February 10, 1997

Extending LSQR to generalised least-squares and Schur complement problems without resorting to Cholesky decompositions

Steve Benbow *

School of Mathematics,

University of Bath, Claverton Down, Bath, BA2 7AY

E-mail: `sjb@maths.bath.ac.uk`

An iterative method for solving problems of the form

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

in a generalised least-squares sense is given, where the $n \times n$ symmetric positive-definite matrix A and the $n \times m$ full column rank matrix B are large and sparse. The method, LSQR(A^{-1}) , is shown to be a natural extension of the LSQR algorithm of Paige and Saunders, with generalised orthogonality properties so that the Cholesky factor of A is not required. Instead it is only assumed that some method of calculating the effect of A^{-1} on a vector is available.

Numerical experiments comparing LSQR(A^{-1}) with similar preconditioned Krylov methods are described which demonstrate that LSQR(A^{-1}) exhibits superior numerical properties when the Schur complement matrix, $B^T A^{-1} B$, is ill-conditioned.

Subject classification: AMS(MOS) 65F10, 65F20, 65F50

Keywords: linear least-squares problem, generalised least-squares problem, Schur complement, Krylov subspaces, iterative methods, large sparse systems, LSQR, MINRES, GMRES

1 Introduction

Linear systems of the form

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{1.1}$$

*The author's research was supported by an Engineering and Physical Sciences Research Council of the United Kingdom CASE studentship at Bath University and AEA Technology, Harwell.

arise frequently in the numerical solution of problems in applied mathematics. The matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ (typically $n \geq m$) are assumed to be sparse, and further it is assumed that the matrix A is symmetric positive-definite and B is of full column rank. Systems of this type occur in many applications, for example in optimisation and computational fluid dynamics.

One way in which a system of the form (1.1) can occur in optimisation is in the constrained minimisation

$$\text{minimise } \frac{1}{2}x^T Ax - x^T b \text{ subject to } B^T x = 0,$$

the variables y in (1.1) being the Lagrange multipliers in the associated saddle point problem. In optimisation applications the matrix A is often diagonal, most obviously in the context of weighted least-squares problems, and also when solving linear programming problems by interior point methods [7] where systems of the form (1.1) arise at each stage with the diagonal matrix A changing and becoming increasingly badly scaled.

Two classic examples from computational fluid dynamics in which systems of the form (1.1) can arise are Darcy's law for incompressible flow,

$$\frac{\mu}{k}u + \nabla p = 0, \quad \nabla \cdot u = 0,$$

and the steady Stokes' equations for incompressible flow,

$$-\mu \triangle u + \nabla p = f, \quad \nabla \cdot u = 0,$$

both in 2 or $3D$ together with some appropriate boundary conditions. Here the variables x of (1.1) represent the discretised velocity u and y represent the discretised pressure p. The constant μ is the fluid viscosity and k is a discontinuous permeability function. In the case of Darcy's law the equations are typically discretised using finite difference (F.D.) or finite element (F.E.) methods. The matrix A representing the discretisation of the μ/k operator can be diagonal for appropriate F.D. discretisations or typically tridiagonal in $2D$ (pentadiagonal in $3D$) for appropriate F.E. bases, and is often badly scaled because of the discontinuous permeability function. It is possible to reformulate Darcy's law as an elliptic boundary value problem by eliminating the velocity term. In [15] it is argued that finite element discretisations of such elliptic boundary value problems are possible which give rise to diagonal A by determining the gradients of p by interpolation of the nodal values. The Stokes equations are often solved using F.E. discretisations where, for example, the velocity bases are quadratic and the pressure bases constant so as to satisfy the Babuska-Brezzi constraints (see [1] for more details). The higher dimension of the Stokes velocity-basis functions causes a greater level of fill-in in the matrix A and hence makes the efficient use of A^{-1} less practical.

It can be seen (§2) that the y component of the solution of (1.1) is the solution of the *generalised least-squares* (see for example [6]) problem

$$\textit{minimise}\ _{y\in\mathbb{R}^m} ||By-b||_{A^{-1}}\,. \tag{1.2}$$

The LSQR algorithm [12] introduced in §3 is a method for solving least-squares problems of the form $\textit{minimise}\ _{y\in\mathbb{R}^m} ||Cy-c||$ and hence could be used naïvely to solve the generalised least-squares problem above, if the Cholesky factorisation $A = LL^T$ were known, upon setting $C = L^{-1}B$ and $c = L^{-1}b$. This approach would require two operations involving L^{-1} per iteration step. In §4 it is shown that a variation of the LSQR algorithm exists which can be applied to the generalised least-squares problem and which doesn't require knowledge of the Cholesky factor of A, rather it only assumes that some method of calculating the action of A^{-1} exists. Further only one application of A^{-1} is needed per iteration step. This is favourable in the case where the level of fill-in in the Cholesky factor makes its use impractical, for example when A is an 'arrowhead' matrix L can contain non-zero terms in each of its lower diagonal entries and so operations involving the Cholesky factor will be expensive whereas an iterative method which calculates the effect of A^{-1} using only operations involving A can be relatively inexpensive.

Assuming that it is possible to calculate the effect of A^{-1} is equivalent to assuming that (1.1) can be preconditioned by an augmented matrix containing A. This assumption is made in [3] where the authors conclude that if it is feasible to operate with A^{-1} then the most efficient way to solve (1.1) using a minimum residual approach (see for example [11, 13]) is to solve the Schur complement system $B^TA^{-1}By = B^TA^{-1}b$ using the MINRES algorithm [11] (and then recover x). In §5 this approach is compared with the generalised least-squares approach outlined above, and it is noted that since the Schur complement system is simply the normal equations for the generalised least-squares problem it is more favourable to solve (1.1) using a least-squares approach, as the worse condition of the Schur complement matrix than that of the coefficient matrix in the generalised least-squares problem can lead to poor numerical performance.

2 The generalised least-squares connection

Notice that x can be eliminated from (1.1) to form an equation in y only, namely

$$B^TA^{-1}By = B^TA^{-1}b. \tag{2.1}$$

The matrix $B^TA^{-1}B$ is often referred to as the Schur complement matrix. If y solves (2.1) the solution $(x^T\ \ y^T)^T$ of (1.1) can be recovered using $x = A^{-1}(b-By)$. From (2.1),

$$p^TB^TA^{-1}By = p^TB^TA^{-1}b \quad \forall p\in\mathbb{R}^m,$$

so that, on defining the inner-product $\langle \cdot\ ,\ \cdot \rangle$ on $\mathbb{R}^n$ by $\langle u\ ,\ v \rangle = u^T A^{-1} v$, (2.1) can be reformulated as

$$\textit{Find } y \in \mathbb{R}^m \textit{ such that } \quad \langle Bp\ ,\ By \rangle = \langle Bp\ ,\ b \rangle \quad \forall p \in \mathbb{R}^m. \tag{2.2}$$

Setting $r = b - By$ implies that if y solves (2.2), $\langle Bp\ ,\ r \rangle = 0 \quad \forall p \in \mathbb{R}^m$, so that the residual r is orthogonal to the span of the columns of B, denoted $sp(B)$, with respect to $\langle \cdot\ ,\ \cdot \rangle$. This condition implies that By is the closest representation of b in $sp(B)$ with respect to the norm induced by $\langle \cdot\ ,\ \cdot \rangle$, i.e. that y is determined by the generalised least-squares problem,

$$\textit{Find } y \in \mathbb{R}^m \textit{ such that } \min_{p \in \mathbb{R}^m} ||b - Bp||_{A^{-1}} = ||b - By||_{A^{-1}}\ , \tag{2.3}$$

where $||u||_{A^{-1}} = \langle u\ ,\ u \rangle^{\frac{1}{2}} = (x^T A^{-1} x)^{\frac{1}{2}}$. The reverse argument, that (2.3) is equivalent to (2.1), can be seen by minimising the functional $F(p) = \frac{1}{2} ||b - Bp||^2_{A^{-1}}$.

3 Golub-Kahan bidiagonalisation and LSQR

As is the case for many iterative solution methods for symmetric systems (for example CG [5], SYMMLQ and MINRES [11]), the LSQR algorithm [12] for the solution of least-squares problems can be seen to have its roots buried in the Lanczos process [8] for reducing a symmetric matrix to tridiagonal form. In §3.1 the Lanczos process is reintroduced and in §3.2 it is explained how Golub and Kahan used the Lanczos process to reduce a given matrix to bidiagonal form. §3.3 then describes how Paige and Saunders extended this bidiagonalisation process to an iterative solution method for least-squares problems.

3.1 The Lanczos process

Given a symmetric matrix M and a starting vector b, the Lanczos process [8] reduces M to tridiagonal form as follows.

The Lanczos process

Set $\hat{\beta}_1 \hat{v}_1 = b$,

For $i = 1, 2, \ldots$

$$\begin{aligned} w_i &= M\hat{v}_i - \hat{\beta}_i \hat{v}_{i-1}, \\ \hat{\alpha}_i &= \hat{v}_i^T w_i, \\ \hat{\beta}_{i+1}\hat{v}_{i+1} &= w_i - \hat{\alpha}_i \hat{v}_i, \end{aligned}$$

where $\hat{v}_0 := 0$ and $\hat{\beta}_i \geq 0$ is chosen so that $||\hat{v}_i|| = 1, \ \ i = 1, 2, \ldots$. The recurrences of the Lanczos process can be written more succinctly as

$$M\hat{V}_k = \hat{V}_k T_k + \hat{\beta}_{k+1}\hat{v}_{k+1}e_k^T, \tag{3.1}$$

where $\hat{V}_k = [\hat{v}_1, \ldots, \hat{v}_k]$ and

$$T_k = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_2 & & & \\ \hat{\beta}_2 & \hat{\alpha}_2 & \hat{\beta}_3 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \hat{\beta}_k \\ & & & \hat{\beta}_k & \hat{\alpha}_k \end{bmatrix}.$$

The set $\{\hat{v}_i\}_{i=1}^k$ can be seen to be constructed so that $\hat{v}_i^T\hat{v}_j = \delta_{ij}$ (in the absence of rounding error) so that $\hat{V}_k^T\hat{V}_k = I_k$.

As is mentioned above, many iterative solution methods for the system

$$Mz = b \tag{3.2}$$

can be derived from the Lanczos process. If $\hat{p}_k$ is the solution of the system

$$T_k\hat{p}_k = \hat{\beta}_1 e_1, \tag{3.3}$$

and if $z_k := \hat{V}_k\hat{p}_k$ then (3.1) gives

$$Mz_k = b + \hat{\beta}_{k+1}\hat{\eta}_k\hat{v}_{k+1} \tag{3.4}$$

where η_k is the last entry in $\hat{p}_k$. Hence z_k is the exact solution of (3.4) and will solve (3.2) whenever $\hat{\beta}_{k+1}\hat{\eta}_k$ is negligible.

Different solution methods for (3.3) give rise to various solution methods for (3.2). For example when M (and therefore T_k) is positive definite, using the Cholesky decomposition to solve (3.3) is equivalent to the conjugate gradient method, whilst the use of orthogonal decompositions of T_k are the basis of the SYMMLQ and MINRES algorithms. The way in which different decompositions of T_k can be used to construct these iterative methods is discussed in detail in [11].

3.2 Golub-Kahan bidiagonalisation

The singular value decomposition of a matrix B is $B = U\Sigma V^T$ where Σ is a rectangular, diagonal matrix having the same dimensions as B with non-negative entries, and U, V are unitary matrices of appropriate size. In [4] Golub and Kahan suggest an iterative procedure for generating a bidiagonal matrix, E, having the same singular values as B. The matrix E can be seen to be the result of applying the Lanczos process to the matrix

$$\mathcal{A}_{LS} = \begin{bmatrix} I & B \\ B^T & \end{bmatrix}$$

as follows.

If the Lanczos process is applied to $\mathcal{A}_{LS}$ with starting vector $(b^T, 0^T)^T$ it can be seen that the orthonormal vectors $\hat{v}_i$ are of the form

$$\hat{v}_i = \begin{cases} \begin{bmatrix} u_j \\ 0 \end{bmatrix} & i = 2j-1, \quad j \in \mathbb{N} \\ \begin{bmatrix} 0 \\ v_j \end{bmatrix} & i = 2j, \quad j \in \mathbb{N} \end{cases}$$

and that

$$\hat{\alpha}_i = \begin{cases} 1 & i = 2j-1, \quad j \in \mathbb{N} \\ 0 & i = 2j, \quad j \in \mathbb{N} \end{cases}.$$

Notice that since $\{\hat{v}_i\}_{i=1}^{k}$ is an orthonormal set, both $\{u_j\}_{j=1}^{[\frac{k+1}{2}]}$ and $\{v_j\}_{j=1}^{[\frac{k}{2}]}$ are orthonormal sets. It can also be verified that $\{u_j\}_{j=1}^{l}$ spans the Krylov space $\mathcal{K}^l(BB^T, b)$ whilst $\{v_j\}_{j=1}^{l}$ spans the space $\mathcal{K}^l(B^TB, B^Tb)$. Renaming the $\hat{\beta}_i$'s as

$$\hat{\beta}_i = \begin{cases} \beta_j & i = 2j-1, \quad j \in \mathbb{N} \\ \alpha_j & i = 2j, \quad j \in \mathbb{N} \end{cases}$$

allows the Lanczos process to be expressed in the following form, to be referred to as the Golub-Kahan lower-bidiagonalisation procedure (GKLB).

The Golub-Kahan lower-bidiagonalisation procedure (GKLB)

$$\begin{aligned} &\text{Set } \beta_1 u_1 = b, \quad \alpha_1 v_1 = B^T u_1, \\ &\text{For } j = 1, 2, \ldots \\ &\quad \beta_{j+1} u_{j+1} = B v_j - \alpha_j u_j, \\ &\quad \alpha_{j+1} v_{j+1} = B^T u_{j+1} - \beta_{j+1} v_j, \end{aligned}$$

where $\alpha_j, \beta_j > 0$ are chosen so that $||u_j|| = ||v_j|| = 1$ respectively. Notice that one step of GKLB is equivalent to performing two Lanczos steps on $\mathcal{A}_{LS}$ but requires only two multiplications involving B as opposed to four. Defining $U_l = [u_1, \ldots, u_l]$, $V_l = [v_1, \ldots, v_l]$ and

$$E_l = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \cdot & \cdot & \\ & & \cdot & \alpha_l \\ & & & \beta_{l+1} \end{bmatrix}$$

allows the recurrence relations of GKLB to be expressed as

$$\begin{aligned} U_{l+1}(\beta_1 e_1) &= b, \\ BV_l &= U_{l+1}E_l, \\ B^T U_{l+1} &= V_l E_l^T + \alpha_{l+1} v_{l+1} e_{l+1}^T. \end{aligned}$$

The algorithm GKLB owes the special lower-bidiagonal form of the matrix E_l to the starting vector $(b^T, 0^T)^T$. In fact if the starting vector $(0^T, B^T b)^T$ was instead used to start the Lanczos process for $\mathcal{A}_{LS}$ a similar algorithm to GKLB would result which reduces B to *upper*-bidiagonal form. This algorithm will be referred to as Golub-Kahan upper-bidiagonalisation (GKUB), the detail of its derivation is omitted since it is essentially the same as that above. The recurrences of GKUB are as follows.

The Golub-Kahan upper-bidiagonalisation procedure (GKUB)

Set $\theta_1 v_1 = B^T b, \quad \rho_1 p_1 = B^T v_1,$
For $j = 1, 2, \ldots$

$$\begin{aligned} \theta_{j+1} v_{j+1} &= B^T p_j - \rho j v_j, \\ \rho_{j+1} p_{j+1} &= B v_{j+1} - \theta_{j+1} p_j, \end{aligned}$$

with $\theta_j, \rho_j \geq 0$ chosen so that $||v_j|| = ||p_j|| = 1$. If $P_l = [p_1, \ldots, p_l]$ and

$$R_l = \begin{bmatrix} \rho_1 & \theta_2 & & & \\ & \rho_2 & \theta_3 & & \\ & & \cdot & \cdot & \\ & & & \cdot & \theta_l \\ & & & & \rho_l \end{bmatrix}$$

then

$$\begin{aligned} V_l(\theta_1 e_1) &= B^T b, \\ BV_l &= P_l R_l, \\ B^T P_l &= V_l R_l^T + \theta_{l+1} v_{l+1} e_l^T, \end{aligned}$$

hold and $P_l^T P_l = I_l$.

There is an important connection between the matrices E_l and R_l. Using the orthogonality properties of GKLB and GKUB it is simple to deduce that

$$E_l^T E_l = R_l^T R_l.$$

Now suppose that $Q_l E_l = \begin{bmatrix} \hat{R}_l \\ 0^T \end{bmatrix}$ is the QR decomposition of E_l where $\hat{R}_l$ has positive diagonal entries (so that $\hat{R}_l$ is unique [14]). Then $\hat{R}_l^T \hat{R}_l = R_l^T R_l$. Further since the Cholesky decomposition of a positive definite matrix is unique upon insisting that the diagonal entries of the Cholesky factor are positive [14], it follows that $\hat{R}_l = R_l$. Hence the the matrix R_l of GKUB applied to $\mathcal{A}_{LS}$ is the upper-bidiagonal matrix obtained in the QR decomposition of E_l. It will be shown in §3.3 that the entries of R_l can be obtained trivially from the entries of E_l without requiring GKUB to be carried out.

3.3 The LSQR algorithm

The solution of the system

$$\begin{bmatrix} I & B \\ B^T & \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{3.5}$$

is now considered. Since the Lanczos vectors satisfy $\hat{V}_k^T M \hat{V}_k = T_k$, the GKLB vectors have the property that

$$\begin{bmatrix} u_1^T & 0^T \\ 0^T & v_1^T \\ u_2^T & 0^T \\ 0^T & v_2^T \\ \cdot & \cdot \\ \cdot & \cdot \\ u_{l+1}^T & 0^T \end{bmatrix} \begin{bmatrix} I & B \\ B^T & \end{bmatrix} \begin{bmatrix} u_1 & 0 & \dots & u_{l+1} \\ 0 & v_1 & \dots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \alpha_1 & & & & & \\ \alpha_1 & 0 & \beta_2 & & & & \\ & \beta_2 & 1 & \alpha_2 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \alpha_l & 0 & \beta_{l+1} \\ & & & & & \beta_{l+1} & 1 \end{bmatrix}$$

and by permuting rows and columns the above can be rewritten in the form

$$\begin{bmatrix} U_{l+1}^T & \\ & V_l^T \end{bmatrix} \begin{bmatrix} I & B \\ B^T & \end{bmatrix} \begin{bmatrix} U_{l+1} & \\ & V_l \end{bmatrix} = \begin{bmatrix} I & E_l \\ E_l^T & \end{bmatrix},$$

so that the Lanczos subproblem (3.3) corresponds to

$$\begin{bmatrix} I & E_l \\ E_l^T & \end{bmatrix} \begin{bmatrix} s_{l+1} \\ t_l \end{bmatrix} = \beta_1 \begin{bmatrix} e_1 \\ 0 \end{bmatrix} \tag{3.6}$$

where $(s_{l+1}^T, t_l^T)^T$ is the permutation of p_{2l+1}. Hence t_l is the solution of the least-squares problem

$$\text{minimise } \|E_l t - \beta_1 e_1\|. \tag{3.7}$$

Notice that once t_l is known, $s_{l+1} = \beta_1 e_1 - E_l t_l$ can be recovered and then estimates to the solution of (3.5) are given by $y_l = V_l t_l$ and $x_l = U_{l+1} s_{l+1}$. The connection between the matrix R_l of GKUB and the QR decomposition of E_l provided Paige and Saunders with an attractive method for the solution of (3.7). First define $f_l = (\varphi_1, \ldots, \varphi_l)^T$ and $\overline{\varphi}_{l+1}$ by

$$Q_l(\beta_1 e_1) = \begin{bmatrix} f_l \\ \overline{\varphi}_{l+1} \end{bmatrix}, \tag{3.8}$$

where Q_l is the orthonormal matrix in the QR decomposition of E_l. Then

$$\begin{aligned} \|E_l t - \beta_1 e_1\| &= \|Q_l(E_l t - \beta_1 e_1)\| \\ &= \left\| \begin{bmatrix} R_l \\ 0^T \end{bmatrix} t - \begin{bmatrix} f_l \\ \overline{\varphi}_{l+1} \end{bmatrix} \right\| \end{aligned}$$

and so upon noticing $f_l \in sp(R_l)$, t_l is defined by

$$R_l t_l = f_l,$$

and it is trivial to show that

$$s_{l+1} = Q_l^T \begin{bmatrix} 0 \\ \overline{\varphi}_{l+1} \end{bmatrix}.$$

Since t_l changes elementwise it would appear that $y_l = V_l t_l$ needs to be recalculated at every step. This is not desirable, it is more convenient to rewrite the above method in such a way that solution estimates can be updated. To this end notice that

$$[R_l \;\; f_l] = \begin{bmatrix} R_{l-1} & \vdots & f_{l-1} \\ \cdots & \cdot & \cdot \end{bmatrix}$$

where the dots indicate the inclusion of a new row and column, so that

$$y_l = V_l t_l = (V_l R_l^{-1}) f_l =: D_l f_l,$$

where $D_l = [d_1, \ldots, d_l]$. Defining $d_0 := 0$, the rule

$$d_l = \frac{1}{\rho_l}(v_l - \theta_l d_{l-1}),$$

is obtained for developing D_l, whence if $y_0 := 0$,

$$y_l = y_{l-1} + \varphi_l d_l$$

is the update rule for solution approximations.

The final observation before the LSQR algorithm of Paige and Saunders can be presented is the following. The matrix Q_l in the QR decomposition of E_l can be expressed as a product of plane rotations, $Q_l = Q_{l,l+1} \dots Q_{2,3}Q_{1,2}$, where each $Q_{j,j+1}$ operates on the transformed E_l to destroy the sub-diagonal β_{j+1} term. Hence c_l and s_l, the non-trivial entries in $Q_{l,l+1}$ must satisfy

$$\begin{bmatrix} c_l & s_l \\ s_l & -c_l \end{bmatrix} \begin{bmatrix} \overline{\rho}_l & \\ \beta_{l+1} & \alpha_{l+1} \end{bmatrix} = \begin{bmatrix} \rho_l & \theta_{l+1} \\ & \overline{\rho}_{l+1} \end{bmatrix} \tag{3.9}$$

where $\overline{\rho}_l = (R_l)_{ll}$. Notice also that (3.8) implies that

$$\begin{bmatrix} c_l & s_l \\ s_l & -c_l \end{bmatrix} \begin{bmatrix} \overline{\varphi}_l \\ 0 \end{bmatrix} = \begin{bmatrix} \varphi_l \\ \overline{\varphi}_{l+1} \end{bmatrix}. \tag{3.10}$$

Using (3.9) , (3.10) and the fact that the c_l, s_l matrix is unitary it is easy to verify that given $\alpha_{l+1}, \beta_{l+1}, \overline{\rho}_l$ and $\overline{\varphi}_l$, the new coefficients $\rho_l, c_l, s_l, \theta_{l+1}, \overline{\rho}_{l+1}, \varphi_l$ and $\overline{\varphi}_{l+1}$ can be calculated at trivial cost by the rules given in algorithm LSQR below.

The LSQR algorithm

Set $y_0 = 0$, $\beta_1 u_1 = b$, $\alpha_1 v_1 = B^T u_1$, $d_1 = v_1$, $\overline{\varphi}_1 = \beta 1$, $\overline{\rho}_1 = \alpha_1$

For j=1,2, ...

(Bidiagonalisation step)

$$\begin{aligned} \beta_{j+1} u_{j+1} &= B v_j - \alpha_j u_j \\ \alpha_{j+1} v_{j+1} &= B^T u_{j+1} - \beta_{j+1} v_j \end{aligned}$$

(Orthogonal transformation step)

$$\begin{aligned} \rho_j &= \left(\overline{\rho}_j^2 + \beta_{j+1}^2\right)^{\frac{1}{2}} \\ c_j &= \overline{\rho}_j / \rho j \\ s_j &= \beta_{j+1} / \rho_j \\ \theta_{j+1} &= s_j \alpha_{j+1} \\ \overline{\rho}_{j+1} &= -c_j \alpha_{j+1} \\ \varphi_j &= c_j \overline{\varphi}_j \\ \overline{\varphi}_{j+1} &= s_j \overline{\varphi}_j \end{aligned}$$

(Solution update)

$$\begin{aligned} y_j &= y_{j-1} + (\varphi_j / \rho_j) d_j \\ d_{j+1} &= v_{j+1} - (\theta_{j+1} / \rho_j) d_j \end{aligned}$$

After each solution update a convergence test needs to be carried out and the iteration terminated if some stopping criteria is satisfied. An analysis of suitable stopping criteria can be found in [12].

It should be noted that it is also possible to reduce the system (3.5) to a bidiagonal problem by using GKUB instead of GKLB. The resulting algorithm for the solution of (3.5) is then the LSCG algorithm of Paige [9]. This algorithm corresponds to solving a version of the normal equations associated with (3.5) and is therefore not as computationally attractive as the LSQR algorithm above.

4 Solution of generalised least-squares problems

In this section the bidiagonalisation procedure GKLB will be applied to a particular matrix associated with the generalised least-squares problem (2.3) to obtain a bidiagonalisation procedure for B with different orthogonality properties. In the same way that LSQR was constructed from GKLB, an iterative solution method for generalised least-squares problems will then be derived from the new bidiagonalisation procedure.

4.1 A bidiagonalisation procedure with a different inner-product

Recall from §2 that solving the system (1.1) is equivalent to solving the generalised least-squares problem (2.3),

$$\textit{Find } y \in \mathbb{R}^m \textit{ such that } \min_{p \in \mathbb{R}^m} ||b - Bp||_{A^{-1}} = ||b - By||_{A^{-1}} .$$

Typically to solve such problems it is first assumed that the Cholesky factorisation $A = LL^T$, where L is lower-triangular, is available. Then the problem (2.3) can be recast as the linear least-squares problem

$$\textit{Find } y \in \mathbb{R}^m \textit{ such that } \min_{p \in \mathbb{R}^m} \left|\left|L^{-1}(b - Bp)\right|\right| = \left|\left|L^{-1}(b - By)\right|\right| \tag{4.1}$$

which is then solved using any favourite method for linear least-squares. An example of solving (2.3) in this way is given in [10].

From §2 it is obvious that (4.1) is equivalent to the linear system

$$\begin{bmatrix} I & L^{-1}B \\ B^T L^{-T} & 0 \end{bmatrix} \begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} L^{-1}b \\ 0 \end{bmatrix} \tag{4.2}$$

and clearly LSQR could be the method used to solve (4.1). The solution $(x^T \;\; y^T)^T$ of (1.1) is then recovered by evaluating $z = L^{-1}(b - By)$, $x = L^{-T}z$. However there is another way in which an LSQR type method can be applied to solve (2.3) without using the Cholesky factor of A which sacrifices some of the Euclidean orthonormality properties of the LSQR vectors for a similar orthonormality condition.

Defining the vectors $w_i := Mu_i$, where the choice of the matrix M will be discussed shortly, the GKLB procedure can be applied to the matrix $L^{-1}B$ as follows

GKLB applied to $L^{-1}B$ with transformed u_j

$$\begin{array}{l} \text{Set } \beta_1 w_1 = ML^{-1}b, \quad \alpha_1 v_1 = B^T L^{-T} M^{-1} w_1, \\ \text{For } j = 1, 2, \ldots \\ \quad \begin{array}{rcl} \beta_{j+1} w_{j+1} &=& ML^{-1}Bv_j - \alpha_j w_j, \\ \alpha_{j+1} v_{j+1} &=& B^T L^{-T} M^{-1} w_{j+1} - \beta_{j+1} v_j, \end{array} \end{array}$$

where again $\alpha_j \geq 0$ is chosen so that $||v_j|| = 1$ and $\beta_j \geq 0$ is chosen such that $||w_j||_{(MM^T)^{-1}} = ||u_j|| = 1$. Since the vectors u_j form an orthonormal set it is true that

$$w_i^T \left(MM^T\right)^{-1} w_j = \delta_{ij}. \tag{4.3}$$

It is clear that judicious choices for M are $M \in \{L, L^{-T}\}$. Choosing $M = L$ will cancel all applications of L^{-1} and all occurrences of $L^{-T}M^{-1}$ become applications of A^{-1}, similarly choosing $M = L^{-T}$ cancels all L^{-T} operations and ML^{-1} becomes A^{-1}. In both cases the Cholesky factor L is completely removed from the picture. The lower-bidiagonalisation procedure resulting from the choice $M = L$ will be referred to as GKLB(A^{-1}) (since the vectors w_j are A^{-1}-orthonormal by (4.3)) and that resulting from the choice $M = L^{-T}$ will be referred to as GKLB(A) (since then the w_j are A-orthonormal). These two algorithms are presented below.

GKLB(A^{-1})

$$\begin{array}{l} \text{Set } \beta_1 w_1 = b, \quad \alpha_1 v_1 = B^T A^{-1} w_1, \\ \text{For } j = 1, 2, \ldots \\ \quad \begin{array}{rcl} \beta_{j+1} w_{j+1} &=& Bv_j - \alpha_j w_j, \\ \alpha_{j+1} v_{j+1} &=& B^T A^{-1} w_{j+1} - \beta_{j+1} v_j, \end{array} \end{array}$$

GKLB(A)

$$\begin{array}{l} \text{Set } \beta_1 w_1 = A^{-1}b, \quad \alpha_1 v_1 = B^T w_1, \\ \text{For } j = 1, 2, \ldots \\ \quad \begin{array}{rcl} \beta_{j+1} w_{j+1} &=& A^{-1}Bv_j - \alpha_j w_j, \\ \alpha_{j+1} v_{j+1} &=& B^T w_{j+1} - \beta_{j+1} v_j, \end{array} \end{array}$$

where in both algorithms $\alpha_j \geq 0$ is chosen so that $||v_j|| = 1$ and with $\beta_j \geq 0$ chosen so that $||w_j||_{A^{-1}} = 1$ in GKLB(A^{-1}) and $||w_j||_A = 1$ in GKLB(A) . The recurrences of GKLB(A^{-1}) are more nicely expressed as

$$\begin{aligned} W_{k+1}(\beta_1 e_1) &= b, \\ BV_k &= W_{k+1}E_k, \\ B^T A^{-1} W_{k+1} &= V_k E_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T. \end{aligned}$$

and those of GKLB(A) ,

$$\begin{aligned} W_{k+1}(\beta_1 e_1) &= A^{-1}b, \\ A^{-1}BV_k &= W_{k+1}E_k, \\ B^T W_{k+1} &= V_k E_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T. \end{aligned}$$

The procedure for carrying out the calculation of $\beta_{j+1}w_{j+1}$ in GKLB(A^{-1}) is

$$\begin{aligned} \hat{w_{j+1}} &= Bv_j - \alpha_j w_j \\ t &= A^{-1}\hat{w_{j+1}} \\ \beta_{j+1} &= (\hat{w_{j+1}}^T t)^{\frac{1}{2}} \\ w_{j+1} &= \hat{w_{j+1}}/\beta_{j+1} \\ A^{-1}w_{j+1} &= t/\beta_{j+1} \end{aligned}$$

so that $A^{-1}w_{j+1}$ is already known before it is required in the calculation of v_{j+1}. The calculation for GKLB(A) is similar.

Since both bidiagonalisation procedures are computationally equivalent only GKLB(A^{-1}) will be considered further, although all the results given are equally true of both methods. The last note on GKLB(A) is the following. At each iteration step one solve of the form $Af = g$ is required. Suppose that $f_k = W_k h_k$ is an approximation to f from $sp(W_k)$, then it may be hoped that $AW_k h_k \approx g$ (in some sense). If the matrix W_k is that obtained from GKLB(A) , $h_k = W_k^T g$ and so $f_k = (W_k W_k^T)g$. This observation has been seen to be of little use in practise.

4.2 Extension of the bidiagonalisation procedure to an iterative solution method

In exactly the same way that LSQR was derived from GKLB, GKLB(A^{-1}) forms the basis for an iterative solution method for generalised least-squares problems. Let $y_k = V_k z_k$ be an approximation to the solution of (2.3). The optimal z_k is then the solution of

$$\min_{z \in \mathbb{R}^k} ||b - BV_k z||_{A^{-1}} ,$$

and using the recurrences of GKLB(A^{-1}) ,

$$||b - BV_k z||_{A^{-1}} = ||b - W_{k+1}E_k z||_{A^{-1}} .$$

Defining $F(z) = \frac{1}{2}\left|\left|b - W_{k+1}E_k z\right|\right|^2_{A^{-1}}$ it can be seen that $\frac{d}{dz}F(z_k) = 0$ implies

$$E_k^T\left(E_k z_k - W_{k+1}^T A^{-1} b\right) = 0,$$

so that z_k is the least-squares solution of $\min_{z\in\mathbb{R}^k}\left|\left|E_k z - W_{k+1}^T A^{-1} b\right|\right|$. Further since $W_{k+1}^T A^{-1} b = \beta_1 e_1$, z_k is the least squares solution of

$$\min_{z\in\mathbb{R}^k}\left|\left|E_k z - \beta_1 e_1\right|\right|,$$

which is exactly the same subproblem which is solved by LSQR (c.f. (3.7)) so that only the bidiagonalisation step of LSQR needs to be swapped for GKLB(A^{-1}) in order to apply an LSQR-type algorithm to the generalised least squares problem (2.3). The algorithm described above will be referred to as LSQR(A^{-1}) and is given below for completeness.

The LSQR(A^{-1}) algorithm

Set $y_0 = 0$, $\beta_1 w_1 = b$, $\alpha_1 v_1 = B^T A^{-1} w_1$, $d_1 = v_1$, $\overline{\varphi}_1 = \beta_1$, $\overline{\rho}_1 = \alpha_1$

For j=1,2, ...

(Bidiagonalisation step)

$$\begin{aligned} \beta_{j+1} w_{j+1} &= B v_j - \alpha_j w_j \quad \text{with } \beta_{j+1} \text{ chosen so that } \left|\left|w_{j+1}\right|\right|_{A^{-1}} = 1 \\ \alpha_{j+1} v_{j+1} &= B^T A^{-1} w_{j+1} - \beta_{j+1} v_j \end{aligned}$$

(Orthogonal transformation step)

$$\begin{aligned} \rho_j &= \left(\overline{\rho}_j^2 + \beta_{j+1}^2\right)^{\frac{1}{2}} \\ c_j &= \overline{\rho}_j / \rho j \\ s_j &= \beta_{j+1} / \rho_j \\ \theta_{j+1} &= s_j \alpha_{j+1} \\ \overline{\rho}_{j+1} &= -c_j \alpha_{j+1} \\ \varphi_j &= c_j \overline{\varphi}_j \\ \overline{\varphi}_{j+1} &= s_j \overline{\varphi}_j \end{aligned}$$

(Solution update)

$$\begin{aligned} y_j &= y_{j-1} + (\varphi_j / \rho_j) d_j \\ d_{j+1} &= v_{j+1} - (\theta_{j+1} / \rho_j) d_j \end{aligned}$$

4.3 Preconditioning

When solving systems of the form

$$Mz = b,$$

where $M \in \mathbb{R}^{n\times n}$, it is usual to introduce a nonsingular preconditioning matrix N, or if N is positive definite its Cholesky factor L, and instead solve one of the systems

$$\begin{aligned} N^{-1}Mz_L &= N^{-1}b && \text{(left preconditioning)} \\ MN^{-1}z_R &= b && \text{(right preconditioning)} \\ L^{-1}ML^{-T}z_C &= L^{-1}b && \text{(centrally preconditioning)} \end{aligned}$$

where $z_L = z$, $z_R = Nz$ and $z_C = L^T z$; the idea in all cases being that the matrix $N^{-1}M$ has better condition than M.

It is clear that of these preconditioning strategies only right preconditioning is applicable to (generalised) least-squares problems since any other form of preconditioning affects the norm in which the minimisation is taking place. Hence preconditioned generalised least-squares problems,

$$\min \left|\left|BN^{-T}p - b\right|\right|_{A^{-1}} \tag{4.4}$$

can be considered when solving (1.1), where $p = N^T y$. Analogous to the extension from least-squares to generalised least-squares, an iterative method for (4.4) can be devised from the Golub-Kahan bidiagonalisation procedures by altering the inner product in which the m-vectors, $\{v_i\}$, are orthonormal. GKLB can be applied to the matrix BN^{-T} and if $z_i := Sv_i$, and $S = N$, is chosen with $H = NN^T$ the resulting algorithm, to be referred to as GKLB(A^{-1}, H^{-1}), is

GKLB(A^{-1}, H^{-1})

Set $\beta_1 w_1 = b, \quad \alpha_1 z_1 = B^T A^{-1} w_1,$

For $j = 1, 2, \ldots$

$$\begin{aligned} \beta_{j+1}w_{j+1} &= BH^{-1}z_j - \alpha_j w_j, \\ &\qquad (\beta_{j+1} \text{ chosen so that } ||w_{j+1}||_{A^{-1}} = 1) \\ \alpha_{j+1}z_{j+1} &= B^T A^{-1} w_{j+1} - \beta_{j+1} z_j \\ &\qquad (\alpha_{j+1} \text{ chosen so that } ||z_{j+1}||_{H^{-1}} = 1). \end{aligned}$$

The procedure for carrying out the calculation of $\alpha_{j+1}z_{j+1}$ is similar to that given for calculating $\beta_{j+1}w_{j+1}$ in GKLB(A^{-1}) .

A similar algorithm, GKLB(A^{-1}, H), to GKLB(A^{-1}, H^{-1}) can be obtained on taking $S = N^{-T}$ above. Similarly GKLB(A, H^{-1}) and GKLB(A, H) are also defined. Any of these bidiagonalisation procedures can be plugged into the bidiagonalisation step of LSQR(A^{-1}) to form an algorithm for the preconditioned system (4.4) the one which corresponds to GKLB(A^{-1}, H^{-1}) being denoted LSQR(A^{-1}, H^{-1}) e.t.c. .

5 Comparing LSQR(A^{-1}) with preconditioned Krylov methods

The main factor influencing whether the LSQR(A^{-1}) method is applicable to a particular system of form (1.1) is that the action of A^{-1} must be easy to perform. In effect this is equivalent to assuming that a specific preconditioning of (1.1) is possible, and hence it would make sense to compare the LSQR(A^{-1}) method with preconditioned Krylov methods, where the preconditioner contains the matrix A. In [3] preconditioners of the form

$$\begin{bmatrix} \frac{1}{\eta^2}A & 0 \\ 0 & \pm H \end{bmatrix}, \quad \eta \in \mathbb{R}^+, \tag{5.1}$$

are considered, where H is symmetric positive definite and is typically an approximation to the Schur complement matrix $B^T A^{-1} B$. The preconditioner (5.1) can be applied centrally to (1.1) to form the preconditioned system

$$\begin{bmatrix} \eta I & \hat{B} \\ \pm \hat{B}^T & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} \hat{b} \\ 0 \end{bmatrix}, \tag{5.2}$$

where $\hat{B} = L^{-1} B Q^{-T}$, $\hat{x} = \frac{1}{\eta} L^T x$, $\hat{y} = Q^T y$ and $\hat{b} = L^{-1} b$, here L and Q are the Cholesky factors of A and H respectively. The choice of $\pm H$ in (5.1) can impose different conditions on $\hat{\mathcal{A}}_\pm$, the coefficient matrix in (5.2). It is easy to see that choosing $+H$ ensures that $\hat{\mathcal{A}}_+$ is symmetric, whilst the identity

$$\begin{bmatrix} u^T & v^T \end{bmatrix} \hat{\mathcal{A}}_- \begin{bmatrix} u \\ v \end{bmatrix} = \eta u^T u > 0, \quad u \neq 0,$$

shows that choosing $-H$ gives rise to a positive semi-definite coefficient matrix. The dilemma of selecting symmetric or positive semi-definite preconditioners is addressed in [3] and some elegant results are presented, the most relevant being that if MINRES (§3.1 , [11]) is used to solve the symmetrically preconditioned version of (1.1) with $\hat{\mathcal{A}}_+$ as coefficient matrix and GMRES [13] is used to solve the positive semi-definite problem with $\hat{\mathcal{A}}_-$ as coefficient matrix, then the two resulting solution methods for (1.1) are equivalent in the sense that

$$||r_k^{\text{MINRES}}|| = ||r_k^{\text{GMRES}}||,$$

and further convergence is independent of the choice of η, and

$$\left|\left|r_{2k+1}^{\text{MINRES}}\right|\right| = ||r_{2k}^{\text{MINRES}}|| \quad \left(= ||r_{2k}^{\text{GMRES}}|| = \left|\left|r_{2k+1}^{\text{GMRES}}\right|\right|\right),$$

so that both methods make no progress every second step. These observations make MINRES the most attractive of the two methods since it is cheaper to implement than GMRES.

The system (5.2) is equivalent to the preconditioned generalised least-squares problem

$$\min \left|\left| BQ^{-T}\hat{x} - \hat{b} \right|\right|_{A^{-1}},$$

(c.f. (4.4)) and hence LSQR(A^{-1}, H^{-1}) can be applied. The matrix H will certainly affect the convergence of both the least-squares and the MINRES methods of solution, but since only a comparison of these two approaches is required the matrix H plays no part as will hopefully become clear. Therefore it will be assumed that $H = I$. The MINRES approach then corresponds to solving the system

$$\begin{bmatrix} I & L^{-1}B \\ B^T L^{-T} & 0 \end{bmatrix} \begin{bmatrix} L^T x \\ y \end{bmatrix} = \begin{bmatrix} L^{-1}b \\ 0 \end{bmatrix}.$$

Recall now (§3.1) that the MINRES algorithm is a development of the Lanczos process and by (3.3) solves a system

$$T_k \hat{p} = \beta_1 e_1$$

at each stage of the Lanczos process. Now following the analysis of §3.3 , the tridiagonal matrix T_k can be seen to be of the form

$$T_{2k} = \begin{bmatrix} 1 & \alpha_1 & & & & \\ \alpha_1 & 0 & \beta_2 & & & \\ & \beta_2 & 1 & \alpha_2 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \beta_l & 1 & \alpha_l \\ & & & & \alpha_l & 0 \end{bmatrix}$$

$$T_{2k+1} = \begin{bmatrix} 1 & \alpha_1 & & & & & \\ \alpha_1 & 0 & \beta_2 & & & & \\ & \beta_2 & 1 & \alpha_2 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \alpha_l & 0 & \beta_{l+1} \\ & & & & & \beta_{l+1} & 1 \end{bmatrix}.$$

Hence at every odd step the MINRES approach will solve a system of the form $T_{2k+1}\hat{p}_{2k+1} = \beta_1 e_1$ and as was remarked in §3.3 there exists a permutation matrix P_{2k+1} such that

$$P_{2k+1} T_{2k+1} P_{2k+1} = \begin{bmatrix} I & E_k \\ E_k^T & 0 \end{bmatrix}$$

so that the $2k{+}1^{th}$ MINRES subproblem is exactly the same as the k^{th} LSQR(A^{-1}) subproblem. Hence if x_k were to be recovered at every step of LSQR(A^{-1}) it would be seen that

$$\begin{bmatrix} x_{k+1}^{\mathrm{LSQR}(A^{-1})} \\ y_{k+1}^{\mathrm{LSQR}(A^{-1})} \end{bmatrix} = \begin{bmatrix} x_{2k+1}^{\mathrm{MINRES}} \\ y_{2k+1}^{\mathrm{MINRES}} \end{bmatrix}$$

so that the LSQR(A^{-1}) approach to solving (1.1) requires half the number of iterations of the preconditioned MINRES approach to reach exactly the same approximate solution, i.e. the LSQR(A^{-1}) method steps over the redundant MINRES iterations found in [3]. It should be noted that it is not usually practical to update $x_k^{\mathrm{LSQR}(A^{-1})}$ at every step, however once $y_k^{\mathrm{LSQR}(A^{-1})}$ is deemed sufficiently accurate, $x_k^{\mathrm{LSQR}(A^{-1})}$ can be recovered at a cost of less than one LSQR(A^{-1}) step. It can be seen ([2]) that MINRES on the preconditioned system (5.2) requires

- Matrix-vector products $2\ (n \times m)$
- A^{-1} operations 1
- Dot products $2\ (n)\ ,\ 2\ (m)$
- Additional flops $12n + 12m$
- Stored vectors $7\ (n)\ ,\ 7\ (m)$

per step, where $(\cdot)$ indicates the dimension of the operation, whereas the operation count for one step of LSQR(A^{-1}) is

- Matrix-vector products $2\ (n \times m)$
- A^{-1} operations 1
- Dot products $1\ (n)\ ,\ 1\ (m)$
- Additional flops $4n + 8m$
- Stored vectors $1\ (n)\ ,\ 4\ (m)$

so that the LSQR(A^{-1}) method seems to be more attractive than either of the preconditioned MINRES or GMRES approaches.

In [3], the authors also note that applying MINRES to the matrix

$$\hat{\mathcal{A}}_+(\hat{\mathcal{A}}_+ - \eta I) = \begin{bmatrix} \hat{B}\hat{B}^T & 0 \\ 0 & \hat{B}^T\hat{B} \end{bmatrix}$$

generates the same iterates as MINRES applied the the matrix $\hat{\mathcal{A}}_+$. Notice that if an initial vector of the form $(0\ \ r_0^T)^T$ were used to generate a Krylov subspace together with the matrix above then each Krylov vector would retain the zero component in its upper part. Therefore applying MINRES to $\hat{\mathcal{A}}_+(\hat{\mathcal{A}}_+ - \eta I)$ is equivalent to applying MINRES to $\hat{B}^T\hat{B} = B^TA^{-1}B$ (when $H = I$, the case $H \neq I$ being similar), that is applying MINRES to the Schur complement matrix.

Obviously when B is of full column rank the cheaper conjugate gradient method may be preferred. This observation led to the conclusion that the most efficient way to solve (1.1) (if the preconditioner (5.1) is viable) is to apply MINRES to the Schur complement system. However notice that MINRES applied to the Schur complement system satisfies (at the k^{th} step)

$$\min_{y \in \mathcal{K}_1^k} \left|\left|B^T A^{-1} B y - B^T A^{-1} b\right|\right| = \left|\left|B^T A^{-1} B y_k^{\text{MINRES}} - B^T A^{-1} b\right|\right|$$

whereas at the k^{th} step of LSQR(A^{-1}),

$$\min_{y \in \mathcal{K}_2^k} ||By - b||_{A^{-1}} = \left|\left|By_k^{\text{LSQR}(A^{-1})} - b\right|\right|_{A^{-1}}.$$

where $\mathcal{K}_1^k$ and $\mathcal{K}_2^k$ are the Krylov subspaces associated with MINRES and LSQR(A^{-1}) respectively. It can be verified that with the choice of $y_0 = 0$ the two Krylov subspaces are the same, indeed $\mathcal{K}_1^k = \mathcal{K}_2^k = \mathcal{K}^k(B^T A^{-1} B, B^T A^{-1} b)$. Hence by previous remarks both methods would return the same iterates in exact arithmetic. Notice further that the Schur complement system

$$B^T A^{-1} B y = B^T A^{-1} b$$

is simply the normal equations corresponding to the generalised least-squares problem (2.3). This observation leads to the conclusion that in inexact arithmetic, MINRES on the Schur complement system will be numerically inferior to LSQR(A^{-1}) on ill-conditioned problems since the Schur complement matrix can have worse condition than the coefficient matrix $L^{-1}B$ of the generalised least-squares problem.

6 Numerical experiments and implementation

Following from the final comments of §5 , little difference between applying MINRES to the Schur complement and LSQR(A^{-1}) to the generalised least-squares problems is to be expected when the Schur complement matrix is well conditioned. Experiments have shown this to be the case, with LSQR(A^{-1}) proving slightly better at minimising $||y - y_k||$ and $||By_k - b||_{A^{-1}}$.

The advantage of LSQR(A^{-1}) over the MINRES approach is more obvious when considering more ill-conditioned problems. Figure 1 illustrates the performance of LSQR(A^{-1}) and MINRES at minimising the error, $||y - y_k||$, and residual, $||r_k||_{A^{-1}} = ||By_k - b||_{A^{-1}}$, for a finite element discretisation of flow in a porous region containing non-porous 'fingers', as governed by Darcy's law. The resulting generalised least-squares problem is obviously incompatible ($By \neq b$) since $x = 0$ would correspond to a flow with zero velocity. Similar performance is displayed initially by each method, however after approximately 10 iterations LSQR(A^{-1}) can be seen to reduce the error faster than MINRES. Both methods perform similarly in reducing the residual with LSQR(A^{-1}) being only slightly faster. The fact that the generalised least-squares problem is so incompatible (in this case $||By - b||_{A^{-1}} \approx 6 \times 10^{-1}$) and that $||r_0||_{A^{-1}}$ is small makes a more precise comparison of each method's residual reducing properties difficult.

For the purpose of these experiments the A^{-1}-norm of the residual has been calculated exactly using an A^{-1} solve at each step of the iteration. In practise however, the estimate

$$||r_k||_{A^{-1}} = \overline{\varphi}_{k+1} = \beta_1 s_k s_{k-1} \ldots s_1$$

should be used since it is virtually free and has been observed to agree closely with the calculated value of $||r_k||_{A^{-1}}$. This bound is analogous to that given in [12] for $||r_k||$ in LSQR and its derivation is the same.

The system described above was used to construct a more compatible problem by altering b so that $||By - b|| < 6 \times 10^{-3}$. Since the system is more compatible, comparison of the abilities of LSQR(A^{-1}) and MINRES at reducing the residual is possible. Figure 2 shows the performance of both methods on this system. Again the advantage of LSQR(A^{-1}) is clear after only a small number of iterations.

A similar estimate to that above for $||r_k||_{A^{-1}}$ is available for the residual of the Schur complement equations (the normal equations for the generalised-least squares problem),

$$\left|\left|B^T A^{-1} r_k\right|\right| = \overline{\varphi}_{k+1} \alpha_{k+1} |c_k|.$$

This estimate is a useful stopping criterion since $\left|\left|B^T A^{-1} r_k\right|\right|$ should be small for good approximations, y_k, to the solution. It is to be expected that MINRES will be the superior of the two methods at minimising $\left|\left|B^T A^{-1} r_k\right|\right|$ since this is precisely the quantity which is minimised at each step when MINRES is applied to the Schur complement system. This can be observed to be the case in Figure 3 which shows $\left|\left|B^T A^{-1} r_k\right|\right|$ for the problem considered in Figure 2. As expected, $\left|\left|B^T A^{-1} r_k^{\text{MINRES}}\right|\right|$ is monotonically decreasing, whereas $\left|\left|B^T A^{-1} r_k^{\text{LSQR}(A^{-1})}\right|\right|$ exhibits oscillations. Hence care should be taken when using the residual of the Schur complement equations as a stopping criterion (as MINRES was seen to be inferior when minimising the error in Figure 2).

As a final example it is shown that LSQR(A^{-1}) can succeed in solving a moderately ill-conditioned problem upon which MINRES fails. Figure 4 depicts what happens when both methods are applied to a system with $B \in \mathbb{R}^{32 \times 16}$ and with $\kappa(B^T A^{-1} B) = 5.3 \times 10^4$. Although LSQR($A^{-1}$) requires a greater number of iterates than expected it successfully minimises $||y - y_k||$ whereas the MINRES approach fails to noticeably reduce the error. This trend was observed to continue up to 200 iterations at which point the computation was terminated.

7 Summary

It has been demonstrated that considering augmented systems of the form (1.1) as generalised least-squares problems of the form (1.2) can be advantageous when constructing iterative solution methods. In §4 it was seen that the standard LSQR algorithm for least-squares problem could be extended to algorithm LSQR(A^{-1}) for generalised least-squares problems without assuming availability of the Cholesky decomposition of the weight matrix A, which would be required if the LSQR algorithm were to be applied naïvely to the problem. This fact also results in a saving when the weight matrix is diagonal, the LSQR(A^{-1}) method requiring one less diagonal matrix multiplication per iteration than the LSQR approach. The results of §5 show that the LSQR(A^{-1}) method for (1.2) is essentially twice as fast as the corresponding preconditioned MINRES method for (1.1) as it is able to step over the redundant steps in the MINRES approach. Since the condition of the generalised least-squares problem is better than that of the equivalent Schur complement system it has been seen (§6) that the LSQR(A^{-1}) method also has nicer numerical properties than MINRES applied to the Schur complement system.

In conclusion, if it is reasonable to precondition (1.1) using the matrix A, algorithm LSQR(A^{-1}) should be used to solve (1.2) in favour of MINRES on the Schur complement equations since it is more reliable in practise.

Acknowledgements

The author is very grateful to Professor Gene Golub for his useful discussions and hospitality during a visit to Stanford in 1996, and to Professor Michael Saunders for his helpful comments and introduction to LSQR. The author also wishes to thank his thesis supervisors Professor Alastair Spence and Andrew Cliffe of AEA Technology, Harwell for their advice and encouragement on this work.

References

[1] I. Duff, *The solution of augmented systems*, Tech. Report RAL-93-084, Rutherford Appleton Laboratory, Nov. 1993.

[2] B. Fischer, *Polynomial based iteration methods for symmetric linear systems*, Advances in numerical mathematics, Wiley-Teubner, 1996.

[3] B. Fischer, A. Ramage, D. Silvester, and A.J. Wathen, *Minimum residual methods for augmented systems*, Tech. Report 15, University of Strathclyde, June 1995.

[4] G. Golub and W. Kahan, *Calculating the singular values and pseudoinverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[5] M. Hestenes and E.Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. N.B.S., 49 (1952), pp. 409–436.

[6] J. Johnston, *Econometric methods*, McGraw-Hill, New York, 2nd ed., 1972.
[7] N. Karmarkar, *A new polynomial-time algorithm for linear-programming*, Combinatorica, 4 (1984), pp. 373–395.
[8] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. N.B.S., 45 (1950), pp. 255–282.
[9] C. Paige, *Bidiagonalisation of matrices and solution of linear equations*, SIAM J. Numer. Anal., 11 (1974), pp. 197–209.
[10] ———, *Fast numerically stable computations for generalised linear least squares problems*, SIAM J. Numer. Anal., 16 (1979), pp. 165–171.
[11] C. Paige and M. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[12] ———, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., 8 (1982), pp. 43–71.
[13] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[14] G. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[15] S. Vavasis, *Stable finite elements for problems with wild coefficients*, SIAM J. Numer. Anal., 33 (1996), pp. 890–916.

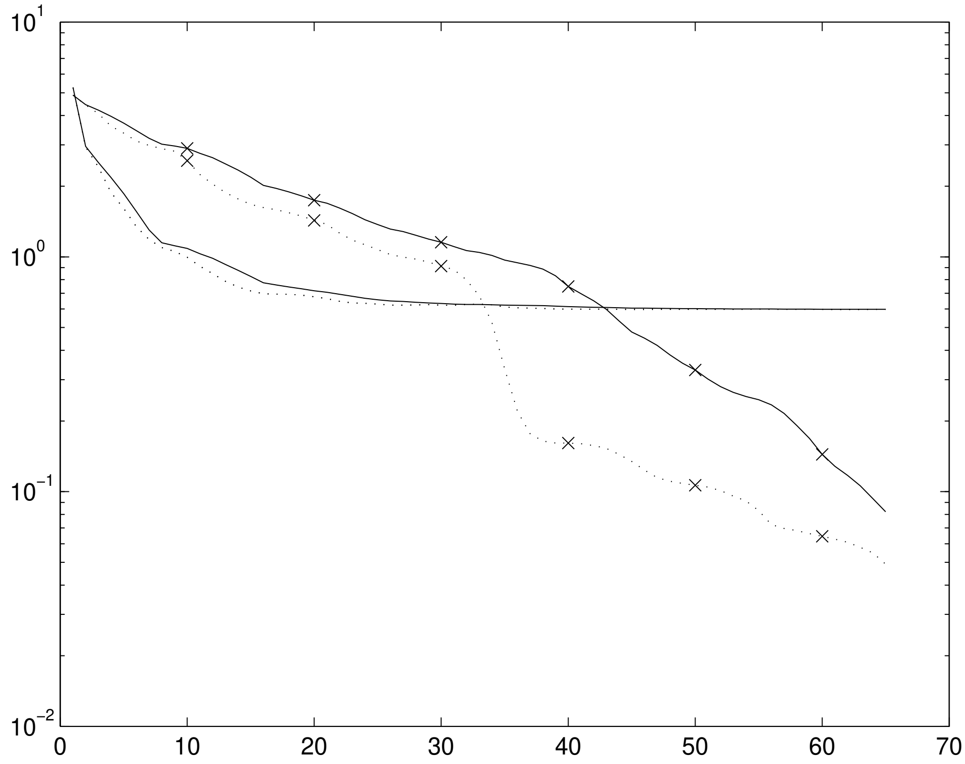

Figure 1: *Graphs comparing the performance of MINRES on the Schur complement system and LSQR(A^{-1}) on the generalised least-squares problem for a groundwater flow system with $\kappa(B^T A^{-1} B) = 7.5 \times 10^2$. The (two) crossed lines show the errors, $||y - y_k||$, for both methods. The (uncrossed) lines representing the residuals, $||r_k||_{A^{-1}}$, are also plotted. Solid lines (—) show the performance of MINRES against iteration and dotted lines ($\cdots$) represent LSQR(A^{-1}) . LSQR(A^{-1}) is faster to reduce the error and slightly faster at reducing the residual norms.*

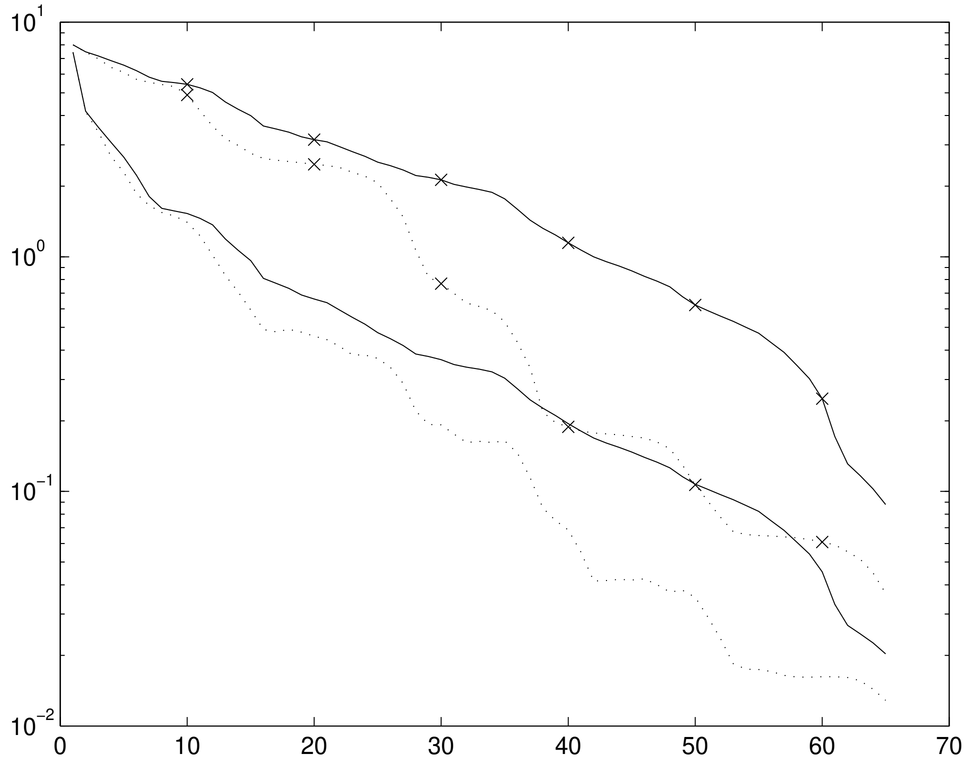

Figure 2: *Graphs comparing the performance of MINRES on the Schur complement system and LSQR(A^{-1}) for an 'almost compatible' generalised least-squares problem ($||By - b|| < 6 \times 10^{-3}$), $\kappa(B^T A^{-1} B) = 7.5 \times 10^2$. The (two) crossed lines show the errors, $||y - y_k||$, for both methods. The (uncrossed) lines representing the residuals, $||r_k||_{A^{-1}}$, are also plotted. Solid lines (—) show the performance of MINRES against iteration and dotted lines ($\cdots$) represent LSQR(A^{-1}) . LSQR(A^{-1}) is noticeably faster at reducing the error and residual norms.*

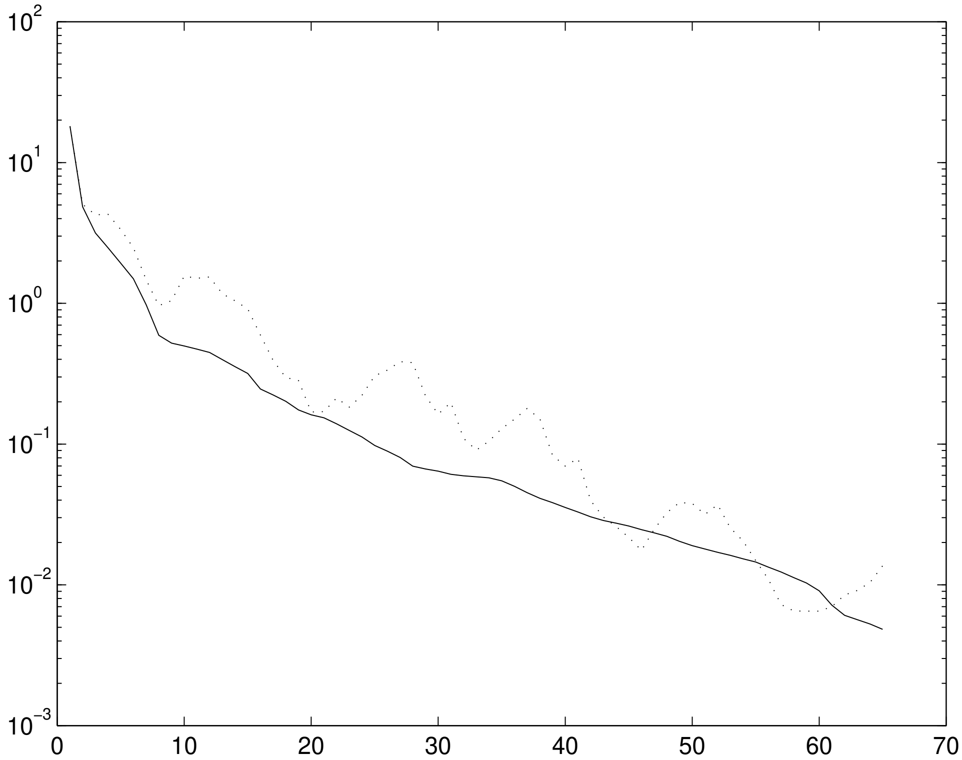

Figure 3: *Graphs showing* $||B^T A^{-1} r_k||$ *for the generalised least-squares problem in Figure 2. Solid lines (—) show the performance of MINRES against iteration and dotted lines (···) represent LSQR(*A^{-1}*) . MINRES performs better at minimising* $||B^T A^{-1} r_k||$ *as expected from its minimisation property.*

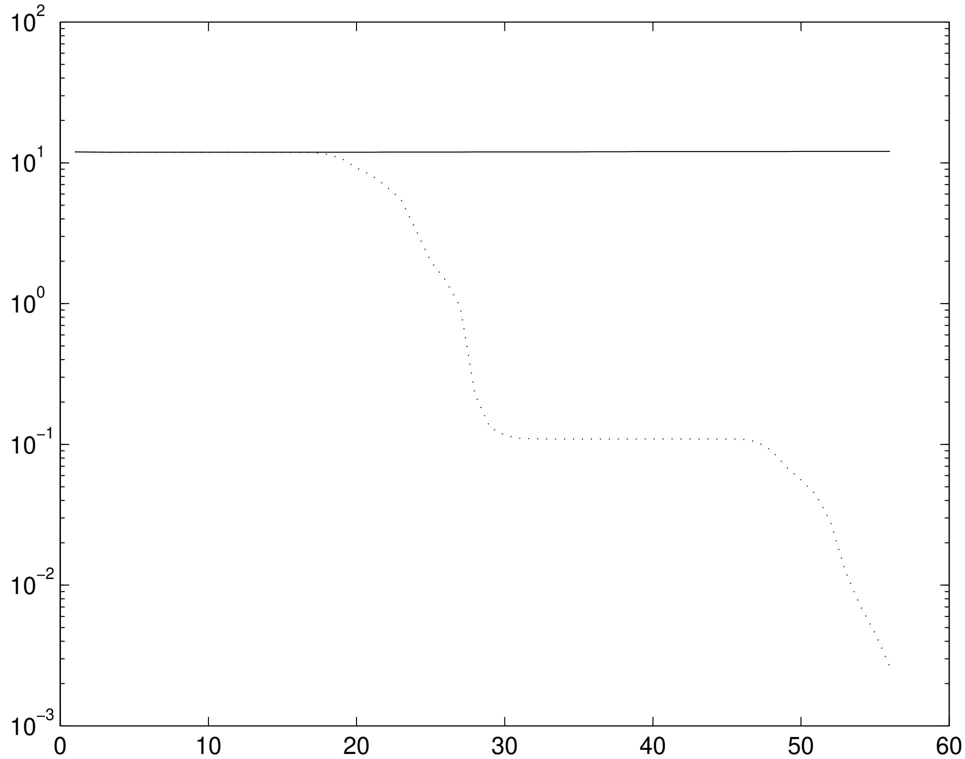

Figure 4: *Graphs showing $||y - y_k||$ for a moderately ill-conditioned problem, $\kappa(B^T A^{-1} B) = 5.3 \times 10^4$. Solid lines (—) show the performance of MINRES against iteration and dotted lines $(\cdots)$ represent LSQR(A^{-1}) . MINRES fails to reduce the norm of the error, LSQR(A^{-1}) succeeds eventually.*