Optimization Methods & Software
Vol. 24, No. 1, February 2009, 123–143

An iterative solver-based long-step infeasible primal-dual path-following algorithm for convex QP based on a class of preconditioners

Zhaosong Lu[a]*, Renato D.C. Monteiro[b] and Jerome W. O'Neal[c]

[a]*Department of Mathematics, Simon Fraser University, Burnaby, Canada;* [b]*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA;* [c]*Columbus, OH, USA*

(*Received 28 August 2006; final version received 11 August 2008*)

In this paper, we present a long-step infeasible primal-dual path-following algorithm for convex quadratic programming (CQP) whose search directions are computed by means of a preconditioned iterative linear solver. In contrast to the authors' previous paper [Z. Lu, R.D.C. Monteiro, and J.W. O'Neal. *An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming*, SIAM J. Optim. 17(1) (2006), pp. 287–310], we propose a new linear system, which we refer to as the *hybrid augmented normal equation* (HANE), to determine the primal-dual search directions. Since the iterative linear solver can only generate an approximate solution to the HANE, this solution does not yield a primal-dual search direction satisfying all equations of the primal-dual Newton system. We propose a recipe to compute an inexact primal-dual search direction, based on a suitable approximate solution to the HANE. The second difference between this paper and [Z. Lu, R.D.C. Monteiro, and J.W. O'Neal. *An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming*, SIAM J. Optim. 17(1)(2006), pp. 287–310] is that, instead of using the maximum weight basis (MWB) preconditioner in the aforesaid recipe for constructing the inexact search direction, this paper proposes the use of any member of a whole class of preconditioners, of which the MWB preconditioner is just a special case. The proposed recipe allows us to: (i) establish a polynomial bound on the number of iterations performed by our path-following algorithm and (ii) establish a uniform bound, depending on the quality of the preconditioner, on the number of iterations performed by the iterative solver.

Keywords: convex quadratic programming; iterative linear solver; primal-dual path-following methods; interior-point methods; hybrid augmented normal equation; inexact search directions; polynomial convergence

AMS Subject Classification: 65F10; 65F35; 90C20; 90C25; 90C51

1. Introduction

In this paper, we develop a long-step infeasible primal-dual path-following (IPDPF) algorithm for solving convex quadratic programming (CQP) based on inexact search directions. The CQP

*Corresponding author. Email: zhaosong@sfu.ca

ISSN 1055-6788 print/ISSN 1029-4937 online

DOI: 10.1080/10556780802414049
http://www.informaworld.com

problem we consider has the form

$$\min_x \left\{ \tfrac{1}{2} x^T \mathbf{Q} x + c^T x : \quad Ax = b, x \geq 0 \right\}, \tag{1}$$

where the data are $\mathbf{Q} \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$, and the decision vector is $x \in \mathbb{R}^n$. We assume that $\mathbf{Q}$ is given in the form $\mathbf{Q} = VE^2V^T + Q$, where $V \in \mathbb{R}^{n\times l}$, E is a $l \times l$ positive diagonal matrix, and Q is a $n \times n$ positive semidefinite matrix.

In [15], the authors also developed an inexact IPDPF algorithm for solving Equation (1) with $\mathbf{Q}$ assumed to be given in the form $\mathbf{Q} = VE^2V^T$, or equivalently $Q = 0$. This inexact IPDPF algorithm is essentially the long-step IPDPF algorithm in [10,28], the only difference being that the search directions are computed by means of an iterative linear solver. We refer to the iterations of the iterative linear solver as the *inner iterations*, and the iterations performed by the actual path-following method as the *outer iterations*. The main step in the inexact IPDPF algorithm in [15] is the computation of a primal-dual search direction $(\Delta x, \Delta s, \Delta y, \Delta z)$, whose subvector $(\Delta y, \Delta z)$ can be found by solving the so-called *augmented normal equation*, or ANE. This ANE is of the form $\tilde{A}\tilde{D}^2\tilde{A}^T(\Delta y, \Delta z) = g$, where $\tilde{D}$ is a positive diagonal matrix, and $\tilde{A}$ is a 2×2 block matrix whose blocks consist of A, V^T, the zero matrix, and the identity matrix. In contrast to IPDPF methods based on exact search directions, the inexact IPDPF algorithm in [15] assumes that an approximate solution to the ANE is obtained via an iterative linear solver. Since the condition number of the ANE matrix may become excessively large on degenerate QP problems (see e.g. [14]), the maximum weight basis (MWB) preconditioner T introduced in [22,25,27] is used to better precondition the matrix. A suitable approximate solution can then be determined within a uniformly bounded number of iterations of an iterative linear solver. Since the ANE is solved only approximately, it cannot yield a search direction that satisfies all equations of the primal-dual Newton system. Thus, we developed a recipe in [15] for determining an inexact search direction, based on an approximate solution to the ANE and the MWB preconditioner, which accomplishes the following two goals: (i) problem (1) can be solved within a polynomial number of iterations, and (ii) the required approximate solution to the ANE can be found within a uniformly bounded number of inner iterations.

This paper extends the authors' previous work [15] in the following two ways. The first extension that we present in this paper is to introduce a new linear system, which we refer to as the *hybrid augmented normal equation* (HANE), as a means to determine the search directions for the IPDPF algorithm studied in this paper. The development of the HANE stems from the desire to take into account the structure of $\mathbf{Q}$, given by $\mathbf{Q} = VE^2V^T + Q$, in the computation of the search direction. To motivate the approach based on the HANE, we will assume in this paragraph that Q is a non-negative diagonal matrix. Consider the two extreme cases where $V = 0$ or $Q = 0$. In the first case, since $\mathbf{Q} = Q$ is diagonal, computing the search directions via the standard normal equation is appealing, since it has the same structure as the one corresponding to a linear programming problem. In the second case, the approach based on the ANE developed in [15] provides a viable alternative for computing the search direction. The approach based on the HANE combines the ideas involved in these two extreme cases in order to handle the mixed structure of $\mathbf{Q}$ as stated before. The second extension, which is the major contribution of this paper, is to show that a large class of preconditioners can be used instead of the MWB preconditioner in the recipe for determining inexact search directions proposed in [15]. In this regard, this extension will be done in the more general context of the HANE equation, rather than in the context of the ANE used by [15]. We will also discuss the situation where the preconditioned conjugate gradient method is used in conjunction with the partial update preconditioner proposed by Karmarkar in [8] (see also [6,11,18]) and derive the corresponding inner iteration complexity bound.

We observe that the use of iterative linear solvers to compute the primal-dual Newton search directions of interior-point path-following algorithms has been extensively studied in

[1,3–5,13,21–23,25]. The use of inexact search directions in interior-point methods has been investigated in the context of conic programming problems (see, e.g. [1,2,5,13,17,21,26,29]). For feasibility problems of the form $\{x \in \mathcal{H}_1 : \mathcal{A}x = b, x \in \mathcal{C}\}$, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are Hilbert spaces, $\mathcal{C} \subseteq \mathcal{H}_1$ is a closed convex cone satisfying some mild assumptions, and $\mathcal{A} : \mathcal{H}_1 \to \mathcal{H}_2$ is a continuous linear operator. Renegar [24] has proposed an interior-point method where the Newton system that determines the search directions is approximately solved by performing a uniformly bounded number of iterations of the conjugate gradient (CG) method.

Our paper is divided into five sections. In Subsection 1.1, we present some terminology and notation that will be used throughout this paper. In Section 2, we present an inexact IPDPF algorithm based on a class of inexact search directions, and we also partially describe a recipe based on the HANE for determining inexact search directions for our algorithm. In Section 3, we introduce the class of preconditioners used in a crucial step of the aforesaid recipe for constructing a vector of a required size, thereby providing the final details of the aforementioned recipe. Section 4 gives proofs of some of the results presented in Section 3. Finally, some concluding remarks are given in Section 5.

1.1 *Terminology and notation*

Throughout this paper, upper-case Roman letters denote matrices, lower-case Roman letters denote vectors, and lower-case Greek letters denote scalars. We let $\mathbb{R}^n$, $\mathbb{R}^n_+$, and $\mathbb{R}^n_{++}$ denote the set of n-vectors having real, non-negative real, and positive real components, respectively. Also, we let $\mathbb{R}^{m\times n}$ denote the set of $m \times n$ matrices with real entries, and let $\mathcal{S}^n_+$ denote the set of $n \times n$ positive semidefinite real symmetric matrices. For a vector $v \in \mathbb{R}^n$, we let $|v|$ denote the vector whose ith component is $|v_i|$, for every $i = 1, \dots, n$, and we let $\mathrm{Diag}(v)$ denote the diagonal matrix whose ith diagonal element is v_i, for every $i = 1, \dots, n$. In addition, given vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$, we denote by (u, v) the vector $(u^T, v^T)^T \in \mathbb{R}^{m+n}$.

If a matrix $Z \in \mathbb{R}^{m\times m}$ has all positive eigenvalues, we denote by $\kappa(Z)$ its spectral condition number, i.e. its maximum eigenvalue divided by its minimum eigenvalue. Given a matrix $Z \in \mathbb{R}^{m\times n}$, the range space $\{Zv : v \in \mathbb{R}^m\}$ of Z will be denoted by $\mathcal{R}(Z)$. Also, if a matrix $W \in \mathbb{R}^{m\times m}$ is symmetric ($W = W^T$) and positive definite (resp., positive semidefinite), we write $W \succ 0$ (resp., $W \succeq 0$). Certain matrices bear special mention, namely the matrices X and S. These matrices are the diagonal matrices corresponding to the vectors x and s, respectively, as described in the previous paragraph. The symbol 0 will be used to denote a scalar, vector, or matrix of all zeroes; its dimensions should be clear from the context. Also, we denote by e the vector of all 1s, and by I the identity matrix; their dimensions should be clear from the context.

We will use several different norms throughout the paper. For a vector $z \in \mathbb{R}^n$, $\|z\| = \sqrt{z^T z}$ is the Euclidean norm and $\|z\|_\infty = \max_{i=1,\dots,n} |z_i|$ is the 'infinity norm'. Also, given a matrix $C \succ 0$, we define the norm $\|z\|_C = \sqrt{z^T C z}$. Finally, given a matrix $V \in \mathbb{R}^{m\times n}$, $\|V\|$ denotes the operator norm associated with the Euclidean norm: $\|V\| = \max_{z:\|z\|=1} \|Vz\|$.

2. Outer iteration framework

In this section, we introduce an inexact IPDPF algorithm based on a class of inexact search directions and discuss its iteration complexity. This section is divided into two subsections. In Subsection 2.1, we introduce the class of inexact search directions, state the inexact IPDPF algorithm based on it, and give its iteration complexity result. In Subsection 2.2, we will discuss how the HANE naturally appears as a way of computing the exact search direction. We will also describe how an approximate solution to the HANE can be used to compute an approximate search direction for the inexact IPDPF algorithm.

2.1 *An inexact IPDPF algorithm for CQP*

Consider the following primal-dual pair of QP problems:

$$\min_{x} \left\{ \frac{1}{2} x^T \mathbf{Q} x + c^T x : \quad Ax = b,\ x \geq 0 \right\}, \tag{2}$$

$$\max_{(\hat{x},s,y)} \left\{ -\frac{1}{2} \hat{x}^T \mathbf{Q} \hat{x} + b^T y : \quad A^T y + s - \mathbf{Q}\hat{x} = c,\ s \geq 0 \right\}, \tag{3}$$

where the data are $\mathbf{Q} \in \mathcal{S}^n_+$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, and the decision variables are $x \in \mathbb{R}^n$ and $(\hat{x}, s, y) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m$. We will assume that $\mathbf{Q}$ is given in the form $\mathbf{Q} = VE^2V^T + Q$ for some $V \in \mathbb{R}^{n\times l}$, $E \in \mathrm{Diag}(\mathbb{R}^l_{++})$, and $Q \in \mathcal{S}^n_+$. In addition, we will make the following two assumptions throughout the paper.

ASSUMPTION 2.1 *$rank(A) = m < n$.*

ASSUMPTION 2.2 *the set of optimal solutions of Equations* (2) *and* (3) *is non-empty.*

It is well known that if x^* is an optimal solution for Equation (2) and $(\hat{x}^*, s^*, y^*)$ is an optimal solution for Equation (3), then (x^*, s^*, y^*) is also an optimal solution for Equation (3). Now, let $\mathcal{S}$ denote the set of all vectors $w := (x, s, y, z) \in \mathbb{R}^{2n+m+l}$ satisfying

$$Ax = b, \quad x \geq 0, \tag{4}$$

$$A^T y + s + Vz - Qx = c, \quad s \geq 0, \tag{5}$$

$$Xs = 0, \tag{6}$$

$$EV^T x + E^{-1} z = 0. \tag{7}$$

It is clear that $w \in \mathcal{S}$ if and only if x is optimal for Equation (2), (x, s, y) is optimal for Equation (3), and $z = -E^2V^Tx$. (Throughout this paper, the symbol w will always denote the quadruple (x, s, y, z), where the vectors lie in the appropriate dimensions; similarly, $\Delta w = (\Delta x, \Delta s, \Delta y, \Delta z)$, $w^k = (x^k, s^k, y^k, z^k)$, etc.)

For a point $w \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$, let us define

$$\mu := \mu(w) = \frac{x^T s}{n}, \tag{8}$$

$$r_p := r_p(w) = Ax - b, \tag{9}$$

$$r_d := r_d(w) = A^T y + s + Vz - Qx - c, \tag{10}$$

$$r_V := r_V(w) = EV^T x + E^{-1} z, \tag{11}$$

$$r := r(w) = (r_p(w), r_d(w), r_V(w)). \tag{12}$$

Given a point $u \in \mathcal{R}(Q)$, it is easy to show that the function $t^T Qt$ is constant over the manifold $\{t \in \mathbb{R}^n : Qt = u\}$. Hence, the function $|||\cdot|||_Q : \mathcal{R}(Q) \mapsto \mathbb{R}_+$ given by

$$|||u|||_Q = \sqrt{t^T Qt} \quad \text{for any } t \in \mathbb{R}^n \text{ such that } Qt = u \tag{13}$$

is well defined. The following proposition shows that this function is a norm on $\mathcal{R}(Q)$.

PROPOSITION 2.1 *Let $|||\cdot|||_Q$ be as defined in Equation* (13), *and let $u \in \mathcal{R}(Q)$. Then, the following statements hold*:

1. *Given a factorization $Q = \widetilde{V}\widetilde{V}^T$, where $\widetilde{V}$ has full column rank, we have that $|||u|||_Q = \|\mathbf{v}\|$, where $\mathbf{v}$ is the unique vector satisfying $\widetilde{V}\mathbf{v} = u$*;
2. $|||\cdot|||_Q$ *defines a norm on $\mathcal{R}(Q)$*; *and*
3. $\|u\| \leq \|Q\|^{1/2}|||u|||_Q$.

Proof Let $u \in \mathcal{R}(Q)$ be given, and let $\mathbf{v}$ be the unique vector such that $\widetilde{V}\mathbf{v} = u$. Using the assumption that $\widetilde{V}$ has full column rank, we easily see that $\mathbf{v} = \widetilde{V}^T t$ for any vector t satisfying $Qt = u$. Then the assumption that $Q = \widetilde{V}\widetilde{V}^T$ along with Equation (13) implies that

$$|||u|||_Q = \sqrt{t^T Q t} = \|\widetilde{V}^T t\| = \|\mathbf{v}\|, \tag{14}$$

and statement 1 is proven.

Since $u = \widetilde{V}\mathbf{v}$ and $\widetilde{V}$ has full column rank, it is clear that $\mathbf{v} = [\widetilde{V}^T\widetilde{V}]^{-1}\widetilde{V}^T u$. This together with statement 1 immediately implies that $|||\cdot|||_Q$ is a seminorm on $\mathcal{R}(Q)$. It is indeed a norm, since, in view of Equation (14), $|||u|||_Q = 0$ implies that $\mathbf{v} = 0$, and hence that $u = \widetilde{V}\mathbf{v} = 0$.

To prove the third statement, let t be a vector such that $Qt = u$. Then Equation (13) implies that

$$\|u\| = \|Qt\| \leq \|Q^{1/2}\|\|Q^{1/2}t\| = \|Q\|^{1/2}\sqrt{t^T Q t} = \|Q\|^{1/2}|||u|||_Q. \qquad \blacksquare$$

Next, given a point $w \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$ and scalars $\sigma \in [0, 1]$, $\tau_p > 0$, and $\tau_q > 0$, we will say that a search direction Δw is a (τ_p, τ_q)-*search direction* at w (with centrality parameter σ) if Δw satisfies

$$A\Delta x = -r_p, \tag{15}$$

$$A^T\Delta y + \Delta s + V\Delta z - Q\Delta x = -r_d - g, \tag{16}$$

$$X\Delta s + S\Delta x = -Xs + \sigma\mu e - p, \tag{17}$$

$$EV^T\Delta x + E^{-1}\Delta z = -r_V + q \tag{18}$$

for some $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$ such that

$$\|p\|_\infty \leq \tau_p\mu, \quad |||g|||_Q^2 + \|q\|^2 \leq \tau_q^2\mu, \tag{19}$$

where μ is given by Equation (8). Note that while p and q can vary over the whole Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^l$, respectively, the error g is required to be in $\mathcal{R}(Q)$.

We will now point out the relationship between the definition before and the definition of a (τ_p, τ_q)-search direction given in paper [15]. It is clear that system (32)–(35) in [15] for determining an inexact search direction can be viewed as a special case of system (15)–(18) by setting $Q = 0$, which also implies that $g = 0$ due to the fact that $g \in \mathcal{R}(Q)$. However, it is also possible to transform system (15)–(18) into a system of the form specified by Equations (32)–(35) of [15] (see the proof of Theorem 2.1 in Subsection 4.1. Hence, these two systems for defining inexact search directions are essentially equivalent. We consider system (15)–(18) in this paper because it naturally lends itself to the development of the HANE as a means to determine the search direction Δw (see Subsection 2.2).

Next, given $\eta \in [0, 1]$, $\gamma \in (0, 1)$, $\theta > 0$, and an initial point $w^0 \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$, we define the following set:

$$\mathcal{N}_{w^0}(\eta, \gamma, \theta) := \left\{ w \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l} : \begin{array}{c} Xs \geq (1-\gamma)\mu e, r_p = \eta r_p^0, \eta \leq \mu/\mu_0, \\ r_d - \eta r_d^0 \in \mathcal{R}(Q), \\ |||r_d - \eta r_d^0|||_Q^2 + \|r_V - \eta r_V^0\|^2 \leq \theta^2 \mu. \end{array} \right\}, \tag{20}$$

where $\mu = \mu(w)$, $\mu_0 = \mu(w^0)$, $r = r(w)$, and $r^0 = r(w^0)$. The central path neighbourhood used by the inexact IPDPF algorithm described next is given by

$$\mathcal{N}_{w^0}(\gamma, \theta) = \bigcup_{\eta \in [0,1]} \mathcal{N}_{w^0}(\eta, \gamma, \theta). \tag{21}$$

We are now ready to state the inexact IPDPF algorithm.

Inexact IPDPF Algorithm

(i) **Start:** *Let* $\epsilon > 0$ *and* $0 < \underline{\sigma} \leq \overline{\sigma} < 4/5$ *be given. Choose* $\gamma \in (0, 1)$, $\theta > 0$ *and* $w^0 \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$ *such that* $w^0 \in \mathcal{N}_{w^0}(\gamma, \theta)$. *Set* $k = 0$.

(ii) **While** $\mu_k := \mu(w^k) > \epsilon$ **do**

(a) *Let* $w := w^k$ *and* $\mu := \mu_k$; *choose* $\sigma := \sigma_k \in [\underline{\sigma}, \overline{\sigma}]$.

(b) *Set*

$$\tau_p = \frac{\gamma \sigma}{4} \tag{22}$$

and

$$\tau_q = \left[\sqrt{1 + (1 - 0.5\gamma)\sigma} - 1 \right] \theta. \tag{23}$$

(c) *Compute a* (τ_p, τ_q)*-search direction* $\Delta w := \Delta w^k$.

(d) *Compute* $\tilde{\alpha} := argmax\{\alpha \in [0, 1] : w + \alpha' \Delta w \in \mathcal{N}_{w^0}(\gamma, \theta), \ \forall \alpha' \in [0, \alpha]\}$.

(e) *Compute* $\bar{\alpha} := argmin\{(x + \alpha \Delta x)^T (s + \alpha \Delta s) : \alpha \in [0, \tilde{\alpha}]\}$.

(f) *Let* $w^{k+1} = w + \bar{\alpha} \Delta w$, *and set* $k \leftarrow k + 1$.

End (while).

The following result gives a bound on the number of iterations needed by the inexact IPDPF algorithm to obtain an ϵ-solution to the KKT conditions (4)–(7). Its proof will be given in Subsection 4.1.

Theorem 2.1 *Assume that the constants* $\gamma, \underline{\sigma}, \overline{\sigma}$ *and* θ *are such that*

$$\max \left\{ \gamma^{-1}, (1-\gamma)^{-1}, \underline{\sigma}^{-1}, \left(1 - \tfrac{5}{4}\overline{\sigma}\right)^{-1} \right\} = \mathcal{O}(1), \quad \theta = \mathcal{O}(\sqrt{n}), \tag{24}$$

and that the initial point $w^0 \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$ *satisfies* $(x^0, s^0) \geq (x^*, s^*)$ *for some* $w^* \in \mathcal{S}$. *Then, the inexact IPDPF algorithm generates an iterate* $w^k \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$ *satisfying* $\mu_k \leq \epsilon \mu_0$, $\|r_p^k\| \leq \epsilon \|r_p^0\|$, $\|r_d^k\| \leq \epsilon \|r_d^0\| + \epsilon^{1/2} \theta \|Q\|^{1/2} \mu_0^{1/2}$ *and* $\|r_V^k\| \leq \epsilon \|r_V^0\| + \epsilon^{1/2} \theta \mu_0^{1/2}$ *within* $\mathcal{O}\left(n^2 \log \epsilon^{-1}\right)$ *iterations.*

It is possible to show that if w^0 is a strictly feasible point, i.e. $w^0 \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$ and $r^0 = 0$, then the iteration complexity of the aforementioned algorithm is bounded by $\mathcal{O}(n \log \epsilon^{-1})$ iterations. It is also possible to develop a primal-dual short-step path-following algorithm based on the inexact search directions introduced before, which would have iteration complexity bounds $\mathcal{O}(n \log \epsilon^{-1})$ and $\mathcal{O}(\sqrt{n} \log \epsilon^{-1})$ for infeasible and feasible starting points, respectively. One interesting characteristic of the feasible algorithms discussed in this paragraph is that, although the algorithms start with a primal- and dual-feasible point w^0, the algorithms only maintain primal feasibility throughout, while the dual residuals satisfy $\|r_d\| = \mathcal{O}(\sqrt{\mu})$. For the sake of brevity, we will only deal with the long-step IPDPF algorithm stated before.

2.2 *Framework for computing an inexact search direction*

In this subsection, we will provide a framework for computing inexact search directions and give sufficient conditions for them to be (τ_p, τ_q)-search directions.

We begin by defining the following matrices:

$$D := (Q + X^{-1}S)^{-1/2}, \tag{25}$$

$$\hat{D} := \begin{pmatrix} D & 0 \\ 0 & E^{-1} \end{pmatrix} \in \mathbb{R}^{(n+l)\times(n+l)}, \tag{26}$$

$$\hat{A} := \begin{pmatrix} A & 0 \\ V^T & I \end{pmatrix} \in \mathbb{R}^{(m+l)\times(n+l)}, \tag{27}$$

$$H := \hat{A}\hat{D}^2\hat{A}^T, \tag{28}$$

and the vector

$$h := \hat{A}\begin{pmatrix} D^2(s - \sigma\mu X^{-1}e - r_d) \\ 0 \end{pmatrix} - \begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix}. \tag{29}$$

One approach to compute an exact search direction, i.e. a direction Δw satisfying (15)–(18) with $(g, p, q) = 0$, is as follows. First, we solve the following system of equations for $(\Delta y, \Delta z)$:

$$H\begin{pmatrix} \Delta y \\ \Delta z \end{pmatrix} = h.$$

This system is what we refer to as the HANE. (We observe that if $V = 0$, i.e. $\mathbf{Q} = Q$, then this system reduces to the standard normal equation for QP, while if $Q = 0$, i.e. $\mathbf{Q} = VE^2V^T$, it reduces to the ANE in [15].) Once $(\Delta y, \Delta z)$ is determined, we obtain Δx and Δs according to formulae (31) and (32) with $g = p = 0$.

Suppose now that the HANE is solved only inexactly, i.e. that the vector $(\Delta y, \Delta z)$ satisfies

$$H\begin{pmatrix} \Delta y \\ \Delta z \end{pmatrix} = h + f \tag{30}$$

for some error vector f. We then compute Δx and Δs according to the following formulae:

$$\Delta x = D^2\left(r_d + A^T\Delta y + V\Delta z - s + \sigma\mu X^{-1}e + g - X^{-1}p\right), \tag{31}$$

$$\Delta s = -r_d - A^T\Delta y + Q\Delta x - V\Delta z - g, \tag{32}$$

where the pair of correction vectors $(g, p) \in \mathcal{R}(Q) \times \mathbb{R}^n$ will be required to satisfy some conditions that we describe next. Clearly, the search direction $\Delta w = (\Delta x, \Delta s, \Delta y, \Delta z)$ computed as before satisfies Equation (16) in view of Equation (32). Moreover, Equation (17) is satisfied, since Equations (25), (31), and (32) imply that

$$\begin{aligned} X\Delta s + S\Delta x &= -Xr_d - XA^T\Delta y - XV\Delta z - Xg + (XQ + S)\Delta x \\ &= -Xr_d - XA^T\Delta y - XV\Delta z - Xg + XD^{-2}\Delta x \\ &= -Xs + \sigma\mu e - p. \end{aligned}$$

To motivate the conditions, we will impose on the pair $(g, p) \in \mathcal{R}(Q) \times \mathbb{R}^n$; we note that Equations (26)–(32) imply that

$$\begin{aligned} &\hat{A}\begin{pmatrix} \Delta x \\ E^{-2}\Delta z \end{pmatrix} + \begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix} \\ &= \hat{A}\begin{pmatrix} D^2\left((A^T\Delta y + V\Delta z) + (-s + \sigma\mu X^{-1}e + r_d) + (g - X^{-1}p)\right) \\ E^{-2}\Delta z \end{pmatrix} + \begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix} \\ &= \hat{A}\hat{D}^2\begin{pmatrix} A^T\Delta y + V\Delta z \\ \Delta z \end{pmatrix} - h - \hat{A}\begin{pmatrix} D^2(X^{-1}p - g) \\ 0 \end{pmatrix} \\ &= H\begin{pmatrix} \Delta y \\ \Delta z \end{pmatrix} - h - \hat{A}\begin{pmatrix} D^2(X^{-1}p - g) \\ 0 \end{pmatrix} = f - \hat{A}\begin{pmatrix} D^2(X^{-1}p - g) \\ 0 \end{pmatrix}. \end{aligned} \tag{33}$$

Our strategy will be to choose the pair $(g, p) \in \mathcal{R}(Q) \times \mathbb{R}^n$ so that the first component of Equation (33) is zero, and hence that Equation (15) is satisfied. Specifically, let us partition $f = (f_1, f_2) \in \mathbb{R}^m \times \mathbb{R}^l$. We will choose $(g, p) \in \mathcal{R}(Q) \times \mathbb{R}^n$ such that

$$AD^2(X^{-1}p - g) = f_1. \tag{34}$$

Observe that g and p are not uniquely defined. Letting

$$q = E\left(f_2 - V^T D^2(X^{-1}p - g)\right)$$

and using Equation (27), we easily see that Equation (34) is equivalent to

$$f = \hat{A}\begin{pmatrix} D^2(X^{-1}p - g) \\ E^{-1}q \end{pmatrix}. \tag{35}$$

Then, using Equations (27), (33), and (35), we conclude that

$$\begin{aligned} \hat{A}\begin{pmatrix} \Delta x \\ E^{-2}\Delta z \end{pmatrix} + \begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix} &= f - \hat{A}\begin{pmatrix} D^2(X^{-1}p - g) \\ E^{-1}q \end{pmatrix} + \hat{A}\begin{pmatrix} 0 \\ E^{-1}q \end{pmatrix} \\ &= \hat{A}\begin{pmatrix} 0 \\ E^{-1}q \end{pmatrix} = \begin{pmatrix} 0 \\ E^{-1}q \end{pmatrix}, \end{aligned} \tag{36}$$

from which we see that the first component of Equation (33) is set to 0 and the second component is exactly $E^{-1}q$. We have thus shown that this construction yields a search direction Δw satisfying Equations (15)–(18).

Before ending this subsection, we provide a framework for computing a triple $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$ satisfying Equation (35). First, choose a vector $v := (v_1, v_2) \in \mathbb{R}^n \times \mathbb{R}^l$

satisfying

$$\hat{A}v = f. \tag{37}$$

Next, we choose the triple $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$ according to

$$g := -Qv_1, \quad p := Sv_1, \quad q := Ev_2. \tag{38}$$

Then, Equations (25), (37), and (38) imply that

$$\hat{A}\begin{pmatrix} D^2(X^{-1}p - g) \\ E^{-1}q \end{pmatrix} = \hat{A}\begin{pmatrix} D^2(X^{-1}S + Q)v_1 \\ v_2 \end{pmatrix} = \hat{A}v = f,$$

i.e. $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$ satisfies Equation (35). Note that in view of Assumption 2.1 and Equation (27), system Equation (37) has multiple solutions. Strategies for choosing a specific vector v satisfying Equation (37) will be discussed in Subsection 3.1.

The following result relates the size of $\hat{D}^{-1}v$ with the magnitude of the triple $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$, and gives a sufficient condition for the search direction described to be a (τ_p, τ_q)-search direction.

Proposition 2.2 *Let $w \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$ be given, and consider the vector $v \in \mathbb{R}^{n+l}$ and the triple $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$ as defined in Equations* (37) *and* (38)*. Then, we have*

$$\|p\| \leq \sqrt{n\mu}\|\hat{D}^{-1}v\|, \quad |||g|||_Q^2 + \|q\|^2 \leq \|\hat{D}^{-1}v\|^2. \tag{39}$$

As a consequence, if $\|\hat{D}^{-1}v\| \leq \xi\sqrt{\mu}$, where ξ is defined as

$$\xi := \min\{n^{-1/2}\tau_p, \tau_q\}, \tag{40}$$

then the corresponding inexact search direction Δw as described before is a (τ_p, τ_q)-search direction.

Proof Using Equation (25) and the fact that $(x, s) > 0$, we conclude that $Q \preceq Q + X^{-1}S = D^{-2}$. Next, the first identity in Equation (38) along with Equation (13) implies that $|||g|||_Q^2 = v_1^T Q v_1$. Using these facts along with Equations (26) and (38), we obtain

$$|||g|||_Q^2 + \|q\|^2 = v_1^T Q v_1 + \|Ev_2\|^2 \leq v_1^T D^{-2} v_1 + \|Ev_2\|^2 = \|D^{-1}v_1\|^2 + \|Ev_2\|^2 = \|\hat{D}^{-1}v\|^2.$$

Similarly, we have $X^{-1}S \preceq D^{-2}$, which clearly implies that $D^2 \preceq XS^{-1}$. This result, along with the fact that $x_i s_i \leq n\mu$ for all i, implies that $SD^2S \preceq XS \preceq n\mu I$, and hence that $\|SD\| = \|SD^2S\|^{1/2} \leq \sqrt{n\mu}$. We use this result along with Equation (26) and the second relation in Equation (38) to obtain

$$\|p\| = \|Sv_1\| \leq \|SD\|\|D^{-1}v_1\| \leq \sqrt{n\mu}\|D^{-1}v_1\| \leq \sqrt{n\mu}\|\hat{D}^{-1}v\|.$$

Thus Equation (39) is proven. The second part of the proposition follows from the fact that Equations (39), (40), and the assumption that $\|\hat{D}^{-1}v\| \leq \xi\sqrt{\mu}$ imply that Equation (19) holds. ∎

3. Inner iteration complexity

In this section, we complete the description of the recipe given in Subsection 2.2 to determine a (τ_p, τ_q)-search direction Δw. The section is divided into two subsections. In Subsection 3.1, we derive a uniform upper bound on the number of iterations that a generic iterative linear solver requires to obtain a sufficiently accurate solution $(\Delta y, \Delta z)$ to the HANE, which will then yield a (τ_p, τ_q)-search direction Δw, as required in step 2(d) of the inexact IPDPF algorithm. One of the key ideas in this paper, which is described in Subsection 2.1, is the use of a suitable approximation F of $\hat{D}^2$ to define the vector v as a linear function of u. In Subsection 2.2, we present two examples of matrices F that are suitable approximations of $\hat{D}^2$. We also obtain specific expressions for the iteration complexity developed in Subsection 2.1 when the iterative solver used to obtain an approximate solution to the HANE is the preconditioned conjugate gradient (PCG) method with preconditioner given by $\hat{A}F\hat{A}^T$.

3.1 *Inner iteration complexity analysis*

In this subsection, we will complete the description of the recipe given in Subsection 2.2 to determine a (τ_p, τ_q)-search direction Δw. For simplicity of notation, in this section we will denote the variable of unknowns in the HANE by u, so that the HANE takes the form $Hu = h$, where H and h are given by Equations (28) and (29), respectively. Recall that the only thing that was unspecified in the recipe of Subsection 2.2 was the choice of a vector v satisfying Equation (37). Recall also from Lemma 2.2 that by choosing v such that $\|\hat{D}^{-1}v\| \leq \xi\sqrt{\mu}$, where ξ is given by Equation (40), the corresponding inexact search direction Δw is guaranteed to be a (τ_p, τ_q)-search direction, simply by choosing (g, p, q) according to Equation (38). One of the key ideas in this paper, which is described in this subsection, is the use of a generic preconditioner for H to define the vector v as a linear function of u. This subsection also discusses the iteration complexity of a generic iterative solver to obtain an iterate u so that the corresponding $v = v(u)$ satisfies the condition $\|\hat{D}^{-1}v\| \leq \xi\sqrt{\mu}$. We also discuss an appropriate choice of the starting point u^0 and conditions on the generic preconditioner for H, which guarantee that the inner iteration complexity bound is uniformly bounded throughout the iterations of the inexact IPDPF algorithm.

We will first discuss the criterion we use to measure the complexity of an iterative solver to obtain an approximate solution to a system of the form $Hu = h$. A common way of measuring the closeness of u to $u^* := H^{-1}h$ is by the distance $\|u - u^*\|_H = \|f(u)\|_{H^{-1}}$, where $f(u) := Hu - h$. Many algorithms for solving the system $Hu = h$ produce a sequence of iterates that decrease this distance at every step (see [7,9,16]). Other equivalent distances could be used in our following discussion, but we will only consider the previous one without any loss of generality. We will say that the complexity of an iterative solver (with respect to the earlier distance) is bounded above by a non-decreasing function $\Upsilon : [1, \infty) \mapsto \mathbb{Z}_+$ if, for any $\delta \geq 1$, $\Upsilon(\delta)$ denotes an upper bound on the number of iterations required by the iterative solver, started at any u^0, to obtain an iterate u such that $\|f(u)\|_{H^{-1}} \leq \delta^{-1}\|f(u^0)\|_{H^{-1}}$.

Next, we will discuss a way of choosing a vector v satisfying Equation (37) and the condition

$$\|\hat{D}^{-1}v\| \leq K\|f(u)\|_{H^{-1}} \tag{41}$$

for some suitable constant $K \geq 1$. For fixed $f(u)$, consider the ideal case for which we set $v = v_{LS}$, where $v_{LS} = \operatorname{argmin}\{\|\hat{D}^{-1}v\| : \hat{A}v = f(u)\}$. It is straightforward to show that

$$v_{LS} = \hat{D}^2\hat{A}^T H^{-1} f(u) = \hat{D}^2\hat{A}^T(\hat{A}\hat{D}^2\hat{A}^T)^{-1} f(u), \tag{42}$$

where H is given by Equation (28). Thus, we have that

$$\|\hat{D}^{-1}v_{LS}\| = \sqrt{f(u)^T(\hat{A}\hat{D}^2\hat{A}^T)^{-1}f(u)} = \|f(u)\|_{H^{-1}}, \quad (43)$$

and hence Equation (41) is satisfied with $K = 1$. Unfortunately, the computation of v_{LS} requires the computation of $H^{-1}f(u)$, or equivalently the solution of a system of linear equations with the same coefficient matrix as the HANE we are trying to solve. To remedy this problem, we will approximate $\hat{D}^2$ by a matrix $F \succeq 0$ such that $G := \hat{A}F\hat{A}^T \succ 0$ and $G^{-1}f(u)$ is much cheaper to compute than $H^{-1}f(u)$. We then replace $\hat{D}^2$ in Equation (42) by F to obtain a vector v according to

$$v := v(F, u) = F\hat{A}^T G^{-1} f(u) = F\hat{A}^T(\hat{A}F\hat{A}^T)^{-1}f(u). \quad (44)$$

It is clear that v defined in this manner satisfies Equation (37). By imposing some conditions on the approximation F according to the following definition, v will also satisfy Equation (41) for some constant $K \geq 1$. We will require that F approximate $\hat{D}^2$ in the following sense.

Definition 3.1 *Let constants* $0 < \lambda_L \leq \lambda_U$ *be given. We will say that a matrix* F *is a* (λ_L, λ_U)*-approximation of* $\hat{D}^2$ *if* $0 \preceq F \preceq \lambda_U \hat{D}^2$ *and* $\hat{A}F\hat{A}^T \succeq \lambda_L \hat{A}\hat{D}^2\hat{A}^T$.

Using this definition, we can now state the following result.

Lemma 3.1 *Suppose that a matrix* F *is a* (λ_L, λ_U)*-approximation of* $\hat{D}^2$*. Then the vector* v *given by Equation* (44) *satisfies Equation* (41) *with* $K = \sqrt{\lambda_U/\lambda_L}$.

Proof Recall that $G = \hat{A}F\hat{A}^T$, and recall the definition of H in Equation (28). Using the assumption that F is a (λ_L, λ_U)-approximation of $\hat{D}^2$ and Definition 3.1, we have that $G^{-1} \preceq \lambda_L^{-1}H^{-1}$ and $\hat{D}^{-1}F\hat{D}^{-1} \preceq \lambda_U I$. Using these inequalities along with Equation (44), we conclude that

$$\begin{aligned}
\|\hat{D}^{-1}v\| &\leq \|\hat{D}^{-1}F^{1/2}\|\|F^{1/2}\hat{A}^T G^{-1}f(u)\| \\
&= \|\hat{D}^{-1}F^{1/2}\|\sqrt{f(u)^T G^{-1}(\hat{A}F\hat{A}^T)G^{-1}f(u)} \\
&= \|\hat{D}^{-1}F\hat{D}^{-1}\|^{1/2}\sqrt{f(u)^T G^{-1}f(u)} \leq \sqrt{\frac{\lambda_U}{\lambda_L}}\sqrt{f(u)^T H^{-1}f(u)} \\
&= \sqrt{\frac{\lambda_U}{\lambda_L}}\|f(u)\|_{H^{-1}}.
\end{aligned}$$

■

Note that if u is a point such that $\|f(u)\|_{H^{-1}} \leq \delta^{-1}\|f(u^0)\|_{H^{-1}}$, and if v is formed according to Equation (44), where F is a (λ_L, λ_U)-approximation of $\hat{D}^2$, we have

$$\frac{\|\hat{D}^{-1}v\|}{\|f(u^0)\|_{H^{-1}}} \leq \sqrt{\frac{\lambda_U}{\lambda_L}}\frac{\|f(u)\|_{H^{-1}}}{\|f(u^0)\|_{H^{-1}}} \leq \delta^{-1}\sqrt{\frac{\lambda_U}{\lambda_L}} \quad (45)$$

in view of Lemma 3.1. The issues to be considered now are: (i) the choice of the starting point u^0 and (ii) the choice of δ. Regarding (i), we will show that a starting point u^0 can always be chosen so that

$$\|f(u^0)\|_{H^{-1}} \le \Psi\sqrt{\mu} \tag{46}$$

for some universal constant Ψ. Assuming this fact, the constant δ in issue (ii) can be chosen as

$$\delta = \left(\frac{\Psi}{\xi}\right)\sqrt{\frac{\lambda_U}{\lambda_L}}, \tag{47}$$

where ξ is given by Equation (40). Indeed, by Equations (45)–(47), it follows that the resulting vector v satisfies $\|\hat{D}^{-1}v\| \le \xi\sqrt{\mu}$, as desired.

We will now concentrate our attention on the construction of a starting point u^0 satisfying Equation (46). First, compute a point $w' = (x', y', s', z')$ satisfying the following system of linear equations:

$$\hat{A}\begin{pmatrix} x' \\ E^{-2}z' \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \quad A^T y' + s' + Vz' - Qx' = c. \tag{48}$$

We then define

$$u^0 = -\eta\begin{pmatrix} y^0 - y' \\ z^0 - z' \end{pmatrix}, \tag{49}$$

where $\eta = \|r_p\|/\|r_p^0\|$. Notice that all of the starting points generated by the above formulae are multiples of the same vector, which can be computed once at the beginning of the inexact IPDPF algorithm. Moreover, if the starting point w^0 of the algorithm is feasible to Equations (2) and (3), then we may choose $w' = w^0$, and hence $u^0 = 0$. The following lemma gives a bound on $\|f(u^0)\|_{H^{-1}}$.

LEMMA 3.2 *Assume that w^0 and w' are such that $(x^0, s^0) \ge |(x', s')|$ and $(x^0, s^0) \ge (x^*, s^*)$ for some $w^* \in \mathcal{S}$. Further, assume that $w \in \mathcal{N}_{w^0}(\gamma, \theta)$ for some $\gamma \in (0, 1)$ and $\theta > 0$, and that H, h, and u^0 are given by Equations* (28)*,* (30)*, and* (49)*, respectively. Then, $f(u^0)$ satisfies Equation* (46)*, where μ is given by Equation* (8) *and Ψ is defined as*

$$\Psi := \frac{6}{\sqrt{1-\gamma}}n + \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)^{1/2}\sqrt{n} + \frac{\theta^2}{2\sqrt{1-\gamma}} + \theta. \tag{50}$$

The proof of this lemma will be given in Subsection 4.2. Our next lemma provides insight into the size of the ratio Ψ/ξ in Equation (47).

LEMMA 3.3 *Suppose that* $\max\{\sigma, \sigma^{-1}, \gamma^{-1}, (1-\gamma)^{-1}, \theta^{-1}\} = \mathcal{O}(1)$ *and* $\theta = \mathcal{O}(\sqrt{n})$ *in the inexact IPDPF algorithm, and that τ_p, τ_q, ξ, and Ψ are as defined in Equations* (22), (23), (40), *and* (50), *respectively. Then, we have that* $\Psi/\xi = \mathcal{O}(n^{3/2})$.

Proof Under the assumptions shown, it is easy to see that $\Psi = \mathcal{O}(n)$ and $\xi^{-1} = \mathcal{O}(\sqrt{n})$, and the result follows immediately. ■

We summarize the results of this subsection in the following theorem.

THEOREM 3.1 *Suppose that the conditions of Lemmas* 3.1–3.3 *are met. Then, an iterative solver with complexity bounded by $\Upsilon(\cdot)$ generates an iterate u such that $v = v(F, u)$ satisfies $\|\hat{D}^{-1}v\| \le \xi\sqrt{\mu}$ in at most*

$$\Upsilon\left(\mathcal{O}\left(n^{3/2}\sqrt{\lambda_U/\lambda_L}\right)\right)$$

iterations.

It is important to observe that, although the requirements given in this subsection are sufficient to ensure that the resulting Δw is a (τ_p, τ_q)-search direction, they are not necessary. Indeed, it is only necessary to check the sizes of $\|p\|_\infty$ and $|||g|||_Q^2 + \|q\|^2$ to ensure that the resulting Δw is a (τ_p, τ_q)-search direction. Once a candidate vector v is generated, then $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$ (and their corresponding magnitudes) can be easily computed according to Equation (38).

3.2 *Specific applications*

In this subsection, we present two examples of matrices F, which are (λ_L, λ_U)-approximations of $\hat{D}^2$, and an estimation of their corresponding constants λ_L and λ_U. As a consequence, we will obtain specific expressions for the iteration complexity developed in Theorem 3.1 when the iterative solver used to obtain an approximate solution to the HANE is the PCG method with preconditioner given by $\hat{A}F\hat{A}^T$.

The first example of a matrix F we will consider in this subsection is the MWB preconditioner originally proposed by Vaidya [27] (see also [25]). For the purposes of this example only, we will assume that Q is diagonal, which clearly implies that $\hat{D}$ is also diagonal. MWB is a basis B of $\hat{A}$ formed by giving higher priority to columns of $\hat{A}$ corresponding to larger diagonal elements of $\hat{D}$. The MWB preconditioner is then given by $\hat{T}^{-1}\hat{T}^{-T}$, where $\hat{T} = \hat{D}_{\mathcal{B}}^{-1}B^{-1}$ and $\hat{D}_{\mathcal{B}}$ is the diagonal submatrix of $\hat{D}$ corresponding to the columns of B. (See [20] for a complete description of the MWB.) Note that this preconditioner can be written as

$$G = B\hat{D}_{\mathcal{B}}^2 B^T = \hat{A}\begin{pmatrix} \hat{D}_{\mathcal{B}}^2 & 0 \\ 0 & 0 \end{pmatrix}\hat{A}^T = \hat{A}F\hat{A}^T,$$

where

$$F = \begin{pmatrix} \hat{D}_{\mathcal{B}}^2 & 0 \\ 0 & 0 \end{pmatrix}.$$

It is clear from this definition that $0 \preceq F \preceq \hat{D}^2$. Next, Lemma 2.1 in [20] implies that $\|\hat{T}\hat{A}\hat{D}\| \leq \varphi_{\hat{A}}$, where $\varphi_{\hat{A}}$ is defined as

$$\varphi_{\hat{A}} := \max\{\|B^{-1}\hat{A}\|_F : B \text{is a basis for} \hat{A}\}.$$

It follows that $\hat{T}H\hat{T}^T = \hat{T}(\hat{A}\hat{D}^2\hat{A}^T)\hat{T}^T \preceq \varphi_{\hat{A}}^2 I$, which implies that $G \succeq \varphi_{\hat{A}}^{-2}H$. In view of Definition 3.1, we have thus shown that F is a $(\varphi_{\hat{A}}^{-2}, 1)$-approximation of $\hat{D}^2$.

Another way of obtaining an approximation of $\hat{D}^2$ is by using the partial updating strategy that was first proposed by Karmarkar [8] (see also Gonzaga [6]) in the context of primal-only interior-point methods, and extended by Monteiro and Adler [18] and Kojima *et al.* [11] to the context of primal-dual path-following methods. At each iteration of a path-following algorithm, the strategy consists of generating a diagonal matrix $\bar{D}$ satisfying

$$\rho^{-1}\frac{s_i}{x_i} \leq \bar{D}_{ii} \leq \rho\frac{s_i}{x_i}, \quad \text{for all } i = 1, \ldots, n \tag{51}$$

for some constant $\rho \geq 1$, and using

$$F := \begin{pmatrix} (Q + \bar{D})^{-1} & 0 \\ 0 & E^{-2} \end{pmatrix} \tag{52}$$

as the approximation for $\hat{D}^2$. The current approximation $\bar{D}$ is obtained by updating the approximation used at the previous iterate in the following manner. If the ith diagonal element of $\bar{D}$

used at the previous iterate violates Equation (51), then it is changed to s_i/x_i; otherwise it is left unchanged. Using Equations (25), (26), (51), and (52), we easily see that $\rho^{-1}\hat{D}^2 \preceq F \preceq \rho\hat{D}^2$, which implies that $G = \hat{A}F\hat{A}^T \succeq \rho^{-1}H$. Hence, F is a (ρ^{-1}, ρ)-approximation of $\hat{D}^2$.

In the remainder of this subsection, we will obtain specific expressions for the iteration complexity developed in Theorem 3.1 when the iterative solver used to obtain an approximate solution to the HANE is the PCG method with preconditioner $\hat{A}F\hat{A}^T$, where F is obtained via the MWB and partial update methods, respectively. It should be noted that under exact arithmetic, the PCG algorithm is in fact a finite termination algorithm, achieving an exact solution to the HANE in at most $m + l$ iterations, since $H \in \mathcal{S}^{m+l}_{++}$ (see, e.g. [9,16]). For our purposes, we will view the PCG method as an iterative method, which is known to satisfy the following convergence property: if $G \in \mathcal{S}^{m+l}_{++}$ is used as a preconditioner for the HANE, then the method obtains an iterate u such that $\|f(u)\|_{H^{-1}} \le \delta^{-1}\|f(u^0)\|_{H^{-1}}$ in at most

$$\Upsilon(\delta) = \mathcal{O}\left\{\sqrt{\kappa(G^{-1}H)}\log\delta\right\} \tag{53}$$

iterations, where we recall that $\kappa(\cdot)$ represents the spectral condition number of $(\cdot)$. The following lemma gives a bound on the spectral condition number of $G^{-1}H$ when $G = \hat{A}F\hat{A}^T$ and F is a (λ_L, λ_U)-approximation of $\hat{D}^2$.

LEMMA 3.4 *Suppose that F is a (λ_L, λ_U)-approximation of $\hat{D}^2$, and define $G := \hat{A}F\hat{A}^T$. Then, $\kappa(G^{-1}H) \le \lambda_U/\lambda_L$.*

Proof Let L be an invertible matrix such that $LL^T = G^{-1}$. We observe that $G^{-1}H$ and L^THL are similar, and hence $\kappa(L^THL) = \kappa(G^{-1}H)$. Since F is a (λ_L, λ_U)-approximation of $\hat{D}^2$, we have that $F \preceq \lambda_U\hat{D}^2$ and $G \succeq \lambda_L H$. These relations, along with (28) and the definition of G, imply that $\lambda_L H \preceq G \preceq \lambda_U H$. This observation together with the fact that $G = L^{-T}L^{-1}$ then implies that $\lambda_U^{-1}I \preceq L^THL \preceq \lambda_L^{-1}I$, and hence that $\kappa(G^{-1}H) = \kappa(L^THL) \le \lambda_U/\lambda_L$. ∎

Using Lemma 3.4 along with Equation (53), we see that Theorem 3.1 yields the iteration complexity bound

$$\mathcal{O}\left\{\sqrt{\frac{\lambda_U}{\lambda_L}}\log\left(n\frac{\lambda_U}{\lambda_L}\right)\right\} \tag{54}$$

for the PCG method with preconditioner $G = \hat{A}F\hat{A}^T$, where F is a (λ_L, λ_U)-approximation of $\hat{D}^2$. For the MWB and partial update preconditioners, this bound becomes $\mathcal{O}(\varphi_{\hat{A}}\log(n\varphi_{\hat{A}}))$ and $\mathcal{O}(\rho\log(n\rho))$ iterations, since the respective matrices F are $(\varphi_{\hat{A}}^{-2}, 1)$- and (ρ^{-1}, ρ)-approximations of $\hat{D}^2$, respectively. We observe that the resulting bound for the MWB preconditioner is precisely the same as the one obtained in [15].

In the remaining part of this subsection, we will make a few observations about the inner iteration complexity bound (Equation (54)). As mentioned in Subsection 2.1, it is possible to develop a short-step method based on the inexact search directions introduced in Subsection 2.1. When this method is started from a feasible point, it can then be shown that the inner iteration complexity bound is the same as Equation (54), but with the factor n removed from the logarithm. Recall that the term $\log n$ in Equation (54) follows from the fact that the ratio Ψ/ξ in Lemma 3.3 is $\mathcal{O}(n^{3/2})$, which in turn follows from the fact that Ψ in Lemma 3.2 and ξ^{-1} in Equation (40) satisfy $\Psi = \mathcal{O}(n)$ and $\xi^{-1} = \mathcal{O}(\sqrt{n})$. In the context of a short-step feasible method, it is possible to show that for an appropriate choice of σ, γ, and θ, $\Psi = \mathcal{O}(1)$ and $\xi^{-1} = \mathcal{O}(1)$. The latter follows from the fact that the bound derived in Equation (39) for $\|p\|$ can be reduced by a factor of $\mathcal{O}(\sqrt{n})$, and hence that ξ can be chosen as $\Theta(\min\{\tau_p, \tau_q\})$.

In view of the discussion in the previous paragraph, the short-step variant of the inexact IPDPF algorithm, started from a feasible point, has inner iteration complexity bound $\mathcal{O}(\rho \log \rho)$ if the partial update preconditioner is used to solve the HANE. It is interesting to compare this bound with the inner iteration complexity bound of the inexact path-following method presented by Anstreicher in [1]. His paper presents a short-step, dual-only, path-following method with feasible starting point, where the normal equation is solved by the PCG method using the partial update preconditioner. It shows that the outer and inner complexity bounds are $\mathcal{O}(\sqrt{n} \log \epsilon^{-1})$ and $\mathcal{O}(\rho)$ iterations, respectively. In order to minimize the overall arithmetic complexity of his method, including the work of updating the preconditioner through a series of rank-one updates, Anstreicher [1] shows that the best choice for ρ is $\rho = \mathcal{O}(m^\beta)$ for some $\beta \in (0, 1/2)$, which yields the optimal arithmetic complexity of $\mathcal{O}((n^3/\log n) \log \epsilon^{-1})$.

Note that the inner iteration complexity bound in [1] is a factor of $\log \rho = \mathcal{O}(\log(\lambda_U/\lambda_L))$ better than the same bound in our method. The main reason for this difference is that, while Anstreicher's [1] method generates an iterate u satisfying

$$\frac{\|f(u)\|_{H^{-1}}}{\|f(u^0)\|_{H^{-1}}} \leq \delta^{-1}, \tag{55}$$

where $\delta = \mathcal{O}(1)$, our method generates an iterate u such that $\|\hat{D}^{-1} v(F, u)\|/(\xi\sqrt{\mu}) \leq 1$. Noting that Lemmas 3.1 and 3.2 and inequality Equation (46) imply that

$$\frac{\|\hat{D}^{-1} v(F, u)\|}{\xi\sqrt{\mu}} = \frac{\Psi}{\xi} \cdot \frac{\|\hat{D}^{-1} v(F, u)\|}{\Psi\sqrt{\mu}} \leq \frac{K\Psi}{\xi} \frac{\|f(u)\|_{H^{-1}}}{\|f(u^0)\|_{H^{-1}}},$$

where $K = \sqrt{\lambda_U/\lambda_L}$, our requirement on the iterate u can be accomplished by enforcing Equation (55) with $\delta = K\Psi/\xi$. Since, for a short-step method with a feasible starting point, we have that this choice of δ satisfies $\delta = \mathcal{O}(\rho)$, it follows that our inner iteration complexity has an additional $\log \delta = \mathcal{O}(\log \rho)$ factor compared with the complexity of [1]. Note that if the ideal choice of $v = v_{LS}$ given by Equation (42) is made, then $K = 1$ in view of Equation (43) and $\delta = \mathcal{O}(1)$. Then we would have an inner iteration complexity bound of $\mathcal{O}(\rho)$, the same as in [1]. Hence, the dual-only method in [1] can be thought of as being comparable, in terms of the number of inner iterations, with the inexact IPDPF algorithm proposed in this paper, with this ideal (but expensive) choice of inexact search direction. Note that, since the left-hand side of Equation (55) cannot be computed, and hence cannot be used to check for early termination of the PCG method, exactly $\Upsilon(\delta)$ iterations of the PCG method must be performed at each outer iteration of Anstreicher's [1] algorithm, where $\Upsilon(\delta)$ is given by Equation (53). In this respect, our approach is preferable to the one in [1], since it has a measurable termination criterion, namely $\|\hat{D}^{-1} v(F, u)\|/(\xi\sqrt{\mu}) \leq 1$. It is possible to incorporate a measurable stopping criterion into Anstreicher's [1] approach, but in that case, the resulting inner iteration complexity bound would increase to $\mathcal{O}(\rho \log \rho)$, the same bound as in our method.

4. Technical results

In this subsection, we present the proofs of Theorem 2.1 and Lemma 3.2. Subsection 4.1 presents the proof of Theorem 2.3, while Subsection 4.2 gives the proof of Lemma 3.2.

4.1 *Proof of Theorem 2.1*

In this subsection, we prove Theorem 2.1 by showing that the inexact IPDPF algorithm of Subsection 2.1 is completely equivalent to the algorithm presented in [15], and hence has similar convergence properties as the latter one.

Proof of Theorem 2.1 Let $\widetilde{V} \in \mathbb{R}^{n\times\tilde{l}}$ be a matrix of full column rank such that $Q = \widetilde{V}\widetilde{V}^T$. It is clear that we may write $\mathbf{Q} = \mathbf{V}\mathbf{E}^2\mathbf{V}^T$, where

$$\mathbf{V} := \left(V\widetilde{V}\right), \quad \mathbf{E} := \begin{pmatrix} E & 0 \\ 0 & I \end{pmatrix}.$$

Note that $\mathbf{Q}$ has the form required for the inexact IPDPF algorithm in [15]. Recall that the algorithm in [15] generates a sequence of iterates $\mathbf{w}^k = (x^k, s^k, y^k, (z^k, \tilde{z}^k))$ to approximate a solution of the equivalent reformulation of the optimality conditions (4)–(7):

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + s + Vz + \widetilde{V}\tilde{z} &= c, \quad s \geq 0, \\ Xs &= 0, \\ EV^T x + E^{-1} z &= 0, \\ \widetilde{V}^T x + \tilde{z} &= 0. \end{aligned}$$

More specifically, the algorithm in [15] generates a sequence of points $\mathbf{w}^k$, which lie in the neighbourhood $\mathbf{N}_{\mathbf{w}^0}(\gamma, \theta) := \cup_{\eta\in[0,1]}\mathbf{N}_{\mathbf{w}^0}(\eta, \gamma, \theta)$, where

$$\mathbf{N}_{\mathbf{w}^0}(\eta, \gamma, \theta) := \left\{ \mathbf{w} \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l+\tilde{l}} : \begin{array}{c} Xs \geq (1-\gamma)\mu e, (r_p, \mathbf{r}_d) = \eta(r_p^0, \mathbf{r}_d^0), \eta \leq \dfrac{\mu}{\mu_0}, \\ \|r_V - \eta r_V^0\|^2 + \|r_{\widetilde{V}} - \eta r_{\widetilde{V}}^0\|^2 \leq \theta^2\mu \end{array} \right\},$$

and the residuals $\mathbf{r}_d$ and $r_{\widetilde{V}}$ are defined as

$$\begin{aligned} \mathbf{r}_d &:= A^T y + s + Vz + \widetilde{V}\tilde{z} - c, \\ r_{\widetilde{V}} &:= \widetilde{V}^T x + \tilde{z}. \end{aligned}$$

Given a point $\mathbf{w} \in \mathbf{N}_{\mathbf{w}^0}(\gamma, \theta)$, the inexact algorithm in [15] generates a (τ_p, τ_q)-search direction $\mathbf{\Delta w} = (\Delta x, \Delta s, \Delta y, (\Delta z, \Delta\tilde{z}))$, which in that context means a search direction satisfying

$$\begin{aligned} A\Delta x &= -r_p, \\ A^T\Delta y + \Delta s + V\Delta z + \widetilde{V}\Delta\tilde{z} &= -\mathbf{r}_d, \\ X\Delta s + S\Delta x &= -Xs + \sigma\mu e - p, \\ EV^T\Delta x + E^{-1}\Delta z &= -r_V + q, \\ \widetilde{V}^T\Delta x + \Delta\tilde{z} &= -r_{\widetilde{V}} + \tilde{q}, \end{aligned}$$

for some vectors p, q, and $\tilde{q}$ satisfying $\|p\|_\infty \leq \tau_p\mu$ and $\|(q, \tilde{q})\| \leq \tau_q\sqrt{\mu}$, where τ_p and τ_q are defined in Equations (22) and (23), respectively. The inexact IPDPF algorithm in [15] determines a stepsize α in exactly the same manner as steps (d) and (e) of the inexact algorithm in Subsection 2.1, but with w, Δw, and $\mathcal{N}_{w^0}(\gamma, \theta)$ replaced by $\mathbf{w}$, $\mathbf{\Delta w}$, and $\mathbf{N}_{\mathbf{w}^0}(\gamma, \theta)$, respectively, and determines the next iterate $\mathbf{w}^+$ according to $\mathbf{w}^+ = \mathbf{w} + \alpha\mathbf{\Delta w}$.

It is straightforward to show that the inexact IPDPF algorithm in Subsection 2.1, started at w^0, is completely equivalent to the inexact IPDPF algorithm in [15], started at $\mathbf{w}^0 = (x^0, s^0, y^0, (z^0, \tilde{z}^0))$, where $\tilde{z}^0 = -\widetilde{V}^T x^0$, due to the following claims.

1. A vector $w = (x, s, y, z) \in \mathcal{N}_{w^0}(\eta, \gamma, \theta)$ if and only if there exists a vector $\tilde{z}$ such that $\mathbf{w} = (x, s, y, (z, \tilde{z})) \in \mathbf{N}_{\mathbf{w}^0}(\eta, \gamma, \theta)$, in which case $\tilde{z}$ is unique.
2. If w and $\mathbf{w}$ are related as in statement 1 here, a search direction $\Delta w = (\Delta x, \Delta s, \Delta y, \Delta z)$ is a (τ_p, τ_q)-search direction at w if and only if there exists a vector $\Delta\tilde{z}$ such that the search direction $\mathbf{\Delta w} = (\Delta x, \Delta s, \Delta y, (\Delta z, \Delta\tilde{z}))$ is a (τ_p, τ_q)-search direction at $\mathbf{w}$ (in the sense of [15]), in which case $\Delta\tilde{z}$ is unique.

The proofs of claims 1 and 2 are based on the following observations, which are valid under the assumption that $\tilde{z}^0 = -\widetilde{V}^T x^0$, or equivalently $r^0_{\widetilde{V}} = 0$.

- If $w \in \mathcal{N}_{w^0}(\eta, \gamma, \theta)$, let t be the unique vector such that $\widetilde{V} t = r_d - \eta r^0_d$, and define $\tilde{z} = -\widetilde{V}^T x - t$. Then $\mathbf{w} \in \mathbf{N}_{\mathbf{w}^0}(\eta, \gamma, \theta)$.
- If $\mathbf{w} \in \mathbf{N}_{\mathbf{w}^0}(\eta, \gamma, \theta)$, then we have that $\mathbf{r}_d = \eta \mathbf{r}^0_d = \eta r^0_d = r_d + \widetilde{V} r_{\widetilde{V}}$. Thus $r_d - \eta r^0_d \in \mathcal{R}(Q)$, and statement 1 of Proposition 2.1 and the fact that $r^0_{\widetilde{V}} = 0$ imply that $|||r_d - \eta r^0_d|||_Q = \|r_{\widetilde{V}}\| = \|r_{\widetilde{V}} - \eta r^0_{\widetilde{V}}\|$. It follows that $w \in \mathcal{N}_{w^0}(\eta, \gamma, \theta)$.
- Let Δw be a (τ_p, τ_q)-search direction with error terms $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$, let $\tilde{q}$ be the unique vector such that $\widetilde{V}\tilde{q} = g$, and let $\Delta\tilde{z}$ be given by $\Delta\tilde{z} = -\widetilde{V}^T \Delta x - r_{\widetilde{V}} + \tilde{q}$. Then $\mathbf{\Delta w}$ is a (τ_p, τ_q)-search direction at $\mathbf{w}$ with error terms $(p, (q, \tilde{q}))$.
- Let $\mathbf{\Delta w}$ be a (τ_p, τ_q)-search direction at $\mathbf{w}$ with error terms $(p, (q, \tilde{q}))$, and let $g = \widetilde{V}\tilde{q}$. It follows that Δw is a (τ_p, τ_q)-search direction with error terms $(g, p, q) \in \mathcal{R}(Q) \times \mathbb{R}^n \times \mathbb{R}^l$.

We leave a detailed proof of claims 1 and 2 to the reader.

Given $\epsilon > 0$, Theorem 2.2 of [15] claims that the inexact algorithm in [15] finds a point $\mathbf{w}^k \in \mathbf{N}_{\mathbf{w}^0}(\gamma, \theta)$ satisfying $\mu_k \le \epsilon\mu_0$ in at most $\mathcal{O}(n^2 \log \epsilon^{-1})$ iterations. Translated to the inexact IPDPF algorithm in Subsection 2.1, this means that a point $w^k \in \mathcal{N}_{w^0}(\gamma, \theta)$ satisfying $\mu_k \le \epsilon\mu_0$ can be found in at most $\mathcal{O}(n^2 \log \epsilon^{-1})$ iterations. The remaining conditions on w^k in our theorem follow from the definition of $\mathcal{N}_{w^0}(\gamma, \theta)$ in Equation (21), the fact that $\mu_k \le \epsilon\mu_0$, and statement 3 of Proposition 2.1. ■

4.2 *Proof of Lemma 3.2*

In this subsection, we present the proof of Lemma 3.2. We first present some technical lemmas.

LEMMA 4.1 *Suppose that $w^0 \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$ such that $(x^0, s^0) \ge (x^*, s^*)$ for some $w^* \in \mathcal{S}$. Then, for any $w \in \mathcal{N}_{w^0}(\eta, \gamma, \theta)$ with $\eta \in [0, 1]$, $\gamma \in (0, 1)$, and $\theta > 0$, we have*

$$\eta(x^T s^0 + s^T x^0) \le \left(3n + \frac{\theta^2}{4}\right)\mu.$$

Proof Recall from Subsection 4.1 that any point $w \in \mathcal{N}_{w^0}(\eta, \gamma, \theta)$ can be mapped into a point $\mathbf{w} \in \mathbf{N}_{\mathbf{w}^0}(\eta, \gamma, \theta)$, such that the x and s components of w and $\mathbf{w}$ are precisely the same. The result now follows by applying Lemma 4.1 of [15] to $\mathbf{w}$. ■

LEMMA 4.2 *Let H be defined as in Equation (28), and suppose that $(x, s, y, z) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$. Then, for any $w \in \mathbb{R}^{n+l}$ we have that $\|\hat{A}\hat{D}w\|_{H^{-1}} \le \|w\|$.*

Proof Observe that $\hat{D}\hat{A}^T H^{-1}\hat{A}\hat{D}$ is a projection matrix, which implies that $\hat{D}\hat{A}^T H^{-1}\hat{A}\hat{D} \preceq I$. Thus, for any $w \in \mathbb{R}^{n+l}$ we have that

$$\|\hat{A}\hat{D}w\|_{H^{-1}} = \sqrt{w^T(\hat{D}\hat{A}^T H^{-1}\hat{A}\hat{D})w} \le \sqrt{w^T w} = \|w\|. \qquad \blacksquare$$

For the purpose of the next proof, let us define

$$J(\sigma) := -(XS)^{1/2}e + \sigma\mu(XS)^{-1/2}e. \tag{56}$$

LEMMA 4.3 *Suppose* $w^0 \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^{m+l}$, $w \in \mathcal{N}_{w^0}(\eta, \gamma, \theta)$ *for some* $\eta \in [0, 1]$, $\gamma \in (0, 1)$, *and* $\theta > 0$, *and* w' *satisfies Equation* (48). *Let* H, h, *and* u^0 *be given by Equations* (28), (30), *and* (49), *respectively. Then,*

$$Hu^0 - h = \hat{A}\hat{D}\begin{pmatrix} DX^{-1/2}S^{1/2}J(\sigma) + \eta DX^{-1}\left[X(s^0 - s') + S(x^0 - x')\right] + D(r_d - \eta r_d^0) \\ r_V - \eta r_V^0 \end{pmatrix}. \tag{57}$$

Proof Using the fact that $w \in \mathcal{N}_{w^0}(\eta, \gamma, \theta)$ along with Equations (20), (27), and (48), we easily obtain that

$$\begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix} = \begin{pmatrix} \eta r_p^0 \\ \eta E^{-1}r_V^0 + E^{-1}(r_V - \eta r_V^0) \end{pmatrix}$$

$$= \eta\hat{A}\begin{pmatrix} x^0 - x' \\ E^{-2}(z^0 - z') \end{pmatrix} + \hat{A}\begin{pmatrix} 0 \\ E^{-1}(r_V - \eta r_V^0) \end{pmatrix} \tag{58}$$

$$s^0 - s' = -A^T(y^0 - y') + Q(x^0 - x') - V(z^0 - z') + r_d^0. \tag{59}$$

From Equation (56), we easily see that

$$-s + \sigma\mu X^{-1}e = X^{-1/2}S^{1/2}J(\sigma). \tag{60}$$

Equation (25) implies that

$$I - D^2Q = D^2(D^{-2} - Q) = D^2X^{-1}S. \tag{61}$$

Using relations (20), (26), (27), (28), (29), (49), (58), and (59), we obtain

$$\begin{aligned}
Hu^0 - h &= \hat{A}\hat{D}^2\hat{A}^T u^0 - \hat{A}\begin{pmatrix} D^2(s - \sigma\mu X^{-1}e - r_d) \\ 0 \end{pmatrix} + \begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix} \\
&= -\eta\hat{A}\hat{D}^2\hat{A}^T\begin{pmatrix} y^0 - y' \\ z^0 - z' \end{pmatrix} - \hat{A}\begin{pmatrix} D^2(s - \sigma\mu X^{-1}e - \eta r_d^0 - (r_d - \eta r_d^0)) \\ 0 \end{pmatrix} + \begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix} \\
&= -\eta\hat{A}\begin{pmatrix} D^2\left(A^T(y^0 - y') - Q(x^0 - x') + V(z^0 - z') - r_d^0\right) \\ E^{-2}(z^0 - z') \end{pmatrix} \\
&\quad - \hat{A}\begin{pmatrix} D^2\left(\eta Q(x^0 - x') - (r_d - \eta r_d^0)\right) \\ 0 \end{pmatrix} - \hat{A}\begin{pmatrix} D^2(s - \sigma\mu X^{-1}e) \\ 0 \end{pmatrix} + \begin{pmatrix} r_p \\ E^{-1}r_V \end{pmatrix} \\
&= -\eta\hat{A}\begin{pmatrix} -D^2(s^0 - s') \\ E^{-2}(z^0 - z') \end{pmatrix} - \hat{A}\begin{pmatrix} D^2\left(\eta Q(x^0 - x') - (r_d - \eta r_d^0)\right) \\ 0 \end{pmatrix} \\
&\quad - \hat{A}\begin{pmatrix} D^2(s - \sigma\mu X^{-1}e) \\ 0 \end{pmatrix} + \eta\hat{A}\begin{pmatrix} x^0 - x' \\ E^{-2}(z^0 - z') \end{pmatrix} + \hat{A}\begin{pmatrix} 0 \\ E^{-1}(r_V - \eta r_V^0) \end{pmatrix} \\
&= \hat{A}\begin{pmatrix} -D^2(s - \sigma\mu X^{-1}e) + \eta D^2(s^0 - s') + \eta(I - D^2Q)(x^0 - x') + D^2(r_d - \eta r_d^0) \\ E^{-1}(r_V - \eta r_V^0) \end{pmatrix},
\end{aligned}$$

which together with Equations (26), (60), and (61) yields Equation (57), as desired. $\blacksquare$

We now turn to the proof of Lemma 3.2.

Proof of Lemma 3.2 The fact that $w \in \mathcal{N}_{w^0}(\gamma,\theta)$ implies that $w \in \mathcal{N}_{w^0}(\eta,\gamma,\theta)$ for some $\eta \in [0,1]$. By Lemmas 4.2 and 4.3, we have that

$$\begin{aligned}
&\|Hu^0 - h\|_{H^{-1}} \\
&= \left\| \hat{A}\hat{D} \begin{pmatrix} DX^{-1/2}S^{1/2}J(\sigma) + \eta DX^{-1}\left[X(s^0-s') + S(x^0-x')\right] + D(r_d - \eta r_d^0) \\ r_V - \eta r_V^0 \end{pmatrix} \right\|_{H^{-1}} \\
&\le \left\| \begin{pmatrix} DX^{-1/2}S^{1/2}J(\sigma) + \eta DX^{-1}\left[X(s^0-s') + S(x^0-x')\right] + D(r_d - \eta r_d^0) \\ r_V - \eta r_V^0 \end{pmatrix} \right\| \\
&\le \|DX^{-1/2}S^{1/2}\|\,\|J(\sigma)\| + \eta\|DX^{-1}\|\|S(x^0-x') + X(s^0-s')\| + \left\| \begin{pmatrix} D(r_d-\eta r_d^0) \\ r_V - \eta r_V^0 \end{pmatrix} \right\|.
\end{aligned} \tag{62}$$

We will examine each norm in Equation (62) in turn. First, since $w \in \mathcal{N}_{w^0}(\gamma,\theta)$, we have that $x_i s_i \ge (1-\gamma)\mu$ for all i. It follows from a well-known result (see, e.g. [12]) that

$$\|J(\sigma)\| \le \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)^{1/2} \sqrt{n\mu}. \tag{63}$$

Moreover, using Equation (25) and the facts that $Q \succeq 0$ and $x_i s_i \ge (1-\gamma)\mu$ for all i, we obtain that

$$\begin{aligned}
\|DX^{-1}\| &= \|X^{-1}D^2X^{-1}\|^{1/2} = \|X^{-1}(Q + X^{-1}S)^{-1}X^{-1}\|^{1/2} \\
&\le \|(XS)^{-1}\|^{1/2} \le \frac{1}{\sqrt{(1-\gamma)\mu}}.
\end{aligned} \tag{64}$$

Similarly, we have

$$\max\{\|DX^{-1/2}S^{1/2}\|, \|DQ^{1/2}\|\} \le 1. \tag{65}$$

Using the fact that $(x^0,s^0) \ge |(x',s')|$ and $(x^0,s^0) \ge (x^*,s^*)$ together with Lemma 4.1, we obtain that

$$\begin{aligned}
\eta\|S(x^0-x') + X(s^0-s')\| &\le \eta\left(\|S(x^0-x')\| + \|X(s^0-s')\|\right) \le 2\eta\left(\|Sx^0\| + \|Xs^0\|\right) \\
&\le 2\eta(x^Ts^0 + x^Ts^0) \le \left(6n + \frac{\theta^2}{2}\right)\mu.
\end{aligned} \tag{66}$$

The fact that $\|DQ^{1/2}\| \le 1$ implies that $Q^{1/2}D^2Q^{1/2} \preceq I$, which in turn implies that $QD^2Q \preceq Q$. Next, the fact that $w \in \mathcal{N}_{w^0}(\gamma,\theta)$ implies that $r_d - \eta r_d^0 = Qt$ for some vector t. We use these facts along with Equation (13) to observe that

$$\begin{aligned}
\left\| \begin{pmatrix} D(r_d-\eta r_d^0) \\ r_V - \eta r_V^0 \end{pmatrix} \right\| &= \left[t^T(QD^2Q)t + \|r_V - \eta r_V^0\|^2\right]^{1/2} \\
&\le \left[t^TQt + \|r_V - \eta r_V^0\|^2\right]^{1/2} \\
&= \left[|||r_d - \eta r_d^0|||_Q^2 + \|r_V - \eta r_V^0\|^2\right]^{1/2} \le \theta\sqrt{\mu}.
\end{aligned} \tag{67}$$

The result now follows by combining bounds (63)–(67) into Equation (62). ■

5. Concluding remarks

In this paper, we have presented two important extensions to the results of [15]. First, we have extended the available choices of preconditioners in the recipe for constructing inexact search directions to a whole class of preconditioners, which includes the MWB preconditioner used in [15] as a special case. These preconditioners are indexed by a positive semidefinite matrix F, and convergence using these preconditioners depends on how well F approximates $\hat{D}^2$. Second, we have presented the HANE as a new method to determine an approximate search direction in the inexact IPDPF algorithm.

In the specific case of LP, the results presented in this paper can be simplified considerably. First, note that in this case Equation (18) is not present, and that Equation (16) reduces to $A^T\Delta y + \Delta s = -r_d$, since $V = 0$ and $Q = 0$, and hence $g = 0$. Furthermore, Equation (19) reduces to $\|p\|_\infty \leq \gamma\sigma\mu/4$, i.e. the second inequality in Equation (19) disappears. Second, the HANE reduces to the standard normal equation. Third, the last inequality in the definition of $\mathcal{N}_{w^0}(\eta, \gamma, \theta)$ in Equation (20) disappears, and hence we may choose $\theta = 0$. Finally, noting that the z-component of u^0 in Equation (49) is not involved in LP, by choosing $y' = y^0$ (and s^0 sufficiently large so that the conditions of Lemma 3.2 hold), we see that $u^0 = 0$ is a viable starting point for the iterative solver.

One feature of the MWB preconditioner $\hat{T}$ discussed in Subsection 3.2 is that it satisfies $\hat{T}H\hat{T}^T \succeq I$, as was shown in [20]. Thus, the Adaptive PCG (APCG) method in [19] may be used as the iterative solver to determine an approximate solution to the preconditioned HANE. The APCG method, applied to the preconditioned HANE with initial preconditioner $\hat{T}$, determines a solution u such that $\|f\|_{H^{-1}} \leq \delta^{-1}\|f^0\|_{H^{-1}}$ in at most

$$\mathcal{O}(\log\det(\hat{T}H\hat{T}^T) + (m+l)^{1/2}\log\delta)$$

iterations (see [19]). Since

$$\log\det(\hat{T}H\hat{T}^T) \leq (m+l)\log\lambda_{\max}(\hat{T}H\hat{T}^T) \leq 2(m+l)\log\varphi_{\hat{A}},$$

it follows that a suitable approximate solution to the HANE can be found in at most

$$\mathcal{O}((m+l)\log\varphi_{\hat{A}} + (m+l)^{1/2}\log(n\varphi_{\hat{A}})) \tag{68}$$

iterations of the APCG method. One unique feature of the APCG method is that the preconditioner $\hat{T}$ is periodically updated to better condition the HANE matrix. The bound (68) assumes that we form v according to Equation (44) using the preconditioner $G = \hat{T}^{-1}\hat{T}^{-T}$ employed at the beginning of the APCG method. It would be interesting to investigate whether v could be formed using the updated preconditioners generated during the course of the APCG method. One question that would need to be addressed is whether the updated preconditioner fits into the form $G = \hat{A}F\hat{A}^T$ required for the results in Section 3 to hold. Exploring adaptive preconditioning strategies, such as the one employed by the APCG method, for generating inexact search directions in the context of the inexact IPDPF algorithm, is certainly an interesting area for future research.

Acknowledgements

Z. Lu was partially supported by SFU Presidential Grant and NSERC Discovery Grant. R.D.C. Monteiro was partially supported by NSF Grants CCR-0203113 and CCF-0430644 and CCF-0808863 and ONR Grants N00014-05-1-0183 and N00014-08-1-0033. J.W. O'Neal was supported in part by the NDSEG Fellowship Program sponsored by the Department of Defense.

References

[1] K.M. Anstreicher, *Linear programming in O(n3L/ln n) operations*, SIAM J. Optim. 9(4) (1999), pp. 803–812.
[2] V. Baryamureeba and T. Steihaug, *On the convergence of an inexact primal-dual interior point method for linear programming*, J. Optim. Theory Appl., to appear.
[3] V. Baryamureeba, T. Steihaug, and Y. Zhang, *Properties of a class of preconditioners for weighted least squares problems*, Tech. Rep. 16, Department of Computational and Applied Mathematics, Rice University, 1999.
[4] L. Bergamaschi, J. Gondzio, and G. Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Comput. Optim. Appl. 28(2) (2004), pp. 149–171.
[5] R.W. Freund, F. Jarre, and S. Mizuno, *Convergence of a class of inexact interior-point algorithms for linear programs*, Math. Oper. Res. 24(1) (1999), pp. 50–71.
[6] C.C. Gonzaga. *An algorithm for solving linear programming problems in* $O(n^3L)$ *operations*, in *Progress in Mathematical Programming: Interior-Point and Related Methods*, ch. 1, 1989, pp. 1–28.
[7] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, 1997.
[8] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica 4 (1984), pp. 373–395.
[9] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.
[10] M. Kojima, N. Megiddo, and S. Mizuno, *A primal-dual infeasible-interior-point algorithm for linear programming*, Math. Program. 61(3) (1993), pp. 263–280.
[11] M. Kojima, S. Mizuno, and A. Yoshise, *A polyonimal-time algorithm for a class of linear complementarity problems,* Math. Program. 44(1) (1989), pp. 1–26.
[12] ———, *A primal-dual interior point algorithm for linear programming*, in *Progress in Mathematical Programming: Interior-Point and Related Methods*, ch. 2, 1989, pp. 29–47.
[13] J. Korzak, *Convergence analysis of inexact infeasible-interior-point algorithms for solving linear programming problems*, SIAM J. Optim. 11(1) (2000), pp. 133–148.
[14] V.V. Kovacevic-Vujcic and M.D. Asic, *Stabilization of interior-point methods for linear programming*, Comput. Optim. Appl. 14 (1999), pp. 331–346.
[15] Z. Lu, R.D.C. Monteiro, and J.W. O'Neal, *An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming*, SIAM J. Optim. 17(1) (2006), pp. 287–310.
[16] D.G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, 1984.
[17] S. Mizuno and F. Jarre, *Global and polynomial-time convergence of an infeasible-interior-point algorithm using inexact computation,* Math. Program. 84 (1999), pp. 357–373.
[18] R.D.C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms, part I: linear programming*, Math. Program. 44 (1989), pp. 27–41.
[19] R.D.C. Monteiro, J.W. O'Neal, and A.S. Nemirovski, *A new conjugate gradient algorithm incorporating adaptive ellipsoid preconditioning*, Tech. Rep., Georgia Institute of Technology, 2004.
[20] R.D.C. Monteiro, J.W. O'Neal, and T. Tsuchiya, *Uniform boundedness of a preconditioned normal matrix used in interior point methods*, SIAM J. Optim. 15(1) (2004), pp. 96–100.
[21] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, 1995.
[22] A.R.L. Oliveira and D.C. Sorensen, *Computational experience with a preconditioner for interior point methods for linear programming,* Tech. Rep. 28, Department of Computational and Applied Mathematics, Rice University, 1997.
[23] L.F. Portugal, M.G.C. Resende, G. Veiga, and J.J. Judice, *A truncated primal-infeasible dual feasible network interior point method*, Networks, 35 (2000), pp. 91–108.
[24] J. Renegar, *Condition numbers, the barrier method, and the conjugate-gradient method,* SIAM J. Optim. 6 (1996), pp. 879–912.
[25] M.G.C. Resende and G. Veiga, *An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite uncapacitated networks*, SIAM J. Optim. 3 (1993), pp. 516–537.
[26] K.C. Toh, R.H. Tütüncü, and M.J. Todd, *Inexact primal-dual path-following algorithms for a special class of convex quadratic SDP and related problems*, Tech. Rep., Cornell University, 2005.
[27] P.M. Vaidya, *Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners*, Tech. Rep., a talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, October 1991, Minneapolis.
[28] Y. Zhang, *On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem*, SIAM J. Optim. 4(1) (1994), pp. 208–227.
[29] G. Zhou and K.-C. Toh, *Polynomiality of an inexact infeasible interior point algorithm for semidefinite programming*, Math. Program. 99 (2004), pp. 261–282.