Properties of a class of preconditioners for weighted least squares problems*

Venansius Baryamureeba[†] Trond Steihaug[‡] Yin Zhang[§]

April 30, 1999
(Revised July 6, 1999)

Abstract

A sequence of weighted linear least squares problems arises from interior-point methods for linear programming where the changes from one problem to the next are the weights and the right hand side. One approach for solving such a weighted linear least squares problem is to apply a preconditioned conjugate gradient method to the normal equations where the preconditioner is based on a low-rank correction to the Cholesky factorization of a previous coefficient matrix. In this paper, we establish theoretical results for such preconditioners that provide guidelines for the construction of preconditioners of this kind. We also present preliminary numerical experiments to validate our theoretical results and to demonstrate the effectiveness of this approach.

Key Words. Weighted linear least squares, Preconditioner, Preconditioned conjugate gradient method, Linear programming, interior-point algorithms.

*This is Technical Report No. 170, Department of Informatics, University of Bergen, N-5020 Bergen, Norway and Technical Report No. TR99-16, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005, USA.

[†]Department of Informatics, University of Bergen, N-5020 Bergen, Norway.

[‡]Department of Informatics, University of Bergen, N-5020 Bergen, Norway. This author was supported in part by the Research Council of Norway.

[§]Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005, USA. This author was supported in part by DOE Grant DE-FG03-97ER25331.

1 Introduction

This paper concerns solving linear systems of equations arising from interior-point algorithms for linear programming by preconditioned conjugate gradient methods combined with a direct method. The central issue is to study the theoretical properties of a particular class of preconditioners.

1.1 Interior-Point Methods for Linear Programming

Consider the standard form primal linear program

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to:} & Ax = b \\ & x \geq 0 \end{array}$$

and its dual program

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to:} & A^T y + z = c \\ & z \geq 0 \end{array}$$

where $A \in \Re^{m \times n}$ $(m < n)$, $b \in \Re^m$ and $c \in \Re^n$. For convenience, we will assume that A has full rank m.

The optimality (Karush-Kuhn-Tucker or KKT) conditions for the above linear program pair can be written as a linear-quadratic system of equations plus nonnegativity constraints,

$$F(x, y, z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe \end{pmatrix} = 0, \quad (x, z) \geq 0,$$

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

We will mainly consider primal-dual interior-point algorithms that have proven to be the most efficient for linear programming, although the results in this paper can be applied to primal or dual interior-point algorithms as well. The basic framework of primal-dual interior-point methods can be viewed as applying Newton's method to the so-called perturbed KKT conditions

$$F_\mu(x, y, z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe - \mu e \end{pmatrix} = 0, \quad (x, z) > 0, \tag{1}$$

for a sequence of positive μ values that decreases to zero, while keeping x and z positive. The parameter μ is called the centering (or barrier) parameter, and the set of triples (x, y, z) that satisfy (1) for all $\mu > 0$ is called the (primal-dual) central path. References on primal-dual interior-point methods include [14, 6, 19], for example.

At a given iterate (x, y, z), where $x, z > 0$, and for a fixed value of μ, the linear equation that defines a Newton step for (1) is

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe - \mu e \end{pmatrix}. \tag{2}$$

The next iterate is obtained by the update

$$(x, y, z) \leftarrow (x, y, z) + \alpha(\Delta x, \Delta y, \Delta z)$$

where the step length α is chosen to ensure that the new x and z remain positive.

The most widely-used approach for solving (2) is to use a block Gaussian elimination to reduce it to a smaller system

$$(AGA^T)\Delta y = AG(c - A^T y - \mu X^{-1} e) + (b - Ax) \tag{3}$$

where $G = Z^{-1}X$ is positive definite diagonal matrix. After Δy is determined from solving the above linear system, Δx and Δz can then be easily computed. Hence, in this approach, the primary computation at each iteration is to solve (3).

It is worth noting that whenever there holds $b = A\tilde{x}$ for some $\tilde{x}$, system (3) can be written into

$$(AGA^T)\Delta y = AGh \tag{4}$$

where $h = (c - A^T y - \mu X^{-1} e) + G^{-1}(\tilde{x} - x)$. Evidently, (4) is the normal equations for the weighted least squares problem

$$\min_{\Delta y} \|A^T \Delta y - h\|_G^2, \tag{5}$$

with the weighted norm $\|\cdot\|_G \equiv \|G^{1/2}(\cdot)\|_2$. For this reason, we can say that the primary computation for this class of interior-point algorithms is to solve a sequence of weighted least squares problems (5) where the weights G and the right-hand side h vary from iteration to iteration.

In this paper, we consider solving (4) by a preconditioned conjugate gradient method. The preconditioner is based on a low-rank correction of a matrix AHA^T where (i) H is a positive definte diagonal matrix from a previous iteration and is not "vastly different" from G; (ii) the Cholesky factorization of AHA^T is already available. We will derive bounds on the condition number of the preconditioned coefficient matrix and provide guidelines for the choice of low-rank corrections. We report some preliminary computational results to validate our theoretical results and to demonstrate the effectiveness of the approach.

1.2 Organization and Notation

The outline of the paper is as follows: In Section 2, we introduce the construction of the preconditioners based on low-rank corrections. In Section 3, we give theoretical results to guide the choice of low-rank corrections. In Section 4, we briefly discuss the computation involved with the preconditioners. In section 5, we report preliminary numerical results.

Throughout this paper we use the following notation. The symbol $\min_i$ or $\max_i$ is for all i for which the argument is defined. For any matrix A, A_{ij} is the element in the i-th row and j-th column, A_j is the j-th column, $A_{j\bullet}$ is the j-th row, and $\mathrm{nnz}(A)$ is the number of nonzero elements in A. The vector norm $\|\cdot\|$ is the Euclidean norm $\|x\|^2 = x^T x$. The notation $x > 0$ ($x \geq 0$) means that all components of the vector x are positive (nonnegative). The symbol 0 will be used to denote the number zero, the zero vector, and the zero matrix. The symbol I is used to denote the (square) identity matrix; its size will always be apparent from the context. For any square matrix X with real eigenvalues, $\lambda_i(X)$ are the eigenvalues of X arranged in nondecreasing order. $\lambda_{\min}(X)$ and $\lambda_{\max}(X)$ denote the smallest and largest eigenvalues of X respectively; i.e.,

$$\lambda_{\min}(X) \equiv \lambda_1(X) \leq \lambda_2(X) \leq \cdots \leq \lambda_m(X) \equiv \lambda_{\max}(X).$$

If X is symmetric and positive definite, then the above arrangement gives

$$\lambda_i(X^{-1}) = 1/\lambda_{m-i+1}(X). \tag{6}$$

In addition, we will denote the spectral condition number of X by $\kappa(X)$ where by definition

$$\kappa(X) \equiv \lambda_{\max}(X)/\lambda_{\min}(X).$$

The letters L and R represent lower triangular factors or unit lower triangular factors of a symmetric, positive definite matrix. The type of factors will always be apparent from the context.

2 Low-Rank Corrections as Preconditioners

2.1 Motivation

In most implementations of primal-dual interior-point algorithms for linear programming, the direct method of Cholesky factorization is used to solve the system (4). The use of iterative methods of conjugate-gradient type have also been considered (see for example [5, 13, 15]). However, the success of iterative methods has been, at best, very limited for interior-point algorithms due to the difficulty in constructing general-purpose, effective preconditioners.

It is well known that effective preconditioners are critical for accelerating the convergence rate of conjugate-gradient type methods. In search for a preconditioner, one has to balance two conflicting goals: (i) to minimize the condition number of the preconditioned coefficient matrix, and (ii) to minimize the cost of solving linear systems with the preconditioner as the coefficient matrix. We refer the readers to the book [10] for a discussion on the preconditioned conjugate-gradient method.

In this paper, we investigate a mixed strategy for solving linear systems of the form (4), in which a direct method and an iterative method are combined in some manner. Specifically, we will combine Cholesky factorization (or LDL^T) with preconditioned conjugate-gradient method. A simple, but not necessarily effective, combination would be that the Cholesky factorization is used at every even iteration and the preconditioned conjugate-gradient method at every odd iteration.

Suppose that we have calculated the Cholesky factorization of AHA^T at a previous iteration and we are to solve a new linear system with coefficient matrix AGA^T where G is different but not "too far away" from H. Then it is advantageous to utilize the existing Cholesky factorization of AHA^T in the construction of a preconditioner for the conjugate-gradient method. Moreover, a good candidate for such a preconditioner is a low-rank correction (update) to AHA^T because of two reasons. Firstly, a good correction may be able to at least partially account for the changes from H to G. Secondly, given the availability of the Cholesky factors, a low-rank update will keep the cost low for solving linear systems with the preconditioner as the coefficient matrix.

2.2 The Preconditioner

We now formally introduce low-rank correction preconditioners. Let $G, H \in \Re^{n\times n}$ be positive definite diagonal matrices. As stated above, we already have the Cholesky factorization of AHA^T and we want to solve a linear system with coefficient matrix AGA^T. We employ a preconditioned conjugate gradient method with a preconditioner AKA^T, where K is an $n \times n$ positive definite diagonal matrix constructed from H and G so that the difference between AKA^T and AHA^T is a low rank matrix. In this case, the Cholesky factorization of AHA^T can be effectively used to solve linear systems with the preconditioner AKA^T as the coefficient matrix. We now give our construction of the matrix K.

For a given index set

$$\mathcal{Q} \subseteq \{j : 1 \leq j \leq n \text{ and } G_{jj} \neq H_{jj}\},$$

let the $n \times n$ diagonal matrices D and K be given, respectively, by

$$D_{jj} = \begin{cases} G_{jj} - H_{jj}, & \text{if } j \in \mathcal{Q}, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

and

$$K = H + D. \tag{8}$$

Note that $K_{jj} = G_{jj}$ if $j \in \mathcal{Q}$, otherwise $K_{jj} = H_{jj}$. Clearly, AKA^T is a symmetric positive definite matrix.

Let $\bar{A} \in \Re^{m \times q}$, where $q = |\mathcal{Q}|$, consist of all columns A_j such that $j \in \mathcal{Q}$ and let $\bar{D} \in \Re^{q \times q}$ be the diagonal matrix corresponding to the nonzero diagonal elements of D. In this notation,

$$AKA^T = A(H + D)A^T = AHA^T + \bar{A}\bar{D}\bar{A}^T, \tag{9}$$

namely, AKA^T is a rank q-update of AHA^T. In particular, if

$$\mathcal{Q} = \{j : 1 \leq j \leq n \text{ and } G_{jj} \neq H_{jj}\},$$

then $AKA^T = AGA^T$. However, as it will become clear later, we will always keep q fairly small in our algorithms. Further, the elements in the class of preconditioners are determined by the choice of the index set $\mathcal{Q}$.

2.3 Related Work

The idea of using low-rank updates is not new. In his seminal paper on the first polynomial-time interior-point method for linear programming, Karmarkar [12] already proposed to use low-rank updates to decrease the theoretical complexity bounds. His idea has been pursued by many later papers (for example, Goldfarb and Liu [9]) for the similar reason. Moreover, a combination of a direct method and an iterative method was reported by Karmarkar and Ramakrishnan [13].

A closely related recent work to this present paper is by Wang and O'Leary [17] where the authors propose a strategy of preconditioning the normal equation system based on low-rank corrections similar to what is discussed in this section. In their implementation, they choose the index set $\mathcal{Q}$ to consist of the indices corresponding to the largest values of $|G_{jj} - H_{jj}|$. When selecting indices for $\mathcal{Q}$, we will show that one should look at the magnitude of G_{ii}/H_{ii} instead of $|H_{ii} - G_{ii}|$.

3 Condition Number of the Preconditioned Matrix

The quality of a preconditioner is measured by the condition number of the preconditioned matrix $(AKA^T)^{-1}AGA^T$. In this section, we will derive lower and upper bounds for this condition number. We will argue that these bounds, especially the upper bound, will provide useful information on how to choose the index set $\mathcal{Q}$.

3.1 An Upper Bound

The following technical lemma is needed for the proof of Theorem 3.1 which gives bounds on the eigenvalues and the condition number of the preconditioned matrix $(AKA^T)^{-1}AGA^T$.

Lemma 3.1 *Let $G, H \in \Re^{n\times n}$ be symmetric. If H is positive definite, then*

$$\lambda_{\min}(H^{-1}G) \le \lambda_i((AHA^T)^{-1}AGA^T) \le \lambda_{\max}(H^{-1}G). \tag{10}$$

Moreover, (10) also holds if G is positive definite, and H and AHA^T are nonsingular.

Proof: First we note that the eigenvalues of $(AHA^T)^{-1}AGA^T$ are all real because it is similar to the symmetric matrix

$$(AHA^T)^{-1/2}(AGA^T)(AHA^T)^{-1/2}.$$

Let $\tilde{A} = AH^{1/2}$ and $\hat{G} = H^{-1/2}GH^{-1/2}$, then

$$(AHA^T)^{-1}AGA^T = (\tilde{A}\tilde{A}^T)^{-1}\tilde{A}\hat{G}\tilde{A}^T. \tag{11}$$

Let

$$\tilde{A} = U\Sigma V^T = U\Sigma_1 V_1^T \tag{12}$$

be the singular value decomposition of $\tilde{A}$, where $U \in \Re^{m\times m}$ and $V \in \Re^{n\times n}$ are orthogonal matrices and $\Sigma \in \Re^{m\times n}$ is a positive definite diagonal matrix. Let $\Sigma_1 \in \Re^{m\times m}$ and $V_1 \in \Re^{n\times m}$ consist of the first m columns of Σ and V respectively. Substituting (12) into the right hand side of (11), and noting that $V_1^T V_1 = I$ and $U^T U = I$ gives

$$(AHA^T)^{-1}AGA^T = (U\Sigma_1^{-1})V_1^T \hat{G} V_1(\Sigma_1 U^T).$$

Therefore by similarity, $\lambda_i((AHA^T)^{-1}AGA^T) = \lambda_i(V_1^T\hat{G}V_1)$. Thus, there exists a unit eigenvector $y_i \in \Re^m$ of $V_1^T\hat{G}V_1$ such that

$$\lambda_i((AHA^T)^{-1}AGA^T) = \lambda_i(V_1^T\hat{G}V_1) = y_i^T V_1^T \hat{G} V_1 y_i \geq \lambda_{\min}(\hat{G}),$$

where the last inequality follows from the fact that $V_1 y_i$ is a unit vector in $\Re^n$. Similarly, we can prove that $\lambda_i((AHA^T)^{-1}AGA^T) \leq \lambda_{\max}(\hat{G})$. Finally we note that $\hat{G}$ is similar to $H^{-1}G$.

Considering the matrix $(AGA^T)^{1/2}(AHA^T)^{-1}(AGA^T)^{1/2}$, the proof for the second assertion follows from a similar argument. ■

Theorem 3.1 *Let $G, H \in \Re^{n\times n}$ be positive definite diagonal matrices. For a given $\mathcal{Q}$ let K be defined as in (8). The eigenvalues of $(AKA^T)^{-1}AGA^T$ satisfy*

$$\min\left\{1, \min_{j\notin\mathcal{Q}} \frac{G_{jj}}{H_{jj}}\right\} \leq \lambda_i((AKA^T)^{-1}AGA^T) \leq \max\left\{1, \max_{j\notin\mathcal{Q}} \frac{G_{jj}}{H_{jj}}\right\}.$$

Moreover,

$$\kappa((AKA^T)^{-1}AGA^T) \leq \max\left\{1, \max_{j\notin\mathcal{Q}} \frac{G_{jj}}{H_{jj}}\right\} / \min\left\{1, \min_{j\notin\mathcal{Q}} \frac{G_{jj}}{H_{jj}}\right\}.$$

Proof: Since K defined by equation (8) is positive definite and diagonal, Lemma 3.1 implies that

$$\min_j \{G_{jj}/K_{jj}\} \leq \lambda_i((AKA^T)^{-1}AGA^T) \leq \max_j \{G_{jj}/K_{jj}\}. \tag{13}$$

For $j \in \mathcal{Q}$ then $G_{jj}/K_{jj} = 1$, and if $j \notin \mathcal{Q}$ then $G_{jj}/K_{jj} = G_{jj}/H_{jj}$. We thus have

$$\max_j \left\{\frac{G_{jj}}{K_{jj}}\right\} = \max\left\{1, \max_{j\notin\mathcal{Q}} \{G_{jj}/H_{jj}\}\right\}$$

and

$$\min_j \left\{ \frac{G_{jj}}{K_{jj}} \right\} = \min \left\{ 1, \min_{j \notin \mathcal{Q}} \{G_{jj}/H_{jj}\} \right\}.$$

Moreover, the upper bound on the condition number follows directly from the definition of the condition number and the above eigenvalue bounds. This completes the proof. ∎

Corollary 3.1 *Let $G, H \in \Re^{n\times n}$ be positive definite and diagonal. Let γ_j be the sorted elements of G_{jj}/H_{jj} in nondecreasing order:*

$$\min_j \{G_{jj}/H_{jj}\} \equiv \gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_n \equiv \max_j \{G_{jj}/H_{jj}\}.$$

Let $q = q_1 + q_2$ and $\mathcal{Q}$ consist of the indices corresponding to the q_1 largest and q_2 smallest diagonal elements of $H^{-1}G$. Then

$$\min\{\gamma_{q_2+1}, 1\} \leq \lambda_i((AKA^T)^{-1}AGA^T) \leq \max\{\gamma_{n-q_1}, 1\}.$$

Moreover, if $\gamma_{n-q_1} \geq 1$, $\gamma_{q_2+1} \leq 1$ and $q \leq m$, then

$$\kappa((AKA^T)^{-1}AGA^T) \leq \frac{\gamma_{n-q_1}}{\gamma_{q_2+1}}. \tag{14}$$

Proof: Observe that

$$\max_{j \notin \mathcal{Q}} \{G_{jj}/K_{jj}\} = \max_{j \notin \mathcal{Q}} \{G_{jj}/H_{jj}\} = \gamma_{n-q_1}$$

and

$$\min_{j \notin \mathcal{Q}} \{G_{jj}/K_{jj}\} = \min_{j \notin \mathcal{Q}} \{G_{jj}/H_{jj}\} = \gamma_{q_2+1}.$$

The result follows from Theorem 3.1 and the definition of the spectral condition number. ∎

3.2 A Lower Bound

Lemma 3.2 gives bounds on the eigenvalues of a matrix after a series of rank-one corrections. This result will be used in the proof of Theorem 3.2 to give bounds on the eigenvalues and the condition number of the preconditioned matrix $(AKA^T)^{-1}AGA^T$.

Lemma 3.2 *Suppose $B_q = B_0 + \sum_{j=1}^{q} \tau_j c_j c_j^T$, where $c_j \in \Re^m$, $\tau_j \neq 0$, $q < m$, and $B_0 \in \Re^{m\times m}$ is symmetric. Let $q = q_1 + q_2$, where q_1 is the number of indices such that $\tau_j > 0$. Then*

$$\lambda_{\min}(B_q) \leq \lambda_{q_1+1}(B_0) \quad \textit{and} \quad \lambda_{\max}(B_q) \geq \lambda_{m-q_2}(B_0).$$

Proof: Observe that

$$B_j = B_{j-1} + \tau_j c_j c_j^T,\ 1 \le j \le q.$$

The proof follows from Wilkinson ([18], pp. 94-97). ■

Theorem 3.2 gives a lower bound on $\lambda_{\max}((AKA^T)^{-1}AGA^T)$ and an upper bound on $\lambda_{\min}((AKA^T)^{-1}AGA^T)$.

Theorem 3.2 *Let $G, H \in \Re^{n\times n}$ be positive definite diagonal matrices. Let*

$$|\mathcal{Q}| = q = q_1 + q_2 \le m,$$

where q_1 is the number of indices in $\mathcal{Q}$ such that $G_{jj} > H_{jj}$. Then

$$\lambda_{\min}((AKA^T)^{-1}AGA^T) \le \lambda_{q_2+1}((AHA^T)^{-1}AGA^T)$$

and

$$\lambda_{\max}((AKA^T)^{-1}AGA^T) \ge \lambda_{m-q_1}((AHA^T)^{-1}AGA^T).$$

Moreover,

$$\kappa((AKA^T)^{-1}AGA^T) \ge \frac{\lambda_{m-q_1}((AHA^T)^{-1}AGA^T)}{\lambda_{q_2+1}((AHA^T)^{-1}AGA^T)}.$$

Proof: Consider again

$$AKA^T = AHA^T + \bar{A}(\bar{G} - \bar{H})\bar{A}^T = AHA^T + \bar{A}\bar{D}\bar{A}^T.$$

Let $\bar{D}_1$ and $-\bar{D}_2$ be diagonal matrices with the positive and negative diagonal elements of $\bar{D}$ on their diagonals, respectively. Clearly, since $\mathcal{Q}$ contains no index such that $G_{jj} = H_{jj}$ $\bar{D}_1$ and $\bar{D}_2$ are positive definite matrices of sizes $q_1 \times q_1$ and $q_2 \times q_2$, respectively. Corresponding to the indices of $\bar{D}_1$ and $\bar{D}_2$, we similarly define $\bar{A}_1$ and $\bar{A}_2$. Then

$$AKA^T = AHA^T + \bar{A}_1\bar{D}_1\bar{A}_1^T - \bar{A}_2\bar{D}_2\bar{A}_2^T. \tag{15}$$

Let

$$B_0 = (AGA^T)^{-1/2}AHA^T(AGA^T)^{-1/2}$$

and

$$B_q = (AGA^T)^{-1/2}AKA^T(AGA^T)^{-1/2}.$$

We define

$$c_j = \begin{cases} \left[(AGA^T)^{-1/2}\bar{A}_1\bar{D}_1^{1/2}\right]_j, & j = 1, 2, \cdots, q_1, \\ \left[(AGA^T)^{-1/2}\bar{A}_2\bar{D}_2^{1/2}\right]_{j-q_1}, & j = q_1+1, \cdots, q_1+q_2, \end{cases}$$

and

$$\tau_j = \begin{cases} 1, & j = 1, 2, \cdots, q_1, \\ -1, & j = q_1 + 1, \cdots, q_1 + q_2. \end{cases}$$

It follows from (15) that

$$B_q = B_0 + \sum_{j=1}^{q} \tau_j c_j c_j^T.$$

From Lemma 3.2 and using (6)

$$\lambda_{\max}(B_q^{-1}) \;=\; \frac{1}{\lambda_{\min}(B_q)} \geq \frac{1}{\lambda_{q_1+1}(B_0)} = \lambda_{m-q_1}(B_0^{-1}).$$

Similarly

$$\lambda_{\min}(B_q^{-1}) \;=\; \frac{1}{\lambda_{\max}(B_q)} \leq \frac{1}{\lambda_{m-q_2}(B_0)} = \lambda_{q_2+1}(B_0^{-1}).$$

The theorem follows from the similarity between the matrices B_q^{-1} and $(AKA^T)^{-1}AGA^T$, and between B_0^{-1} and $(AHA^T)^{-1}AGA^T$. ∎

3.3 Interpretation of the Results

Theorem 3.1 implies that

$$\kappa((AKA^T)^{-1}AGA^T) \leq \frac{\max_{j \notin \mathcal{Q}}\{G_{jj}/H_{jj}\}}{\min_{j \notin \mathcal{Q}}\{G_{jj}/H_{jj}\}}, \tag{16}$$

whenever the numerator is greater than or equal to one and the denominator is less than or equal to one. Recall that $\{\gamma_j\}$ are the sorted elements of $\{G_{jj}/H_{jj}\}$. Combining Theorem 3.2 and Corollary 3.1 we have that if $\mathcal{Q}$ consist of the indices corresponding to the q_1 largest and q_2 smallest diagonal elements of $H^{-1}G$, $\gamma_{n-q_1} \geq 1$, $\gamma_{q_2+1} \leq 1$ and $q = q_1 + q_2 \leq m$, then the condition number of the preconditioned matrix $(AKA^T)^{-1}AGA^T$ satisfies

$$\frac{\lambda_{m-q_1}((AHA^T)^{-1}AGA^T)}{\lambda_{q_2+1}((AHA^T)^{-1}AGA^T)} \leq \kappa((AKA^T)^{-1}AGA^T) \leq \frac{\gamma_{n-q_1}}{\gamma_{q_2+1}}. \tag{17}$$

From the above two inequalities, the following observations are now in order.

1. When selecting indices for $\mathcal{Q}$, one should look at the magnitude of G_{ii}/H_{ii} instead of $|H_{ii} - G_{ii}|$ as is suggested in [17]. This is not a surprise because $|H_{ii} - G_{ii}|$ measures the absolute change while the departure of G_{ii}/H_{ii} from the unity measures the relative change.

2. In order to reduce the upper bound, the index set $\mathcal{Q}$ should include, if possible, either indices corresponding to the largest elements of $\{G_{ii}/H_{ii}\}$ satisfying $G_{ii}/H_{ii} > 1$, or indices corresponding to the smallest elements of $\{G_{ii}/H_{ii}\}$ satisfying $G_{jj}/H_{jj} < 1$, or both.

3. One can choose $\mathcal{Q}$ to minimize the upper bound for a given q by varying q_1 and q_2 such that $q = q_1 + q_2$. However, we note that a smaller upper bound does not necessarily imply a smaller condition number.

4. The lower bound sets a limit on how well one can do in terms of making the condition number small for given q_1 and q_2. Unfortunately, the lower bound is not as easily computable as the upper bound is.

4 Computing with the Preconditioner

We shall consider two approaches of computing $(AKA^T)^{-1}d$: using the Sherman Morrison-Woodbury formula or updating the Cholesky factor (or the LDL^T factors).

4.1 Using the Sherman-Morrison-Woodbury Formula

To simplify the notation in the rest of this section, let $M = (AKA^T)^{-1}$ be the inverse of the preconditioner. For $AKA^T = AHA^T + \bar{A}\bar{D}\bar{A}^T$ (see (9)) the Sherman Morrison-Woodbury formula (see [10] pp. 50, for example) reads

$$M = (AHA^T)^{-1}[I - \bar{A}(\bar{D}^{-1} + \bar{A}^T(AHA^T)^{-1}\bar{A})^{-1}\bar{A}^T(AHA^T)^{-1}].$$

Assume that we have a Cholesky factorization $AHA^T = LL^T$, and let $V = L^{-1}\bar{A}$. Then

$$\begin{aligned} M &= (LL^T)^{-1} - L^{-T}V(\bar{D}^{-1} + V^TV)^{-1}V^TL^{-1} \\ &= L^{-T}(I - VF^{-1}V^T)L^{-1}, \end{aligned} \tag{18}$$

where $F = \bar{D}^{-1} + V^TV$ is $q \times q$ and q is usually small. For an m-vector d we have

$$Md = L^{-T}[r - (V(F^{-1}(V^Tr)))],$$

where $r = L^{-1}d$. Hence, in computing Md we need to store L, V, and the factors of F. Note that F is nonsingular and symmetric but may be indefinite since $\bar{D}$ may have negative elements on its diagonal.

The cost of computing and using M is dominated by the number of solves carried out. By a solve we mean either a backward or forward substitution involving L^{-1} or L^{-T}. The initial cost to compute M consists of q solves to compute $V = L^{-1}\bar{A}$, and the cost to form and factor the q-by-q matrix F. The computation cost of Md consists of 2 solves plus approximately $4\text{nnz}(V) + 2q^2 + m$ floating point operations where $\text{nnz}(V)$ is the number of nonzero elements in V.

Numerical testing on large Netlib test problems shows that V is usually a sparse matrix. The direct approach for computing $V = L^{-1}\bar{A}$ is to carry out q solves with q columns of $\bar{A}$. These will be solves with very sparse right-hand sides and will require much fewer floating point operations than solved with dense right-hand sides that we have in computing Md.

An alternative approach can exploit even more of the sparsity structure of $\bar{A}$. Assume that the number of nonzero rows in $\bar{A}$ is p which is small relative to q, i.e., $p \ll q$. Let e_i be the i-th unit vector in $\Re^m$ corresponding to the index i of a nonzero row of $\bar{A}$. Let E be the $m \times p$ matrix consisting of the columns e_i for all i corresponding to the nonzero rows of $\bar{A}$. Then $E^T\bar{A}$ is a $p \times q$ sub-matrix of $\bar{A}$ obtained from deleting all zero rows of $\bar{A}$, and $L^{-1}\bar{A} = (L^{-1}E)(E^T\bar{A})$. Hence we only need to carry out p solves with E and then multiply the result by $E^T\bar{A}$. Again note that the sparsity in the right hand sides should be exploited.

4.2 Updating the Triangular Factors

The sparsity structure of AKA^T is the same as that of AHA^T. When we update the triangular factors of AHA^T the sparsity structure of these factors is preserved. There are cases where it is more efficient to update the triangular factors than using the technique in the previous subsection. That is, given $AHA^T = LDL^T$ we want to compute R and T such that

$$RTR^T = LDL^T + \bar{A}\bar{D}\bar{A}^T,$$

where R and L are lower triangular matrices and T and D are positive definite diagonal matrices.

Bennett [3] presents an algorithm for updating triangular factors of a general matrix of the form $B = W + XCY^T$, when the factors of W are given. For W and C symmetric and $X = Y$, the algorithm can be simplified and the number of floating point operations is halved. This algorithm is stable [8] when applied to AKA^T and AHA^T since both are symmetric and positive definite. Bennett's algorithm is generalized to the sparse case by Baryamureeba and Steihaug [2]. The sparse algorithm is developed for the special class of updates where $\bar{A}$ is a sub-matrix of A. This algorithm is given below:

Algorithm 4.1 *Triangular factor update for sparse matrices*

Set $V = \bar{A}$, $C = \bar{D}$ *and* $R = 0$.
for $i = 1, \ldots, m$ **do**
  if $V_{i\bullet} \neq 0$ **then**
    $p = (V_{i\bullet} C)^T$
    $T_{ii} = D_{ii} + V_{i\bullet}\, p$
    $u = (1/T_{ii})\, p$
    $C \leftarrow C - u\, p^T$
    for $j = i+1, \ldots, m$ **do**
      if $L_{ji} \neq 0$ **then**
        $V_{j\bullet} \leftarrow V_{j\bullet} - L_{ji} V_{i\bullet}$
        $R_{ji} = L_{ji} + V_{j\bullet}\, u$
      end if
    end for
  end if
end for

The final V in Algorithm 4.1 can be shown to satisfy $V = L^{-1}\bar{A}$. Algorithm 4.1 is very effective when there are either many zero rows in $\bar{A}$ or few nonzero elements in the unit lower triangular factor L. It is important to note that in Algorithm 4.1, the sparsity structure of the triangular factors is preserved for the special case where $\bar{A}$ consists of columns from A and $\bar{D}$ is a diagonal matrix. Since the matrices K and H are diagonal, AKA^T and AHA^T have the same sparsity structure. Thus if R and L are Cholesky factors of AKA^T and AHA^T respectively, then they have the same sparsity structure. This is achieved in Algorithm 4.1 by setting $R_{ji} = 0$ whenever $L_{ji} = 0$.

5 Numerical Results

In this section, we report some numerical results. The purposes of the numerical experiments are to validate the relevance of our theoretical results and to determine the viability of the mixed strategy of combining Cholesky factorization and conjugate gradient with low-rank correction preconditioners. Specifically, our numerical experiments are designed to address the following issues.

1. We want to compare the spectral distribution of $(AKA^T)^{-1}AGA^T$ with that of $(AHA^T)^{-1}AGA^T$ to determine the impact of the diagonal update from H to K.

2. In our theoretical analysis, the extreme eigenvalues of $K^{-1}G$ are used to derive bounds on those of $(AKA^T)^{-1}AGA^T$. We want to see how sharp or loose the bounds are in practice to get a sense of the sharpness of our bounds.

3. We want to compare two techniques for selecting the index set $\mathcal{Q}$: the one based on $G - H$ proposed in [17] and the one based on $H^{-1}G$ proposed in this paper.

4. We want to compare numerical performance of two approaches: (i) using a mixture of a direct method and an iterative method, and (ii) using a direct method only.

We emphasize that the numerical results in this paper are preliminary and should be considered to be diagnostic and qualitative.

5.1 Implementation Details

We have implemented a primal-dual Newton algorithm (PDN), where the Cholesky factorization is used at all iterations, and variants of the mixed-strategy primal-dual Newton algorithm, where Cholesky factorization and preconditioned conjugate gradient method are used alternatively at the even and odd iterations. This is a simple strategy and more sophisticated strategies are discussed in [17]. The implementation was done in Matlab.

We choose a starting point (x_0, y_0, z_0) that is identical to the one given by Mehrotra [15]. We terminate the interior-point iterations when the relative error ϵ, as defined in (19) below, is less than or equal to the prescribed tolerance $\epsilon^* = 10^{-5}$, or when we exceed the maximum number of interior-point iterations which is set to 300 in our experiments. The relative error is defined as

$$\epsilon = \max\left\{ \frac{\|Ax - b\|}{\max\{1, \|b\|\}}, \frac{\|A^T y + z - c\|}{\max\{1, \|c\|\}}, \frac{|c^T x - b^T y|}{\max\{1, |c^T x|\}} \right\}. \tag{19}$$

At each iteration, to compute the step $(\Delta x, \Delta y, \Delta z)$ through (2) we set the barrier parameter $\mu = \sigma(x^T z)/n$ for $\sigma = 0.1$. The step length α is chosen as a parameter $\tau \in (0, 1)$ times the step length to the boundary of the nonnegative orthant for $\tau = 0.99995$.

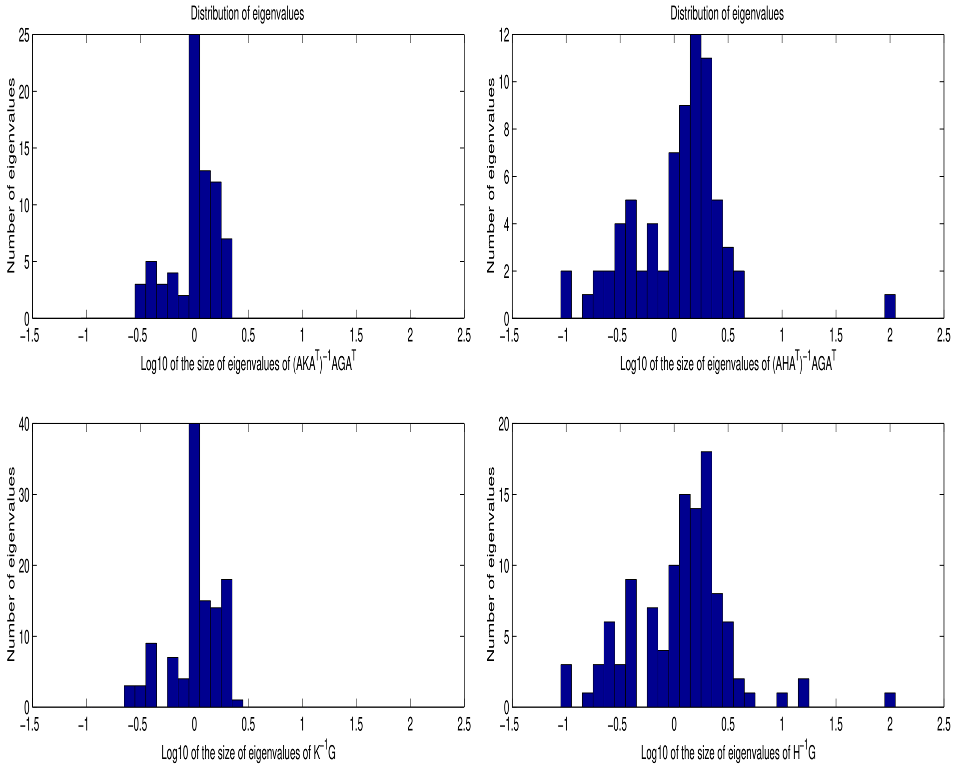

Figure 1: The matrices are extracted at an interior-point iteration in the middle stage for test problem `blend`. We set $q = 30$ and choose $\mathcal{Q}$ to consist of indices corresponding to the 20 largest and 10 smallest diagonal elements of $H^{-1}G$.

We used the Matlab function `symmmd` to compute the symmetric multiple minimum degree ordering for matrix AHA^T and the sparse Cholesky solver `chol`. The iterative method is the preconditioned conjugate gradient method for least squares (PCGLS) [4]. The termination criteria in the PCGLS routine are when either the number of PCGLS iterations has reached a prescribed integer t or when $\|v\| \leq 10^{-5}$ where $v = AGA^T\Delta y - AGh$ is the residual.

The test problems used in our experiments are all from the Netlib set of linear programs [7]. All the numerical experiments are performed on a Sun Ultra 10 Workstation.

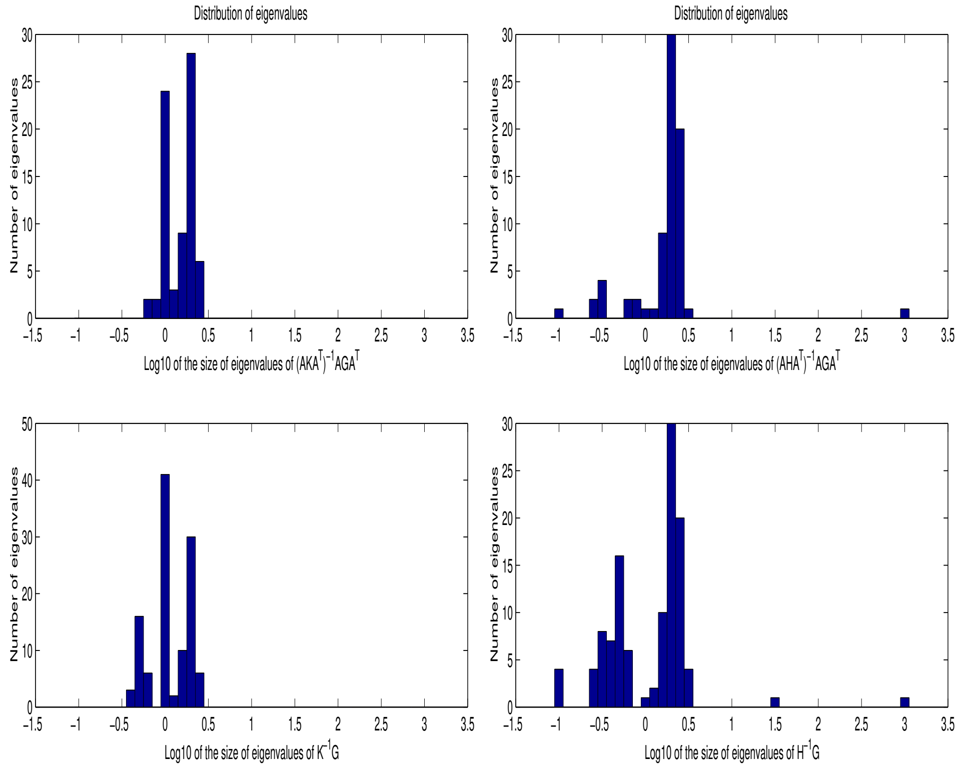

Figure 2: The matrices are extracted at an interior-point iteration in the last stage for test problem `blend`. We set $q = 40$ and choose $\mathcal{Q}$ to consist of indices corresponding to the 20 largest and 20 smallest diagonal elements of $H^{-1}G$.

5.2 How Good Are the Corrections and Bounds?

In Figures 1 and 2, we give some examples that demonstrate the effectiveness of the low-rank correction preconditioners for the weighted least squares problems and the sharpness of the bounds based on $H^{-1}G$. The results are generated from one of the Netlib test problem `blend` at a particular interior-point iteration.

The results in Figures 1 and 2 verify Lemma 3.1 and Corollary 3.1. We observe that the matrix $(AKA^T)^{-1}AGA^T$ has more eigenvalues near 1 than $(AHA^T)^{-1}AGA^T$. The spectrum of $(AKA^T)^{-1}AGA^T$ is contained in that of $K^{-1}G$ and is considerably narrower than that of $(AHA^T)^{-1}AGA^T$ which is contained in that of $H^{-1}G$. Clearly, $(AKA^T)^{-1}AGA^T$ has a smaller condition number than $(AHA^T)^{-1}AGA^T$. The bound on this condition number depends on the choice of the index set $\mathcal{Q}$ as seen in (17).

5.3 How Should the Index Set Be Selected?

We compare two techniques for selecting the index set for the low-rank update. The first technique is based on choosing the indices corresponding to the largest diagonal elements of $|G-H|$, as discussed in [17]. The second technique is based on choosing indices corresponding to the smallest and the largest diagonal elements of $H^{-1}G$ as proposed in this paper. We present an example based on the Netlib problem **sc205** and mention that the observed behavior seems to be typical out of many similar experiments we conducted on different test problems.

From the graph on the left of Figure 5.3, we see that using the choice in [17] to construct the preconditioner, the preconditioned conjugate gradient method requires considerably more iterations to achieve the prescribed accuracy than using our choice. This is well explained by the two bar graphs on the right where we see that the spectrum of the preconditioned matrix corresponding to M_2 (based on $H^{-1}G$) is more clustered than that corresponding to M_1 (based on $|G-H|$).

5.4 How Does the Mixed Strategy Perform?

In Table 1, we compare the performance of the standard primal-dual Newton (PDN) algorithm with three variants of the mixed primal-dual Newton (Mixed PDN) algorithm: Mixed PDN^i, $i = 1, 2, 3$. When $\epsilon \geq 0.1$ we set $t = 7$, $q = 6$ for Mixed PDN^1; $t = 5$, $q = 20$ for Mixed PDN^2; and $t = 5$, $q = 40$ for Mixed PDN^3; where t is the maximum number of PCGLS iterations allowed. When $\epsilon < 0.1$ we set $t = 40$ for all three with their q values unchanged.

Problem	PDN		Mixed PDN^1		Mixed PDN^2		Mixed PDN^3	
name	iter	time	iter	time	iter	time	iter	time
czprob	56	89.01	59	59.43	57	56.47	55	55.30
d2q06c	54	544.85	61	432.95	61	387.23	58	395.09
d6cube	42	317.58	47	230.59	45	221.98	46	230.57
stocfor2	41	113.26	43	92.20	42	87.56	46	99.35
scsd8	16	21.41	20	16.62	18	15.34	19	17.86

Table 1: Comparison of four algorithms. The preconditioner M used is given by (18).

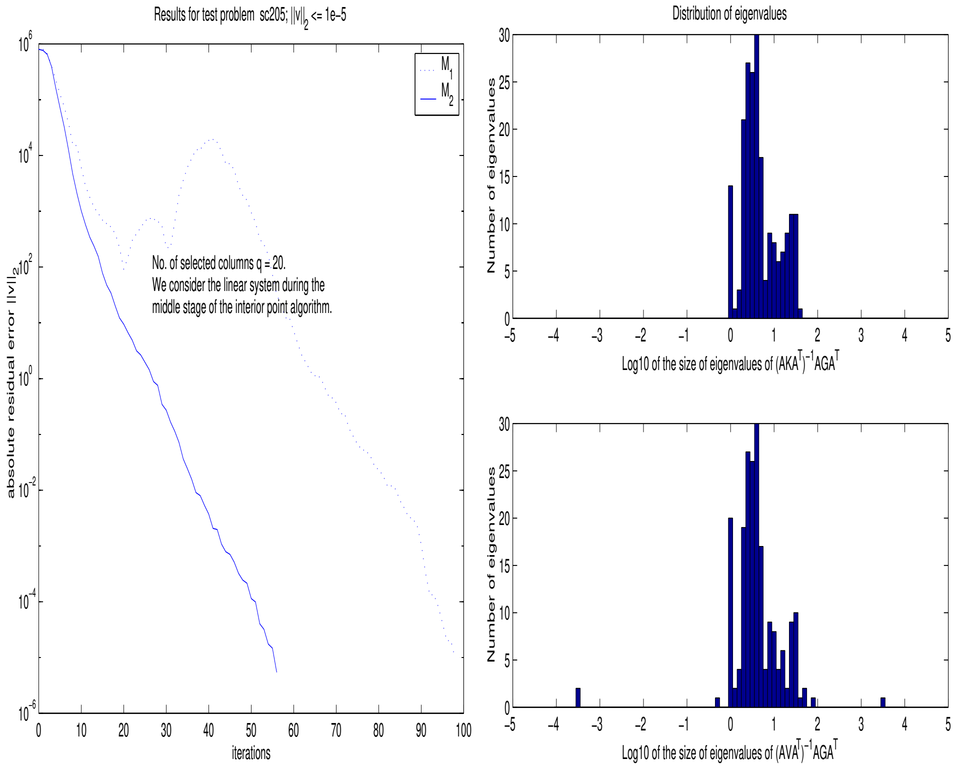

Figure 3: For preconditioner $M_1 = AVA^T$, we choose indices corresponding to 20 largest diagonal elements of $|G - H|$. For preconditioner $M_2 = AKA^T$, we choose indices corresponding to 10 largest and 10 smallest diagonal elements of $H^{-1}G$ at that same iteration. The test problem is obtained at an interior point iteration in the middle stage for test problem **sc205** from the Netlib.

Recall that the termination criteria in PCGLS routine are when either the number of PCGLS iterations has reached a prescribed integer t or when $\|v\| \leq 10^{-5}$ where $v = AGA^T \Delta y - AGh$.

When $\epsilon \geq 0.1$, in most cases PCGLS iterations are terminated by the number of iterations reaching t. On the other hand, when $\epsilon < 0.1$ PCGLS iterations are terminated by the error tolerance. We have set the maximum number of PCGLS iterations per step to 40. For some of the large problems the number of PCGLS iterations per step may reach 40 for small values of q. Numerical experiments show that we do not necessarily need an accurate PCGLS step when $\epsilon \geq 0.1$. The results in Table 1 show that by carefully choosing q and t we can keep the number of interior-point iterations (iter) for mixed PDN algorithms close to that of PDN algorithm, and yet still obtain better CPU time with mixed PDN methods (time). When we increase q we can decrease t and vice versa to reduce the cost.

At present, the most practically efficient interior-point algorithmic framework for linear programming is the predictor-corrector algorithm of Mehrotra [16], along with Gondzio's multiple corrections [11]. See also [1]. Our comparison is done with the less efficient, but easier to implement, primal-dual Newton algorithm. Nevertheless, we believe that the above test results show that the mixed PDN approach is promising.

There are many possibilities for improving the efficiency of the mixed PDN approach. Obviously, the even-odd alternation is not generally optimal. Dynamic schemes of alternation based on easily computable bounds on condition numbers should be investigated. In addition, a mixture of the PCGLS steps with the predictor-corrector steps may also be beneficial.

6 Final Remarks

Our preliminary numerical results indicate that the theoretical bounds on the spectrum of the preconditoned matrix obtained in this paper are quite satisfactory. We believe that they also strongly support the principle of selecting the index set $\mathcal{Q}$ based on the magnitude of $H^{-1}G$. Moreover, the numerical results demonstrate that the strategy of mixing direct and iterative methods is promising and merits further study.

References

[1] E.D. Andersen, J. Gondzio, C. Meszaros, X. Xu, *Implementation of interior-point methods for large scale linear programs,* T.Terlaky(ed), Interior Point Methods of Mathematical Programming pp. 189-252. Kluwer Academic Publishers, 1996.

[2] V. Baryamureeba, and T. Steihaug, *Computational issues for a new class of preconditioners,* To appear in the Proceedings for the 2nd Workshop on Large-Scale Scientific Computations, 1999.

[3] J.M. Bennett, *Triangular factors of modified matrices*, Numerische Mathematik Vol. 7, pp. 217-221, 1965.

[4] Å. Björck, *Numerical methods for least squares problems*, SIAM, 1996.

[5] T.J. Carpenter and D.F. Shanno, *An interior point method for quadratic programs based on conjugate projected gradients*, Computational Optimization and Applications, Vol. 2, pp. 5-28, 1993.

[6] A.S. El-Bakry, R.A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation of the primal-dual Newton interior-point method for nonlinear programming,* J. Optimization Theory and Applications, Vol. 89:507-541, 1996.

[7] D.M. Gay, *Electronic mail distribution of linear programming test problems,* Mathematical Programming Society COAL Newsletter, No. 13, pp.10-12, 1985.

[8] P.E. Gill, G.H. Golub, W. Murray, and M.A. Saunders, *Methods for modifying matrix factorizations,* Mathematics of Computation, Vol. 28, pp. 505-535, 1974.

[9] D. Goldfarb, and S. Liu, *An $O(n^3L)$ primal interior-point algorithm for convex quadratic programming,* Mathematical Programming, Vol. 49, pp. 325–340, 1991.

[10] G.H. Golub, and C.H. Van Loan, *Matrix computations,* Third Edition, 1996.

[11] J. Gondzio, *Multiple centrality corrections in a primal-dual method for linear programming,* Computational Optimization and Applications. To appear.

[12] N. Karmarkar, *A new polynomial time algorithm for linear programming.* Combinatorica, 4 (1984) 373-395.

[13] N.K. Karmarkar and K.G. Ramakrishnan, *Computational results of an interior point algorithm for large-scale linear programming,* Mathematical Programming, Vol. 52, pp. 555-586, 1991.

[14] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point method for linear programming.* In Nimrod Megiddo, editor, *Progress in Mathematical programming, interior-point and related methods*, pages 29–47. Springer-Verlag, New York, 1989.

[15] S. Mehrotra, *Implementations of affine scaling methods: Approximate solutions of systems of linear equations using preconditioned conjugate gradient methods,* ORSA J. Comput., 4(1992), pp. 103-118.

[16] S. Mehrotra, *On the implementation of a primal-dual interior point method,* SIAM J. Optimization, Vol. 2, pp. 575-601, 1992.

[17] W. Wang, and D. P. O'Leary, *Adaptive use of iterative methods in predictor-corrector interior point methods for linear programming,* Technical Report No. CS-TR-4011, Computer Science Department, University of Maryland, April 1999. (Revised version of TR-3560, November 1995)

[18] J.H. Wilkinson, *The Eigenvalue Problem,* Oxford University Press, London, 1965.

[19] S. Wright, *Primal-dual interior-point methods.* SIAM, Philadelphia, 1997.