Preconditioning indefinite systems in interior point methods for large scale linear optimisation

Al-Jeiroudi, G, Gondzio, J & Hall, J 2008, 'Preconditioning indefinite systems in interior point methods for large scale linear optimization' Optimization Methods and Software, vol 23, no. 3, pp. 345-363.

Preconditioning Indefinite Systems in Interior Point Methods for Large Scale Linear Optimization

Ghussoun Al-Jeiroudi Jacek Gondzio Julian Hall

June 12, 2006, revised February 23, 2007 and June 1, 2007

MS-2006-003

For other papers in this series see http://www.maths.ed.ac.uk/preprints

Preconditioning Indefinite Systems in Interior Point Methods for Large Scale Linear Optimization

Ghussoun Al-Jeiroudi* Jacek Gondzio† Julian Hall‡

School of Mathematics
The University of Edinburgh
Mayfield Road, Edinburgh EH9 3JZ
United Kingdom.

Technical Report MS-2006-003

June 12th, 2006, revised February 23rd, 2007

*Email: G.Al-Jeiroudi@sms.ed.ac.uk
†Email: J.Gondzio@ed.ac.uk, URL: http://maths.ed.ac.uk/~gondzio/
‡Email: J.A.J.Hall@ed.ac.uk, URL: http://maths.ed.ac.uk/hall/

Preconditioning Indefinite Systems in Interior Point Methods for Large Scale Linear Optimization

Abstract

We discuss the use of preconditioned conjugate gradients method for solving the reduced KKT systems arising in interior point algorithms for linear programming. The (indefinite) augmented system form of this linear system has a number of advantages, notably a higher degree of sparsity than the (positive definite) normal equations form. Therefore we use the conjugate gradients method to solve the augmented system and look for a suitable preconditioner.

An explicit null space representation of linear constraints is constructed by using a nonsingular basis matrix identified from an estimate of the optimal partition in the linear program. This is achieved by means of recently developed efficient basis matrix factorisation techniques which exploit hyper-sparsity and are used in implementations of the revised simplex method.

The approach has been implemented within the HOPDM interior point solver and applied to medium and large-scale problems from public domain test collections. Computational experience is encouraging.

1 Introduction

We are concerned in this paper with the use of the primal-dual interior point method (IPM for short) to solve large-scale linear programming problems. The primal-dual method is applied to the primal-dual formulation of the linear program

Primal		Dual	
min	$c^T x$	max	$b^T y$
s.t.	$Ax = b$,	s.t.	$A^T y + s = c$,
	$x \geq 0$;		y free, $s \geq 0$,

where $A \in \mathcal{R}^{m \times n}$, $x, s, c \in \mathcal{R}^n$ and $y, b \in \mathcal{R}^m$. We assume that $m \leq n$. The primal-dual algorithm is usually faster and more reliable than the pure primal or pure dual method [2, 27]. The main computational cost of this algorithm is the computation of the primal-dual Newton direction. Applying standard transformations [27] leads to the following linear system that must be solved at each iteration

$$\begin{bmatrix} \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -\Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1}$$

where $\Theta = XS^{-1}$ and X and S are diagonal matrices in $\mathcal{R}^{n \times n}$ with the elements of vectors x and s respectively on the diagonal, $f = c - A^T y - s + X^{-1}(XSe - \mu e)$, $g = Ax - b$ and μ is the average complementarity gap $\mu = x^T s/n$.

In many implementations, (1) is further reduced to the normal equations form

$$A\Theta A^T \Delta y = A\Theta f + g. \tag{2}$$

The matrix involved in the *augmented system* (1) is indefinite while the matrix involved in the *normal equations* (2) is positive definite. Since the matrix A displays a high degree of sparsity, both linear systems are sparse and can, in most cases, be solved by a very efficient direct method [2]. The solution strategies include the Bunch-Parlett factorisation [6] applied to (1), avoiding 2×2 pivots, Cholesky-like factorisation of a quasidefinite matrix [1, 25], or straightforward Cholesky factorisation applied to the positive definite matrix $A\Theta A^T$ [8]. However, there exist linear optimization problems for which both (1) and (2) display disastrous fill-in and make direct solution inefficient. We are concerned with these situations in this paper and consider the use of iterative methods instead.

The matrix involved in the normal equations (2) is positive definite and the method of conjugate gradients [14] is a natural candidate among iterative methods to be applied to this system. On the other hand, the normal equations become extremely ill-conditioned as the interior point method approaches the solution. Unless the conjugate gradients method is preconditioned, it will struggle while solving this system during the final iterations of the interior point method. Several attempts have been made to precondition the normal equations system, see for example [4, 5, 26] and the references therein.

In this paper we are going to use a Krylov-subspace method [16, 21, 24] to solve the augmented system equation (1). There are at least four important reasons why we decide to work with this particular system: (i) better conditioning, (ii) additional flexibility in exploiting sparsity, (iii) the

possibility of extension to the quadratic/nonlinear programming case, and (iv) the availability of a wider selection of preconditioners. We discuss them briefly below.

The first argument comes from the fact that after appropriate scaling, as suggested in [3], the augmented system (1) may have much better conditioning than the normal equations system (2). We expect that having better conditioning properties in (1) should make it easier to precondition the system for an iterative method.

The second reason comes from the comparison of *direct* methods applied to augmented system and normal equations. There are a number of examples which show that factoring the augmented system (1) often produces significant savings in the number of nonzero entries over factoring normal equations. The presence of dense columns in A provides a straightforward example of such a situation, see [2, 8] and the references therein. Compared with Cholesky factorisation for the normal equations, the augmented system factorisation enjoys an additional degree of freedom resulting from the ability to interchange pivots between diagonal elements of Θ and diagonal elements of the already filled $(2,2)$ block in (1). We expect that these sparsity advantages may be exploited when constructing preconditioners.

The third argument follows the observation that the reduction of the augmented system to the normal equations makes sense only when the $(1,1)$ block in (1) is a diagonal matrix. This is of course the case when linear programs are solved; however, when nonseparable quadratic or nonlinear programs are solved this block is a general sparse matrix. Using an iterative method for the augmented system opens the possibility of future extension to quadratic or nonlinear programming. We would like to mention at this point that there exists a class of constraint preconditioners which could be used in such context, see [12, 15, 17] to mention just a few.

The final reason comes from the analysis by Oliveira and Sorensen [18] who propose a preconditioner for the augmented system (1), and then reduce the preconditioned system to positive definite normal equations, allowing them to use the conjugate gradients method to solve (2). They show in [18] that all preconditioners for the normal equations system have an equivalent for the augmented system, while the opposite is not true. More precisely, they show that the whole classes of (different) preconditioners for the augmented system can result in the same preconditioner for the normal equations. We consider this to be a strong argument for constructing a preconditioner for the augmented system.

In this paper, we will design the preconditioner for the augmented system and we will go a step further than [18]. Instead of reducing the augmented system to normal equations and then applying an iterative method, we will use the conjugate gradients (CG) method to solve the indefinite system (1). We are aware of the disadvantages associated with applying the CG method to indefinite systems [9]. However, we are motivated by the recent analyses of Lukšan and Vlček [17] and Rozložník and Simoncini [20] showing that short recurrence iterative methods such as conjugate gradients can be applied to indefinite systems in certain situations. We will show in particular that the analysis of [20] may be applied to the preconditioner proposed in this paper.

There exists a wide range of iterative methods which can be used to solve the indefinite system (1). The family of Krylov-subspace methods [16, 21, 24] enjoys a particularly good reputation among different iterative methods. Since we plan to solve large systems of equations, we prefer to use a short recurrence method rather than a long recurrence one. The full recurrence

methods such as GMRES [22] occasionally do not manage to converge fast enough and become unacceptably expensive. Among the short recurrence methods we considered MINRES [19] and CG [14, 21, 24]. Bearing in mind that, whichever method is used, preconditioning is necessary, we decided not to use MINRES because this method requires a symmetric positive definite preconditioner, a restriction we would like to avoid. Summing up, encouraged by recent analyses [17, 20] we will apply the preconditioned conjugate gradients (PCG) method directly to the indefinite system (1).

The paper is organised as follows. In Section 2, we briefly review the primal-dual interior point method for linear programming. In Section 3, we introduce the indefinite preconditioner. In Section 4, we perform a spectral analysis of the preconditioned matrix. In Section 5, we take a closer look at the behaviour of conjugate gradients on the indefinite system: we follow [20] in the analysis of our preconditioner. In Section 6, we discuss the issues involved in the identification of a suitable subset of columns to produce a well-conditioned matrix B. We have implemented the conjugate gradients method with the indefinite preconditioner (10) in the context of the HOPDM interior point solver [11] and we have applied it to solve a number of medium and large-scale linear programming problems. In Section 7, we discuss our computational experience. In Section 8, we present the conclusions.

2 The interior point method for linear programming

It is widely accepted that the primal-dual interior point method is the most efficient variant of interior point method for linear programming [2, 27]. The usual transformation in interior point methods consists of replacing inequality constraints by the logarithmic barrier. The primal barrier problem becomes:

$$\begin{aligned} \min \quad & c^T x - \mu \sum_{j=1}^{n} \ln x_j \\ \text{s.t.} \quad & Ax = b, \end{aligned}$$

where $\mu \geq 0$ is a barrier parameter. The Lagrangian associated with this problem has the form:

$$L(x, y, \mu) = c^T x - y^T (Ax - b) - \mu \sum_{j=1}^{n} \ln x_j$$

and the conditions for a stationary point are

$$\begin{aligned} \nabla_x L(x, y, \mu) &= c - A^T y - \mu X^{-1} e = 0 \\ \nabla_y L(x, y, \mu) &= Ax - b = 0, \end{aligned}$$

where $X^{-1} = \text{diag}\{x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\}$. Denoting

$$s = \mu X^{-1} e, \qquad \text{i.e.} \qquad XSe = \mu e,$$

where $S = \text{diag}\{s_1, s_2, \ldots, s_n\}$ and $e = (1, 1, \ldots, 1)^T$, the first order optimality conditions (for the barrier problem) are:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= \mu e \\ (x, s) &\geq 0. \end{aligned} \tag{3}$$

The interior point algorithm for linear programming applies Newton's method to solve this system of nonlinear equations and gradually reduces the barrier parameter μ to guarantee convergence to the optimal solution of the original problem. The Newton direction is obtained by solving the system of linear equations:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix}, \tag{4}$$

where

$$\begin{aligned} \xi_p &= b - Ax, \\ \xi_d &= c - A^T y - s, \\ \xi_\mu &= \mu e - XSe. \end{aligned}$$

By eliminating

$$\Delta s = X^{-1}(\xi_\mu - S\Delta x) = -X^{-1}S\Delta x + X^{-1}\xi_\mu,$$

from the second equation we get the symmetric indefinite augmented system of linear equations

$$\begin{bmatrix} \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -\Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}. \tag{5}$$

where $\Theta = XS^{-1} \in \mathcal{R}^{n\times n}$ is a diagonal scaling matrix and the right-hand-side vectors satisfy $f = \xi_d - X^{-1}\xi_\mu$ and $g = -\xi_p$. To simplify notation we will drop a minus sign in Δx for the rest of this paper.

3 Indefinite preconditioner

To design the preconditioner for any form of reduced KKT system in interior point methods, we first observe that the ill-conditioning in linear systems (1) and (2) is a consequence of the properties of the diagonal scaling matrix Θ. From the complementarity condition for linear programs we know that, at the optimum, $\hat{x}_j\hat{s}_j = 0, \forall j \in \{1, 2, \ldots, n\}$. The condition $\hat{x}_j\hat{s}_j = 0$ is satisfied if at least one of the variables $\hat{x}_j$ and $\hat{s}_j$ is zero. Primal-dual interior point methods identify a strong optimal partition [27], that is, they produce an optimal solution with the property $\hat{x}_j + \hat{s}_j > 0, \forall j$. In other words, only one of $\hat{x}_j$ and $\hat{s}_j$ is zero. The set of indices $\{1, 2, \ldots, n\}$ can therefore be partitioned into two disjoint subsets:

$$\mathcal{B} = \{j \in \{1, 2, ..., n\} : \hat{x}_j > 0\} \text{ and } \mathcal{N} = \{j \in \{1, 2, ..., n\} : \hat{s}_j > 0\}.$$

In fact, the optimal partition is closely related (but not equivalent to) the basic-nonbasic partition in the simplex method.

Unlike the simplex method which satisfies the complementarity condition at each iteration, the interior point method satisfies this condition only in the limit. The primal-dual interior point method identifies a strong optimal partition near the optimal solution. Below we will summarise its asymptotic behaviour and use the arrow to denote "converges to". If at the optimal solution

$j \in \mathcal{B}$, then $x_j \to \hat{x}_j > 0$ and $s_j \to 0$, hence the corresponding element $\theta_j \to \infty$. If at the optimal solution $j \in \mathcal{N}$, then $x_j \to 0$ and $s_j \to \hat{s}_j > 0$ and $\theta_j \to 0$. Summing up,

$$\theta_j \to \begin{cases} \infty, & \text{if } j \in \mathcal{B} \\ 0, & \text{if } j \in \mathcal{N}, \end{cases} \qquad \text{and} \qquad \theta_j^{-1} \to \begin{cases} 0, & \text{if } j \in \mathcal{B} \\ \infty, & \text{if } j \in \mathcal{N}. \end{cases} \tag{6}$$

This property of interior point methods is responsible for a number of numerical difficulties. In particular, it causes both linear systems (1) and (2) to become very ill-conditioned when an interior point method approaches the optimal solution [2]. However, it may be used to advantage when constructing a preconditioner for the iterative method.

We partition the matrices and vectors:

$$A = [A_{\mathcal{B}}, A_{\mathcal{N}}], \quad \Theta = \begin{bmatrix} \Theta_{\mathcal{B}} & 0 \\ 0 & \Theta_{\mathcal{N}} \end{bmatrix}, \quad x = [x_{\mathcal{B}}, x_{\mathcal{N}}], \quad \text{and} \quad s = [s_{\mathcal{B}}, s_{\mathcal{N}}]$$

according to the partition of $\{1, 2, \ldots, n\}$ into sets $\mathcal{B}$ and $\mathcal{N}$. With this notation, from (6) we conclude that $\Theta_{\mathcal{N}} \approx 0$ and $\Theta_{\mathcal{B}}^{-1} \approx 0$. Consequently, the matrix in the augmented system (1) can be approximated as follows:

$$\begin{bmatrix} \Theta_{\mathcal{B}}^{-1} & & A_{\mathcal{B}}^T \\ & \Theta_{\mathcal{N}}^{-1} & A_{\mathcal{N}}^T \\ A_{\mathcal{B}} & A_{\mathcal{N}} & \end{bmatrix} \approx \begin{bmatrix} & & A_{\mathcal{B}}^T \\ & \Theta_{\mathcal{N}}^{-1} & A_{\mathcal{N}}^T \\ A_{\mathcal{B}} & A_{\mathcal{N}} & \end{bmatrix}, \tag{7}$$

and the matrix in the normal equations system (2) can be approximated as follows:

$$A\Theta A^T = A_{\mathcal{B}}\Theta_{\mathcal{B}}A_{\mathcal{B}}^T + A_{\mathcal{N}}\Theta_{\mathcal{N}}A_{\mathcal{N}}^T \approx A_{\mathcal{B}}\Theta_{\mathcal{B}}A_{\mathcal{B}}^T. \tag{8}$$

If the matrix $A_{\mathcal{B}}$ were square and nonsingular then equations (7) and (8) would suggest obvious preconditioners for the augmented system and normal equations, respectively. However, there is no guarantee that this is the case. On the contrary, in practical applications it is very unlikely that the matrix $A_{\mathcal{B}}$ corresponding to the optimal partition is square and nonsingular. Moreover, the optimal partition is known only when an IPM approaches the optimal solution of the linear program.

To construct a preconditioner to (1) with a structure similar to the approximation (7) we need to guess an optimal partition and, additionally, guarantee that the matrix B which approximates $A_{\mathcal{B}}$ is nonsingular. We exploit the difference in magnitude of elements in Θ to design a preconditioner. We sort the elements of Θ in non-increasing order: $\theta_1 \geq \theta_2 \geq \theta_3 \geq \cdots \geq \theta_n$. Hence the elements of Θ^{-1} satisfy $\theta_1^{-1} \leq \theta_2^{-1} \leq \theta_3^{-1} \leq \cdots \leq \theta_n^{-1}$. If the primal-dual iterate is sufficiently close to an optimal solution, then the first elements θ_j^{-1} in this list correspond to variables x_j which are most likely to be nonzero at the optimum, and the last elements in the list correspond to variables which are likely to be zero at the optimum. We select the first m linearly independent columns of the matrix A, when permuted according to the order of θ_j^{-1}, and we construct a nonsingular matrix B from these columns. The submatrix of A corresponding to all the remaining columns is denoted by N. Therefore we assume that a partition $A = [B, N]$ is known such that B is nonsingular and the entries θ_j^{-1} corresponding to columns of B are chosen from the smallest elements of Θ^{-1}. According to this partitioning of A and Θ (and after a symmetric row and column permutation) the indefinite matrix in (1) can be rewritten in the following form

$$K = \begin{bmatrix} \Theta_B^{-1} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix}. \tag{9}$$

By construction, the elements of Θ_B^{-1} are supposed to be among the smallest elements of Θ^{-1}, hence we may assume that $\Theta_B^{-1} \approx 0$. The following easily invertible block-triangular matrix

$$P = \begin{bmatrix} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \tag{10}$$

is a good approximation to K. Hence P is an attractive preconditioner for K. We should mention that Oliveira and Sorensen [18] use a similar partitioning process to derive their preconditioner for the normal equations. They order the columns of the matrix $A\Theta^{-1}$ from the smallest to the largest with respect to the 1-norm and then scan the columns of A in this order to select the first m that are linearly independent.

Since the matrix B was constructed from columns corresponding to the smallest possible elements of Θ^{-1} we may expect that $\|\Theta_B^{-1}\|_F \ll \|\Theta_N^{-1}\|_F$, where $\|.\|_F$ denotes the Frobenius norm of the matrix. Using (9) and (10) we derive the following bound on the square of the Frobenius norm of the difference of matrices K and P:

$$\|K - P\|_F^2 = \|\Theta_B^{-1}\|_F^2 \ll \|P\|_F^2 < \|K\|_F^2. \tag{11}$$

Summing up, P is a good approximation to K (since the approximation error is small in relation to $\|P\|_F^2$ and $\|K\|_F^2$) and we may consider it as a possible preconditioner of K.

Naturally there are further requirements that a successful preconditioner should satisfy: it should be easily invertible and it should capture the numerical properties of (9). P is easily invertible because it is block-triangular with nonsingular diagonal blocks B, Θ_N^{-1} and B^T. We conclude this section by giving explicit formulae for the solution of equations with the preconditioner (10) and leave the analysis of spectral properties of the preconditioned matrix $P^{-1}K$ to Section 4.

3.1 Solving equations with P

The matrix (10) is block triangular and its diagonal blocks B, Θ_N^{-1} and B^T are invertible. Let $d = [d_B, d_N, d_y]$ and $r = [r_B, r_N, r_y]$ and consider the system of equations

$$\begin{bmatrix} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \begin{bmatrix} d_B \\ d_N \\ d_y \end{bmatrix} = \begin{bmatrix} r_B \\ r_N \\ r_y \end{bmatrix}. \tag{12}$$

The solution of (12) can easily be computed by exploiting the block-triangular structure of the matrix:

$$\begin{aligned} B^T d_y = r_B &\quad\Rightarrow\quad d_y = B^{-T} r_B \\ \Theta_N^{-1} d_N + N^T d_y = r_N &\quad\Rightarrow\quad d_N = \Theta_N r_N - \Theta_N N^T d_y \\ B d_B + N d_N = r_y &\quad\Rightarrow\quad d_B = B^{-1}(r_y - N d_N). \end{aligned} \tag{13}$$

The operation $d = P^{-1}r$ involves solving two equations (one with B and one with B^T) and a couple of matrix-vector multiplications. These operations will be performed at every iteration of the conjugate gradients procedure hence they should be implemented in the most efficient way. The issues of choosing a well-conditioned basis matrix B with sparse factored inverse are addressed in Section 6.

4 Spectral analysis

We have observed earlier that if Θ_B is chosen carefully and $\|\Theta_B^{-1}\|_F \ll \|\Theta_N^{-1}\|_F$ then the preconditioner (10) is a good approximation to K in (9). To assess the quality of the preconditioner we need a better understanding of the relation between P and K.

We will therefore analyse the spectral properties of the preconditioned matrix $P^{-1}K$. Let us use the notation $Kt = r$ to denote the system (1), where $t = [\Delta x, \Delta y]$ and $r = [f, g]$. Given a starting approximation $t^{(0)}$ and the associated residual $r^{(0)} = r - Kt^{(0)}$ the indefinite preconditioner may be applied either from the right, yielding the system

$$KP^{-1}\hat{t} = r, \qquad t = P^{-1}\hat{t}, \tag{14}$$

or from the left, so that the system to be solved becomes

$$P^{-1}Kt = P^{-1}r. \tag{15}$$

The right and the left preconditioned matrices KP^{-1} and $P^{-1}K$ have the same eigenvalues so general spectral results can be given in terms of either of the two formulations. The following theorem shows that the eigenvalues of the $P^{-1}K$ matrix are real and positive. Moreover they are bounded away from zero.

Theorem 1 *Let λ be an eigenvalue of $P^{-1}K$. Then λ is real and $\lambda \geq 1$.*

Proof Let v be an eigenvector of $P^{-1}K$ corresponding to the eigenvalue λ, that is, $P^{-1}Kv = \lambda v$. Let $\lambda = 1 + \tau$ and, applying the usual partitioning $v = [v_B, v_N, v_y]$, the eigensystem can be written as $Kv = (1 + \tau)Pv$:

$$\begin{bmatrix} \Theta_B^{-1} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \begin{bmatrix} v_B \\ v_N \\ v_y \end{bmatrix} = (1+\tau) \begin{bmatrix} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \begin{bmatrix} v_B \\ v_N \\ v_y \end{bmatrix}$$

which yields

$$\begin{aligned} \Theta_B^{-1} v_B &= \tau B^T v_y \\ \tau(\Theta_N^{-1} v_N + N^T v_y) &= 0 \\ \tau(B v_B + N v_N) &= 0. \end{aligned}$$

We consider two cases. When $\tau = 0$ clearly $\lambda = 1$ so the claim is true. Otherwise, when $\tau \neq 0$, the equation system can be simplified:

$$\begin{aligned} \Theta_B^{-1} v_B &= \tau B^T v_y \\ \Theta_N^{-1} v_N + N^T v_y &= 0 \\ B v_B + N v_N &= 0, \end{aligned}$$

and solved for τ. From the third equation we get $v_B = -B^{-1}N v_N$ and, substituting this in the first equation, yields $N v_N = -\tau B \Theta_B B^T v_y$. Next, we use the second equation to substitute for $v_N = -\Theta_N N^T v_y$ giving

$$(N \Theta_N N^T) v_y = \tau (B \Theta_B B^T) v_y.$$

If $v_y = 0$ then (using $\tau \neq 0$) we deduce that $v_B = 0$ and $v_N = 0$, that is the eigenvector is zero. We can exclude such a situation and safely assume that $v_y \neq 0$. In this case, we multiply both sides of the equation by v_y^T to get

$$v_y^T(N\Theta_N N^T)v_y = \tau v_y^T(B\Theta_B B^T)v_y.$$

Since all the elements of Θ are positive and B is nonsingular, the matrix $B\Theta_B B^T$ is symmetric positive definite and the matrix $N\Theta_N N^T$ is symmetric positive semidefinite. Hence we conclude that

$$\tau = \frac{v_y^T(N\Theta_N N^T)v_y}{v_y^T(B\Theta_B B^T)v_y} \geq 0, \tag{16}$$

which completes the proof. □

The proof reveals the importance of the correct partitioning of $A = [B, N]$. Indeed, this partition should have a number of desirable features:

- B should be nonsingular and well-conditioned since we should operate accurately with the preconditioner;
- All elements in Θ_B^{-1} should be small in comparison with those in Θ_N^{-1}.

The condition $\|\Theta_B^{-1}\|_F \ll \|\Theta_N^{-1}\|_F$ is relatively easy to satisfy. However, (16) indicates that we need a stronger property: we would like to bound τ from above and, in that way, cluster all eigenvalues of $P^{-1}K$ in an interval $[1, \lambda_{max}]$, with λ_{max} kept as small as possible. This opens questions regarding the necessary concessions to be made when the matrix B and the corresponding Θ_B are chosen. The ability to identify a well-conditioned matrix B consisting of columns for which the θ_j are "large" is crucial for the good/efficient behaviour of our approach. We discuss these issues in detail in Section 6.

5 The preconditioned conjugate gradient method

We are dealing with large and sparse problems and we are looking for an iterative method from the Krylov-subspace family which can solve the augmented system (1) efficiently. Moreover, we would like to use a short recurrence method to avoid excessive storage requirements. Between MINRES [19] and CG [14] we choose the latter because it allows for the use of an indefinite preconditioner. We follow [20] and show that this can be done under the condition that a specific starting point is used so that a zero block in the residual vector is preserved. We discuss the details below.

5.1 The preconditioned CG method for the the indefinite system

Rozložník and Simoncini [20] used the BiCG method to solve an indefinite system such as (1) preconditioned from the right. They show that the right preconditioned BiCG method reduces to the standard preconditioned CG method if the following two properties hold. The first property is that the preconditioned matrix $H = KP^{-1}$ is J-symmetric, where $J = P^{-1}$, and the second

is that $g = 0$. The reason behind this is that when $g = 0$ the residual of PCG has a zero block and can be expressed as $r^{(j)} = [s^{(j)}, 0]$. Although in our case $g \neq 0$, the initial iterate $t^{(0)}$ can be chosen so that the corresponding residual has the form $r^{(0)} = [s^{(0)}, 0]$. Furthermore, the preconditioned matrix $H = KP^{-1}$ is J-symmetric, since $H^T J = JH$. See [20].

Let us consider the following starting point for CG:

$$t^{(0)} = \begin{bmatrix} \Delta x_B^{(0)} \\ \Delta x_N^{(0)} \\ \Delta y^{(0)} \end{bmatrix} = \begin{bmatrix} B^{-1}g \\ 0 \\ 0 \end{bmatrix},$$

where $\Delta x = \begin{bmatrix} \Delta x_B \\ \Delta x_N \end{bmatrix}$. The initial residual $r^{(0)} = r - Kt^{(0)}$ may then be written as

$$r^{(0)} = \begin{bmatrix} f_B \\ f_N \\ g \end{bmatrix} - \begin{bmatrix} \Theta_B^{-1} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \begin{bmatrix} B^{-1}g \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f_B - \Theta_B^{-1}B^{-1}g \\ f_N \\ 0 \end{bmatrix}.$$

Note two interesting properties of the preconditioned matrix KP^{-1} stated as two Lemmas below. Multiplying by the preconditioned matrix KP^{-1} preserves a zero block in the third component of the vector.

Lemma 2 *Let* $t = \begin{bmatrix} v_B \\ v_N \\ 0 \end{bmatrix}$. *Then* $KP^{-1}t = \begin{bmatrix} z_B \\ z_N \\ 0 \end{bmatrix}$.

Proof We note first that, by using (12)-(13), we may write $u = P^{-1}t$ as

$$u = \begin{bmatrix} B^{-1}N\Theta_N N^T B^{-T} v_B - B^{-1}N\Theta_N v_N \\ -\Theta_N N^T B^{-T} v_B + \Theta_N v_N \\ B^{-T} v_B \end{bmatrix}.$$

Hence

$$\begin{aligned} KP^{-1}t &= Ku = \begin{bmatrix} \Theta_B^{-1} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \begin{bmatrix} B^{-1}N\Theta_N N^T B^{-T} v_B - B^{-1}N\Theta_N v_N \\ -\Theta_N N^T B^{-T} v_B + \Theta_N v_N \\ B^{-T} v_B \end{bmatrix} \\ &= \begin{bmatrix} (I + \Theta_B^{-1}B^{-1}N\Theta_N N^T B^{-T})v_B - \Theta_B^{-1}B^{-1}N\Theta_N v_N \\ v_N \\ 0 \end{bmatrix}, \end{aligned}$$

which completes the proof. □

Furthermore, using the initial approximate solution

$$t^{(0)} = \begin{bmatrix} \Delta x_B^{(0)} \\ \Delta x_N^{(0)} \\ \Delta y^{(0)} \end{bmatrix} = \begin{bmatrix} B^{-1}(g - N\Theta_N f_N) \\ \Theta_N f_N \\ 0 \end{bmatrix},$$

the residuals will have two zero blocks, $r = \begin{bmatrix} r_B \\ 0 \\ 0 \end{bmatrix}$.

The initial residual $r^{(0)} = r - Kt^{(0)}$ may be written:

$$r^{(0)} = \begin{bmatrix} f_B \\ f_N \\ g \end{bmatrix} - \begin{bmatrix} \Theta_B^{-1} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \begin{bmatrix} B^{-1}(g - N\Theta_N f_N) \\ \Theta_N f_N \\ 0 \end{bmatrix} = \begin{bmatrix} f_B - \Theta_B^{-1}B^{-1}g + \Theta_B^{-1}B^{-1}N\Theta_N f_N \\ 0 \\ 0 \end{bmatrix}.$$

We observe an important property of the preconditioned matrix: multiplying with the matrix KP^{-1} preserves the zero blocks in the second and third components of the vector.

Lemma 3 *Let* $t = \begin{bmatrix} v_B \\ 0 \\ 0 \end{bmatrix}$. *Then* $KP^{-1}t = \begin{bmatrix} z_B \\ 0 \\ 0 \end{bmatrix}$.

Proof We note first that, by using (12)-(13), we may write $u = P^{-1}t$ as

$$u = \begin{bmatrix} B^{-1}N\Theta_N N^T B^{-T} v_B \\ -\Theta_N N^T B^{-T} v_B \\ B^{-T} v_B \end{bmatrix}$$

hence

$$KP^{-1}t = Ku = \begin{bmatrix} \Theta_B^{-1} & & B^T \\ & \Theta_N^{-1} & N^T \\ B & N & \end{bmatrix} \begin{bmatrix} B^{-1}N\Theta_N N^T B^{-T} v_B \\ -\Theta_N N^T B^{-T} v_B \\ B^{-T} v_B \end{bmatrix} = \begin{bmatrix} (I + \Theta_B^{-1}B^{-1}N\Theta_N N^T B^{-T}) v_B \\ 0 \\ 0 \end{bmatrix},$$

which completes the proof. □

The residual in each PCG iteration is computed as a linear combination of the previous residual and the matrix KP^{-1} applied to the previous residual. This implies that $r^{(j)} = [s^{(j)}, 0]$ for $j = 0,1,\ldots$ and, hence, we can use the standard PCG method along with (10) to solve (1).

Below, we recall the general preconditioned conjugate gradients algorithm.

PCG Algorithm.

$$\begin{aligned} t^{(0)} &= \begin{bmatrix} \Delta x^{(0)} \\ \Delta y^{(0)} \end{bmatrix} \\ r^{(0)} &= r - Kt^{(0)} = \begin{bmatrix} s^{(0)} \\ 0 \end{bmatrix} \\ d^{(0)} &= P^{-1}r^{(0)} \\ \text{For } j = 0, 1, \ldots & \\ \alpha^{(j)} &= (r^{(j)})^T P^{-1} r^{(j)} / (d^{(j)})^T K d^{(j)} \\ t^{(j+1)} &= t^{(j)} + \alpha^{(j)} d^{(j)} \\ r^{(j+1)} &= r^{(j)} - \alpha^{(j)} K d^{(j)} \\ \beta^{(j)} &= (r^{(j+1)})^T P^{-1} r^{(j+1)} / (r^{(j)})^T P^{-1} r^{(j)} \\ d^{(j+1)} &= P^{-1} r^{(j+1)} + \beta^j d^{(j)}. \end{aligned} \tag{17}$$

Note that we use a customised form of this algorithm which exploits the special structure (zero blocks) in the right hand side (Lemma 3).

5.2 The convergence of the PCG method

In this section, we analyse the behaviour of the PCG method for the indefinite system (1) and give explicit formulae describing the convergence of the method. We follow [20] and apply a similar analysis to our preconditioner. The convergence analysis of the PCG method is important because both K and P are indefinite matrices.

The PCG algorithm (17) generates iterates $t^{(j)}, j = 0, 1, \ldots$ with residuals $r^{(j)} = r - Kt^{(j)}$. The error corresponding to each PCG iteration has the form $e^{(j)} = t^{(j)} - t^*$, where t^* is the solution of (1), and the residual can be written as $r^{(j)} = -Ke^{(j)}$ since $Ke^{(j)} = Kt^{(j)} - Kt^* = -r^{(j)}$. In Lemma 4 we prove that the indefinite K-inner product of the error $e^{(j)}$ in the PCG algorithm is always non-negative so we can write $\|e^{(j)}\|_K = \sqrt{< e^{(j)}, Ke^{(j)} >}$, even though K is not positive definite. In Theorem 5 we show that the K-norm of the error $e^{(j)}$ is minimized over the eigenvalues of the symmetric positive definite matrices. In other words, the error term displays asymptotic convergence similar to that observed when PCG is applied to positive definite systems.

Lemma 4 *The indefinite K-inner product $< e^{(j)}, Ke^{(j)} >$ is non-negative for any error $e^{(j)}$ hence it defines a norm*

$$\|e^{(j)}\|_K = \sqrt{< e^{(j)}, Ke^{(j)} >} = \|e_1^{(j)}\|_{\Theta^{-1}}. \tag{18}$$

Proof We have shown in Lemma 2 that, for a suitable initial solution, the residual has the form $r^{(j)} = [s^{(j)}, 0]$. Hence

$$r^{(j)} = -Ke^{(j)} = -\begin{bmatrix} \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} e_1^{(j)} \\ e_2^{(j)} \end{bmatrix} = \begin{bmatrix} -\Theta^{-1}e_1^{(j)} - A^T e_2^{(j)} \\ -Ae_1^{(j)} \end{bmatrix},$$

implies $Ae_1^{(j)} = 0$. Simple calculations give the following result

$$\begin{aligned} < e^{(j)}, Ke^{(j)} > &= (e^{(j)})^T Ke^{(j)} = \begin{bmatrix} (e_1^{(j)})^T & (e_2^{(j)})^T \end{bmatrix} \begin{bmatrix} \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} e_1^{(j)} \\ e_2^{(j)} \end{bmatrix} \\ &= (e_1^{(j)})^T \Theta^{-1} e_1^{(j)} + (e_1^{(j)})^T A^T e_2^{(j)} + (e_2^{(j)})^T A e_1^{(j)} \\ &= (e_1^{(j)})^T \Theta^{-1} e_1^{(j)} \\ &= (e_B^{(j)})^T \Theta_B^{-1} e_B^{(j)} + (e_N^{(j)})^T \Theta_N^{-1} e_N^{(j)} \geq 0 \end{aligned} \tag{19}$$

because Θ^{-1} is positive definite. This gives $\|e^{(j)}\|_K = \|e_1^{(j)}\|_{\Theta^{-1}}$, which completes the proof. □

Let D_j be the Krylov subspace $D_j = span\{d^{(0)}, d^{(1)}, ..., d^{(j-1)}\}$. Then $D_1 = span\{d^{(0)}\} = span\{P^{-1}r^{(0)}\}$. $D_2 = span\{d^{(0)}, d^{(1)}\}$, where the direction $d^{(1)}$ is a linear combination of the previous direction and $P^{-1}r^{(1)}$, while $r^{(1)}$ is a linear combination of the previous residual and $Kd^{(0)}$. This implies that $d^{(1)}$ is a linear combination of $d^{(0)}$ and $P^{-1}KP^{-1}r^{(0)}$, which gives $D_2 =$

$span\{P^{-1}r^{(0)}, P^{-1}KP^{-1}r^{(0)}\}$. By the same argument $d^{(j-1)}$ is a linear combination of $d^{(j-2)}$ and $(P^{-1}K)^{j-1}P^{-1}r^{(0)}$, giving $D_j = span\{P^{-1}r^{(0)}, P^{-1}KP^{-1}r^{(0)}, ..., (P^{-1}K)^{j-1}P^{-1}r^{(0)}\}$. Moreover, $r^{(0)} = -Ke^{(0)}$ so $D_j = span\{P^{-1}Ke^{(0)}, (P^{-1}K)^2e^{(0)}, \ldots, (P^{-1}K)^je^{(0)}\}$.

The error can be written as $e^{(j)} = e^{(j-1)} + \alpha^{(j-1)}d^{(j-1)}$, hence $e^{(j)} = e^{(0)} + \sum_{k=0}^{j-1}\alpha^{(k)}d^{(k)}$. Since $d^{(j)} \in D_{j+1}$ the error can be written as $e^{(j)} = (I + \sum_{k=1}^{j}\psi^{(k)}(P^{-1}K)^k)e^{(0)}$, where the coefficient $\psi^{(k)}$ is related to $\alpha^{(k)}$ and $\beta^{(k)}$. Hence the error term can be expressed as

$$e^{(j)} = \phi_j(P^{-1}K)e^{(0)}, \tag{20}$$

where ϕ_j is a polynomial of degree j and we require that $\phi_j(0) = 1$.

Theorem 5 *Let $e^{(0)}$ be the initial error of PCG. Then*

$$\|e^{(j)}\|_K^2 \leq \min_{\phi\in P_j,\phi(0)=1}\max_{\lambda\in\Lambda(S_1)}[\phi(\lambda)]^2\|e_B^{(0)}\|^2_{\Theta_B^{-1}} + \min_{\phi\in P_j,\phi(0)=1}\max_{\lambda\in\Lambda(S_2)}[\phi(\lambda)]^2\|e_N^{(0)}\|^2_{\Theta_N^{-1}}, \tag{21}$$

where $S_1 = I_m + \Theta_B^{-1/2}B^{-1}N\Theta_N N^TB^{-T}\Theta_B^{-1/2}$ and $S_2 = I_{n-m} + \Theta_N^{1/2}N^TB^{-T}\Theta_B^{-1}B^{-1}N\Theta_N^{1/2}$ are symmetric positive definite matrices.

Proof First, we observe that $Ae_1^{(0)} = 0$, that is $Be_B^{(0)} + Ne_N^{(0)} = 0$, and hence we write

$$Ke^{(0)} = \begin{bmatrix} \Theta_B^{-1}e_B^{(0)} + B^Te_2^{(0)} \\ \Theta_N^{-1}e_N^{(0)} + N^Te_2^{(0)} \\ 0 \end{bmatrix}$$

and, using (13), we get

$$P^{-1}Ke^{(0)} = \begin{bmatrix} B^{-1}N\Theta_N N^TB^{-T}\Theta_B^{-1}e_B^{(0)} - B^{-1}Ne_N^{(0)} \\ -\Theta_N N^TB^{-T}\Theta_B^{-1}e_B^{(0)} + e_N^{(0)} \\ B^{-T}\Theta_B^{-1}e_B^{(0)} + e_2^{(0)} \end{bmatrix}.$$

Since $Be_B^{(0)} + Ne_N^{(0)} = 0$, that is $e_B^{(0)} = -B^{-1}Ne_N^{(0)}$ and $Ne_N^{(0)} = -Be_B^{(0)}$, we obtain

$$\begin{aligned} P^{-1}Ke^{(0)} &= \begin{bmatrix} B^{-1}N\Theta_N N^TB^{-T}\Theta_B^{-1}e_B^{(0)} - B^{-1}(-Be_B^{(0)}) \\ -\Theta_N N^TB^{-T}\Theta_B^{-1}(-B^{-1}Ne_N^{(0)}) + e_N^{(0)} \\ B^{-T}\Theta_B^{-1}e_B^{(0)} + e_2^{(0)} \end{bmatrix} \\ &= \begin{bmatrix} \Theta_B(\Theta_B^{-1} + \Theta_B^{-1}B^{-1}N\Theta_N N^TB^{-T}\Theta_B^{-1})e_B^{(0)} \\ \Theta_N(\Theta_N^{-1} + N^TB^{-T}\Theta_B^{-1}B^{-1}N)e_N^{(0)} \\ B^{-T}\Theta_B^{-1}e_B^{(0)} + e_2^{(0)} \end{bmatrix}. \end{aligned} \tag{22}$$

Let us define $C_1 = \Theta_B^{-1} + \Theta_B^{-1}B^{-1}N\Theta_N N^TB^{-T}\Theta_B^{-1}$ and $C_2 = \Theta_N^{-1} + N^TB^{-T}\Theta_B^{-1}B^{-1}N$. It is easy to prove that C_1 and C_2 are symmetric and positive definite matrices. By repeating a similar argument to the one used to derive (22) we obtain

$$\phi(P^{-1}K)e^{(0)} = \begin{bmatrix} \phi(\Theta_B C_1)e_B^{(0)} \\ \phi(\Theta_N C_2)e_N^{(0)} \\ * \end{bmatrix}. \tag{23}$$

We observe that it is not necessary to compute the last component of the vector $P^{-1}Ke^{(0)}$ because Lemma 4 guarantees that this component does not contribute to $\|e^{(j)}\|_K^2$.

Using (23) to compute the K-norm of the error (19) we obtain

$$\|\phi_j(P^{-1}K)e^{(0)}\|_K^2 \;=\; \|\phi_j(\Theta_B C_1)e_B^{(0)}\|_{\Theta_B^{-1}}^2 + \|\phi_j(\Theta_N C_2)e_N^{(0)}\|_{\Theta_N^{-1}}^2. \tag{24}$$

Let us observe that $(\Theta_B C_1)^k = \Theta_B^{1/2}(\Theta_B^{1/2}C_1\Theta_B^{1/2})^k\Theta_B^{-1/2} = \Theta_B^{1/2}S_1^k\Theta_B^{-1/2}$, where $S_1 = \Theta_B^{1/2}C_1\Theta_B^{1/2}$ is a symmetric and positive definite matrix. Analogously, we define another symmetric positive definite matrix $S_2 = \Theta_N^{1/2}C_2\Theta_N^{1/2}$ and observe that $(\Theta_N C_2)^k = \Theta_N^{1/2}S_2^k\Theta_N^{-1/2}$. Using these definitions, the two terms on the right-hand-side of (24) can be simplified as follows

$$\begin{aligned}
\|\phi_j(\Theta_B C_1)e_B^{(0)}\|_{\Theta_B^{-1}}^2 &= \|\Theta_B^{1/2}\phi_j(S_1)\Theta_B^{-1/2}e_B^{(0)}\|_{\Theta_B^{-1}}^2 = \|\phi_j(S_1)\Theta_B^{-1/2}e_B^{(0)}\|^2,\\
\|\phi_j(\Theta_N C_2)e_N^{(0)}\|_{\Theta_N^{-1}}^2 &= \|\Theta_N^{1/2}\phi_j(S_2)\Theta_N^{-1/2}e_N^{(0)}\|_{\Theta_N^{-1}}^2 = \|\phi_j(S_2)\Theta_N^{-1/2}e_N^{(0)}\|^2,
\end{aligned}$$

hence the K-norm error in (24) is

$$\|e^{(j)}\|_K^2 = \|\phi_j(S_1)\Theta_B^{-1/2}e_B^{(0)}\|^2 + \|\phi_j(S_2)\Theta_N^{-1/2}e_N^{(0)}\|^2. \tag{25}$$

From (20) we have $\|e^{(j)}\|_K^2 = \|\phi_j(P^{-1}K)e^{(0)}\|_K^2$, where ϕ_j is a polynomial of degree j and $\phi_j(0) = 1$. Now we can write

$$\|e^{(j)}\|_K^2 \le \min_{\phi\in P_j,\phi(0)=1}\max_{\lambda\in\Lambda(S_1)}[\phi(\lambda)]^2\|\Theta_B^{-1/2}e_B^{(0)}\|^2 + \min_{\phi\in P_j,\phi(0)=1}\max_{\lambda\in\Lambda(S_2)}[\phi(\lambda)]^2\|\Theta_N^{-1/2}e_N^{(0)}\|^2$$

and the claim is proved after substituting $\|\Theta_B^{-1/2}e_B^{(0)}\|^2 = \|e_B^{(0)}\|_{\Theta_B^{-1}}^2$ and $\|\Theta_N^{-1/2}e_N^{(0)}\|^2 = \|e_N^{(0)}\|_{\Theta_N^{-1}}^2$. □

The K-norm of the error $e^{(j)} = \phi_j(P^{-1}K)e^{(0)}$ is minimized over the eigenvalues of the symmetric positive definite matrices S_1 and S_2 so the error term decreases asymptotically.

6 Identifying and factorising the matrix B

The preconditioner P was derived on the assumption that it should be significantly cheaper to compute sparse factors of just the matrix B than computing a Cholesky factorisation of the coefficient matrix of the normal equations. Assuming that A has full row rank, we can find an m by m non-singular sub-matrix B.

The matrix B is given by the first m linearly independent columns of the matrix $\widetilde{A}$, where the columns of $\widetilde{A}$ are those of the constraint matrix A, ordered by increasing value of θ_j^{-1}. The set of columns forming B is identified by applying Gaussian elimination to the matrix $\widetilde{A}$, as described below. Although this yields an LU factorisation of B, the factorisation is not efficient with respect to sparsity and its use in subsequent PCG iterations would be costly. This potential cost is reduced significantly by using the Tomlin matrix inversion procedure [23] to determine the factorisation of B for use in PCG iterations. The Tomlin procedure is a relatively simple method of triangularisation and factorisation that underpins the highly efficient implementation of the revised simplex method described by Hall and McKinnon [13]. Since the matrix B is analogous to a simplex basis matrix, the use of the Tomlin procedure in this paper is expected to be similarly advantageous.

6.1 Identifying the columns of B via Gaussian elimination

When applying Gaussian elimination to the matrix $\widetilde{A}$ in order to identify the set of columns forming B, it is important to stress that the matrix $\widetilde{A}$ is not updated when elimination operations are identified. The linear independence of a particular column of $\widetilde{A}$, with respect to columns already in B, is determined as follows.

Suppose that k columns of B have been determined and let L_k be the current lower triangular matrix of elimination multipliers. Let a_q be the first column of $\widetilde{A}$ that has not yet been considered for inclusion in B. The system $L_k\hat{a}_q = a_q$ is solved and the entries of the pivotal column $\hat{a}_q$ are scanned for a good pivotal value. If there are no acceptable pivots, indicating that a_q is linearly dependent on the columns already in B, then a_q is discarded. Otherwise, a pivot is chosen and a_q is added to the set of columns forming B.

At least m systems of the form $L_k\hat{a}_q = a_q$ must be solved in order to identify all the columns of B. For some problems, a comparable number of linearly dependent columns of $\widetilde{A}$ are encountered before a complete basis is formed. Thus the efficiency with which $L_k\hat{a}_q = a_q$ is solved is crucial. Additionally, the ill-conditioning of B may lead to PCG being prohibitively expensive. This issue of efficiency is addressed in the following two ways.

Firstly, in order to reduce the number of nonzeros in the matrices L_k, the pivotal entry in $\hat{a}_q$ is selected from the set of acceptable pivots on grounds of sparsity. If the matrix $\widetilde{A}$ were updated with respect to elimination operations, then the acceptable pivot of minimum row count could be chosen. Since this is not known, a set of approximate row counts is maintained and used to discriminate between acceptable pivots. This set of approximate row counts is initialised to be correct and then, as elimination operations are identified, updated according to the maximum fill-in that could occur were $\widetilde{A}$ to be updated.

Secondly, since a_q is sparse, consideration is given to the likelihood that $\hat{a}_q$ is also sparse. This is trivially the case when $k = 0$ since $\hat{a}_q = a_q$. Since the columns of L_k are subsets of the entries in pivotal columns, it follows that for small values of k, $\hat{a}_q$ will remain sparse. For some important classes of LP problems, this property holds for all k and is analogous to what Hall and McKinnon term hyper-sparsity [13]. Techniques for exploiting hyper-sparsity when forming $\hat{a}_q$ analogous to those described in [13] have been used when computing the preconditioner and have led to significant improvements in computational performance.

7 Numerical results

The method discussed in this paper has been implemented in the context of HOPDM [11]. We have implemented the preconditioned conjugate gradients method for the augmented system given a specific starting point. In the implementation, the starting point with two zero blocks in its residual is used, see Section 5.1, Lemma 3. We consider a subset of the linear programming problems from the Netlib [10], Kennington [7] and other public test sets used in [18]. In this section we indicate that the new approach can be very effective in some cases, and that the new approach is an important option for some classes of problems.

In the initial iterations of the interior point method the normal equations are solved using the

Problem	Dimensions			Nonzeros in Factors		Memory
	m	n	nz(A)	nz(B)	nz(L)	Ratio
aircraft	3754	7517	24034	9754	1417131	145
chr12a	947	1662	5820	5801	78822	14
chr12b	947	1662	5820	4311	85155	20
chr12c	947	1662	5820	6187	80318	13
chr15a	1814	3270	11460	10533	218060	21
chr15b	1814	3270	11460	9574	218023	23
chr15c	1814	3270	11460	9979	219901	22
chr18a	3095	5679	19908	19559	531166	27
chr18b	3095	5679	19908	9139	527294	58
chr20a	4219	7810	27380	38477	885955	23
chr20b	4219	7810	27380	63243	893674	14
chr20c	4219	7810	27380	23802	926034	39
chr22a	5587	10417	36520	33685	1392239	41
chr22b	5587	10417	36520	38489	1382161	36
chr25a	8148	15325	53725	49605	2555662	52
fit1p	628	1677	10894	5002	196251	39
fit2p	3001	13525	60784	34303	4498500	131
fome10	6071	12230	35632	114338	1610864	145
fome11	14695	24460	71264	237844	3221728	14
fome12	24285	48920	167492	445156	6443456	15
pds-06	9882	28655	82269	22020	580116	26
pds-10	16559	48763	140063	37123	1626987	44
pds-20	33875	105728	304153	77352	6960089	90
route	20894	23923	187686	14876	3078015	207
scr10	689	1540	5940	13653	124559	9
scr12	1151	2784	10716	20437	330483	16
scr15	2234	6210	24060	77680	125514	2
scr20	5079	15980	61780	446686	6561431	15

Table 1: Comparing the number of nonzero elements in the LU factorisation of the basis B and in the Cholesky factorisation of the normal equations matrix $A\Theta A^T$.

direct approach by forming the Cholesky factorisation LDL^T for the normal equations matrix. As the interior point method approaches optimality, the normal equations matrix becomes extremely ill-conditioned due to the value of the entries in Θ. At this point, we switch to the iterative solver. In practice, we switch to PCG when two conditions are satisfied: firstly, there are enough small elements in Θ^{-1} (we have at least $3m/4$ small entries θ_j^{-1}). Secondly, the relative duality gap is less than or equal to 10^{-2}.

Initially, the termination criterion for the PCG method (17) is set as $\|r_k\|/\|r^{(0)}\| < 10^{-2}$. When the relative duality gap becomes less than or equal to 10^{-3} the termination criterion is changed to $\|r_k\|/\|r^{(0)}\| < 10^{-3}$ and, finally, when the relative duality gap falls below 10^{-4} the termination criterion becomes $\|r_k\|/\|r^{(0)}\| < 10^{-4}$.

In Table 1, we report the problem sizes: m, n and $nz(A)$ denote the number of rows, columns

and nonzeros in the constraint matrix A. In the next two columns, $nz(B)$ denotes the number of nonzeros in the LU factorisation of the basis matrix B and $nz(L)$ denotes the number of nonzero elements in the Cholesky factor of the normal equations matrix. We report results for problems which benefit from the use of the iterative approach presented in this paper. As shown in the last column of Table 1, the iterative method is storage-efficient, requiring one or two orders of magnitude less storage than the Cholesky factorisation. If the PCG approach were used for all IPM iterations, this memory advantage would allow certain problems to be solved for which the memory requirement of Cholesky would be prohibitive. In addition, it is essential that the LU factors are smaller by a significant factor since they will have to be applied twice for each PCG iteration when solving for the Newton direction, whereas the direct method using Cholesky factors requires the L factor to be used just twice to compute the Newton direction. The relative memory requirement can also be viewed as a measure of the maximum number of PCG iterations that can be performed while remaining competitive with the direct method using Cholesky factors.

Problem	Direct approach		Mixed approach			Ratio
	Time	IPM-iters	Time	IPM-iters	IPM-pcg	
aircraft	33.15	17	24.94	17	5	1.33
chr12a	0.304	14	0.290	14	2	1.05
chr12b	0.402	16	0.354	16	3	1.14
chr12c	0.256	11	0.254	11	1	1.01
chr15a	1.274	17	1.316	22	9	0.97
chr15b	1.263	17	1.196	17	2	1.06
chr15c	1.231	17	1.194	17	2	1.03
chr18a	6.480	29	5.747	30	5	1.23
chr18b	3.520	16	3.213	16	3	1.10
chr20a	13.69	28	9.292	28	14	1.47
chr20b	11.31	27	9.895	27	8	1.22
chr20c	11.91	23	11.76	23	4	1.01
chr22a	25.59	28	24.73	28	2	1.03
chr22b	48.78	52	27.09	33	2	1.80
chr25a	81.04	39	71.92	39	5	1.13
fit1p	3.49	20	2.01	20	9	1.74
fit2p	583.33	25	211.93	25	12	2.75
fome10	281.96	45	124.01	43	17	2.27
fome11	827.85	48	288.44	44	17	2.87
fome12	1646.29	48	604.98	44	17	2.72
pds-06	60.81	44	28.12	43	21	2.16
pds-10	198.08	38	103.34	53	29	1.92
pds-20	2004.87	47	770.83	66	38	2.60
route	53.98	25	48.99	24	4	1.10
scr10	0.839	19	0.685	19	8	1.22
scr12	3.092	14	2.951	14	2	1.05
scr15	50.79	26	41.22	26	7	1.23
scr20	614.56	25	517.62	26	4	1.19

Table 2: Solution statistics.

The results of comparing our mixed approach against the direct approach are given in Table 2. In all reported runs we have asked for eight digits of accuracy in the solution. For each test problem we report the number of interior point iterations and the total CPU time in seconds needed to solve the problem. Additionally, for the mixed approach we also report the number of interior point iterations in which preconditioned conjugate gradients method was used (IPM-pcg). For the problem `fit2p`, for example, 12 of the 25 interior point iterations used the iterative solution method: the remaining 13 iterations used the direct method. In the last column of Table 2 we report the ratio of solution time for the two methods. If this ratio is greater than one the mixed approach is faster than the pure direct one. For the problem `fit2p`, for example, this ratio is equal to 2.75 which implies that the mixed approach is 64% faster than the pure direct approach.

As we report in the column headed "Mixed approach" of Table 2, we use the PCG method only in the final iterations of the interior point method, while the rest of the interior point iterations are made using the direct method. For most problems, the numbers of IPM iterations required when using the direct and mixed approaches to solve a given problem are the same or differ only slightly. However, for `chr15a`, `pds-10` and `pds-20`, the mixed approach requires more iterations, significantly so in the case of the latter two problems. In the case of `chr15a` this accounts for the only ratio below 1 in Table 2. For one problem, `chr22b`, using the mixed approach leads to significantly fewer IPM iterations being required.

In order to give an insight into the behaviour of the preconditioned conjugate gradients, in Table 3 we report the number of PCG iterations needed to solve a particular linear system. First, we report separately this number for the last interior point iteration when our preconditioner is supposed to behave best. The following three columns correspond to the minimum, the average, and the maximum number of PCG iterations encountered throughout all iterative solves.

Finally, in Table 4 we report results for most of the problems solved with the pure iterative method. In these runs we have ignored the spread of elements in the diagonal matrix Θ and the distance to optimality, and we have forced the use of the PCG method in all interior point iterations. Such an approach comes with a risk of failure of the PCG method because the preconditioner does not have all its attractive properties in the earlier iterations. Indeed, we would not advise its use in the general context. However, for several problems in our collection such an approach has been very successful.

8 Conclusions

We have proposed in this paper a new sparse preconditioner for the (indefinite) augmented system arising in interior point algorithms for linear programming. This preconditioner takes advantage of the fact that a subset of elements in the matrix Θ^{-1} converge to zero as the solution of the linear program is approached. We replace these elements with zeros in the preconditioner. As a result, we have obtained a sparse and easily invertible block-triangular matrix. The constraint matrix A has been partitioned into $[B,\ N]$, where B is an m by m nonsingular matrix. The matrix B is obtained from m linearly independent columns of A which correspond to small θ_j^{-1}. By following the analysis of Rozložník and Simoncini [20] closely, we have shown that the PCG method can be applied to a non-symmetric indefinite matrix for a specific starting point. In addition, we have analysed the behaviour of the error term.

Problem	PCG Iterations			
	lastIPM	min	average	max
aircraft	10	8	9	10
chr12a	19	18	20	23
chr12b	29	28	29	29
chr12c	26	26	26	26
chr15a	37	37	38	41
chr15b	33	31	38	36
chr15c	32	31	32	32
chr18a	37	35	37	38
chr18b	57	53	56	57
chr20a	39	38	56	82
chr20b	32	32	63	104
chr20c	45	42	44	45
chr22a	48	46	49	53
chr22b	45	39	42	46
chr25a	51	46	50	55
fit1p	2	2	3	6
fit2p	4	3	15	43
fome10	142	129	243	519
fome11	169	123	205	494
fome12	111	111	210	500
pds-06	60	36	53	71
pds-10	66	45	60	86
pds-20	111	44	78	145
route	85	30	60	92
scr10	19	16	19	23
scr12	44	44	45	45
scr15	43	43	61	78
scr20	200	141	181	291

Table 3: The number of PCG iterations during the interior point method iterations.

This analysis reveals that, although we work with the indefinite system preconditioned with the indefinite matrix, the error converges to zero and, asymptotically, behaves in a similar way to the classical case when PCG is applied to a positive definite system. Finally, we have illustrated the feasibility of our approach on a set of medium to large-scale linear problems.

Based on these results we conclude that it is advantageous to apply the preconditioned conjugate gradient method to indefinite KKT systems arising in interior point algorithms for linear programming.

The use of an iterative method in this context makes an essential difference in the implementation of the interior point algorithm. This requires a better understanding of IPM convergence properties in a situation when directions are inexact. A study of these properties will be the subject of our research in the near future.

Problem	Direct approach		Pure iterative approach		Ratio
	Time	IPM-iters	Time	IPM-iters	
aircraft	33.15	17	2.87	15	11.55
chr12b	0.402	16	0.306	14	1.31
chr15a	1.274	17	0.856	16	1.49
chr15b	1.263	17	0.944	16	1.34
chr15c	1.231	17	0.959	18	1.28
chr18a	6.480	29	3.119	29	2.08
chr18b	3.520	16	2.255	18	1.56
chr20a	13.69	28	5.721	34	2.39
chr20b	11.31	27	5.721	30	1.98
chr20c	11.91	23	4.800	22	2.48
chr22a	25.59	28	6.725	31	3.81
chr22b	48.78	52	8.232	36	5.93
chr25a	81.04	39	17.54	41	4.62
fit1p	3.49	20	0.38	19	9.18
fit2p	583.33	25	19.09	26	30.56
scr10	0.839	19	0.633	19	1.33
scr12	3.092	14	1.701	15	1.82
scr15	50.79	26	16.55	26	3.07

Table 4: Efficiency of the pure iterative method.

Acknowledgements
We are grateful to the anonymous referee for constructive comments that have resulted in an improved presentation. We have also received valuable advice from Miro Rozloznik and Nick Gould.

References

[1] A. Altman and J. Gondzio, *Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization*, Optimization Methods and Software, 11-12 (1999), pp. 275–302.

[2] E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu, *Implementation of interior point methods for large scale linear programming*, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, 1996, pp. 189–252.

[3] M. Arioli, I. S. Duff, and P. P. M. de Rijk, *On the augmented system approach to sparse least-squares problems*, Numerische Mathematik, 55 (1989), pp. 667–684.

[4] V. Baryamureeba, *Solution of large-scale weighted least-squares problems*, Numerical Linear Algebra with Applications, 9 (2002), pp. 93–106.

[5] V. Baryamureeba, T. Steihaug, and Y. Zhang, *A class of preconditioners for weighted least squares problems*, Tech. Rep. 170, Department of Informatics, University of Bergen, Norway, 1999.

[6] J. R. Bunch and B. N. Parlett, *Direct methods for solving symemtric indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 8 (1971), pp. 639–655.

[7] W. J. Carolan, J. E. Hill, J. L. Kennington, S. Niemi, and S. J. Wichmann, *An empirical evaluation of the KORBX algorithms for military airlift applications*, Operations Research, 38 (1990), pp. 240–248.

[8] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Oxford University Press, New York, 1987.

[9] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis Dundee 1975, G. Watson, ed., Springer-Verlag, Berlin, New York, 1976, pp. 73–89.

[10] D. M. Gay, *Electronic mail distribution of linear programming test problems*, Mathematical Programming Society COAL Newsletter, 13 (1985), pp. 10–12.

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

[12] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic problems arising in optimization*, SIAM Journal on Scientific Computing, 23 (2001), pp. 1375–1394.

[13] J. A. J. Hall and K. I. M. Mckinnon, *Hyper-sparsity in the revised simplex method and how to exploit it*, Computational Optimization and Applications, 32 (2005), pp. 259–283.

[14] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand, 49 (1952), pp. 409–436.

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

[16] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, vol. 16 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[17] L. Lukšan and J. Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems*, Numerical Linear Algebra with Applications, 5 (1998), pp. 219–247.

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

[19] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

[20] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM Journal of Matrix Analysis and Applications, 24 (2002), pp. 368–391.

[21] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2nd ed., 2003.

[22] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[23] J. A. Tomlin, *Pivoting for size and sparsity in linear programming inversion routes*, Journal of Mathematics and Applications, 10 (1972), pp. 289–295.

[24] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, Cambridge, 2003.

[25] R. J. Vanderbei, *Symmetric quasidefinite matrices*, SIAM Journal on Optimization, 5 (1995), pp. 100–113.

[26] W. Wang and D. P. O'Leary, *Adaptive use of iterative methods in predictor-corrector interior point methods for linear programming*, Numerical Algorithms, 25 (2000), pp. 387–406.

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