A PRECONDITIONER FOR LINEAR SYSTEMS ARISING FROM INTERIOR POINT OPTIMIZATION METHODS

TIM REES* AND CHEN GREIF†

Abstract. We explore a preconditioning technique applied to the problem of solving linear systems arising from primal-dual interior point algorithms in linear and quadratic programming. The preconditioner has the attractive property of improved eigenvalue clustering with increased ill-conditioning of the (1,1) block of the saddle point matrix. It fits well into the optimization framework since the interior point iterates yield increasingly ill-conditioned linear systems as the solution is approached. We analyze the spectral characteristics of the preconditioner, utilizing projections onto the null space of the constraint matrix, and demonstrate performance on problems from the NETLIB and CUTEr test suites. The numerical experiments include results based on inexact inner iterations.

Key words. block preconditioners, saddle point systems, primal-dual interior point methods, augmentation

AMS subject classifications. 65F10, 65K05

1. Introduction. Interior point methods for solving linear and quadratic programming problems have been gaining popularity in the last two decades. These methods have forged connections between previously disjoint fields and allowed for a fairly general algebraic framework to be used; see, for example, [11] for a comprehensive survey. The size of many problems of interest is very large and the matrices involved are frequently sparse and often have a special structure. As a result, there is an increasing interest in iterative solution methods for the saddle point linear systems that arise throughout the iterations.

The general optimization framework is as follows. Consider the quadratic programming (QP) problem

$$\begin{aligned} &\min_x \frac{1}{2} x^T H x + c^T x \\ &\text{subject to :} \\ &\quad Ax = b,\ Cx \geq d. \end{aligned}$$

Here $x, c \in \mathbb{R}^n$ and H is an $n \times n$ Hessian, often symmetric positive semidefinite; the constraint matrix A is $m_1 \times n$ with $m_1 < n$, and we assume it has rank m_1. Inequality constraints are expressed in the $m_2 \times n$ matrix C. Often simple bounds on the variables are given, so C could be an identity matrix or its concatenation with a negative identity matrix.

When H is symmetric positive semidefinite and the constraints are linear, satisfaction of the first-order KKT conditions is sufficient to guarantee global optimality of a solution [23, Chap. 16]. If Lagrange multipliers y, z and slack variables s are

*Department of Applied Mathematics, University of Waterloo, Waterloo ON, N2L 3G1, Canada, twrees@math.uwaterloo.ca. The work of this author was done while he was at the University of British Columbia, in partial fulfillment of the requirements for an M.Sc. degree.

†Department of Computer Science, University of British Columbia, Vancouver BC, V6T 1Z4, Canada, greif@cs.ubc.ca. The work of this author was supported in part by the Natural Sciences and Engineering Research Council of Canada.

introduced, the KKT conditions for this problem are

$$\begin{gathered} Hx - A^T y - C^T z = -c, \\ Ax = b, \\ Cx - s = d, \\ s \geq 0,\ z \geq 0,\ s^T z = 0. \end{gathered}$$

Typical interior point methods [23, 29] for QPs define a function whose roots coincide with the KKT conditions and take Newton steps to progressively approach an optimal solution. One possible way of solving the problem is by performing predictor and corrector solves at each step to ensure sufficient progress towards the optimal point. After elimination of some of the unknowns we obtain the step equations, a saddle point system of the form

$$\begin{pmatrix} H + C^T S^{-1} Z C & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \tag{1.1}$$

where S and Z are diagonal and keep changing throughout the QP iteration. The right hand side vector is related to residual measures of the KKT conditions and depends on whether a predictor or corrector step is being carried out [23, 29].

While we will focus primarily on QPs, it is worth also considering the linear programming (LP) problem, which is formulated as

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} c^T x \\ &\text{subject to :} \\ &Ax = b,\ x \geq d. \end{aligned}$$

LPs share similarities with QPs; in fact they can be classified as simple subproblems, with a zero Hessian and further simplifications. It is convenient to present the corresponding linear system in the 2×2 block form

$$\begin{pmatrix} S^{-1} Z & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \tag{1.2}$$

where $S^{-1}Z$ is diagonal and changes throughout the LP iteration, and the right hand side vector is constructed in a similar manner to (1.1).

For both LPs and QPs the saddle point matrix becomes highly ill-conditioned as the solution is approached, due to increased ill-conditioning of the $(1,1)$ block. In the typical case when inequality constraints are simple bounds on the primal variables, $H + C^T S^{-1} Z C$ is a diagonal perturbation of H. The complementarity of S and Z gives rise to the simultaneous presence of extremely small and extremely large values in $S^{-1}Z$. For both LPs and QPs the matrices $S^{-1}Z$ and $H + C^T S^{-1} Z C$ will never be exactly singular, but will approximate singularity as the solution is approached. This is a key property that must be taken into account.

Detailed description of the theory involved in interior point methods for linear and quadratic programming can be found, for example, in [23, 29]. The methodology of using a predictor and corrector step calculation at each iteration of the solve was presented in [21]. These references point out the importance of efficiently solving the step equations, and identify the difficulties involved. Many software packages such as IPOPT [28], LIPSOL [30], and OOQP [13] use direct solvers to solve the step equations. While this is a sound approach for many problems, it may suffer the

combined ailments of poor scaling with problem size and deterioration in conditioning and numerical stability as the QP or LP solution is approached. In particular, special care must be taken in matrix factorizations to deal with the presence of large and small pivots [22]. These factors motivate the study of iterative methods in the context of optimization. Modern solution techniques like Krylov subspace methods rely on the ease of sparse matrix-vector products, and converge in a rate dependent on the number of distinct eigenvalues of the preconditioned matrix [7, 26].

In this paper we study a preconditioner that has the property that the more ill-conditioned the (1,1) block of the saddle point matrix is, the faster a minimum residual solver such as MINRES converges (in terms of outer iterations). Therefore, the corresponding solver is particularly effective in the last few iterations of the LP or QP solver. Our approach is based on augmentation of the (1,1) block using a weight matrix. Augmentation has been used in several areas of applications and in many flavors (see for example [3] and references therein). Our particular methodology extends recent work done by Greif and Schötzau [17, 18] into new directions and introduces a technique that well fits the algebraic framework of interior point methods for optimization problems.

Our preconditioner is part of a growing set of preconditioned iterative approaches for solving optimization problems. A preconditioning technique that has emerged recently as a popular choice is the class of constraint preconditioners (see Lukšan and Vlček [20] or Keller, Gould and Wathen [19] and references therein), which rely on leaving the constraints intact, and seeking to replace the $(1, 1)$ block by a matrix that is much easier to invert. Recent work by Dollar and Wathen [9] introduces implicit factorizations that further facilitate the use of constraint preconditioners. Similar factorizations are applied to regularized saddle point systems by Dollar, Gould, Schilders and Wathen in [8]. Forsgren, Gill and Griffin [10] extend constraint-based preconditioners to deal with regularized saddle point systems using an approximation of the $(1, 1)$ block coupled with an augmenting term (related to a product with the constraint matrix and regularized $(2, 2)$ block). The technique is intended for interior point methods for general constrained optimization problems. In [5] Bergamaschi, Gondzio, and Zilli employ constraint preconditioners with diagonal $(1, 1)$ blocks, which allow for factoring the preconditioner or its reduced normal equation form. Approximate constraint preconditioners are further explored in [4].

Other block structured preconditioning approaches are also available. For example, Oliveira and Sorensen [24] consider linear programming and make use of block triangular preconditioners that have the constraint matrix in their $(1, 2)$ block and easy to invert matrices in the main diagonal. The preconditioned matrix has an eigenvalue $\lambda = 1$ with a high algebraic multiplicity, and since for linear programs the $(1, 1)$ block of the saddle point system is diagonal, the preconditioner can be factored and solved with efficiently. The use of iterative methods in constrained optimization also relies on the notion of inexact interior point methods. These have been investigated in [1, 2, 12] and other papers. The findings justify the use of approximate solutions at each step of the method, and we explore this experimentally.

The remainder of this paper is organized as follows. In Section 2 the preconditioner and its general form are presented, and algebraic properties are derived. In Section 3 the choice of the weight matrix W and inner solves are discussed. This is followed by two schemes for reducing fill-in in the preconditioner when dense rows are present. In Section 4 we present numerical results demonstrating performance of the preconditioner. In Section 5 we draw some conclusions.

2. The Preconditioning Approach. We will adopt the general notation

$$\mathcal{A} = \begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \tag{2.1}$$

to represent the saddle point matrices of equations (1.1) and (1.2). We assume that G is symmetric and positive semidefinite with nullity p, and that A is of size $n \times m$ and has full row rank. Note that the assumption that $\mathcal{A}$ is nonsingular implies that $\text{null}(A) \cap \text{null}(G) = \{0\}$, which we use in our analysis below.

2.1. A Block Triangular Preconditioner. Consider the preconditioner

$$\mathcal{M} = \begin{pmatrix} G + A^T W^{-1} A & kA^T \\ 0 & W \end{pmatrix},$$

where k is a scalar, and W is an $m \times m$ symmetric positive definite weight matrix. The eigenvalues of the preconditioned matrix $\mathcal{M}^{-1}\mathcal{A}$ satisfy the generalized eigenvalue problem

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} G + A^T W^{-1} A & kA^T \\ 0 & W \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{2.2}$$

The second block row gives $y = \frac{1}{\lambda} W^{-1} A x$, and substituting it into the first block equation gives

$$\lambda(\lambda - 1)Gx + (\lambda^2 + k\lambda - 1)A^T W^{-1} A x = 0.$$

Regardless of the choice of k, we see that $\lambda = 1$ with algebraic multiplicity $n - m$. From the nullity of G it follows that there are p linearly independent null vectors of G. For each such null vector we can find two λ values satisfying $\lambda^2 + k\lambda - 1 = 0$. Thus we have

$$\lambda_{\pm} = \frac{-k \pm \sqrt{k^2 + 4}}{2},$$

each with algebraic multiplicity p. The remaining $2(m - p)$ eigenvalues satisfy

$$-\frac{\lambda^2 - \lambda}{\lambda^2 + k\lambda - 1} Gx = A^T W^{-1} A x. \tag{2.3}$$

Since G is positive semidefinite and $A^T W^{-1} A$ is positive definite, we must have

$$-\frac{\lambda^2 - \lambda}{\lambda^2 + k\lambda - 1} > 0,$$

thus we can write $\frac{\lambda^2 - \lambda}{\lambda^2 + k\lambda - 1} = -\mu^2$ for some $\mu \in \mathbb{R},\ \mu > 0$. We can rearrange this to

$$(1 + \mu^2)\lambda^2 + (k\mu^2 - 1)\lambda - \mu^2 = 0,$$

giving

$$\lambda_{\pm} = \frac{-(k\mu^2 - 1) \pm \sqrt{(k\mu^2 - 1)^2 + 4\mu^2(1 + \mu^2)}}{2(1 + \mu^2)}. \tag{2.4}$$

This expression gives an explicit formula in terms of the generalized eigenvalues of (2.3) and can be used to identify the intervals in which the eigenvalues lie. To illustrate this, we examine the case $k = -1$, which corresponds to setting the $(1, 2)$ block of the preconditioner to be $-A^T$. We have $\lambda = 1$ with multiplicity $n - m$, and $\lambda_\pm = \frac{1\pm\sqrt{5}}{2}$, each with multiplicity p. By (2.4) we have

$$\lambda_\pm = \frac{1 \pm \sqrt{1 + \frac{4\mu^2}{1+\mu^2}}}{2}.$$

Since λ_+ is a strictly increasing function of μ on $(0, \infty)$ (and λ_- are strictly decreasing), the intervals containing the remaining eigenvalues can be found using $\lim_{\mu\to 0,\infty} \lambda_\pm(\mu)$. From this one finds that the remaining eigenvalues lie in the intervals $(\frac{1-\sqrt{5}}{2}, 0) \cup (1, \frac{1+\sqrt{5}}{2})$. It is worth noting that since G is typically highly singular, many of the generalized eigenvalues are large, in which case the corresponding eigenvalues $\lambda_\pm$ are bounded away from zero. For example, many of the negative ones will tend to $\frac{1-\sqrt{5}}{2}$. Since the $2p$ eigenvalues $(k \pm \sqrt{k^2 + 4})/2$ are unbounded as k goes to ∞, we conclude that k should be of moderate size.

2.2. A Block Diagonal Preconditioner. The choice $k = 0$ yields a block diagonal symmetric positive definite preconditioner of the form

$$\mathcal{M} = \begin{pmatrix} G + A^T W^{-1} A & 0 \\ 0 & W \end{pmatrix}. \tag{2.5}$$

It is suitable for use with minimal residual methods based on short recurrences such as MINRES. This preconditioner and its spectral properties in a general algebraic context have been recently studied in [17] and applied (in a slightly different form) to the Maxwell equations in [18]. In the context of optimization, it is useful to examine the behavior on the reduced space generated by projections onto the null space of A. We offer such results below, and prove them using orthogonal transformations, by taking similar steps to those taken in [19].

As before, we assume that G is positive semidefinite with nullity p. Suppose further that A has full rank, and choose W to be symmetric positive definite. It is straightforward to show that if $\mathcal{A}$ is non-singular, then $G + A^T W^{-1} A$ must be symmetric positive definite. As is shown in [17], the eigenvalues of the preconditioned matrix $\mathcal{M}^{-1}\mathcal{A}$ and their multiplicities can be found directly through the generalized eigenvalue problem

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \lambda \begin{pmatrix} G + A^T W^{-1} A & 0 \\ 0 & W \end{pmatrix} \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}. \tag{2.6}$$

Let $QR = A^T$ be an "economy size" QR factorization of A^T, where Q is $n \times m$ and R is $m \times m$. Define Z to be a $n \times (n - m)$ orthogonal basis for the null space of A. Since $Z \cup Q$ forms an orthogonal basis for $\mathbb{R}^n$, any vector $x \in \mathbb{R}^n$ can be written as $x = Zx_z + Qx_q$.

Following the spirit of the proof of [19, Thm. 2.1], we define the $(n+m) \times (n+m)$ matrix

$$P = \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix}, \tag{2.7}$$

and perform a similarity transformation as follows. We express $\bar{x} = Z\bar{x}_z + Q\bar{x}_q$, and let $v = (x_z, x_q, y)^T$ where $Pv = (\bar{x}_z, \bar{x}_q, \bar{y})^T$. The generalized eigenvalue problem can then be written as $P^T\mathcal{A}Pv = \lambda P^T\mathcal{M}Pv$. This yields:

$$\begin{pmatrix} Z^TGZ & Z^TGQ & 0 \\ Q^TGZ & Q^TGQ & R \\ 0 & R^T & 0 \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = \lambda \begin{pmatrix} Z^TGZ & Z^TGQ & 0 \\ Q^TGZ & Q^TGQ + RW^{-1}R^T & 0 \\ 0 & 0 & W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix}. \tag{2.8}$$

By inspection, we observe that by setting $\lambda = 1$ the system reduces to

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & -RW^{-1}R^T & R \\ 0 & R^T & -W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Let e_i denote the i^{th} column of the identity matrix. Evidently there are $n - m$ corresponding eigenvectors that can be written in the form

$$(x_z, x_q, y) = (e_i, 0, 0). \tag{2.9}$$

In addition, m linearly independent eigenvectors can be written in the form:

$$(x_z, x_q, y) = (0, e_i, W^{-1}R^Te_i). \tag{2.10}$$

Now consider $\lambda = -1$. Equation (2.8) reduces to

$$\begin{pmatrix} 2Z^TGZ & 2Z^TGQ & 0 \\ 2Q^TGZ & 2Q^TGQ + RW^{-1}R^T & R \\ 0 & R^T & W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Any vector $x^* = Zx_z^* + Qx_q^*$ in the null space of G satisfies $G(Zx_z^* + Qx_q^*) = 0$. There are p such vectors, so p linearly independent eigenvectors of the form

$$(x_z, x_q, y) = (x_z^*, x_q^*, -W^{-1}R^Tx_q^*) \tag{2.11}$$

will satisfy (2.8) with $\lambda = -1$.

By [17, Theorem 2.2] the remaining eigenvalues, $\lambda \neq \pm 1$, lie in the interval $(-1, 0)$. To derive an expression for the corresponding eigenvectors we reduce equation (2.8) to an eigenvalue problem in x_q. From the block row in (2.8), $y = \frac{1}{\lambda}W^{-1}R^Tx_q$. The first line of the equation can be reduced to

$$x_z = -(Z^TGZ)^{-1}Z^TGQx_q.$$

Substituting this into the second line of (2.8) and simplifying yields

$$RW^{-1}R^Tx_q = \lambda\left[C^T(Z^TGZ)^{-1}C - Q^TGQx_q - RW^{-1}R^T\right]x, \tag{2.12}$$

where $C = Z^TGQ$. We have actually proved the following theorem.

THEOREM 2.1. *The preconditioned matrix $\mathcal{M}^{-1}\mathcal{A}$ has eigenvalues $\lambda = 1$ with multiplicity n with eigenvectors (2.9) and (2.10), and $\lambda = -1$ with multiplicity p, with corresponding eigenvectors given in (2.11). The remaining eigenvalues lie in the interval $(-1, 0)$ and satisfy the generalized eigenvalue problem (2.12).*

Theorem 2.1 illustrates the strong spectral clustering when the $(1, 1)$ block of $\mathcal{A}$ is singular. A well-known difficulty associated with interior point methods is the increased ill-conditioning of the $(1, 1)$ block as the solution is approached. Our claim is that the preconditioner performs robustly even as the problem becomes more ill-conditioned; in fact the outer iteration count decreases. On the other hand, solving for the augmented (1,1) block may be more computationally difficult and requires an effective approach, for example inexact solves.

3. Practical Considerations and Computational Cost. In this section we discuss the choice of the weight matrix W and ways of reducing the cost of inner iterations. We also describe procedures for dealing with a dense row.

3.1. The Inner Iteration and Choices of the Weight Matrix. There are two critical issues to consider in the application of the preconditioner. First, the weight matrix W must be chosen. Then, given a weight matrix, an efficient method of factoring or iteratively solving systems with the preconditioner must be sought. These considerations are motivated by the fact that each iteration of a preconditioned Krylov subspace method requires solutions to linear systems of the form $\mathcal{M}x = b$; based on the block structure of $\mathcal{M}$, this requires solving systems with $G + A^T W^{-1} A$ and W.

The simplest choice of a weight matrix is diagonal, and it clearly makes inverting W trivial. A simple, one-parameter choice is a scaled identity. Letting $W = \gamma I$, γ could be chosen so that the augmenting term $\frac{1}{\gamma} A^T A$ is of norm comparable to G. See, for example, [14] for a general algebraic discussion. Note that since G changes at each step, γ must also be updated.

For LPs the matrix G is diagonal and choosing $1/\gamma$ related to an ordered statistic such as the mean, median or maximum of the diagonal entries in G, has proven to be effective in our experiments in reducing the number of MINRES iterations required at each step. This is illustrated in Figure 3.1 for the Netlib problem "tuff", where the MINRES residual norms (in the predictor step) are plotted against each step of the LP solve. An arbitrary fixed choice of $1/\gamma$ leads to a "hump" with a large number of intermediate outer iteration counts. As predicted by our theoretical results, close to the solution the near-singularity of the (1,1) block results in fast convergence regardless of the choice of γ. But as is illustrated in the figure, it is the dynamic choice of γ^{-1} as the maximal entry in G that yields rapid convergence of MINRES throughout the LP iteration, and in fact the "hump" is flattened in this case. The choice $1/\gamma = \max(G)$ results in a set of values monotonically increasing from approximately 1 to approximately 10^{10}, and the iteration significantly outperforms other choices in terms of number of MINRES iterations, while the cost of each inner solve does not change.

For QPs, similar approaches are possible. With $W = \gamma I$, a choice of γ such as $\|A\|^2/\|G\|$ (or an approximation thereof) ensures the norm of the augmenting term is not too small in comparison with G. (See [14] for an analysis of the effect of scaling of this sort on the condition number of the matrix.) For solving $\mathcal{M}x = b$, iterative and direct methods are possible. In an iterative scheme the inner iteration can be solved using the preconditioned conjugate gradient (PCG) method. If a direct solver is preferred, the symmetric positive definiteness of the preconditioner allows for use of a sparse Cholesky factorization.

3.2. Dealing with a Dense Row in A. The presence of even a single dense row in A can lead to a fully dense augmenting matrix $A^T W^{-1} A$. We present two possible approaches for dealing with dense rows in the situation that it is desirable to explicitly form the $(1,1)$ block of the preconditioner.

First, we present an asymmetric preconditioner, motivated by the analysis of Section 2.1. With a_i denoting the dense column i of A^T, and e_i being the i^{th} column of an $m \times m$ identity, we define a preconditioner

$$\hat{\mathcal{M}} = \begin{pmatrix} G + A^T \bar{W} A & -a_i e_i^T \\ 0 & W \end{pmatrix}. \tag{3.1}$$

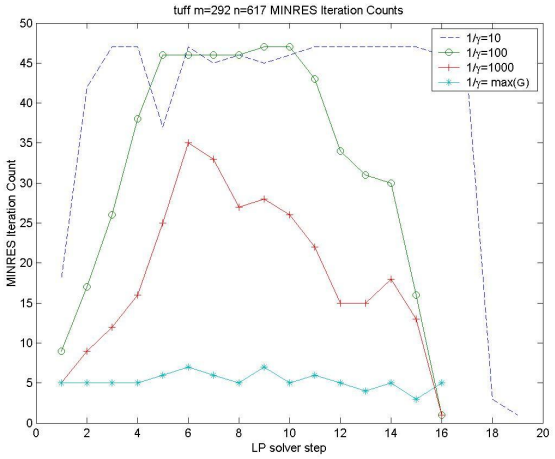

FIG. 3.1. *MINRES iteration counts with various* $W = \gamma I$. *Problem "tuff" has* $m = 292$, $n = 617$ *after LIPSOL preprocessing. MINRES iteration counts are plotted at each LP step for the various choices of* γ. *MINRES error tolerance was set to* 10^{-8}. *In final iterations,* $\max(G) > 10^8$.

Suppose $W = \gamma I$ for some $\gamma > 0$, and let $\bar{W} = \frac{1}{\gamma} I - \frac{1}{\gamma} e_i e_i^T$. Assuming $\hat{\mathcal{M}}$ is non-singular, the eigenvalues of the preconditioned matrix are given by the following theorem.

THEOREM 3.1. *The preconditioned matrix* $\hat{\mathcal{M}}^{-1}\mathcal{A}$ *has* $\lambda = 1$ *with multiplicity* $n-1$ *and* $\lambda = -1$ *with multiplicity* p. *Corresponding eigenvectors can be explicitly found in terms of the null space and column space of* A.

Proof. Exactly as in the proof of Theorem 2.1 we define Q, R, Z and transform the generalized eigenvalue problem using P as in (2.7). This yields

$$\begin{pmatrix} Z^TGZ & Z^TGQ & 0 \\ Q^TGZ & Q^TGQ & R \\ 0 & R^T & 0 \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = \lambda \begin{pmatrix} Z^TGZ & Z^TGQ & 0 \\ Q^TGZ & Q^TGQ + \frac{1}{\gamma} RR^T - \frac{1}{\gamma} r_i r_i^T & -r_i e_i^T \\ 0 & 0 & \gamma I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix}, \tag{3.2}$$

where r_i denotes the i^{th} column of R.

As before, by inspection we check $\lambda = 1$, which reduces the equation to

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & -\frac{1}{\gamma} RR^T + \frac{1}{\gamma} r_i r_i^T & R + r_i e_i^T \\ 0 & R^T & -\gamma I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = 0.$$

Immediately we see $n-m$ corresponding eigenvectors of the form $(x_z, x_q, y) = (u, 0, 0)$, for $(n-m)$ linearly independent vectors u. An additional $m-1$ linearly independent eigenvectors can be seen by finding consistent solutions in the free variables x_q, y to the equation

$$\begin{pmatrix} -\frac{1}{\gamma} RR^T + \frac{1}{\gamma} r_i r_i^T & R + r_i e_i^T \\ R^T & -\gamma I \end{pmatrix} \begin{pmatrix} x_q \\ y \end{pmatrix} = 0.$$

Substituting $y = \frac{1}{\gamma} R^T x_q$, this requires $2\frac{1}{\gamma} r_i r_i^T x_q = 0$. In general we can find exactly $m-1$ eigenvectors orthogonal to r_i. That is, there are $m-1$ eigenvectors of the form $(x_z, x_q, y) = (0, x_q^*, \frac{1}{\gamma} x_q^*)$, where x_q^* is orthogonal to r_i, corresponding to $\lambda = 1$.

The p eigenvectors corresponding to $\lambda = -1$ are also evident by simple inspection. Substituting $\lambda = -1$ requires finding a solution to:

$$\begin{pmatrix} 2Z^TGZ & 2Z^TGQ & 0 \\ 2Q^TGZ & 2Q^TGQ + \frac{1}{\gamma}RR^T - \frac{1}{\gamma}r_ir_i^T & R - r_ie_i^T \\ 0 & R^T & \gamma I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = 0$$

Vectors x_z, x_q, y can be found to solve this equation. Consider any $x^* = Zx_z^* + Qx_q^*$ in the null space of G. Then $GZx_z^* + GQx_q^* = 0$, and we are left with finding a y such that

$$\begin{pmatrix} \frac{1}{\gamma}RR^T - \frac{1}{\gamma}r_ir_i^T & R - r_ie_i^T \\ R^T & \gamma I \end{pmatrix} \begin{pmatrix} x_q^* \\ y \end{pmatrix} = 0$$

for the fixed x_q^*. The choice $y = -\frac{1}{\gamma}R^Tx_q^*$ correctly cancels the left hand side, and it becomes apparent why the minus sign was chosen for the $(1,2)$ block of $\hat{\mathcal{M}}$; without it, we could not explicitly find a suitable y value. Since the p vectors in the null space of G are linearly independent, we have constructed a p-dimensional eigenbasis for $\hat{\mathcal{M}}^{-1}\mathcal{A}$, corresponding to $\lambda = -1$. □

Theorem 3.1 shows that $\hat{\mathcal{M}}$ is sparse and at the same time maintains strong spectral clustering. The preconditioner is asymmetric, though, and it is desirable to find a simpler form (that still retains the strong spectral properties).

To this end, consider replacing preconditioner $\hat{\mathcal{M}}$ from Equation (3.1) with

$$\bar{\mathcal{M}} = \begin{pmatrix} G + A^T\bar{W}A & 0 \\ 0 & W \end{pmatrix}$$

where $\bar{W}$ is again an approximation to W^{-1}. Similarly to the asymmetric case, consider the choice

$$\bar{W} = W^{-1} - \frac{1}{\gamma}e_ie_i^T.$$

The matrix $\bar{W}$ is diagonal but singular, since each of its rows that corresponds to a dense row of A is identically zero. As a result, the matrix $A^T\bar{W}A$ no longer experiences fill-in from the product $a_ia_i^T$.

This modification does not result in significant changes in the spectral clustering of the preconditioned matrix. Since $\bar{\mathcal{M}}$ is a rank-1 perturbation of $\mathcal{M}$ by (2.5), it follows that $\bar{\mathcal{M}}^{-1}\mathcal{A}$ is just a rank-1 perturbation of $\mathcal{M}^{-1}\mathcal{A}$ and we can apply interlacing results.

If we let μ_i denote the i^{th} largest eigenvalue of $\bar{\mathcal{M}}^{-1}\mathcal{A}$, and λ_i be the i^{th} largest eigenvalue of $\mathcal{M}^{-1}\mathcal{A}$, interlacing guarantees that $\lambda_{i-1} \leq \mu_i \leq \lambda_i$. Since the eigenvalues λ_i are known and have high algebraic multiplicities, so are the eigenvalues μ_i, and for each multiple eigenvalue λ_i the multiplicity of the corresponding μ_i goes down by at most 1, due to interlacing. Thus, if preconditioned MINRES is used, we have strong spectral clustering without introducing any fill-in. We can summarize our findings as follows.

PROPOSITION 3.2. *Assume $\bar{\mathcal{M}}$ is non-singular. Then the preconditioned matrix $\bar{\mathcal{M}}^{-1}\mathcal{A}$ has $\lambda = 1$ with multiplicity at least $n-1$, and $\lambda = -1$ with multiplicity at least $p-1$.*

4. Numerical Experiments. Numerical experiments were done on problems from the CUTEr test suite [16], using MATLAB, on an Intel 2.5GHZ processor with 2GB of RAM. In our experiments we focused on QP test problems with a non-diagonal and semidefinite $(1,1)$ block, for which our preconditioner is suitable. In this section we illustrate how the number of iterations required by MINRES drops to its theoretical limit and how inexact inner iterations reduce the overall computational work. (For the latter we settle for experimental observations; see, for example, [15] or [27] for general theory and analysis.) We also include results for the row removal scheme discussed in Section 3.2.

We used a variety of methods for solving the inner iteration $\mathcal{M}x = b$, but most of our experiments made use of ILUPACK [6]. This package uses multilevel incomplete LU factorizations as preconditioners for CG and GMRES, and was found to be efficient and easy to use. While a range of values were tested, the results we present here were collected with an ILUPACK backwards error setting of 1e-06.

Tables 4.1–4.4 demonstrate several measures of the work required in applying our method. In these tables, n, m_1, m_2 denote the dimensions of the problem being solved, as defined in the Introduction, and N_{QP} denotes the total number of QP steps required for convergence. The average number of Krylov solver iterations per QP steps is given by N_K. The average number of iterations of PCG used by ILUPACK in the inner iteration, and the total number summed over all QP steps are given by N_I and $Tot(N_I)$ respectively. The time (in seconds) required to solve a problem is given in the column Time(s).

Tables 4.1–4.2 show results for applying BiCG-STAB, once with a tight outer tolerance of 10^{-6} and once with a loose outer tolerance of 10^{-2}. Tables 4.3–4.4 show results using the same tolerances, with MINRES. The following observations can be made. In general, loosening tolerance for the Krylov solver increases the overall number of QP iterations only modestly, and at the same time substantially reduces (in most tested examples) the overall number of solves. We mention here that loosening the convergence tolerance of the inner-most iterations did not result in a similar reduction of computational work. We therefore observe in our experiments that inexactness is more effective on the level of the outer Krylov iterations rather than on the level of the inner-most iterations.

Comparing the performance of BiCG-STAB to the performance of MINRES is not within our stated goals, but having results using more than one Krylov solver allows us to confirm the consistency of convergence behavior for most problems. The two solvers do behave in a similar manner, and the modest running times indicate that the proposed preconditioner seems to be efficient and robust.

Next, to demonstrate the application of the row removal scheme proposed in Section 3.2 we consider the "blockqp" set of problems. These problems are characterized by a Hessian with two non-zero diagonals, and a constraint matrix with a single non-zero diagonal, a dense row, and a dense column. As a result, if the augmentation preconditioner is fully formed, it will be dense. To avoid this, the symmetric row removal scheme is used. This leads to a preconditioner with a nearly diagonal $(1,1)$ block, which can be approximately factored with ILUPACK in an efficient manner. Table 4.5 presents results confirming that the row removal scheme can lead to a convergent solution while avoiding the intensive memory requirements of fully forming the augmenting $(1,1)$ block of the preconditioner.

Theorem 2.1 guarantees increased spectral clustering of the preconditioned matrix $\mathcal{M}^{-1}\mathcal{A}$ when the $(1,1)$ block of $\mathcal{A}$ is singular. The LP and QP saddle point matrices,

Problem	n	m_1	m_2	N_{QP}	N_K	N_I	$Tot(N_I)$	Time (s)
$avgasa$	12	10	18	5	2.80	2.00	112	0.22
$avgasb$	12	10	18	5	2.75	2.00	110	0.21
$blockqp1-10$	26	11	51	4	1.75	2.71	76	0.15
$blockqp2-10$	26	11	51	4	1.81	3.07	89	0.14
$blockqp3-10$	26	11	51	6	2.00	3.42	164	0.25
$blockqp4-10$	26	11	51	5	2.50	3.44	172	0.22
$blockqp5-10$	26	11	51	6	2.38	3.25	185	0.28
$cvxqp1-100$	100	50	200	34	3.66	4.67	3840	3.75
$cvxqp2-100$	100	25	200	13	3.06	3.57	567	0.81
$cvxqp3-100$	100	75	200	13	4.06	3.47	732	1.10
$dual1$	85	1	170	7	1.75	4.26	362	0.99
$dual2$	96	1	192	5	1.50	3.37	101	0.57
$dual3$	111	1	222	5	1.50	3.07	92	0.71
$dual4$	75	1	150	5	1.45	3.55	103	0.37
$gouldqp2$	699	349	1398	10	1.82	2.08	152	1.00
$gouldqp2-30$	59	29	118	7	1.64	2.59	119	0.25
$gouldqp3$	699	349	1398	11	1.52	3.03	203	1.07
$gouldqp3-30$	59	29	118	5	1.55	2.71	84	0.17
$steenbra$	432	108	432	11	1.86	5.28	433	2.12

TABLE 4.1

Solver results using BiCG-STAB. Problems were solved to a tolerance of 1.0e-06. BiCG-STAB tolerance was fixed at 1.0e-02.

Problem	n	m_1	m_2	N_{QP}	N_K	N_I	$Tot(N_I)$	Time (s)
$avgasa$	12	10	18	3	12.00	2.00	288	0.53
$avgasb$	12	10	18	4	10.56	2.00	338	0.48
$blockqp1-10$	26	11	51	4	2.12	2.59	88	0.15
$blockqp2-10$	26	11	51	4	2.31	3.08	114	0.16
$blockqp3-10$	26	11	51	5	7.60	3.01	457	0.53
$blockqp4-10$	26	11	51	5	5.60	3.96	444	0.39
$blockqp5-10$	26	11	51	6	6.75	3.09	500	0.56
$cvxqp1-100$	100	50	200	13	21.44	3.94	5002	4.74
$cvxqp2-100$	100	25	200	12	10.27	4.01	2008	1.91
$cvxqp3-100$	100	75	200	11	22.91	3.73	4107	4.30
$dual1$	85	1	170	7	2.07	5.42	716	1.38
$dual2$	96	1	192	5	1.50	3.37	101	0.56
$dual3$	111	1	222	5	1.50	3.07	92	0.73
$dual4$	75	1	150	5	3.50	4.40	308	0.64
$gouldqp2$	699	349	1398	9	4.28	2.05	315	1.59
$gouldqp2-30$	59	29	118	5	5.00	2.00	200	0.35
$gouldqp3$	699	349	1398	10	3.45	2.69	371	1.64
$gouldqp3-30$	59	29	118	5	3.55	2.70	192	0.29
$steenbra$	432	108	432	11	6.82	5.30	1591	5.31

TABLE 4.2

Solver results using BiCG-STAB. Problems were solved to a tolerance of 1.0e-06. BiCG-STAB tolerance was fixed at 1.0e-06.

Problem	n	m_1	m_2	N_{QP}	N_K	N_I	$Tot(N_I)$	Time (s)
$avgasa$	12	10	18	4	4.00	2.00	96	0.19
$avgasb$	12	10	18	5	3.70	2.00	114	0.23
$blockqp1-10$	26	11	51	4	1.50	2.86	80	0.16
$blockqp2-10$	26	11	51	4	1.38	3.07	86	0.15
$blockqp3-10$	26	11	51	6	1.92	3.79	178	0.24
$blockqp4-10$	26	11	51	5	2.00	3.70	148	0.20
$blockqp5-10$	26	11	51	6	2.58	3.45	190	0.26
$cvxqp1-100$	100	50	200	21	8.74	4.29	2000	2.40
$cvxqp2-100$	100	25	200	13	3.31	3.59	496	0.77
$cvxqp3-100$	100	75	200	16	5.94	3.68	946	1.47
$dual1$	85	1	170	7	1.57	5.91	384	0.88
$dual2$	96	1	192	5	1.00	3.27	98	0.58
$dual3$	111	1	222	5	1.00	3.07	92	0.74
$dual4$	75	1	150	5	2.80	5.02	266	0.57
$gouldqp2$	699	349	1398	10	1.65	2.14	156	1.09
$gouldqp2-30$	59	29	118	6	1.50	2.33	98	0.21
$gouldqp3$	699	349	1398	13	1.35	3.14	273	1.48
$gouldqp3-30$	59	29	118	6	1.58	2.86	123	0.24
$steenbra$	432	108	432	12	6.38	6.92	1390	4.14

TABLE 4.3

Solver results using MINRES. Problems were solved to a tolerance of 1.0e-06. MINRES tolerance was fixed at 1.0e-02.

Problem	n	m_1	m_2	N_{QP}	N_K	N_I	$Tot(N_I)$	Time (s)
$avgasa$	12	10	18	4	12.25	2.00	236	0.41
$avgasb$	12	10	18	4	11.25	2.00	220	0.38
$blockqp1-10$	26	11	51	4	1.75	2.73	82	0.15
$blockqp2-10$	26	11	51	4	1.75	3.07	92	0.16
$blockqp3-10$	26	11	51	5	8.10	2.85	288	0.42
$blockqp4-10$	26	11	51	5	7.20	3.96	404	0.42
$blockqp5-10$	26	11	51	6	7.25	2.99	332	0.46
$cvxqp1-100$	100	50	200	14	21.39	4.06	3072	3.69
$cvxqp2-100$	100	25	200	12	14.71	3.99	1641	1.95
$cvxqp3-100$	100	75	200	14	17.82	3.78	2779	3.66
$dual1$	85	1	170	7	2.64	6.66	586	1.08
$dual2$	96	1	192	5	1.00	3.27	98	0.58
$dual3$	111	1	222	5	1.00	3.07	92	0.72
$dual4$	75	1	150	5	4.60	5.65	418	0.72
$gouldqp2$	699	349	1398	9	6.22	2.03	300	2.00
$gouldqp2-30$	59	29	118	5	7.10	2.00	182	0.39
$gouldqp3$	699	349	1398	10	4.50	2.67	347	1.94
$gouldqp3-30$	59	29	118	5	4.80	2.68	182	0.32
$steenbra$	432	108	432	11	31.18	5.93	4524	13.12

TABLE 4.4

Solver results using MINRES. Problems were solved to a tolerance of 1.0e-06. MINRES tolerance was fixed at 1.0e-06.

Problem	N_{QP}	N_K	N_I	Time (s)
*blockqp*1	3	3.00	7.67	1.78
*blockqp*2	4	3.00	8.00	2.38
*blockqp*3	8	30.69	9.44	27.28
*blockqp*4	6	18.88	8.28	17.31
*blockqp*5	8	25.38	8.21	23.32

TABLE 4.5

Results obtained using the symmetric dense row removal scheme of Section 3.2. Problems were solved using MINRES with error tolerance 1.0e-05. Problems solved to accuracy 1.0e-4. For each "blockqp" problem, $n = 2006$, $m_1 = 1001$, $m_2 = 4011$.

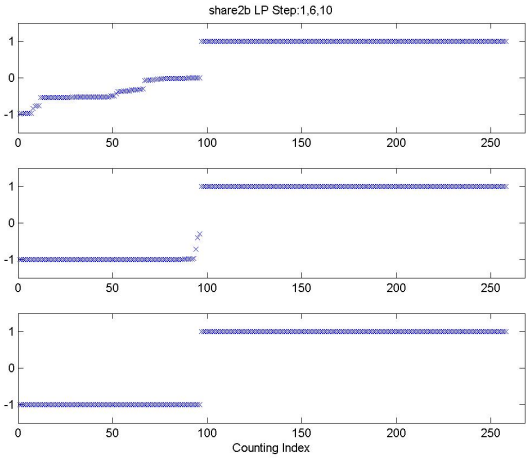

FIG. 4.1. *Eigenvalues of* $\mathcal{M}^{-1}\mathcal{A}$ *at different steps of the LP solution for "share2b". Eigenvalues are plotted in sorted order with values along the y axis. As governed by Theorem 2.1, all unclustered eigenvalues lie in the interval* $(-1, 0)$.

however, only become approximately singular as a solution is approached. It is useful to evaluate whether the strong clustering of the preconditioned eigenvalues will be achieved under approximate conditions. To test this, we examined the eigenvalues of the preconditioned matrix at various steps in the process of solving an LP. Figure 4.1 depicts the sorted eigenvalues at three different steps of the LP solve for the problem "share2b".

Preconditioned eigenvalues at the first, sixth, and tenth LP steps are shown from top to bottom. (The problem took 13 steps to solve.) In confirmation of Theorem 2.1, we see that all eigenvalues lie between -1 and 1. Furthermore, right from the first iteration $\lambda = 1$ has high multiplicity. It is interesting to note that already by the sixth step (the middle subplot), only a handful of unclustered eigenvalues remain. In the tenth LP step, all eigenvalues appear to be ± 1. These observations all confirm Theorem 2.1, and illustrate how tightly clustered the preconditioned eigenvalues can be when the saddle point system is severely ill-conditioned. This also demonstrates that even in early iterations the preconditioner can be effective.

Next, we present preliminary comparisons with constraint preconditioners. We stress that we have only performed a limited set of tests with a specific simple choice of a (1,1) block for constraint preconditioners; a comprehensive set of tests will be

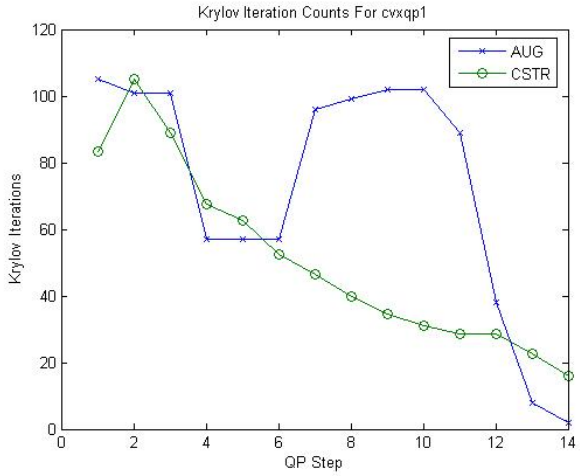

FIG. 4.2. *MINRES iteration counts for "cvxqp1", $n = 1000, m_1 = 250$. Constraint preconditioner iterations are represented by 'o', augmentation preconditioner iterations are represented by 'x'. The constraint preconditioner is consistently better. Note in the final few iterations, though, iteration counts for the augmentation preconditioner significantly decrease. This is due to the increased singularity of the $(1,1)$ block.*

necessary to reach definite conclusions. Our results are mixed, as we illustrate below. Figure 4.2 shows a basic comparison of MINRES iteration counts (of the predictor computation). The plot shows the MINRES iteration counts for each QP step for the problem "cvxqp1". The constraint preconditioner used for this plot was chosen to have a $(1,1)$ block equal to the diagonal of the saddle point system. For both preconditioners, an exact inner solve was applied. For this problem the constraint preconditioner outperformed the augmentation preconditioner in most steps of the QP solve and in terms of overall computational work. In the final few steps, however, where the saddle point matrix was highly ill-conditioned, MINRES iteration counts for our preconditioner dropped significantly and convergence was almost immediate, whereas convergence of the constraint preconditioner was still within approximately 20 iterations. This again confirms Theorem 2.1 and indicates that the proposed preconditioner is most effective when the saddle point matrix is close to singular. This in fact may suggest a hybrid approach in which it may be useful to switch to an augmentation-based preconditioner when iterates approach the solution.

Figure 4.3 shows an example in which our preconditioner is superior the constraint preconditioner throughout the iteration. For the QP problem "cvxqp2", MINRES iteration counts (of the predictor computation) are plotted against the QP step. This is a large problem but throughout the solve no more than 30 iterations are needed per step. In the final few QP steps, the MINRES iteration count approaches its theoretical limit of two. Depicted in the same plot are the corresponding predictor iteration counts for a constraint preconditioner, with $(1,1)$ block set to match the diagonal of the saddle point system. The constraint preconditioner consistently requires more MINRES iterations at each QP step.

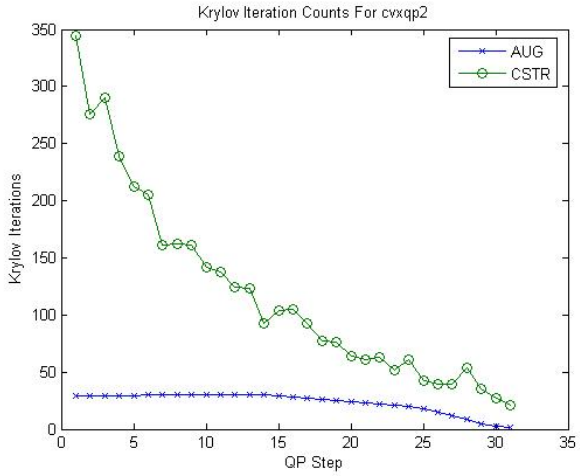

FIG. 4.3. *MINRES iteration counts for "cvxqp2", $n = 10000, m_1 = 2500$. Constraint preconditioner iterations are represented by 'o', augmentation preconditioner iterations are represented by 'x'. The augmentation preconditioner is consistently better and approaches theoretical convergence. The final two iteration counts of MINRES are 3 and 2, respectively.*

5. Conclusions. We have studied preconditioners for linear and quadratic programming problems, and have demonstrated their merits. The preconditioners are well suited for saddle point systems with a highly singular (1,1) block. For interior point iterations, as convergence is approached and the $(1, 1)$ block of the saddle point matrix becomes more ill-conditioned, convergence is the fastest and can theoretically occur within two outer iterations (in the absence of roundoff errors). We have also provided spectral analysis on the null space of the constraint matrix.

The choice of the weight matrix W is crucial and we have tested with simple scaled identity matrices that incorporate a dynamic choice of the parameter throughout the iteration. A good choice may substantially reduce the iteration counts throughout, as we demonstrate by the "flattened hump" for an LP problem in Figure 3.1. Furthermore, we have shown that applying an inexact version of the Krylov solver throughout the iteration, with a convergence tolerance as low as 0.01, significantly reduces the overall amount of computational work.

The excellent performance of the preconditioner in particular near the solution may suggest that regardless of what preconditioner is used throughout the iteration, it may be worthwhile to switch to a preconditioner of the form explored here as the iterates approach the solution.

Acknowledgments. We would like to thank the two referees for their constructive comments.

REFERENCES

[1] S. Bellavia. Inexact interior-point method. *J. Optim. Theory Appl.*, 96(1):109–121, 1998.
[2] S. Bellavia and S. Pieraccini. Convergence analysis of an inexact infeasible interior point method for semidefinite programming. *Compt. Optim. Appl.*, 29(3):289–313, 2004.
[3] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
[4] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli. Inexact constraint preconditioners for

linear systems arising in interior point methods. Technical Report MS-05-002, School of Mathematics, The University of Edinburgh, 2005.
[5] L. Bergamaschi, J. Gondzio, and G. Zilli. Preconditioning indefinite systems in interior point methods for optimization. *Compt. Optim. Appl.*, 28:149–171, 2004.
[6] M. Bolhöfer and Y. Saad. Multilevel preconditioners constructed from inverse based ILUs. *SIAM J. Sci. Compt.*, 27:1627–1650, 2006.
[7] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia PA, 1997.
[8] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen. Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems. *SIAM J. Matrix Anal. Appl.*, 28(1):170–189, 2006.
[9] H. S. Dollar and A. J. Wathen. Approximate factorization constraint preconditioners for saddle-point matrices. *SIAM J. Sci. Compt.*, 27(5):1555–1572, 2006.
[10] A. Forsgren, P. E. Gill, and J. Griffin. Iterative solution of augmented systems arising in interior methods. Technical Report NA-05-03, UCSD Department of Mathematics, 2005.
[11] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Rev.*, 44(4):525–597, 2002.
[12] R. W. Freund, F. Jarre, and S. Mizuno. Convergence of a class of inexact interior-point algorithms for linear programs. *Math. Oper. Res.*, 24(1):50–71, 1999.
[13] E. M. Gertz and S. J. Wright. Object-oriented software for quadratic programming. *ACM Trans. Math. Softw.*, 29(1):58–81, 2003.
[14] G.H. Golub and C. Greif. On solving block structured indefinite linear systems. *SIAM J. Sci. Compt.*, 24(6):2076–2092, 2003.
[15] G.H. Golub and M.L. Overton. The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems. *Numerische Mathematik*, 53:571–593, 1998.
[16] N. I. M. Gould, D. Orban, and P. L. Toint. Cuter and sifdec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Softw.*, 29(4):373–394, 2003.
[17] C. Greif and D. Schötzau. Preconditioners for saddle point linear systems with highly singular (1,1) blocks. *ETNA, Special Volume on Saddle Point Problems*, 22:114–121, 2006.
[18] C. Greif and D. Schötzau. Preconditioners for the discretized time-harmonic Maxwell equations in mixed form. *Numer. Linear Alg. Appl.*, accepted, 2006.
[19] C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Anal. Appl.*, 21(4):1300–1317, 2000.
[20] L. Lukšan and J. Vlček. Indefinitely preconditioned inexact newton method for large sparse equality constrained non-linear programming problems. *Numer. Linear Alg. Appl.*, 5(3):219–247, 1998.
[21] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. Optim*, 2:575–601, 1992.
[22] E. Ng and B. W. Peyton. Block sparse cholesky algorithms on advanced uniprocessor computers. *SIAM J. Sci. Compt.*, 14:617–629, 1995.
[23] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.
[24] A. Oliveira and D. Sorensen. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. Technical Report CRPC-TR97771, Center for Research on Parallel Computation, Rice University, 1997.
[25] T. Rees. Augmentation preconditioning for saddle point systems. Master's thesis, Department of Computer Science, University of British Columbia, August 2006.
[26] Y. Saad. *Iterative Methods for Sparse Linear Systems, Second Edition*. SIAM, Philadelphia PA, 2003.
[27] V. Simoncini and D.B. Szyld. Theory of inexact krylov subspace methods and applications to scientific computing. *SIAM J. Sci. Compt.*, 25:454–477, 2003.
[28] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math. Prog.*, 106(1):25–57, 2006.
[29] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, PA, 1997.
[30] Y. Zhang. Solving large-scale linear programs by interior-point methods under the MATLAB environment. *Optim. Meth. Softw.*, 10:1–31, 1998.