Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming

Joo-Siong Chai *and Kim-Chuan Toh †

Abstract

We propose to compute the search direction at each interior-point iteration for a linear program via a reduced augmented system that typically has a much smaller dimension than the original augmented system. This reduced system is potentially less susceptible to the ill-conditioning effect of the elements in the $(1, 1)$ block of the augmented matrix. A preconditioner is then designed by approximating the block structure of the inverse of the transformed matrix to further improve the spectral properties of the transformed system. The resulting preconditioned system is likely to become better conditioned toward the end of the interior-point algorithm. Capitalizing on the special spectral properties of the transformed matrix, we further proposed a two-phase iterative algorithm that starts by solving the normal equations with PCG in each IPM iteration, and then switches to solve the preconditioned reduced augmented system

*Computational Engineering Program, Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576. (smap0059@nus.edu.sg).

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543, Singapore (mattohkc@math.nus.edu.sg); and Singapore-MIT Alliance, E4-04-10, 4 Engineering Drive 3, Singapore 117576. Research supported in parts by NUS Research Grant R146-000-076-112 and SMA IUP Research Grant.

with symmetric quasi-minimal residual (SQMR) method when it is advantageous to do so. The experimental results have demonstrated that our proposed method is competitive with direct methods in solving large-scale LP problems and a set of highly degenerate LP problems.

1 Introduction

Interior-point methods (IPM) are commonly used to solve large-scale linear programming (LP) problems. The most computationally intensive part at each iteration of an IPM involves the solution of one or more linear systems of equations. There are several ways to solve the linear systems. The standard approach uses direct method to solve the normal equation system based on sparse Cholesky factorization of the symmetric positive definite Schur complement matrix [3], [11], [23]. Another common approach solves the augmented system of equations by Bunch-Parlett decomposition of the symmetric indefinite augmented matrix [4], [20], [21]. However, as the size of the matrix becomes large, the computational effort of direct methods grows in the order of m^3, if the LP data is dense.

One way around this difficulty is to employ iterative methods to solve the linear systems of equations. As justified by Freund and Jarre [5], iterative algorithms have many advantages over direct methods. Iterative methods such as Krylov subspace algorithms have the potential to reduce the computation time by working with approximate directions rather than exact directions (modulo rounding errors). The amount of reduction achievable by any iterative method is determined by the spectral properties of the coefficient matrix, which determine the convergence rate of the iterative method [2]. Hence, it is important to construct an effective preconditioner for the coefficient matrix to achieve a small condition number or good clustering of the eigenvalues for the preconditioned system [16], [19].

It is known that the spectral properties of both the Schur complement matrix and the augmented matrix deteriorate as the interior-point iterates converge to a solution. Hence, effective preconditioning is especially critical for the iterative solver as the IPM converges. In particular, the Schur complement matrix is usually more ill-conditioned

than the augmented matrix. As a result, it is usually more difficult to achieve convergence with the normal equation system, and solving it typically yields less accurate solutions compared to solving the augmented system for a given linear system, although the former has the advantage of being symmetric positive definite and smaller in size than the latter.

Some progress has been achieved in constructing preconditioners for the augmented systems. Practically all of the work in this area exploited the block structure of the augmented matrix in designing the preconditioners, as can be seen in [1], [2], [7], [5], [10], [14] and [18]. Theoretical properties of the resulting preconditioned augmented systems, such as eigenvalue distributions and convergence bounds, have also been derived in [1], [2], [9], [10], [12], [14] and [18]. Oliveira and Sorensen [14] have shown that, in general, it is potentially easier to design an effective preconditioner for the augmented system than the normal equation system. Specifically, they showed that every preconditioner for the normal equation system yields an equivalent preconditioner for the augmented system, but the converse is not true. Based on the augmented system, they proposed a class of symmetric preconditioners that was shown to have the desirable property of improving performance (in terms of number of iterative steps required to solve the linear system) as the IPM approaches a solution. This phenomenon is unusual because the linear systems typically become more ill-conditioned as the interior-point iterates converge to a solution. However, sophisticated techniques are required for their implementation, such as selecting linearly independent columns from the constraint matrix and computing the sparse LU factorization for the resulting submatrix.

In this paper, we propose a preconditioning approach that produces a preconditioned system that also becomes better conditioned as the interior-point iterates converge to a solution. However, we take a different approach from that taken by Oliveira and Sorensen [14]. We first exploit the block structure of the symmetric indefinite augmented matrix to transform the augmented system into an equivalent reduced 2×2 block system, which is potentially better conditioned than the original augmented system. Based on the transformed system, we design the preconditioning matrix by approximating the block structure of the inverse of the transformed matrix. We also

propose a two-phase iterative algorithm that takes advantage of the spectral properties of the transformed matrix to compute the search directions in the interior-point method. Phase one employs existing direct or iterative method to compute the search directions via the normal equations, while phase two computes the search directions based on the transformed equations by some preconditioned iterative method. Computational experiments have produced some encouraging results.

We use $\|\cdot\|$ to denote either the vector or matrix 2-norm. For any two non-negative numbers α and β, we write $\alpha = O(\beta)$ if there is a positive constant c such that $\alpha \leq c\beta$. We write $\alpha = \Omega(\beta)$ to indicate that $\alpha = O(\beta)$ and $\beta = O(\alpha)$. For a matrix M, we write $M = O(\beta)$ to denote $\|M\| = O(\beta)$. We let $\mathcal{R}(M)$ and $\mathcal{N}(M)$ be the column and null spaces of M respectively. If M is nonsingular, the condition number of M is the number $\kappa(M) = \|M\|\|M^{-1}\|$. For a vector x, we use $\mathrm{diag}(x)$ to denote the diagonal matrix whose main diagonal is x. If all the components of x are nonzero, then x^{-1} denotes the vector whose ith component is x_i^{-1}.

The paper is organized as follows. In section 2, we describe the augmented system and normal equation in more detail. In section 3, we derive some results on the condition number of the Schur complement matrix. We present the reduced augmented system and some theoretical results in sections 4 and 5. Then, in section 6, we discuss preconditioner design based on the block structure of the inverse of the reduced augmented matrix. In section 7, we propose a two-phase iterative algorithm that solves the normal equation in the first phase and the reduced augmented system in the second phase, as well as address some implementational issues. Finally, we demonstrate the performance of our approach through some computational results in section 8, before concluding the paper in section 9.

2 Problem formulation

Consider the following LP problem in the standard form:

$$\min\{c^T x \,:\, Ax = b,\ \ x \geq 0\}, \tag{1}$$

where $c, x \in I\!R^n$, $b \in I\!R^m$, $A \in I\!R^{m \times n}$. The dual LP problem of (1) has the form:

$$\max\{b^T y \,:\, A^T y + z = c, \;\; z \geq 0\}, \tag{2}$$

where $z \in I\!R^n$, $y \in I\!R^m$. We assume that A has full row rank throughout this paper.

The first order optimality conditions, also known as the KKT conditions, corresponding to (1) and (2) are:

$$Ax \;=\; b, \quad A^T y + z \;=\; c, \quad XZe \;=\; 0, \tag{3}$$

where $X = \mathrm{diag}(x)$, $Z = \mathrm{diag}(z)$ and e is the n-vector of all ones.

To apply primal-dual IPM to solve the given LP problem, we solve the following perturbed KKT conditions instead:

$$Ax \;=\; b, \quad A^T y + z \;=\; c, \quad XZe \;=\; \mu e, \tag{4}$$

where $\mu > 0$ is a barrier parameter. It can be seen that (4) is identical to (3), except that the complementarity condition is perturbed by μ. In the IPM solution of the LP problem, the parameter $\mu > 0$ is reduced by a certain factor at each iteration of the IPM. For every $\mu > 0$, (4) has a unique solution $(x(\mu), y(\mu), z(\mu))$ which defines the central path for the primal-dual feasible region of (1) and (2). As μ gradually reduces to 0, the solution converges to an optimal solution.

To solve for the search directions $(\Delta x, \Delta y, \Delta z)$, we apply Newton's method to the perturbed KKT conditions (4) to obtain the following Newton equation system:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix} \tag{5}$$

where

$$r_p \;=\; b - Ax, \quad r_d \;=\; c - A^T y - z, \quad r_c \;=\; \mu e - XZe. \tag{6}$$

By substituting $\Delta z \;=\; X^{-1}(r_c - Z\Delta x)$ into the second equation in (5), we can reduce the set of Newton equations to the following augmented system of equations:

$$\underbrace{\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} g \\ r_p \end{bmatrix} \tag{7}$$

where $D = X^{-1}Z$ and $g = r_d - X^{-1}(\mu e - XZe)$.

By eliminating Δx from (7), this can further be reduced to the normal equations

$$\underbrace{AD^{-1}A^T}_{\mathcal{S}}\Delta y = AD^{-1}g + r_p. \tag{8}$$

The matrix $\mathcal{S} = AD^{-1}A^T$ in (8) is the Schur complement of the augmented matrix $\mathcal{A}$ in (7). After solving for Δy in (8), we may compute Δx and Δz via

$$\Delta x = D^{-1}(A^T\Delta y - g), \quad \Delta z = r_d - A^T\Delta y. \tag{9}$$

At each iteration of the primal-dual IPM, the solution of the linear system (7) or (8) have to be computed numerically. This is the most computationally intensive part of the IPM's solution of an LP problem, especially when the coefficient matrix is dense. It should be noted that the only part of the coefficient matrix that changes from one iteration to another of the IPM is the diagonal matrix D. Very often, the matrix D is highly ill-conditioned, particularly when the interior point iterates approach an optimal solution. This typically causes the coefficient matrix to be highly ill-conditioned even if A is well-conditioned (for details, see [22, ch.10, 11]). Applying iterative solutions such as Krylov subspace method to such systems often encounter exceedingly slow convergence. Furthermore, constructing an effective preconditioner for such a system is extremely difficult when D is ill-conditioned.

3 Conditioning of the normal equations

It is well known that the ill-conditioning of the matrix D in (7) is often due to the separation of its diagonal elements into two clusters with different orders of magnitude, where one cluster corresponds to non-active constraints while another corresponds to active constraints. Very often, a highly ill-conditioned D leads to a highly ill-conditioned augmented matrix $\mathcal{A}$ even if A is well-conditioned (see [22, ch.10, 11] for details).

When μ is small, strict complementarity of the iterates (x, z) implies that the diagonal elements of D separate into two groups of magnitude $\Omega(\mu)$ and $\Omega(\frac{1}{\mu})$. Without

loss of generality, we assume that D has the partition

$$D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} \tag{10}$$

where

$$D_1 \;=\; \mu \tilde{D}_1, \qquad D_2 \;=\; \frac{1}{\mu}\tilde{D}_2,$$

with $\mathrm{diag}(\tilde{D}_i) = \Omega(1)$ for $i = 1, 2$. Further, let $A = [A_1, \quad A_2]$ be the partition conforming to that of D. In this section, we shall assume that $\mu \ll 1$. Note that the dimension of D_1 is not greater than m, and if the LP is not severely degenerate at the optimal solution, it is typically close to m. Also note that when μ is sufficiently small, the partition would remain fixed independent of $/mu$. Hence, we may assume that the positive singular values of A_1 to be $\Omega(1)$ as μ tends to zero.

For the analysis below, we consider the SVD: $A_1\tilde{D}_1^{-1/2} = U\Sigma V^T = U_1\Sigma_1 V_1^T$, where Σ_1 is the square diagonal matrix of positive singular values. The former decomposition is the full SVD, while the latter is the reduced SVD. Let U_2 be the matrix whose columns form an orthonormal basis of $\mathcal{N}(\tilde{D}_1^{-1/2}A_1^T) = \mathcal{N}(A_1^T)$ (hence $A_1^T U_2 = 0$). Note that U_2 is a null matrix iff $\mathcal{N}(A_1^T) = \{0\}$, i.e., iff A_1 has full row rank. Let

$$W_{ij} \;=\; U_i^T(A_2\tilde{D}_2^{-1}A_2^T)U_j, \quad \text{for } i, j = 1, 2. \tag{11}$$

Lemma 3.1 *Assuming that A has full row rank, then W_{22} is nonsingular.*

Proof. Since A has full row rank, it is easily shown that $\mathcal{N}(A_1^T) \bigcap \mathcal{N}(A_2^T) = \{0\}$. We show that $A_2^T U_2$ has full column rank by contradiction. Suppose $A_2^T U_2 v = 0$ for some $v \neq 0$. Then $U_2 v \in \mathcal{N}(A_2^T)$. Since we also have $U_2 v \in \mathcal{N}(A_1^T)$ because $A_1^T U_2 = 0$, this implies that $U_2 v \in \mathcal{N}(A_1^T) \bigcap \mathcal{N}(A_2^T) = \{0\}$, which is a contradiction. Thus $U_2^T A_2$ has full row rank and W_{22} is nonsingular. ∎

Proposition 3.1 *Consider the Schur complement matrix $\mathcal{S} = AD^{-1}A^T$ in (8).*

(a) If A_1 has full row rank, then $\kappa(\mathcal{S}) \approx \kappa(A_1\tilde{D}_1^{-1}A_1^T)$.

(b) If A_1 does not have full row rank, then

$$\kappa(\mathcal{S}) \approx \frac{1}{\mu^2}\|A_1\tilde{D}_1^{-1}A_1^T\|\|W_{22}^{-1}\|. \tag{12}$$

Note that $\|W_{22}^{-1}\| = O(1)$ *since* W_{22} *is nonsingular.*

Proof. (a) By expanding the matrix $\mathcal{S}$, we can rewrite it as

$$\mathcal{S} \;=\; A_1 D_1^{-1} A_1^T + A_2 D_2^{-1} A_2^T \;=\; \frac{1}{\mu}\left(A_1 \tilde{D}_1^{-1} A_1^T + \mu^2 A_2 \tilde{D}_2^{-1} A_2^T\right).$$

When A_1 has full row rank, the matrix $A_1 \tilde{D}_1^{-1} A_1^T$ is nonsingular. The required result is now obvious and we shall omit the details.

(b) Using the SVD of $A_1 \tilde{D}_1^{-1/2}$ and the expression in (11), we thus obtain

$$\mathcal{S} \;=\; \frac{1}{\mu} U \begin{bmatrix} \Sigma_1^2 + \mu^2 W_{11} & \mu^2 W_{12} \\ \mu^2 W_{12}^T & \mu^2 W_{22} \end{bmatrix} U^T,$$

and using the result in [16, p. 389], we have

$$\mathcal{S}^{-1} \;=\; \mu U \begin{bmatrix} G^{-1} + \mu^2 G^{-1} W_{12} Q^{-1} W_{12}^T G^{-1} & -G^{-1} W_{12} Q^{-1} \\ -Q^{-1} W_{12}^T G^{-1} & \dfrac{1}{\mu^2} Q^{-1} \end{bmatrix} U^T,$$

where $Q = W_{22} - \mu^2 W_{12}^T G^{-1} W_{12}$ with $G = \Sigma_1^2 + \mu^2 W_{11}$. It is clear that $\|\mathcal{S}\| \approx \|A_1 \tilde{D}_1^{-1} A_1^T\|/\mu$ and $\|\mathcal{S}^{-1}\| \approx \|Q^{-1}\|/\mu \approx \|W_{22}^{-1}\|/\mu$, and the required result (12) follows readily. □

Remark 3.1 *The preceding proposition indicates that in the general case when* A_1 *does not have full row rank, the Schur complement matrix* $\mathcal{S}$ *can be very ill-conditioned as the barrier parameter* μ *approaches zero.*

4 Reduced augmented system

In this section, we show that it is possible to transform the augmented matrix $\mathcal{A}$ in (7) to an equivalent reduced 2×2 block matrix that can potentially be better conditioned. Let the partition in Δx and g corresponding to that of D in (10) be

$$\Delta x = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}. \tag{13}$$

Let E_1 be a given positive definite diagonal matrix with the same dimension as D_1. (Usually we choose E_1 to be a positive multiple of the identity matrix.) Instead

of computing Δx and Δy from (7), we propose to compute them by solving a smaller system which we call the reduced augmented equation (RAE), as given in the next lemma. The main motivation for using the RAE instead of the augmented system (7) is that the former is generally a much smaller system than the latter.

Lemma 4.1 *The solution of (7) can be computed from the following reduced augmented equation (RAE):*

$$\underbrace{\begin{bmatrix} H & B \\ B^T & -\Psi \end{bmatrix}}_{K} \begin{bmatrix} \Delta y \\ \Delta \tilde{x}_1 \end{bmatrix} = \begin{bmatrix} h \\ F_1^{-1/2} g_1 \end{bmatrix} \tag{14}$$

where $F_1 = E_1 + D_1$, *and*

$$\begin{aligned} \Delta \tilde{x}_1 &= F_1^{-1/2} E_1 \Delta x_1 \\ \Psi &= D_1 E_1^{-1} \\ B &= A_1 F_1^{-1/2} \\ H &= A\mathrm{diag}(F_1^{-1},\, D_2^{-1}) A^T = BB^T + A_2 D_2^{-1} A_2^T \\ h &= r_p + A\mathrm{diag}(F_1^{-1},\, D_2^{-1}) g \end{aligned}$$

Once Δy *has been computed,* Δx_2 *can be readily computed from the equation*

$$\Delta x_2 = D_2^{-1}(A_2^T \Delta y - g_2). \tag{15}$$

Proof. By substituting the partitions of D, A and g into (7) and using $\Delta x_2 = D_2^{-1}(A_2^T \Delta y - g_2)$, we get

$$-D_1 \Delta x_1 + A_1^T \Delta y = g_1 \tag{16}$$

$$A_1 \Delta x_1 + A_2 D_2^{-1} A_2^T \Delta y = r_p + A_2 D_2^{-1} g_2. \tag{17}$$

By adding $A_1 F_1^{-1}$ times (16) to (17), we have

$$A_1 F_1^{-1} E_1 \Delta x_1 + A\mathrm{diag}(F_1^{-1},\, D_2^{-1}) A^T \Delta y = r_p + A\mathrm{diag}(F_1^{-1}, D_2^{-1}) g.$$

The above equation, together with (16) scaled by $F^{-1/2}$ gives (14). ∎

Lemma 4.2

$$\|K\| \le 2 \max\Big\{\|H\|,\, \|B\|,\, \|\Psi\|\Big\}.$$

Proof. The proof is easy and we shall omit it. □

Lemma 4.3 *The inverse of the reduced augmented matrix K in (14) is given by*

$$K^{-1} \;=\; \begin{bmatrix} H^{-1/2}(I-P)H^{-1/2} & H^{-1}BS^{-1} \\ S^{-1}B^TH^{-1} & -S^{-1} \end{bmatrix}, \tag{18}$$

where $S = B^TH^{-1}B + \Psi$, and $P = H^{-1/2}BS^{-1}B^TH^{-1/2}$ satisfies the condition $0 \preceq P \preceq I$, i.e., P and $I-P$ are positive semidefinite. Furthermore,

$$\|K^{-1}\| \;\le\; 2\max\Big\{\|H^{-1}\|, \|S^{-1}\|\Big\}$$

Proof. The proof of (18) can be deduced from [16, p. 389]. By the definition of S, we have $0 \preceq S^{-1/2}B^TH^{-1}BS^{-1/2} \preceq I$, and thus $\|H^{-1/2}BS^{-1/2}\| \le 1$. This implies that

$$\begin{aligned} \|H^{-1}BS^{-1}\| \;&\le\; \|H^{-1/2}\|\,\|H^{-1/2}BS^{-1/2}\|\,\|S^{-1/2}\| \;\le\; \|H^{-1/2}\|\|S^{-1/2}\| \\ &\le\; \max\Big(\|H^{-1}\|\,,\,\|S^{-1}\|\Big). \end{aligned}$$

By Lemma 4.2, it is easy to see that

$$\|K^{-1}\| \;\le\; 2\max\Big\{\|H^{-1/2}(I-P)H^{-1/2}\|,\,\|H^{-1}BS^{-1}\|,\,\|S^{-1}\|\Big\} \;\le\; 2\max\Big\{\|H^{-1}\|,\,\|S^{-1}\|\Big\}.$$

Note that

$$\begin{aligned} \|H^{-1}BS^{-1}\| \;&=\; \|H^{-1/2}(H^{-1/2}BS^{-1})\| \;\le\; \|H^{-1}\|^{1/2}\,\|S^{-1}B^TH^{-1}BS^{-1}\|^{1/2} \\ &\le\; \|H^{-1}\|^{1/2}\,\|S^{-1}\|^{1/2} \;\le\; \max\Big\{\|H^{-1}\|,\,\|S^{-1}\|\Big\}. \end{aligned}$$

□

Let the SVD of B be $B = U\Sigma V^T = U_1\Sigma_1V_1^T$, where Σ_1 is the diagonal matrix of positive singular values. Here U_1 and V_1 are the matrices whose columns form an orthonormal basis of $\mathcal{R}(B)$ and $\mathcal{R}(B^T)$, respectively. Let U_2 and V_2 be the columns of U and V other than those in U_1 and V_1, respectively. Then the columns of U_2 and V_2 form an orthonormal basis of $\mathcal{N}(B^T) = \mathcal{N}(A_1^T)$ and $\mathcal{N}(B)$, respectively. We shall next analyze the conditioning of K.

Lemma 4.4 *The following results hold.*

(a) Consider the matrix $H = BB^T + \mu\tilde{W}$, where $\tilde{W} = A_2\tilde{D}_2^{-1}A_2^T$. We have

$$\mu H^{-1} \;=\; U_2 W_{22}^{-1} U_2^T + O(\mu), \tag{19}$$

where $W_{22} = U_2^T\tilde{W}U_2$. Thus $\|H^{-1}\| \approx \|W_{22}^{-1}\|/\mu$ if $\mathcal{N}(A_1^T) \neq \{0\}$.

(b)

$$B^T H^{-1} B \;=\; V_1(I + O(\mu))V_1^T. \tag{20}$$

If B has full column rank, then $B^T H^{-1} B = I + V_1 O(\mu) V_1^T$.

(c) Consider $S = B^T H^{-1} B + \Psi$. We have

$$\mu S^{-1} \;=\; V_2\Big(V_2^T\tilde{D}_1 E_1^{-1} V_2\Big)^{-1} V_2^T + O(\mu).$$

Proof. (a) Let $W := U^T\tilde{W}U$ be written as

$$W \;=\; \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^T & W_{22} \end{bmatrix},$$

where $W_{ij} = U_i^T\tilde{W}U_j$ for $i, j = 1, 2$. We have

$$H \;=\; U \begin{bmatrix} \Sigma_1^2 + \mu W_{11} & \mu W_{12} \\ \mu W_{12}^T & \mu W_{22} \end{bmatrix} U^T.$$

Thus

$$H^{-1} \;=\; U \begin{bmatrix} G^{-1} + \mu G^{-1} W_{12} Q^{-1} W_{12}^T G^{-1} & -G^{-1} W_{12} Q^{-1} \\ -Q^{-1} W_{12}^T G^{-1} & \dfrac{1}{\mu} Q^{-1} \end{bmatrix} U^T$$

where $G = \Sigma_1^2 + \mu W_{11}$ and $Q = W_{22} - \mu W_{12}^T G^{-1} W_{12}$. Thus

$$\mu H^{-1} \;=\; U_2 Q^{-1} U_2^T + O(\mu) \;=\; U_2 W_{22}^{-1} U_2^T + O(\mu).$$

This completes the proof of (19).

(b) Note that we have

$$B^T H^{-1} B \;=\; V_1\Sigma_1\left(G^{-1} + \mu G^{-1} W_{12} Q^{-1} W_{12}^T G^{-1}\right)\Sigma_1 V_1^T.$$

Since $G^{-1} = \Sigma_1^{-2} + O(\mu)$, we get the result in (20) readily. When B has full column rank, V_1 is an orthogonal matrix, and hence $V_1 V_1^T = I$.

(c) It is clear that S can be written in the form

$$S \;=\; V \begin{bmatrix} I + O(\mu) + \mu\tilde{\Psi}_{11} & \mu\tilde{\Psi}_{12} \\ \mu\tilde{\Psi}_{12}^T & \mu\tilde{\Psi}_{22} \end{bmatrix} V^T,$$

where $\tilde{\Psi}_{ij} = V_i^T \tilde{D}_1 E_1^{-1} V_j$, for $i, j = 1, 2$. Using the same proof as in (a), the required result is easily shown. ∎

Proposition 4.1 *Assume that $\mu \ll 1$ so that $\|\Psi\| \leq \|B\|$.*

(a) If A_1 has full row rank, then BB^T is nonsingular and

$$\kappa(K) = \begin{cases} O(1)\max\{1, \|B\|^2\}\max\{\|(BB^T)^{-1}\|, 1\}, & \text{if } A_1 \text{ has full column rank;} \\ \dfrac{O(1)}{\mu}\max\{1, \|B\|^2\}\max\{\mu\|(BB^T)^{-1}\|, \|(V_2^T\tilde{D}_1 E_1^{-1} V_2)^{-1}\|\}, & \text{otherwise.} \end{cases}$$

(b) If A_1 does not have full row rank, then

$$\kappa(K) = \frac{O(1)}{\mu}\max\{1, \|B\|^2\}\max\{\|W_{22}^{-1}\|,\, \|(V_2^T\tilde{D}_1 E_1^{-1} V_2)^{-1}\|\}. \tag{21}$$

Proof. From Lemma 4.2, and noting that $\|H\| \approx \|B\|^2$, it is easy to see that $\|K\| = O\Big(\max\{1, \|B\|^2\}\Big)$.

(a) Since BB^T is nonsingular, we have $\|H^{-1}\| = \|(BB^T)^{-1}\| + O(\mu)$. If A_1 also has full column rank, then $S = B^T H^{-1} B + \Psi = I + \mu\tilde{D}_1 E_1^{-1} + O(\mu)$, implying that $\|S^{-1}\| = 1 + O(\mu)$. Thus Lemma 4.3 implies that $\|K^{-1}\| = O(1)\max\{\|(BB^T)^{-1}\|,\, 1\}$. If A_1 does not have full column rank, then by Lemma 4.4 (c), $\|S^{-1}\| \approx \|(V_2^T\tilde{D}_1 E_1^{-1} V_2)^{-1}\|/\mu$, and the required result follows.

(b) From Lemma 4.4, we have $\|H^{-1}\| \approx \|W_{22}^{-1}\|/\mu$, and $\|S^{-1}\| \approx \|(V_2^T\tilde{D}_1 E_1^{-1} V_2)^{-1}\|/\mu$. Thus $\|K^{-1}\| = O\Big(\max\{\|W_{22}^{-1}\|,\, \|(V_2^T\tilde{D}_1 E_1^{-1} V_2)^{-1}\|\}\Big)/\mu$, and the required result follows readily. ∎

Comparing the results in Prop 3.1 and Prop 4.1, we can see that the effect of an ill-conditioned D on the reduced augmented matrix K in (14) is potentially less prominent than the Schur complement matrix $\mathcal{S}$ in (8).

5 Residual vectors

Suppose the system of Newton equations (5) is solved via the reduced augmented system (14). Then the residual vectors associated with (5) and (14) have the same order of magnitude as articulated in the following lemma.

Lemma 5.1 *Suppose Δy and Δx_1 is computed approximately from the RAE (14) with residual vector*

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} h \\ F_1^{-1/2} g_1 \end{bmatrix} - \begin{bmatrix} H & B \\ B^T & -\Psi \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta \tilde{x}_1 \end{bmatrix}$$

Assume that once Δy and $\Delta \tilde{x}_1$ is given, Δx_2 can be computed without rounding errors from (15), and Δz is also computed exactly from (9). Then the residual vector associated with the Newton equations (5) is given by

$$\left[\xi - A_1 F_1^{-1/2}\eta;\ 0;\ (F_1^{1/2}\eta\ ;\ 0)\right].$$

Proof. It is easily shown that

$$\begin{aligned} -D\Delta x + A^T \Delta y &= g - [F_1^{1/2}\eta\ ;\ 0] \\ A\Delta x &= r_p - \left(\xi - A_1 F_1^{-1/2}\eta\right), \end{aligned}$$

and the required result follows readily from the above equations. □

6 Preconditioning Approach

Most existing preconditioning techniques for the augmented matrix in (7) assume implicitly that D is reasonably well-conditioned. However, as mentioned, Krylov subspace iterative methods often encounter slow convergence due to ill conditioning of $\mathcal{A}$ or $\mathcal{S}$ because of an increasingly ill-conditioned D as the IPM converges to a solution. The matter is made worse because constructing an effective preconditioner for $\mathcal{A}$ or $\mathcal{S}$ is extremely difficult when D is ill-conditioned. As shown in the preceding section, for the RAE system in (14), the effect of an ill-conditioned D is less prominent than that of the normal equations. Hence we expect the task of constructing an effective preconditioner

to be easier based on the theory developed by Oliveira and Sorensen [14] who showed that in general, it is potentially more difficult to design a suitable preconditioner for the Schur complement matrix $\mathcal{S}$ than the augmented matrix $\mathcal{A}$.

We shall adapt the results established by Oliveira and Sorensen [14] to show that every preconditioner for the normal equation system yields an equivalent preconditioner for the RAE system, but the converse is not true. Therefore, the preconditioned normal equation system can be seen as a particular case of the preconditioned RAE system.

Lemma 6.1 *Consider the reduced augmented matrix K in (14). For an arbitrary pair of nonsingular left-right preconditioners (M, N) for the Schur complement matrix $\mathcal{S}$ in (8), we can construct a corresponding pair of preconditioners (P, Q) for K such that*

$$P^{-1}KQ^{-1} = \begin{bmatrix} I & 0 \\ 0 & M\mathcal{S}N^T \end{bmatrix}$$

Proof. Let $\Psi = LL^T$ and consider the matrices

$$P^{-1} = \begin{bmatrix} 0 & -L^{-1} \\ M & MB\Psi^{-1} \end{bmatrix}, \qquad Q^{-1} = \begin{bmatrix} 0 & N^T \\ L^{-T} & \Psi^{-1}B^TN^T \end{bmatrix}.$$

Then we have

$$P^{-1}\begin{bmatrix} H & B \\ B^T & -\Psi \end{bmatrix}Q^{-1} = \begin{bmatrix} I & 0 \\ 0 & M\widehat{S}N^T \end{bmatrix}$$

where $\widehat{S} = B\Psi^{-1}B^T + H$. By expanding the terms in the expression for $\widehat{S}$, it turns out that $\widehat{S}$ is exactly the Schur complement matrix $\mathcal{S}$ in (8). ∎

The next lemma shows that one whole class of preconditioners for the RAE system (14) results in the same preconditioner for the normal equation (8).

Lemma 6.2 *Consider the RAE system (14). For nonsingular matrices M and N, and a matrix Y of appropriate dimensions, we can construct for K a pair of preconditioners (P, Q) of the form*

$$P^{-1} = \begin{bmatrix} 0 & N \\ M & Y \end{bmatrix} \quad \textit{and} \quad Q^{-1} = \begin{bmatrix} 0 & M^T \\ N^T & Y^T \end{bmatrix} = P^{-T}$$

that will result in a normal equation independent of N and Y.

Proof. It can be easily shown that

$$P^{-1}KQ^{-1} = \begin{bmatrix} -N\Psi N^T & NB^TM^T - N\Psi Y^T \\ MBN^T - Y\Psi N^T & MHM^T + YB^TM^T + MBY^T - Y\Psi Y^T \end{bmatrix}.$$

The preconditioned RAE system is thus given by

$$\begin{bmatrix} -N\Psi N^T & NB^TM^T - N\Psi Y^T \\ MBN^T - Y\Psi N^T & MHM^T + YB^TM^T + MBY^T - Y\Psi Y^T \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} q \\ r \end{bmatrix}$$

where

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = Q \begin{bmatrix} \Delta y \\ \Delta \tilde{x}_1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} q \\ r \end{bmatrix} = P^{-1} \begin{bmatrix} h \\ F_1^{-1/2} g_1 \end{bmatrix}$$

By eliminating Δu we get

$$M(B\Psi^{-1}B^T + H)M^T\Delta v \;=\; (MB\Psi^{-1}N^{-1} - YN^{-1})q + r.$$

Again, simple algebraic manipulations will reveal that the matrix $B\Psi^{-1}B^T + H$ on the left-hand side is exactly $\mathcal{S}$ in (8). □

The preceding lemmas imply that in general, the reduced augmented system offers more freedom in designing effective preconditioners than the normal equation system. Next, we shall study an approach of designing the preconditioner that exploits the block structure of the inverse of the reduced augmented matrix. Before doing that, we shall give an overview of some existing preconditioning methods for an augmented system.

6.1 Existing preconditioners

The purpose of preconditioning a linear system is to enable the resulting preconditioned system to converge to a solution faster than a direct method under a suitable iterative method. To achieve this goal, the preconditioner should approximate the numerical properties of the coefficient matrix such that the preconditioned system needs fewer number of iterative steps to solve than the original system. At the same time, the preconditioning step should be computationally cheap, so that the additional work incurred by preconditioning can be compensated by the decrease in the number of steps required for convergence to a solution of desired accuracy. Over the years, various

types of preconditioners have been proposed for the augmented system and other systems of similar structure. Such systems arise in many areas, for example, constrained optimizations and finite element discretizations of partial differential equations. Practically all of the preconditioning approaches are motivated by the 2 by 2 block structure of the augmented matrix. In this section, we give an overview of a few of them that have demonstrated some positive results for augmented systems that are not too ill-conditioned.

Gill et al [7], Murphy et al [12] and Ipsen [9] studied block preconditioners that yield a preconditioned system with three or fewer distinct eigenvalues when exact Schur complement matrices are employed in the preconditioners. Theoretically, this means that optimal iterative algorithms such as Krylov subspace methods will converge to a solution in at most three iterations. However, preconditioning by the exact Schur complement matrices is computationally as expensive as solving the system by a direct method. Such preconditioners are practical only when good and inexpensive approximations of the Schur complement matrices are available. Unfortunately, such effective approximations for Schur complement matrices exist only in some special cases.

Bergamaschi et al [2] and Keller et al [10] considered block preconditioners that replace the $(1,1)$ block by an appropriate approximation, while keeping the remaining sparser blocks unchanged. They derived some theoretical results on the resulting preconditioned systems, such as eigenvalue distributions and convergence rate under Krylov subspace iterative methods. When diagonal approximation was used for the $(1,1)$ block, the resulting preconditioned iterative solutions had comparable performance with the direct methods for certain classes of problems. In particular, the iterative method outperformed the direct method when the $(1,1)$ block is relatively dense for the strictly convex quadratic programming problems considered in [2]. We should emphasize that the work in [2] is not directly applicable to LP because of the lack of a strictly positive definite quadratic term $\frac{1}{2}x^TQx$ to ameliorate the extreme ill-conditioning of the (1,1) block in the augmented matrix $\mathcal{A}$.

On the other hand, Axelsson and Neytcheva [1] considered a block incomplete factorization preconditioning method and an indefinite block preconditioner on regularized Stokes problem. It was found that the indefinite preconditioner performed better on

the problem considered.

A more general and systematic study of block preconditioners for symmetric indefinite systems of linear equations are given by Toh et al [18]. In their work, they presented theoretical and numerical evaluations of three common types of block preconditioners, namely, the block diagonal, block constraint and block triangular preconditioners. They conducted numerical experiments on the preconditioners in Krylov subspace methods to solve large linear systems arising from finite element discretization of Biot's consolidation equations. The experimental results highlight the need to strike a balance between how well a preconditioner approximates a given matrix, and the computational cost of the preconditioning step. While theoretical results serve as good indicators of the numerical properties of the preconditioned system when sophisticated preconditioners are used, they offer little help in identifying this optimal balance when practical, and often crude, approximations are used. One can only rely on numerical experiments to evaluate the performance of the preconditioners.

Oliveira and Sorensen [14] constructed a preconditioner for the augmented system that reduces the preconditioned system to a positive definite one. Numerical experiments applying the preconditioner on conjugate gradient method verified that the resulting iterative solution can outperform the direct approach using Cholesky factorization on some large-scale LP problems when the Cholesky factors are dense. The preconditioned system performs better as the interior point iterates converge to a solution, which is a desirable property since the linear system usually gets increasingly ill-conditioned as the interior point convergence progresses. However, sophisticated techniques are required for the implementation of their preconditioning approach as it involves nontrivial operations such as selecting a set of linearly independent columns from the constraint matrix according to some criteria, and computing the sparse LU factorization for the resulting submatrix.

All the preconditioning methods surveyed in this section either implicitly assume the $(1,1)$ block of the augmented matrix to be reasonably well-conditioned, or that a good preconditioner for the Schur complement matrix can be found, or they involve some elaborate computation in the preconditioning step. One fact demonstrated by them is that optimal preconditioners are largely problem dependent. In the next section, we

propose to construct a preconditioner based on the RAE system we derived in section 4.

6.2 Preconditioner design for the RAE system

In this section, we shall attempt to construct a preconditioner for the reduced augmented matrix by approximating the block structure of its inverse, as proposed in [18], and also in [2] for quadratic problems.

Consider the block structure of the inverse of K:

$$K^{-1} \;=\; \begin{bmatrix} H^{-1} - H^{-1}BS^{-1}B^TH^{-1} & H^{-1}BS^{-1} \\ S^{-1}B^TH^{-1} & -S^{-1} \end{bmatrix} \tag{22}$$

where S is the negative of the Schur complement matrix of K, that is,

$$S \;=\; B^TH^{-1}B + \Psi. \tag{23}$$

This naturally leads us to consider a preconditioner P_c with the following block structure

$$P_c^{-1} \;=\; \begin{bmatrix} \widehat{H}^{-1} - \widehat{H}^{-1}B\widehat{S}^{-1}B^T\widehat{H}^{-1} & \widehat{H}^{-1}B\widehat{S}^{-1} \\ \widehat{S}^{-1}B^T\widehat{H}^{-1} & -\widehat{S}^{-1} \end{bmatrix} \tag{24}$$

where $\widehat{H}$ and $\widehat{S}$ are positive definite approximations of H and S, respectively.

Therefore, if $\widehat{H}$ and $\widehat{S}$ are effective approximations of H and S which are relatively simple to compute, the resulting preconditioned system could be expected to perform better than the original system under an iterative solution method.

Theorem 1 *(a) Suppose $\widehat{S} = B^T\widehat{H}^{-1}B + \Psi$ is used in P_c. Let n_1 be the dimension of D_1 defined in (10). Then $P_c^{-1}K$ has n_1 eigenvalues clustered at* 1. *The remaining m real eigenvalues are those of the matrix*

$$Y := G \,+\, (I - G)\widehat{H}^{-1/2}H\widehat{H}^{-1/2}, \tag{25}$$

where

$$G \;\;=\;\; \widehat{H}^{-1/2}B\,\widehat{S}^{-1}\,B^T\widehat{H}^{-1/2}.$$

(b) Suppose $\widehat{H} = H$ is used in the preconditioner P_c. Then $P_c^{-1}K$ has 1 *as an eigenvalue with multiplicity m. The remaining n_1 eigenvalues are those of the matrix $\widehat{S}^{-1}S$.*

Proof. It is easily verified that

$$P_c^{-1}K \;=\; \begin{bmatrix} \widehat{H}^{-1}H - \widehat{H}^{-1}B\widehat{S}^{-1}B^T(\widehat{H}^{-1}H - I) & \widehat{H}^{-1}B\widehat{S}^{-1}(\widehat{S} - B^T\widehat{H}^{-1}B - \Psi) \\ \widehat{S}^{-1}B^T(\widehat{H}^{-1}H - I) & \widehat{S}^{-1}(B^T\widehat{H}^{-1}B + \Psi) \end{bmatrix}. \quad (26)$$

(a) Since $\widehat{S} = B^T\widehat{H}^{-1}B + \Psi$, the (2,2) block in (26) reduces to I_{n_1}. Thus

$$P_c^{-1}K \;=\; \begin{bmatrix} \widehat{H}^{-1}H - \widehat{H}^{-1}B\widehat{S}^{-1}B^T(\widehat{H}^{-1}H - I) & 0 \\ \widehat{S}^{-1}B^T(\widehat{H}^{-1}H - I) & I_{n_1} \end{bmatrix}. \quad (27)$$

It is clear that $P_c^{-1}K$ has 1 as an eigenvalue with multiplicity n_1, and the remaining eigenvalues are determined by its $(1,1)$ block. Note that the $(1,1)$ block of above preconditioned matrix is similar to

$$G \;+\; (I - G)\widehat{H}^{-1/2}H\widehat{H}^{-1/2}. \quad (28)$$

(b) When $\widehat{H} = H$, the preconditioned matrix in (26) becomes

$$P_c^{-1}K \;\;=\;\; \begin{bmatrix} I_m & H^{-1}B(I - \widehat{S}^{-1}S) \\ 0 & \widehat{S}^{-1}S \end{bmatrix}. \quad (29)$$

With the above expression, it is easy to see that the result stated in the theorem holds true. □

Next, we study the eigenvalue distributions of the preconditioned reduced augmented matrix using simple diagonal approximations for the $(1,1)$ block.

Proposition 6.1 *Suppose we take $\widehat{H} = \mu I$ and $\widehat{S} = B^T\widehat{H}^{-1}B + \Psi$. Then $G = U_1(I + O(\mu^2))U_1^T$, and the matrix*

$$Y = G \;+\; (I - G)H/\mu \quad (30)$$

has n_1 real eigenvalues clustered at $1 + O(\mu)$ and the remaining $m - n_1$ real eigenvalues are those of $W_{22} + O(\mu^2)$.

Proof. For $\widehat{H} = \mu I$,

$$G \;\;=\;\; B\,(B^TB + \mu\Psi)^{-1}\,B^T.$$

It is readily shown by using the SVD of B that

$$(B^TB + \mu\Psi)^{-1} \;=\; V \begin{bmatrix} \Sigma_1^2 + \mu\tilde{\Psi}_{11} & \mu\tilde{\Psi}_{12} \\ \mu\tilde{\Psi}_{12}^T & \mu\tilde{\Psi}_{22} \end{bmatrix}^{-1} V^T$$

where $\tilde{\Psi}_{ij} = V_i^T \Psi V_j$. Note that $\Psi = \mu\tilde{D}_1 E_1^{-1}$. Thus

$$\begin{aligned} G &= [U_1\Sigma_1,\ 0] \begin{bmatrix} \Sigma_1^2 + \mu\tilde{\Psi}_{11} & \mu\tilde{\Psi}_{12} \\ \mu\tilde{\Psi}_{12}^T & \mu\tilde{\Psi}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \Sigma_1 U_1^T \\ 0 \end{bmatrix} \\ &= U \begin{bmatrix} I + O(\mu^2) & 0 \\ 0 & 0 \end{bmatrix} U^T \end{aligned}$$

It follows that

$$\begin{aligned} U^TYU &= \begin{bmatrix} I + O(\mu^2) & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} O(\mu) & 0 \\ 0 & I/\mu \end{bmatrix} \begin{bmatrix} \Sigma_1^2 + \mu W_{11} & \mu W_{12} \\ \mu W_{12}^T & \mu W_{22} \end{bmatrix} \\ &= \begin{bmatrix} I + O(\mu)\Sigma_1^2 & O(\mu^2) \\ W_{12}^T & W_{22} \end{bmatrix}. \end{aligned}$$

The required result follows readily from the above expression. □

Proposition 6.2 *Suppose we take* $\widehat{H} = \mathrm{diag}(H)$ *and* $\widehat{S} = B^T\widehat{H}^{-1}B + \Psi$. *The matrix*

$$Y = G \,+\, (I - G)\widehat{H}^{-1/2}H\widehat{H}^{-1/2} \tag{31}$$

has n_1 *positive real eigenvalues clustered at* $1 + O(\mu)$, *and the remaining* $m - n_1$ *real eigenvalues are* $O(\mu)$.

Proof. Let $J = B^T\widehat{H}^{-1/2}$. Then

$$G \;=\; J^T(JJ^T + \Psi)^{-1}J,$$

which has the same form as the matrix in part (b) of lemma 4.4. Let $P = [P_1\ P_1^{\perp}]$, where the columns of P_1 form an orthonormal basis of $\mathcal{R}(J^T)$ and they form the right

singular vectors of J. The columns of $P_1^{\perp}$ form an orthonormal basis of $\mathcal{N}(J)$. Thus G can be expressed in the form $G = P_1(I + O(\mu))P_1^T$. It is easily deduced that

$$P^T Y P = \begin{bmatrix} I + O(\mu) & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} O(\mu) & 0 \\ 0 & I \end{bmatrix} P^T \widehat{H}^{-1/2} H \widehat{H}^{-1/2} P.$$

Now, recall that $H = BB^T + \mu\tilde{W}$. Hence,

$$\begin{aligned} P^T \widehat{H}^{-1/2} H \widehat{H}^{-1/2} P &= P^T (J^T J) P + \mu P^T (\widehat{H}^{-1/2} \tilde{W} \widehat{H}^{-1/2}) P \\ &= \begin{bmatrix} \Sigma_1^2 + O(\mu) & O(\mu) \\ O(\mu) & O(\mu) \end{bmatrix}, \end{aligned}$$

where Σ_1 is the diagonal matrix of positive singular values of J. Adding up the terms of $P^T Y P$ thus yields the required result. □

Corollary 6.1 *Suppose we take $\widehat{H} = \mu \mathrm{diag}(H)$ and $\widehat{S} = B^T \widehat{H}^{-1} B + \Psi$. The matrix*

$$Y = G + (I - G)\widehat{H}^{-1/2} H \widehat{H}^{-1/2} \tag{32}$$

has n_1 positive real eigenvalues clustered at $1 + O(\mu)$, and the remaining $m - n_1$ real eigenvalues are $O(1)$.

Remark 6.1 *The preceding results show that using $\widehat{H} = \mathrm{diag}(H)$ will cluster the eigenvalues of the preconditioned matrix at 1 and $O(\mu)$ as IPM converges to a solution, whereas using $\widehat{H} = \mu I$ and $\widehat{H} = \mu \mathrm{diag}(H)$ yields an eigenvalue distribution that is independent of μ when the IPM converges (assuming that we approximate S by $\widehat{S} = B^T \widehat{H}^{-1} B + \Psi$ in both cases).*

7 Two-Phase Iterative Algorithm Design

From the theoretical results in the preceding section, we observe that the spectral properties of the preconditioned reduced augmented matrix $P_c^{-1} K$ is likely to improve as the interior-point iterates approach a solution. This desirable property translates to a faster convergence rate when the preconditioned linear system is solved by iterative techniques such as the Krylov subspace methods, when the IPM is converging

to a solution. In contrast, when a Krylov subspace method is applied to a system of preconditioned augmented or normal equations, it typically takes very few iterations to converge in the initial steps of the IPM. However, as the IPM is converging to a solution, the Krylov subspace method typically takes an increasing number of iterations to converge.

Capitalizing on this potential of the preconditioned RAE, we propose a two-phase iterative algorithm to solve for the Newton step directions of a given LP problem in the IPM. In the first phase, we apply an established technique such as the preconditioned conjugate gradient (PCG) method to solve the normal equations. Toward the end of the IPM solution process, when it is advantageous to employ the preconditioned RAE, we switch to phase two, which applies the symmetric quasi-minimal residual (SQMR) method [6] to solve the preconditioned RAE system.

As an illustration, we attempted to solve the problem FIT2P taken from the NETLIB collection of LP problems [13] with two iterative approaches. In the first approach, we used only PCG to solve the normal equations that arose at each IPM step, and in the second approach, we used the two phase iterative algorithm. We will give more details on the preconditioners used and other experimental conditions in the next section. The results for FIT2P are tabulated in Table 1.

Table 1: Iteration counts for iterative solutions of FIT2P. Algorithm switched phase at step 17.

IPM	PCG		Two-phase	
step	predictor	corrector	predictor	corrector
1	29	29	29	29
2	6	9	6	9
3	7	7	7	7
4	13	11	13	11
⋮	⋮	⋮	⋮	⋮
15	36	36	36	36
16	41	41	41	41
17	49	47	107	97
18	53	54	70	74

Table 1: Iteration counts for iterative solutions of FIT2P. Algorithm switched phase at step 17.

IPM	PCG		Two-phase	
step	predictor	corrector	predictor	corrector
19	59	57	21	25
20	79	85	14	12
⋮	⋮	⋮	* Converged *	
25	294	260		
	* Not converged *			

Indeed, as shown in Table 1, the number of PCG iterations taken to solve the normal equations generally increases as the IPM converges to a solution. On the other hand, when the two-phase algorithm switches to the RAE system (which occurs at the 17th IPM step), the number of SQMR iterations taken to solve the preconditioned RAE system generally decreases as the IPM solution converges. Furthermore, while the IPM using the two-phase iterative algorithm has successfully converged to a solution, the PCG approach has failed.

Two important implementational issues associated with the proposed two-phase iterative algorithm are: (1) how to determine the right condition to switch from phase one to phase two, and (2) how to partition the matrix D into D_1 and D_2. We shall address these issues in the following subsections, before presenting the implementation of the algorithm in detail.

7.1 Switching criterion

There are a few reasonable switching criteria to consider:

1. Switch when the duality gap falls below a certain threshold.
2. Switch when phase 1 takes excessive number of PCG iterations to converge.
3. Switch when the elements of matrix D form 2 distinct clusters.

After experimenting on the test data from the NETLIB collection, we found that both criteria 2 and 3 gave consistent performance over the various test problems. Furthermore, by combining the two criteria, we achieved even better results generally than using either one of the criteria alone. Hence, in our computational experiments, we adopted the strategy of switching from phase one to phase two when either criterion 2 or criterion 3 is met. While the performance might turn out better for some particular problem instances when only one of the criteria is used, this strategy is able to achieve satisfactory performance for a reasonably wide range of problems.

7.2 Partitioning of matrix D

When the elements of D form two distinct clusters of different orders of magnitude, the obvious partition is to assign the cluster with smaller value to D_1 and the other cluster to D_2. However, sometimes no obvious cluster can be determined. In that case, one possible partition is to assign elements in D that are less than 1 to D_1, and assign the rest to D_2. This simple strategy works fine on most of the NETLIB LP problems, and thus we adopted it in our computational experiments.

7.3 Implementation

We implemented our two-phase iterative algorithm on the MATLAB-based software package LIPSOL (version 0.6) [23], which is a direct solver for LP problems based on predictor-corrector primal-dual IPM. In the implementation, we replaced the direct solver in LIPSOL with our two-phase iterative algorithm. The algorithm switches from phase one to phase two when the switching condition is satisfied. In the following, we describe a very simple procedure that we used to determine the switching condition.

At the kth IPM step, we assign the n_1 diagonal elements in D that are smaller than one to form the diagonal of D_1 and the remaining n_2 elements to form the diagonal of D_2. Then we compute the logarithm of their geometric means, that is,

$$\bar{d}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1}\log_{10}(D_{1i}), \quad \text{and} \quad \bar{d}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2}\log_{10}(D_{2i})$$

where D_{1i} and D_{2i} are, respectively, the ith component of the diagonal of D_1 and

D_2. In addition, if the algorithm is in phase one, we save the total number of PCG iterations p_k taken to converge in the current IPM step. The switching condition is deemed to be satisfied when $\bar{d}_2 - \bar{d}_1 > 8$, or when $p_{k-1} > \max(120, 0.02m)$, where m is the dimension of the normal matrix. In other words, the switching is triggered if the gap between the two clusters in D is large enough, or if the PCG takes too long to converge in the previous IPM step.

We can thus summarize our algorithm as follows:

Step 1. Initialize the IPM. Set p_0 to zero.

Step 2. At the kth IPM step, partition the elements of D into D_1 and D_2.

Step 3. Compute the geometric means $\bar{d}_1$ and $\bar{d}_2$.

Step 4. If $\bar{d}_2 - \bar{d}_1 < 8$ and $p_{k-1} < \max(120, 0.02m)$, then

Phase 1: Use PCG to solve the normal equations (8) for the search directions;

Save the total number of PCG iterations in p_k.

else

Phase 2: Use SQMR to solve the preconditioned RAE for the search directions.

Step 5. If the IPM converges, then stop the IPM iteration. Otherwise, proceed to the next IPM step to compute the next search directions.

In the next section, we present the computational results of testing the algorithm on a selection of LP problems.

8 Computational Results and Discussions

We have tested our two-phase iterative algorithm on three sets of LP problems. The first set consists of some larger problems of the NETLIB suite, the second set consists of a series of multi-commodity problems of various sizes, while the third set is a collection of highly degenerate test problems. Both the second and the third problem sets are available online at ftp://plato.asu.edu/pub/lptestset/pds/ and http://www.pmf.ni.ac.yu/people/pecko/linprog/linprog.htm, respectively.

Here, we present our computational results to demonstrate our algorithm's capability to solve these three sets of LP problems efficiently, and to compare its performance

with direct solvers based on these problems. All the numerical experiments were carried out in MATLAB version 6.5 on an i686 2.4 GHz processor and 1 GB of RAM. In the following sections, we give more details on the experiments and the results.

8.1 Preconditioning in the two-phase iterative algorithm

Without loss of generality, let the coefficient matrix A be partitioned into sparse and dense columns as $A = [A_s \; A_d]$. Let the corresponding partition of the diagonal matrix D be $D = \text{diag}(D_s, D_d)$. Consider the matrix P_S formed by the part of the Schur complement matrix contributed by A_s and D_s, that is,

$$P_S = A_s D_s^{-1} A_s^T.$$

We compared the performance of the two-phase algorithm when P_S and incomplete Cholesky preconditioner P_{ICT} (with relative drop tolerance $\tau = 10^{-3}$) are used to precondition the PCG in phase one. It turned out that P_{ICT} yielded better results for the multi-commodity problems, whereas P_S generally performed better on the other two sets of problems. Hence, in phase one of the algorithm, we adopted P_{ICT} as the preconditioner for the multi-commodity problems and P_S as the preconditioner for the other two sets of problems. We should emphasize that our purpose of the numerical experiments is not on evaluating the preconditioners for the Schur complement matrix but on evaluating the viability of the RAE approach in computing search directions.

The preconditioner P_c for the reduced augmented matrix and its properties have been discussed in section 6.2. We consider the following variations of $\widehat{H}$ and $\widehat{S}$:

1. $\widehat{H} = \text{diag}(H)$ and $\widehat{S} = B^T \widehat{H}^{-1} B + \Psi$;
2. $\widehat{H} = \text{diag}(H)$ and $\widehat{S} = \text{diag}(B^T \widehat{H}^{-1} B + \Psi)$;
3. $\widehat{H} = H$ and $\widehat{S} = B^T \text{diag}(\widehat{H})^{-1} B + \Psi$;
4. $\widehat{H} = H$ and $\widehat{S} = \text{diag}(B^T \text{diag}(\widehat{H})^{-1} B + \Psi)$;
5. $\widehat{H} = LL^T$ and $\widehat{S} = B^T \text{diag}(\widehat{H})^{-1} B + \Psi$;
6. $\widehat{H} = LL^T$ and $\widehat{S} = B^T \widehat{H}^{-1} B + \Psi$;

where L is the incomplete Cholesky factor of H computed with relative drop tolerance 10^{-3}. After evaluating their performance with a few LP problems, it became clear that option (1) yields the best performance in terms of computational time. While option (6) yields the least number of iterations taken to converge, option (1) outperforms it because it takes significantly less computational effort in the preconditioning step. This underscores the fact that, to construct an effective preconditioner for a given matrix, a balance needs to be struck between how well a preconditioner approximates the matrix, and the computational cost of the preconditioning step.

In addition to the above options of $\widehat{H}$, we also found that using $\widehat{H} = \sqrt{\mu}I$ yielded very similar performance as $\widehat{H} = \text{diag}(H)$. On the other hand, the approximation $\widehat{H} = \mu I$ seemed to be less robust than $\widehat{H} = \sqrt{\mu}I$ and $\widehat{H} = \text{diag}(H)$. The reason that the former performs worse than the latter is because the resulting preconditioner P_c is very ill-conditioned and in finite precision arithmetic, the computed inverse of P_c is not a close approximation to the true inverse. We hence chose preconditioner P_c with $\widehat{H} = \text{diag}(H)$ and $\widehat{S} = B^T\widehat{H}^{-1}B + \Psi$ for the RAE system in phase two of the algorithm. To apply the preconditioner P_c in a Krylov subspace method, the preconditioning step $P_c^{-1}[u; v]$ can be computed efficiently as follows:

$$
\begin{aligned}
&\text{Compute} \quad w = \widehat{H}^{-1}u; \\
&\text{Compute} \quad z = \widehat{S}^{-1}(B^T w - v); \\
&\text{Compute} \quad P_c^{-1}\,[u\,;\,v] = \left[\widehat{H}^{-1}(u - Bz)\,;\,z\right].
\end{aligned}
$$

We avoid explicitly computing $\widehat{S}^{-1}$ by pre-computing the sparse Cholesky factorization of the $n_1 \times n_1$ matrix $\widehat{S}$, and then solve the resulting linear system each time we need to evaluate $z = \widehat{S}^{-1}(B^T w - v)$ in the preconditioning step. Moreover, if the indices of D that are partitioned into D_1 and D_2 in the current IPM step are identical to those in the previous IPM step, then the symbolic factors of $\widehat{S}$ in the previous step may be reused in the sparse Cholesky factorization of the current $\widehat{S}$. This situation tends to occur more frequently as the IPM converges to a solution, and thus reusing the symbolic factors will help cut down the computational effort required in the preconditioning step of the reduced augmented matrix.

8.2 Experimental results

Table 2 and Table 3 display the main statistics of the first and second problem sets after preprocessing by LIPSOL. The first set consists of a selection of larger problems of the NETLIB collection. The second set is a collection of multi-commodity problems with 11 commodities. The suffix of a problem name indicates the number of days being modelled in that problem, for example, PDS-06 models six days, PDS-30 models thirty days and so on. Consequently, the problem dimension ranges from a few thousands to hundreds of thousands. These problems are selected to demonstrate our algorithm's capability in solving LP problems of large dimensions efficiently.

Table 2: Main statistics of some LP problems from NETLIB.

Problem	rows	columns	nonzeros	density
BNL2	2268	4430	14914	1.4844e-03
D2Q06C	2171	5831	33081	2.6132e-03
DEGEN3	1503	2604	25432	6.4980e-03
DFL001	6071	12230	35632	4.7990e-04
FIT2P	3000	13525	50284	1.2393e-03
MAROS-R7	3136	9408	144848	4.9095e-03
PILOT	1441	4657	42300	6.3033e-03
PILOT87	2030	6460	72479	5.5269e-03
SCTAP3	1480	3340	9734	1.9692e-03
SIERRA	1222	2715	7951	2.3965e-03
WOODW	1098	8418	37487	4.0557e-03
CRE-A	3428	7248	18168	7.3122e-04
CRE-C	2986	6411	15977	8.3460e-04
KEN-11	11548	18203	42161	2.0057e-04
KEN-13	23393	37420	84909	9.6998e-05

Table 3: Main statistics of multi-commodity problems.

Problem	rows	columns	nonzeros	density
PDS-02	2788	7551	16230	7.7094e-04
PDS-06	9617	29087	62582	2.2372e-04
PDS-10	16239	49613	106802	1.3256e-04
PDS-20	33250	107627	231155	6.4594e-05
PDS-30	49144	157845	338852	4.3683e-05
PDS-40	65714	216604	464165	3.2610e-05
PDS-50	81823	274800	587936	2.6148e-05
PDS-60	98104	335321	716400	2.1778e-05
PDS-70	114717	390005	833465	1.8629e-05
PDS-80	128954	434580	927826	1.6556e-05
PDS-90	142596	475448	1014136	1.4958e-05

In the experiment, we used the default convergence criterion in LIPSOL for the IPM, which was

$$\varepsilon_{\text{IPM}} = \max\left(\frac{\|r_p\|}{1+\|b\|}, \frac{\|r_d\|}{1+\|c\|}, \frac{x^T z}{n}\right) < 10^{-8},$$

where r_p and r_d are the residual vectors defined in (6), and n is the dimension of x. Given a linear system $\tilde{A}\tilde{x} = \tilde{b}$, let $\tilde{x}^k$ be the computed value for $\tilde{x}$ at the kth iteration of an iterative solution. The PCG and SQMR methods in our two-phase algorithm would stop at the kth iteration if

$$\varepsilon_{\text{IT}} = \|\tilde{b} - \tilde{A}\tilde{x}^k\| < \max\left(10^{-3}\|\tilde{b}\|, 10^{-8}\right).$$

We have generated three sets of results using the sparse Cholesky solver in LIPSOL, our two-phase iterative algorithm, and the PCG method, as summarized in Tables 4 and 5. Each table shows the computational time in seconds (under column 'ctime'), the number of IPM iterations (under column 'it.') required to converge to an optimal solution, as well as the optimal objective values (under column 'op. val'). We should mention that it makes sense to use the direct approach for solving LP as the benchmark

for comparing the performance of our two-phase iterative approach. This is because the direct approach based on sparse Cholesky factorization is highly optimized as it has been researched and developed for more than 20 years.

On the whole, the results indicate that the direct method is generally faster than the two-phase iterative algorithm for the smaller problems, while the converse is true for the larger problems. Both methods take similar number of IPM steps to converge to an optimal solution. The results in Table 4 show that, with the exception of CRE-C, the two-phase iterative algorithm is competitive with the direct method for the selected NETLIB problems. In fact, it even performs better than the direct method in solving problems DFL001 and MAROS-R7.

On the other hand, the results in Table 5 clearly demonstrate the advantages of the two-phase iterative algorithm over the direct approach. Observe that, for this set of problems, the two-phase algorithm is slower than the direct method for the two smallest problems PDS-02 and PDS-06. However, it starts to outperform the latter from PDS-10 onward, and the computational speed advantage increases with the problem size: it is only slightly faster than the direct method in solving PDS-10, about 4 times as fast for PDS-30, and almost 9 times as fast for PDS-70. Furthermore, the direct method runs out of memory for PDS-80 and above. In contrast, the two-phase iterative algorithm is able to solve all the problems.

The PCG method performs well for most of the NETLIB problems in Table 4, except for DFL001 and FIT2P, where it fails to converge due to ill-conditioning in the final IPM steps. For the PDS problems, however, it performs badly on its own.

Based on the experimental results, we thus conclude that the two-phase iterative algorithm is competitive with the direct method in solving some large-scale LP problems. It even has the potential to greatly outperform the direct method in both computational speed and memory requirement for solving very large problems with hundreds of thousands of variables. As mentioned before, the direct approach is highly optimized, whereas our algorithm can still be refined.

Table 4: Computational results for NETLIB problems. An asterisk * indicates failure to converge.

	sparse Cholesky			Two-phase			PCG(P_S)		
Problem	ctime	it.	op. val	ctime	it.	op. val	ctime	it.	op. val
BNL2	2.76	31	1.8112365420e+03	4.62	31	1.8112365328e+03	2.37	31	1.8112365420e+03
D2Q06C	6.31	32	1.2278421082e+05	7.03	32	1.2278421042e+05	5.59	32	1.2278421038e+05
DEGEN3	4.27	20	-9.8729399275e+02	5.26	20	-9.8729399997e+02	7.68	24	-9.8729399996e+02
DFL001	*	*	*	536	49	1.1266396035e+07	*	*	*
FIT2P	8.44	21	6.8464293294e+04	9.54	20	6.8464293315e+04	*	*	*
MAROS-R7	31.59	15	1.4971851665e+06	26.25	15	1.4971851665e+06	30.34	15	1.4971851665e+06
PILOT	9.68	31	-5.5748970661e+02	10.41	31	-5.5748971074e+02	10.40	36	-5.5748972935e+02
PILOT87	31.21	37	3.0171038591e+02	36.88	37	3.0171038569e+02	29.79	37	3.0171038591e+02
SCTAP3	0.59	18	1.4240000000e+03	0.62	18	1.4240000000e+03	0.56	18	1.4240000000e+03
SIERRA	0.75	17	1.5394362184e+07	0.88	17	1.5394362184e+07	0.76	17	1.5394362184e+07
WOODW	3.10	28	1.3044763953e+00	6.82	29	1.3044763333e+00	2.70	28	1.3044763953e+00
CRE-A	2.31	30	2.3595407061e+07	5.58	30	2.3595406913e+07	2.13	30	2.3595406953e+07
CRE-C	2.05	30	2.5275116141e+07	21.0	30	2.5275116138e+07	9.80	34	2.5275116141e+07
KEN-11	9.28	22	-6.9723822622e+09	19.3	22	-6.9723822622e+09	9.10	22	-6.9723822618e+09
KEN-13	28.5	27	-1.0257394790e+10	81.1	27	-1.0257394790e+10	27.8	27	-1.0257394779e+10

Table 5: Computational results for PDS problems. An asterisk * indicates failure to converge. A dash - indicates insufficient memory for the computation.

	sparse Cholesky			Two-phase			PCG(P_{ICT})		
Problem	ctime	it.	op. val	ctime	it.	op. val	ctime	it.	op. val
PDS-02	2.48	29	2.8857862010e+10	4.44	29	2.8857862010e+10	*	*	*
PDS-06	38.9	42	2.7761037600e+10	51.7	43	2.7761037600e+10	*	*	*
PDS-10	181	52	2.6727094976e+10	135	53	2.6727094976e+10	*	*	*
PDS-20	1902	67	2.3821658640e+10	1080	69	2.3821658643e+10	*	*	*
PDS-30	5877	69	2.1385445736e+10	1394	70	2.1385445736e+10	*	*	*
PDS-40	15603	66	1.8855198824e+10	2653	68	1.8855198824e+10	*	*	*

Table 5: Computational results for PDS problems. An asterisk * indicates failure to converge. A dash - indicates insufficient memory for the computation.

	sparse Cholesky			Two-phase			PCG(P_{ICT})		
Problem	ctime	it.	op. val	ctime	it.	op. val	ctime	it.	op. val
PDS-50	29261	72	1.6603525724e+10	5415	74	1.6603525724e+10	*	*	*
PDS-60	47187	72	1.4265904407e+10	8168	71	1.4265904407e+10	*	*	*
PDS-70	85757	84	1.2241162812e+10	9594	83	1.2241162812e+10	*	*	*
PDS-80	-	-	-	11646	78	1.1469077462e+10	*	*	*
PDS-90	-	-	-	12801	73	1.1087561635e+10	*	*	*

In the next experiment, we compare the performance of the two-phase iterative algorithm with two other direct solvers on a set of highly degenerate LP problems which is available online at http://www.pmf.ni.ac.yu/ people/pecko/linprog/linprog.htm. Stanimirovic et al [17] mentioned that these test problems cause serious numerical difficulties in established LP solvers such as PCx [3] and HOPDM [8]. The two direct solvers we used for comparison were LIPSOL and LOQO [21]. The optimality criteria for LOQO was the default value set in LOQO, which were that the primal and dual objective values agree to 8 significant figures, and that the primal and dual are feasible to the level of 10^{-6} in relative error. Table 6 presents the number of IPM iterations taken by LIPSOL, LOQO and the two-phase iterative algorithm to converge, as well as the optimal objective values attained. A dash '-' indicates failure to converge to a solution.

From the results, it is clear that the two-phase iterative algorithm is able to solve most of the highly degenerate problems, while LIPSOL and LOQO failed to solve many of them. Hence, they verified the theoretical results that the preconditioned RAE is less likely to be affected by ill-conditioning as the IPM converges.

Table 6: Comparison of number of IPM iterations taken by three different methods. A dash '-' indicates failure to converge to a solution.

		LIPSOL		LOQO		Two-phase	
Problem	dimensions	it.	op. val	it.	op. val	it.	op. val
07-20-02	9×22	-	-	-	-	11	-4.3064355850e-04
10-20-06	10×20	6	2.3824176224e-11	-	-	6	3.2466363333e-13
10-20-11	12×20	-	-	-	-	12	-3.6297943443e-03
10-20-12	12×20	-	-	19	-1.648554003e-08	7	8.8671810112e-13
10-20-13	12×22	-	-	16	1.734362793e-07	7	1.4363799708e-12
15-30-03	17×32	6	4.0479393466e-10	37	5.336360138e-12	6	8.1024963022e-12
15-30-04	16×31	-	-	-	-	13	8.3732604980e-04
15-30-06	16×31	-	-	-	-	10	1.5613813978e-13
15-30-07	17×32	-	-	-	-	23	2.1362304688e-04
15-60-09	16×61	-	-	-	-	19	2.0275116174e-03
20-40-05	22×42	-	-	-	-	20	4.3007612257e-02
30-60-05	30×59	9	-1.1729041376e-04	-	-	8	1.3038516047e-08
Hager	18×27	-	-	24	7.544225756e-08	11	1.0994492332e-08
Hager1	18×27	-	-	23	0.07999999954	11	8.0000000030e-02
Hager2	18×27	-	-	24	3.84170562e-09	21	0.0000000000e-00
Little	3×5	6	3.1369501780e-14	12	3.16467637e-09	6	2.5527129588e-10

9 Conclusions

We have proposed to compute the search direction at each interior-point iteration for a linear program via a reduced augmented system that has a smaller dimension than the original augmented system. This reduced system is potentially less susceptible to the ill-conditioning effect caused by the vastly varying magnitudes in the diagonal elements of D, the $(1,1)$ block of the augmented matrix. A preconditioner is then designed by approximating the block structure of the inverse of the transformed matrix to further improve the spectral properties of the transformed system. The resulting

preconditioned system is likely to become better conditioned toward the end of the interior-point algorithm.

Capitalizing on the special spectral properties of the transformed matrix, we further proposed a two-phase iterative algorithm that starts by solving the normal equations with PCG in each IPM iteration. As the elements in the matrix D forms two distinct clusters of different magnitudes as the interior point iterates converge to a solution, the two-phase algorithm switches to solve the preconditioned reduced augmented system with SQMR when the magnitude difference is large enough. A simple heuristic has also been developed to determine the point to switch from phase one to phase two.

The experimental results have demonstrated the potential of our proposed method in solving large-scale LP problems as well as highly degenerate LP problems. In particular, we have seen that it is competitive with sparse Cholesky method in terms of computational time and memory requirement taken to solve the large problems. It also outperforms the direct methods in solving some highly degenerate problems. There is still some room to improve the efficiency of the algorithm by refining the numerical computation, such as the handling of dense rows in the matrix involved in SQMR.

Finally, the next step that we can explore to take this iterative approach to the next level may include the following areas: (a) Devise a better switching condition so as to improve the performance by better utilising the advantages of phase one and two; (b) Develop a better preconditioner for RAE to improve the spectral properties at similar, if not reduced computational cost; (c) Develop more effective preconditioner for the Schur complement matrix S used in phase one of our two-phase iterative algorithm.

Acknowledgement

The authors are grateful to Professor A.R.L. Oliveira for his help in accessing the collection of multi-commodity test problem data. We also thank the Referees for their thorough review of the manuscript and constructive comments which have helped to improve the paper.

References

[1] O. Axelsson and M. Neytcheva, *Preconditioning methods for linear systems arising in constrained optimization problems*, Numerical Linear Algebra with Applications, 10, 3-31, 2003.

[2] L. Bergamaschi, J. Gondzio and G. Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications, 28, 149-171, 2004.

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

[4] R. Fourer and S. Mehrotra, *Solving symmetric indefinite systems in an interior-point method for linear programming*, Mathematical Programming, 62, 15-39, 1993.

[5] R.W. Freund and F. Jarre, *A QMR-based interior-point algorithm for solving linear programs*, Mathematical Programming, 76, 183-210, 1997.

[6] R.W. Freund and N.M. Nachtigal, *A new Krylov-subspace method for symmetric indefinite linear systems*, Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics, W.F. Ames (Ed.), IMACS, 1253-1256, 1994.

[7] P.E. Gill, W. Murray, D.B. Ponceleón and M.A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM Journal on Matrix Analysis and Applications, 13, 292-311, 1992.

[8] J. Gondzio, *HOPDM (version 2.12) - A fast LP solver based on a primal-dual interior point method*, European Journal of Operational Research 85, 221-225, 1995.

[9] I.C.F. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM Journal on Scientific Computing, 23, 1050-1051, 2001.

[10] C. Keller, N.I.M. Gould and A.J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM Journal on Matrix Analysis and Applications, 21, 1300-1317, 2000.

[11] I.J. Lustig, R.E. Marsten and D.F. Shanno, *Interior point methods for linear programming: computational state of the art*, ORSA Journal on Computing, 6, 1-14, 1994.

[12] M.F. Murphy, G.H. Golub and A.J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM Journal on Scientific Computing, 21, 1969-1972, 2000.

[13] *NETLIB linear programming library*, available online at http://www.netlib.org/lp/.

[14] A.R.L. Oliveira and D.C. Sorensen, *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra and its Applications, 394, 1-24, 2005.

[15] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12, 617-629, 1975.

[16] Y. Saad, *Iterative methods for sparse linear systems*, PWS Publishing Company, Boston, 1996.

[17] P.S. Stanimirovic, N.V. Stojkovic and V.V. Kovacevic-Vujcic, *Stabilization of Mehrotra's primal-dual algorithm and its implementation*, IMA Workshop 5: Semidefinite Programming and Robust Optimization, Winter 2003.

[18] K.C. Toh, K.K. Phoon and S.H. Chan, *Block preconditioners for symmetric indefinite linear systems*, International Journal for Numerical Methods in Engineering, 60, 1361-1381, 2004.

[19] L.N. Trefethen and D. Bau, *Numerical linear algebra*, SIAM, 1997.

[20] R.J. Vanderbei, *Symmetric Quasidefinite Matrices*, SIAM Journal of Optimization, 5, 100-113, 1995.

[21] R.J. Vanderbei, *LOQO user's manual - version 4.05*, Department of Operations Research and Financial Engineering, Princeton University, 1999.

[22] S.J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, 1997.

[23] Y. Zhang, *User's guide to LIPSOL: linear-programming interior point solver v0.4*, Optimization Methods and Software, 12, 385-396, 1999.