Hamburger Beiträge
zur Angewandten Mathematik

Predictor-Corrector Smoothing Methods for Linear Programs with a More Flexible Update of the Smoothing Parameter

Stephan Engelke and Christian Kanzow

Reihe A
Preprint 162
February 2001

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 LINEAR PROGRAMS WITH A MORE FLEXIBLE UPDATE OF THE SMOOTHING PARAMETER[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

February 28, 2001

Abstract. We consider a smoothing-type method for the solution of linear programs. Its main idea is to reformulate the corresponding central path conditions as a nonlinear system of equations, to which a variant of Newton's method is applied. The method is shown to be globally and locally quadratically convergent under suitable assumptions. In contrast to a number of recently proposed smoothing-type methods, the current work allows a more flexible updating of the smoothing parameter. Furthermore, compared with previous smoothing-type methods, the current implementation of the new method gives significantly better numerical results on the netlib test suite.

Key Words. Linear programs, central path, smoothing method, global convergence, quadratic convergence.

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

1 Introduction

Consider the linear program

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

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$ are the given data and A is assumed to be of full rank, i.e., $\operatorname{rank}(A) = m$. The classical method for the solution of this minimization problem is Dantzig's simplex algorithm, see, e.g., [11, 1]. During the last two decades, however, interior-point methods have become quite popular and are now viewed as being serious alternatives to the simplex method, especially for large-scale problems.

More recently, so-called smoothing-type methods have also been investigated for the solution of linear programs. These smoothing-type methods join some of the properties of interior-point methods. To explain this in more detail, consider the optimality conditions

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

of the linear program (1), and recall that (1) has a solution if and only if (2) has a solution. The most successful interior-point methods try to solve the optimality conditions (2) by solving (inexactly) a sequence of perturbed problems (also called the *central path conditions*)

$$\begin{array}{rcl} A^T\lambda + s &=& c, \\ Ax &=& b, \\ x_i > 0,\, s_i > 0,\, x_i s_i &=& \tau^2 \quad \forall i = 1, \dots, n, \end{array} \tag{3}$$

where $\tau > 0$ denotes a suitable parameter. Typically, interior-point methods apply some kind of Newton method to the equations within these perturbed optimality conditions and guarantee the positivity of the primal and dual variables by an appropriate line search.

Many smoothing-type methods follow a similar pattern: They also try to solve (inexactly) a sequence of perturbed problems (3). To this end, however, they first reformulate the system (3) as a nonlinear system of equations and then apply Newton's method to this reformulated system. In this way, smoothing-type methods avoid the explicit inequality constraints, and therefore the iterates generated by these methods do not necessarily belong to the positive orthant. More details on smoothing methods are given in Section 2.

The algorithm to be presented in this manuscript belongs to the class of smoothing-type methods. It is closely related to some methods recently proposed by Burke and Xu [2, 3] and further investigated by the authors in [13, 14]. In contrast to these methods, however, we allow a more flexible choice for the parameter τ. Since the precise way this parameter is updated within the algorithm has an enormous influence on the entire behaviour of the algorithm, we feel that this is a highly important topic. The second motivation for writing this paper is the fact that our current code gives significantly better numerical results than previous implementations of smoothing-type methods. For some further background on smoothing-type methods, the interested reader is referred to [4, 6, 7, 8, 16, 17, 20, 22, 23] and references therein.

The paper is organized as follows: We develop our algorithm in Section 2, give a detailed statement and show that it is well-defined. Section 3 then discusses the global and local convergence properties of our algorithm. In particular, it will be shown that the method has the same nice global convergence properties as the method suggested by Burke and Xu [3]. Section 4 indicates that the method works quite well on the whole netlib test suite. We then close this paper with some final remarks in Section 5.

A few words about our notation: $\mathbb{R}^n$ denotes the n-dimensional real vector space. For $x \in \mathbb{R}^n$, we use the subscript x_i in order to indicate the ith component of x, whereas a superscript like in x^k is used to indicate that this is the kth iterate of a sequence $\{x^k\} \subseteq \mathbb{R}^n$. Quite often, we will consider a triple of the form $w = (x^T, \lambda^T, s^T)^T$, where $x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m$, and $s \in \mathbb{R}^n$; of course, w is then a vector in $\mathbb{R}^{n+m+n}$. In order to simplify our notation, however, we will usually write $w = (x, \lambda, s)$ instead of using the mathematically more correct formula $w = (x^T, \lambda^T, s^T)^T$. If $x \in \mathbb{R}^n$ is a vector whose components are all nonnegative, we simply write $x \geq 0$; an expression like $x \leq 0$ has a similar meaning. Finally, the symbol $\|\cdot\|$ is used for the Euclidean vector norm.

2 Description of Algorithm

In this section, we want to derive our predictor-corrector smoothing method for the solution of the optimality conditions (2). Furthermore, we will see that the method is well-defined.

Since the main idea of our method is based on a suitable reformulation of the optimality conditions (2), we begin with a very simple way to reformulate this system. To this end, let $\varphi : \mathbb{R}^2 \to \mathbb{R}$ denote the so-called minimum function

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

and let $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ be defined by

$$\phi(x, s) := \big(\varphi(x_1, s_1), \dots, \varphi(x_n, s_n)\big)^T.$$

Since φ has the property that

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

it follows that ϕ can be used in order to get a characterization of the complementarity conditions:

$$x_i \geq 0, s_i \geq 0, x_i s_i = 0 \, (i = 1, \dots, n) \iff \phi(x, s) = 0.$$

Consequently, a vector $w^* = (x^*, \lambda^*, s^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ is a solution of the optimality conditions (2) if and only if it satisfies the nonlinear system of equations

$$\Phi(w) = 0,$$

where $\Phi : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ is given by

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

The main disadvantage of the mapping Φ is that it is not differentiable everywhere. In order to overcome this nonsmoothness, several researchers (see, e.g., [7, 5, 18, 21]) have proposed to approximate the minimum function φ by a continuously differentiable mapping with the help of a so-called smoothing parameter τ. In particular, the function

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

has become quite popular and is typically called the Chen-Harker-Kanzow-Smale smoothing function in the literature [5, 18, 21]. Based on this function, we may define the mappings

$$\phi_\tau(x,s) := \big(\varphi_\tau(x_1,s_1),\ldots,\varphi_\tau(x_n,s_n)\big)^T$$

and

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

Obviously, Φ_τ is a smooth approximation of Φ for every $\tau > 0$, and coincides with Φ in the limiting case $\tau = 0$. Furthermore, it was observed in [18] that a vector $w_\tau = (x_\tau, \lambda_\tau, s_\tau)$ solves the nonlinear system of equations

$$\Phi_\tau(w) = 0 \tag{4}$$

if and only if this vector is a solution of the central path conditions (3). Solving the system (4) by, say, Newton's method, is therefore closely related to several primal-dual path-following methods which have become quite popular during the last 15 years, cf. [24].

However, due to our numerical experience [13, 14] and motivated by some stronger theoretical results obtained by Burke and Xu [3], we prefer to view τ as an independent variable (rather than a parameter). To make this clear in our notation, we write

$$\varphi(a,b,\tau) := \varphi_\tau(a,b)$$

and, similarly,

$$\phi(x,s,\tau) := \phi_\tau(x,s)$$

from now on. Since the nonlinear system (4) contains only $n+m+n$ equations and $\Phi(x,\lambda,s,\tau) := \Phi_\tau(x,\lambda,s)$ has $n+m+n+1$ variables, we add one more equation and define a mapping

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

cf. [3]. We also need the following generalization of the function Θ:

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

here, $\sigma \in (0,1]$ denotes a suitable centering parameter, and $\psi : [0,\infty) \to \mathbb{R}$ is a function having the following properties:

(P.1) ψ is continuously differentiable with $\psi(0) = 0$.

(P.2) $\psi'(\tau) > 0$ for all $\tau \in [0,\infty)$.

(P.3) $\psi(\tau) \leq \psi'(\tau) \cdot \tau$ for all $\tau \in [0,\infty)$.

(P.4) For each $\tau_0 > 0$, there is a constant $\gamma > 0$ (possibly depending on τ_0) such that $\psi(\tau) \geq \gamma \cdot \psi'(\tau) \cdot \tau$ for all $\tau \in [0, \tau_0]$.

The following functions satisfy all these properties:

$$\begin{aligned} \psi(\tau) &:= \tau, \\ \psi(\tau) &:= (1+\tau)^2 - 1, \\ \psi(\tau) &:= \exp(\tau) - 1. \end{aligned}$$

In fact, it is quite easy to see that all three examples satisfy properties (P.1), (P.2), and (P.3). Furthermore, the mapping $\psi(\tau) = \tau$ satisfies (P.4) with $\gamma := 1$ being independent of τ_0. Also the mapping $\psi(\tau) := (1+\tau)^2 - 1$ satisfies (P.4) with $\gamma := 1/2$ being independent of τ_0. On the other hand, a simple calculation shows that the third example does satisfy (P.4) with $\gamma := (1 - \exp(-\tau_0))/\tau_0$, i.e., here γ depends on τ_0.

Note that the choice $\psi(\tau) = \tau$ corresponds to the one used in [2, 3], whereas here we aim to generalize the approach from [2, 3] in order to allow a more flexible procedure to decrease τ. Since the precise reduction of τ has a significant influence on the overall performance of our smoothing-type method, we feel that such a generalization is very important from a computational point of view.

Before we give a precise statement of our algorithm, let us add some further comments on the properties of the function ψ: (P.1) is obviously needed since we want to apply a Newton-type method to the system of equations $\Theta_{\sigma,\psi}(x,\lambda,s,\tau) = 0$, hence ψ has to be sufficiently smooth. The second property (P.2) implies that ψ is strictly monotonically increasing. Together with $\psi(0) = 0$ from property (P.1), this means that the nonlinear system of equations

$$\Theta_{\sigma,\psi}(x,\lambda,s,\tau) = 0$$

is equivalent to the optimality conditions (2) themselves (and not to the central path conditions (3)) since the last row immediately gives $\tau = 0$. The third property (P.3) will be used in order to show that the algorithm to be presented below is well-defined, cf. the proof of Lemma 2.2 (c). Furthermore, properties (P.3) and (P.4) together will guarantee that the sequence $\{\tau_k\}$ is monotonically decreasing and converges to zero, see the proof of Theorem 3.3.

We now return to the description of the algorithm. The method to be presented below is a predictor-corrector algorithm with the predictor step being responsible for the local fast rate of convergence, and with the corrector step guaranteeing global convergence. More precisely, the predictor step consists of one Newton iteration applied to the system $\Theta(x,\lambda,s,\tau) = 0$, followed by a suitable update of τ which tries to reduce τ as much as possible. The corrector step then applies one Newton iteration to the system $\Theta_{1,\psi}(x,\lambda,s,\tau)=0$, but with the usual right-hand side $\Theta_{1,\psi}(x,\lambda,s,\tau)$ being replaced by $\Theta_{\sigma,\psi}(x,\lambda,s,\tau)$ for some centering parameter $\sigma\in(0,1)$. This Newton step is followed by an Armijo-type line search.

The computation of all iterates is carried out in such a way that they belong to the neighbourhood

$$\mathcal{N}(\beta) := \{(x,\lambda,s,\tau)\,|\,A^T\lambda+s=c, Ax=b, \|\phi(x,s,\tau)\|\le\beta\tau\}$$

of the central path, where $\beta>0$ denotes a suitable constant. In addition, we will see later that all iterates automatically satisfy the inequality $\phi(x,s,\tau)\le 0$, which will be important in order to establish a result regarding the boundedness of the iterates, cf. Lemma 3.1 and Proposition 3.2 below.

The precise statement of our algorithm is as follows (recall that Θ and $\Theta_{\sigma,\psi}$ denote the mappings from (5) and (6), respectively).

Algorithm 2.1 *(Predictor-Corrector Smoothing Method)*

(S.0) (Initialization)
Choose $w^0 := (x^0,\lambda^0,s^0)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^n$ and $\tau_0>0$ such that $A^T\lambda^0+s^0=c, Ax^0=b$, and $\phi(x^0,s^0,\tau_0)\le 0$, select $\beta\ge\|\phi(x^0,s^0,\tau_0)\|/\tau_0, \rho\in(0,1), 0<\hat\sigma_{\min}<\hat\sigma_{\max}<1, \varepsilon\ge 0$, and set $k:=0$.

(S.1) (Termination Criterion)
If $\|\phi(x^k,s^k,0)\|\le\varepsilon$: STOP.

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

$$\Theta'(w^k,\tau_k)\begin{pmatrix}\Delta w\\ \Delta\tau\end{pmatrix} = -\Theta(w^k,\tau_k). \tag{7}$$

If $\|\phi(x^k+\Delta x^k,s^k+\Delta s^k,0)\|=0$: STOP. Otherwise, if

$$\|\phi(x^k+\Delta x^k,s^k+\Delta s^k,\tau_k)\|>\beta\tau_k,$$

then set

$$\hat w^k := w^k,\ \ \hat\tau_k:=\tau_k,\ \ \eta_k:=1,$$

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

$$\begin{aligned}\|\phi(x^k+\Delta x^k,s^k+\Delta s^k,\rho^j\tau_k)\| &\le \beta\rho^j\tau_k \quad \forall j=0,1,2,\dots,\ell_k \text{ and}\\ \|\phi(x^k+\Delta x^k,s^k+\Delta s^k,\rho^{\ell_k+1}\tau_k)\| &> \beta\rho^{\ell_k+1}\tau_k,\end{aligned}$$

and set $\hat\tau_k:=\eta_k\tau_k$ and

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

(S.3) (Corrector Step)
Choose $\hat{\sigma}_k \in [\hat{\sigma}_{\min}, \hat{\sigma}_{\max}]$, *and compute a solution* $(\Delta\hat{w}^k, \Delta\hat{\tau}_k) = (\Delta\hat{x}^k, \Delta\hat{\lambda}^k, \Delta\hat{s}^k, \Delta\hat{\tau}_k) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}$ *of the linear system*

$$\Theta'_{1,\psi}(\hat{w}^k, \hat{\tau}_k) \begin{pmatrix} \Delta\hat{w} \\ \Delta\hat{\tau} \end{pmatrix} = -\Theta_{\hat{\sigma}_k,\psi}(\hat{w}^k, \hat{\tau}_k). \tag{8}$$

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

$$\|\phi(\hat{x}^k + \hat{t}_k\Delta\hat{x}^k, \hat{s}^k + \hat{t}_k\Delta\hat{s}^k, \hat{\tau}_k + \hat{t}_k\Delta\hat{\tau}_k)\| \le \beta(\hat{\tau}_k + \hat{t}_k\Delta\hat{\tau}_k). \tag{9}$$

Set $w^{k+1} := \hat{w}^k + \hat{t}_k\Delta\hat{w}^k$ *and* $\tau_{k+1} := \hat{\tau}_k + \hat{t}_k\Delta\hat{\tau}_k$.

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

Algorithm 2.1 is closely related to some other methods recently investigated by different authors. For example, if we take $\psi(\tau) = \tau$, then the above algorithm is almost identical with a method proposed by Burke and Xu [3]. It is not completely identical since we use a different update for $\hat{w}^k$ in the predictor step, namely for the case $\ell_k = 0$. This is necessary in order to prove our global convergence results, Theorem 3.3 and Corollary 3.4 below. On the other hand, Algorithm 2.1 is similar to a method used by the authors in [14]; in fact, taking once again $\psi(\tau) = \tau$, we almost have the method from [14]. The only difference that remains is that we use a different right-hand side in the predictor step, namely $\Theta(w^k, \tau_k)$, whereas [14] uses $\Theta(w^k, 0)$. The latter choice seems to give slightly better local properties, however, the current version allows to prove better global convergence properties.

From now on, we always assume that the termination parameter ε in Algorithm 2.1 is equal to zero and that Algorithm 2.1 generates an infinite sequence $\{(x^k, \lambda^k, s^k, \tau_k)\}$, i.e., we assume that the stopping criteria in Steps (S.1) and (S.2) are never satisfied. This is not at all restrictive since otherwise w^k or $w^k + \Delta w^k$ would be a solution of the optimality conditions (2).

We first note that Algorithm 2.1 is well-defined.

Lemma 2.2 *The following statements hold for any* $k \in \mathbb{N}$:

- *(a) The linear systems (7) and (8) have a unique solution.*
- *(b) There is a unique* η_k *satisfying the conditions in Step (S.2).*
- *(c) The stepsize* $\hat{t}_k$ *in (S.3) is uniquely defined.*

Consequently, Algorithm 2.1 is well-defined.

Proof. Taking into account the structure of the Jacobians $\Theta'(w, \tau)$ and $\Theta'_{\sigma,\psi}(w, \tau)$ and using the fact that $\psi'(\tau) > 0$ by property (P.2), part (a) is an immediate consequence of, e.g., [12, Proposition 3.1]. The second statement follows from [13, Proposition 3.2] and is essentially due to Burke and Xu [3]. In order to verify the third statement, assume there is an iteration index k such that

$$\|\phi(\hat{x}^k + \rho^\ell \Delta\hat{x}^k, \hat{s}^k + \rho^\ell \Delta\hat{s}^k, \hat{\tau}_k + \rho^\ell \Delta\hat{\tau}_k)\| > \beta(\hat{\tau}_k + \rho^\ell \Delta\hat{\tau}_k)$$

for all $\ell \in \mathbb{N}$. Since $\|\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k)\| \leq \beta\hat{\tau}_k$, we obtain from property (P.3) that

$$\begin{aligned}
\beta(\hat{\tau}_k + \rho^\ell \Delta\hat{\tau}_k) &= \beta\big(\hat{\tau}_k - \rho^\ell \hat{\sigma}_k \psi(\hat{\tau}_k)/\psi'(\hat{\tau}_k)\big) \\
&\geq \beta\big(\hat{\tau}_k - \rho^\ell \hat{\sigma}_k \hat{\tau}_k\big) \\
&\geq \big(1 - \hat{\sigma}_{\max}\rho^\ell\big)\beta\hat{\tau}_k \\
&\geq \big(1 - \hat{\sigma}_{\max}\rho^\ell\big)\|\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k)\|.
\end{aligned}$$

Taking this inequality into account, the proof can now be completed by using a standard argument for the Armijo line search rule. $\square$

We next state some simple properties of Algorithm 2.1 to which we will refer a couple of times in our subsequent analysis.

Lemma 2.3 *The sequences $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ and $\{\tau_k\}$ generated by Algorithm 2.1 have the following properties:*

(a) $A^T\lambda^k + s^k = c$ *and* $Ax^k = b$ *for all* $k \in \mathbb{N}$.

(b) $\tau_k \leq \tau_0(1 - \gamma\hat{\sigma}_0\hat{t}_0)\eta_0 \cdots (1 - \gamma\hat{\sigma}_{k-1}\hat{t}_{k-1})\eta_{k-1}$ *for all* $k \in \mathbb{N}$, *where* $\gamma > 0$ *denotes the constant from property (P.4).*

(c) $\|\phi(x^k, s^k, \tau_k)\| \leq \beta\tau_k$ *for all* $k \in \mathbb{N}$.

Proof. Part (a) holds for $k = 0$ by our choice of the starting point (x^0, λ^0, s^0). Hence it holds for all $k \in \mathbb{N}$ since Newton's method solves linear systems exactly. In order to verify statement (b), we first note that we get

$$\Delta\hat{\tau}_k = -\hat{\sigma}_k \psi(\hat{\tau}_k)/\psi'(\hat{\tau}_k) \tag{10}$$

from the fourth block row of the linear equation (8). Since $\tau_k, \hat{\tau}_k \in [0, \tau_0]$ for all $k \in \mathbb{N}$, it therefore follows from property (P.4) and the updating rules in steps (S.2) and (S.3) of Algorithm 2.1 that

$$\begin{aligned}
\tau_{k+1} &= \hat{\tau}_k + \hat{t}_k \Delta\hat{\tau}_k \\
&= \hat{\tau}_k - \hat{t}_k \hat{\sigma}_k \psi(\hat{\tau}_k)/\psi'(\hat{\tau}_k) \\
&\leq \hat{\tau}_k - \gamma\hat{t}_k \hat{\sigma}_k \hat{\tau}_k \\
&= (1 - \gamma\hat{t}_k\hat{\sigma}_k)\eta_k\tau_k.
\end{aligned}$$

Using a simple induction argument, we see that (b) holds. Finally, statement (c) is a direct consequence of the updating rules in Algorithm 2.1. $\square$

3 Convergence Properties

In this section, we analyze the global and local convergence properties of Algorithm 2.1. Since the analysis for the local rate of convergence is essentially the same as in [3] (recall that our predictor step is identically to the one from [3]), we focus on the global properties. In particular, we will show that all iterates (x^k, λ^k, s^k) remain bounded under a strict feasibility assumption. This was noted by Burke and Xu [3] for a particular member of our class of methods (namely for the choice $\psi(\tau) := \tau$), but is not true for many other smoothing-type methods like those from [5, 6, 7, 8, 13, 14, 22, 23].

The central observation which allows us to prove the boundedness of the iterates (x^k, λ^k, s^k) is that they automatically satisfy the inequality

$$\phi(x^k, s^k, \tau_k) \le 0$$

for all $k \in \mathbb{N}$ provided this inequality holds for $k = 0$. This is precisely the statement of our first result.

Lemma 3.1 *The sequences $\{w^k\} = \{(x^k, \lambda^k, s^k)\}, \{\tau_k\}, \{\hat{w}^k\} = \{(\hat{x}^k, \hat{\lambda}^k, \hat{s}^k)\}$ and $\{\hat{\tau}_k\}$ generated by Algorithm 2.1 have the following properties:*

(a) $\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) \le 0$ for all $k \in \mathbb{N}$.

(b) $\phi(x^k, s^k, \tau_k) \le 0$ for all $k \in \mathbb{N}$.

Proof. We first derive some useful inequalities, and then verify the two statements simultaneously by induction on k.

We begin with some preliminary discussions regarding statement (a). To this end, let $k \in \mathbb{N}$ be fixed for the moment, and assume that we take $\hat{w}^k = w^k + \Delta w^k$ in Step (S.2) of Algorithm 2.1. Since each component of the function ϕ is concave, we then obtain

$$\begin{aligned}
&\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) \\
&\quad = \phi(x^k + \Delta x^k, s^k + \Delta s^k, \eta_k \tau_k) \\
&\quad = \phi(x^k + \Delta x^k, s^k + \Delta s^k, \tau_k + (\eta_k - 1)\tau_k) \\
&\quad \le \phi(x^k, s^k, \tau_k) + \phi'(x^k, s^k, \tau_k) \begin{pmatrix} \Delta x^k \\ \Delta s^k \\ (\eta_k - 1)\tau_k \end{pmatrix} \\
&\quad = \phi(x^k, s^k, \tau_k) + \phi'(x^k, s^k, \tau_k) \begin{pmatrix} \Delta x^k \\ \Delta s^k \\ \Delta \tau_k \end{pmatrix} + \phi'(x^k, s^k, \tau_k) \begin{pmatrix} 0 \\ 0 \\ (\eta_k - 1)\tau_k - \Delta \tau_k \end{pmatrix}.
\end{aligned} \tag{11}$$

From the third block row of (7), we have

$$\phi'(x^k, s^k, \tau_k) \begin{pmatrix} \Delta x^k \\ \Delta s^k \\ \Delta \tau_k \end{pmatrix} = -\phi(x^k, s^k, \tau_k).$$

Hence we get from (11):

$$\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) \le \phi'(x^k, s^k, \tau_k) \begin{pmatrix} 0 \\ 0 \\ (\eta_k - 1)\tau_k - \Delta\tau_k \end{pmatrix}. \tag{12}$$

We claim that the right-hand side of (12) is nonpositive. To prove this statement, we first note that

$$\phi'(x^k, s^k, \tau_k) \begin{pmatrix} 0 \\ 0 \\ (\eta_k - 1)\tau_k - \Delta\tau_k \end{pmatrix} = \big((\eta_k - 1)\tau_k - \Delta\tau_k\big) d_\tau^k$$

with

$$\begin{aligned} d_\tau^k &:= \left(\frac{\partial\varphi}{\partial\tau}(x_1^k, s_1^k, \tau_k), \dots, \frac{\partial\varphi}{\partial\tau}(x_n^k, s_n^k, \tau_k)\right)^T \\ &= \left(\frac{-4\tau_k}{\sqrt{(x_1^k - s_1^k)^2 + 4\tau_k^2}}, \dots, \frac{-4\tau_k}{\sqrt{(x_n^k - s_n^k)^2 + 4\tau_k^2}}\right)^T \\ &\le 0. \end{aligned}$$

Hence it remains to show that

$$(\eta_k - 1)\tau_k - \Delta\tau_k \ge 0.$$

However, this is obvious since the last row of the linear system (7) implies $\Delta\tau_k = -\tau_k$.

We next derive some useful inequalities regarding statement (b). To this end, we still assume that $k \in \mathbb{N}$ is fixed. Using once again the fact that ϕ is a concave function in each component, we obtain from (8)

$$\begin{aligned} \phi(x^{k+1}, s^{k+1}, \tau_{k+1}) &= \phi(\hat{x}^k + \hat{t}_k\Delta\hat{x}^k, \hat{s}^k + \hat{t}_k\Delta\hat{s}^k, \hat{\tau}_k + \hat{t}_k\Delta\hat{\tau}_k) \\ &\le \phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) + \hat{t}_k\phi'(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) \begin{pmatrix} \Delta\hat{x}^k \\ \Delta\hat{s}^k \\ \Delta\hat{\tau}_k \end{pmatrix} \\ &= \phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) - \hat{t}_k\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) \\ &= (1 - \hat{t}_k)\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k), \end{aligned} \tag{13}$$

and this completes our preliminary discussions.

We now verify statements (a) and (b) by induction on k. For $k = 0$, we have $\phi(x^0, s^0, \tau_0) \le 0$ by our choice of the starting point $w^0 = (x^0, \lambda^0, s^0)$ and the initial smoothing parameter $\tau_0 > 0$ in Step (S.0) of Algorithm 2.1. Therefore, if we set $\hat{w}^0 = w^0$ in Step (S.2) of Algorithm 2.1, we also have $\hat{\tau}_0 = \tau_0$ and therefore $\phi(\hat{x}^0, \hat{s}^0, \hat{\tau}_0) \le 0$. On the other hand, if we set $\hat{w}^0 = w^0 + \Delta w^0$ in Step (S.2), the argument used in the beginning of this proof shows that the inequality $\phi(\hat{x}^0, \hat{s}^0, \hat{\tau}_0) \le 0$ also holds in this case.

Suppose that we have $\phi(x^k, s^k, \tau_k) \le 0$ and $\phi(\hat{x}^k, \hat{s}^k, \hat{\tau}_k) \le 0$ for some $k \in \mathbb{N}$. Then (13) immediately implies that we have $\phi(x^{k+1}, s^{k+1}, \tau_{k+1}) \le 0$. Consequently, if we have $\hat{w}^{k+1} = w^{k+1}$ in Step (S.2) of Algorithm 2.1, we obviously have $\phi(\hat{x}^{k+1}, \hat{s}^{k+1}, \hat{\tau}_{k+1}) \le 0$. Otherwise, i.e., if we set $\hat{w}^{k+1} = w^{k+1} + \Delta w^{k+1}$ in Step (S.2), the argument used in the beginning part of this proof shows that the same inequality holds. This completes the formal proof by induction. □

We next show that the sequence $\{w^k\}$ generated by Algorithm 2.1 remains bounded provided that there is a strictly feasible point for the optimality conditions (2), i.e., a vector $\hat{w} = (\hat{x}, \hat{\lambda}, \hat{s})$ satisfying $A^T\hat{\lambda} + \hat{s} = c, A\hat{x} = b$ and $\hat{x} > 0, \hat{s} > 0$.

Proposition 3.2 *Assume that there is a strictly feasible point for the optimality conditions (2). Then the sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ generated by Algorithm 2.1 is bounded.*

Proof. The statement is essentially due to Burke and Xu [3], and we include a proof here only for the sake of completeness.

Assume that the sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ generated by Algorithm 2.1 is unbounded. Since $\{\tau_k\}$ is monotonically decreasing by Lemma 2.3 (b), it follows from Lemma 2.3 (c) that

$$\|\phi(x^k, s^k, \tau_k)\| \le \beta\tau_k \le \beta\tau_0 \tag{14}$$

for all $k \in \mathbb{N}$. The definition of the (smoothed) minimum function therefore implies that there is no index $i \in \{1, \dots, n\}$ such that $x_i^k \to -\infty$ or $s_i^k \to -\infty$ on a subsequence, since otherwise we would have $\varphi(x_i^k, s_i^k, \tau_k) \to -\infty$ which, in turn, would imply $\|\phi(x^k, s^k, \tau_k)\| \to +\infty$ on a subsequence in contrast to (14). Therefore, all components of the two sequences $\{x^k\}$ and $\{s^k\}$ are bounded from below, i.e.,

$$x_i^k \ge \gamma \quad \text{and} \quad s_i^k \ge \gamma \quad \forall i = 1, \dots, n, \ \forall k \in \mathbb{N}, \tag{15}$$

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 $i \in \{1, \dots, n\}$ 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 $k \in \mathbb{N}$ (cf. Lemma 2.3 (a)) 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 (2) whose existence is guaranteed by our assumption. 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 $k \in \mathbb{N}$ by Lemma 2.3 (a), we get

$$A^T(\hat{\lambda} - \lambda^k) + (\hat{s} - s^k) = 0 \quad \text{and} \quad A(\hat{x} - x^k) = 0 \tag{16}$$

by subtracting these equations. Premultiplying the first equation in (16) with $(\hat{x} - x^k)^T$ and taking into account the second equation in (16) gives

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

Reordering this equation, we obtain

$$\hat{s}^T x^k + \hat{x}^T s^k = (x^k)^T s^k + \hat{x}^T \hat{s} \tag{17}$$

for all $k \in \mathbb{N}$. Using (15) 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 (17) and the fact that $x_i^k \to +\infty$ or $s_i^k \to +\infty$ on a subsequence for at least one index $i \in \{1, \ldots, n\}$ that

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

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

$$x_j^k s_j^k \to +\infty \tag{18}$$

on a suitable subsequence.

Now, using Lemma 3.1 (b), we have

$$\phi(x^k, s^k, \tau_k) \leq 0$$

for all $k \in \mathbb{N}$. Taking into account the definition of ϕ and looking at the j-th component, this implies

$$x_j^k + s_j^k \leq \sqrt{(x_j^k - s_j^k)^2 + 4\tau_k^2} \tag{19}$$

for all $k \in \mathbb{N}$. Using (18) and (15), we see that we necessarily have $x_j^k > 0$ and $s_j^k > 0$ for all those k belonging to the subsequence for which (18) holds. Therefore, taking the square in (19), we obtain

$$4x_j^k s_j^k \leq 4\tau_k^2$$

after some simplifications. However, since the right-hand side of this expression is bounded by $4\tau_0^2$, this gives a contradiction to (18). $\square$

We next prove a global convergence result for Algorithm 2.1. Note that this result is different from the one provided by Burke and Xu [3] and is more in the spirit of those from [22, 13, 14]. (Burke and Xu [3] use a stronger assumption in order to prove a global linear rate of convergence for the sequence $\{\tau_k\}$.)

Theorem 3.3 *Assume that the sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ generated by Algorithm 2.1 has at least one accumulation point. Then $\{\tau_k\}$ converges to zero.*

Proof. Since the sequence $\{\tau_k\}$ is monotonically decreasing (by Lemma 2.3 (b)) and bounded from below by zero, it converges to a number $\tau_* \geq 0$. If $\tau_* = 0$, we are done.

So assume that $\tau_* > 0$. Then the updating rules in Step (S.2) of Algorithm 2.1 immediately give

$$\hat{w}^k = w^k, \quad \hat{\tau}_k = \tau_k, \quad \text{and} \quad \eta_k = 1 \tag{20}$$

for all $k \in \mathbb{N}$ sufficiently large. Subsequencing if necessary, we assume without loss of generality that (20) holds for all $k \in \mathbb{N}$. Then Lemma 2.3 (b) and $\hat{\sigma}_k \geq \hat{\sigma}_{\min}$ yield

$$\tau_k \leq \tau_0 \prod_{j=0}^{k-1}(1-\gamma\hat{\sigma}_j\hat{t}_j) \leq \tau_0 \prod_{j=0}^{k-1}(1-\gamma\hat{\sigma}_{\min}\hat{t}_j). \tag{21}$$

Since $\tau_k \to \tau_* > 0$ by assumption, it follows from (21) that $\lim_{k\to\infty}\hat{t}_k = 0$. Therefore, the stepsize $\hat{\alpha}_k := \hat{t}_k/\rho$ does not satisfy the line search criterion (9) for all $k \in \mathbb{N}$ large enough. Hence we have

$$\|\phi(\hat{x}^k + \hat{\alpha}_k\Delta\hat{x}^k, \hat{s}^k + \hat{\alpha}_k\Delta\hat{s}^k, \hat{\tau}_k + \hat{\alpha}_k\Delta\hat{\tau}_k)\| > \beta(\hat{\tau}_k + \hat{\alpha}_k\Delta\hat{\tau}_k) \tag{22}$$

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

Now let $w^* = (x^*, \lambda^*, s^*)$ be an accumulation point of the sequence $\{w^k\}$, and let $\{w^k\}_K$ be a subsequence converging to w^*. Since $\hat{\sigma}_k \in [\hat{\sigma}_{\min}, \hat{\sigma}_{\max}]$ for all $k \in \mathbb{N}$, we can assume without loss of generality that the subsequence $\{\hat{\sigma}_k\}_K$ converges to some number $\hat{\sigma}_* \in [\hat{\sigma}_{\min}, \hat{\sigma}_{\max}]$. Furthermore, since $\tau_* > 0$, it follows from (20) and Lemma 2.2 (a) that the corresponding subsequence $\{(\Delta\hat{w}^k, \Delta\hat{\tau}_k)\}_K$ converges to a vector $(\Delta\hat{w}^*, \Delta\hat{\tau}_*) = (\Delta\hat{x}^*, \Delta\hat{\lambda}^*, \Delta\hat{s}^*, \Delta\hat{\tau}_*)$, where $(\Delta\hat{w}^*, \Delta\hat{\tau}_*)$ is the unique solution of the linear equation

$$\Theta'_{1,\psi}(w^*, \tau_*)\begin{pmatrix}\Delta\hat{w}\\ \Delta\hat{\tau}\end{pmatrix} = -\Theta_{\hat{\sigma}_*,\psi}(w^*, \tau_*), \tag{23}$$

cf. (8). Using $\{\hat{\alpha}_k\}_K \to 0$ and taking the limit $k \to \infty$ on the subset K, we then obtain from (20) and (22) that

$$\|\phi(x^*, s^*, \tau_*)\| \geq \beta\tau_* > 0. \tag{24}$$

On the other hand, we get from (22), (10), property (P.3), (20), Lemma 2.3 (c), and $\hat{\sigma}_k \leq \hat{\sigma}_{\max}$ that

$$\begin{aligned}
\|\phi(\hat{x}^k + \hat{\alpha}_k\Delta\hat{x}^k, \hat{s}^k + \hat{\alpha}_k\Delta\hat{s}^k, \hat{\tau}_k + \hat{\alpha}_k\Delta\hat{\tau}_k)\| &> \beta\big(\hat{\tau}_k + \hat{\alpha}_k\Delta\hat{\tau}_k\big)\\
&= \beta\big(\hat{\tau}_k - \hat{\alpha}_k\hat{\sigma}_k\psi(\hat{\tau}_k)/\psi'(\hat{\tau}_k)\big)\\
&\geq \beta\big(\hat{\tau}_k - \hat{\alpha}_k\hat{\sigma}_k\hat{\tau}_k\big)\\
&= (1-\hat{\sigma}_k\hat{\alpha}_k)\beta\tau_k\\
&\geq (1-\hat{\sigma}_k\hat{\alpha}_k)\|\phi(x^k, s^k, \tau_k)\|\\
&\geq (1-\hat{\sigma}_{\max}\hat{\alpha}_k)\|\phi(x^k, s^k, \tau_k)\|
\end{aligned}$$

for all $k \in \mathbb{N}$ sufficiently large. Using (20), this implies

$$\frac{\|\phi(x^k + \hat{\alpha}_k\Delta\hat{x}^k, s^k + \hat{\alpha}_k\Delta\hat{s}^k, \tau_k + \hat{\alpha}_k\Delta\hat{\tau}_k)\| - \|\phi(x^k, s^k, \tau_k)\|}{\hat{\alpha}_k} \geq -\hat{\sigma}_{\max}\|\phi(x^k, s^k, \tau_k)\|.$$

Since $\|\phi(\cdot,\cdot,\cdot)\|$ is a continuously differentiable function at (x^*, s^*, τ_*) due to (24), taking the limit $k \to \infty$ for $k \in K$ then gives

$$\frac{\phi(x^*, s^*, \tau_*)^T}{\|\phi(x^*, s^*, \tau_*)\|}\phi'(x^*, s^*, \tau_*)\begin{pmatrix}\Delta\hat{x}^*\\ \Delta\hat{s}^*\\ \Delta\hat{\tau}_*\end{pmatrix} \geq -\hat{\sigma}_{\max}\|\phi(x^*, s^*, \tau_*)\|,$$

where $(\Delta\hat{w}^*, \Delta\hat{\tau}_*) = (\Delta\hat{x}^*, \Delta\hat{\lambda}^*, \Delta\hat{s}^*, \Delta\hat{\tau}_*)$ denotes the solution of the linear system (23). Using (23) then gives

$$-\|\phi(x^*, s^*, \tau_*)\| \geq -\hat{\sigma}_{\max}\|\phi(x^*, s^*, \tau_*)\|.$$

Since $\hat{\sigma}_{\max} \in (0, 1)$, this implies $\|\phi(x^*, s^*, \tau_*)\| = 0$, a contradiction to (24). Hence we cannot have $\tau_* > 0$. □

Due to Proposition 3.2, the assumed existence of an accumulation point in Theorem 3.3 is automatically satisfied if there is a strictly feasible point for the optimality conditions (2). An immediate consequence of Theorem 3.3 is the following result.

Corollary 3.4 *Every accumulation point of a sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ generated by Algorithm 2.1 is a solution of the optimality conditions (2).*

Proof. The short proof is essentially the same as in [14], for example, and we include it here for the sake of completeness. — Let $w^* = (x^*, \lambda^*, s^*)$ be an accumulation point of the sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$, and let $\{w^k\}_K$ denote a subsequence converging to w^*. Then we have $\tau_k \to 0$ in view of Theorem 3.3. Hence Lemma 2.3 (c) implies

$$\|\phi(x^*, s^*, 0)\| = \lim_{k\in K} \|\phi(x^k, s^k, \tau_k)\| \leq \beta \lim_{k\in K} \tau_k = 0,$$

i.e., we have $x^* \geq 0, s^* \geq 0$ and $x_i^* s_i^* = 0$ for $i = 1, \dots, n$ due to the definition of ϕ. Since Lemma 2.3 (a) also shows that we have $A^T\lambda^* + s^* = c$ and $Ax^* = b$, we see that $w^* = (x^*, \lambda^*, s^*)$ is indeed a solution of the optimality conditions (2). □

We finally state our local rate of convergence result. Since our predictor step coincides with the one by Burke and Xu [3], the proof of this result is essentially the same as in [3], and we therefore omit the details here.

Theorem 3.5 *Let the parameter β satisfy the inequality $\beta > 2\sqrt{n}$, assume that the optimality conditions (2) have a unique solution $w^* = (x^*, \lambda^*, s^*)$, and suppose that the sequence $\{w^k\} = \{(x^k, \lambda^k, s^k)\}$ generated by Algorithm 2.1 converges to w^*. Then $\{\tau_k\}$ converges globally linearly and locally quadratically to zero.*

The central observation in order to prove Theorem 3.5 is that the sequence of Jacobian matrices $\Theta'(w^k, \tau_k)$ converges to a nonsingular matrix under the assumption of Theorem 3.5. In fact, as noted in [3, 12], the convergence of this sequence to a nonsingular Jacobian matrix is equivalent to the unique solvability of the optimality conditions (2).

4 Numerical Results

We implemented Algorithm 2.1 in C. In order to simplify the work, we took the PCx code from [10, 9] and modified it in an appropriate way. PCx is a predictor-corrector interior-point solver for linear programs, written in C and calling a FORTRAN subroutine in order to solve certain linear systems using the sparse Cholesky method by Ng and Peyton [19]. Since the linear systems occuring in Algorithm 2.1 have essentially the same structure as those arising in interior-point methods, it was possible to use the numerical linear algebra part from PCx for our implementation of Algorithm 2.1. We also apply the preprocessor from PCx before starting our method.

The initial point $w^0 = (x^0, \lambda^0, s^0)$ is the same as the one used for our numerical experiments in [14] and was constructed in the following way:

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

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

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

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

Note that this starting point is feasible in the sense that it satisfies the linear equations $A^T\lambda + s = c$ and $Ax = b$. Furthermore, the initial smoothing parameter was set to

$$\tau_0 := \|\phi(x^0, s^0)\|_\infty,$$

i.e., τ_0 is equal to the initial residual of the optimality conditions (2) (recall that the starting vector satisfies the linear equations in (2) exactly, at least up to numerical inaccuracies). In order to guarantee that $\phi(x^0, s^0, \tau_0) \leq 0$, however, we sometimes have to enlarge the value of τ_0 so that it satisfies the inequalities

$$\tau_0 \geq \sqrt{x_i^0 s_i^0} \quad \forall i \in \{1, \dots, n\} \text{ with } x_i^0 > 0, s_i^0 > 0.$$

Note that the same was done in [14]. We also took the stopping criterion from [14], i.e., we terminate the iteration if one of the following conditions hold:

(a) $\tau_k < 10^{-4}$ or

(b) $\|\Phi(w^k)\|_\infty < 10^{-4}$ or

(c) $\|\Phi(w^k)\|_\infty < 10^{-3}$ and $\|\Phi(w^k)\|_\infty / \|\Phi(w^0)\|_\infty < 10^{-6}$.

Finally, the centering parameter $\hat{\sigma}_k$ was chosen as follows: We let $\hat{\sigma}_{\min} = 0.4, \hat{\sigma}_{\max} = 0.6, \gamma = 0.1$, start with $\hat{\sigma}_0 = 0.5$ and set

$$\hat{\sigma}_{k+1} := \min\{\hat{\sigma}_{\max}, \hat{\sigma}_k + \gamma\}$$

if the predictor step was successful (i.e., if we were allowed to take $\hat{w}^{k+1} = w^k + \Delta w^k$), and

$$\hat{\sigma}_{k+1} := \max\{\hat{\sigma}_{\min}, \hat{\sigma}_k - \gamma\}$$

otherwise. This strategy guarantees that all centering parameters belong to the interval $[\hat{\sigma}_{\min}, \hat{\sigma}_{\max}]$. According to our experience, a larger value of $\hat{\sigma}_k$ usually gives faster convergence, but the entire behaviour of our method becomes more unstable, whereas a smaller value of the centering parameter gives a more stable behaviour, while the overall number of iterations increases. The dynamic choice of $\hat{\sigma}_k$ given above tries to combine these observations in a suitable way.

The remaining parameters from Step (S.0) of Algorithm 2.1 were chosen as follows:

$$\rho = 0.79 \quad \text{and} \quad \beta := \|\phi(x^0, s^0, \tau_0)\|/\tau_0.$$

We first consider the function $\psi(\tau) := \tau$ (this, more or less, corresponds to the method from [14])

All test runs were done on a SUN Enterprise 450 with 480 MHz. Table 1 contains the corresponding results, with the columns of Table 1 having the following meanings:

problem:	name of the test problem in the netlib collection,
m:	number of equality constraints (after preprocessing),
n:	number of variables (after preprocessing),
k:	number of iterations until termination,
P:	number of accepted predictor steps,
τ_f:	value of τ_k at the final iterate,
$\|\Phi(w^f)\|_\infty$:	value of $\|\Phi(w^k)\|_\infty$ at the final iterate,
primal objective:	value of the primal objective function at final iterate.

Moreover, we give the number of iterations needed by the related method from [14] in parantheses after the number of iterations used by our new method.

Table 1: Numerical results for Algorithm 2.1

problem	m	n	k	P	τ_f	$\|\Phi(w^f)\|_\infty$	primal objective
25fv47	788	1843	27 (34)	11	1.0e−03	1.758e−04	5.50184589e+03
80bau3b	2140	11066	29 (29)	15	6.0e−04	2.527e−04	9.87224192e+05
adlittle	55	137	14 (15)	12	3.3e−02	2.239e−04	2.25494963e+05
afiro	27	51	12 (10)	12	1.9e−02	5.456e−06	−4.64753155e+02
agg	390	477	22 (23)	17	3.8e−02	6.257e−04	−3.59917673e+07
agg2	514	750	22 (25)	16	2.1e−02	4.992e−04	−2.02392524e+07
agg3	514	750	21 (30)	16	3.1e−02	5.673e−04	1.03121159e+07
bandm	240	395	15 (20)	12	2.6e−04	9.065e−05	−1.58628032e+02
beaconfd	86	171	21 (18)	18	2.5e−03	5.156e−04	3.35924858e+04
blend	71	111	10 (13)	9	8.1e−04	8.746e−05	−3.08121828e+01
bnl1	610	1491	30 (26)	16	8.2e−04	1.737e−04	1.97762956e+03
bnl2	1964	4008	25 (26)	10	9.3e−04	4.893e−04	1.81123495e+03
boeing1	331	697	18 (26)	13	1.7e−03	3.652e−04	−3.35213568e+02
boeing2	126	265	18 (16)	13	2.7e−03	4.166e−06	−3.15018729e+02
bore3d	81	138	14 (28)	11	5.9e−03	3.980e−05	1.37308039e+03
brandy	133	238	16 (19)	14	2.1e−03	3.469e−04	1.51850990e+03
capri	241	436	15 (20)	14	4.0e−03	9.161e−04	2.69000997e+03
cycle	1420	2773	30 (39)	14	8.2e−05	8.349e−05	−5.22639964e+00
czprob	671	2779	17 (22)	13	5.7e−03	5.207e−05	2.18519670e+06
d2q06c	2132	5728	48 (57)	18	1.8e−04	5.591e−04	1.22784214e+05
d6cube	403	5443	13 (25)	10	5.3e−04	3.157e−05	3.15491667e+02

Table 1 (continued): Numerical results for Algorithm 2.1

problem	m	n	k	P	τ_f	$\|\Phi(w^f)\|_\infty$	primal objective
degen2	444	757	10 (23)	10	1.9e−03	2.901e−05	−1.43517800e+03
degen3	1503	2604	10 (16)	10	9.2e−04	7.742e−05	−9.87294001e+02
dfl001	—	—	— (—)	—	—	—	—
e226	198	429	14 (27)	13	2.4e−04	1.889e−05	−2.58649291e+01
etamacro	334	669	20 (26)	13	8.8e−05	1.625e−03	−7.55715232e+02
fffff800	322	826	28 (36)	17	1.2e−03	5.876e−04	5.55679564e+05
finnis	438	935	20 (31)	17	2.0e−03	7.843e−04	1.72791066e+05
fit1d	24	1049	14 (20)	14	2.7e−03	8.491e−05	−9.14637809e+03
fit1p	627	1677	17 (19)	14	7.0e−05	3.469e−02	9.14648712e+03
fit2d	25	10524	17 (22)	17	1.4e−03	7.494e−04	−6.84642932e+04
fit2p	3000	13525	19 (20)	19	1.5e−03	9.397e−05	6.84642933e+04
forplan	121	447	26 (28)	17	2.2e−03	4.722e−04	−6.64218959e+02
ganges	1113	1510	20 (25)	19	2.4e−03	1.218e−04	−1.09585736e+05
gfrd-pnc	590	1134	17 (23)	15	3.2e−02	4.308e−04	6.90223600e+06
greenbea	—	—	— (25)	—	—	—	—
greenbeb	1932	4154	43 (35)	13	1.7e−03	9.559e−04	−4.30226026e+06
israel	174	316	17 (27)	15	1.0e−02	4.732e−04	−8.96644822e+05
kb2	43	68	15 (32)	10	1.6e−03	1.653e−06	−1.74990013e+03
lotfi	133	346	23 (35)	12	3.2e−03	7.087e−04	−2.52647043e+01
maros	655	1437	22 (37)	14	2.4e−03	1.738e−04	−5.80637437e+04
maros-r7	2152	7440	39 (22)	14	4.0e−03	8.053e−04	1.49718517e+06
modszk1	665	1599	21 (26)	17	7.2e−03	3.330e−04	3.20619729e+02
nesm	654	2922	46 (52)	9	4.7e−04	4.718e−04	1.40760365e+07
perold	593	1374	26 (33)	12	2.1e−03	6.564e−04	−9.38075527e+03
pilot	1368	4543	71 (81)	9	9.0e−05	1.600e−02	−5.57274205e+02
pilot.ja	810	1804	30 (76)	14	7.1e−04	9.749e−04	−6.11313652e+03
pilot.we	701	2814	36 (61)	10	2.6e−03	9.981e−04	−2.72010753e+06
pilot4	396	1022	26(132)	12	1.7e−03	6.888e−04	−2.58113924e+03
pilot87	1971	6373	67 (63)	11	9.1e−05	6.095e−03	3.01715468e+02
pilotnov	848	2117	15 (27)	15	2.3e−03	4.059e−04	−4.49727619e+03
recipe	64	123	11 (14)	10	1.2e−03	4.205e−05	−2.66616000e+02
sc105	104	162	18 (19)	13	1.2e−03	2.793e−05	−5.22020617e+01
sc205	203	315	24 (22)	14	6.8e−04	1.030e−04	−5.22020617e+01
sc50a	49	77	14 (15)	11	4.7e−03	8.546e−05	−6.45750802e+01
sc50b	48	76	15 (14)	10	7.1e−03	7.714e−06	−7.00000047e+01
scagr25	469	669	31 (19)	13	4.7e−03	1.049e−04	−1.47534331e+07
scagr7	127	183	15 (19)	14	3.6e−03	4.563e−04	−2.33138982e+06
scfxm1	305	568	15 (20)	13	8.4e−03	6.230e−04	1.84167590e+04
scfxm2	610	1136	18 (26)	15	2.7e−03	1.834e−04	3.66602616e+04
scfxm3	915	1704	20 (26)	15	5.4e−03	9.098e−04	5.49012545e+04
scorpion	340	412	19 (21)	14	2.4e−04	1.815e−05	1.87812482e+03
scrs8	421	1199	17 (21)	14	1.9e−03	2.169e−04	9.04293215e+02

Table 1 (continued): Numerical results for Algorithm 2.1

problem	m	n	k	P	τ_f	$\|\Phi(w^f)\|_\infty$	primal objective
scsd1	77	760	12 (22)	12	4.2e−03	7.203e−06	8.66666364e+00
scsd6	147	1350	11 (15)	8	3.8e−04	1.937e−05	5.05000001e+01
scsd8	397	2750	9 (13)	9	8.7e−03	3.131e−05	9.04999988e+02
sctap1	284	644	12 (24)	12	9.4e−03	8.910e−06	1.41224999e+03
sctap2	1033	2443	11 (18)	11	7.2e−03	8.233e−05	1.72480714e+03
sctap3	1408	3268	12 (18)	12	3.8e−03	1.051e−05	1.42400000e+03
seba	448	901	19 (23)	12	2.5e−03	1.550e−06	1.57116000e+04
share1b	112	248	29 (43)	14	2.2e−03	3.762e−04	−7.65893186e+04
share2b	96	162	15 (16)	10	2.5e−03	8.099e−05	−4.15732240e+02
shell	487	1451	19 (22)	16	3.4e−01	4.313e−04	1.20882535e+09
ship04l	292	1905	22 (20)	16	5.3e−03	7.616e−04	1.79332454e+06
ship04s	216	1281	16 (20)	13	6.9e−03	1.561e−04	1.79871470e+06
ship08l	470	3121	25 (21)	15	2.1e−03	7.592e−04	1.90905521e+06
ship08s	276	1604	15 (20)	13	3.0e−02	7.416e−04	1.92009821e+06
ship12l	610	4171	21 (21)	13	7.0e−03	2.670e−04	1.47018792e+06
ship12s	340	1943	18 (20)	14	5.3e−03	7.548e−05	1.48923613e+06
sierra	1212	2705	20 (22)	16	7.7e−03	2.548e−05	1.53943622e+07
stair	356	532	18 (19)	14	6.9e−04	2.105e−04	−2.51266950e+02
standata	314	796	11 (13)	10	3.9e−02	1.699e−05	1.25769927e+03
standgub	314	796	11 (13)	10	3.9e−02	1.699e−05	1.25769927e+03
standmps	422	1192	14 (18)	12	9.6e−03	4.418e−05	1.40601750e+03
stocfor1	102	150	13 (16)	13	2.1e−02	2.014e−05	−4.11319832e+04
stocfor2	1980	2868	14 (29)	13	2.0e−03	1.481e−06	−3.90243999e+04
stocfor3	15362	22228	23 (63)	19	2.8e−04	5.514e−05	−3.99767839e+04
stocfor3old	15362	22228	23 (70)	19	2.8e−04	5.514e−05	−3.99767839e+04
truss	1000	8806	16 (19)	13	2.5e−03	3.621e−04	4.58815785e+05
tuff	257	567	20 (32)	15	1.9e−04	4.977e−05	2.92147852e−01
vtp.base	72	111	10 (19)	10	2.9e−01	3.126e−04	1.29831462e+05
wood1p	171	1718	22 (13)	11	1.5e−04	9.009e−05	1.44286460e+00
woodw	708	5364	36 (34)	10	1.0e−04	5.104e−03	1.30440832e+00

Table 1 clearly indicates that our current implementation works much better than our previous code from [14]. In fact, for almost all examples we were able to reduce the number of iterations considerably.

We finally state some results for the function $\psi(\tau) := (1+\tau)^2 - 1$. Rather than giving another complete list, however, we illustrate the typical behaviour of this method by presenting the corresponding results for those test examples why lie between `kb2` and `scagr7` (this list includes the difficult `pilot*` problems) in Table 2.

Table 2: Numerical results with quadratic function

problem	m	n	k	P	τ_f	$\|\Phi(w^f)\|_\infty$	primal objective
kb2	43	68	15	9	2.0e−03	2.458e−05	−1.74990013e+03
lotfi	133	346	22	9	3.0e−03	6.715e−04	−2.52647449e+01
maros	655	1437	20	11	3.0e−03	3.805e−04	−5.80637438e+04
maros-r7	2152	7440	24	8	1.8e−03	9.450e−04	1.49718510e+06
modszk1	665	1599	26	11	2.5e−03	3.087e−05	3.20619729e+02
nesm	654	2922	45	4	7.5e−04	8.144e−04	1.40760365e+07
perold	593	1374	55	11	5.7e−05	2.585e−04	−9.38075528e+03
pilot	1368	4543	53	7	1.4e−04	2.953e−04	−5.57310815e+02
pilot.ja	810	1804	39	5	2.9e−04	1.480e−04	−6.11313633e+03
pilot.we	701	2814	43	4	9.8e−04	9.283e−04	−2.72010754e+06
pilot4	396	1022	31	7	1.4e−03	5.672e−04	−2.58113925e+03
pilot87	1971	6373	99	3	9.1e−05	1.835e−02	3.02675463e+02
pilotnov	848	2117	23	6	1.4e−03	3.573e−04	−4.49727619e+03
recipe	64	123	10	8	1.1e−03	1.928e−05	−2.66616001e+02
sc105	104	162	15	10	1.7e−03	1.193e−04	−5.22020686e+01
sc205	203	315	18	11	3.0e−04	2.486e−05	−5.22020615e+01
sc50a	49	77	13	10	1.7e−03	3.224e−05	−6.45750795e+01
sc50b	48	76	11	9	4.1e−03	4.955e−05	−7.00000201e+01
scagr25	469	669	20	10	1.1e−02	1.060e−04	−1.47534331e+07
scagr7	127	183	16	10	3.1e−03	9.326e−05	−2.33138982e+06

5 Concluding Remarks

We have presented a class of smoothing-type methods for the solution of linear programs. This class of methods has similar convergence properties as the one by Burke and Xu [3], for example, but allows a more flexible choice for the updating of the smoothing parameter τ. The numerical results presented for our implementation of this smoothing-type method are very encouraging and, in particular, significantly better than for all previous implementations. The results also indicate that the precise updating of the smoothing parameter plays a very important role for the overall behaviour of the methods. However, this subject certainly needs to be investigated further.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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

[6] 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.

[7] 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.

[8] 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.

[9] J. Czyzyk, S. Mehrotra, M. Wagner, and S.J. Wright: *PCx: An interior-point code for linear programming.* Optimization Methods and Software 11 & 12, 1999, pp. 397–430.

[10] J. Czyzyk, S. Mehrotra, and S.J. Wright: *PCx User Guide.* Technical Report OTC 96/01, Optimization Technology Center, Argonne National Laboratory and Northwestern University, May 1996.

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

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

[13] S. Engelke and C. Kanzow: *Predictor-corrector smoothing methods for the solution of linear programs.* Preprint 153, Institute of Applied Mathematics, University of Hamburg, Hamburg, March 2000.

[14] S. Engelke and C. Kanzow: *Improved smoothing-type methods for the solution of linear programs.* Numerische Mathematik, to appear.

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

[16] K. Hotta, M. Inaba and A. Yoshise: *A complexity analysis of a smoothing method using CHKS-functions for monotone linear complementarity problems.* Computational Optimization and Applications 17, 2000, pp. 183–201.

[17] 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.

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

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

[20] 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.

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

[22] 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.): *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods.* Kluwer Academic Publishers, 1999, pp. 381–404.

[23] 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.

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