Scientific Research Publishing

Advances in Linear Algebra & Matrix Theory, 2016, 6, 104-115
http://www.scirp.org/journal/alamt
ISSN Online: 2165-3348
ISSN Print: 2165-333X

A QMR-Type Algorithm for Drazin-Inverse Solution of Singular Nonsymmetric Linear Systems

Alireza Ataei

Department of Mathematics, Faculty of Science, Persian Gulf University, Bushehr, Iran
Email: ataei@pgu.ac.ir

How to cite this paper: Ataei, A. (2016) A QMR-Type Algorithm for Drazin-Inverse Solution of Singular Nonsymmetric Linear Systems. *Advances in Linear Algebra & Matrix Theory*, **6**, 104-115.
http://dx.doi.org/10.4236/alamt.2016.64011

Received: October 8, 2016
Accepted: November 26, 2016
Published: November 29, 2016

Abstract

In this paper, we propose DQMR algorithm for the Drazin-inverse solution of consistent or inconsistent linear systems of the form $Ax=b$ where $A\in\mathbb{C}^{N\times N}$ is a singular and in general non-hermitian matrix that has an arbitrary index. DQMR algorithm for singular systems is analogous to QMR algorithm for non-singular systems. We compare this algorithm with DGMRES by numerical experiments.

Keywords

Singular Linear Systems, DGMRES Method, Quasi-Minimal Residual Methods, Drazin-Inverse Solution, Krylov Subspace Methods

1. Introduction

Consider the linear system

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

where $A\in\mathbb{C}^{N\times N}$ is a singular matrix and $\mathrm{ind}(A)$ is arbitrary. Here $\mathrm{ind}(A)$, the index of A is the size of the largest Jordan block corresponding to the zero eigenvalue of A. We recall that the Drazin-inverse solution of (1) is the vector A^Db, where A^D is the Drazin-inverse of the singular matrix A. For the Drazin-inverse and its properties, we can refer to [1] or [2]. In the important special case $\mathrm{ind}(A)$, this matrix is called the group inverse of A and denoted by A^g. The Drazin-inverse has various applications in the theory of finite Markov chains [2], the study of singular differential and difference equations [2], the investigation of Cesaro-Neumann iterations [3], cryptography [4], iterative methods in numerical analysis, [5] [6], multibody system dynamics [7] and so forth. The problem of finding the solution of the form A^Db for (1) is very common in the literature and many different techniques have been developed

in order to solve it.

In [6] [8] [9] [10] [11], authors presented some Krylov subspace methods [9] to solve singular linear system with some restriction. However, the treatment of singular linear inconsistent system by Krylov subspace has been proved extremely hard. In [12], Sidi had not put any restrictions on the matrix A and the system (1). In his paper, the spectrum of A can have any shape and no restrictions are put on the linear system (1). The only assumption is that $\text{index}(A)$ is known. Although the $\text{index}(A)$ of A is overestimated, the method is valid.

In [12], Sidi proposed a general approach to Krylov subspace methods to compute Drazin-inverse solution. In addition, he presented several Krylov subspace methods of Arnoldi, DGCR and Lancoze types. Furthermore, in [13] [14], Sidi has continued to drive two Krylov subspace methods to compute $A^D b$. One is DGMRES method, which is the implementation of the DGCR method for singular systems which is analogues to GMRES for non-singular systems. Other is DBI-CG method which is Lanczos type algorithm [13]. DGMRES, like, GMRES method, is a stable numerically and economical computationally, which is a storage wise method. DBI-CG method, also like BI-CG for non-singular systems, is a fast algorithm, but when we need a high accuracy, the algorithm is invalid. DFOM algorithm is another implementation of the projection method for singular linear systems is analogues to Arnoldi for non-singular systems. DFOM algorithm may be less accurate but faster than DGMRES, and more precise and slower than DBI-CG [15].

In the present paper, the Drazin-Quasi-minimal residual algorithm (DQMR hereafter) is another implementation of the projection method for singular linear systems is analogues to Lanczos algorithm for non-singular systems. DGMRES algorithm, in practice, cannot afford to run the full algorithm and it is necessary to use restart. For difficult problems, in most cases, this results in extremely slow convergence, While DQMR algorithm can be implemented using only short recurrences and hence it can be computed with little work and low storage requirements per iteration.

The outline of this paper is as follows. In Section 2, we will provide a brief of summary of the review of the theorem and projection method in [12] which is relevant to us. We shall discuss the projection methods approach to solve (1) in general and DQMR particular. In Section 3, we will drive the DQMR method. We design DQMR when we set $\text{ind}(A)=0$ throughout, DQMR reduces to QMR. In this sense, DQMR is an extension of QMR that archives the Drazin-inverse solution of singular systems. In Section 4, by numerical examples, we show that the computation time and iteration number of DQMR algorithm is substantially less than that of DGMRES algorithm. Section 5 is devoted to concluding remarks.

2. Some Basic Theorem and Projection Methods for $A^D b$

The method we are interested in starts with an arbitrary initial vector x_0 and generate sequences of vector $x_1, x_2, \cdots$ according to

$$x_m = x_0 + q_{m-1}(A) r_0, \quad r_0 = b - Ax_0, \tag{2}$$

where $q_{m-1}(\lambda)$ is a polynomial in λ of degree at most $m-1$, given by

$$q_{m-1}(\lambda)=\sum_{i=1}^{m-a}c_i\lambda^{a+i-1},\quad a=\operatorname{ind}(A). \tag{3}$$

Let us define

$$p_{m-1}(\lambda)=1-\lambda q_{m-1}(\lambda)=1-\sum_{i=1}^{m-a}c_i\lambda^{a+i}. \tag{4}$$

We call $p_m(\lambda)$ the m^{th} residual polynomial since

$$r_m=b-Ax_m=\left[I-q_{m-1}(A)\right]r_0=p_m(A)r_0. \tag{5}$$

Note that

$$p_m(0)=1\quad\text{and}\quad p_m^{(i)}(0)=0,\quad i=1,2,\cdots,a \tag{6}$$

The condition (6) is due to Eiermann *et al.* [5].

For convenience we denote by Π_m the class of polynomials of degree at most m and define

$$\Pi_m^0=\left\{p\in\Pi_m : p_m(0)=1\text{ and }p_m^{(i)}(0)=0, i=1,2,\cdots,a\right\}. \tag{7}$$

Thus, the polynomial p_m described above is in Π_m^0.

The projection methods of [12] are now defined by demanding that the vector $A^a r_m$ to be orthogonal to a given W subspace of dimension $m-a$. In addition, If we denote by W the $N\times(m-a)$ matrix whose columns span the subspace W, then this orthogonality demand is equivalent to $W^*A^a r_0=0$. As we have $r_m=r_0+\sum_{i=1}^{m-a}c_iA^{i+a}r_0$ from (4), $W^*A^a r_m=0$ amounts to the requirement that $c_1,c_2,\cdots,c_{m-a}$ satisfy the linear system

$$W^*A^{a+1}Vc=W^*A^a r_0, \tag{8}$$

where $V\in\mathbb{C}^{N\times(m-a)}$ and $c\in\mathbb{C}^{m-a}$ are given by

$$V=\left[A^a r_0\mid A^{a+1}r_0\mid\cdots\mid A^{m-a}r_0\right]\quad\text{and}\quad c=\left[c_1,c_2,\cdots,c_{m-a}\right]^{\mathrm{T}}. \tag{9}$$

We see that unique solution for c exists provided $\det\left(W^*A^{a+1}V\right)\neq 0$, and when it does we have $x_m=x_0+V\left(W^*A^{a+1}V\right)^{-1}W^*A^a r_0$.

As we choose different W, we have a different algorithm: for DGMRES, we choose $W=A^{(a+1)}V$, for DBI-CG, we choose $W=\operatorname{span}\left\{A^*\breve{r}_0,\left(A^*\right)^2\breve{r}_0,\cdots,\left(A^*\right)^{m-a}\breve{r}_0\right\}$.

In this paper, for DQMR, we choose

$$W=\operatorname{span}\left\{A^a r_0, A^*\left(A^a r_0\right),\cdots,\left(A^*\right)^{m-a}A^a r_0\right\}.$$

We will mention several definitions and theorems, which have projection method converge below.

We will denote by $\hat{\mathcal{L}}$ the direct sum of the variant subspaces of A corresponding to its non-zeros eigenvalues, and by $\tilde{\mathcal{L}}$, its invariant subspaces corresponding to its zeros eigenvalue. Thus, $\hat{\mathcal{L}}$ is $R\left(A^a\right)$, the range of A^a, and $\tilde{\mathcal{L}}$ is $N\left(A^a\right)$, the null space of A^a. So every vector in $\mathbb{C}^N$ can be expressed as the sum of two vector, one in $\hat{\mathcal{L}}$

and other in $\tilde{\mathcal{L}}$.

Definition 1 [12]. *Let A be singular and* $\operatorname{ind}(A)=a$, *and let* $\hat{u}\in\hat{\mathcal{L}}$ *be given. Then a polynomial* $p(\lambda)$ *will be called the minimal a—incomplete polynomial of A with respect to the vector* $\hat{u}$ *if* $p(\lambda)\in\Pi_m^0$ *and m is the smallest possible one so that* $p(A)\hat{u}=0$.

The following theorems will ensure the success of projection method.

Theorem 1 [12]. $p(\lambda)$ *exists and is unique. Furthermore, its degree m satisfies* $q\le m\le q+a$, *where q is the degree of the minimal polynomial of A with respect to* $\hat{u}$.

The following result that is the justification of the above-mentioned projection approach is Theorem 4.2 in [12].

Theorem 2 *Let* $x_0=\hat{x}_0+\tilde{x}_0$, *where* $\hat{x}_0\in\hat{\mathcal{L}}$ *and* $\tilde{x}_0\in\tilde{\mathcal{L}}$, *are the initial vector in the projection method to compute* A^Db. *Moreover, Let also* $P(\lambda)$ *the minimal a—incomplete polynomial of A with respect to* $\hat{x}_0+A^Db$, *and let m be its degree. Finally, let* x_m *be the vector generated by the projection method through (2)-(8) with* $\det\left(W^*A^{a+1}V\right)\neq 0$. *Then* $x_m=A^Db+\tilde{x}_0$.

Obviously, of Theorem 2 if $\det\left(W^*A^{a+1}V\right)\neq 0$, the projection method will terminate successfully in a finite number of steps, this number being at most *N*. If we pick $x_0=0$, which also forces $\tilde{x}_0=0$, they produce the Drazine-inverse solution A^Db for (1) upon termination.

Theorem 3 [14]. *The vector* x_m *exists uniquely and unconditionally for all* $m\le m_0$, m_0 *being the degree of the minimal a-incomplete polynomial of A with respect to* $\hat{x}_0-A^Db\in\hat{\mathcal{L}}$. *Furthermore,* $x_m=\hat{x}_m+\tilde{x}_m$ *with* $\hat{x}_m\in\hat{\mathcal{L}}$ *and* $\tilde{x}_m=\tilde{x}_0$ *for all m.*

3. DQMR Algorithm

In this section, we will introduce a different implementation of projection method. The algorithm is analogous to QMR algorithm. We must note that in spite of the analogy, DQMR seems to be quite different from QMR, which is for non-singular systems.

As $x_m=x_0+\sum_{i=1}^{m-a}c_iA^{a+i-1}r_0$, we start with $v_1=w_1=A^ar_0/\left\|A^ar_0\right\|_2$, the lanczos algorithm [16] generates two sequences of vectors $v_1,v_2,\cdots,v_n$ and $w_1,w_2,\cdots,w_n,n=1,2,\cdots,$ that satisfy

$$\begin{aligned}&\operatorname{span}\{v_1,v_2,\cdots,v_k\}=\operatorname{span}\left\{A^ar_0,A^{a+1}r_0,\cdots,A^{a+k-1}r_0\right\}=K_k\left(A;A^ar_0\right)\\&\operatorname{span}\{w_1,w_2,\cdots,w_k\}=\operatorname{span}\left\{A^ar_0,A^*\left(A^ar_0\right),\cdots,\left(A^*\right)^{a+k-1}A^ar_0\right\}=K_k\left(A^*;A^ar_0\right).\end{aligned}\tag{10}$$

where they are clear that $K_k\left(A;A^ar_0\right)$ and $K_k\left(A^*;A^ar_0\right)$ denote the Krylov subspaces

$$\operatorname{span}\left\{A^ar_0,A^{a+1}r_0,\cdots,A^{a+k-1}r_0\right\}\ \text{and}\ \operatorname{span}\left\{A^ar_0,A^*A^ar_0,\cdots,\left(A^*\right)^{a+k-1}A^ar_0\right\},$$

respectively.

If we define that $N\times k$ matrix $\hat{V}_k$ by

$$\hat{V}_k=\begin{bmatrix}v_1 & v_2 & \cdots & v_k\end{bmatrix},\quad k=1,2,\cdots$$

then, we can write for $m\le m_0$

$$x_m = x_0 + \hat{V}_{m-a}\zeta_m, \quad \text{for some } \zeta_m \in C^{m-a}.$$

Therefore, it is obvious that we need to determine ζ_m. Since $r_m = r_0 - A\hat{V}_{m-a}\zeta_m$ we have

$$A^a r_m = A^a r_0 - A^{a+1}\hat{V}_{m-a}\zeta_m = \gamma_1 v_1 - A^{a+1}\hat{V}_{m-a}\zeta_m, \tag{11}$$

where $\gamma_1 = \left\|A^a r_0\right\|_2$.

Moreover, provided that $k \le q-1$, from (11) we can write

$$A\hat{V}_k = \hat{V}_{k+1}\overline{T}_k, \quad \hat{T}_m = \begin{bmatrix} \hat{t}_{11} & \hat{t}_{12} & \cdots & \hat{t}_{1,a+2} & 0 & \cdots & 0 \\ \hat{t}_{21} & \hat{t}_{22} & \cdots & \hat{t}_{2,a+2} & \hat{t}_{2,a+3} & \ddots & \vdots \\ \vdots & \hat{t}_{32} & \ddots & \vdots & \ddots & \ddots & \vdots \\ \hat{t}_{a+2,1} & \vdots & \ddots & \hat{t}_{a+2,a+2} & \vdots & \ddots & 0 \\ 0 & \hat{t}_{a+3,2} & \ddots & \hat{t}_{a+3,a+2} & \ddots & \vdots & \hat{t}_{m-2a-1,m-a} \\ \vdots & 0 & \ddots & \vdots & \ddots & \ddots & \hat{t}_{m-2a,m-a} \\ \vdots & & \ddots & \hat{t}_{2a+3,a+2} & \vdots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \vdots & \hat{t}_{m-a,m-a} \\ \vdots & & & & \ddots & \ddots & \hat{t}_{m-a+1,m-a} \\ \vdots & & & & \ddots & \ddots & \vdots \\ \vdots & & & & & \ddots & \vdots \\ \vdots & & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \quad \hat{t}_{m+1,m-a} \end{bmatrix}. \tag{12}$$

Note that $\hat{T}_m \in C^{(m+1)\times(m-a)}$. Since the vectors Av_1 Av_2 $\cdots$ Av_k are linearly independent when $k \le q-1$, we have $\operatorname{rank}\left(A\hat{V}_k\right) = k$. Since $\operatorname{rank}\left(\hat{V}_{k+1}\right) = k+1$, and $\operatorname{rank}\left(A\hat{V}_k\right) \le \min\left\{\operatorname{rank}\left(\hat{V}_k\right), \operatorname{rank}\left(\overline{T}_k\right)\right\}$, we also have that $\operatorname{rank}\left(\overline{T}_k\right) = k$. In other words, $\overline{T}_k$ has full rank. In additon, if we apply (12) to $A^{a+1}\hat{V}_{m-a}$. Provided that $m \le q-1$, we have:

$$\begin{aligned} A^{a+1}\hat{V}_{m-a} &= A^a\hat{V}_{m-a+1}\overline{T}_{m-a} = A^{a-1}\hat{V}_{m-a+2}\overline{T}_{m-a+1}\overline{T}_{m-a} \\ &= \cdots = \hat{V}_{m+1}\hat{V}_m, \quad \hat{V}_m \equiv \overline{T}_m\overline{T}_{m-1}\cdots\overline{T}_{m-a}. \end{aligned}$$

Consequently, provided $m \le q-1$, from (11) we can write:

$$A^a r_m = \gamma_1 v_1 - \hat{V}_{m+1}\hat{T}_m\zeta_m$$

and since $v_1 = \hat{V}_{m+1}e_1$, where $e_1 = [1,0,\cdots,0]^{\mathrm{T}} \in C^{m+1}$, hence

$$\left\|A^a r_m\right\|_2 = \left\|\hat{V}_{m+1}\left(\gamma_1 e_1 - \hat{T}_m\zeta_m\right)\right\|_2$$

If the column vectors of $\hat{V}_{m+1}$ were orthonormal, then we would have:

$$\left\|A^a r_m\right\|_2 = \left\|\gamma_1 e_1 - \hat{T}_m\zeta_m\right\|_2$$

as in GMRES. Therefore, a least, squares solution could be obtained from the krylove space $K_m\left(A; A^a r_0\right)$ by minimizing $\left\|\gamma_1 e_1 - \hat{T}_m\zeta\right\|_2$ over ζ. In the Lanczos algorithm, the v_i's are not orthonormal. However, it is still a reasonable idea to minimize the function

$$T(\zeta) = \left\|\beta e_1 - \hat{T}_m\zeta\right\|_2$$

over ζ and compute the corresponding approximate solution $x_m = x_0 + \hat{V}_{m-a}\zeta_m$. The resulting solution is called the Drazin-Quasi-minimal residual (hereafter DQMR) approximation. Since $\overline{T}_m \in C^{(m+1)\times m}$ is a tridiagonal matrix, Therefore, the $\hat{T}_m \in C^{(m+1)\times(m-a)}$ is a matrix with $2a+3$ diagonal to form (12).

Similar to [14], $\hat{T}_{m+1}$ can be obtained as a simple update of $\hat{T}_m$ by first appending a row of zeros at the bottom of $\hat{T}_m$ and following that by appending the $(m+1)$-vector $\left[\hat{t}_{m-a,m+1-a}, \hat{t}_{m+1-2a,m+1-a}, \cdots, \hat{t}_{m+2,m+1-a}\right]^{\mathrm{T}}$ as the $(m+1-a)^{th}$ column as follows.

Let us define

$$G_k^{(0)} = \overline{T}_k,\ k = 1, 2, \cdots \tag{13}$$

$$G_k^{(j)} = \overline{T}_k G_{k-1}^{(j-1)},\ k = j+1,\ j+2, \cdots;\ j = 1, 2, \cdots \tag{14}$$

where $G_k^{(j)} \in C^{(k+1)\times(k-j)}$ also $G_m^{(a)} = \hat{T}_m$ for each $m \geq a+1$.

Since $\hat{T}_m$ is tridiagonal matrix we have:

$$\overline{T}_{m+1} = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \delta_2 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{m+1} \\ & & \delta_{m+1} & \alpha_{m+1} \\ & & & \delta_{m+2} \end{bmatrix} = \left[\begin{array}{c|c} \overline{T}_m & g_{m+1}^{(0)} \\ \hline 0_m^{\mathrm{T}} & \alpha_{m+1}^{(0)} \end{array}\right],$$

where, certainly, $g_{m+1}^{(0)} = \begin{bmatrix} 0 & 0 & \cdots & \beta_{m+1} & \alpha_{m+1} \end{bmatrix}^{\mathrm{T}} \in C^{m+1}$, 0_k denotes the k-dimensional (column) zeros vector, and $\alpha_{m+1}^{(0)} = \delta_{m+2}$ that is scalar.

$$\text{Supposed that } T_{m+1}^{(s+1)} = \left[\begin{array}{c|c} T_m^{(s+1)} & g_{m+1}^{(s+1)} \\ \hline O_{m-s-1}^{\mathrm{T}} & \alpha_{m+1}^{(s+1)} \end{array}\right],\ m \geq s+2. \tag{15}$$

$$g_{m+1}^{(s+1)} = \overline{T}_m g_m^{(s)} + \alpha_m^{(s)} g_{m+1}^{(0)}, \tag{16}$$

$$\text{From [14], we have: } \alpha_{m+1}^{(s+1)} = \alpha_{m+1}^{(0)} \alpha_m^{(s)}. \tag{17}$$

Equation (16) can be simplified as follows:

$$\begin{aligned} \overline{T}_m g_m^{(s)} &= \begin{bmatrix} \alpha_1 & \beta_2 & \cdots & \cdots & 0 \\ \delta_2 & \alpha_2 & \ddots & & \vdots \\ 0 & \delta_3 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \beta_m \\ \vdots & & \ddots & \ddots & \alpha_m \\ 0 & \cdots & \cdots & 0 & \delta_m \end{bmatrix} \begin{bmatrix} g_{1,m}^{(s)} \\ g_{2,m}^{(s)} \\ \vdots \\ g_{m,m}^{(s)} \end{bmatrix} = \begin{bmatrix} \alpha_1 g_{1,m}^{(s)} + \beta_2 g_{2,m}^{(s)} \\ \delta_2 g_{1,m}^{(s)} + \alpha_2 g_{2,m}^{(s)} + \beta_3 g_{3,m}^{(s)} \\ \delta_3 g_{2,m}^{(s)} + \alpha_3 g_{3,m}^{(s)} + \beta_4 g_{4,m}^{(s)} \\ \vdots \\ \delta_m g_{m-1,m}^{(s)} + \alpha_m g_{m,m}^{(s)} \\ \delta_{m+1} \cdot g_{m,m}^{(s)} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \delta_2 \cdot g_{1,m}^{(s)} \\ \delta_3 \cdot g_{2,m}^{(s)} \\ \vdots \\ \delta_m \cdot g_{m-1,m}^{(s)} \\ \delta_{m+1} \cdot g_{m,m}^{(s)} \end{bmatrix} + \begin{bmatrix} \alpha_1 \cdot g_{1,m}^{(s)} \\ \alpha_2 \cdot g_{2,m}^{(s)} \\ \alpha_3 \cdot g_{3,m}^{(s)} \\ \vdots \\ \alpha_m \cdot g_{m,m}^{(s)} \\ 0 \end{bmatrix} + \begin{bmatrix} \beta_2 \cdot g_{2,m}^{(s)} \\ \beta_3 \cdot g_{3,m}^{(s)} \\ \vdots \\ \beta_m \cdot g_{m,m}^{(s)} \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{18}$$

By using the Hadamard product Equation (18) is reduced. For this purpose, we first introduce the concepts of Hadamard matrix product.

Definition 2 *Let A and B be* $m\times n$ *matrices with entries in C. The Hadamard product of A and B is defined by as follows*

$$A\odot B=\begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1n}b_{1n} \\ \vdots & \vdots & & \vdots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & \cdots & a_{mn}b_{mn} \end{pmatrix}.$$

Let us denote

$$\Delta_m=\begin{bmatrix}\delta_2 & \delta_3 & \cdots & \delta_{m+1}\end{bmatrix}^{\mathrm{T}},\quad \bar{\alpha}_m=\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_m\end{bmatrix}^{\mathrm{T}},\quad \bar{\beta}_m=\begin{bmatrix}\beta_2 & \beta_3 & \cdots & \beta_m\end{bmatrix}^{\mathrm{T}},$$

$$g_m^{(s)}=\begin{bmatrix}g_{1,m}^{(s)} & g_{2,m}^{(s)} & \cdots & g_{m,m}^{(s)}\end{bmatrix}^{\mathrm{T}},\quad \underline{g}_m^{(s)}=\begin{bmatrix}g_{2,m}^{(s)} & g_{3,m}^{(s)} & \cdots & g_{m,m}^{(s)}\end{bmatrix}^{\mathrm{T}}.$$

Now, we can be simplified (18) as follows

$$\bar{T}_m g_m^{(s)}=\begin{bmatrix}0\\ \Delta_m\odot g_m^{(s)}\end{bmatrix}+\begin{bmatrix}\bar{\alpha}_m\odot g_m^{(s)}\\ 0\end{bmatrix}+\begin{bmatrix}\bar{\beta}_m\odot \underline{g}_m^{(s)}\\ 0\\ 0\end{bmatrix} \tag{19}$$

For solution system

$$\hat{T}_m\zeta=\beta e_1 \tag{20}$$

We must reduce the band matrix, $\hat{T}_m$, into upper triangular by using Givens rotation. $\hat{T}_m$ matrix has bandwidth $2a+3$. To reduce the matrix $\hat{T}_m$ to a upper triangular matrix we need to $(m-a)(a+1)$ Givens rotations matrix. We denote with $\bar{g}_0=\beta e_1$ right-hand side (20), and we multiply both sides of (20) from left by Givens rotations. To update the mth column of matrix $\hat{T}_m$, we must first multiply the previous Givens rotations by this column and then we annihilate the main subdiagonal elements with appropriate rotations. It should be noted that number of the previous rotations is $\min(m-1,2a+2)\times(a+1)$, and the number of the rotations to annihilate the main subdiagonal elements is $(a+1)$. Finally, the mth end step we have an upper triangular matrix as follows

$$\hat{R}_m=\begin{bmatrix}
\hat{r}_{1,1} & \hat{r}_{1,2} & \cdots & \hat{r}_{1,2a+3} & 0 & \cdots & \cdots & 0\\
0 & \hat{r}_{2,2} & \cdots & \hat{r}_{2,2a+3} & \hat{r}_{2,2a+4} & \cdots & \cdots & 0\\
\vdots & \ddots & \ddots & \vdots & & \ddots & & \vdots\\
\vdots & & \ddots & \hat{r}_{2a+3,2a+3} & & & \ddots & 0\\
 & & & \ddots & \hat{r}_{2a+4,2a+4} & & & \hat{r}_{m-3a-2,m-a}\\
 & & & & \ddots & \ddots & & \vdots\\
 & & & & & \ddots & \ddots & \hat{r}_{m-a-1,m-a}\\
 & & & & & & \ddots & \hat{r}_{m-a,m-a}\\
 & & & & & & & 0\\
 & & & & & & & \vdots\\
0 & \cdots & & & & & & 0
\end{bmatrix} \tag{21}$$

Generally, if we define $\Omega^{(m-a)}$ the product of matrices $\Omega_{i+1,i}^{(m)}$, then

$$\Omega^{(m-a)} = \Omega_{m-a+1,m-a}^{(m-a)} \cdots \Omega_{m,m-1}^{(m-a)} \Omega_{m+1,m}^{(m-a)}, \quad i = m+1, m, \cdots, m-a$$

where $\Omega_{i+1,i}^{(m)}$ be the Givens matrices use to transform $\hat{T}_m$ into an upper triangular $\hat{R}_m$ and the vector of $\overline{g}_m = \Omega^{(m-a)}(\gamma e_1)$. Finally , the approximate solution is given by

$$x_m = x_0 + \hat{V}_{m-a} R_{m-a}^{-1} g_{m-a},$$

where R_{m-a} and g_{m-a} are obtained by removing the $a+1$ row of the matrix $\hat{R}_m$ and right-hand side $\overline{g}_m$. The approximation solution x_m can be updated at each step by the relation,

$$x_m = x_{m-1} + p_{m-a}\gamma_{m-a}. \tag{22}$$

Since if we assume $P_{m-a} = [p_1 \cdots p_{m-a}]$, then we have:

$$P_{m-a} = \hat{V}_{m-a} R_{m-a}^{-1}.$$

Consequently,

$$[p_1 \cdots p_{m-a}] R_{m-a} = [v_1 \cdots v_{m-a}],$$

and

$$p_{m-a} = \left(v_{m-a} - \sum_{i=m-3a-2}^{m-a-1} p_i \hat{r}_{i,m-a} \right) \Big/ \hat{r}_{m-a,m-a},$$

where p_{m-a} is the last column of P_{m-a}. Therefore, it can be written

$$x_m = x_0 + P_{m-a} g_{m-a} = x_0 + [P_{m-a-1}, p_{m-a}] \begin{bmatrix} g_{m-a-1} \\ \gamma_{m-a} \end{bmatrix}$$

or

$$x_m = x_{m-1} + p_{m-a}\gamma_{m-a}. \tag{23}$$

Thus, x_m can be updated easily at each step, via the relation (23) using x_{m-1}.

This gives the following algorithm, which we call the Drazin-QMR for Drazin-inverse solution of singular nonsymmetric linear equations.

Algorithm 4.1 DQMR Algorithm

1. *Pick* x_0 *and compute* $r_0 = b - Ax_0$ *and* $A^a r_0$.
2. *Compute* $\gamma_1 = \left\| A^a r_0 \right\|_2$ *and set* $v_1 = (A^a r_0)/\gamma_1$.
3. *For* $i = 1, 2, 3, \cdots, m$.
4. *Compute* $\alpha_i, \delta_{i+1}, \beta_{i+1}$ *and* v_{i+1}, w_{i+1} as in ([17], Algorithm 7.1).
5. *For* $k = 1, 2, \cdots, m$ *form the matrices* $\widehat{V}_k \in R^{n\times k}$ *and* $\overline{T}_k \in R^{(k+1)\times k}$:

$$\widehat{V}_k = [v_1\ v_2 \cdots v_k] \text{ and } \overline{T}_k = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & \cdots & 0 \\ \delta_2 & \alpha_2 & \beta_3 & \ddots & & \vdots \\ 0 & \delta_3 & \ddots & \ddots & & 0 \\ \vdots & \ddots & & \ddots & & \beta_m \\ \vdots & & \ddots & & & \alpha_m \\ 0 & \cdots & 0 & & & \delta_{m+1} \end{bmatrix}.$$

6. *Compute the matrix* $\widehat{T}_m = \overline{T}_m \overline{T}_{m-1} \cdots \overline{T}_{m-a}$ *by using* (15).

7. *Update the QR factorization of* $\widehat{T}_m$ *,i.e,*

8. *Set* $i_1 = \min(m-1, 2a+2)$.

9. *Apply* $\Omega^{(i)}$ $, i = i_1 : m-a+1$ *to the* $(m-a)$th *column of* $\widehat{T}_m$.

10. *For* $i = 1 : a+1$.

11.*Compute the rotation coefficients* s_i, c_i, *of rotation matrix* $\Omega^{m-a}_{m+2-i,m+1-i}$.

12.*Apply rotation* $\Omega^{(m-a)}_{m+2-i,m+1-i}$ *to the* $(m-a)$th *column of* $\widehat{T}_m$ *and* $\overline{g}_m$.

13. EndFor.

14. *Compute* $p_{m-a} = (v_{m-a} - \sum_{i=m-3a-2}^{m-a-1} p_i \widehat{t}_{i,m-a}) / \widehat{t}_{m-a,m-a}$.

15. *Compute* $x_m = x_{m-1} - p_{m-a}\gamma_{m-a}$.

16. End Do.

4. Numerical Examples

In this section, we will compute the linear system $Ax = b$ by discretization Poisson equation with Neumann boundary conditions:

$$\begin{cases} \left(\dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2}\right) u(x,y) = f(x,y), & (x,y) \in \Omega = [0,1] \times [0,1] \\ \dfrac{\partial}{\partial n} u(x,y) = \varphi(x,y) & x, y \in \partial\Omega. \end{cases}$$

This linear system has also been computed by Sidi [14] for testing DGMRES algorithm. The problem has also been considered by Hank and Hochbruck [18] for testing the Chebyshev-type semi-iterative method. The numerical computations are performed in MATLAB (R213a) with double precision. The results were obtains by running the code on an Intel (R) Core (TM) i7-2600 CPU Processor running 3.40 GHz with 8 GB of RAM memory using Windows 7 professional 64-bit operating system. The initial vector x_0 is the zero vector. All the tests were stopped as soon as

$$\text{Relative Error} = \frac{\left\| x_n - A^D b \right\|_\infty}{\left\| A^D b \right\|_\infty} \le 10^{-8}.$$

Let M be an odd integer, we discretize the Poisson equation on a uniform grid of mesh size $h = 1/M$ via central differences, and then by taking the unknowns in the red-black order we obtain the system $Ax = b$, where the $(M+1)^2 \times (M+1)^2$ non-symmetric matrix A is as follows

$$A=\begin{bmatrix}
4I & o & \cdots & \cdots & \cdots & \cdots & \cdots & o & T_1 & -2I & o & \cdots & \cdots & \cdots & \cdots & o \\
o & 4I & \ddots & & & & & \vdots & -I & T_2 & -I & o & & & & \vdots \\
\vdots & \ddots & \ddots & \ddots & & & & \vdots & o & -I & T_1 & -I & o & & & \vdots \\
\vdots & & \ddots & \ddots & \ddots & & & \vdots & \vdots & o & -I & T_2 & -I & o & & \vdots \\
\vdots & & & \ddots & \ddots & \ddots & & \vdots & \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & & \ddots & \ddots & \ddots & \vdots & \vdots & & & \ddots & \ddots & \ddots & \ddots & o \\
\vdots & & & & & \ddots & 4I & o & \vdots & & & & o & -I & T_1 & -I \\
o & \cdots & \cdots & \cdots & \cdots & \cdots & o & 4I & o & \cdots & \cdots & \cdots & \cdots & o & -2I & T_2 \\
T_2 & -2I & o & \cdots & \cdots & \cdots & \cdots & o & 4I & o & \cdots & \cdots & \cdots & \cdots & \cdots & o \\
-I & T_1 & -I & o & & & & \vdots & o & 4I & \ddots & & & & & \vdots \\
o & -I & T_2 & -I & o & & & \vdots & \vdots & \ddots & \ddots & \ddots & & & & \vdots \\
\vdots & o & -I & T_1 & -I & o & & \vdots & \vdots & & \ddots & \ddots & \ddots & & & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & & & \ddots & \ddots & \ddots & & \vdots \\
\vdots & & & \ddots & \ddots & \ddots & \ddots & o & \vdots & & & & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & & o & -I & T_2 & -I & \vdots & & & & & \ddots & 4I & o \\
o & \cdots & \cdots & \cdots & \cdots & o & -2I & T_1 & o & \cdots & \cdots & \cdots & \cdots & \cdots & o & 4I
\end{bmatrix}. \quad (24)$$

Here, I and 0 denote, respectively, the $(M+1)/2\times(M+1)/2$ identity and zero matrices and the $(M+1)/2\times(M+1)/2$ matrices T_1 and T_2 are given by

$$T_1=\begin{bmatrix}
-2 & o & \cdots & \cdots & o \\
-1 & -1 & \ddots & & \vdots \\
o & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & & -1 & o \\
o & & o & -1 & -1
\end{bmatrix}, \qquad T_2=\begin{bmatrix}
-1 & -1 & o & \cdots & o \\
o & -1 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & o \\
\vdots & & & -1 & -1 \\
o & \cdots & \cdots & o & -2
\end{bmatrix}.$$

The numerical experiment was performed for $M=31,63$.

It should be noted A is singular with 1D null space spanned by the vector $e=[1,\cdots,1]^{\mathrm{T}}$. Furthermore, $\mathrm{ind}(A)=1$, as mentioned in [13] [14] [15] [18]. Even if continues problem has a solution, the discretized problem need not be consistent. In the sequel we consider only the Drazin-inverse solution of the system for arbitrary right side b, not necessarily related to f and φ.

We first construct a consistent system with the known solution $\hat{s}\in R(A)$ via $\hat{s}=Ay$, where $y=[0,\cdots,0,1]^{\mathrm{T}}$. Then we perturb $A\hat{s}$, the right-hand side of $Ax=A\hat{s}=\hat{b}$, with a constant multiple of the null space vector e and we obtain the right-hand side

$$b=\hat{b}+\delta\frac{e}{\|e\|_2}.$$

Consequently the system $Ax=\hat{b}+\delta\dfrac{e}{\|e\|_2}$ is solved for x. The perturbation parameter δ is selected as 10^{-2} in our experiments. The solution we intend to obtain is the vector $\hat{s}$, whose components are zeros except

$$\hat{s}_{2\hat{M}^2-\hat{M}}=-1,\ \hat{s}_{2\hat{M}^2-1}=-1,\ \hat{s}_{2\hat{M}^2}=-2,\ \hat{s}_{4\hat{M}^2}=4,\ \text{where } \hat{M}=(M+1)/2.$$

In Table 1, Table 2, we give the number of iterations (Its), the CPU time (Time)

Table 1. Application of DQMR implementation to the consistent singular system.

Size of A		1024×1024		4096×4096		
Method	Its	Time	RE	Its	Time	RE
DQMR	155	0.61	5.9674e−09	267	7.39	8.8234e−09
DGMRES	165	0.88	3.0294e−09	307	10.89	8.1578e−09

Table 2. Application of DQMR implementation to the inconsistent singular system.

Size of A		4096×4096		16384×16384		
Method	Its	Time	RE	Its	Time	RE
DQMR	155	0.53	5.9674e−09	267	6.88	8.8234e−09
DGMRES	165	0.81	3.0294e−09	307	10.34	8.1578e−09

required for convergence, the relative error (RE), for the DGMRES and DQMR methods. As shown in Table 1, Table 2 the DQMR algorithm is effective and less expensive than the DGMRES algorithm.

5. Conclusion

In this paper, we presented a new method, called DQMR, for Drazin-inverse solution of singular nonsymmetric linear systems. The DQMR algorithm for singular systems is analogous to the QMR algorithm for non-singular systems. Numerical experiments indicate that the Drazin-inverse solution obtained by this method is reasonably accurate, and its computation time is less than that of solution obtained by the DGMRES method. Thus, we can conclude that the DQMR algorithm is a robust and efficient tool to compute the Drazin-inverse solution of singular linear systems.

Acknowledgements

First, we would like to thank professor F. Toutounian for her comments that improved our results. Second, we thank the editor and the referees for their carefully reading and useful comments.

References

[1] Ben-Israel, A. and Grevile, T.N.E. (2003) Generalized Inverses: Theory and Applications. 2nd Edition, Springer-Verlag, New York.

[2] Campell, S.L. and Meyer, C.D. (1979) Generalized Inverses of Linear Transformations. Pitman (Advanced Publishing Program), Boston.

[3] Hartwig, R.E. and Hall, F. (1982) Applications of the Drazin Inverse to Cesaro-Neumann Iterations. In: Campbell, S.L., Ed., *Recent Applications of Generalized Inverses*, **66**, 145-195.

[4] Hartwig, R.E. and Levine, J. (1981) Applications of the Drazin Inverse to the Hill Cryptographic system. *Part III*, *Cryptologia*, **5**, 67-77. https://doi.org/10.1080/0161-118191855850

[5] Eiermann, M., Marek, I. and Niethammer, W. (1988) On the Solution of Singular Linear Systems of Algebric Equations by Semiiterative Methods. *Numerische Mathematik*, **53**, 265-283. https://doi.org/10.1007/BF01404464

[6] Freund, R.W. and Hochbruck, M. (1994) On the Use of Two QMR Algorithms for Solving Singular Systems and Applications in Markov Chain Modeling. *Numerical Linear Algebra with Applications*, **1**, 403-420. https://doi.org/10.1002/nla.1680010406

[7] Simeon, B., Fuhrer, C. and Rentrop, P. (1993) The Drazin Inverse in Multibody System Dynamics. *Numerische Mathematik*, **64**, 521-539. https://doi.org/10.1007/BF01388703

[8] Brown, P.N. and Walker, H.F. (1997) GMRES on (Nearly) Singular Systems. *SIAM Journal on Matrix Analysis and Applications*, **18**, 37-51. https://doi.org/10.1137/S0895479894262339

[9] Ipsen, I.F. and Meyer, C.D. (1998) The Idea behind Krylov Methods. *The American Mathematical Monthly*, **105**, 889-899. https://doi.org/10.2307/2589281

[10] Smoch, L. (1999) Some Result about GMRES in the Singular Case. *Numerical Algorithms*, **22**, 193-212. https://doi.org/10.1023/A:1019162908926

[11] Wei, Y. and Wu, H. (2000) Convergence Properties of Krylov Subspace Methods for Singular System with Arbitrary Index. *Journal of Computational and Applied Mathematics*, **114**, 305-318. https://doi.org/10.1016/S0377-0427(99)90237-6

[12] Sidi, A. (1999) A Unified Approach to Krylov Subspace Methods for the Drazin-Inverse Solution of Singular Non-Symmetric Linear Systems. *Linear Algebra and Its Applications*, **298**, 99-113. https://doi.org/10.1016/S0024-3795(99)00153-6

[13] Sidi, A. and Kluzner, V. (1999) A Bi-CG Type Iterative Method for Drazin Inverse Solution of Singular Inconsistent Non-Symmetric Linear Systems of Arbitrary Index. *The Electronic Journal of Linear Algebra*, **6**, 72-94.

[14] Sidi, A. (2001) DGMRES: A GMRES-Type Algorithm for Drazin-Inverse Solution of Singular Non-Symmetric Linear Systems. *Linear Algebra and Its Applications*, **335**, 189-204. https://doi.org/10.1016/S0024-3795(01)00289-0

[15] Zhou, J. and Wei, Y. (2004) DFOM Algorithm and Error Analysis for Projection Methods for Solving Singular Linear System. *Applied Mathematics and Computation*, **157**, 313-329. https://doi.org/10.1016/j.amc.2003.08.036

[16] Lanczos, C. (1950) An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *Journal of Research of the National Bureau of Standards*, **45**, 255-282. https://doi.org/10.6028/jres.045.026

[17] Saad, Y. (2003) Iterative Methods for Sparse Linear Systems. 2nd Edition, SIAM, Philadelphia. https://doi.org/10.1137/1.9780898718003

[18] Hank, M. and Hochbruck, M. (1993) A Chebyshev-Like Semiiteration for Inconsistent Linear Systems. *Electronic Transactions on Numerical Analysis*, **1**, 315-339.