



**National Institute of Informatics**

---

**NII Technical Report**

**Preconditioned GMRES Methods  
for Least Squares Problems**

**Tokushi ITO and Ken HAYAMI**

**NII-2004-006E  
May 2004**

# PRECONDITIONED GMRES METHODS FOR LEAST SQUARES PROBLEMS\*

TOKUSHI ITO<sup>†</sup>      KEN HAYAMI<sup>‡</sup>

**Abstract.** For least squares problems of minimizing  $\|\mathbf{b} - A\mathbf{x}\|_2$  where  $A$  is a large sparse  $m \times n$  ( $m \geq n$ ) matrix, the common method is to apply the conjugate gradient method to the normal equation  $A^\top A\mathbf{x} = A^\top \mathbf{b}$ . However, the condition number of  $A^\top A$  is square of that of  $A$ , and convergence becomes problematic for severely ill-conditioned problems even with preconditioning. In this paper, we propose two methods for applying the GMRES method to the least squares problem by using a  $n \times m$  matrix  $B$ . We give the necessary and sufficient condition that  $B$  should satisfy in order that the proposed methods give a least squares solution. Then, for implementations for  $B$ , we propose an incomplete QR decomposition  $\text{IMGS}(l)$ . Numerical experiments show that the simplest case  $l = 0$ , which is equivalent to  $B = (\text{diag}(A^\top A))^{-1}A^\top$ , gives best results, and converges faster than previous methods for severely ill-conditioned problems.

**Key words.** Least squares problems, GMRES, preconditioning, incomplete QR-decomposition, singular systems

**AMS subject classifications.** 65F10, 65F20, 65F25, 65F35, 65F50, 15A06, 15A12, 15A23

---

\*This paper dated 30 April 2004.

<sup>†</sup>Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan (Graduated) ([zau80636@bird.zero.ad.jp](mailto:zau80636@bird.zero.ad.jp)).

<sup>‡</sup>National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan, also affiliated to the Department of Informatics, Graduate University for Advanced Studies (Sokendai) ([hayami@nii.ac.jp](mailto:hayami@nii.ac.jp)). The research of this author was supported by the Grant-in-Aid for Scientific Research of the Ministry of Education, Culture Sports, Science and Technology, Japan.

# 1 Introduction

In this paper, we consider the linear least squares problem

$$\min_{\mathbf{x} \in \mathbf{R}^n} \|\mathbf{b} - A\mathbf{x}\|_2 \quad (1.1)$$

where  $A$  is a  $m \times n$  ( $m \geq n$ ) large sparse matrix with full column rank.

The popular methods to solve the problem (1.1) are, the direct method[2] and the iterative method applying the Conjugate Gradient (CG) method[5] to the normal equation

$$A^\top A\mathbf{x} = A^\top \mathbf{b}. \quad (1.2)$$

This iterative method is called the CGLS method[2].

The direct method is reliable, but for large problems, computation time and storage become huge, while the iterative method generally requires small storage, and can be fast in execution time. However, this applies only if the convergence proceeds fast enough. The condition number of  $A^\top A$  is square of that of  $A$ , so the convergence becomes slow for ill-conditioned problems, and preconditioning is necessary.

Instead of using the normal equation, Zhang and Oyanagi[10, 11, 12] introduced mapping the least squares problem (1.1) to a system of linear equations with a (large)  $m \times m$  square coefficient matrix  $AB$  by a  $n \times m$  matrix  $B$  and then applying the Orthomin( $k$ ) method. By selecting the mapping matrix  $B$  properly, the condition number becomes small, and the convergence can be improved.

Following their work, in this paper, we will first consider applying the Generalized Minimal Residual (GMRES( $k$ )) method to the system of linear equations with the (large)  $m \times m$  coefficient matrix  $AB$ . Next, we will propose using the (small)  $n \times n$  matrix  $BA$ , which is equivalent to applying the GMRES( $k$ ) method to the system of linear equations  $BA\mathbf{x} = B\mathbf{b}$ . Numerical experiments show that this latter approach requires less computation. We will also derive the necessary and sufficient condition that the mapping matrix  $B$  should satisfy in order that both methods give a least squares solution. As the mapping matrix  $B$ , we propose using an incomplete QR decomposition. Numerical experiments show that the proposed method is faster and robust compared to previous methods, for severely ill-conditioned problems.

## 2 Previous methods for the linear least squares problems

### 2.1 Direct methods

The common direct method to solve (1.1) is to use QR decomposition[2]. The QR decomposition of a  $m \times n$  rectangular matrix  $A$  is given by

$$A = QR,$$

where  $Q$  is a  $m \times n$  rectangular matrix whose column vectors are orthonormal, i.e.  $Q^T Q = I_n$ , where  $I_n$  is the  $n \times n$  identity matrix, and  $R$  is a  $n \times n$  upper triangular matrix. If this decomposition can be carried out, the normal equation (1.2) becomes

$$R^T R \mathbf{x} = R^T Q^T \mathbf{b}, \quad (2.1)$$

and when  $A$  is full column rank,  $R$  is nonsingular, and (2.1) becomes

$$R \mathbf{x} = Q^T \mathbf{b}. \quad (2.2)$$

Since  $R$  is an upper triangular matrix, (2.2) can be solved by backward substitution. The QR decomposition can be carried out in a finite number of steps. There are various algorithms for implementing the QR decomposition, among which the Householder method and the (modified) Gram-Schmidt method is commonly used. The direct method is reliable, but when the problem is large, the required computational work and memory become huge.

### 2.2 Iterative methods using the normal equation

By transforming to the normal equation (1.2), the problem becomes a system of linear equations whose coefficient matrix is square and symmetric positive definite. The method of applying the conjugate gradient (CG) method to the normal equation is called the CGLS method[2]. The algorithm of the CGLS method is as follows.

**Algorithm 2.1** : *CGLS method*

Choose  $\mathbf{x}_0$ .  
 $\tilde{\mathbf{r}}_0 = A^T (\mathbf{b} - A \mathbf{x}_0)$   
 $\mathbf{p}_0 = \tilde{\mathbf{r}}_0$   
for  $i = 0, 1, 2, \dots$  until convergence  
 $\alpha_i = \frac{(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_i)}{(\mathbf{p}_i, A^T A \mathbf{p}_i)}$   
 $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$   
 $\tilde{\mathbf{r}}_{i+1} = \tilde{\mathbf{r}}_i - \alpha_i A^T A \mathbf{p}_i$   
 $\beta_i = \frac{(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_i)}{(\tilde{\mathbf{r}}_{i-1}, \tilde{\mathbf{r}}_{i-1})}$   
 $\mathbf{p}_{i+1} = \tilde{\mathbf{r}}_{i+1} + \beta_i \mathbf{p}_i$   
end

Iterative methods generally require relatively small memory and may be computationally efficient. However, this is true only when the convergence is sufficiently fast. The condition number of  $A^T A$  is square of the condition number of  $A$ . Hence, the convergence may be slow for ill-conditioned problems so that preconditioning becomes necessary. As preconditioners, diagonal scaling, incomplete Cholesky decomposition, and incomplete QR decomposition are commonly used[2]. Recently, a robust preconditioner was proposed in [1]. In this paper, we will look at preconditioning methods based on incomplete QR decomposition, in comparing the use of the GMRES method to the use of the CG method for least squares problems.

### 2.3 Incomplete QR decomposition

The incomplete QR decomposition as a preconditioner was proposed by Jennings[6], and it can be classified roughly into two ways. One is the IMGS (Incomplete Modified Gram-Schmidt decomposition) method in which the incomplete decomposition is realized by an incomplete modified Gram-Schmidt method. The other is the IG (Incomplete Givens) method in which the incomplete decomposition is realized by Givens rotations[6].

The IMGS method uses a threshold for the elements in  $R$ , where the magnitude of each off-diagonal element  $r_{ij}$  is compared against a threshold  $\tau$  times the norm of the corresponding column norm  $d_i \equiv \|\mathbf{a}_i\|_2$ , i.e., elements which satisfy  $\|r_{ij}\|_2 < \tau d_i$  are dropped, where  $A = [\mathbf{a}_1, \dots, \mathbf{a}_n]$ [2]. In the IMGS method,  $A$  is approximated by  $QR$ , where  $R$  is an upper triangular matrix with positive diagonal elements, and

$$\langle \mathbf{q}_1, \dots, \mathbf{q}_n \rangle = \langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle,$$

where  $\langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle$  denotes the subspace spanned by  $\mathbf{a}_1, \dots, \mathbf{a}_n$ . The algorithm of the IMGS method is as follows, where  $\mathbf{a}_i^{(1)} \equiv \mathbf{a}_i$  ( $i = 1, \dots, n$ ).

**Algorithm 2.2** : *IMGS method (Jennings' version)*

```

for  $i = 1, 2, \dots, n$ 
     $r_{ii} = \|\mathbf{a}_i^{(i)}\|_2, \quad \mathbf{q}_i = \frac{\mathbf{a}_i^{(i)}}{r_{ii}}$ 
    for  $j = i + 1, \dots, n$ 
         $r_{ij} = \mathbf{q}_i^\top \mathbf{a}_j^{(i)}$ 
        If  $|r_{ij}| < \tau d_i$ , then  $r_{ij} = 0$ , end.
         $\mathbf{a}_j^{(i+1)} = \mathbf{a}_j^{(i)} - r_{ij} \mathbf{q}_i$ 
    end
end

```

If  $A$  is full column rank, this algorithm never breaks down.

Saad proposed an alternative algorithm in which the  $p_Q$  largest (in absolute value) elements in a column of  $Q$  and the  $p_R$  largest elements in a row of  $R$  are kept, where  $p_Q$  and  $p_R$  are two chosen parameters[8]. This algorithm is given as follows.

**Algorithm 2.3** : *IMGS method (Saad's version)*

for  $i = 1, 2, \dots, n$   
 $r_{ii} = \|\mathbf{a}_i^{(i)}\|_2, \quad \mathbf{q}_i = \frac{\mathbf{a}_i^{(i)}}{r_{ii}}$   
*Determine the  $p_Q$  largest elements of  $\mathbf{q}_i$*   
*and assign a zero value to the other elements.*  
for  $j = i + 1, \dots, n$   
 $r_{ij} = \mathbf{q}_i^\top \mathbf{a}_j^{(i)}$   
*Determine the  $p_R$  largest  $r_{ij}$ 's for  $i + 1 \leq j \leq n$ ,*  
*and assign a zero value to the others.*  
 $\mathbf{a}_j^{(i+1)} = \mathbf{a}_j^{(i)} - r_{ij} \mathbf{q}_i$   
end  
end

When these IMGS preconditioners are applied to the normal equation, we have

$$R^{-\top} A^\top A R^{-1} R \mathbf{x} = R^{-\top} A^\top \mathbf{b},$$

or

$$\tilde{A} \tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \quad \tilde{\mathbf{x}} = R \mathbf{x}, \quad (2.3)$$

where  $\tilde{A} = R^{-\top} A^\top A R^{-1}$ ,  $\tilde{\mathbf{x}} = R \mathbf{x}$ ,  $\tilde{\mathbf{b}} = R^{-\top} A^\top \mathbf{b}$ .

Then, the conjugate gradient (CG) method is applied to (2.3).

## 2.4 CR-LS( $k$ ) method

Zhang and Oyanagi[10, 11, 12] proposed a different type of method in which they applied the Orthomin( $k$ ) method to (1.1) by introducing a  $n \times m$  mapping matrix  $B$ , instead of solving the normal equation, so that the condition number becomes smaller and improved convergence of the iterative method is expected. This method is called the CR-LS( $k$ ) method. The algorithm of the CR-LS( $k$ ) method is as follows.

**Algorithm 2.4** : *CR-LS( $k$ ) method*

Choose  $\mathbf{x}_0$ .  
 $\mathbf{r}_0 = \mathbf{b} - A \mathbf{x}_0, \quad \mathbf{p}_0 = B \mathbf{r}_0$

for  $i = 0, 1, 2, \dots$  until convergence

$$\alpha_i = \frac{(\mathbf{r}_i, A\mathbf{p}_i)}{(A\mathbf{p}_i, A\mathbf{p}_i)}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i A\mathbf{p}_i$$

for  $j = 0$  to  $\min(k-1, i)$

$$\beta_{i,i-j} = -\frac{(A\mathbf{B}\mathbf{r}_{i+1}, A\mathbf{p}_{i-j})}{(A\mathbf{p}_{i-j}, A\mathbf{p}_{i-j})}$$

end

$$\mathbf{p}_{i+1} = B\mathbf{r}_i + \sum_{j=0}^{\min(k-1, i)} \beta_{i,i-j} \mathbf{p}_{i-j}$$

end

### 3 The GMRES( $k$ )-LS method

The GMRES( $k$ ) method[9] is an efficient and robust Krylov subspace method for solving systems of linear equations

$$A\mathbf{x} = \mathbf{b},$$

where  $A$  is square, nonsingular and nonsymmetric.

In this section, we propose algorithms which apply the GMRES( $k$ ) method to the linear least squares problem (1.1) by using a  $n \times m$  mapping matrix  $B$ .

In a linear least squares problem, we seek the solution  $\mathbf{x}^*$  which minimizes the objective function

$$S(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_2^2$$

of  $n$  variables  $\mathbf{x} = (x_1, x_2, \dots, x_n)^\top$ , where  $A$  is a  $m \times n$  ( $m \geq n$ ) matrix, and  $\mathbf{b}$  is a  $m$ -dimensional vector.

#### 3.1 GMRES( $k$ )-LS method 1

If one were to apply the GMRES( $k$ ) method directly to the linear least squares problem in which  $A$  is a  $m \times n$  rectangular matrix, and the initial residual  $\mathbf{r}_0$  is a  $m$ -dimensional vector, one cannot create a Krylov subspace, just by multiplying  $\mathbf{r}_0$  by  $A$ .

There are two possible ways of overcoming this problem. The first is to use a  $n \times m$  mapping matrix  $B$  to create a Krylov subspace

$$\langle \mathbf{r}_0, AB\mathbf{r}_0, \dots, (AB)^{i-1}\mathbf{r}_0 \rangle$$

in the (larger)  $m$ -dimensional space, where  $AB$  is a  $m \times m$  matrix, as in the CR-LS( $k$ ) method[10, 11, 12], and to apply the GMRES( $k$ ) method using this Krylov subspace.

This gives the following algorithm.

**Algorithm 3.1** : *GMRES( $k$ )-LS method 1*

- Choose  $\mathbf{x}_0$ .
- \*  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
- $\beta = \|\mathbf{r}_0\|_2, \quad \mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta}$
- For  $i = 1, 2, \dots, k$  until convergence
- $h_{j,i} = (AB\mathbf{v}_i, \mathbf{v}_j) \quad (j = 1, 2, \dots, i)$
- $\hat{\mathbf{v}}_{i+1} = AB\mathbf{v}_i - \sum_{j=1}^i h_{j,i}\mathbf{v}_j$
- $h_{i+1,i} = \|\hat{\mathbf{v}}_{i+1}\|_2$
- $\mathbf{v}_{i+1} = \frac{\hat{\mathbf{v}}_{i+1}}{h_{i+1,i}}$
- Find  $\mathbf{y}_i \in \mathbf{R}^i$  which minimizes  $\|\mathbf{r}_i\|_2 = \|\beta\mathbf{e}_i - \bar{H}_i\mathbf{y}\|_2$



*end*  
 $\mathbf{x}_k = \mathbf{x}_0 + B[\mathbf{v}_1, \dots, \mathbf{v}_k]\mathbf{y}_k$   
 $\mathbf{x}_0 = \mathbf{x}_k$   
*Go to \*.*

Here,  $\bar{H}_i = (h_{pq}) \in \mathbf{R}^{(i+1) \times i}$ , and  $\mathbf{e}_i = (1, 0, \dots, 0)^\top \in \mathbf{R}^{i+1}$ .

Note here that

$$\min_{\mathbf{x} \in \mathbf{R}^n} \|\mathbf{b} - A\mathbf{x}\|_2 \leq \min_{\mathbf{z} \in \mathbf{R}^m} \|\mathbf{b} - AB\mathbf{z}\|_2, \quad (3.1)$$

where the equality holds if and only if

$$\text{rank } B = n.$$

This is because  $\mathbf{x}^*$  gives the minimum of the left hand side of (3.1) if and only if

$$A^\top A\mathbf{x}^* - A^\top \mathbf{b} = \mathbf{0},$$

where as  $\mathbf{z}^*$  gives the minimum of the right hand side of (3.1) if and only if

$$B^\top (A^\top AB\mathbf{z}^* - A^\top \mathbf{b}) = \mathbf{0}.$$

Hence,

$$\begin{aligned} \text{rank } B = \text{rank } B^\top = n &\iff A^\top AB\mathbf{z}^* - A^\top \mathbf{b} = \mathbf{0} \\ \iff B\mathbf{z}^* &\text{ is a least squares solution of the left hand side of (3.1).} \end{aligned}$$

Therefore, we will assume that  $\text{rank } B = n$ . Then, the above GMRES( $k$ )-LS method 1 is equivalent to applying the GMRES( $k$ ) method to the system

$$AB\mathbf{z} = \mathbf{b}$$

with initial approximation  $\mathbf{z}_0$ , such that

$$B\mathbf{z}_0 = \mathbf{x}_0.$$

Note that if  $\text{rank } B = n$ , there exists  $\mathbf{z}_0$  such that  $B\mathbf{z}_0 = \mathbf{x}_0$ . This is because if we denote

$$B = [\mathbf{b}_1, \dots, \mathbf{b}_n]^\top,$$

and assume that  $\text{rank } B = n$ , there exist  $\mathbf{b}_{n+1}, \dots, \mathbf{b}_m$  such that

$$\tilde{B} = [\mathbf{b}_1, \dots, \mathbf{b}_m]^\top$$

is nonsingular. Then

$$\mathbf{z}_0 = \tilde{B}^{-1} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{t} \end{bmatrix}$$

satisfies  $B\mathbf{z}_0 = \mathbf{x}_0$ , where  $\mathbf{t} \in \mathbf{R}^{m-n}$  is an arbitrary vector. Note that the line

$$\mathbf{x}_k = \mathbf{x}_0 + B[\mathbf{v}_1, \dots, \mathbf{v}_k]\mathbf{y}_k$$

in Algorithm 3.1 corresponds to

$$\mathbf{z}_k = \mathbf{z}_0 + [\mathbf{v}_1, \dots, \mathbf{v}_k]\mathbf{y}_k,$$

where

$$\mathbf{x}_k = B\mathbf{z}_k$$

and

$$\mathbf{x}_0 = B\mathbf{z}_0.$$

Then, we have the following theorem.

**Theorem 3.1** *If  $\text{rank } A = \text{rank } B = n$ , the GMRES-LS method 1 determines the least squares solution of*

$$\min_{\mathbf{x} \in \mathbf{R}^n} \|\mathbf{b} - A\mathbf{x}\|_2 \quad (1.1)$$

for arbitrary  $\mathbf{b}$  and  $\mathbf{x}_0$ , if and only if  $B$  can be expressed as

$$B = CA^\top \quad (3.2)$$

where  $C$  is a nonsingular matrix.

Note that the GMRES-LS method 1 corresponds to GMRES( $\infty$ )-LS method 1 (no restarts).

**Proof.** It is sufficient to prove that the GMRES method determines the least squares solution of

$$\min_{\mathbf{z} \in \mathbf{R}^m} \|\mathbf{b} - AB\mathbf{z}\|_2$$

for arbitrary  $\mathbf{b}$  and  $\mathbf{z}_0$ , where  $B\mathbf{z}_0 = \mathbf{x}_0$ .

From Theorem 2.4 of [3], it is sufficient to prove that

$$\mathcal{N}(AB) = \mathcal{N}(B^\top A^\top),$$

where  $\mathcal{N}(\cdot)$  denotes the null space. Let

$$A = [\mathbf{a}_1, \dots, \mathbf{a}_n]$$

and

$$B^\top = [\mathbf{b}_1, \dots, \mathbf{b}_n].$$

Then, note that

$$\begin{aligned} \mathbf{z} \in \mathcal{N}(AB) &\iff AB\mathbf{z} = \mathbf{0} \\ &\iff B\mathbf{z} = \mathbf{0} \\ &\iff \mathbf{z} \perp \langle \mathbf{b}_1, \dots, \mathbf{b}_n \rangle, \end{aligned}$$

where the second equivalence is due to  $\text{rank } A = n$ . Note also that,

$$\begin{aligned} \mathbf{z} \in \mathcal{N}(B^\top A^\top) &\iff B^\top A^\top \mathbf{z} = \mathbf{0} \\ &\iff A^\top \mathbf{z} = \mathbf{0} \\ &\iff \mathbf{z} \perp \langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle, \end{aligned}$$

where the second equivalence is due to  $\text{rank } B^\top = \text{rank } B = n$ .

Hence,

$$\begin{aligned} \mathcal{N}(AB) = \mathcal{N}(B^\top A^\top) &\iff \langle \mathbf{b}_1, \dots, \mathbf{b}_n \rangle = \langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle \\ &\iff B^\top = AC^\top \text{ where } C^\top \text{:nonsingular} \\ &\iff B = CA^\top \text{ where } C \text{:nonsingular.} \end{aligned}$$

□

Here we note that there is a related method of applying the GMRES method to overdetermined systems due to Calvetti, Lewis and Reichel[4]. The method essentially appends  $m - n$  zero columns to  $A$  to obtain a  $m \times m$  square singular matrix

$$\tilde{A} = [A, 0],$$

and then applies the GMRES method to the system

$$\tilde{A}\mathbf{z} = \mathbf{b}.$$

This method can be considered as a special case of our GMRES( $k$ )-LS method 1, where

$$B = [I_n, 0].$$

However, this special choice of  $B$  does not necessarily meet the condition (3.2) in Theorem 3.1, so that their method may break down before reaching the least squares solution of (1.1).

### 3.2 GMRES( $k$ )-LS method 2

The other alternative is to use the  $n \times m$  mapping matrix  $B$  to map  $\mathbf{r}_0$  to

$$\tilde{\mathbf{r}}_0 = B\mathbf{r}_0,$$

and then to create a Krylov subspace

$$\langle \tilde{\mathbf{r}}_0, BA\tilde{\mathbf{r}}_0, \dots, (BA)^{i-1}\tilde{\mathbf{r}}_0 \rangle$$

in the (smaller)  $n$ -dimensional space, where  $BA$  is a  $n \times n$  matrix, and to apply the GMRES( $k$ ) method using this Krylov subspace as follows.

**Algorithm 3.2** : *GMRES(k)-LS method 2*

Choose  $\mathbf{x}_0$ .

\*  $\tilde{\mathbf{r}}_0 = B(\mathbf{b} - A\mathbf{x}_0)$

$\beta = \|\tilde{\mathbf{r}}_0\|_2, \quad \mathbf{v}_1 = \frac{\tilde{\mathbf{r}}_0}{\beta}$

for  $i = 1, 2, \dots, k$  until convergence

$h_{j,i} = (BA\mathbf{v}_i, \mathbf{v}_j) \quad (j = 1, 2, \dots, i)$

$\hat{\mathbf{v}}_{i+1} = BA\mathbf{v}_i - \sum_{j=1}^i h_{j,i} \mathbf{v}_j$

$h_{i+1,i} = \|\hat{\mathbf{v}}_{i+1}\|_2$

$\mathbf{v}_{i+1} = \frac{\hat{\mathbf{v}}_{i+1}}{h_{i+1,i}}$

Find  $\mathbf{y}_i \in \mathbf{R}^i$  which minimizes  $\|\tilde{\mathbf{r}}_i\|_2 = \|\beta \mathbf{e}_i - \bar{H}_i \mathbf{y}\|_2$

end

$\mathbf{x}_k = \mathbf{x}_0 + [\mathbf{v}_1, \dots, \mathbf{v}_k] \mathbf{y}_k$

$\mathbf{x}_0 = \mathbf{x}_k$

Go to \*.

Here,  $\bar{H}_i$ , and  $\mathbf{e}_i$  are defined as in Algorithm 3.1.

As will be shown in the numerical experiments, it turns out that the GMRES(k)-LS method 2, which uses the Krylov space

$$\langle \tilde{\mathbf{r}}_0, BA\tilde{\mathbf{r}}_0, \dots, (BA)^{i-1}\tilde{\mathbf{r}}_0 \rangle$$

in the (smaller)  $n$ -dimensional space requires less computational work compared to the GMRES(k)-LS method 1.

Note that the GMRES(k)-LS method 2 is equivalent to applying the GMRES(k) method to

$$BA\mathbf{x} = B\mathbf{b}, \tag{3.3}$$

where  $BA$  is a  $n \times n$  matrix.

The following theorem gives the necessary and sufficient condition for the mapping matrix  $B$ , in order that equation (3.3) is equivalent to the original least squares problem (1.1).

**Theorem 3.2** *The solution of*

$$BA\mathbf{x} = B\mathbf{b}, \tag{3.3}$$

*is a (least squares) solution of*

$$\min_{\mathbf{x} \in \mathbf{R}^n} \|\mathbf{b} - A\mathbf{x}\|_2 \tag{1.1}$$

if and only if  $B$  can be expressed as

$$B = CA^\top, \quad (3.4)$$

where  $C$  is a nonsingular matrix.

**Proof.**  $\mathbf{x}$  is a least squares solution of

$$\min_{\mathbf{x} \in \mathbf{R}^n} \|\mathbf{b} - A\mathbf{x}\|_2 \quad (1.1)$$

if and only if  $\mathbf{x}$  satisfies the normal equation

$$A^\top A\mathbf{x} = A^\top \mathbf{b}. \quad (1.2)$$

which is equivalent to

$$A^\top \mathbf{r} = \mathbf{0} \quad (3.5)$$

where  $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ .

(3.5), in turn, is equivalent to

$$\langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle \perp \mathbf{r} \quad (3.6)$$

where  $\langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle$  is the subspace spanned by the column vectors  $\mathbf{a}_1, \dots, \mathbf{a}_n$  of  $A$ .

On the other hand,

$$BA\mathbf{x} = B\mathbf{b}, \quad (3.3)$$

is equivalent to

$$B\mathbf{r} = \mathbf{0},$$

which, in turn, is equivalent to

$$\langle \mathbf{b}_1, \dots, \mathbf{b}_n \rangle \perp \mathbf{r} \quad (3.7)$$

where  $\mathbf{b}_1, \dots, \mathbf{b}_n$  are the column vectors of  $B^\top$ .

Now, (3.6) and (3.7) are equivalent, if and only if  $\langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle$  and  $\langle \mathbf{b}_1, \dots, \mathbf{b}_n \rangle$  are the same subspace, which is true, if and only if

$$A = [\mathbf{a}_1, \dots, \mathbf{a}_n] = [\mathbf{b}_1, \dots, \mathbf{b}_n] C' = B^\top C',$$

where  $C'$  is a nonsingular  $n \times n$  matrix.

Hence, (3.3) and (1.2) are equivalent if and only if

$$A = B^\top C',$$

where  $C'$  is a nonsingular  $n \times n$  matrix, or, in other words,

$$AC'^{-1} = B^\top$$

or

$$B = C'^{-\top} A^\top.$$

Thus, the solution of (3.3) is a least squares solution of (1.1), if and only if

$$B = CA^\top, \quad (3.4)$$

where  $C = C'^{-\top}$  is nonsingular.  $\square$

### 3.3 Incomplete QR decomposition

Besides satisfying the condition (3.2) or (3.4), it is natural to choose a mapping matrix  $B$  which satisfies  $BA \approx I_n$ , so that we may expect fast convergence of the iterative method.

In [12],  $B = (\text{diag}(A^\top A))^{-1}A^\top$  was proposed for the CR-LS( $k$ ) method, where  $\text{diag}(A^\top A)$  is the diagonal matrix obtained by taking the diagonal part of  $A^\top A$ , and is nonsingular when  $A$  is full rank. It is obvious that this choice of  $B$  fulfills the condition (3.2) and (3.4). For this  $B$ , (3.3) is equivalent to carrying out the preconditioning of row diagonal scaling to the normal equation (1.2).

In this paper, we consider using the incomplete QR decomposition of  $A$  given by

$$A = QR + E,$$

where  $Q$  is a  $m \times n$  matrix,  $R$  is a  $n \times n$  upper triangular matrix, and  $E$  is the error matrix. The column vectors of  $Q$  may not be perfectly orthonormal. Previous methods for solving the linear least squares problem (1.1) using the incomplete QR decomposition used  $R$  as the preconditioning matrix for the normal equation  $A^\top A \mathbf{x} = A^\top \mathbf{b}$  [6]. That is, to solve

$$(R^{-\top} A^\top A R^{-1}) R \mathbf{x} = R^{-\top} A^\top \mathbf{b}, \quad (3.8)$$

or

$$\tilde{A} \tilde{\mathbf{x}} = \tilde{\mathbf{b}}$$

where  $\tilde{A} = R^{-\top} A^\top A R^{-1}$ ,  $\tilde{\mathbf{x}} = R \mathbf{x}$ ,  $\tilde{\mathbf{b}} = R^{-\top} A^\top \mathbf{b}$ , using the conjugate gradient method.

In our case, we set  $B = R^{-1}Q^\top$ . In the GMRES( $k$ )-LS method 2, this is equivalent to solving

$$R^{-1}Q^\top A \mathbf{x} = R^{-1}Q^\top \mathbf{b} \quad (3.9)$$

using the GMRES( $k$ ) method. We expect that (3.9) is better conditioned compared to (3.8).

If we take  $E = 0$ , the QR decomposition is performed completely, i.e.  $A = QR$ , so that  $\mathbf{x} = R^{-1}Q^\top \mathbf{b}$ , and only one iteration is required, but the computational work and storage would become prohibitive for large problems. Hence, we set  $E \neq 0$ , and do an incomplete QR decomposition. The nearer the incomplete QR decomposition is to the complete QR decomposition, the required number of iterations will decrease, but more time and memory would be required to do the incomplete QR decomposition. Hence, we need to strike a balance.

Consider the QR decomposition using the modified Gram-Schmidt process. In this paper, we will realize the incomplete QR decomposition by making the current column vector of  $Q$  to be orthogonal to the previous  $l$  column vectors, where the incompleteness of the decomposition is adjusted by the parameter  $l$ . If  $l = n - 1$ , it is equivalent to the complete QR decomposition. We call this procedure the IMGSL( $l$ ) method. The algorithm is as follows.

**Algorithm 3.3** : *IMGS*( $l$ ) method

```

for  $i = 1, 2, \dots, n$ 
   $r_{ii} = \|\mathbf{a}_i\|_2, \quad \mathbf{q}_i = \frac{\mathbf{a}_i}{r_{ii}}$ 
  for  $j = i + 1, \dots, \min(i + l, n)$ 
     $r_{ij} = \mathbf{q}_i^\top \mathbf{a}_j$ 
     $\mathbf{a}_j = \mathbf{a}_j - r_{ij} \mathbf{q}_i$ 
  end
end
end

```

At each step of the *IMGS*( $l$ ) method, the column vector of  $Q$  is computed as a linear combination of the column vectors of  $A$  such that

$$\langle \mathbf{q}_1, \dots, \mathbf{q}_n \rangle = \langle \mathbf{a}_1, \dots, \mathbf{a}_n \rangle,$$

where  $\langle \mathbf{q}_1, \dots, \mathbf{q}_n \rangle$  denotes the subspace spanned by the vectors  $\mathbf{q}_1, \dots, \mathbf{q}_n$ .

Thus, we have  $Q = A\tilde{C}$ , where  $\tilde{C}$  is a nonsingular matrix. Thus, we have

$$Q^\top = \tilde{C}^\top A^\top \tag{3.10}$$

where  $\tilde{C}^\top$  is nonsingular.

Now, consider

$$B \equiv R^{-1}Q^\top \tag{3.11}$$

obtained by the *IMGS*( $l$ ) decomposition. Since  $r_{ii} \neq 0$ ,  $i = 1, \dots, n$ ,  $R = (r_{ij})$  is nonsingular. From (3.11) and (3.10), we have

$$B = R^{-1}Q^\top = R^{-1}\tilde{C}^\top A^\top = CA^\top,$$

where  $C = R^{-1}\tilde{C}^\top$  is nonsingular. Hence,  $B$  satisfies the condition (3.2) and (3.4).

Here, we note that *IMGS*(0) is equivalent to the diagonal scaling proposed in [12].

**Lemma.** The *IMGS*(0) method is equivalent to the diagonal scaling given by  $B = (\text{diag}(A^\top A))^{-1}A^\top$ .

**Proof.** From Algorithm 3.3, the *IMGS*(0) method is given by

```

for  $i = 1, 2, \dots, n$ 
   $r_{ii} = \|\mathbf{a}_i\|_2, \quad \mathbf{q}_i = \frac{\mathbf{a}_i}{r_{ii}}$ 
end

```

which gives

$$R = \text{diag}(\|\mathbf{a}_1\|_2, \dots, \|\mathbf{a}_n\|_2)$$

and

$$Q = [\mathbf{a}_1/\|\mathbf{a}_1\|_2, \dots, \mathbf{a}_n/\|\mathbf{a}_n\|_2],$$

so that

$$B = R^{-1}Q^\top = \begin{bmatrix} \mathbf{a}_1^\top/\|\mathbf{a}_1\|_2^2 \\ \vdots \\ \mathbf{a}_n^\top/\|\mathbf{a}_n\|_2^2 \end{bmatrix}.$$

On the other hand,

$$\text{diag}(A^\top A) = \text{diag}(\|\mathbf{a}_1\|^2, \dots, \|\mathbf{a}_n\|^2),$$

so that

$$\left(\text{diag}(A^\top A)\right)^{-1} A^\top = \begin{bmatrix} \mathbf{a}_1^\top/\|\mathbf{a}_1\|_2^2 \\ \vdots \\ \mathbf{a}_n^\top/\|\mathbf{a}_n\|_2^2 \end{bmatrix}.$$

Hence,

$$B = R^{-1}Q^\top = \left(\text{diag}(A^\top A)\right)^{-1} A^\top.$$

□



## 4 Numerical experiments

In this section, we give numerical experiment results solving linear least squares problems

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2, \quad \mathbf{A} \in \mathbf{R}^{m \times n}, \quad m \geq n,$$

in order to demonstrate the effectiveness of the proposed method. The following sparse matrices were used for  $\mathbf{A}$ .

The first set of matrices were taken from the Harwell-Boeing test collection which is widely used in numerical experiments on linear least squares problems. These matrices are available from Matrix Market[7]. The matrices of the form “ARTF $n$ nnn” are artificial matrices made of two Harwell-Boeing matrices such as

$$\mathbf{A} = \begin{pmatrix} \text{WELL}_{nnnn} \\ 0 \quad \text{ASH}_{nnn} \end{pmatrix}.$$

The characteristics of these matrices are given in Table 4.1, where ‘Condition’ denotes the condition number, and ‘Density’ is the percentage of non-zero elements. The elements of the matrices “ABB” and “ASH” are random numbers uniformly distributed in  $[-1, 1]$ . The elements of the matrices “WELL” and “ILLC” are the same as the original. The structure of the non-zero elements of the matrices “WELL” and “ILLC” are the same, but the values of the elements are different.

Table 4.1: Characteristics of test matrices (1).

No	Name	$m$	$n$	Density	Condition	Description
1	WELL1033	1033	320	1.4%	$2 \times 10^2$	Gravity-meter obs.
2	ILLC1033	1033	320	1.4%	$2 \times 10^4$	“
3	ARTF1252	1252	320	1.3%	$4 \times 10^1$	WELL1033+ASH219
4	ARTF1346	1346	320	1.2%	$4 \times 10^1$	WELL1033+ABB313
5	ARTF1641	1641	320	1.1%	$4 \times 10^0$	WELL1033+ASH608
6	ARTF1991	1991	320	1.0%	$4 \times 10^0$	WELL1033+ASH958
7	ARTF2808	2808	712	0.53%	$3 \times 10^1$	WELL1850+ASH958

The second set of matrices are newly made artificial matrices. These matrices were generated in order to test the methods for problems with varying condition numbers. They were produced by the MATLAB command “sprandn” which makes a  $m \times n$  nonsymmetric matrix of a certain density and condition number. The non-zero elements of the matrices are random numbers of normal distribution, whose mean is zero and variance is one, and the pattern of the non-zero elements is also random. The characteristics of these matrices are shown in Table 4.2.

Table 4.2: Characteristics of test matrices (2).

No	Name	$m$	$n$	Density	Condition	Description
8	RANDS2	1000	320	4.9%	$2 \times 10^2$	“sprandn”
9	RANDS4	1000	320	4.9%	$1 \times 10^4$	”
10	RANDS6	1000	320	4.9%	$1 \times 10^6$	”
11	RANDS8	1000	320	4.9%	$1 \times 10^8$	”

The third and fourth set of matrices are also newly made artificial matrices including even larger matrices. These matrices were also generated in order to test the methods for problems with varying condition numbers. We produced the matrices by the MATLAB command “sprandn” as in the previous set. The characteristics of these matrices are shown in Table 4.3 and Table 4.4.

Using different methods, we solved the linear least squares problem corresponding to  $A\mathbf{x} = \mathbf{b}$  that has the right hand side  $\mathbf{b}$  computed from the exact solution  $\mathbf{x} = (1, \dots, 1)^\top$ . In order to perform a fair comparison among different methods, in all the following experiments, convergence of the numerical solution was assessed by comparing them to the exact solution. More specifically, the criterion

$$\frac{\|\mathbf{x} - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2} < 10^{-6}$$

using the relative error between the approximated solution  $\mathbf{x}$  and the exact solution  $\mathbf{x}^*$ , was used. The initial approximation was set to  $\mathbf{x}_0 = \mathbf{0}$ . In practice, convergence can be monitored using the (relative) residual corresponding to each method.

Table 4.3: Characteristics of test matrices (3).

No	Name	$m$	$n$	Density	Condition	Description
12	RANDM1	3000	1000	4.9%	$8 \times 10^1$	“sprandn”
13	RANDM2	3000	1000	4.9%	$6 \times 10^2$	”
14	RANDM3	3000	1000	4.9%	$2 \times 10^3$	”
15	RANDM4	3000	1000	4.9%	$2 \times 10^4$	”
16	RANDM5	3000	1000	4.9%	$2 \times 10^5$	”
17	RANDM6	3000	1000	4.9%	$2 \times 10^6$	”
18	RANDM7	3000	1000	4.9%	$2 \times 10^7$	”
19	RANDM8	3000	1000	4.9%	$1 \times 10^8$	”

Table 4.4: Characteristics of test matrices (4).

No	Name	$m$	$n$	Density	Condition	Description
20	RANDL1	10000	1000	1.5%	$6 \times 10^1$	“sprandn”
21	RANDL2	10000	1000	1.5%	$4 \times 10^2$	”
22	RANDL3	10000	1000	1.5%	$3 \times 10^3$	”
23	RANDL4	10000	1000	1.5%	$3 \times 10^4$	”
24	RANDL5	10000	1000	1.5%	$2 \times 10^5$	”
25	RANDL6	10000	1000	1.5%	$2 \times 10^6$	”
26	RANDL7	10000	1000	1.5%	$2 \times 10^7$	”

The following methods were compared.

- (1) The CGLS method, which is equivalent to applying the conjugate gradient (CG) method to the normal equation

$$A^\top A \mathbf{x} = A^\top \mathbf{b}.$$

- (2) The method of applying the CG method to the normal equation using the IMGS preconditioning (Jennings' version)[2],[6]. It is equivalent to applying the CG method to

$$R^{-\top} A^\top A R^{-1} R \mathbf{x} = R^{-\top} A^\top \mathbf{b},$$

or to

$$\tilde{A} \tilde{\mathbf{x}} = \tilde{\mathbf{b}},$$

where  $\tilde{A} = R^{-\top} A^\top A R^{-1}$ ,  $\tilde{\mathbf{x}} = R \mathbf{x}$ , and  $\tilde{\mathbf{b}} = R^{-\top} A^\top \mathbf{b}$ .

- (3) The method of using the IMGS( $l$ ) method of section 3.3 for the preconditioning, in place of the IMGS method in (2).
- (4) The proposed GMRES( $k$ )-LS method 1 with the IMGS( $l$ ) as the preconditioner for  $B = R^{-1} Q^\top$ .
- (5) The proposed GMRES( $k$ )-LS method 2 with the IMGS( $l$ ) as the preconditioner for  $B = R^{-1} Q^\top$ . It is equivalent to solving

$$R^{-1} Q^\top A \mathbf{x} = R^{-1} Q^\top \mathbf{b}$$

by the GMRES( $k$ ) method.

Saad's version of the IMGS (Algorithm 2.3) was also tested with various parameters, but it required more computational time compared to Jennings' version, so we do not give the results here. We coded the algorithm in MATLAB 6 and ran the experiments on a NEC PC with mobile AMD AthlonXP 2000+, 736MB RAM.

First, we tested the IMGS method (Jennings' version) and the IMGS( $l$ ) method in combination with the GMRES-LS method 2, changing the parameter values for the preconditioners. The results are shown in Table 4.5 and Table 4.6. The results were obtained using the matrix "WELL1033" and "RANDS6", respectively. The results show the computational time and the number of iterations required to obtain relative error less than  $10^{-6}$ .

Table 4.5: Computation time required by different parameter values for incomplete QR decomposition (WELL1033, time in seconds).

Method	preconditioning time	iteration time	number of iterations	total time
Jennings' IMGS				
$\tau = 1$	2.94	1.01	98	3.95
$\tau = 0.5$	2.99	1.02	99	4.03
$\tau = 0.4$	2.98	1.04	100	4.02
$\tau = 0.3$	2.98	1.13	104	4.01
$\tau = 0.2$	3.04	0.99	95	4.03
$\tau = 0.1$	3.36	0.67	67	4.03
$\tau = 0.07$	4.72	1.02	54	5.74
$\tau = 0.01$	7.61	0.41	17	8.02
$\tau = 0$	8.77	0.09	1	8.86
IMGS( $l$ )				
$l = 0$	0.02	0.82	98	0.84
$l = 1$	0.07	1.20	100	1.27
$l = 10$	0.60	1.85	104	2.45
$l = 50$	2.40	2.75	113	5.15
$l = 100$	3.97	2.30	97	6.27
$l = 150$	5.97	2.06	83	8.03
$l = 200$	8.20	1.16	47	9.36
$l = 250$	8.30	0.62	26	8.92
$l = 300$	8.40	0.30	11	8.70
$l = 320$	8.45	0.10	1	8.55

Table 4.6: Computation time required by different parameter values for incomplete QR decomposition (RANDS6, time in seconds).

Method	preconditioning time	iteration time	number of iterations	total time
Jennings' IMGS				
$\tau = 1$	5.50	21.94	320	27.44
$\tau = 0.5$	7.59	23.30	320	30.89
$\tau = 0.4$	8.51	23.82	320	32.33
$\tau = 0.3$	9.82	23.81	317	33.63
$\tau = 0.2$	12.60	21.76	302	34.36
$\tau = 0.1$	17.88	20.67	287	38.55
$\tau = 0.07$	22.11	18.82	270	40.93
$\tau = 0.01$	47.66	12.33	200	59.99
$\tau = 0$	170.08	0.34	1	170.42
IMGS( $l$ )				
$l = 0$	0.04	17.70	320	17.74
$l = 1$	1.40	31.18	320	32.58
$l = 10$	3.11	33.93	320	37.04
$l = 50$	15.61	33.88	320	49.49
$l = 100$	40.50	34.04	320	74.54
$l = 150$	73.24	25.01	277	98.25
$l = 200$	113.50	13.14	193	126.64
$l = 250$	149.22	6.27	110	155.49
$l = 300$	170.61	1.44	25	172.05
$l = 320$	169.18	0.33	1	169.51

In the IMGS method (Jennings’ version), the smaller the threshold parameter  $\tau$  becomes, the less incomplete the incomplete QR decomposition becomes, and the number of required iterations tends to decrease, but more time is required for the incomplete decomposition. However, the effect of changing  $\tau$  is relatively small. For the following experiments, we fixed  $\tau$  at 0.1 for the method (2).

Similarly, for the IMGS( $l$ ) method, as  $l$  is increased, the incompleteness of the incomplete QR decomposition becomes small, and the number of iterations tends to decrease, but the computation time for the decomposition increases. It turns out that  $l = 0$ , which is equivalent to the diagonal scaling  $B = (\text{diag}(A^T A))^{-1} A^T$  is optimal in terms of total time.

Among the different preconditioning methods compared, the IMGS( $l$ ) method with  $l = 0$  was the fastest. For all the following experiments,  $l$  was fixed at  $l = 0$  for the IMGS( $l$ ) method.

Table 4.7: Results for changing  $k$  of the GMRES( $k$ )-LS method 2.

RANDL2	$k$	10	50	100	150	200	250	$\geq 262$
	iter	559	369	298	306	287	279	262
	time	35.93	23.65	21.69	25.30	26.01	30.95	31.75
RANDL3	$k$	50	100	200	300	400	500	$\geq 574$
	iter	1336	988	898	743	706	652	574
	time	83.20	69.67	79.05	83.86	110.41	148.94	204.79
RANDL4	$k$	400	500	600	700	800	900	$\geq 993$
	iter	41149	29909	19095	22379	15168	9802	993
	time	7247.13	5525.54	4785.23	6690.14	6013.33	4113.34	1211.83
RANDL5	$k$	400	500	600	700	800	900	$\geq 998$
	iter	46704	25971	28086	23767	16566	9563	998
	time	8170.12	4752.82	6912.20	7739.61	4955.51	3570.22	1228.72

Next, we demonstrate the effect of changing the restarting cycle  $k$  of the GMRES( $k$ )-LS method 2 in Table 4.7. In each cell of the table, the middle number (iter) is the number of iterations, and the bottom number (time) is the computation time (in seconds). Although restarting is effective for less ill-conditioned problems “RANDL2” and “RANDL3”, the full GMRES is faster for more ill-conditioned problems. Hence, in the following, we used  $k = \infty$  (the full GMRES).

Table 4.8: Results for test matrices (1).

Matrix	(1)CG	(2)CG IMGS	(3)CG IMGS(0)	(4)GMRES 1 IMGS(0)	(5)GMRES 2 IMGS(0)
WELL1033	159 0.14	68 4.03	158 0.25	98 2.03	98 0.84
ILLC1033	3317 1.12	181 3.42	3342 2.09	256 17.72	257 9.05
ARTF1252	150 0.14	25 4.36	72 0.14	70 1.31	70 0.51
ARTF1346	202 0.22	33 6.21	104 0.22	92 2.37	94 0.86
ARTF1641	115 0.12	17 6.26	46 0.10	45 0.72	45 0.24
ARTF1991	34 0.04	11 8.54	14 0.06	14 0.13	14 0.06
ARTF2808	142 0.29	15 34.05	51 0.26	51 2.36	52 0.61

The computation results for the test matrices (1) of Table 4.1 from the Matrix Market are shown in Table 4.8. In each cell of the table, the upper number is the number of iterations, and the lower number is the computation time (in seconds) required for convergence. The CG method (1) and the CG method with IMGS(0) preconditioning (3) were the fastest since the problems are relatively well conditioned. (2) required the least number of iterations, but was computationally expensive, due to the time for preconditioning and backward and forward substitutions.



Table 4.9: Results for test matrices (2).

Matrix	(1)CG	(2)CG IMGS	(3)CG IMGS(0)	(4)GMRES 1 IMGS(0)	(5)GMRES 2 IMGS(0)
RANDS2	550 0.72	33 12.63	158 0.55	137 4.42	141 2.15
RANDS4	6411 6.99	203 15.02	1002 2.59	319 30.75	319 17.65
RANDS6	61281 64.11	686 21.38	4038 10.12	320 31.75	320 17.74
RANDS8	485766 490.65	1903 29.63	16546 43.17	320 31.90	334 20.45

Next, results for the test matrices (2) of Table 4.2 with varying condition numbers are given in Table 4.9. For the very ill-conditioned problem “RANDS8”, the CG method (1) required many iterations and computation time. On the other hand, the GMRES 2 method with IMGS(0) (5) converged quickly and was the fastest. Comparing (3) and (5), it is observed that when using the same preconditioning of IMGS(0), our approach of applying the GMRES method requires less iterations and computation time compared to the usual approach of applying CG to the normal equation. This may be explained by the fact that the coefficient matrix  $R^{-1}Q^T A$  for the GMRES method is better conditioned compared to the coefficient matrix  $R^{-1}A^T A R^{-1}$  for the CG method. Another explanation may be that GMRES performs the Gram-Schmidt orthogonalization explicitly, where as the CG relies on the three term recurrence, so that GMRES is more robust against loss of orthogonality due to rounding error, especially in ill-conditioned problems. (cf. convergence graphs in Fig. 4.3 and Fig. 4.4.) For less ill conditioned problems “RANDS4” and “RANDS6”, the same phenomenon is observed in the sense that GMRES requires less iterations than CG, but CG (3) is faster than GMRES 2 (5) in computation time because the (modified) Gram-Schmidt process of the GMRES is time consuming.

Table 4.10: Results for test matrices (3).

Matrix	(1)CG	(2)CG IMGS	(3)CG IMGS(0)	(4)GMRES 1 IMGS(0)	(5)GMRES 2 IMGS(0)
RANDM1	145	33	82	81	83
	1.33	651.18	3.88	9.20	5.33
RANDM2	865	80	343	292	301
	7.26	638.32	12.95	74.45	37.56
RANDM3	2904	230	807	540	556
	24.00	991.58	30.01	330.26	174.51
RANDM4	15319	513	2750	875	879
	125.39	1053.98	96.49	1260.92	769.30
RANDM5	45140	1668	8040	922	926
	369.21	1055.62	281.66	1362.47	927.19
RANDM6	129380	4179	23401	959	963
	1070.30	1422.33	818.38	1624.99	1090.49
RANDM7	606754	11169	80370	981	997
	5529.10	1827.32	2821.29	1677.47	1210.97
RANDM8	1000000*	30641	168264	983	1005
		3090.80	5869.60	1598.08	1294.39

Finally, experiment results for test matrices (3) of Table 4.3 and (4) of Table 4.4 including even larger matrices with varying condition numbers are given in Table 4.10 and Table 4.11, respectively. (The \* indicates that convergence was not obtained within the number of iterations.) Similar results are observed, where the GMRES 2 method with IMGS(0) preconditioning of (5) was the fastest for the severely ill-conditioned problems “RANDM7”, “RANDM8”, “RANDL6” and “RANDL7”.

Table 4.11: Results for test matrices (4).

Matrix	(1)CG	(2)CG IMGS	(3)CG IMGS(0)	(4)GMRES 1 IMGS(0)	(5)GMRES 2 IMGS(0)
RANDL1	146 1.48	21 401.04	61 3.99	61 14.12	63 5.06
RANDL2	994 9.18	51 535.74	270 12.83	254 200.01	262 31.80
RANDL3	7801 70.71	134 727.50	737 36.17	554 990.38	574 204.79
RANDL4	40851 368.96	611 1094.75	4558 218.15	991 4234.94	993 1211.83
RANDL5	157470 1444.30	1101 1180.55	9954 478.87	998 3858.91	998 1228.72
RANDL6	583250 5274.50	2733 1841.82	27129 1302.85	1000 4562.32	1000 1296.03
RANDL7	1000000* 2021.13	7018 2021.13	75995 3645.71	1000 4300.09	1060 1572.70

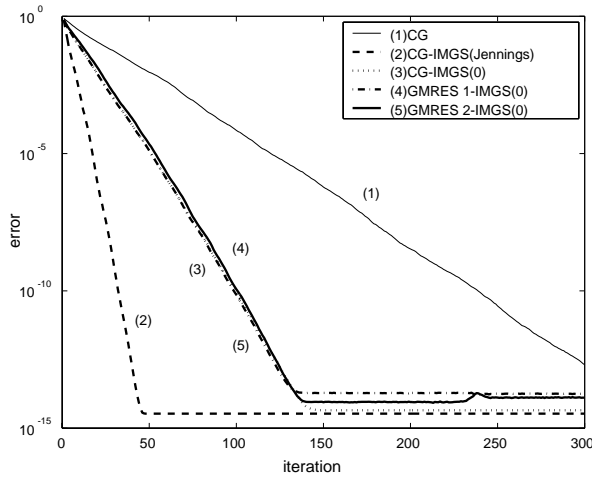


Figure 4.1: Relative error versus the number of iterations for the problem RANDL1.

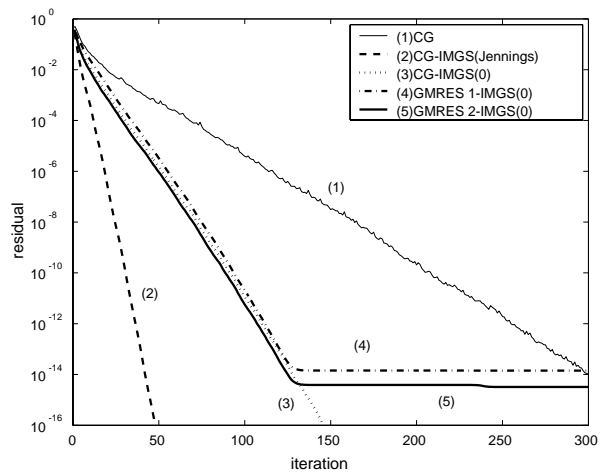


Figure 4.2: Relative residual versus the number of iterations for the problem RANDL1.

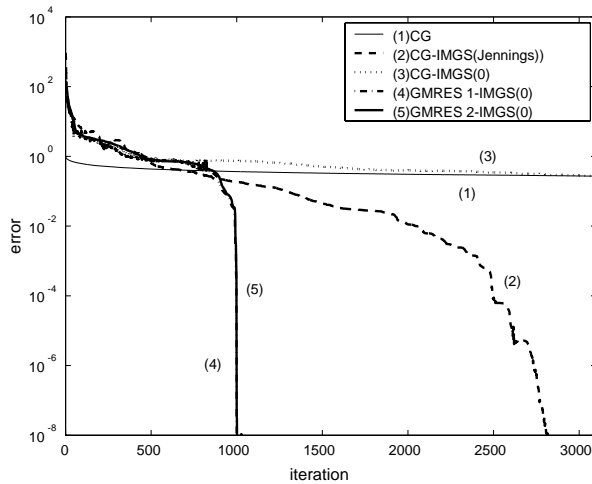


Figure 4.3: Relative error versus the number of iterations for the problem RANDL6.

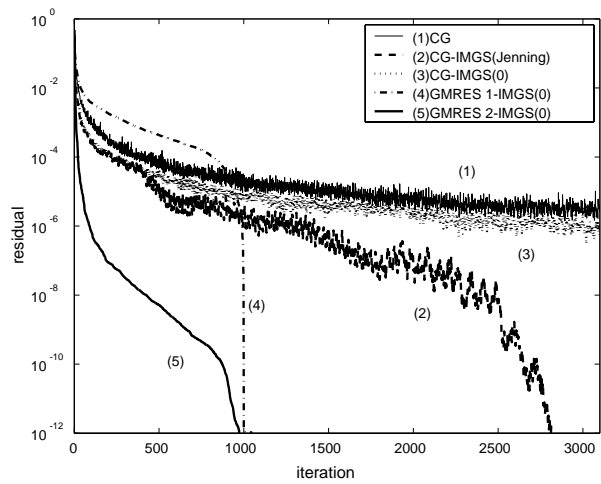


Figure 4.4: Relative residual versus the number of iterations for the problem RANDL6.

Fig. 4.1 shows the relative error vs. iterations, and Fig. 4.2 shows the relative residual vs. iterations for the case “RANDL1” in Table 4.4 and 4.11. Fig. 4.3 shows the relative error vs. iterations, and Fig. 4.4 shows the relative residual vs. iterations for the case “RANDL6” in Table 4.4 and 4.11. Again, it is shown that the GMRES converges better than CG with the same IMGS(0) preconditioning for the severely ill-conditioned problem “RANDL6”.

From these experiments, we conclude that the proposed GMRES 2 method with IMGS(0) preconditioning, which is equivalent to applying the GMRES method to the diagonally scaled normal equation, becomes superior to previous methods particularly for severely ill-conditioned problems.

## 5 Conclusions

In this paper, we proposed two methods for applying the GMRES method to the least squares problem by using a mapping matrix  $B$ . The first method applies GMRES to  $AB\mathbf{z} = \mathbf{b}$ , and the second method applies GMRES to  $BA\mathbf{x} = B\mathbf{b}$ . Next, we gave the necessary and sufficient condition which should be satisfied by  $B$ , in order that the methods give a least squares solution.

As an example for  $B$ , we proposed using  $\text{IMGS}(l)$ , which is a kind of incomplete QR decomposition. In fact,  $\text{IMGS}(0)$  and GMRES applied to  $BA\mathbf{x} = B\mathbf{b}$ , which is equivalent to applying the GMRES method to the diagonally scaled normal equation, turned up to be the fastest.

Numerical experiments showed that the proposed method is superior to previous methods, particularly when the coefficient matrix is severely ill-conditioned.

## Acknowledgements

We would like to thank Professor Kokichi Sugihara for an important remark which lead to Theorem 3.2, and Dr. Shao-Liang Zhang for discussions and useful information.

## References

- [1] M. BENZI AND M. TUMA, *A robust preconditioner with low memory requirements for large sparse least squares problems*, SIAM J. Sci Comput., 25 (2003), pp. 499–512.
- [2] A. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [3] P. N. BROWN AND H. F. WALKER, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.
- [4] D. CALVETTI, B. LEWIS, AND L. REICHEL, *GMRES-type methods for inconsistent systems*, Linear Algebra Appl., 316 (2000), pp. 157–169.
- [5] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 409–436.
- [6] A. JENNINGS AND M. A. AJIZ, *Incomplete methods for solving  $A^T Ax = b$* , SIAM J. Sci. Stat. Comput., 5 (1984), pp. 978–987.
- [7] NATIONAL INSTITUTE OF STANDARDS,  
*Matrix Market*, test matrices database available on line at <http://math.nist.gov/MatrixMarket/data/>, Gaithersburg, MD.
- [8] Y. SAAD, *Preconditioning techniques for nonsymmetric and indefinite linear systems*, J. Comput. Appl. Math., 24 (1988), pp. 89–105.
- [9] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
- [10] S.-L. ZHANG, *Generalization of the Conjugate Residual Method*, Ph.D. Thesis, University of Tsukuba, Japan, 1989 (in Japanese).
- [11] S.-L. ZHANG AND Y. OYANAGI, *A necessary and sufficient convergence condition of orthomin( $k$ ) methods for least squares problem with weight*, Ann. Inst. Statist. Math., 42 (1990), pp. 805–811.
- [12] S.-L. ZHANG AND Y. OYANAGI, *Orthomin( $k$ ) method for linear least squares problem*, J. Inf. Proc., 14 (1991), pp. 121–125.