Hindawi
Journal of Mathematics
Volume 2022, Article ID 5435076, 9 pages
https://doi.org/10.1155/2022/5435076

Research Article

Structure Preprocessing Method for the System of Unclosed Linear Algebraic Equations

Ling Li and Yongxian Li

School of Mathematics & Physics, Henan University of Urban Construction, Pingdingshan 467036, China

Correspondence should be addressed to Ling Li; liling@mju-edu.cn

Received 23 September 2022; Revised 30 November 2022; Accepted 14 December 2022; Published 27 December 2022

Academic Editor: Firdous A. Shah

The complexity of open linear algebraic equations makes it difficult to obtain analytical solutions, and preprocessing techniques can be applied to coefficient matrices, which has become an effective method to accelerate the convergence of iterative methods. Therefore, it is important to preprocess the structure of open linear algebraic equations to reduce their complexity. Open linear algebraic equations can be divided into symmetric linear equations and asymmetric linear equations. The former is based on 2×2. The latter is preprocessed by the improved QMRGCGS method, and the applications of the two methods are analyzed, respectively. The experimental results show that when the step is 500, the pretreatment time of quasi-minimal residual generalized conjugate gradient square 2 method is 34.23 s, that of conjugate gradient square 2 method is 35.14 s, and that of conjugate gradient square method is 45.20 s, providing a new reference method and idea for solving and preprocessing non-closed linear algebraic equations.

1. Introduction

With the development of computer science and technology, computational mathematics has become increasingly important. The numerical calculation method in computational mathematics is commonly adopted to deal with scientific and engineering calculation problems [1]. The effective solution methods of large-scale unclosed algebraic systems are considered one of the important research topics in computational mathematics because many scientific and engineering fields are inseparable from the numerical solution of differential equations or integral equations, such as structural mechanics, computational fluid mechanics, electromagnetic field calculation, material simulation and design, life science, aerodynamics, system science, medical science, astronomy, financial engineering, social science, and other soft sciences. For linear partial differential equation or integral equation, it is difficult to find their analytical solutions because of their high complexity. The numerical method is adopted as a tool to solve this problem. The common numerical methods in computational mathematics, such as finite element, finite difference, finite volume, moment method, and meshless discretization methods, have been extensively studied. These computational problems are ultimately transformed into solving one or a group of large-scale open linear algebraic equations [2].

In practical application and engineering calculation, the iterative method is commonly used to solve linear equations. However, with the rapid development of science and technology and the increasing scale of the required solutions, the low-efficiency solution of the steady iteration method can no longer meet the needs of scientific calculation, and in fact it has rarely been used alone in the solution of equations. At the same time, tropical algebra can be used to solve combinatorial optimization problems. It combines the concepts and methods of statistical physics, machine learning, and other fields and has been well applied in noise removal and optimal control. In addition, the homotopy perturbation method transforms the problem of solving some non-linear partial differential equations into the initial value problem of solving ordinary differential equations through traveling wave transformation and homotopy perturbation theory, which is a relatively common method for solving non-linear problems. Later, Young proposed the definition and basic concept of non-stationary iterative method. The non-stationary Richardson iterative method is the first non-stationary iterative method, which can be directly extended to the steepest descent method, Chebyshev semi-iterative method, preconditioned conjugate gradient (PCG) method, and generalized CG (GCG) method. Based on the stationary iterative method of Krylov subspace represented by the CG method, the convergence rate of the iterative method is still related to the spectral distribution of the coefficient matrix. When the distribution of the eigenvalues of the iterative matrix is more concentrated, the convergence speed of the iterative method is faster. When the spectral distribution is more dispersed, the convergence speed of the non-stationary iterative method tends to be slow in general, and sometimes the iterative method does not converge. In this case, the preprocessing technology can be applied to the coefficient matrix to make the spectral radius of the iterative matrix tend to gather, which is an effective way to solve the dispersion of spectral distribution and accelerate the convergence speed of the iterative method [3]. In order to further reduce the complexity of open algebraic equations and effectively optimize the convergence performance of this method, this paper studies the structure preprocessing method of the open linear algebraic equations. According to the type of equations, the treatment can be divided into two parts, that is, the preprocessing of symmetric linear equations and the preprocessing of asymmetric linear equations.

2. The Structure Preprocessing Method for the System of Unclosed Linear Algebraic Equations

In the fields of natural science, aerodynamics, economic management, engineering technology, fluid mechanics, structural mechanics, and aerospace engineering, many practical problems can be solved by solving linear equations.

$$Ax = b, \tag{1}$$

where matrix A is an $n \times n$ matrix and x and b are n-dimensional column vectors.

The methods for solving linear equations mainly include square root method, elimination method, direct triangular decomposition method, determinant and matrix inversion, Seidel iteration method, Jacobi iteration method, super relaxation iteration method, and other iteration methods [4].

However, the above methods are not suitable for all linear equations. Strictly speaking, they have great limitations in solving some specific problems. Therefore, some scholars have made efforts on developing preprocessing methods for linear equations [5]. These preprocessing methods are generally to split coefficient matrix 8 of linear equations in different ways, so that the iterative method converges to obtain the solution of linear equations. In [6], the general coefficient matrix of a linear equation system is transformed into a symmetric positive definite matrix, so that the question of solving the original linear equation system can be transformed into the question of finding the minimum value of an equivalent variational question. In addition, in [7], various algorithms for solving a linear equation system are given. Among them, the authors in [8] developed an improved Gauss–Seidel iterative method to solve the non-convergent linear equations and selected appropriate processing factors to realize the iterative convergence of the linear equations.

Scholars have made great efforts on solving large sparse linear equations. GMRES algorithm, preconditioned conjugate gradient method, ICCG method, and other common methods are used to solve large sparse linear equations. Among them, GMRES algorithm is one of the most effective algorithms for solving large sparse asymmetric linear equations at present, and the Krylov subspace method is commonly used to solve the preprocessing problem of large sparse linear equations. Moreover, various iterative methods have been developed on this basis, including the conjugate gradient method, the generalized minimal residual method, and so on. Conjugate gradient is a method between the steepest descent method and Newton method. It only uses the information of the first derivative, but it overcomes the shortcomings of the slowest descent method and Newton method, which need to store, calculate, and inverse Hesse matrix. The generalized minimum residual method is also a kind of unsteady method, which has the advantages of saving storage space and having less computation, and is suitable for parallel computing. However, the convergence of the two algorithms is easily affected by applications and boundary conditions [9–13].

The preprocessing method of linear equations generally involves the calculation of the iterative method. In such a rapidly developing information age, there are more requirements for solving linear equations, and the requirements for solving speed are becoming higher than ever. Therefore, how to make the solution closer to the practical application, faster, and more accurate is a major problem to be considered.

For the preprocessing methods for solving linear equations, the search and selection of preprocessor is the key link. The preprocessor G in this paper selects the approximation to the inverse of coefficient matrix A of linear equations. The method to obtain the inverse matrix of coefficient matrix is elaborated in detail in algebra book of undergraduate course; in fact, the elementary transformation method is the most common one. It is one of the teaching contents of undergraduate course to find the inverse matrix by elementary transformation, but the scale of matrix that can be transformed by elementary transformation is generally small, and it can even be deduced and calculated in the exercise book. However, in the fields of natural science, economic management, engineering technology, fluid mechanics, aerodynamics, structural mechanics, aerospace engineering, etc., the matrices used in practice are usually large-scale matrices or matrices with relatively large condition numbers, which can hardly be transformed [14]. According to the previous considerations, the mathematical knowledge is combined with the current computer knowledge, and the algorithm is constructed to realize the operation of matrix inversion in the computer software, so as to reduce the difficulty of matrix inversion encountered in the actual calculation process [15].

In this section, the preprocessing methods of symmetric linear equations are discussed. To be specific, the coefficient matrix is divided into component block forms, and then the commonly used preprocessing methods are applied to the saddle point question after partition, and three kinds of preprocessors are obtained.

2.1. Principle and Method. At present, the preprocessors for saddle point problems can be divided into block diagonal preprocessor, block triangle preprocessor, constraint preprocessor, and so on. For the two types of linear equations obtained previously, they can be uniformly written in the following form by repartitioning:

$$\underbrace{\begin{bmatrix} F & B^T \\ C & 1 \end{bmatrix}}_{A}\begin{bmatrix} x \\ p \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}. \tag{2}$$

In general, the above linear system can be considered as saddle point question, where A is saddle point matrix [16]. B^T is the scrambling parameter, (F, C) is a point in the set, x is the penalty parameter of the matrix, p is the constraint coefficient, c is the penalty parameter after the union, and d is the penalty parameter after the union.

For a matrix with the above structure, the corresponding block diagonal preprocessor and block lower triangle preprocessor are constructed:

$$\begin{aligned} P_D &= \begin{bmatrix} F & 0 \\ 0 & CF^{-1}B^T \end{bmatrix}, \\ P_t &= \begin{bmatrix} F & 0 \\ C & -CF^{-1}B^T \end{bmatrix}, \end{aligned} \tag{3}$$

where P_D is the block diagonal preprocessor and P_t is the lower triangular preprocessor of the block.

If the coefficient matrix A is symmetric, a constraint-type preprocessor with the following forms can also be constructed:

$$P_C = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}, \tag{4}$$

where P_C is the constraint-type preprocessor and G is a symmetric matrix, which is the approximate value of F.

For the block diagonal preprocessor, the following conclusion can be obtained.

Lemma 1. *If P_D is used to preprocess A, then the coefficient matrix $T = P_D^{-1}$ after preprocessing satisfies*

$$T(T-1)\left(T^2 - T - 1\right) = 0. \tag{5}$$

This theorem shows that T is diagonalizable and has at most four different eigenvalues: 0, 1, and $1/2 \pm \sqrt{5}/2$ [17].

When T is non-singular, there are only three non-zero eigenvalues. This shows that when T is non-singular, the Krylov subspace is

$$K(T, r) = \operatorname{span}\left(r, Tr, T^2 r, \ldots\right). \tag{6}$$

The number of dimensions is no more than 3, where r is the initial residual. Therefore, the Krylov subspace method is terminated in at most 3 steps [18].

Similarly, for the subblock triangle preprocessor, we have the following conclusions.

Lemma 2. *Set*

$$\begin{aligned} A &= \begin{bmatrix} F & B^T \\ C & 0 \end{bmatrix}, \\ P_t &= \begin{bmatrix} F & 0 \\ C & -CF^{-1}B^T \end{bmatrix}. \end{aligned} \tag{7}$$

If $F \in R^{m \cdot m}$ is non-singular and $B \in R^{m \cdot m}$ is full rank, then the $m + n$ eigenvalues of matrix $P_t^{-1}A$ after preprocessing are equal to 1.

For the constrained preprocessor, the following conclusion can be obtained.

Lemma 3. *Let A be a symmetric indefinite matrix in the following form:*

$$\begin{aligned} A &= \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix}, \\ P_C &= \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}, \end{aligned} \tag{8}$$

where $B \in R^{n \cdot m}$ is full rank, $G \in R^{m \cdot m}$ is symmetric, $G \neq F$, and P_C represents non-singular. Let $Y \in R^{m \cdot (m-n)}$ be a group of bases of the zero space of B, i.e., span$\{Y\}$ = null(B); then, $P_C^{-1}A$ has the following properties:

(1) $P_C^{-1}A$ has 2n eigenvalues of 1.

(2) The remaining $m - n$ eigenvalues satisfy the following generalized eigenvalue question:

$$Y^T F = aY^T G, \tag{9}$$

where Y^T is the residual matrix and a is the coefficient.

2.2. Application Research. The following three preprocessing methods are applied to symmetric algebraic linear equations. Let equations be written as follows:

$$\underbrace{\begin{bmatrix} \delta Q & 0 & -Q_{sd}^T \\ 0 & Q & H_{sd}^T \\ -Q_{sd}^T & H_{sd}^T & 0 \end{bmatrix}}_{A}\begin{bmatrix} f \\ u \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ d \end{bmatrix}. \tag{10}$$

Let

$$F=\begin{bmatrix}\delta Q & 0\\ 0 & Q\end{bmatrix},\qquad B=\begin{bmatrix}-Q_{sd}^{T} & H_{sd}^{T}\end{bmatrix}, \tag{11}$$

where δ represents the coefficient, Q represents the mass matrix of symmetric positive definite, Q_{sd}^{T} and H_{sd}^{T} are the mass matrices of symmetric indefinite, and $\begin{bmatrix}f\\ u\\ p\end{bmatrix}$ is the stagnation point of the Lagrange function [19].

Then, A can be written as

$$A=\begin{bmatrix}F & B^{T}\\ B & 0\end{bmatrix}. \tag{12}$$

For linear equation (10), block diagonal and block bottom triangle preprocessors are expressed as follows:

$$P_D=\begin{bmatrix}\delta Q & 0 & 0\\ 0 & Q & 0\\ 0 & 0 & \dfrac{Q_{SD}Q^{-1}Q_{sd}^{T}}{\delta}+H_{sd}Q^{-1}H_{sd}^{T}\end{bmatrix},$$

$$P_t=\begin{bmatrix}\delta Q & 0 & 0\\ 0 & Q & 0\\ -Q_{sd}H_{sd} & -\dfrac{Q_{sd}Q^{-1}Q_{sd}^{T}}{\delta} & -H_{sd}Q^{-1}H_{sd}^{T}\end{bmatrix}. \tag{13}$$

For constrained preprocessor, select

$$Y=\begin{bmatrix}-Q_{sd}H_{sd}\\ L\end{bmatrix}, \tag{14}$$

where L is the constraint matrix. Then, Y column becomes zero space of B, and

$$\begin{aligned}Y^{T}FZ&=\begin{bmatrix}-Q_{sd}^{T}H_{sd}^{T}L\end{bmatrix}\begin{bmatrix}\delta Q & 0\\ 0 & Q\end{bmatrix}\begin{bmatrix}-Q_{sd}^{-1}H_{sd}\\ L\end{bmatrix}\\ &=Q+\delta\left(-Q_{sd}^{-1}H_{sd}\right)^{T}Q\left(-Q_{sd}^{-1}H_{sd}\right),\end{aligned} \tag{15}$$

where Z is a zero-space vector. So, the concrete form of P_C is

$$P_C=\begin{bmatrix}0 & 0 & -Q_{sd}^{T}\\ 0 & Q+\delta\left(-Q_{sd}^{-1}H_{sd}\right)^{T}Q\left(-Q_{sd}^{-1}H_{sd}\right) & H_{sd}^{T}\\ -Q_{sd} & H_{sd} & 0\end{bmatrix}. \tag{16}$$

3. Preprocessing of Unsymmetrical Algebraic Linear Equations

In this section, the preprocessing of the system of positive definite linear equations is studied. The BCG method and its CGS method of unsymmetrical linear equations have typical irregular convergence behavior. Freund and Nachtigal put forward quasi-minimal residual method (QMR) to remedy the convergence of the BCG method and produce smooth convergence curve. However, like the BCG method, the QMR method uses coefficient matrix A and the product of transposition AT and vector. In order to solve this problem, Freund proposed the TFQMR method, which has the property of quasi-minimum residuals and does not use the product of AT and vectors. It is a Krylov subspace algorithm and performs well in the experiment of solving large sparse linear equations [20–26].

In order to improve the convergence of the CGS method, Fokkema et al. extended the CGS method and derived the generalized CGS method, which is called the GCGS method. The GCGS method effectively changes the convergence of the CGS method and BiCGSTAB method. At the same time, two new methods, CGS2 and SCGS, have been derived, but they do not have quasi-minimum surplus, which leads to irregular convergence behavior for some complex problems [27, 28]. In this section, quasi-minimal residuals are introduced into the GCGS method, yielding the QMRGCGS method. This kind of method includes the TFQMR method and QMRCGSTAB method as special cases. At the same time, two new methods, namely, QMRGCGS2 method and QMRSCGS method, are derived based on the QMRGCGS method.

3.1. GCGS Method. The orientation quantity $x_0 \in R^N$ is taken as the initial solution of equation $Ax = b$, and its residual vector is $r_0 = b - Ax_0$. The residual vector of the n-th iteration of BCG method is $r_0^{BCG} = b - Ax_n$. It can be expressed in the form of $r_n^{BCG} = \phi_n(A)x_0$, where $\phi_n \in P_n$ and $\phi_n(0) = 1$. The residual vector of the n-th iteration of the CGS method is $r_n^{CGS} = (\phi_n(A)^2 r_0$. The residual vector of the n-th iteration of the GCGS method is different from that of the CGS method, and its expression is $r_n^{GCGS} = \phi_n(A)\psi_n(A)r_0$. Among them, $\phi_n(A)$ is still BCG polynomial, and $\phi_n(A)$ is the approximate form of BCG polynomial, so the GCGS method can be derived, and GCGS is given by Algorithm 1 [29].

In Algorithm 1, taking $\tilde{\alpha}_k = \alpha_k$ and then $\tilde{\beta} = \beta_{\underline{k}}$, the CGS method is derived. Taking $\tilde{\alpha}_k = (e_k, t_k)/(t_k, t_k)$, $\beta_k = 0$ and the BiCGSTAB method is derived. Related BCG polynomial is used, and $\tilde{\alpha}_k = \tilde{\rho}_k/\tilde{\sigma}_k$ and $\tilde{\beta}_k = \tilde{\rho}_k/(\alpha_{k-1}\tilde{\sigma}_{k-1})$ are taken correspondingly, where $\tilde{\rho}_k = (r_k, \tilde{e}_0)$, $\tilde{\sigma}_k = (Av_k, \tilde{e}_0)$, and then the GCGS2 method is derived. When $k = 0$, $\tilde{\alpha}_k = b$ (b is the reciprocal of the maximum eigenvalue of coefficient

```
Select an initial guess $x_0$ and some $\widetilde{r}_0$.
$r_0 = b - Ax_0$
$u_{-1} = w_{-1} = s_{-1} = 0$
$\alpha_{-1} = \sigma_{-1} = \widetilde{\alpha}_{-1} = \widetilde{\sigma}_{-1} = 1$
for $k = 0, 1, 2, \ldots,$ do
$\rho_k = (r_k, \widetilde{r}_0)$
$\beta_k = (-1/\widetilde{r}_0)(\rho_k/\sigma_{k-1})$
$v_k = r_k - \beta_k u_{k-1}$
Select $\widetilde{\beta}_k$
$t_k = r_k - \widetilde{\beta}_k e_{k-1}$
$w_k = t_k - \beta_k (u_{k-1} - \widetilde{\beta}_k w_{k-1})$
$c = Aw_k$
$\sigma_k = (c, \widetilde{r}_0)$
$\alpha_k = \rho_k/\sigma_k$
$e_k = t_k - \alpha_k c$
Choose $\widetilde{\alpha}_k$
$u_k = u_k - \widetilde{\alpha}_k c$
$u_{k+1} = x_k + \alpha_k v_k + \widetilde{\alpha}_k e_k$
If $x_{k+1}$ is accurate enough, then quit
$r_{k+1} = r_k - A(\alpha_k v_k + \widetilde{\alpha}_k e_k)$
End
```

Algorithm 1: GCGS method.

matrix) and $\widetilde{\beta}_k = 0$; when $k > 0$, $\widetilde{\alpha}_k = \alpha_{k-1}$ and $\widetilde{\beta}_k = \beta_{k-1}$, so that the SCGS method is exported [30–32].

3.2. QMRGCGS Method.

From the above analysis, it can be seen that the CGS method uses the square of BCG polynomial, and the GCGS method has been extended on this basis. Using the product of BCG polynomials and approximate BCG polynomials, the GCGS method does not have quasi-minimal residuals [33, 34]. Therefore, quasi-minimal residuals are introduced into the GCGS method, and the QMRGCGS method is derived [35, 36].

Set

$$s_m = \begin{cases} e_{k-1}, \text{ if } m = 2k - 1 \text{ is odd,} \\ v_k, \text{ if } m = 2k \text{ is even,} \end{cases} \tag{17}$$

where s_m represents a QMRGCGS function, e_{k-1} represents a function, and v_k represents a function, and

$$p_m = \begin{cases} \alpha_{k-1}, & \text{if } m = 2k - 1 \text{ is odd,} \\ \alpha_k, & \text{if } m = 2k \text{ is even,} \end{cases} \tag{18}$$

where P_m represents the logarithmic equation of the QMRGCGS function, a_{k-1} represents a function, and a_k represents a function.

Let

$$\begin{aligned} S_m &= \{s_0, s_1, \ldots, s_{m-1}\}, \\ P_m &= \{p_0, p_1, \ldots, p_{m-1}\}^T. \end{aligned} \tag{19}$$

According to Algorithm 1,

$$\begin{aligned} r_m &= r_0 - AS_m P_m \\ &= r_{m-1} - p_{m-1} As_{m-1}, \end{aligned} \tag{20}$$

where r_m represents the weight matrix. So,

$$As_m = \frac{(r_{m-1} - r_{m+1})}{p_m}. \tag{21}$$

Let

$$R_m = (r_0, r_1, \ldots, r_{m-1}), \tag{22}$$

where R_m represents the weight set. Then, it can be written as

$$AS_m = R_{m+1} \overline{C}_m, \tag{23}$$

where $\overline{C}_m$ represents the function matrix, wherein

$$\overline{C}_m = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & O \quad O & \\ & -1 & 1 \\ & & -1 \end{bmatrix} \cdot \left\{ \frac{1}{p_0}, \frac{1}{p_1}, \ldots, \frac{1}{p_{m-1}} \right\}. \tag{24}$$

For the GCGS method, the following relation holds.

$$v_{k-1} = \phi_{k-1}(A) \psi_{k-1}(A) r_0. \tag{25}$$

Since matrix ϕ_k and matrix ψ_k are of order k, according to the definition of s_m, there is the following equation:

$$M_m(r_0, A) = \operatorname{span}\{s_1, s_2, \ldots, s_m\} = \{S_m PP \in R_m\}, \tag{26}$$

where M_m represents a vector and P represents a point on the weight set. Quasi-minimal residuals are introduced. From equation (26), the iteration result $x_m \in x_0 + M_m(r_0, A)$ of Step m can be written in the following form, and there is a certain $p \in R_m$, so that

$$x_m = x_0 + S_mP. \tag{27}$$

Thus,

$$r_m = r_0 - AS_mP = R_{m+1}\left[h_1^{m+1} - \overline{C}_mP\right], \tag{28}$$

where $h_1^{m+1} = (1,0,...,0)^T \in R_{m+1}$. In fact, p of equation (27) can be selected to make $\|r_m\|$ minimal. However, since R_{m+1} is dense and its column vectors are not orthogonal to each other, the amount of calculation is too large, and quasi-minimal residuals are introduced.

$$Z_{m+1} = \mathrm{diag}\{v_1, v_2, \dots, v_{m-1}\}, \quad v_k \geq 0,\ k = 1, 2, \dots, m+1. \tag{29}$$

As a weight matrix, take $v_k = 1/\|r_k\|$, $k = 1, 2, ..., m+1$, and equation (28) can be written as

$$r_m = R_{m+1}Z_m^{-1}\left(v_1h_1^{m+1} - Z_{m+1}\overline{C}_mP\right). \tag{30}$$

Let $P_m = R_m$ be the solution of minimal quadratic question, that is, P_m satisfies

$$r_m = \left\|v_1h_1^{m+1} - Z_{m+1}\overline{C}_mP_m\right\| = \min_{p\in R_m}\left\|v_1h_1^{m+1} - Z_{m+1}\overline{C}_mP\right\|. \tag{31}$$

So, the solution x_m of the QMRGCGS method can be expressed as

$$x_m = x_0 + S_mP_m, \tag{32}$$

where P_m satisfies equation (31).

The Givens transform is applied to $Z_{m+1}\overline{C}_m$, and a series of Givens transform is set to $G_0, G_1, \dots, G_m$. $Z_{m+1}\overline{C}_m$ is transformed into a matrix $\overline{\xi}_m$ in the form of upper triangle, namely,

$$\begin{aligned}\overline{\xi}_m &= G_m \dots G_0G_1Z_{m+1}\overline{C}_m,\\ \overline{\zeta}_m &= G_n \dots G_0G_1v_1h_1^{m+1}.\end{aligned} \tag{33}$$

Let ξ_m be the matrix obtained by removing the last row of ξ_m and ζ_m be the vector obtained by removing the last element of $\overline{\zeta}_m$; then, P_m satisfying equation (31) can be expressed as

$$P_m = \xi_m^{-1}\zeta_m. \tag{34}$$

There are

$$x_m = x_0 + S_m\xi_m^{-1}\zeta_m. \tag{35}$$

Let

$$D_m = S_m\xi_m^{-1}, \tag{36}$$

where D_m is a minimum function. Then,

$$\begin{aligned}x_m &= x_0 + D_m\zeta_m,\\ \zeta_m &= \begin{cases}\zeta_{m-1},\\ I_m,\end{cases}\end{aligned} \tag{37}$$

where $I_m = J_mI_m^{(m+1)}$, $I_m^{(m+1)}$ is the last element of vector $\overline{\zeta}_{m-1}$, and J_m is the cosine of G_m; then,

$$x_m = x_0 + D_{m-1}\zeta_{m-1} + I_mp_m. \tag{38}$$

Thus,

$$x_m = x_{m-1} + I_mp_m. \tag{39}$$

In this way, the result of preprocessing of the unsymmetrical algebraic linear equations can be obtained.

3.2.1. Application Test of QMRCGS2 Method. The QMRGCGS2 method is used to preprocess the following asymmetric algebraic linear equations.

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + \gamma\left(x\frac{\partial u}{\partial x} + y\frac{\partial u}{\partial y}\right)nu = f. \tag{40}$$

The equation has Dirichlet boundary condition in $(0,1) \times (0,1)$. The size of the grid is $h = 1/64$, $n = 200$, and $\gamma = 100$. On this basis, the Reynolds number on each element is less than 1. A stable discrete scheme is generated by using the central difference scheme [37, 38]. Figure 1 shows the relative residual vectors and iterative steps obtained by using CGS, CGS2, and QMRGCGS2 methods for 200 steps. The relative residual norm is $\|r_n\|/\|r_0\|$, and the initial value is $x_0 = 0$. The workload and storage of every two iterations of the QMRCGS2 method are equal to those of the CGS method and CGS2 method, so that k and n satisfy the following relationship: $k = 2n - 1$ or $k = 2n$ [39, 40]. The iterative steps are shown in Figure 1.

From Figure 1, it can be seen that the relative residual norm of CGS2,SCGS, GCGS CGS, and GCGS2 methods shows irregular convergence behavior, while the relative residual norm of the QMRGCGS2 method tends to show regular convergence behavior. That is to say, the QMRGCGS2 method has smoother convergence behavior than the CGS method and GCGS2 method [41–43]. Finally, CGS2 and CGS methods with good performance are selected to compare their performance with the proposed QMRGCGS2 method for practical application problems. The actual problem selected is Grenoble, which comes from the Harwell-Boeing sparse matrix set with extensive application background. The order is 115, and the number of non-zero elements is 421. The relative residual norm and iteration steps obtained by 150 steps of the three methods are shown in Figure 2.

It can be seen from Figure 2 that the relative residual norm of CGS2 and CGS methods in the practical application of Grenoble shows an irregular convergence behavior, while the relative residual norm of the proposed QMRGCGS2 method tends to a constant, that is, the method has a smoother convergence. Then, calculate the pretreatment time to verify the effectiveness of the pretreatment. The convergence criterion requires that the relative residual norm is less than 10–8. Table 1 shows the iteration and calculation time required for convergence. The problem in Table 1 is the two-dimensional three-temperature energy discretization linear algebraic equation. The scale size of the grid node is 8000 (160 × 53).

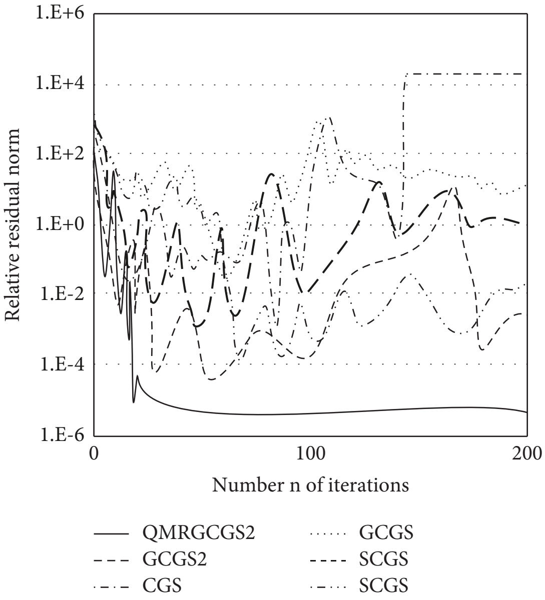

Figure 1: The relative residual norm of the equation.

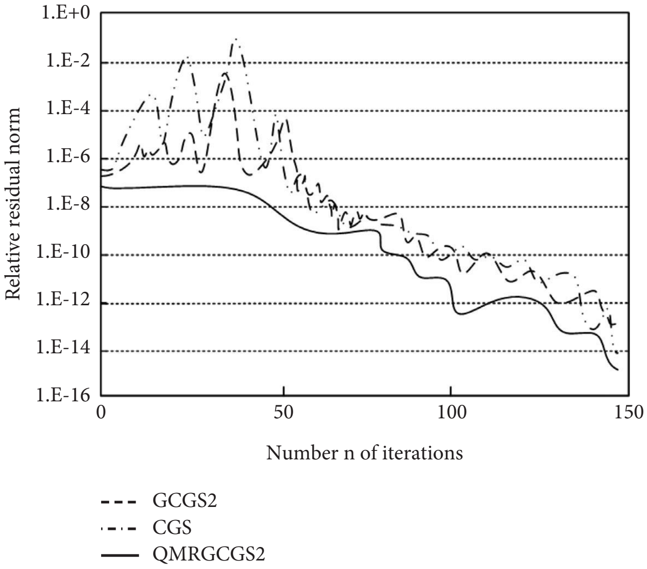

Figure 2: Relative residual norm and iteration step results obtained by 150 steps of three methods.

Table 1: Preprocessing time.

Steps	QMRGCGS2 (s)	CGS2 (s)	CGS (s)
100	15.36	20.36	22.36
200	20.30	23.34	27.14
300	25.47	27.56	32.25
400	28.36	30.21	38.17
500	34.23	35.14	45.20
600	38.11	40.12	50.31
700	42.36	45.21	58.74
800	45.10	48.25	66.62
900	48.51	55.36	70.36
1000	52.36	60.47	76.00

According to Table 1, when the step is 100, the pretreatment time of the QMRGCGS2 method is 15.36 s, that of the CGS2 method is 20.36 s, and that of the CGS method is 22.36 s. When the step is 500, the pretreatment time of the QMRGCGS2 method is 34.23 s, that of the CGS2 method is 35.14 s, and that of the CGS method is 45.20 s. It can be seen from the test results that the QMRGCGS2 method is faster than CGS, and the CGS2 method has a faster convergence speed, indicating higher efficiency of the preprocessing method proposed in this paper.

4. Conclusions

In this paper, the preconditioning of symmetric linear equations and asymmetric linear equations is studied, respectively. The experiment shows that when the step size is 500, the pretreatment time of the QMRGCGS2 method is 34.23 s, that of the CGS2 method is 35.14 s, and that of the CGS method is 45.20 s. The QMRGCGS2 method has more smooth convergence behavior and faster convergence speed than the CGS method and GCGS2 method. It provides a new method for solving asymmetric linear equations and has certain promotion value.

However, in the research of asymmetric linear equations, only m-step polynomial preprocessing numerical experiment has been done. In future research work, the effectiveness of the preprocessor when it is applied to other Krylov subspace methods will be studied. For the polynomial preprocessor in this paper, how to choose an optimal value is a question worthy of further study.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was funded by the Natural Science Foundation of China (Study on non-smooth decentralized control for a class of uncertain non-linear large-scale systems) (no. 61503122) and Tackling of Key Scientific and Technical Project of Henan Province (Non-smooth intelligent control of non-linear system and its application) (no. 202102210142).

References

[1] O. O. Kuznetsov, Y. I. Gorbenko, I. M. Bilozertsev, A. V. Andrushkevych, and O. P. Narizhnyi, "Algebraic immunity of non-linear blocks of symmetric ciphers," *Telecommunications and Radio Engineering*, vol. 77, no. 4, pp. 309–325, 2018.

[2] M. Grigoryan and V. Yedoyan, "On the interaction of cracks and stringers with a heterogeneous wedge-shaped body, manufactured from different materials at an antiplane deformation," *Key Engineering Materials*, vol. 828, no. 1, pp. 31–39, 2019.

[3] X. F. Zong, T. Li, G. Yin, L. Y. Wang, and J. F. Zhang, "Stochastic consent ability of linear systems with time delays and multiplicative noises," *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1059–1074, 2018.

[4] D. D. Thuan, C. Nguyen, N. T. Ha, N. H. Du, and N. H. Du, "Robust stability of linear time-varying implicit dynamic equations: a general consideration," *Mathematics of Control, Signals, and Systems*, vol. 31, no. 3, pp. 385–413, 2019.

[5] J. C. Arceo, M. Sanchez, V. Estrada-Manzo, and M. Bernal, "Convex stability analysis of nonlinear singular systems via linear matrix inequalities," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1740–1745, 2019.

[6] F. Mirzaee, S. Alipour, and N. Samadyar, "A numerical approach for solving weakly singular partial integro differential equations via two dimensional orthonormal Bernstein polynomials with the convergence analysis," *Numerical Methods for Partial Differential Equations*, vol. 35, no. 2, pp. 615–637, 2019.

[7] P. Manfredi, D. De Zutter, and D. Vande Ginste, "On the relationship between the stochastic Galerkin method and the pseudo-spectral collocation method for linear differential algebraic equations," *Journal of Engineering Mathematics*, vol. 108, no. 1, pp. 73–90, 2018.

[8] S. A. Gutnik and V. A. Sarychev, "Application of computer algebra methods to investigate the dynamics of the system of two connected bodies moving along a circular orbit," *Programming and Computer Software*, vol. 45, no. 2, pp. 51–57, 2019.

[9] G. Huang, J. Li, C. Luo, X. Chen, and X. H. Chen, "Regularisation parameter adaptive selection and its application in the prestack AVO inversion," *Exploration Geophysics*, vol. 49, no. 3, pp. 323–335, 2018.

[10] F. Meng, A. Pang, X. Dong, C. Han, and X. Sha, "H_{∞} optimal performance design of an unstable plant under bode integral constraint," *Complexity*, vol. 2018, Article ID 4942906, 10 pages, 2018.

[11] F. Meng, D. Wang, P. Yang, and G. Xie, "Application of sum of squares method in nonlinear H_{∞} control for satellite attitude maneuvers," *Complexity*, vol. 2019, Article ID 5124108, 10 pages, 2019.

[12] C. Zhao, X. Liu, S. Zhong, K. Shi, D. Liao, and Q. Zhong, "Secure consensus of multi-agent systems with redundant signal and communication interference via distributed dynamic event-triggered control," *ISA Transactions*, vol. 112, pp. 89–98, 2021.

[13] S. Lv and F. Song, "Particle swarm intelligence and the evolution of cooperation in the spatial public goods game with punishment," *Applied Mathematics and Computation*, vol. 412, Article ID 126586, 2022.

[14] M. A. Medvedeva, T. E. Simos, C. Tsitouras, and C. Tsitouras, "Exponential integrators for linear inhomogeneous problems," *Mathematical Methods in the Applied Sciences*, vol. 44, no. 1, pp. 937–944, 2021.

[15] M. Chen, W. Deng, and S. Serra-Capizzano, "Uniform convergence of v-cycle multigrid algorithms for two-dimensional fractional feynman-kac equation," *Journal of Scientific Computing*, vol. 74, no. 2, pp. 1034–1059, 2018.

[16] G. Shen, F. Liu, Z. Fu, B. Yu, and B. Yu, "Visual cryptograms of random grids via linear algebra," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12871–12899, 2018.

[17] C. W. Lee, Y. Hong, M. Hayrapetyan, X. G. Yang, and Z. Xi, "Derivation and tuning of a solvable and compact differential

- algebraic equations model for LiFePO 4 - graphite Li - ion batteries," *Journal of Applied Electrochemistry*, vol. 48, no. 3, pp. 365–377, 2018.

[18] A. Sahoo and V. Narayanan, "A min - max approach to event - and self - triggered sampling and regulation of linear systems," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5433–5440, 2019.

[19] A. N. Kovalenko, "Projection method for constructing full - wave models of striplines," *Journal of Communications Technology and Electronics*, vol. 64, no. 2, pp. 93–99, 2019.

[20] G. Casale and J. A. Weil, "Galoisian methods for testing irreducibility of order two nonlinear differential equations," *Pacific Journal of Mathematics*, vol. 297, no. 2, pp. 299–337, 2018.

[21] K. Wang and S. Li, "Robust distributed modal regression for massive data," *Computational Statistics & Data Analysis*, vol. 160, Article ID 107225, 2021.

[22] S. Li, "Efficient algorithms for scheduling equal-length jobs with processing set restrictions on uniform parallel batch machines," *Mathematical Biosciences and Engineering*, vol. 19, no. 11, pp. 10731–10740, 2022.

[23] K. Wang, H. Wang, and S. Li, "Renewable quantile regression for streaming datasets," *Knowledge-Based Systems*, vol. 235, Article ID 107675, 2022.

[24] S. He, F. Guo, Q. Zou, and D. Hui, "MRMD2.0: a Python tool for machine learning with feature ranking and reduction," *Current Bioinformatics*, vol. 15, no. 10, pp. 1213–1221, 2020.

[25] Z. Wu, A. Song, J. Cao, J. Luo, and L. Zhang, "Efficiently translating complex SQL query to MapReduce jobflow on cloud," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 508–517, 2020.

[26] X. Wu, W. Zheng, X. Xia, and D. Lo, "Data quality matters: a case study on data label correctness for security bug report prediction," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2541–2556, 2022.

[27] B. Zogheib and E. Tohidi, "Modal Hermite spectral collocation method for solving multi-dimensional hyperbolic telegraph equations," *Computers & Mathematics with Applications*, vol. 75, no. 10, pp. 3571–3588, 2018.

[28] W. Dang, J. Guo, M. Liu et al., "A semi-supervised extreme learning machine algorithm based on the new weighted kernel for machine smell," *Applied Sciences*, vol. 12, no. 18, p. 9213, 2022.

[29] V. L. Girko and L. D. Shevchuk, "The inverse tangent law for the solutions of systems of linear algebraic equations with independent random coefficients," *Random Operators and Stochastic Equations*, vol. 26, no. 2, pp. 125–130, 2018.

[30] X. Yang, "Numerical contour integral methods for unsteady Stokes equations," *Computers & Mathematics with Applications*, vol. 75, no. 12, pp. 4414–4426, 2018.

[31] R. J. Attaullah and A. Jabeen, "Solution of the hiv infection model with full logistic proliferation and variable source term using galerkin scheme," *Matrix Science Mathematic*, vol. 4, no. 2, pp. 37–43, 2020.

[32] S. Lu, J. Guo, S. Liu et al., "An improved algorithm of drift compensation for olfactory sensors," *Applied Sciences*, vol. 12, no. 19, p. 9529, 2022.

[33] Y. Liu, B. Zhang, Y. N. Feng et al., "Development of 340-GHz transceiver front end based on gaas monolithic integration technology for THz active imaging array," *Applied Sciences*, vol. 10, no. 21, p. 7924, 2020.

[34] L. Sun, J. Hou, C. Xing, Z. Fang, and Z. Fang, "A robust hammerstein-wiener model identification method for highly nonlinear systems," *Processes*, vol. 10, no. 12, p. 2664, 2022.

[35] E. Grylonakis, C. Filelis-Papadopoulos, G. Gravvanis, and G. A. Gravvanis, "A class of unified transform techniques for solving linear elliptic PDEs in convex polygons," *Applied Numerical Mathematics*, vol. 129, pp. 159–180, 2018.

[36] Y. Wang, H. Cheng, Q. Hu et al., "Pore structure heterogeneity of Wufeng-Longmaxi shale, Sichuan Basin, China: evidence from gas physisorption and multifractal geometries," *Journal of Petroleum Science and Engineering*, vol. 208, Article ID 109313, 2022.

[37] T. M. Ni, Y. Yao, H. Chang et al., "LCHR-TSV: novel low cost and highly repairable honeycomb-based TSV redundancy architecture for clustered faults," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2938–2951, 2020.

[38] C. Böhning and H. C. Graf von Bothmer, "On stable rationality of some conic bundles and moduli spaces of Prymcurves," *Commentarii Mathematici Helvetici*, vol. 93, no. 1, pp. 133–155, 2018.

[39] X. Xu and L. Chen, "Projection of long-term care costs in China, 2020-2050, based on the bayesian quantile regression method," *Sustainability*, vol. 11, no. 13, p. 3530, 2019.

[40] H. B. Xu, C. W. Zhang, H. Li, and J. P. Ou, "Real-time hybrid simulation approach for performance validation of structural active control systems: a linear motor actuator based active mass driver case study," *Structural Control and Health Monitoring*, vol. 21, no. 4, pp. 574–589, 2014.

[41] Y. Rao, S. Kosari, A. Khan, N. Abbasizadeh, and N. Abbasizadeh, "A study on special kinds of derivations in ordered hyperrings," *Symmetry*, vol. 14, no. 10, p. 2205, 2022.

[42] M. Kornilova, V. Kovalnogov, R. Fedorov et al., "Zeroing neural network for pseudoinversionof an arbitrary timevarying matrix based on singular value decomposition," *Mathematics*, vol. 10, no. 8, p. 1208, 2022.

[43] Y. Rao, S. Kosari, J. Anitha, I. Rajasingh, H. Rashmanlou, and H. Rashmanlou, "Forcing parameters in fully connected cubic networks," *Mathematics*, vol. 10, no. 8, p. 1263, 2022.