuscript, φ always denotes either the minimum function (6) or the Fischer-Burmeister function (7), while φ_τ always denotes the corresponding smooth approximation given in (8) and (9), respectively.

Next, let us define the operator $\Phi_\tau : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ by

$$\Phi_\tau(x,\lambda,s) := \begin{pmatrix} A^T\lambda + s - c \\ Ax - b \\ \phi_\tau(x,s) \end{pmatrix}, \tag{10}$$

where

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

Obviously, Φ_τ may be viewed as a continuously differentiable approximation of the nonsmooth operator Φ.

The next result states that the smoothed functions φ_τ are indeed good approximations of the nonsmooth functions φ (see also [11]).

Lemma 2.2 *There exists a constant $c > 0$ (independent of τ and (a,b)) such that*

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

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

Proof. In case of the minimum function, we obtain

$$\begin{aligned}
|\varphi(a,b) - \varphi_\tau(a,b)| &= \left| a+b-\sqrt{(a-b)^2} - \left(a+b-\sqrt{(a-b)^2+4\tau^2}\right)\right| \\
&= \sqrt{(a-b)^2+4\tau^2} - \sqrt{(a-b)^2} \\
&= \frac{(a-b)^2+4\tau^2-(a-b)^2}{\sqrt{(a-b)^2+4\tau^2}+\sqrt{(a-b)^2}} \\
&\leq \frac{4\tau^2}{\sqrt{4\tau^2}} \\
&= 2\tau
\end{aligned}$$

for all $(a,b) \in \mathbb{R}^2$ and all $\tau > 0$, i.e., the statement holds with $c := 2$.

In a similar way, one can verify that the statement holds with $c := \sqrt{2}$ for the Fischer-Burmeister function. $\square$

As a direct consequence, we obtain the following result, where Φ and Φ_τ denote the mappings defined in (5) and (10), respectively.

Lemma 2.3 *There exists a constant $\kappa > 0$ (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. Using Lemma 2.2 and the definition of the mappings Φ and Φ_τ, we obtain for the minimum function

$$\begin{aligned}
\|\Phi(w) - \Phi_\tau(w)\| &= \sqrt{\sum_{i=1}^{n} [\varphi(x_i, s_i) - \varphi_\tau(x_i, s_i)]^2} \\
&\leq \sqrt{\sum_{i=1}^{n} c^2\tau^2} \\
&= \sqrt{nc^2\tau^2} \\
&= \sqrt{n}c\tau,
\end{aligned}$$

i.e., the statement holds with $\kappa := c\sqrt{n} = 2\sqrt{n}$.

In a similar way, one can show that the assertion holds for the Fischer-Burmeister function with $\kappa := \sqrt{2n}$. $\square$

Note that the constant κ introduced in Lemma 2.3 is actually known. In fact, this will play an important role in the design of our algorithm where this constant will be used explicitly.

We are now in the position to state our algorithm for the solution of (3). Basically, this method is a Newton-type method for the solution of the system $\Phi(x, \lambda, s) = 0$. However, instead of solving the corresponding Newton equation

$$\Phi'(x^k, \lambda^k, s^k) \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = -\Phi(x^k, \lambda^k, s^k)$$

at each iteration with a possibly singular or not existing Jacobian $\Phi'(x^k, \lambda^k, s^k)$, we solve a linear system of the form

$$\Phi'_{\tau_k}(x^k, \lambda^k, s^k) \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = -\Phi(x^k, \lambda^k, s^k)$$

for some $\tau_k > 0$. This guarantees that the Jacobian of Φ_{τ_k} exists; moreover, as we will see later, this matrix is always nonsingular. However, the search direction computed in this way is, in general, not a descent direction for the natural merit function

$$\Psi(x, \lambda, s) := \frac{1}{2}\Phi(x, \lambda, s)^T\Phi(x, \lambda, s) = \frac{1}{2}\|\Phi(x, \lambda, s)\|^2,$$

but it turns out to be a descent direction for the smoothed merit function

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

with $\tau = \tau_k$. The precise algorithm is as follows, see also [5].

Algorithm 2.4 *(Jacobian Smoothing Method)*

(S.0) Choose $\rho, \alpha, \eta \in (0, 1), \varepsilon > 0, \sigma \in (0, 1 - \alpha)$ *and* $w^0 := (x^0, \lambda^0, s^0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$. *Set* $\beta_0 := \|\Phi(w^0)\|, \tau_0 := \frac{\alpha}{2\kappa}\beta_0$ *and* $k := 0$.

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

(S.2) 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*

$$\Phi'_{\tau_k}(w^k)\Delta w = -\Phi(w^k). \tag{11}$$

(S.3) Compute $t_k = \max\{\rho^\ell \mid \ell = 0, 1, 2, \dots\}$ *such that*

$$\Psi_{\tau_k}(w^k + t_k\Delta w^k) \leq \Psi_{\tau_k}(w^k) - 2\sigma t_k\Psi(w^k) \tag{12}$$

and set $w^{k+1} := w^k + t_k\Delta w^k$.

(S.4) If

$$\|\Phi(w^{k+1})\| \leq \max\{\eta\beta_k, \|\Phi(w^{k+1}) - \Phi_{\tau_k}(w^{k+1})\|/\alpha\}, \tag{13}$$

then set

$$\beta_{k+1} := \|\Phi(w^{k+1})\| \tag{14}$$

and choose τ_{k+1} *such that*

$$\tau_{k+1} \in \left(0, \min\left\{\frac{\alpha}{2\kappa}\beta_{k+1}, \frac{\tau_k}{2}\right\}\right). \tag{15}$$

Otherwise (i.e., if (13) is not satisfied) set $\beta_{k+1} := \beta_k$ *and* $\tau_{k+1} := \tau_k$.

(S.5) Set $k \leftarrow k + 1$ *and go to Step (S.1).*

Step (S.0) of Algorithm 2.4 is the initialization (with κ being the constant from Lemma 2.3), while Step (S.1) contains the termination criterion. Step (S.2) computes the Newton-type search direction which is the main computational effort of Algorithm 2.4. Step (S.3) then calculates a stepsize by using an Armijo-type condition for the smoothed merit function Ψ_{τ_k}. Step (S.4) contains the updating rule for the smoothing parameter τ_k. This updating rule looks somewhat complicated, however, it is exactly the rule that is needed in the theoretical analysis. More precisely, the current updating rule for τ_k will be used in order to establish a global convergence result for Algorithm 2.4, see the next section for further details. To guarantee local fast convergence, however, one has to impose some further conditions on the choice of τ_k. Loosely speaking, τ_k has to go to zero sufficiently fast. This issue will be discussed in more detail in Section 4.

3 Global Convergence

Throughout this section, we assume that the termination parameter ε is equal to zero and that Algorithm 2.4 does not terminate in a finite number of iterations. Under this assumption, we will show that Algorithm 2.4 is well-defined and globally convergent in the sense that any accumulation point of a sequence generated by the Jacobian smoothing method is a solution of the optimality conditions (3).

Our first result states that the linear systems (11) can always be solved uniquely.

Proposition 3.1 *The Jacobian matrices $\Phi_\tau'(w)$ are nonsingular for all $w = (x, \lambda, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ and all $\tau > 0$.*

Proof. It is easy to see that Φ_τ is differentiable with

$$\Phi_\tau'(x, \lambda, s) = \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ D_{a,\tau} & 0 & D_{b,\tau} \end{pmatrix} \tag{16}$$

with diagonal matrices

$$D_{a,\tau} := \operatorname{diag}\left(\dots, \frac{\partial \varphi_\tau}{\partial a}(x_i, s_i), \dots\right) \in \mathbb{R}^{n \times n}$$

and

$$D_{b,\tau} := \operatorname{diag}\left(\dots, \frac{\partial \varphi_\tau}{\partial b}(x_i, s_i), \dots\right) \in \mathbb{R}^{n \times n}.$$

Since

$$\frac{\partial \varphi_\tau}{\partial a}(a, b) \in (0, 2) \quad \text{and} \quad \frac{\partial \varphi_\tau}{\partial b}(a, b) \in (0, 2)$$

for all $(a, b) \in \mathbb{R}^2$ and both functions φ_τ defined in (8) and (9), it follows that the diagonal matrices $D_{a,\tau}$ and $D_{b,\tau}$ are positive definite. Now let $q = (q^{(1)}, q^{(2)}, q^{(3)}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ be an appropriately partitioned vector with $\Phi_\tau'(w)q = 0$. Then (16) implies

$$A^T q^{(2)} + q^{(3)} = 0, \tag{17}$$

$$A q^{(1)} = 0, \tag{18}$$

$$D_{a,\tau} q^{(1)} + D_{b,\tau} q^{(3)} = 0. \tag{19}$$

Premultiplying (17) by $(q^{(1)})^T$ and taking into account (18) gives

$$(q^{(1)})^T q^{(3)} = 0. \tag{20}$$

On the other hand, solving (19) for $q^{(1)}$ and substituting into (20) results into

$$(q^{(3)})^T D_{a,\tau}^{-1} D_{b,\tau} q^{(3)} = 0. \tag{21}$$

Since $D_{a,\tau}, D_{b,\tau}$ are positive definite, we obtain $q^{(3)} = 0$ from (21). This implies $q^{(1)} = 0$ because of (19). Hence we also get $q^{(2)} = 0$ from (17) and the full rank assumption of A. □

We next state a technical inequality which, however, will turn out to be very helpful in our subsequent analysis. Its simple proof can be found in [5] or [12].

Lemma 3.2 *Let $\{w^k\}$ be a sequence generated by Algorithm 2.4. Then*

$$\|\Phi(w^k) - \Phi_{\tau_k}(w^k)\| \leq \alpha \|\Phi(w^k)\|$$

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

The following result guarantees that the line search in Step (S.3) of Algorithm 2.4 is well-defined. Its proof can be found in [5, 12] and is based on Lemma 3.2.

Proposition 3.3 *At each iteration k, there exists a finite exponent ℓ_k such that the stepsize $t_k = \rho^{\ell_k}$ satisfies the line search criterion (12).*

Propositions 3.1 and 3.3 together imply that Algorithm 2.4 is at least well-defined. In the remaining part of this section, we will show that it is also globally convergent in the sense that any accumulation point of a sequence generated by Algorithm 2.4 is a solution of the optimality conditions (3). To this end, it will be convenient to use the index set

$$K := \{0\} \cup \{k \in \mathbb{N} \mid \|\Phi(w^k)\| \leq \max\{\eta\beta_{k-1}, \|\Phi_{\tau_{k-1}}(w^k) - \Phi(w^k)\|/\alpha\}. \tag{22}$$

Our next result says that, although $\|\Phi(w^k)\|$ does not necessarily decrease monotonically (since our line search is based on $\|\Phi_{\tau_k}(\cdot)\|$), a possible increase cannot be too dramatic.

Proposition 3.4 *All iterates w^k generated by Algorithm 2.4 belong to the level set*

$$\mathcal{L} := \{w \mid \|\Phi(w)\| \leq (1+\alpha)\|\Phi(w^0)\|\}.$$

Proof. The proof is essentially the same as the ones given in [5, 12] for the corresponding results in that papers. Nevertheless, we include it here not only for the sake of completeness, but also since we will derive an important inequality which will be used in our subsequent analysis.

Let us partition the index set K from (22) into $K = \{0\} \cup K_1 \cup K_2$, where

$$K_1 := \{k \in K \mid \eta\beta_{k-1} \geq \|\Phi_{\tau_{k-1}}(w^k) - \Phi(w^k)\|/\alpha\}$$

and

$$K_2 := \{k \in K \mid \eta\beta_{k-1} < \|\Phi_{\tau_{k-1}}(w^k) - \Phi(w^k)\|/\alpha\}.$$

Assume that K consists of $k_0 = 0 < k_1 < k_2 < \dots$ (note that K might be finite or infinite). Let $k \in \mathbb{N}$ be arbitrarily given and k_j be the largest number in K such that $k_j \le k$. Then we have

$$\tau_k = \tau_{k_j} \quad \text{and} \quad \beta_k = \beta_{k_j} = \|\Phi(w^{k_j})\|$$

in view of our updating rules in Step (S.4) of Algorithm 2.4. Using the line search criterion (12), we have

$$\|\Phi_{\tau_{k_j}}(w^k)\| \le \|\Phi_{\tau_{k_j}}(w^{k_j})\|.$$

Hence we obtain from Lemma 2.3

$$\begin{aligned} \|\Phi(w^k)\| &\le \|\Phi_{\tau_k}(w^k)\| + \|\Phi(w^k) - \Phi_{\tau_k}(w^k)\| \\ &= \|\Phi_{\tau_{k_j}}(w^k)\| + \|\Phi(w^k) - \Phi_{\tau_{k_j}}(w^k)\| \\ &\le \|\Phi_{\tau_{k_j}}(w^k)\| + \kappa\tau_{k_j} \\ &\le \|\Phi_{\tau_{k_j}}(w^{k_j})\| + \kappa\tau_{k_j} \\ &\le \|\Phi(w^{k_j})\| + \|\Phi_{\tau_{k_j}}(w^{k_j}) - \Phi(w^{k_j})\| + \kappa\tau_{k_j} \\ &\le \|\Phi(w^{k_j})\| + \kappa\tau_{k_j} + \kappa\tau_{k_j} \\ &= \beta_{k_j} + 2\kappa\tau_{k_j}. \end{aligned} \tag{23}$$

Now, if $j = 0$, we have $\beta_{k_j} = \beta_0, \tau_{k_j} = \tau_0$ and therefore

$$\|\Phi(w^k)\| \le \beta_0 + 2\kappa\tau_0 = (1+\alpha)\|\Phi(w^0)\|$$

from (23) and Step (S.0) of Algorithm 2.4. On the other hand, if $j \ge 1$, we get from Step (S.4) of Algorithm 2.4 that

$$\tau_{k_j} \le \frac{1}{2}\tau_{k_j-1} = \frac{1}{2}\tau_{k_{j-1}}$$

and either

$$\beta_{k_j} \le \eta\beta_{k_j-1} = \eta\beta_{k_{j-1}} \qquad \text{if } k_j \in K_1$$

or, using Lemma 2.3 again,

$$\beta_{k_j} \le \|\Phi_{\tau_{k_j-1}}(w^{k_j}) - \Phi(w^{k_j})\|/\alpha \le \frac{\kappa}{\alpha}\tau_{k_j-1} = \frac{\kappa}{\alpha}\tau_{k_{j-1}} \le \frac{1}{2}\beta_{k_{j-1}} \qquad \text{if } k_j \in K_2.$$

Let us define

$$r := \max\{1/2, \eta\}.$$

Then it follows from the definitions of τ_0 and β_0 that, for $j \ge 1$, we have

$$\tau_{k_j} \le \frac{1}{2^{j-1}}\tau_0 = \frac{1}{2^j}\frac{\alpha}{\kappa}\|\Phi(w^0)\| \tag{24}$$

and

$$\beta_{k_j} \le r^{j-1}\beta_0 = r^{j-1}\|\Phi(w^0)\|. \tag{25}$$

Therefore, using (23) and $r \geq 1/2$, we obtain for $j \geq 1$

$$\|\Phi(w^k)\| \leq \left(r^{j-1} + \frac{\alpha}{2^{j-1}}\right) \|\Phi(w^0)\| \leq r^{j-1}(1+\alpha)\|\Phi(w^0)\|. \tag{26}$$

This shows that the inequality

$$\|\Phi(w^k)\| \leq (1+\alpha)\|\Phi(w^0)\|$$

holds in any case. □

We are now in the position to prove our main global convergence result.

Theorem 3.5 *Every accumulation point of a sequence $\{w^k\}$ generated by Algorithm 2.4 is a solution of the optimality conditions (3).*

Proof. If the index set K is infinite, then the statement follows immediately from (26). So consider the case where K is finite. Let $\hat{k}$ be the largest number in K. Then it follows from our updating rules in Step (S.4) of Algorithm 2.4 that the following relations hold for all $k \geq \hat{k}$:

$$\begin{aligned}
\tau_k &= \tau_{\hat{k}}, && (27)\\
\beta_k &= \beta_{\hat{k}} = \|\Phi(w^{\hat{k}})\|, && (28)\\
\|\Phi(w^k)\| &> \eta\beta_k = \eta\|\Phi(w^{\hat{k}})\| > 0, && (29)\\
\alpha\|\Phi(w^k)\| &> \eta\|\Phi_{\tau_{\hat{k}}}(w^k) - \Phi(w^k)\|. && (30)
\end{aligned}$$

Let w^* be an accumulation point of $\{w^k\}$ and $\{w^k\}_L$ be a subsequence converging to w^*. Without loss of generality, we assume that $k \geq \hat{k}$ for all $k \in L$. Let

$$t_* := \liminf_{k \in L} t_k$$

(note that we take the limes inferior on the subset L only). We now distinguish two cases.

Case 1: $t_* > 0$.
Since $t_k \geq t_*/2$ for all $k \in L$ sufficiently large, we then get from our line search rule that

$$\Psi_{\tau_{\hat{k}}}(w^{k+1}) - \Psi_{\tau_{\hat{k}}}(w^k) \leq -2\sigma t_k \Psi(w^k) \leq -\sigma t_* \Psi(w^k) < 0. \tag{31}$$

On the other hand (since $\tau_{\hat{k}}$ is fixed), the merit function $\Psi_{\tau_{\hat{k}}}$ is bounded from below (by zero) and $\{\Psi_{\tau_{\hat{k}}}(w^k)\}_{k \in \mathbb{N}}$ is a monotonically decreasing sequence. Hence $\{\Psi_{\tau_{\hat{k}}}(w^k)\}$ is convergent. In particular, we therefore have

$$\Psi_{\tau_{\hat{k}}}(w^{k+1}) - \Psi_{\tau_{\hat{k}}}(w^k) \to 0 \quad \text{for } k \to \infty.$$

Hence, taking the limit $k \to \infty$ for $k \in L$ in (31), we get

$$0 \leq -\sigma t_* \Psi(w^*) \leq 0$$

and therefore $\Psi(w^*) = 0$.

Case 2: $t_* = 0$.
Since $\tau_k = \tau_{\hat{k}} > 0$ for all $k \in L$, $\Phi'_{\tau_{\hat{k}}}(w^*)$ is nonsingular by Proposition 3.1, and $\{w^k\}_{k\in L} \to w^*$, it follows from a standard perturbation result that $\Phi'_{\tau_k}(w^k)$ is nonsingular with

$$\|\Phi'_{\tau_k}(w^k)^{-1}\| \le \kappa_1$$

for all $k \in L$ sufficiently large and a suitable constant $\kappa_1 > 0$. This implies

$$\begin{aligned}
\|\Delta w^k\| &\le \|\Phi'_{\tau_k}(w^k)^{-1}\Phi(w^k)\| \\
&\le \kappa_1\|\Phi(w^k)\| \\
&\le \kappa_1(1+\alpha)\|\Phi(w^0)\| \\
&=: \kappa_2
\end{aligned}$$

for all large enough $k \in L$ by Step (S.2) of Algorithm 2.4 and Proposition 3.4. Hence we can assume without loss of generality that

$$\{\Delta w^k\}_{k\in L} \to \Delta w^*$$

for some vector $\Delta w^* \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$. On the other hand, subsequencing if necessary, we have $\{t_k\}_{k\in L} \to 0$. Therefore the stepsize t_k/ρ does not satisfy the Armijo-like condition (12), i.e., we have

$$-2\sigma\frac{t_k}{\rho}\Psi(w^k) < \Psi_{\tau_{\hat{k}}}(w^k + \frac{t_k}{\rho}\Delta w^k) - \Psi_{\tau_{\hat{k}}}(w^k)$$

for all $k \in L$ sufficiently large. Using the Cauchy-Schwarz inequality and Lemma 3.2, this implies

$$\begin{aligned}
&-2\sigma\Psi(w^k) \\
<\;& \frac{\Psi_{\tau_{\hat{k}}}(w^k + \frac{t_k}{\rho}\Delta w^k) - \Psi_{\tau_{\hat{k}}}(w^k)}{\frac{t_k}{\rho}} \\
=\;& \nabla\Psi_{\tau_{\hat{k}}}(w^k)^T\Delta w^k + \left(\frac{\Psi_{\tau_{\hat{k}}}(w^k + \frac{t_k}{\rho}\Delta w^k) - \Psi_{\tau_{\hat{k}}}(w^k)}{\frac{t_k}{\rho}} - \nabla\Psi_{\tau_{\hat{k}}}(w^k)^T\Delta w^k\right) \\
=\;& -\Phi_{\tau_{\hat{k}}}(w^k)^T\Phi(w^k) + \left(\frac{\Psi_{\tau_{\hat{k}}}(w^k + \frac{t_k}{\rho}\Delta w^k) - \Psi_{\tau_{\hat{k}}}(w^k)}{\frac{t_k}{\rho}} - \nabla\Psi_{\tau_{\hat{k}}}(w^k)^T\Delta w^k\right) \\
=\;& -2\Psi(w^k) + \Phi(w^k)^T\left(\Phi(w^k) - \Phi_{\tau_{\hat{k}}}(w^k)\right) \\
&+ \left(\frac{\Psi_{\tau_{\hat{k}}}(w^k + \frac{t_k}{\rho}\Delta w^k) - \Psi_{\tau_{\hat{k}}}(w^k)}{\frac{t_k}{\rho}} - \nabla\Psi_{\tau_{\hat{k}}}(w^k)^T\Delta w^k\right) \\
\le\;& -2\Psi(w^k) + 2\alpha\Psi(w^k) \\
&+ \left(\frac{\Psi_{\tau_{\hat{k}}}(w^k + \frac{t_k}{\rho}\Delta w^k) - \Psi_{\tau_{\hat{k}}}(w^k)}{\frac{t_k}{\rho}} - \nabla\Psi_{\tau_{\hat{k}}}(w^k)^T\Delta w^k\right).
\end{aligned}$$

Taking the limit $k \to \infty$ for $k \in L$ and using $w^k \to w^*, \Delta w^k \to \Delta w^*$, and $t_k \to 0$ on this subsequence, we obtain from the continuous differentiability of $\Psi_{\tau_{\hat{k}}}$ that

$$-2\sigma\Psi(w^*) \leq -2\Psi(w^*) + 2\alpha\Psi(w^*) + \nabla\Psi_{\tau_{\hat{k}}}(w^*)^T\Delta w^* - \nabla\Psi_{\tau_{\hat{k}}}(w^*)^T\Delta w^*$$

and therefore

$$\sigma\Psi(w^*) \geq (1-\alpha)\Psi(w^*).$$

By our choice of σ in Step (S.0) of Algorithm 2.4, this implies $\Psi(w^*) = 0$.

We therefore obtain in both cases that $\Psi(w^*) = 0$ and therefore also $\Phi(w^*) = 0$. However, by (29) and continuity, we have $\|\Phi(w^*)\| > 0$. This contradiction shows that K is infinite, so the proof is complete. □

Note that the previous proof showed the following: Whenever the sequence $\{w^k\}$ generated by Algorithm 2.4 has an accumulation point, then the index set K is infinite. Hence K can be finite only if $\{w^k\}$ is unbounded.

4 Rate of Convergence

The aim of this section is to prove local quadratic convergence of our Jacobian smoothing method from Algorithm 2.4 under suitable assumptions. To this end, we first establish the following result which is also of interest by its own.

Theorem 4.1 *Let (x^*, λ^*, s^*) be a solution of the optimality conditions (3). Then the following statements are equivalent:*

(a) Φ is continuously differentiable at (x^*, λ^*, s^*), and the Jacobian $\Phi'(x^*, \lambda^*, s^*)$ is nonsingular.

(b) (x^*, λ^*, s^*) is the unique solution of the optimality conditions (3).

Proof. (a) $\Longrightarrow$ (b): Let Φ be continuously differentiable at (x^*, λ^*, s^*) with $\Phi'(x^*, \lambda^*, s^*)$ being nonsingular. Then, by standard results [6], there exists a constant $c > 0$ with

$$\|\Phi(x, \lambda, s)\| \geq c\|(x^*, \lambda^*, s^*) - (x, \lambda, s)\|$$

for all (x, λ, s) sufficiently close to (x^*, λ^*, s^*). This inequality shows that, locally, (x^*, λ^*, s^*) is the unique solution of the optimality conditions (3). However, since it is easy to see that the solution set of (3) is convex, it follows that (x^*, λ^*, s^*) is the unique solution of (3) also from a global point of view.

(b) $\Longrightarrow$ (a): By the Goldman-Tucker Theorem (see, e.g., [18]), the unique solution (x^*, λ^*, s^*) of the optimality conditions (3) satisfies the strict complementarity condition

$$x_i^* + s_i^* > 0 \quad \forall i = 1, \dots, n.$$

This implies that Φ is continuously differentiable at (x^*, λ^*, s^*).

In order to see that the Jacobian of Φ is nonsingular at the point (x^*, λ^*, s^*), we assume throughout this proof that φ denotes the Fischer-Burmeister function from (7). The proof for the minimum function from (6) is very similar and therefore omitted here.

Define the two index sets

$$\begin{aligned}\mathcal{B} &:= \{i \in \{1, \dots, n\} \mid x_i^* > 0\}, \\ \mathcal{N} &:= \{i \in \{1, \dots, n\} \mid s_i^* > 0\}.\end{aligned}$$

By strict complementarity, we have

$$\mathcal{N} = \{1, \dots, n\} \setminus \mathcal{B}.$$

Note that

$$\Phi'(x^*, \lambda^*, s^*) = \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ D_a & 0 & D_b \end{pmatrix}$$

with

$$\begin{aligned}D_a &:= \operatorname{diag}\left(\dots, \frac{\partial \varphi}{\partial a}(x_i^*, s_i^*), \dots\right), \\ D_b &:= \operatorname{diag}\left(\dots, \frac{\partial \varphi}{\partial b}(x_i^*, s_i^*), \dots\right).\end{aligned}$$

The definitions of the index sets $\mathcal{B}$ and $\mathcal{N}$ together with the definition of the Fischer-Burmeister function shows that

$$\frac{\partial \varphi}{\partial a}(x_i^*, s_i^*) = \frac{x_i^*}{\sqrt{(x_i^*)^2 + (s_i^*)^2}} - 1 = \begin{cases} 0 & \text{if } i \in \mathcal{B}, \\ -1 & \text{if } i \in \mathcal{N}, \end{cases} \tag{32}$$

$$\frac{\partial \varphi}{\partial b}(x_i^*, s_i^*) = \frac{s_i^*}{\sqrt{(x_i^*)^2 + (s_i^*)^2}} - 1 = \begin{cases} -1 & \text{if } i \in \mathcal{B}, \\ 0 & \text{if } i \in \mathcal{N}. \end{cases} \tag{33}$$

Then $\Phi'(x^*, \lambda^*, s^*)q = 0$ for an appropriately partitioned vector $q = (q^{(1)}, q^{(2)}, q^{(3)})$ implies

$$A^T q^{(2)} + q^{(3)} = 0, \tag{34}$$

$$A q^{(1)} = 0, \tag{35}$$

$$(D_a)_{\mathcal{B}} q_{\mathcal{B}}^{(1)} + (D_b)_{\mathcal{B}} q_{\mathcal{B}}^{(3)} = 0, \tag{36}$$

$$(D_a)_{\mathcal{N}} q_{\mathcal{N}}^{(1)} + (D_b)_{\mathcal{N}} q_{\mathcal{N}}^{(3)} = 0, \tag{37}$$

where $q_{\mathcal{B}}^{(1)}$ denotes the $|\mathcal{B}|$-dimensional subvector of $q^{(1)}$ consisting of the components $q_i^{(1)}$ $(i \in \mathcal{B})$; similarly, $(D_a)_{\mathcal{B}}$ denotes the $|\mathcal{B}| \times |\mathcal{B}|$-diagonal matrix containing the diagonal entries a_{ii} $(i \in \mathcal{B})$ from the matrix D_a. The other subvectors and submatrices occuring in the above formulars are defined in an analogous way.

Using (32), (33), (36), and (37), we obtain

$$q_{\mathcal{B}}^{(3)} = 0 \quad \text{and} \quad q_{\mathcal{N}}^{(1)} = 0. \tag{38}$$

We will use (38) in order to show that the vector

$$(x^*(t), \lambda^*(t), s^*(t)) := (x^*, \lambda^*, s^*) + t(q^{(1)}, q^{(2)}, q^{(3)})$$

is also a solution of the optimality conditions (3) for all $t > 0$ sufficiently small. This then implies that $q = (q^{(1)}, q^{(2)}, q^{(3)}) = (0, 0, 0)$ since (x^*, λ^*, s^*) was assumed to be the only solution of (3).

Obviously, the equations $Ax^*(t) = b$ and $A^T\lambda^*(t) + s^*(t) = c$ are satisfied for any $t > 0$ in view of (34) and (35), respectively. Moreover, $x^*(t) \geq 0$ and $s^*(t) \geq 0$ for all $t > 0$ sufficiently small follows from the definition of the index sets $\mathcal{B}$ and $\mathcal{N}$ together with (38). Finally, we also have

$$x^*(t)^T s^*(t) = \left(x^*_{\mathcal{B}} + tq^{(1)}_{\mathcal{B}}\right)^T \left(s^*_{\mathcal{B}} + tq^{(3)}_{\mathcal{B}}\right) + \left(x^*_{\mathcal{N}} + tq^{(1)}_{\mathcal{N}}\right)^T \left(s^*_{\mathcal{N}} + tq^{(3)}_{\mathcal{N}}\right) = 0$$

since (x^*, λ^*, s^*) satisfies the optimality conditions (3), $x^*_{\mathcal{N}} = 0, s^*_{\mathcal{B}} = 0$ and because of (38). This completes the proof. □

A result similar to Theorem 4.1 was also presented by Burke and Xu [2] on a recent conference; in fact, it was pointed out by Burke that Theorem 4.1 can be derived from a similar result by Burke and Xu [3], where the authors consider a smoothing-type method for linear complementarity problems based on the minimum function.

Theorem 4.1 allows the following interpretation: Assume that a sequence $\{(x^k, \lambda^k, s^k)\}$ generated by Algorithm 2.4 converges to a solution (x^*, λ^*, s^*) satisfying the strict complementarity condition $x^*_i + s^*_i > 0$ for all $i = 1, \ldots, n$, so that Φ is continuously differentiable around this solution point. Then Theorem 4.1 states that the sequence of Jacobian matrices $\{\Phi'(x^k, \lambda^k, s^k)\}$ converges to a singular matrix whenever (x^*, λ^*, s^*) is not the unique solution of (3).

In the remaining part of this section, we want to show that Algorithm 2.4 is locally quadratically convergent if one of the two equivalent conditions from Theorem 4.1 is satisfied and if the smoothing parameter τ_k is updated in an appropriate way. The latter is made more precise in our next two results.

In order to motivate this result, assume that Φ is continuously differentiable at a point (x, λ, s). Then $\Phi'_\tau(x, \lambda, s) \to \Phi'(x, \lambda, s)$ for $\tau \to 0$. Hence, for any $\delta > 0$, there exists a constant $\bar{\tau} > 0$ such that

$$\|\Phi'_\tau(x, \lambda, s) - \Phi'(x, \lambda, s)\| \leq \delta$$

for all $\tau \in [0, \bar{\tau}]$. However, the existence of such a constant $\bar{\tau}$ does not guarantee that this $\bar{\tau}$ can be computed easily. On the other hand, our local rate of convergence result assumes that such a constant is computable. Therefore, our next two results give explicit values of this $\bar{\tau}$ provided that the difference between the matrices $\Phi'_\tau(x, \lambda, s)$ and $\Phi'(x, \lambda, s)$ is measured in the Frobenius norm.

The first of these two results deals with the case where Φ and Φ_τ are defined via the Fischer-Burmeister function (7) and its smooth counterpart (9).

Lemma 4.2 *Let Φ and Φ_τ be defined using the Fischer-Burmeister-type functions (7) and (9), respectively. Furthermore, let $(x, \lambda, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ be any vector with $x_i^2 + s_i^2 > 0$ for all $i = 1, \dots, n$, and let $\delta > 0$ be arbitrarily given. Then we have*

$$\|\Phi'_\tau(x, \lambda, s) - \Phi'(x, \lambda, s)\|_F \le \delta$$

for all $\tau \in [0, \bar{\tau}]$, where $\bar{\tau} = \bar{\tau}(x, s, \delta) > 0$ is given by

$$\bar{\tau}(x, s, \delta) := \frac{\delta \min_{i=1,\dots,n}\{x_i^2 + s_i^2\}}{2\sqrt{n \max_{i=1,\dots,n}\{x_i^2 + s_i^2\}}}.$$

Proof. Since $x_i^2 + s_i^2 > 0$ for all $i = 1, \dots, n$, the mapping Φ is continuously differentiable at (x, λ, s). Hence its Jacobian $\Phi'(x, \lambda, s)$ exists at (x, λ, s), and an elementary calculation shows that

$$\begin{aligned} \|\Phi'_\tau(x, \lambda, s) - \Phi'(x, \lambda, s)\|_F^2 \;=\; & \textstyle\sum_{i=1}^n \left(\frac{\partial \varphi_\tau}{\partial a}(x_i, s_i) - \frac{\partial \varphi}{\partial a}(x_i, s_i)\right)^2 + \\ & \textstyle\sum_{i=1}^n \left(\frac{\partial \varphi_\tau}{\partial b}(x_i, s_i) - \frac{\partial \varphi}{\partial b}(x_i, s_i)\right)^2. \end{aligned} \tag{39}$$

Throughout this proof, let us use the notation

$$\alpha_{xs} := \min_{i=1,\dots,n}\{x_i^2 + s_i^2\} > 0.$$

Then we obtain

$$\begin{aligned} \left|\frac{\partial \varphi_\tau}{\partial a}(x_i, s_i) - \frac{\partial \varphi}{\partial a}(x_i, s_i)\right| &= \left|\frac{x_i}{\sqrt{x_i^2 + s_i^2 + 2\tau^2}} - \frac{x_i}{\sqrt{x_i^2 + s_i^2}}\right| \\ &= |x_i| \left(\frac{1}{\sqrt{x_i^2 + s_i^2}} - \frac{1}{\sqrt{x_i^2 + s_i^2 + 2\tau^2}}\right) \\ &\le |x_i| \left(\frac{1}{\sqrt{\alpha_{xs}}} - \frac{1}{\sqrt{\alpha_{xs} + 2\tau^2}}\right) \\ &= |x_i| \frac{\sqrt{\alpha_{xs} + 2\tau^2} - \sqrt{\alpha_{xs}}}{\sqrt{\alpha_{xs}}\sqrt{\alpha_{xs} + 2\tau^2}} \\ &\le |x_i| \frac{\sqrt{2}\tau}{\alpha_{xs}} \end{aligned}$$

for $i = 1, \dots, n$, where the first inequality follows from the fact that the function

$$f(a) := \frac{1}{\sqrt{a}} - \frac{1}{\sqrt{a + 2\tau^2}}$$

is strictly decreasing for $a > 0$ (since $f'(a) < 0$ for $a > 0$), and the second inequality follows from

$$\sqrt{a + b} \le \sqrt{a} + \sqrt{b}$$

for all $a, b \geq 0$. In a similar way, we get

$$\left| \frac{\partial \varphi_\tau}{\partial b}(x_i, s_i) - \frac{\partial \varphi}{\partial b}(x_i, s_i) \right| \leq |s_i| \frac{\sqrt{2}\tau}{\alpha_{xs}}$$

for $i = 1, \ldots, n$. Using the definition of $\bar{\tau}$, we therefore obtain for any $\tau \in [0, \bar{\tau}]$ and any $i \in \{1, \ldots, n\}$:

$$\begin{aligned}
\left(\frac{\partial \varphi_\tau}{\partial a}(x_i, s_i) - \frac{\partial \varphi}{\partial a}(x_i, s_i) \right)^2 &\leq \frac{x_i^2 2\tau^2}{\alpha_{xs}^2} \\
&\leq \frac{2x_i^2 \bar{\tau}^2}{\alpha_{xs}^2} \\
&\leq \frac{2x_i^2 \delta^2 \alpha_{xs}^2}{4n\alpha_{xs}^2 \max_{i=1,\ldots,n}\{x_i^2 + s_i^2\}} \\
&\leq \frac{\delta^2}{2n}
\end{aligned}$$

and, similarly,

$$\left(\frac{\partial \varphi_\tau}{\partial b}(x_i, s_i) - \frac{\partial \varphi}{\partial b}(x_i, s_i) \right)^2 \leq \frac{\delta^2}{2n}.$$

Using (39), this implies

$$\begin{aligned}
& \|\Phi_\tau'(x, \lambda, s) - \Phi'(x, \lambda, s)\|_F \\
= & \sqrt{\sum_{i=1}^n \left(\frac{\partial \varphi_\tau}{\partial a}(x_i, s_i) - \frac{\partial \varphi}{\partial a}(x_i, s_i) \right)^2 + \sum_{i=1}^n \left(\frac{\partial \varphi_\tau}{\partial b}(x_i, s_i) - \frac{\partial \varphi}{\partial b}(x_i, s_i) \right)^2} \\
\leq & \sqrt{\frac{n\delta^2}{2n} + \frac{n\delta^2}{2n}} \\
= & \delta.
\end{aligned}$$

This completes the proof. □

The next result deals with the case where Φ and Φ_τ are defined by the minimum function (6) and its smooth counterpart (8).

Lemma 4.3 *Let Φ and Φ_τ be defined using the minimum-type functions (6) and (8), respectively. Furthermore, let $(x, \lambda, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ be any vector with $x_i \neq s_i$ for all $i = 1, \ldots, n$, and let $\delta > 0$ be arbitrarily given. Then we have*

$$\|\Phi_\tau'(x, \lambda, s) - \Phi'(x, \lambda, s)\|_F \leq \delta$$

for all $\tau \in [0, \bar{\tau}]$, where $\bar{\tau} = \bar{\tau}(x, s, \delta) > 0$ is given by

$$\bar{\tau}(x, s, \delta) := \frac{\delta \min_{i=1,\ldots,n}\{(x_i - s_i)^2\}}{2\sqrt{2n \max_{i=1,\ldots,n}\{(x_i - s_i)^2\}}}.$$

Proof. The proof is essentially the same as the one given for Lemma 4.2. In fact, if we define

$$\alpha_{xs} := \min_{i=1,\dots,n} \{(x_i - s_i)^2\} > 0,$$

then the previous proof goes through with some minor modifications. $\square$

Lemmas 4.2 and 4.3 enable us to state the following local convergence result for Algorithm 2.4 if the smoothing parameter τ_k gets updated sufficiently fast in Step (S.4) of Algorithm 2.4.

Theorem 4.4 *Let (x^*, λ^*, s^*) be the unique solution of the optimality conditions (3), and assume that (x^*, λ^*, s^*) is an accumulation point of a sequence $\{(x^k, \lambda^k, s^k)\}$ generated by Algorithm 2.4. Then the entire sequence $\{(x^k, \lambda^k, s^k)\}$ converges to (x^*, λ^*, s^*). Moreover, if τ_k gets updated such that*

$$\tau_{k+1} \in \left(0, \min\left\{\frac{\alpha}{2\kappa}\beta_{k+1}, \frac{\tau_k}{2}, \bar{\tau}_k\right\}\right)$$

for all k sufficiently large, where $\bar{\tau}_k := \bar{\tau}(x^{k+1}, s^{k+1}, \beta_{k+1})$ denotes the constant defined in Lemma 4.2 or Lemma 4.3 (depending on whether we choose the Fischer-Burmeister or the minimum function), then $\{(x^k, \lambda^k, s^k)\}$ converges quadratically to (x^*, λ^*, s^*).

Proof. First note that, since the solution (x^*, λ^*, s^*) of the optimality conditions (3) satisfies strict complementarity in view of our assumptions, it follows that the two conditions $x_i^2 + s_i^2 > 0$ and $x_i \neq s_i$ for all $i = 1, \dots, n$ used in Lemmas 4.2 and 4.3, respectively, are satisfied in a sufficiently small neighbourhood of (x^*, λ^*, s^*). Hence we can apply these two results in our situation. Doing this and taking into account Theorem 4.1, it is possible to verify the statement in exactly the same way as this is done in [5, 12] for the methods presented there. We therefore skip the proof here. $\square$

There is a recent result by Tseng [17] which indicates that the search direction used in our Jacobian smoothing method is a good one and, in fact, provides a superlinear rate of convergence even if the solution set of the optimality conditions (3) is not a singleton. Unfortunately, however, it is currently not possible to apply the result by Tseng [17] to our framework.

5 Numerical Results

When implementing the Jacobian smoothing method from Algorithm 2.4, it is quite helpful to observe that the linear systems we have to solve at each iteration have exactly the same structure as those in primal-dual interior-point methods, see also the discussion in the next section. Hence it is possible to take the linear algebra subroutines from existing interior-point codes. In particular, for our preliminary testing, we took the LIPSOL solver by Zhang [20, 21]. This is a MATLAB program which, however, calls a FORTRAN sparse Cholesky code in order to solve the linear systems of equations at each iteration (note that it is not necessary to solve the linear systems (11) directly; instead, one can easily use the special structure of this system in order to see that one has to solve only a positive definite system of dimension m at each iteration). Obviously, we had to change the main program in LIPSOL completely in order to implement the Jacobian smoothing method. The implementation of Algorithm 2.4 uses the following parameters:

$$\alpha = 0.99995, \quad \eta = 0.31, \quad \rho = 0.9, \quad \sigma = 10^{-4},$$

and the definition of Φ and Φ_τ are based on the minimum function (6) and its smooth counterpart (8), respectively. The termination criterion used in our code is

$$\|\Phi(w^k)\| \leq \varepsilon \quad \text{with} \quad \varepsilon = 10^{-3}.$$

This is a much weaker condition than what is typically used in corresponding complementarity software. However, due to possible singularity problems (cf. the discussion in Section 4), it seems that one should not use a too strong termination criterion. Moreover, according to our experience, the approximate solutions found by using the above stopping rule seem to have the same accuracy as those provided by interior-point solvers. Nevertheless, we stress that a suitable stopping criterion for the Jacobian smoothing method is a nontrivial task since the iterates are no longer guaranteed to be feasible.

On the other hand, since we choose the starting point (x^0, λ^0, s^0) in such a way that at least the linear equations

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

are satisfied at $(x, \lambda, s) = (x^0, \lambda^0, s^0)$, it follows that these linear equations are satisfied at all iterates $(x, \lambda, s) = (x^k, \lambda^k, s^k)$. Consequently, the only infeasibility which can occur is in the complementarity conditions

$$x_i \geq 0, \; s_i \geq 0, \; x_i s_i = 0 \quad (i = 1, \ldots, n).$$

The precise way we choose our starting point is as follows:

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

(b) Set $x^0 := A^Ty^0$ (hence we have $Ax^0 = b$).

(c) Define $\lambda^0 := 0$ and $s^0 := c$ (so that we also have $A^T\lambda^0 + s^0 = c$).

This may not be the best choice for a starting point to be used within our Jacobian smoothing method, but it seems to be a reasonable and relatively simple choice.

We tested our method on all problems from the netlib collection with less than $m = 600$ rows (in the original formulation; the program automatically reformulates all problems in a certain standard form by, e.g., introducing nonnegative slack variables for all upper bound constraints, and this procedure can increase the dimension of some problems dramatically). Table 1 presents the corresponding results:

- Column 1 contains the name of the test example.
- Column 2 gives the number of iterations until termination.

Table 1: Numerical results for the Jacobian smoothing method

Problem	k	$\|\Phi(w^f)\|$	primal objective
adlittle	18	1.1090e-05	2.2549496316e+05
afiro	8	4.8102e-07	-4.6475314311e+02
agg	56	4.0555e-04	-3.5991767287e+07
agg2	33	7.7409e-04	-2.0239252356e+07
agg3	31	2.1183e-07	1.0312115935e+07
bandm	60	7.6182e-04	-1.5862797925e+02
beaconfd	31	1.8352e-06	3.3592485807e+04
blend	27	4.9101e-05	-3.0812150098e+01
boeing1	81	1.1542e-04	-3.3521356138e+02
boeing2	37	4.1229e-05	-3.1501872799e+02
bore3d	43	6.4748e-05	1.3730803943e+03
brandy	47	3.0906e-04	1.5185098971e+03
capri	43	1.2333e-04	2.6900129008e+03
degen2	27	6.0299e-05	-1.4351780000e+03
e226	69	4.7034e-04	-1.8751928767e+01
etamacro	79	5.5534e-04	-7.5571519859e+02
finnis	52	3.0823e-04	1.7279106560e+05
forplan	44	5.0031e-04	-6.6421884104e+02
israel	162	1.8494e-04	-8.9664482186e+05
lotfi	185	9.2147e-05	-2.5264706074e+01
recipe	13	9.8645e-05	-2.6661599983e+02
sc105	39	5.7189e-04	-5.2202061282e+01
sc205	82	3.4097e-04	-5.2202061208e+01

- Column 3 provides the value of $\|\Phi(w^f)\|$ at the final iterate w^f.
- Column 4 shows the value of the primal objective function at the final iterate w^f.

We think that the results in Table 1 are not too bad. Although interior-point software has a better behaviour on most of these problems, the reader should take into account that, basically, all implementations of interior-point methods for linear programs are predictor-corrector methods, whereas here we solve only one linear system of equations at each iteration. Moreover, the best choice for the parameters in Algorithm 2.4 (especially the values for α and η) are currently not clear to us, whereas interior-point methods are well-understood in the meantime. Some of these things will actually be part of our future research.

Finally, we mention that there are three more test problems with less than $m = 600$ rows which are not listed in Table 1. These are `kb2`, `pilot4`, and `seba`. On these three problems, our method failed either because the maximum number of iterations was reached (like in `kb2` and in `pilot4`) or because the stepsize t_k became too small (like in `seba`). We stress, however, that all problems can be solved by different parameter settings.

Table 1 (continued): Numerical results for the Jacobian smoothing method

Problem	k	$\|\Phi(w^f)\|$	primal objective
sc50a	20	8.3305e-05	-6.4575077292e+01
sc50b	27	9.4160e-09	-7.0000000000e+01
scagr25	99	2.5476e-04	-1.4753433061e+07
scagr7	37	9.5727e-07	-2.3313898243e+06
scfxm1	50	4.6251e-04	1.8416759030e+04
scorpion	87	7.8545e-04	1.8781248227e+03
scrs8	139	2.4730e-04	9.0429695385e+02
scsd1	10	3.4190e-04	8.6666666747e+00
scsd6	13	4.2885e-04	5.0500000078e+01
scsd8	20	9.4377e-04	9.0500000003e+02
sctap1	39	3.9728e-04	1.4122500002e+03
share1b	162	1.2334e-04	-7.6589318579e+04
share2b	34	2.7101e-05	-4.1573224073e+02
shell	48	9.7677e-07	1.2088253460e+09
ship04l	47	6.3126e-07	1.7933245380e+06
ship04s	43	2.6374e-06	1.7987147004e+06
stair	92	2.0571e-04	-2.5126695120e+02
standata	12	1.7527e-04	1.2576994995e+03
standgub	23	3.5092e-06	1.2576995000e+03
standmps	36	2.4049e-04	1.4060175000e+03
stocfor1	51	8.2467e-10	-4.1131976219e+04
tuff	40	5.8251e-04	2.9549410844e-01
vtpbase	38	1.1010e-05	1.2983146246e+05
wood1p	16	6.7710e-05	1.4429024111e+00

6 Comparison with Interior-Point Methods

Our Jacobian smoothing method turns out to be closely related to interior-point methods (primal-dual path-following methods, to be more precise). To see this, we recall that interior-point methods typically perturb the complementarity conditions within the optimality conditions (3) in order to deal with a system of the form

$$\begin{array}{rcl} A^T\lambda + s & = & c, \\ Ax & = & b, \\ x_i > 0,\ s_i > 0,\ x_i s_i & = & \tau^2 \quad \forall i = 1, \dots, n \end{array} \tag{40}$$

(here we use τ^2 instead of τ just for technical reasons). Interior-point methods then apply some kind of Newton method to the *equations* within these perturbed optimality conditions and deal with the nonnegativity of the x- and s-variables separately by a suitable line search. By reducing τ in an appropriate way, interior-point methods have a strong theoretical background and an outstanding numerical performance, see, e.g., the excellent book [18] by Wright for further details.

The relation to our Jacobian smoothing method comes from an observation made in [10]: The perturbed optimality conditions (40) can be rewritten as

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

where φ_τ still denotes one of the two smoothing functions defined in (8) and (9). Hence the system (40) is completely equivalent to the nonlinear system of equations

$$\Phi_\tau(x, \lambda, s) = 0 \tag{41}$$

which does not contain any nonnegativity constraints (at least not explicitly) and which is the basis of our Algorithm 2.4.

As we already said in the abstract, both the theoretical background and the numerical performance of interior-point methods are currently stronger than for our Jacobian smoothing method. However, interior-point methods were not born in just one day, and we are in the starting phase of our research. In fact, we believe that Jacobian smoothing methods have some definite advantages if compared with interior-point methods which should turn out to be helpful in the near future. In particular, we think that the following points are worth being mentioned:

- The Newton-type search direction computed by using the system of equations (41) includes explicitly the information that the x- and s-variables should stay nonnegative, whereas the Newton-type direction computed by interior-point methods completely disregard this point. We therefore believe that our search direction is actually the better direction, at least from a local point of view. In fact, in view of our preliminary numerical experience, the Jacobian smoothing method seems to converge locally somewhat faster than interior-point methods.

- The system (41) is an *unconstrained* reformulation of the perturbed optimality conditions (40). Hence we can allow negative components in our iterates. In particular, there is no further restriction on the length of the stepsize, in contrast to interior-point methods. Therefore, we hope to accept larger steps than those used in interior-point methods. However, our feeling is that this point has not been exploited completely by our Jacobian smoothing method, and we plan to elaborate on this as part of our future research.

- Since, as mentioned in our previous point, our iterates are not required to belong to the positive orthant, it is relatively easy to combine Jacobian smoothing methods with an active set strategy. This cannot be done in an easy way by interior-point methods since, basically, any active set method will project at least some components on the boundary of the positive orthant. If this procedure does not give the solution of the linear program, interior-point methods cannot do much with the information provided by such a strategy, whereas Jacobian smoothing methods can start easily from this projected point (even if the projected point is not a solution of the linear program, it might be much closer to a solution and therefore be an attractive point).

- Finally, if one solves a sequence of similar linear programs (like in branch-and-bound techniques for the solution of integer or mixed integer programs), one typically wants to use the solution of the previous problem as a starting point for the next one. This, however, is usually not possible for interior-point methods because the solution of the previous problem does, in general, not belong to the positive orthant of the next problem, whereas we can easily deal with this situation since we can start at an arbitrary point. In fact, this might even be an advantage if compared with simplex-type schemes. We will certainly work on this topic as part of our future research.

Acknowledgement. The authors would like to thank Jim Burke and Paul Tseng for some discussions on the topic of this paper, and for pointing out some relevant references.

References

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

[2] 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.

[3] J.V. Burke and S. Xu: *A non-interior-predictor-corrector path following algorithm for the monotone linear complementarity problem.* Mathematical Programming, to appear.

[4] 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.

[5] 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.

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

[7] M.C. Ferris and C. Kanzow: *Complementarity and related problems.* In: P.M. Pardalos and M.G.C. Resende (eds.): *Handbook on Applied Optimization.* Oxford University Press, to appear.

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

[9] A. Fischer: *A Newton-type method for positive semidefinite linear complementarity problems.* Journal of Optimization Theory and Applications 86, 1995, pp. 585–608.

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

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

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

[13] B. Kummer: *Newton's method for nondifferentiable functions.* In: J. Guddat et al. (eds.): *Advances in Mathematical Optimization.* Akademie–Verlag, Berlin, Germany, 1988, pp. 114–125.

[14] L. Qi: *Convergence analysis of some algorithms for solving nonsmooth equations.* Mathematics of Operations Research 18, 1993, pp. 227–244.

[15] L. Qi and J. Sun: *A nonsmooth version of Newton's method.* Mathematical Programming 58, 1993, pp. 353–367.

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

[17] 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.

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

[19] N. Yamashita and M. Fukushima: *Modified Newton methods for solving a semismooth reformulation of monotone complementarity problems.* Mathematical Programming 76, 1997, pp. 469–491.

[20] Y. Zhang: *User's guide to LIPSOL.* Technical Report 95-19, Department of Mathematics and Statistics, University of Baltimore County, Baltimore, MD, 1995.

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