ELSEVIER Journal of Computational and Applied Mathematics 66 (1996) 571–584

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Numerical methods for generalized least squares problems

Jin Yun Yuan*,[1]

Departamento de Matemática, Universidade Federal do Paraná, Centro Politécnico, CP: 19.081, Curitiba, CEP: 81531-990, Brazil

Received 27 July 1994; revised 30 June 1995

Abstract

Usually generalized least squares problems are solved by transforming them into regular least squares problems which can then be solved by well-known numerical methods. However, this approach is not very effective in some cases and, besides, is very expensive for large scale problems. In 1979, Paige suggested another approach which consists of solving an equivalent equality-constrained least squares problem by the orthogonal decomposition, the BNP algorithm or the James' implicit nullspace iterative methods. In this paper, we present some new developments of the numerical methods, for example, 2-cycle SOR method and preconditioned conjugate gradient method, for generalized least squares problems. Some numerical comparisons are included as well.

Keywords: Preconditioned conjugate gradient method; Generalized least squares problems; SOR method; Implicit nullspace method; Paige's method

AMS classification: 65F10; 6F05

1. Introduction

The generalized least squares problem

$$\min_{y \in \mathbb{R}^n} (Ay - b)^{\mathrm{T}} W^{-1}(Ay - b), \tag{1.1}$$

where $b \in \mathbb{R}^m$ is a given vector, $A \in \mathbb{R}^{m \times n}$ a given matrix and $W \in \mathbb{R}^{m \times m}$ a known symmetric and positive-definite matrix, is equivalent to a regular least squares problem with respect to a general

* E-mail: jin@mat.ufpr.br, yuan@impa.br.

[1] The work of the author was supported by CNPq grants 301035/93-8.

SSDI 0377-0427(95)00167-0

elliptic norm, rather than the Euclidean one. Usually problem (1.1) is transformed into a regular least squares problem

$$\min_{y \in \mathbb{R}^n} \|B^{-1}(Ay - b)\|_2, \tag{1.2}$$

where $W = BB^T$, solved by well known numerical methods as the SVD method, the orthogonal transformation methods, the block SOR method, the block AOR method, the conjugate gradient method and so on. However, this approach does not work very well in some cases, and is also very expensive for large scale problems. Hence Paige [14, 15] has proposed another approach to change problem (1.1) into an equality-constrained least squares problem with the same solution, and then to solve the new problem by the direct methods given in [12, 16], and the iterative methods, such as the algorithm of Barlow et al. [2] (which we will call BNP) and the implicit nullspace method of James [10, 11, 13].

Since the BNP algorithm and James' implicit nullspace method have some limitations for solving problem (1.1), we presented the 2-cyclic SOR method and the preconditioned conjugate gradient method.

As regard to the requirement of preconditioned matrix A_1 about all block iterative methods for least squares problems and problem (1.1), Björck and Yuan [4] present some algorithms to get preconditioned matrix A_1 by LU factorization.

In this paper, we first summarize Paige's method and James' method in Section 2, and then present the 2-cyclic SOR method and the preconditioned conjugate gradient method for (1.1), respectively, in Sections 3 and Section 4, and some numerical results, remarks, and open problems are given in the last section. Here A is always assumed to be of full rank.

2. Paige's method and James' method

2.1. Paige's method

Suppose that $W \in \mathbb{R}^{m \times m}$ has a factorization

$$W = BB^T, \tag{2.1}$$

where $B \in \mathbb{R}^{m \times k}$, and $\operatorname{rank}(B) = \operatorname{rank}(W) = k$. Therefore, problem (1.1) is equivalent to

$$\begin{aligned} &\min_{x,v} \|v\|_2 \\ &\text{s.t. } Ax + Bv = b \end{aligned} \tag{2.2}$$

where $x \in \mathbb{R}^n$ and $v \in \mathbb{R}^k$. This is a more general formulation and even allows for a rank deficient B.

By the QR decomposition of A, problem (2.2) becomes

$$\min_{v,x} \|v\|_2 \quad \text{s.t.} \quad C_2^T v = c_2, \tag{2.3}$$

and

$$Rx = c_1 - C_1^T v, \tag{2.4}$$

where

$$Q^{\mathrm{T}}b = \begin{pmatrix} R \\ 0 \end{pmatrix}, \qquad Q = (Q_1, Q_2) \text{ orthogonal}, \tag{2.5}$$

R is nonsingular upper triangular, and

$$Q^{\mathrm{T}}b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \quad \text{and} \quad Q^{\mathrm{T}}B = \begin{pmatrix} C_1^{\mathrm{T}} \\ C_2^{\mathrm{T}} \end{pmatrix}, \tag{2.6}$$

$c_1 \in \mathbb{R}^l$, $C_1 \in \mathbb{R}^{k\times l}$ and $l = \operatorname{rank}(R)$. For solving (2.3), obtain an orthogonal matrix $P \in \mathbb{R}^{k\times k}$ such that

$$P^{\mathrm{T}}C_2 = \begin{pmatrix} 0 \\ S^{\mathrm{T}} \end{pmatrix} \quad \text{and} \quad P^{\mathrm{T}}v = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \tag{2.7}$$

where $P = (P_1, P_2)$ and S is nonsingular upper triangular. Hence, the final solution is given by

$$\begin{aligned} &u_1 = 0, \\ &u_2 = S^{-1}c_2, \\ &v = P_2u_2, \\ &x = R^{-1}(c_1 - C_1^{\mathrm{T}}v). \end{aligned} \tag{2.8}$$

The algorithm for (1.1) is as follows.

Algorithm 2.1, Paige's Algorithm.

(1) Factorize W into $W = BB^{\mathrm{T}}$ by (incomplete) Cholesky decomposition where B is upper triangular matrix;

(2) Apply orthogonal transformation to (A, B, b) such that

$$Q^{\mathrm{T}}(A, B, b) = \begin{pmatrix} R & C_1^{\mathrm{T}} & c_1 \\ 0 & C_2^{\mathrm{T}} & c_2 \end{pmatrix};$$

(3) Apply orthogonal transofrmation P to C_2 such that

$$P^{\mathrm{T}}C_2 = \begin{pmatrix} 0 \\ S^{\mathrm{T}} \end{pmatrix};$$

(4) Solve $Su_2 = c_2$;

(5) Solve $Rx = c_1 - C_1^{\mathrm{T}}P_2u_2$, where $P = (P_1 \quad P_2)$.

It follows from perturbation analysis result given in [15] that Paige's algorithm is numerically stable.

2.2. *James' method*

Based on (2.3) and (2.4), the BNP algorithm [2] and the implicit nullspace iterative methods of James [13] for equality-constraints least squares problem

$$\begin{aligned} &\min \|Gy - c\|_2, \\ &\text{s.t. } Ey = b, \end{aligned} \tag{2.9}$$

can be applied to solve (1.1) with special matrices $G = (0 \quad I)$, $E = (A \quad B)$ and

$$y = \begin{pmatrix} x \\ v \end{pmatrix}.$$

Then we can establish the James' algorithm for (1.1) as follows.

Algorithm 2.2.

(1) Use Gauss elimination or orthogonal reduction on $(A \quad B)$ and b to replace $(A \quad B)$ by its staircase form E_s;

(2) Choose a convenient augmentation matrix $M_1 \in \mathbb{R}^{(n+k-m)\times(n+k)}$, and store the interlacing information in a a permutation vector;

(3) Form

$$B_1 = P\begin{pmatrix} E_s \\ M_1 \end{pmatrix}, \quad b_0 = P\begin{pmatrix} b \\ 0 \end{pmatrix}, \quad P = [P_L, P_R],$$

where P is a permutation matrix such that, with the staircase matrix E_s, it satisfies

$$E_s P = [E_s P_L, E_s P_R = E_L, E_R] = E_t,$$

E_t is the trapezoidal matrix obtained from E by Gauss elimination (or orthogonal reduction) with column pivoting, and E_L is upper triangular and nonsingular;

(4) Initialization

$$y_0 = 0,$$

$$d_0 = P_R^T (0 \quad B_2)\left(c - \begin{pmatrix} 0 \\ B_2 \end{pmatrix} b_0\right),$$

$s_0 = d_0$, where B_2 consists of last k rows of B_1^{-1};

(5) Iteration for $k = 0, 1, 2, \ldots$, until $d_k^T d_k <$ tolerance

$$q_k = P_R^T B_1^{-T}\begin{pmatrix} 0 \\ B_2 \end{pmatrix} P_R s_k,$$

$$\alpha_k = \frac{d_k^T d_k}{s_k^T q_k},$$

$$\begin{pmatrix} y_{k+1} \\ d_{k+1} \end{pmatrix} = \begin{pmatrix} y_k \\ d_k \end{pmatrix} + \alpha_k \begin{pmatrix} s_k \\ -q_k \end{pmatrix},$$

$$\beta_{k+1} = \frac{d_{k+1}^T d_{k+1}}{d_k^T d_k},$$

$$s_{k+1} = d_{k+1} + \beta_{k+1} s_k;$$

(6) Recover

$$x = B_1^{-1}(P_R y = b_0).$$

Remark. We mention that the matrix B_1 consists of the submatrix of A and submatrix of B in Algorithm 3.1. The augmentation matrix M_1 can be chosen as simple as possible such that

$$\begin{pmatrix} E_s \\ M_1 \end{pmatrix}$$

is nonsingular, for example, M_1 consisting of the identity matrix and zero matrix.

However, we will meet some unnecessary numerical degradation and difficulties for analysis of the practical problem when Algorithm 2.2 is applied to solve (1.1) because in this case, James' method deals with (1.1) by combining the matrices A and B but not considering them separately [16].

3. SOR-type methods

In order to utilize the original data in problem (1.1) and to overcome difficulties in the end of last section, the preconditioned block, SOR-type methods and preconditioned conjugate gradient method are considered in next two sections (also see [21–24]). Suppose that A has the following splitting:

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \tag{3.1}$$

where $A_1 \in \mathbb{R}^{n\times n}$ is nonsingular and $A_2 \in \mathbb{R}^{(m-n)\times n}$. W and b have corresponding splittings:

$$W = \begin{pmatrix} W_{11} & W_{12} \\ W_{12}^{\mathrm{T}} & W_{22} \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \tag{3.2}$$

where $W_{11} \in \mathbb{R}^{n\times n}$ and $W_{22} \in \mathbb{R}^{(m-n)\times(m-n)}$ are symmetric and positive-definite because W is symmetric and positive definite, and $W_{12} \in \mathbb{R}^{n\times(m-n)}$. Here we assume that the submatrix A_1 is given. For general case, some idea of obtaining the nonsingular submatrix A_1 and A by LU-decomposition is presented in [4]. In terms of the block structures of (3.1) amd (3.2), the normal equation of (1.1) is

$$\begin{pmatrix} A_1 & W_{12} & W_{11} \\ A_2 & W_{22} & W_{12}^{\mathrm{T}} \\ 0 & A_2^{\mathrm{T}} & A_1^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} x \\ r_2 \\ r_1 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix}, \tag{3.3}$$

where

$$r = W^{-1}(b - Ax) = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

is weighted residual vector, corresponding to splitting form (3.1) of A. Premultiplying (3.3) by D^{-1} where

$$\begin{pmatrix} A_1 & 0 & W_{11} \\ 0 & W_{22} & 0 \\ 0 & 0 & A_1^{\mathrm{T}} \end{pmatrix},$$

we obtain the 2-cyclic system

$$\begin{pmatrix} I & A_1^{-1}(W_{12} - W_{11}P^{\mathrm{T}}) & 0 \\ W_{22}^{-1}A_2 & I & W_{22}^{-1}W_{12}^{\mathrm{T}} \\ 0 & P^{\mathrm{T}} & I \end{pmatrix} \begin{pmatrix} x \\ r_2 \\ r_1 \end{pmatrix} = \begin{pmatrix} A_1^{-1}b_1 \\ W_{22}^{-1}b_2 \\ 0 \end{pmatrix}, \tag{3.4}$$

where $P = A_2A_1^{-1}$. We apply the SOR method to solve (3.4), and get the 2-cyclic SOR algorithm for problem (1.1) as follows.

Algorithm 3.1 (2-cyclic block SOR Algorithm).

(1) Factorize A_1 and W_{22}, $x^{(0)} = 0$, $r^{(0)} = 0$;

(2) Select a relaxation parameter ω;

(3) Iterate for $k = 0, 1, \ldots$, until "convergence"

$$x^{(k+1)} = (1 - \omega)x^{(k)} + \omega A_1^{-1}[b_1 - (W_{12} - W_{11}P^{\mathrm{T}})r_2^{(k)}],$$

$$r_2^{(k+1)} = (1 - \omega)r_2^{(k)} + \omega W_{22}^{-1}[b_1 - W_{12}^{\mathrm{T}}r_1^{(k)} - A_2x^{(k+1)}),$$

$$r_1^{(k+1)} = (1 - \omega)r_1^{(k)} - \omega P^{\mathrm{T}}r_1^{(k+1)}.$$

For system (3.4), the associated Jacobi matrix J is given by

$$J = \begin{pmatrix} 0 & -A_1^{-1}(W_{12} - W_{11}P^{\mathrm{T}}) & 0 \\ W_{22}^{-1}A_2 & 0 & -W_{22}^{-1}W_{12}^{\mathrm{T}} \\ 0 & -P^{\mathrm{T}} & 0 \end{pmatrix}. \tag{3.5}$$

It is easy to show the following lemma.

Lemma 3.1. *The eigenvalues μ of the associated Jacobi matrix J of the 2-cyclic SOR method are either real numbers or pure imaginary numbers (i.e. $\mu^2 \in \mathbb{R}$) such that*

(a) *if W is symmetric and positive definite, then*

$$\mu^2 < 1;$$

(b) *if W is symmetric and positive semidefinite and W_{22} is symmetric and positive definite, then*

$$\mu^2 \leqslant 1.$$

Proof. Suppose that

$$\mu \quad \text{and} \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

are an eigenpair of J. It follows from (3.5) and

$$J\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mu \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{3.6}$$

that

$$W_{22}^{-1}[PW_{12} + W_{12}^{\mathrm{T}}P^{\mathrm{T}} - PW_{11}P^{\mathrm{T}}]y = \mu^2 y. \tag{3.7}$$

Since

$$W_{22}^{-1}[PW_{12} + W_{12}^{\mathrm{T}}P^{\mathrm{T}} - PW_{11}P^{\mathrm{T}}]$$

is similar to

$$W_{22}^{-1/2}[PW_{12} + W_{12}^{\mathrm{T}}P^{\mathrm{T}} - PW_{11}P^{\mathrm{T}}]W_{22}^{-1/2},$$

whose eigenvalues are real, the eigenvalue μ^2 of the matrix

$$W_{22}^{-1}[PW_{12} + W_{12}^{\mathrm{T}}P^{\mathrm{T}} - PW_{11}P^{\mathrm{T}}]$$

is real. Therefore, μ is pure real if $\mu^2 \leqslant 0$, and μ is pure imaginary if $\mu^2 < 0$. Also $y \neq 0$ is real.

Adding $-y$ to both sides of (3.7), we get

$$\{W_{22}^{-1}[PW_{12} + W_{12}^{\mathrm{T}}P^{\mathrm{T}} - PW_{11}P^{\mathrm{T}}] - I\}y = (\mu^2 - 1)y. \tag{3.8}$$

It follows from (3.8) and (3.2) that

$$(P, -I)W\begin{pmatrix} P^{\mathrm{T}} \\ -I \end{pmatrix} y = (1 - \mu^2)W_{22}y. \tag{3.9}$$

Premultiplying (3.9) by y, there is

$$z^{\mathrm{T}}W_z = (1 - \mu^2)y^{\mathrm{T}}W_{22}y, \tag{3.10}$$

where

$$0 \neq z = \begin{pmatrix} P^{\mathrm{T}} \\ -I \end{pmatrix} y \in \mathbb{R}^m.$$

Hence,

$$1 - \mu^2 = \frac{z^{\mathrm{T}}W_z}{y^{\mathrm{T}}W_{22}y}. \tag{3.11}$$

(a) has been shown because of (3.11) and positive definiteness of W_{22} and W. Since W_{22} is positive definite, and W is positive semi-definite, (b) follows from (3.11) immediately. □

From Varga's p-cyclic theory for SOR method [18, 19] and Lemma 4.1, we obtain the following convergence result of 2-cyclic SOR method for problem (1.1).

Theorem 3.2. *For problem* (1.1), *the 2-cyclic block SOR method of Algorithm* 3.1 *converges for*

$$0 < \omega < \frac{2}{1 + \rho(J)},$$

where $\rho(J)$ *is the spectral radius of the associated Jacobi matrix* J. *The optimum factor* ω_b *is given by*

$$\omega_b = \frac{2}{1 + \sqrt{1 + \alpha^2 - \beta^2}} \quad \text{and} \quad \rho(\mathcal{L}_{\omega_b}) = \left(\frac{\alpha + \beta}{+\sqrt{1 + \alpha^2 - \beta^2}}\right)^2, \tag{3.12}$$

where $\beta = \max|\mathrm{Re}(\mu)|$, $\alpha = \max|\mathrm{Im}(\mu)|$ *and* μ *is any eigenvalue of the associated Jacobi matrix* J. *Moreover, the behavior of* $\rho(\mathcal{L}_\omega)$ *is given by*

$$\rho(\mathcal{L}_\omega) = \begin{cases} \left[\dfrac{\omega\beta + \sqrt{4(1-\omega) + \omega^2\beta^2}}{2}\right]^2 & \text{if } 0 < \omega \leqslant \omega_b, \\ \left[\dfrac{\omega\alpha + \sqrt{4(\omega-1) + \omega^2\alpha^2}}{2}\right]^2 & \text{if } \omega_b \leqslant \omega < 2/(1+\alpha), \end{cases} \tag{3.13}$$

where $\mathcal{L}_\omega$ *is the iterative matrix of Algorithm* 3.1.

Proof. By Lemma 3.1, all the eigenvalues of J are either real or pure imaginary. Since the method in Algorithm 3.1 is 2-cyclic and consistently ordered, the first part of the theorem follows from Theorem 4.1 of Young in [20]. The optimal parameter ω_b and $\rho(\mathcal{L}_{\omega_b})$ in (3.12) is a straight extension of Young's result in [20]. Yuan [21] and Yuan and Iusem [23] give different proofs for this part.

By Varga's eigenvalueship [18, 19], for each eigenvalue λ of $\mathcal{L}_\omega$ there exists an eigenvalue μ of J, such that

$$(\lambda + \omega - 1)^2 = \lambda\mu^2\omega^2. \tag{3.14}$$

The last part of the theorem is obtained by analysis of (3.14). For details see [21, Chapter 5, 23]. □

4. Preconditioned conjugate gradient method

Premultiplying (3.3) by D^{-1} and using $r_1 = -P^{\mathrm{T}}r_2$, we get two systems

$$[W_{22} + PW_{11}P^{\mathrm{T}} - PW_{12} - W_{12}^{\mathrm{T}}P^{\mathrm{T}}]r_2 = b_2 - Pb_1, \tag{4.1}$$

$$A_1 x = b_1 - (W_{12} - W_{11}P^{\mathrm{T}})r_2, \tag{4.2}$$

where

$$D = \begin{pmatrix} A_1 & 0 & 0 \\ A_2 & I & 0 \\ 0 & 0 & A_1^{\mathrm{T}} \end{pmatrix} \quad \text{and} \quad P = A_2 A_1^{-1}.$$

Since

$$W_{22} + PW_{11}P^{\mathrm{T}} - PW_{12} - W_{12}^{\mathrm{T}}P^{\mathrm{T}} = (P \quad -I)W\begin{pmatrix} P^{\mathrm{T}} \\ -I \end{pmatrix},$$

and W is symmetric and positive definite, (4.1) is symmetric and positive-definite system because of $x^{\mathrm{T}}(P \quad -I)W(P \quad -I)^{\mathrm{T}}x = y^{\mathrm{T}}Wy > 0$, where $y = (P \quad -I)^{\mathrm{T}}x \neq 0$, for all $x \neq 0$. So we can apply the Hestenes and Stiefel's conjugate gradient method [9] to solve system (4.1), and get the preconditioned conjugate gradient algorithm for (1.1).

Algorithm 4.1 (CG-algorithm for GLSP).

(1) Factorize A_1, set $r_2^{(0)} = 0$, $v^{(0)} = b_2 - Pb_1$, $p^{(0)} = v^{(0)}$;

(2) Iterate for $k = 0, 1, \ldots$, until $v^{(k+1)} = 0$ (or $\|v^{(k+1)}\| \leqslant$ tolerance),

$$q = (P \quad -I)W\begin{pmatrix} P^{\mathrm{T}} \\ -I \end{pmatrix}p^{(k)},$$

$$\lambda_k = \frac{\|v^{(k)}\|_2^2}{\langle p^{(k)}, q\rangle},$$

$$v^{(k+1)} = v^{(k)} - \lambda_k q,$$

$$r_2^{k+1} = r_2^k + \lambda_k p^{(k)},$$

$$\alpha_{k+1} = \frac{\|v^{(k+1)}\|_2^2}{\|v^{(k)}\|_2^2},$$

$$p^{(k+1)} = v^{(k+1)} + \alpha_{k+1}p^{(k)};$$

(3) Solve the extra subsystem

$$A_1 x = b_1 + (W_{11}P^{\mathrm{T}} - W_{12})r_2^{(l)},$$

where $r_2^{(l)}$ is the solution obtained in step 2.

The standard errors bounds based on the Chebyshev polynomials for the CG method applied to problem (1.1) is given by

$$\frac{\|A(x^* - x^{(k)})\|_{W^{-1}}}{\|A(x^* - x^{(0)})\|_{W^{-1}}} \leqslant 2\frac{\left[\dfrac{(1+\alpha^2)\beta - 1}{(1+\sqrt{(1+\alpha^2)\beta})^2}\right]^k}{1 + \left[\dfrac{(1+\alpha^2)\beta - 1}{(1+\sqrt{(1+\alpha^2)\beta})^2}\right]^{2k}}, \tag{4.3}$$

where $\alpha = \|P\|_2 = \|A_2A_1^{-1}\|_2$, $\beta = \kappa(W) = \mu_{\max}(W)/\mu_{\min}(W)$ is the spectral condition number of W, $x^{(0)}$ is a vector corresponding to arbitrary initial vector $r_2^{(0)}$ and x^* is the solution of problem (1.1).

It follows from (4.1) that the sequence $x_{CG}^{(x)}$ generated by Algorithm 4.1 is in the Krylov subspace $x^{(0)} + A_1^{-1}(W_{12} - W_{11}P^T)K_{k-1}(v^{(0)}; E)$, where $v^{(0)}$ is a vector dependent of $x^{(0)}$ and $E = W_{22}^{-1}[P(W_{12} - W_{11}P^T) + W_{12}^T P^T]$. If we take

$$B_1 = \begin{pmatrix} -A_1^{-1}(W_{12} - W_{11}P^T) \\ -P^T \end{pmatrix} \quad \text{and} \quad B_2 = -(W_{22}^{-1}A_2 \quad W_{22}^{-1}W_{12}^T)$$

for (3.4) in Freund's Lemma [6], we know that $x_{SOR}^{(k)}$ generated by Algorithm 3.1 is in the same Krylov subspace as $x_{CG}^{(k)}$. By the minimal property of the conjugate gradient method in Krylov subspace, we can get theoretical comparison result about the 2-cyclic SOR method and the pre-conditioned conjugate gradient method for problem (1.1), which confirms Björck's conjecture [3] that Freund's result for least squares problems is also true for problem (1.1).

Theorem 4.1. *If the 2-cyclic SOR Algorithm 3.1 and the preconditioned conjugate gradient Algorithm 4.1 are all started with the same vector $x^{(0)} \in \mathbb{R}^n$, then kth iterates satisfy*

$$\|b - Ax_{CG}^{(k)}\|_{W^{-1}} \leqslant \|b - Ax_{SOR}^{(k+1)}\|_{W^{-1}}, \quad k = 0, 1, \ldots \tag{4.4}$$

where $x_{CG}^{(k)}$ and $x_{SOR}^{(k)}$ are generated by the conjugate gradient Algorithm 4.1 and 2-cyclic SOR Algorithm 3.1 respectively.

Similarly, for problem (1.1), between the preconditioned conjugate gradient Algorithm 4.1 and the generalized SO algorithms [21, 23], there are analog comparison results (cf. [21]). Therefore, the preconditioned conjugate gradient Algorithm 4.1 is better than the SOR-type methods for problem (1.1). The numerical experiments also verify the conclusion (cf. [21, 22]).

5. Numerical results, remarks and conclusions

5.1. Numerical results

We give numerical experiments for dense cases and sparse cases in UNIX SUN workstation and IBM4081, respectively, in FORTRAN 77. For $m = 125$ and $n = 50$, Algorithms 4.1 and 3.1 obtain the same accurate solution as the Paige's method (see Table 1). But the CPU time of Algorithms 3.1 and 4.1 is much less than that of the Paige's method even if A and W are dense matrices. For sparse cases, Algorithm 4.1 is much better than Algorithms 2.1 and 4.1 (Table 2). We also considered rank deficient of W for Algorithm 4.1 and different parameters ω for Algorithm 3.1 (see [21, 22]). All experiments have shown that the pre-conditioned conjugate gradient Algorithm 4.1 is one of the most efficient, even for some rank$(W) < m - n$ cases (cf. [21, 22]). It follows from (4.1) that the algorithm just needs A_1^{-1}, but neither W^{-1} nor inverse of some submatrix of W. The experiments in [21,22] have shown that Algorithm 4.1 is much better than others in the sense of CPU time, accuracy, and storage requirements (cf. [22]). Of course, the numerical stability of the Paige's method is better than of Algorithm 4.1 (see Table 3). In all tables, IT is the number of iterations, m and n are numbers of rows and columns, respectively, of A. The CPU time does not include the time of computing the optimal parameter ω_b. In Table 1, case 1 means all matrices A and W are

Table 1
Dense cases for $m = 125$ and $n = 50$

Method	Case 1			Case 2		
	IT	CPU	$e(10^{-14})$	IT	CPU	$e(10^{-14})$
Paige		32.388	0.252		34.740	0.279
SOR	16	3.982	0.093	19	4.732	0.100
CG	14	2.684	0.092	15	2.749	0.101

Note: $e = \|A^T W^{-1}(b - Ax)\|_2$ and the SOR method is with ω_b.

Table 2
Sparse cases

m	n	CG		SOR	
		IT	TIME	IT	TIME
400	250	25	0.342	22	0.386
550	250	32	0.351	21	0.447
650	250	33	0.386	21	0.453
850	250	33	0.461		

Note: TIME is all user time.

Table 3
A and W are Hilbert matrix

m	n	IT	CG		Paige	
			CPU	e	CPU	e
6	4	3	0.39043	0.195234E − 09	0.78130	0.205795E − 09
7	6	0	0.00500	0.930158E − 08	0.78130	0.575991E − 08
8	7	1	0.00555	0.149882E − 07	0.01055	0.160000E − 07
9	7	4	0.00550	0.555993E − 07	0.78130	0.532440E − 07
10	7	2	0.78125	0.177156E + 46	0.78130	0.728930E + 36

Note: 0.195234E − 09 means 0.195234×10^{-9}.

dominant, and case 2 does not. All matrices were generated by random numbers. All data in the tables are statistic data based on more than 25 examples. The tolerance for all iterative methods is $\varepsilon = 10^{-15}$.

We should also point out that all results in Sections 3 and 4 will reduce to the well-known results for least squares problems when $W = I$.

5.2. Remarks and conclusions

The Paige's method is numerically stable method and can work for any deficient problem. The method needs to decompose the weight matrix W if W is given, but not B. The Paige's method is not convenient for large sparse problems because it is a direct method.

Algorithm 2.2 is a combination of direct method and iterative method, such as the Gauss elimination and the conjugate gradient method. In principle, Algorithm 2.2 can also work for rank deficient problems. However, Algorithm 2.2 sometimes fails because the full rank condition of the matrix $(A \quad B)$ and $m \leqslant n + k \leqslant m + k$ cannot be guaranteed by rank $(W) < m$. Also we first need to decompose the matrix W into $W = BB^{\mathrm{T}}$ which will destroy the sparsity pattern of W. Algorithm 2.2 still requires the inverse of some submatrix of matrix (A, B) or itself which in general is hard to compute.

In fact, for sparse full rank problems, another good approach is the Lanczos based SYMMLQ method given by Paige and Saunders in [17]. Here we will not say more about this method. Based on system (3.4), the QMR method [7] can be applied to solve full rank problem (1.1) with $\operatorname{rank}(A) = n$.

Like James' method, p-cyclic SOR method, AOR method, and other methods for linear equality-constraints least squares problems also can be applied to solve problem (1.1). But they are not the same as the preconditioned SOR-type methods in Section 3 and the preconditioned conjugate gradient method in Section 4. The size ($3m + n$ for full rank) of p-cyclic SOR method and AOR method for (1.1) is much bigger than the size (here $m + n$) of preconditioned SOR-type method. And preconditioner $A_1 \in \mathbb{R}^{n \times n}$ for (3.4) is different from the preconditioner $A_1 \in \mathbb{R}^{(m+n)\times(m+n)}$, which combines the matrices A and B, for those methods in the sense of the size, structure and elements.

All block iterative methods for least squares problems and generalized least squares problems require the preconditioner matrix A_1. In general, we just know the matrix A, but not A_1. Up to now there is little literature to discuss how to get the preconditioner matrix A_1. Recently Björck and Yuan [4] suggested some algorithms to handle the problem by LU decomposition. How to effectively get the preconditioner matrix A_1 is still one open problem especially for large sparse problems.

Algorithms 3.1 and 4.1 just work for full rank problems. Rank deficient problem is unsolved yet by direct iterative methods. For this open problem, Golub has one conjecture [8] that the QMR method may be applied to solve problem (1.1) with $\operatorname{rank}(A) < n$. Suppose that $\operatorname{rank}(W) \geqslant \operatorname{rank}(W_{11}) = n$, it follows from (3.3) and preconditioned matrix

$$\begin{pmatrix} W_{11} & 0 & 0 \\ W_{12}^{\mathrm{T}} & I & 0 \\ A_1^{\mathrm{T}} & 0 & I \end{pmatrix}$$

that

$$\begin{pmatrix} I & W_{11}^{-1}W_{12} & W_{11}^{-1}A_1 \\ 0 & W_{22} - W_{22}^{\mathrm{T}}W_{11}^{-1}W_{12} & A_2 - W_{12}^{\mathrm{T}}W_{11}^{-1}A_1 \\ 0 & A_2^{\mathrm{T}} - A_1^{\mathrm{T}}W_{11}^{-1}W_{12} & -A_1^{\mathrm{T}}W_{11}^{-1}A_1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ x \end{pmatrix} = \begin{pmatrix} W_{11}^{-1}b_1 \\ b_2 - W_{12}^{\mathrm{T}}W_{11}^{-1}b_1 \\ -A_1^{\mathrm{T}}W_{11}^{-1}b_1 \end{pmatrix} \tag{5.1}$$

is reducible, and can be solved by the methods given in [1, 5, 7].

From Theorem 4.1, we know that two different systems are equivalent in the sense that the sequences generatedby the conjugate gradient method and the SOR method are in the same Krylov subspace. There is one question: if one system for one method is given, how to get the equivalent system for another given method such that the approximate solution sequences generated by these two methods are in the same Krylov subspace? For example, we know one symmetric and positive-definite system with ill-conditioning to which conjugate gradient method can be applied. In fact we cannot obtain the desired solution effectively because of ill-conditioning when we apply the conjugate gradient method. Now we want to look for equivalent system on which the SOR method or other methods can work very well. It is also one interesting problem.

Acknowledgements

The author wishes to express his sincere thanks to Professor Alfredo N. Iusem and Professor Apostolos Hadjidimos for their very kind help on this paper. The author also thanks Professor Gene Golub and Professor Alvaro De Pierro for their very helpful discussions. The author is very grateful to the referees for their most valuable comments and suggestions that significantly improved the quality of the paper, and to one referee for picking up references [13, 16, 17].

References

[1] V.A. Barker, Numerical solution of sparse singular systems of equations arising from ergodic Markov chains, *Comm. Statist. Stochastic Models* **5** (1989) 335–381.
[2] J.L. Barlow, N.K. Nichols and R.J. Plemmons, Iterative methods for equality constrained least squares problems, *SIAM J. Sci. Statist. Comput.* **9** (1988) 892–906.
[3] Å. Björck, private communication, 1992.
[4] Å. Björck and J.Y. Yuan, Preconditioner for least squares problems by LU factorization, to be published.
[5] M. Eiermann, I. Marek and W. Niethammer, On the solution of singular linear systems of algebraic equations by semiiterative methods, *Numer. Math.* **53** (1988) 265–283.
[6] R. Freund, A note on two block SOR methods for sparse least-squares problems, *Linear Algebra Appl.* **88/89** (1987) 211–221.
[7] R.W. Freund and M. Hockbruck, On the use of the QMR algorithms for solving singular systems and applications in Markov chain modeling, *J. Numer. Linear Algebra Appl.* **1** (1994) 403–420.
[8] G.H. Golub, private communication, 1993.
[9] M. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards* **49** (1952) 409–436.
[10] D.J. James, Conjugate gradient methods for constrained least squares problems, Ph.D. Dissertation, Dept. of Math., North Carolina State University, Raleigh, NC, 1990.
[11] D.J. James, Order-reducing conjugate gradients versus block AOR for constrained least-squares problems, *Linear Algebra Appl.* **154–156** (1991) 23–43.
[12] D.J. James, Implicit nullspace iterative methods for constrained least squares problems, *SIAM J. Matrix Anal. Appl.* **13** (1992) 962–978.
[13] C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems* (Prentice-Hall, Englewood Cliffs, NJ, 1974).
[14] C.C. Paige, Fast numerically stable computations for generalized least-squares problems, *SIAM J. Numer. Anal.* **16** (1979) 165–171.
[15] C.C. Paige, Computer solution and pertubation analysis of generalized linear least-squares problems, *Math. Comput.* **33** (1979) 171–184.

[16] C.C. Paige, Some aspects of generalized QR factorizations, in: Core and Hammarling, Eds., *Reliable Numerical Computations* (Oxford Univ. Press, New York, 1990).
[17] C.C. Paige and M. Saunders, Solution of sparse indefinite systems of linear equation, *SIAM J. Numer. Anal.* **12** (1975) 617–629.
[18] R.S. Varga, p-cyclic matrices: A generalization of the Young–Frankel successive overrelaxation scheme, *Pacific J. Math.* **9** (1959) 617–628.
[19] R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1962).
[20] D.M. Young, *Iterative Solution of Large Linear Systems* (Academic Press, New York, 1971).
[21] J.Y. Yuan, Iterative methods for generalized least squares problems, Ph.D. Thesis, IMPA, Brazil, 1993.
[22] J.Y. Yuan, Numerical comparisons of numerical methods for generalized least-squares problems, Tech. Report MAT94-06-01, Federal University of Paraná, Curitiba, Brazil, 1994.
[23] J.Y. Yuan and A. Iusem, SOR-type methods for generalized least-squares problems, Tech. Report MAT94-03-01, Federal University of Paraná, Curitiba, Brazil, 1994.
[24] J.Y. Yuan and A. Iusem, Preconditioned conjugate gradient method for generalized least-squares problems, Tech. Report MAT94-04-01, Federal University of Paraná, Curitiba, Brazil, 1994.