A New Class of Preconditioners for Large-Scale Linear Systems from Interior Point Methods for Linear Programming

A.R.L. Oliveira and D.C. Sorensen

CRPC-TR97771
November 1997

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Submitted September 1998; Available as Rice CAAM TR97-27; In review for *SIAM J. Optimization*

A NEW CLASS OF PRECONDITIONERS FOR LARGE-SCALE LINEAR SYSTEMS FROM INTERIOR POINT METHODS FOR LINEAR PROGRAMMING

A. R. L. OLIVEIRA* AND D. C. SORENSEN†

Abstract. A new class of preconditioners is proposed for the iterative solution of linear systems arising from interior point methods. In many cases, these linear systems are symmetric and indefinite. Typically, these indefinite systems can be reduced to an equivalent Schur complement system which is positive definite. We show that all preconditioners for the Schur complement system have an equivalent for the augmented system while the opposite is not true. This suggests it may be better to work with the augmented system. We develop some theoretical properties of the new preconditioners to support this. Computationally, we have verified that this class works better near a solution of the linear programming problem when the linear systems are highly ill-conditioned. Preliminary experiments which illustrate these features are presented. The techniques developed for a competitive implementation are presented in a follow up paper along with numerical experiments on several classes of linear programming problems.

Key words. linear programming, interior-point methods, preconditioners, augmented system

AMS subject classifications. 65F50, 90C05, 90C06

Abbreviated Title: Preconditioners for Interior Point Methods.

1. Introduction.

Interior point methods have been used successfully for solving linear programming problems for about a decade now. Their good performance in practice and their theoretical properties have motivated the implementation of sophisticated codes to solve large scale problems. These methods are effective because they converge in relatively few iterations. However an iteration of an interior point method is more expensive than the iterations of the traditional simplex method.

Each iteration of an interior point method involves the solution of one or more linear systems. There are several ways for solving the linear systems. The most common approach reduces the indefinite augmented system to a smaller positive definite one called the Schur complement. After this reduction, the solution for the linear system is computed via the Cholesky factorization for the majority of the implementations.

In this work, we study efficient ways to solve the augmented linear system by iterative methods. Special attention is given to the choice of a preconditioner. This is a key point in applying iterative methods for solving linear systems. We will show that every preconditioner for the reduced system yields an equivalent preconditioner for the augmented system but the converse is not true. Therefore, we choose to work with the augmented system because of the greater opportunity to find an effective preconditioner.

We propose a new class of preconditioners which avoid computing the Schur complement. These preconditioners rely on an LU factorization of a subset of columns of the constraint matrix instead. We show some theoretical properties of the preconditioned matrix and reduce it to positive definite systems. The discussion on how

* State University of Campinas UNICAMP, São Paulo 13083-970 (aurelio@densis.fee.unicamp.br)
This work was supported in part by the Brazilian Council for the Development of Science and Technology CNPq, and by FAPESP Fundação de Amparo à Pesquisa do Estado de São Paulo,

† Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 77005-1892 (sorensen@caam.rice.edu). This work was supported in part by NSF cooperative agreement CCR-9120008, and by ARPA contract number DAAL03-91-C-0047 (administered by the U.S. Army Research Office) .

to implement these preconditioners efficiently is presented in [17]. We report a few preliminary numerical results to illustrate the features of this preconditioner.

We use the following notation throughout this work. Lower case Greek letters denote scalars, lower case Latin letters denote vectors and upper case Latin letters denote matrices. Components of matrices and vectors are represented by the corresponding Greek letter with subscripts. The symbol 0 will denote the scalar zero, the zero column vector and the zero matrix, its dimension will be clear from context. The identity matrix will be denoted by I, a subscript will determine its dimension when it is not clear from context. The Euclidean norm is represented by $\|\cdot\|$ which will also represent the 2-norm for matrices. The relation $X = diag(x)$ means that X is a diagonal matrix whose the diagonal entries are the components of x. On the other hand, $diag(A)$ means the column vector formed by the diagonal entries of A. A superscript k for a scalar, vector or matrix will denote their value at the kth step of an iterative procedure.

2. Linear Programming Problems. Consider the linear programming problem in the *standard form*:

$$\begin{array}{ll} \text{minimize} & c^t x \\ \text{subject to} & Ax = b, \quad x \geq 0, \end{array} \tag{2.1}$$

where A is a full row rank $m \times n$ matrix and c, b and x are column vectors of appropriate dimension. Associated with problem (2.1) is the dual linear programming problem

$$\begin{array}{ll} \text{maximize} & b^t y \\ \text{subject to} & A^t y + z = c, \quad z \geq 0, \end{array} \tag{2.2}$$

where y is a m-vector of free variables and z is the n-vector of dual slack variables.

The optimality conditions for (2.1) and (2.2) can be written as a nonlinear system of equations with some nonnegative variables:

$$\begin{pmatrix} Ax - b \\ A^t y + z - c \\ XZe \end{pmatrix} = 0, \quad (x, z) \geq 0, \tag{2.3}$$

where $X = diag(x)$ and $Z = diag(z)$ and e is the vector of all ones.

2.1. Mehrotra's Predictor-Corrector Method. The majority of the primal-dual interior point methods found in the literature can be seen as variants of Newton's method applied to the optimality conditions (2.3). Soon after its appearance, the predictor-corrector variant [16] became the most popular variant of the primal-dual interior point methods. The search directions are obtained by solving two linear systems. First we solve

$$\begin{pmatrix} 0 & I & A^t \\ Z^k & X^k & 0 \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta \tilde{x}^k \\ \Delta \tilde{z}^k \\ \Delta \tilde{y}^k \end{pmatrix} = \begin{pmatrix} r_d^k \\ r_a^k \\ r_p^k \end{pmatrix} \tag{2.4}$$

where,

$$\left\{ \begin{array}{lcc} r_d^k & = & c - A^t y^k - z^k \\ r_a^k & = & -X^k Z^k e \\ r_p^k & = & b - Ax^k. \end{array} \right.$$

and $\Delta\tilde{x}^k, \Delta\tilde{z}^k$ and $\Delta\tilde{y}^k$ are called affine directions. Then, the search directions are given by

$$\begin{pmatrix} 0 & I & A^t \\ Z^k & X^k & 0 \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta z^k \\ \Delta y^k \end{pmatrix} = \begin{pmatrix} r_d^k \\ r_m^k \\ r_p^k \end{pmatrix} \tag{2.5}$$

where,

$$r_m^k = \mu^k e - X^k Z^k e - \Delta\tilde{X}^k \Delta\tilde{Z}^k e,$$

and $\mu^k = (\frac{\tilde{\gamma}^k}{\gamma^k})^2(\frac{\tilde{\gamma}^k}{n^{1.5}})$ with $\gamma^k = (x^k)^t z^k$ and $\tilde{\gamma}^k = (x^k + \Delta\tilde{x}^k)^t(z^k + \Delta\tilde{z}^k)$.

This variant significantly reduces the number of iterations for the primal-dual method. This reduction is obtained at the price of solving two linear systems per iteration. However, these linear systems have the same coefficient matrix. In practice the work for solving the extra linear system is easily compensated by the savings on the reduction of the number of iterations.

2.2. The Search Directions. The key step in a given iteration in terms of computational cost is the solution of linear systems (2.4 and (2.5)). Eliminating Δz^k from system (2.4) it reduces to:

$$\begin{pmatrix} -D^k & A^t \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix} = \begin{pmatrix} r_d^k - (X^k)^{-1} r_a^k \\ r_p^k \end{pmatrix} \tag{2.6}$$

where, $D^k = (X^k)^{-1}Z^k$. We refer to (2.6) as the *augmented system*. Eliminating Δx^k from (2.6) we get

$$S^k \Delta y^k = r_p^k + A((D^k)^{-1} r_d^k - (Z^k)^{-1} r_a^k). \tag{2.7}$$

In (2.7) $S^k \equiv A(D^k)^{-1}A^t$ is called the *Schur complement*. A similar reduction can be done for system (2.5).

We like to stress that D^k is a diagonal matrix with positive diagonal entries and it is the only change in the matrices of systems (2.6) and (2.7) at each iteration. Several entries of D^k converge to zero as the method approaches a solution while other entries tend to infinity.

2.3. Approaches for Solving the Linear Systems. Using the Cholesky factorization of S^k to solve for the search directions in interior point methods is by far the most widely used approach (see for example [1, 12, 13, 14]). This approach takes full advantage of S^k being symmetric and positive definite. However, S^k can have much less sparsity and is often more ill-conditioned than the matrix of system (2.6). The most extreme loss of sparsity occurs when A has a dense column. This results in a completely dense Schur complement matrix.

One way around this problem is to use iterative methods. These methods consist of constructing an iterative sequence of approximate solutions for the linear system, until a desirable tolerance is achieved. Since these methods require the matrix only for computing matrix-vector products there is no need to compute the Schur complement unless the preconditioner depends on it. Therefore, loss of sparsity may not be an issue for this approach.

The preconditioned conjugate gradient method is the most popular of the iterative methods for solving positive definite systems. The conjugate gradient method is easy

to implement and will converge rapidly with a good preconditioner. Attempts to solve (2.7) using the preconditioned conjugate gradient method have achieved mixed results [13, 15], mainly because the linear systems become highly ill-conditioned as the interior point method approaches an optimal solution. Another disadvantage of these implementations is that they rely on the computation of the Schur complement to build the preconditioner. Thus, they can suffer from loss of sparsity in the presence of a few dense columns.

For the reasons stated above, several researchers have began to consider the augmented system even though it is indefinite. Consequently, the Cholesky factorization can not be applied since there is no numerically stable way to factor a general indefinite matrix onto LDL^t with D diagonal. However, the reduction to the Schur complement system is embedded in this approach given that the diagonal entries of D are chosen as pivots [20], stability considerations apart. A similar result regarding preconditioners for iterative methods will be shown later.

Implementations using the Bunch-Parlett factorization proved to be more stable but they are slower than solving (2.7) (see [7, 10, 20]). A multifrontal approach applied to the augmented system has been investigated in [5].

The conjugate gradient method is not well defined for indefinite systems. Thus, it is not used for solving (2.6). In [10], SYMMLQ is used to solve the augmented system for a few small problems. We are not aware of any successful implementation of an iterative method for solving the indefinite system arising in interior point methods for large scale linear programming problems. However, this is an active research area and progress along these lines is certainly anticipated. For instance, in [8] it is mentioned that computational results for large scale problems for the preconditioner outlined there will be reported in a forthcoming paper.

All the above discussion refers to solving one of the linear systems. We now discuss what changes when we are solving two systems with the same matrix per iteration.

One important difference is that much work can be saved by computing only one factorization per iteration for the predictor-corrector variant. The reason is that both linear systems share the same matrix thus, the factorization can be used for both. That is the primary reason for popularity of this variant.

It is not completely clear whether the predictor-corrector variant is the most appropriated for the approaches using iterative methods to solve the linear systems. It depends on the expense of computing the preconditioner relative to the work for solving both linear systems and how many interior point iterations the predictor-corrector variant saves.

The linear systems are related since they have the same matrix and the right hand side of the second linear system depends on the solution of the first. With that in mind, it might be possible to obtain better performance when computing the solution of the second linear system. However, we have not found and are not aware of any practical way to accelerate the computation of the second system by using the solution to the first one. This is most likely due to the nonlinearity of the perturbation on the right hand side. Finding a better initial guess for the second linear system will be left as a subject for future research.

2.4. Modeling Linear Programming Problems. Few linear programming problems arise naturally in the standard form (2.1). In practice we have to consider inequality constraints, range constraints, bounded and free variables. It is possible to transform these type of problems into the standard form, but this increases their

dimension. In this section we present the changes in the linear system when considering special structures. These structures will cover all the cited cases except the range constraint which usually is reduced to the bounded variable case by adding one variable for each of these constraints, that is, given the range constraint $\beta_l \leq a^t x \leq \beta_u$, we replace it by $a^t x - \omega = 0$ and $\beta_l \leq \omega \leq \beta_u$.

Applying the primal-dual method on the perturbed optimality conditions of a general problem and eliminating the slack variables the augmented system will be:

$$\begin{pmatrix} -D^k & A^t \\ A & E^k \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix} = \begin{pmatrix} r_1^k \\ r_2^k \end{pmatrix} \tag{2.8}$$

where D^k has zero diagonal entries for the corresponding free variables. E^k is a diagonal matrix with nonzero entries for the corresponding inequality constraint. Bounded variables change the way to compute D^k but they do not change the nonzero structure of the augmented matrix.

For the Schur complement we have

$$S^k \Delta y^k = r_2^k + A(D^k)^\dagger r_1^k$$

where $S^k = A(D^k)^\dagger A^t + E^k$, provided the primal problem has no free variables and $m \leq n$.

Equation (2.8) is the augmented system for problems with inequality constraints, bounded variables and free variables. It contains no slack variables either for inequality constraints or bounds and there is no need to split the free variables into nonnegative variables. It is the smallest system we can get without further information on A or without changing its sparsity pattern.

3. The Augmented System. A slightly more general form for the augmented system (2.6) arises naturally in several areas of applied mathematics such as optimization, fluid dynamics, electrical networks, structural analysis and heat equilibrium. In some of these applications, the matrix D is not necessarily diagonal although it is symmetric. The techniques for solving this system varies widely within these areas since the characteristics of the system changes with the problems. See for example [3, 6, 18, 19, 21] among several others. Still, these systems have many properties in common as we will see next.

3.1. Properties of the Augmented System. The augmented system is nonsingular if its first n columns are linearly independent. This is always the case when D is a diagonal matrix with no zero diagonal entries and A has full row rank. The augmented system is also clearly indefinite. Moreover, it can be seen from the relation

$$\begin{pmatrix} -D & 0 \\ A & I \end{pmatrix} \begin{pmatrix} -D^{-1} & 0 \\ 0 & AD^{-1}A^t + E \end{pmatrix} \begin{pmatrix} -D & A^t \\ 0 & I \end{pmatrix} = \begin{pmatrix} -D & A^t \\ A & E \end{pmatrix}$$

that by Sylvester's law of inertia [11] it has m positive and n negative eigenvalues if D is positive definite and E is positive semi-definite.

It can easily be verified that

$$\begin{pmatrix} -D & A^t \\ A & E \end{pmatrix}^{-1} = \begin{pmatrix} (-D^{-1} + D^{-1}A^t S^{-1} A D^{-1}) & D^{-1}A^t S^{-1} \\ S^{-1}AD^{-1} & S^{-1} \end{pmatrix} \tag{3.1}$$

where $S = AD^{-1}A^t + E$.

3.2. Preconditioning. Iterative methods are very sensitive to the condition number of the matrix $\kappa_p(A) = ||A||_p||A^{-1}||_p$. A matrix with a large condition number is said to be *ill-conditioned*. In most applications it is essential to modify an ill-conditioned linear system into an equivalent better conditioned system. Otherwise, the iteration may be very slow or even fail to converge. This is done in such a way that it is easy to recover the original system solution from the modified one. The technique just described is known as preconditioning.

Consider the following situation: given $Ax = b$, we solve the equivalent linear system $M^{-1}AN^{-1}\tilde{x} = \tilde{b}$, where $\tilde{x} = Nx$ and $\tilde{b} = M^{-1}b$. The system is said to be preconditioned and $M^{-1}AN^{-1}$ is called the preconditioned matrix.

We say that a preconditioner is symmetric if $N^t = M$ because in that case if A is symmetric, the preconditioned matrix $M^{-1}AM^{-t}$ will also be symmetric. In this situation, the preconditioned matrix has the same number of positive (or negative) eigenvalues as the original matrix for symmetric preconditioners. Thus, symmetric preconditioners cannot be used to change an indefinite system into a positive definite one.

There are several classes of preconditioning methods. In this work we are mainly concerned with the class of preconditioners called incomplete factorization methods. This approach has been used successfully to develop preconditioners for problems similar to ours. Usually these preconditioners have been applied to the Schur complement.

The idea of incomplete factorization is to ignore the fill-in entries during the course of the factorization that would create nonzero entries outside of a predetermined pattern. Often, the pre-specified pattern is the original non-zero structure of A. As the factorization proceeds, the result of operations that would cause unwanted fill are simply set to zero. In PDE problems there are physical explanations to justify the use of these type of preconditioners. Understanding of the underlying physical motivation has produced very effective preconditioners. The success of this approach has motivated the use of incomplete factorization on other problems where there is not much information to guide the choice of preconditioners. For symmetric positive definite matrices we compute say $\hat{L}$ and let $M = N^t = \hat{L}$ thus obtaining a (preconditioned) symmetric positive definite matrix which should be closer to the identity than the original matrix A. This preconditioner is called the incomplete Cholesky factorization.

The incomplete Cholesky factorization is a natural choice to apply as a preconditioner for the Schur complement. A slight modification of this approach was investigated in [15]. Here, we are concerned with the incomplete Cholesky factorization version that ignores all fill-ins that may occur on L.

3.3. Existing Preconditioners. Consider the following class of symmetric preconditioners*:

$$\begin{pmatrix} H & 0 \\ F & G \end{pmatrix} \begin{pmatrix} -D & A^t \\ A & E \end{pmatrix} \begin{pmatrix} H^t & F^t \\ 0 & G^t \end{pmatrix} = \begin{pmatrix} -HDH^t & B^t \\ B & C \end{pmatrix} \tag{3.2}$$

where, $B = -FDH^t + GAH^t$ and $C = -FDF^t + FA^tG^t + GAF^t + GEG^t$. We will see next that some of the preconditioners proposed to the augmented system belong to this class.

*From now on we drop the superscript k since we are concerned with one iteration of the primal-dual interior point method.

Preconditioner	H_N	H_B	F_N	F_B	G
M1	$-D_N^{-\frac{1}{2}}$	I	0	0	I
M2	$-D_N^{-\frac{1}{2}}$	B^{-t}	0	0	I
M3	$-D_N^{-\frac{1}{2}}$	B^{-t}	0	$-\frac{1}{2}B^{-t}H_B B^{-1}$	I
M4	$-D_N^{-\frac{1}{2}}$	L^{-1}	0	$-\frac{1}{2}L^{-1}H_B L^{-t}$	U^t
P1	$-D_N^{-\frac{1}{2}}$	$-D_B^{-\frac{1}{2}}$	0	0	$-H_B B^{-1}$
P2	$-D_N^{-\frac{1}{2}}$	$-D_B^{-\frac{1}{2}}$	$-H_B B^{-1} N D_N$	$-H_B$	$H_B^{-1}B^{-1}$

TABLE 3.1
Existing Preconditioners, $B^t = LU$

Gill et al. [10] introduce some preconditioners for the standard form which try to avoid the numerical problems caused by the ill-conditioned behavior of D as the primal-dual method progresses. The first preconditioner they proposed was:

$$\hat{M} = \begin{pmatrix} \hat{H} & 0 \\ 0 & I \end{pmatrix}$$

where $\hat{H}$ is a diagonal $n \times n$ matrix whose nonzero entries are:

$$\hat{\eta}_{ii} = \begin{cases} \delta_{ii}^{-\frac{1}{2}} & \text{for the } n-m \text{ biggest entries of } D \\ 1 & \text{otherwise.} \end{cases}$$

Notice that this is equivalent to choosing $H \equiv \hat{H}$, $F \equiv 0$ and $G \equiv I$ in (3.2).

If we adopt the partition $A = [B\ N]P$ where N corresponds to the $n-m$ variables closest to a bound, and use the notation H_B and H_N to denote the corresponding submatrices for a matrix H, their preconditioners correspond to the choices $M1$ to $M4$ given in table 3.1.

In the same work, they also propose a preconditioner based upon the Bunch-Parlett factorization $\tilde{L}\tilde{D}\tilde{L}^t$ [4] for an approximation of the matrix in the system (2.6). By constructing a diagonal matrix $\hat{D}$ whose eigenvalues are the absolute value of those in $\tilde{D}$ (recall that $\tilde{D}$ is a block diagonal matrix whose blocks have dimension 1 or 2), they obtain a positive definite matrix and therefore a symmetric preconditioner.

In [2] Battermann and Heinkenschloss introduce and analyze another class of preconditioners for the quadratic programming problem. Table 3.1 also gives the choice for the block matrices for the first two preconditioners $P1$ and $P2$ presented by them. The third preconditioner introduced has a different partitioning:

$$P3 = \begin{pmatrix} I & 0 & D_B B^{-1} \\ 0 & 0 & B^{-1} \\ -(B^{-1}N)^t & I & (B^{-1}N)^t D_B B^{-1} \end{pmatrix}.$$

Notice that this is the representation for these preconditioners for linear programming problems. See [2] for a complete description.

The preconditioners in [10] presented on table 3.1 break down for degenerate problems, that is, problems which have an optimal solution where more than m primal variables x are not at their bounds. A consequence of degeneracy is that the number of diagonal entries on D that will converge to zero is unknown. Since these preconditioners assume it occurs for a fixed number $(n-m)$ their performance is sensibly affected by degenerate problems.

3.4. Unsymmetric Preconditioners. Consider the inverse of the augmented system with implicit slack variables:

$$(3.3)\qquad \begin{pmatrix} -D & A^t \\ A & E \end{pmatrix}^{-1} = \begin{pmatrix} (-D^{-1} + D^{-1}A^tS^{-1}AD^{-1}) & D^{-1}A^tS^{-1} \\ S^{-1}AD^{-1} & S^{-1} \end{pmatrix}$$

where $S = AD^{-1}A^t + E$.

From relation (3.3) we can derive an unsymmetric preconditioner by replacing S by an incomplete Cholesky factorization $\hat{L}\hat{L}^t$. We call it unsymmetric because the preconditioned matrix may be unsymmetric although both the original matrix and the preconditioner are symmetric. This preconditioner can be very effective but its matrix-vector product is more expensive than the others since it requires the solution of two linear systems involving $\hat{S}$.

Freund and Jarre [8] present another unsymmetric preconditioner based upon the SSOR where the matrix D is now block diagonal with 1×1 and 2×2 blocks about the diagonal. In order to achieve this, it is necessary to permute the augmented matrix since its last diagonal block consists of all zeros for problems in the standard form. Their preconditioner has the following form $N = I$ and $M = U^tD^{-1}U$ where, $U = (L^t + D)P^t$ and P is a permutation for the augmented matrix.

The block diagonal matrix D in this preconditioner has m (2×2) blocks and $m - n$ (1×1) blocks since the augmented system has m zero diagonal entries. In order to M be a good approximation for A, $||LD^{-1}L^t||$ should be as small as possible. With this goal in mind they propose to choose the permutation by minimizing the product of the largest eigenvalues of the blocks on D^{-1} over all possible choices of 2×2 blocks. This approach gives different permutations at each iteration. The problem of finding the permutation is solved as a weighted bipartite matching problem.

3.5. Why Work with the Augmented System?. The reduction to the Schur complement has the inherent disadvantage of changing the sparse structure of the matrix which can be disastrous for some problems. Even if iterative methods are being applied, the computation of the Schur complement is necessary for obtaining a preconditioner in all implementations we are aware of. The computational time required to construct and apply it may not compensate the reduction of the problem from indefinite to a smaller positive definite system.

The results of this section are very important because they give theoretical support for working with the augmented system. They indicate that considerable important information is lost by reducing to the Schur complement. On the other hand, by studying the augmented system, no information is lost. Moreover, it is possible to return to the Schur complement system when it is advantageous to do so since it is in some sense contained in augmented system.

The following lemma gives a justification for working with the augmented system instead of the Schur complement when applying iterative methods. It will be shown that every preconditioner for the Schur complement can be replicated on the augmented system while the converse statement is not true. Therefore, the preconditioned Schur complement system can be seen as a particular case contained on the more general class of preconditioners for the augmented system.

LEMMA 3.1. *Let*

$$\begin{pmatrix} -D & A^t \\ A & E \end{pmatrix}$$

be nonsingular and D be symmetric positive definite. Then there exists a preconditioner pair M and N such that this matrix can be reduced to:

$$M^{-1}\begin{pmatrix} -D & A^t \\ A & E \end{pmatrix} N^{-1} = \begin{pmatrix} I & 0 \\ 0 & GSH^t \end{pmatrix}$$

where S is the Schur complement and G and H are arbitrarily chosen nonsingular matrices of appropriated dimension.

Proof. Let $D = LL^t$ and consider the preconditioner pair

$$M^{-1} = \begin{pmatrix} -L^{-1} & 0 \\ GAD^{-1} & G \end{pmatrix}$$

and

$$N^{-1} = \begin{pmatrix} L^{-t} & D^{-1}A^tH^t \\ 0 & H^t \end{pmatrix}.$$

Then,

$$M^{-1}\begin{pmatrix} -D & A^t \\ A & E \end{pmatrix} N^{-1} = \begin{pmatrix} I & 0 \\ 0 & GSH^t \end{pmatrix} \tag{3.4}$$

where $S = AD^{-1}A^t + E$. $\square$

Observe that this result even holds for singular matrices G and H. However, it does not make sense to obtain a singular matrix in the context of preconditioning.

The next lemma shows that for the augmented system (2.8). the converse statement is not true.

LEMMA 3.2. *Consider the augmented system given by*

$$\begin{pmatrix} -D & A^t \\ A & E \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

and its Schur complement $S = AD^{-1}A^t + E$ where D is nonsingular and A has full row rank. Then any symmetric block triangular preconditioner

$$\begin{pmatrix} H & 0 \\ F & G \end{pmatrix}$$

leads to a preconditioned system for GSG^t independent of H and F.

Proof. Consider again the class of symmetric preconditioners given by:

$$\begin{pmatrix} H & 0 \\ F & G \end{pmatrix}\begin{pmatrix} -D & A^t \\ A & E \end{pmatrix}\begin{pmatrix} H^t & F^t \\ 0 & G^t \end{pmatrix} = \begin{pmatrix} -HDH^t & B^t \\ B & C \end{pmatrix}$$

where, $B = -FDH^t + GAH^t$ and $C = -FDF^t + FA^tG^t + GAF^t + GEG^t$. Then, the preconditioned system will be as follows:

$$\begin{pmatrix} -HDH^t & B^t \\ B & C \end{pmatrix}\begin{pmatrix} \Delta\tilde{x} \\ \Delta\tilde{y} \end{pmatrix} = \begin{pmatrix} Hr_1 \\ Gr_2 + Fr_1 \end{pmatrix}$$

now, by eliminating $\Delta\tilde{x}$ we get:

$$GSG^t\Delta\tilde{y} = Gr_2 + AD^{-1}r_1 \tag{3.5}$$

this system does not depend on either H or F, thus any choice for these matrices that preserves nonsingularity are valid preconditioners which lead to (3.5). $\square$

Notice that these results can be applied in a more general context. For instance D can be any symmetric positive definite matrix not necessarily diagonal. Systems with these characteristics occur in other fields including nonlinear programming and PDE problems. Moreover, there is no restriction to E whatsoever. However, if E is positive semi-definite as in the augmented system, the Schur complement will be positive definite.

4. A New Class of Preconditioners. In this section we will construct and study a class of symmetric preconditioners for the augmented system which exploits its structure. In view of the discussion on previous sections the preconditioners are designed to avoid forming the Schur complement. Our goal is to obtain a preconditioner that is relatively cheap to compute and retains excellent theoretical and practical properties.

4.1. Building a Preconditioner. Since the augmented system is naturally partitioned into block form, lets start with the most generic possible block symmetric preconditioner for the augmented system

$$M^{-1} = N^{-t} = \begin{pmatrix} F & G \\ H & J \end{pmatrix}$$

and choose the matrix blocks step by step according to our goals. We only consider symmetric preconditioners. We believe there is insufficient motivation for loosing symmetry.

The preconditioned augmented matrix (2.6) will be the following *:

$$\begin{pmatrix} -FDF^t + FA^tG^t + GAF^t & -FDH^t + FA^tJ^t + GAH^t \\ -HDF^t + HA^tG^t + JAF^t & -HDH^t + HA^tJ^t + JAH^t \end{pmatrix}. \tag{4.1}$$

At this point we begin making decisions about the blocks. We start by observing that the right lower block is critical in the sense that the Schur complement $AD^{-1}A^t$ appears in the expression for many reasonable choices of J and H. The Schur complement may be avoided more easily if we set $J = 0$. This choice seems to be rather drastic at first glance leaving few options for the selection of the other blocks. But, as we will soon see, the careful selection of the remaining blocks will lead to promising situations. With $J = 0$ the preconditioned augmented system reduces to

$$\begin{pmatrix} -FDF^t + FA^tG^t + GAF^t & -FDH^t + GAH^t \\ -HDF^t + HA^tG^t & -HDH^t \end{pmatrix}.$$

Lets turn our attention to the off diagonal blocks. If we can make them zero blocks, the preconditioned matrix appears to be closer to the identity matrix and the problem decouples into two smaller linear systems. Thus, one idea is to write $F^t = D^{-1}A^tG^t$. However, it is easy to verify that $\mathcal{N}(F^t) \supset \mathcal{N}(G^t)$ therefore, this choice is not acceptable. A more reasonable choice is $G^t = (HA^t)^{-1}HDF^t$ giving

$$\begin{pmatrix} -FDF^t + FA^tG^t + GAF^t & 0 \\ 0 & -HDH^t \end{pmatrix}.$$

*We will be restricted for now to the problem without inequality constraints and free variables to facilitate the discussion.

Now lets decide how to chose H. The choices A, AD^{-1} or variations of it will not be considered since matrices with the nonzero pattern of the Schur complement will appear in the right lower block and also as part of G. On the other hand, setting $H = [I\ 0]P$ where P is a permutation matrix such that HA^t is nonsingular does not introduce a Schur complement type matrix. The right lower block reduces to a diagonal matrix $-D_B \equiv -HDH^t$ where,

$$PDP^t = \begin{pmatrix} D_B & 0 \\ 0 & D_N \end{pmatrix} \tag{4.2}$$

and we achieve one of the main goals, namely avoiding the Schur complement.

Therefore, we can concentrate on F at the upper left block. The choice $F = D^{-\frac{1}{2}}$ seems to be natural and as we shall see later leads to some interesting theoretical properties for the preconditioned matrix. Summarizing, the final preconditioned matrix takes the form

$$M^{-1}\begin{pmatrix} -D & A^t \\ A & 0 \end{pmatrix} M^{-t} = \begin{pmatrix} -I + D^{-\frac{1}{2}}A^tG^t + GAD^{-\frac{1}{2}} & 0 \\ 0 & -D_B \end{pmatrix} \tag{4.3}$$

where,

$$M^{-1} = \begin{pmatrix} D^{-\frac{1}{2}} & G \\ H & 0 \end{pmatrix}$$

with $G = H^tD_B^{\frac{1}{2}}B^{-1}$, $HP^t = [I\ 0]$ and $AP^t = [B\ N]$. We remark that this notation for the partition of A borrowed from the literature of the simplex method does not mean that the set of columns B form a particular basis for the linear programming problem. The only concern up to this point is that these columns form a nonsingular matrix.

We observe that the goal of avoiding the Schur complement was achieved. The price paid is that now we have to find B and solve linear systems with this matrix. However, the factorization $QB = LU$ is typically easier to compute than the Cholesky factorization. The reason is that the selection of the columns of B causes no change on the structure of the problem, in contrast to the computation of AA^t. In fact, it is known [9] that the sparsity pattern of L^t and U is contained in the sparsity pattern of R, where $AA^t = R^tR$, for any valid permutation Q. Typically in practice, the number of nonzero entries of R is much larger than the combined number of nonzero entries in L and U. Moreover, the LU factorization of B does not depend on D which is the only change in the matrix of the augmented system from one iteration to another. Thus, if we decide to keep the permutation matrix P unchanged, the computation of the next preconditioned matrix is almost free. It only requires computing n square roots.

Another advantage of this preconditioner compared with other augmented system preconditioners is that it reduces the size of the linear system from $n + m$ to n since the lower block equation is easily solved for the diagonal matrix. It still gives a larger system than the Schur complement reduction, but as we shall see, this disadvantage can also be overcome.

In the next section we will study some of the theoretical properties of the upper left block preconditioned matrix (4.3) and later show some strategies on how to choose the permutation matrix, that is, the columns of A that we select to form B. But first we will make a few observations about the preconditioned matrix.

Recall that the augmented matrix (2.6) has m positive and n negative eigenvalues. Thus, the preconditioned matrix (4.3) have the same inertia of eigenvalues. Since the lower right block matrix has m negative eigenvalues, the upper left block has m positive and $n-m$ negative eigenvalues. Therefore the preconditioned matrix is indefinite except for the odd case where $m=n$.

We close this section by showing a property of the matrices in the upper left block of (4.3) which leads to the results of the next section.

LEMMA 4.1. *Let* $A=[B\ N]$ *with* B *nonsingular and* $G=H^tD_B^{\frac{1}{2}}B^{-1}$ *where* $H=[I\ 0]$. *Then* $G^tD^{-\frac{1}{2}}A^t=AD^{-\frac{1}{2}}G=I$.

Proof. It is sufficient to show that $G^tD^{-\frac{1}{2}}A^t=I$.

$$\begin{aligned} B^{-t}D_B^{\frac{1}{2}}[I\ 0]D^{-\frac{1}{2}}[B\ N]^t &= \\ B^{-t}D_B^{\frac{1}{2}}[D_B^{-\frac{1}{2}}\ 0][B\ N]^t &= \\ B^{-t}[I\ 0][B\ N]^t &= I \qquad \square \end{aligned}$$

This result can be easily extended for any permutation $A=[BN]P$ with B nonsingular.

4.2. Theoretical Properties of the Preconditioned System. In this section we will study some properties of matrices of the type

$$\begin{array}{rcccc} K=-I_n & + & U^tV^t & + & VU \\ \text{where } UV & = & V^tU^t & = & I_m \end{array} \tag{4.4}$$

and U, V^t are $m\times n$ matrices. By lemma 4.1 it easy to verify that the upper left block (4.3) belongs to this class of matrices. First we show that the eigenvalues of K are bounded away from zero.

THEOREM 4.2. *Let* λ *be an eigenvalue of* K *given by (4.4) where* U *and* $V^t\in R^{m\times n}$ *then* $|\lambda|\geq 1$.

Proof. Let v be a normalized eigenvector of K associated with λ then,

$$\begin{aligned} Kv &= \lambda v \\ K^2v &= \lambda^2v \\ (I-U^tV^t-VU+U^tV^tVU+VUU^tV^t)v &= \lambda^2v \\ v+(U^tV^t-VU)(VU-U^tV^t)v &= \lambda^2v. \end{aligned}$$

Multiplication on the left by v^t gives

$$\begin{aligned} 1+v^t(U^tV^t-VU)(VU-U^tV^t)v &= \lambda^2 \\ 1+w^tw &= \lambda^2 \end{aligned}$$

where, $w=(VU-U^tV^t)v$. Thus, we obtain $\lambda^2\geq 1$. $\square$

COROLLARY 4.3. *The preconditioned matrix (4.3) is nonsingular.*

Thus, by theorem 4.2 K is not only nonsingular but it has a norm of order 1 and has no eigenvalues in the neighborhood of zero. These are indeed desirable properties for getting good performance with iterative methods for solving linear systems. The following remark is useful for showing other important results.

REMARK 4.1. *Since* U *and* $V^t\in R^{m\times n}$ *are such that* $UV=I_m$, VU *is an oblique projection onto* Range(V). *Thus, if* $x\in$ Range(V), *then* $VUx=x$.

THEOREM 4.4. *The matrix K in (4.4) where U and $V^t \in R^{m\times n}$ has at least one eigenvalue λ such that $|\lambda| = 1$.*

Proof. Observe that if $K + I = U^tV^t + VU$ is singular, then K has at least an eigenvalue $\lambda = -1$. Lets consider three cases:

(i) If $n > 2m$ then $K + I$ is singular, since for any square matrices A and B $rank(A + B) \leq rank(A) + rank(B)$.

(ii) If $n < 2m$, observe that $dim(\text{Range}\,(U^t) \cup \text{Range}\,(V)) \leq n$. Also, U^t and V have rank m since $UV = I_m$. Thus, $dim(\text{Range}(U^t) \cap \text{Range}(V)) > 0$ since $n < 2m$. Hence, there is at least one eigenvector $v \neq 0$ such that $v \in \text{Range}(U^t) \cap \text{Range}(V)$ therefore, by remark 4.1, $Kv = (-I + U^tV^t + VU)v = -v + v + v = v$.

(iii) If $n = 2m$, then from (ii) if $\text{Range}(U^t) \cap \text{Range}(V) \neq 0$ there is a eigenvalue $\lambda = 1$. Otherwise there exist an eigenvector v such that $v_{\text{Range}\,(V)} = 0$ or $v_{\text{Range}\,(U^t)} = 0$. Without loss of generality consider that $v_{\text{Range}\,(V)} = 0$, i.e, $v \in \mathcal{N}(V^t)$ and there is an eigenpair $(\theta = \lambda + 1, v)$ where

$$VUv = \theta v.$$

But then $\lambda + 1 = (0 \text{ or } 1)$ since $UV = I$ and from theorem 4.2 it must be $\lambda = -1$. ◻

COROLLARY 4.5. *The condition number $\kappa_2(K)$ in (4.4) is given by* $\max|\lambda(K)|$.

Proof. The proof is immediate from theorems 4.2, 4.4, the definition of $\kappa_2(K) = \frac{\sigma_{max}}{\sigma_{min}}$ and recalling that K is symmetric. ◻

4.3. Practical Aspects. The iterative methods need access to the matrix only to compute matrix-vector products. In this section we present a more stable way to compute this product versus using the matrix directly as in $w = Kx$. Lets consider for simplicity our matrix to be of the form (4.4). We can write any n-dimensional vector x as $x = x_{\text{Range}\,(V)} + x_{\mathcal{N}(V^t)}$, where $x_{\text{Range}\,(V)} \in \text{Range}\,(V)$ and $x_{\mathcal{N}(V^t)} \in \mathcal{N}(V^t)$. Thus,

$$\begin{aligned} Kx &= -x + U^tV^tx_{\text{Range}\,(V)} + VUx \\ &= -x_{\mathcal{N}(V^t)} + U^tV^tx_{\text{Range}\,(V)} + VUx_{\mathcal{N}(V^t)} \end{aligned}$$

by remark 4.1. Now, since $V^t = [I\ 0]P$ its null and range spaces can be easily represented in a code and all the calculations for it consist in managing some indices properly. Observe that the first two terms do not have nonzero entries in common for any of the positions. Hence, no floating point operations are needed to add them. If we compute the product Kx without these considerations, some round-off error will be introduced for the zero sum $-x_{\text{Range}\,(V)} + UVx_{\text{Range}\,(V)}$ and often this error is large enough compared with the other entries of x. A welcomed side effect is that n floating point operations are saved with this strategy.

Another practical aspect concern the recovering of the solution. The approximate solution for the original system can be easily recovered from the solution for the preconditioned system $(\hat{x}, \hat{y})$ by computing

$$\begin{pmatrix} x \\ y \end{pmatrix} = M^{-t} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}$$

where the error of the solution is given by

$$\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} b_1 + Dx - A^ty \\ b_2 - Ax - Ey \end{pmatrix}.$$

Sometimes the norm of the error $||r||^2 = ||r_1||^2 + ||r_2||^2$ is too large due to the round-off error introduced on computing the preconditioned system and recovering the solution. It can get particularly large at the final iterations of the interior point method. One way to reduce this error is to compute $\tilde{x} = D^{-1}(A^t y - b_1)$ and form a new approximate solution $(\tilde{x}, y)$. The error for the new solution $\tilde{r}$ will be given by $\tilde{r_2} = r_2 + AD^{-1}r_1$ if we assume that $\tilde{r_1} = b_1 + D\tilde{x} - A^t y$ is zero. Thus, we update the solution whenever $||r||$ is above a given tolerance and $||\tilde{r}|| < ||r||$. This update is necessary for the successful convergence of the interior point method for some of the problems tested with this preconditioner. Notice that this strategy can be used for any symmetric preconditioner for the augmented system.

4.4. Reduction to Positive Definite Systems. Consider again the indefinite linear system at the left upper block

$$K = -I + D^{-\frac{1}{2}} A^t G^t + GAD^{-\frac{1}{2}}$$

where $G = H^t D_B^{\frac{1}{2}} B^{-1}$ and $HP^t = [I\ 0]$. It is possible to exploit the structure of the problem even further, reducing it to a smaller positive definite system. If we expand the above equation we obtain the following matrix

$$P^t \begin{pmatrix} I & D_B^{\frac{1}{2}} B^{-1} N D_N^{-\frac{1}{2}} \\ D_N^{-\frac{1}{2}} N^t B^{-t} D_B^{\frac{1}{2}} & -I \end{pmatrix} P$$

where we used relation 4.2.

Therefore the problem can be reduced to solve a positive definite linear system involving either matrix

$$I_m + D_B^{\frac{1}{2}} B^{-1} N D_N^{-1} N^t B^{-t} D_B^{\frac{1}{2}} \tag{4.5}$$

or

$$I_{n-m} + D_N^{-\frac{1}{2}} N^t B^{-t} D_B B^{-1} N D_N^{-\frac{1}{2}}. \tag{4.6}$$

These matrices have some interesting theoretical properties related to the indefinite matrix K.

Lets first define $W = D_N^{-\frac{1}{2}} N^t B^{-t} D_B^{\frac{1}{2}}$. Hence, the positive definite matrices can be written $I + W^t W$ and $I + WW^t$ respectively while the indefinite matrix not considering permutations reduces to

$$\begin{pmatrix} I & W^t \\ W & -I \end{pmatrix}.$$

THEOREM 4.6. *The matrices in (4.5) and (4.6) are positive definite and their eigenvalues are greater or equal to one.*

Proof. Let v be a normalized eigenvector of $I + WW^t$ and θ its associated eigenvalue, then

$$\begin{aligned} (I + WW^t)v &= \theta v \\ v^t(v + WW^t v) &= \theta v^t v \\ 1 + u^t u &= \theta \end{aligned}$$

where $u = W^t v$. Thus $\theta \geq 1$. The proof for $I + W^t W$ is similar. $\square$

Remark 4.2. *The matrices in (4.5) and (4.6) have the same set of eigenvalues with the exception of the extra eigenvalue equal to one for the matrix of higher dimension.*

From the above theorem and theorem 10.2.4 in [11] we can conclude that the conjugate gradient method converges in at most $\min(m, n-m)$ iterations in exact arithmetic for both positive definite matrices. Thus, we do not expect very different behavior between the two linear systems although they have different dimension. Besides the work for computing the matrix-vector product is about the same for all three systems.

The next theorem is also important since it relates the eigenpairs of the indefinite matrix with the eigenpairs of the positive definite matrices.

Theorem 4.7. *Consider the eigenvalue problem*

$$\begin{pmatrix} I & W^t \\ W & -I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \end{pmatrix}$$

then (θ, u) is an eigenpair of $I + W^tW$ and (θ, v) is an eigenpair of $I + WW^t$, where $\theta = \lambda^2$.

Proof. Notice that

$$\begin{pmatrix} I & W^t \\ W & -I \end{pmatrix} \begin{pmatrix} I & W^t \\ W & -I \end{pmatrix} = \begin{pmatrix} I + W^tW & 0 \\ 0 & I + WW^t \end{pmatrix}$$

thus,

$$\begin{pmatrix} I + W^tW & 0 \\ 0 & I + WW^t \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda^2 \begin{pmatrix} u \\ v \end{pmatrix}. \qquad \square$$

Therefore the indefinite system can still be an option for solving the linear system since it has a better eigenvalue distribution than the positive definite systems.

4.5. Choosing the Set of Columns. In this section we discuss how to select the columns of A that form B. The type of matrices in (4.5) and (4.6) suggest a choice for the columns of B by looking at the values of D. If we can chose the columns related to the smallest values of $diag(D)$, both WW^t and W^tW approach zero at the final iterations of the interior point method. Thus, a good strategy consists in taking the first m linearly independent columns of A ordered by the value of δ_{ii} in non decreasing order. This choice of columns tends to produce better conditioned matrices as the interior point method approaches a solution. This is because at least $n - m$ diagonal entries entries of D became large and diminish the importance from WW^t and W^tW.

During the first few iterations, the diagonal values of D are roughly the same. In fact they are all equal to one for the computation of the starting point. Hence, for these linear systems the strategy described before is ineffective. Another way to obtain a good set of columns is to minimize $||W||$. This is problem is hard to solve but it can be approached by a cheap heuristic, that is we choose the first m linearly independent columns of AD^{-1} with smallest 1-norm. This approach is adopted for the computation of the starting point. After that the ordering of columns by the diagonal values of D is used.

Contrary to some of the preconditioners given in [10], degenerated problems do not seem to play an important role in this preconditioner. This is a nice property since the selection of m columns of A naturally leads one to think in terms of basis for the linear programming problem.

IP Iteration	Inner iterations: Incomplete Cholesky	Inner iterations: New LU Preconditioner
0	49	195
1	49	203
2	45	258
3	39	190
4	24	171
5	24	185
6	20	128
7	22	130
8	22	133
9	32	126
10	44	108
11	71	91
12	104	92
13	171	76
14	323	63
15	480	52
16	834	43
17	1433	34
18	2146	30
19	4070	22
20	7274	18
21	11739	17
22	15658	15
23	24102	12
24	13463	10
25	5126	6
Average	3360	84

TABLE 4.1
KEN13 Conjugate Gradient Method Iterations

4.6. Preliminary Experiments. Table 4.1 shows the results for the conjugate gradient method with respect to the number of iterations for the incomplete Cholesky factorization and the new preconditioner as the interior point method progresses. The preconditioned matrix (4.5) was used. The experiments are carried out in C, on a SUN ULTRA-SPARC station. The floating point arithmetic is IEEE standard double precision. The stopping tolerance for the interior point method and preconditioned conjugate gradient is the square root of machine epsilon.

The chosen problem is KEN13 a multi-commodity network flow problem. It can be obtained from netlib. The dimension of the linear system is 14627. The iteration number zero corresponds to the linear system for computing the starting point for the interior point method. Only the number of iterations of the conjugate gradient method for solving the first linear system of the interior point iterations are shown. The number of iterations for solving the second linear system is very close to the number for the first system. An interesting property of these approaches is that the incomplete Cholesky preconditioners in general starts taking few iterations to get convergence and

its behavior deteriorates as the interior point method converges to a solution. With the LU based preconditioner the exact opposite occurs. The last few iterations are the ones where it performs better. This property of the LU preconditioner is highly desirable since the last linear systems are the most ill-conditioned.

The results obtained for problem KEN13 can be considered as typical for both preconditioners. That leads naturally to the idea of a hybrid approach. We start with the incomplete Cholesky preconditioner and change to the LU preconditioner at a certain point. A crucial issue consists in finding a suitable way to decide when to switch the preconditioner from the incomplete Cholesky factorization to the new LU factorization approach. The number of of iterations for the conjugate gradient method to achieve convergence would be a good indicator to determine when is advisable to switch. It could be when it takes more iterations than a given parameter or when a sudden increase on this number occur.

5. Conclusion. We have shown that from the point of view of designing preconditioners it is better to work with the augmented system instead of working with the Schur complement. Two important results support this statement. First, all preconditioners developed for the Schur complement system have in a sense an equivalent for the augmented system. However, the opposite statement is not true. Whole classes of preconditioners for the augmented system can lead to the an unique preconditioner for the Schur complement.

Given this result we decided to design a preconditioner for the augmented system. Our primary goal on developing this preconditioner was to avoid the Schur complement. The main reason to avoid it is the loss of sparsity that can happens by computing it or its factorization. The price for avoiding the Schur complement is that we now have to compute an LU factorization for a set of linearly independent columns of A. We expect that this factorization will be cheaper to compute than computing the Schur complement and its (in)complete Cholesky factorization on many situations found in practice. Moreover, the LU factorization can be used effectively for a number of iterations thus, computing the next few preconditioners is much more cheaper. A feature that is not shared by any known successful approach either to the Schur complement or the augmented system.

A more important advantage of these preconditioners is that they became better in some sense as the interior point method advances towards an optimal solution. This is a very welcome characteristic since these problems are known to be very ill-conditioned close to a solution. Finally, this preconditioner reduces the system to positive definite matrices, and therefore the conjugate gradient method can be applied, making it easier to compete against the Schur complement approach. This reduction on the dimension of the system to be solved to either m or $n - m$ is an advantage over others augmented approaches since it minimizes the overhead of the vector operations on the iterative linear solver method.

For future implementations we would like to make experiments with the hybrid approach, that is we start with an incomplete Cholesky factorization and switch to the LU preconditioner at a certain point. If a suitable way to decide when to switch can be found, this approach will be probably the fastest way for solving many problems. We also are interested on finding variations of the preconditioner by choosing its blocks in a different fashion.

Techniques for computing the new preconditioner efficiently and extensive numerical experiments are reported in a follow up work [17].

REFERENCES

[1] I. Adler, N. K. Karmarkar, M. G. C. Resende, and G. Veiga, *An Implementation of Karmarkar's Algorithm for Linear Programming*, Math. Programming, 44 (1989), pp. 297–335.

[2] A. Battermann and M. Heinkenschloss, *Preconditioners for Karush-Kunh-Tucker Matrices Arising in the Optimal Control of Distributed System*, Technical Report TR96-34, Department of Computational and Applied Mathematics, Rice University, 1996.

[3] D. Braess and P. Peisker, *On the Numerical Solution of the Biharmonic Equation and the Role of Squaring Matrices*, IMA J. Numer. Anal., 6 (1986), pp. 393–404.

[4] J. R. Bunch and B. N. Parlett, *Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

[5] I. S. Duff, *The Solution of Large-Scale Least-Square Problems on Supercomputers*, Annals Oper. Res., 22 (1990), pp. 241–252.

[6] H. C. Elman and G. H. Golub, *Inexact and Preconditioned Uzawa Algorithms for Saddlepoint Problems*, Technical Report CS-TR-3075, Computer Science Department University of Maryland, 1993.

[7] R. Fourer and S. Mehrotra, *Performance on an Augmented System Approach for Solving Least-Squares Problems in an Interior-Point Method for Linear Programming*, Mathematical Programming Society COAL Newsletter, 19 (1992), pp. 26–31.

[8] R. W. Freund and F. Jarre, *A QMR-Based Interior-Point Algorithm for Solving Linear Programs*, AT&T Numerical Analysis Manuscript No. 94-19, Bell Laboratories, Murray Hill, New Jersey, 1995.

[9] A. George and E. Ng, *An Implementation of Gaussian Elimination with Partial Pivoting for Sparse Systems*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 390–409.

[10] P. E. Gill, W. Murray, D. B. Ponceleón and M. A. Saunders, *Preconditioners for Indefinite Systems Arising in Optimization*, SIAM J. Matrix Anal. and Applications, 13 (1992), pp. 292–311.

[11] G. H. Golub and F. Von Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[12] I. J. Lustig, R. E. Marsten and D. F. Shanno, *Computational Experience with a Primal-Dual Interior-Point Method for Linear Programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[13] ———, *On Implementing Mehrotra's Predictor-Corrector Interior Point Method for Linear Programming*, SIAM J. Optimization, 2 (1992), pp. 435–449.

[14] K. A. McShane, C. L. Monma and D. F. Shanno, *An Implementation of a Primal-Dual Interior-Point Method for Linear Programming*, ORSA J. on Comput., 1 (1989), pp. 70–83.

[15] M. 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] ———, *On the Implementation of a Primal-Dual Interior Point Method*, SIAM J. Optimization, 2 (1992), pp. 575–601.

[17] A. R. L. Oliveira and D. C. Sorensen, *Computational Experience with a Preconditioner for Interior Point Methods for Linear Programming*, Technical Report TR97-28, Department of Computational and Applied Mathematics, Rice University, 1997.

[18] T. Rusten and A. J. Winther *A Preconditioned Iterative Method for Saddlepoint Problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.

[19] D. Sylvester and A. Wathen, *Fast Iterative Solution of Stabilized Stokes Systems part II: Using General Block Preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.

[20] R. J. Vanderbei and T. J. Carpenter, *Indefinite Systems for Interior Point Methods*, Math. Programming, 58 (1993), pp. 1–32.

[21] S. A. Vavasis, *Stable Numerical Algorithms for Equilibrium Systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1108–1131.