Journal of Applied Analysis and Computation
Volume 10, Number 3, June 2020, 1047–1059

Website:http://jaac-online.com/
DOI:10.11948/20190185

KRYLOV SUBSPACE METHODS OF HESSENBERG BASED FOR ALGEBRAIC RICCATI EQUATION*

Yajun Xie[1,†], Minhua Yin[2] and Limin Ren[3]

Abstract In this paper, we propose a class of special Krylov subspace methods to solve continuous algebraic Riccati equation (CARE), i.e., the Hessenberg-based methods. The presented approaches can obtain efficiently the solution of algebraic Riccati equation to some extent. The main idea is to apply Kleinman-Newton's method to transform the process of solving algebraic Riccati equation into Lyapunov equation at every inner iteration. Further, the Hessenberg process of pivoting strategy combined with Petrov-Galerkin condition and minimal norm condition is discussed for solving the Lyapunov equation in detail, then we get two methods, namely global generalized Hessenberg (GHESS) and changing minimal residual methods based on the Hessenberg process (CMRH) for solving CARE, respectively. Numerical experiments illustrate the efficiency of the provided methods.

Keywords Continuous algebraic Riccati equation (CARE), Krylov subspace method, Hessenberg-based method, Pivoting strategy, Petrov-Galerkin condition.

MSC(2010) 65H10, 65K05, 49M15.

1. Introduction

In this paper, we will discuss the following continuous algebraic Riccati equation (CARE)

$$XBX - XA - A^TX - C = 0, \tag{1.1}$$

where $A,\ B,\ C \in \mathbb{R}^{n\times n}$, $B,\ C$ are symmetric and positive semidefinite, and the undetermined solution $X \in \mathbb{R}^{n\times n}$ is also symmetric, positive semidefinite and stabilizing, namely, $BX - A$ is stable according to relevant control theory, which implies the eigenvalues of matrix $BX - A$ lie in the open left half-plane [9].

The continuous algebraic equation (1.1) have been investigated extensively due to various scientific and engineering applications, especially for control theory and

†The corresponding author. Email: xieyajun0525@163.com(Y. Xie)
1College of Technology, Fuzhou University of International Studies and Trade, Fuzhou 350202, China
2College of industry and commerce, Fujian Jiangxia university, Fuzhou 350108, China
3Department of Mathematics & Physics, Fujian Jiangxia university, Fuzhou 350108, China
*The project is supported by National Natural Science Foundation of China (Grant No.11071041), Fujian Natural Science Foundation (Grant Nos.2019J01879,2016J01005); The New Century Training Plan of Fujian Province University and The educational reform project (Grant No. 2017[52], J2018B013).

dynamical problems [2, 5, 9, 19]. Normally, it is difficult to get the exactly solution even when n isn't so large, which greatly promotes the substantial developments of constructing various kinds of iterative techniques. Many research work have been investigated in some literatures on fast solvers for the continuous algebraic Riccati equations (1.1). Some of the most classical methods are Newton's method [6,23] and popular structure-preserving doubling algorithm [7, 11]. Chu et al. [8] developed a numerical method for generalized algebraic Riccati equation. Their method consists of computations of the eigendecomposition of the system pencil corresponding to the eigenvalues on the extended imaginary axis and the stable eigenspace of an augmented matrix pencil, which is a generalization of the generalized eigenvalue approach for classical algebraic Riccati equation. Lu in [17] provided the solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory, which is much more efficient than the Gauss-Jacobi method given by Juang in [16]. Furthermore, Bao et al. [4] proposed a modified simple iterative method for nonsymmetric algebraic Riccati equations, which is an improvement version for [17].

To our knowledge, however, there is still not so many ongoing researches from the perspective of Krylov subspace methods. Jbilou proposed the block Krylov subspace methods for large algebraic Riccati equation, which used the block Arnoldi process to construct an orthogonal basis of the corresponding block Krylov subspace and then extracted low rank approximate solutions [13]. Amodei et al. presented an invariant subspace method for large-scale algebraic Riccati equation, which is a new family of low-rank approximations of the solution of the algebraic Riccati equation by considering stable invariant subspaces of the Hamiltonian matrix [1]. In [24], Simoncini gave two numerical methods for the solution of large-scale algebraic Riccati equation, which can be considered as Galerkin projection (GP) method.

As is known to all, some large linear systems, especially for those arising from discretization by finite differences or by finite elements, can be solved rapidly by efficient iterative methods, such as those based on Krylov subspace [10]. Roughly, Krylov subspace methods can be classified as three types: Arnoldi, Lanczos, and Hessenberg based methods. Such as, Generalized Minimum Residual method (GMRES) implements the Arnoldi process and Quasi-Minimal Residual method (QMR) is based on the Lanczos process. However there is possible the occurrence of breakdown or near-breakdown for the Lanczos algorithm. Another efficient subspace technique is Hessenberg method which is based on an upper Hessenberg matrix to construct a basis for the Krylov subspace. The approach has been shown that it requires less work and storage than Arnoldi's method. In [21], Sadok introduced firstly an novel and interesting approach by replacing in QMR the Lanczos algorithm by the Hessenberg process, which was simply denoted as CMRH (Changing Minimal Residual method based on the Hessenberg process). For more details, see [12, 22, 25].

In fact, the above subspace methods can be also generalized to solve large linear matrix systems, such as Sylvester matrix equation or Lyapunov matrix equation, even to solve nonlinear matrix equations, such as algebraic Riccati equation. In this paper, we will first apply the Hessenberg-based methods, including global generalized Hessenberg (GHESS) and CMRH, to solve the continuous algebraic Riccati equation. Our numerical tests illustrate the proposed methods for solving the continuous algebraic Riccati equation are quite efficient.

We use the following notation throughout this paper as a matter of convenience.

Given two real $n \times m$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, the Frobenius inner product of matrices A and B is defined by $\langle A, B\rangle_F = tr(A^T B)$ in which $tr(\cdot)$ and A^T denotes the trace of matrix and transpose of A, respectively. Furthermore, $\|A\|_F$ denotes the Frobenius norm of matrix A. For a reversible matrix A, A^{-1} stands for the inverse of matrix A. Finally, we assume that R is any real $n \times m$ matrix, the K-dimension matrix Krylov subspace associated to the pair $(\mathcal{Q}, R)$ is defined by

$$\mathcal{K}_K(\mathcal{Q}, R) = span\{R, \mathcal{Q}(R), \cdots, \mathcal{Q}^{K-1}(R)\}, \tag{1.2}$$

where $\mathcal{Q}^i(R) = \mathcal{Q}(\mathcal{Q}^{i-1}(R))$ $(i = 0, \cdots, K)$ and $\mathcal{Q}^0(R) = R$.

The remainder of this paper is organized as follows. In Section 2, Kleinman-Newton's method for CARE will be described simply. In Section 3. we will introduce the Hessenberg-based methods, including the CMRH method with maximum strategy and the construction of its preconditioner, to solve the continuous algebraic Riccati equation in detail. Some extensions are proposed in Section 4. In Section 5, numerical tests are provided to illustrate the superiority of the presented iteration methods. Finally, a concluding remark is given in Section 6.

2. Kleinman-Newton's method for CARE

First of all, we describe briefly the general Newton's method for solving a differentiable nonlinear equation $f(x) = 0$. The iterative scheme is

$$x^{k+1} = x^k - (f'(x^k))^{-1} f(x^k), \tag{2.1}$$

where $f'(x^k)$ denotes the Fréchet derivative of the map f at x^k. Now, in order to utilize the Fréchet derivative in continuous algebraic Riccati equations (1.1), we write the following formulas:

$$\mathcal{L}(X) = XBX - XA - A^T X - C, \tag{2.2}$$

$$\mathcal{L}'(X)\widehat{R} = \widehat{R}(BX - A) + (BX - A)^T \widehat{R}, \tag{2.3}$$

where $X^T = X,\ B^T = B$.

On the basis of the following matrix iterative scheme of Newton's method

$$X_{k+1} = X_k - (\mathcal{L}'(X_k))^{-1}\mathcal{L}(X_k), \tag{2.4}$$

we get

$$\mathcal{L}'(X_k)\widehat{R}_k = -\mathcal{L}(X_k), \tag{2.5}$$

where $\widehat{R}_k := X_{k+1} - X_k$. Furthermore, we have

$$\widehat{R}_k(BX_k - A) + (BX_k - A)^T \widehat{R}_k = -X_k B X_k + X_k A + A^T X_k + C. \tag{2.6}$$

Evidently, one can solve the above Lyapunov equation about $\widehat{R}_k$, then getting $X_{k+1} = X_k + \widehat{R}_k$ by given the known point X_k at previous step.

As a matter of fact, if the initial matrix X_0 is symmetric then the sequence $\{X_k\}$ generated by Newton's method is also symmetric. Observing that (2.5) can be rewritten as

$$\mathcal{L}'(X_k)X_{k+1} = \mathcal{L}'(X_k)X_k - \mathcal{L}(X_k). \tag{2.7}$$

Accordingly, it obtains

$$X_{k+1}\mathcal{A} + \mathcal{A}^T X_{k+1} = X_k B X_k + C, \tag{2.8}$$

where $\mathcal{A}_k = BX_k - A$. So, it can be simply described as the following algorithm [6].

Algorithm 1 Kleinman-Newton's method for CARE

Input: Initial guess matrix $X_0 \in \mathbb{R}^{n\times n}$, such that $X_0 = (X_0)^T$ and $BX_0 - A$ is stable.
for $k = 0, 1, \dots \cdots$, until convergence **do**
  Set $\mathcal{A}_k = BX_k - A$;
  Solve $X_{k+1}\mathcal{A}_k + (\mathcal{A}_k)^T X_{k+1} = X_k B X_k + C$;
end for

Next section, we focus on the technique about the Lyapunov equation above Algorithm 1. For the sake of convenience, we first denote

$$\mathcal{Q}(X) := \mathcal{A}^T X + X\mathcal{A}, \tag{2.9}$$

where $\mathcal{A} = BX - A$. Therefore, the step 2 of Algorithm 1 can be seen to the Lyapunov equation as follows

$$\mathcal{Q}(X) = \widetilde{C}, \tag{2.10}$$

where $\widetilde{C} = \overline{X}B\overline{X} + C$, and $\overline{X}$ is given.

3. The description of Hessenberg based process

In this section, we will exhibit the global generalized Hessenberg process, which includes the Arnoldi and global Hessenberg processes as its special cases.

Let $R \in \mathbb{R}^{n\times m}$ and $\mathcal{K}_K(\mathcal{Q}, R) = \text{span}\big\{R, \mathcal{Q}(R), \cdots, \mathcal{Q}^{K-1}(R)\big\}$ be the Krylov subspace and $\{U_i\}_{i=1}^K$ be orthogonal sequence with $U_i \in \mathbb{R}^{n\times m}$ for all $i = 1, \cdots, K$. The global generalized Hessenberg process satisfies the following orthogonality condition

$$\langle R_i, U_1\rangle_F = \langle R_i, U_2\rangle_F = \cdots \langle R_i, U_K\rangle_F = 0,$$

so as to build the basis $\{R_1, R_2, \cdots, R_K\}$ for $\mathcal{K}_K(\mathcal{Q}, R)$.

It is clear that if the previous steps of K iterations of Algorithm 2 are implemented without any occurrence of breakdown, then we can obtain

$$\begin{aligned}[\mathcal{Q}(R_1), \mathcal{Q}(R_2), \cdots, \mathcal{Q}(R_K)] &= \mathcal{R}_{K+1}(\widetilde{H}_K \otimes I_m) \\ &= \mathcal{R}_K(H_K \otimes I_m) + h_{K+1,K}R_{K+1}\big((e^K)^T \otimes I_m\big),\end{aligned} \tag{3.1}$$

where

$$\widetilde{H}_K := \begin{pmatrix} H_K \\ h_{K+1,K}(e^K)^T \end{pmatrix} \in \mathbb{R}^{(K+1)\times K}, \tag{3.2}$$

Algorithm 2 Global generalized Hessenberg process for linear matrix equation

Input: matrix $R \in \mathbb{R}^{n\times m}$ and the dimension of the Krylov subspace K.
Set $\beta = \langle R, U_1\rangle_F$, $R_1 = R/\beta$;
for $j = 1, \cdots, K$ **do**
  $W = \mathcal{Q}(R_j)$;
  for $i = 1, \cdots, j$ **do**
    $h_{ij} = \langle U_i, W\rangle_F / \langle U_i, R_i\rangle_F$;
    $W = W - h_{ij}R_i$;
  end for
  $h_{j+1,j} = \langle W, U_{j+1}\rangle_F$, **if** $h_{j+1,j} = 0$, then **stop**;
  $R_{j+1} = W/h_{j+1,j}$;
end for

$\mathcal{R}_{K+1} = [R_1, R_2, \cdots, R_{K+1}] \in \mathbb{R}^{n\times(Km+K)}$, $H_K \in \mathbb{R}^{K\times K}$ and $\widetilde{H}_K$ is a Hessenberg matrix with entries generated by Algorithm 2. e^K is the K-th column of identity matrix with size $K \times K$.

Now, we will exploit a Petrov-Galerkin condition and a minimal residual norm condition to state the global generalized Hessenberg process. Let X_0 be an initial guess to the continuous algebraic Riccati equations (1.1), $R_0 = \widetilde{C} - \mathcal{Q}(X_0)$ be the corresponding residual, where $\widetilde{C} = X_0BX_0 + C$. At the $k-$th iteration step of Algorithm 2. By the formula

$$X_K = X_0 + \mathcal{R}_K(u^K \otimes I_m),$$

where $\mathcal{R}_K = [R_1, R_2, \cdots, R_K] \in \mathbb{R}^{n\times(Km)}$, then $u^K \in \mathbb{R}^K$ can be obtained by the under Petrov-Galerkin orthogonality condition

$$\langle R_K, U_1\rangle_F = \langle R_K, U_2\rangle_F = \cdots \langle R_K, U_K\rangle_F = 0, \tag{3.3}$$

and

$$\begin{aligned} R_K &= \widetilde{C} - \mathcal{Q}(X_K) \\ &= \widetilde{C} - \mathcal{Q}(X_0 + \mathcal{R}_K(u^K \otimes I_m)) \\ &= R_0 - [\mathcal{Q}(R_1), \mathcal{Q}(R_2), \cdots, \mathcal{Q}(R_K)](u^K \otimes I_m) \\ &= R_0 - \mathcal{R}_{K+1}(\widetilde{H}_K \otimes I_m)(u^K \otimes I_m) \\ &= R_0 - \mathcal{R}_{K+1}(\widetilde{H}_K u^K \otimes I_m) \\ &= R_0 - \mathcal{R}_K(H_K u^K \otimes I_m) + h_{K+1,K}R_{K+1}\big((e^K)^T u^K \otimes I_m\big). \end{aligned}$$

Notice that the relation (3.1) was used in the above equality. Moreover,

$$R_K = \mathcal{R}_K\big((\beta e_1^K - H_K u^K) \otimes I_m\big) + h_{K+1,K}R_{K+1}\big((e^K)^T u^K \otimes I_m\big), \tag{3.4}$$

where β is chosen according to specified algorithm, such as $\beta = \|R_0\|_F$. $\mathcal{R}_K$ satisfies the block-based orthogonality of columns. By calculations, it give rises to

$$\|R_K\|_F^2 = \|\beta e_1^K - H_K u^K\|_2^2 + h_{K+1,K}^2 \sum_{i=1}^{K} (u_K^K)^2.$$

Hence, we can obtain the approximation solution of u^K by solving the system of $H_K u^K = \beta e_1^K$. If $h_{K+1,K} = 0$, then the solution is exact for original problem.

In addition, it follows from (3.4) that

$$R_K = \mathcal{R}_{K+1}\big((\beta e_1^{K+1} - \widetilde{H}_K u^K) \otimes I_m\big) \tag{3.5}$$

and

$$\|R_K\|_F = \|\beta e_1^{K+1} - \widetilde{H}_K u^K\|_2.$$

Therefore, u^K can be also obtained by solving the following least squares problem

$$u^K = \arg \min_{u^K \in \mathbb{R}^K} \|\beta e_1^{K+1} - \widetilde{H}_K u^K\|_2, \tag{3.6}$$

where $\widetilde{H}_K$ is from (3.2). It is worth mentioning that if we take $U_i = R_i$ ($i = 1, 2, \cdots, K$), then the global generalized Hessenberg process in Algorithm 2 will reduce to the particular case, i.e., global Arnoldi process shown in Algorithm 3. In fact, the global generalized Hessenberg process is possible to encounter the breakdown, so a pivoting strategy is necessary to this process. The result is given by Algorithm 4.

Algorithm 3 Arnoldi process for linear matrix equation

Input: matrix $R \in \mathbb{R}^{n\times m}$ and the dimension of the Krylov subspace K.
Set $\beta = \|R\|_F$, $R_1 = R/\beta$;
for $j = 1, \cdots, K$ **do**
  $W = \mathcal{Q}(R_j)$;
  for $i = 1, \cdots, j$ **do**
    $h_{ij} = \langle R_i, W\rangle_F$;
    $W = W - h_{ij}R_i$;
  end for
  $h_{j+1,j} = \|W\|_F$, **if** $h_{j+1,j} = 0$, then **stop**;
  $R_{j+1} = W/h_{j+1,j}$;
end for

Algorithm 4 Hessenberg process with pivoting strategy for linear matrix equation

Input: matrix $R \in \mathbb{R}^{n\times m}$ and the dimension of the Krylov subspace K.
Determine i_0, j_0 such that $|R_{i_0,j_0}| = \max\{|R_{i_0,j_0}|\}_{i=1,\cdots,n}^{j=1,\cdots,m}$;
Set $\beta = R_{i_0,j_0}$, $R_1 = R/\beta$, $p_{1,1} = i_0$, $p_{1,2} = j_0$;
for $j = 1, \cdots, K$ **do**
  $W = \mathcal{Q}(R_j)$;
  for $i = 1, \cdots, j$ **do**
    $h_{ij} = W_{p_{i,1},p_{i,2}}$;
    $W = W - h_{ij}R_i$;
  end for
  Determine i_0, j_0 such that $|W_{i_0,j_0}| = \max\{|W_{i_0,j_0}|\}_{i=1,\cdots,n}^{j=1,\cdots,m}$;
  Set $h_{j+1,j} = W_{i_0,j_0}$, $R_{j+1} = W/h_{j+1,j}$, $p_{j+1,1} = i_0$, $p_{j+1,2} = j_0$;
end for

As in the case of solving linear matrix systems, combining Algorithm 3 with Petrov-Galerkin orthogonality condition (3.3) yields global full orthogonalization method (simply denoted by FOM) [15], while combining Algorithm 3 with minimizing norm condition (3.6) defines GMRES method. On the other hand, combining Algorithm 4 with the Petrov-Galerkin orthogonality condition (3.3) generates global generalized Hessenberg method with pivoting strategy (simply denoted by GHESS), while combining Algorithm 4 with minimizing norm condition (3.6) defines global GMRH method. CMRH was firstly provided to solve nonsymmetric linear systems in [21], which shows that the method is less expensive and needs less storage than GMRES per iteration. It also needs to be restarted to avoid the increasing of number of matrices requiring storage as K.

In this paper, we focus on the GHESS and GMRH, bonded with Kleinman-Newton's method, to solve the continuous algebraic Riccati equation (CARE) (1.1). The descriptions of global GHESS method and CMRH method with restarted version are summarized in Algorithm 5.

Algorithm 5 GHESS/CMRH methods for solving continuous algebraic Riccati equation

Input: Given matrix $A,\ B,\ C \in \mathbb{R}^{n\times n}$, initial guess matrix $X_0 \in \mathbb{R}^{n\times n}$, such that $X_0 = (X_0)^T$and $BX_0 - A$ is stable. Let the dimension of the Krylov subspace be K and the tolerance error $\varepsilon > 0$.
for $k = 0, \cdots, \mathrm{Iter}_{\max}$ **do**
  Compute $R_0 = \widetilde{C} - \mathcal{Q}(X_0)$ where $\widetilde{C} = C - X_0 B X_0$. Set $\mathcal{A}_k = BX_k - A$;
  Compute $\widetilde{H}_K$ and $\mathcal{R}_K$ by applying Algorithm 4 to $\mathcal{K}_K(\mathcal{Q}, R_0)$;
  Solve the problem $\begin{cases} H_K u^K = \beta e_1^K; & \text{(GHESS method)} \\ u^K = \arg\min\limits_{u^K \in \mathbb{R}^K} \|\beta e_1^{K+1} - \widetilde{H}_K u^K\|_2; & \text{(CMRH method)} \end{cases}$
  Compute the approximate solution $X_K = X_0 + \mathcal{R}_K(u^K \otimes I_m)$;
  Compute $R_K = \widetilde{C} - \mathcal{Q}(X_K)$;
  if $\|R_K\|_F < \varepsilon$ **then**
    output the approximate solution X_K;
  else
    $R_0 = R_K,\ \ X_0 = X_K$;
  end if
end for

4. Extension

As generalization of matrix, high dimensional tensor, such as $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, also can be employed to verify adequately our methods. The tensor form of global generalized Hessenberg process apparently deserve discussion and further research. It is nature and feasible to realize the generalization and high dimensional applications from the linear matrix equation to linear tensor equation. Although it may be difficult to tackle the nonlinear tensor product and manage some relevant properties, however, we will firstly prepare to come up with the linear tensor equation with these proposed methods, such as GHESS (or CMRH) for Sylvester tensor equation or Lyapunov tensor equation in future.

It has been shown that the rate of convergence for any Krylov solvers greatly improves since the condition number of matrix decreases when the preconditioning is introduced properly. Assume that $M_k \in \mathbb{R}^{n\times n}$ is an appropriate preconditioner that could vary from one iteration to another iteration. Considering the preconditioners to the equation in the step 2 of Algorithm 1, we can describe as following

$$\mathcal{A}_k^T M_k^{-1} Y_k + M_k^{-1} Y_k \mathcal{A}_k = \widetilde{C} \tag{4.1}$$

or

$$M_k \mathcal{A}_k^T M_k^{-1} Y_k + Y_k \mathcal{A}_k = \mathcal{C}. \tag{4.2}$$

where $Y_k = M_k X_{k+1}$, $\widetilde{C} = C - X_k B X_k$, $\mathcal{C} = M_k \widetilde{C}$, $\mathcal{A}_k = BX_k - A$. Moreover, suppose that $\widehat{\mathcal{A}}_k = M_k \mathcal{A}_k^T M_k^{-1}$, and define the linear mapping $\widehat{\mathcal{Q}}$ as

$$\widehat{\mathcal{Q}}(Y) = \widehat{\mathcal{A}}_k^T Y + Y \mathcal{A}_k. \tag{4.3}$$

After preconditioned, (2.10) will be transformed into

$$\widehat{\mathcal{Q}}(Y) = \mathcal{C}, \tag{4.4}$$

which is a Sylvester matrix equation.

As a result, we give ultimately the following CMRH method with the variable preconditioners for solving continuous algebraic Riccati equation in Algorithm 6.

5. Numerical experiments

In this section, some numerical examples are discussed to validate the performance of effectiveness and advantages of the proposed Hessenberg based methods for solving the algebraic Riccati equation. We compare the convergence of the global generalized Hessenberg method (denoted as 'GHESS'), global CMRH method (still denoted as 'CMRH') and global generalized minimum residual method (denoted as 'GMRES') by the iteration step (denoted as 'IT'), elapsed CPU time in seconds (denoted as 'CPU'), and residual error (denoted as 'RES'). In actual computations, the running is terminated when the current iteration satisfies

$$\text{RES} := \frac{\|R_k\|}{\|R_0\|} < 10^{-10}$$

or if the number of iteration exceeds the prescribed iteration steps $k_{max} = 100$, where $R_k = \mathcal{Q}(X_k) - (X_k B X_k - C)$, $R^0 = \mathcal{Q}(X^0) - (X_0 B X_0 - C)$, X_k denotes the k-th step iteration in Algorithms, X_0 is initial guess. And if the elapsed CPU time is more than 1000 seconds or breakdown of iteration, we denote the situation as $'-'$.

All the numerical experiments have been carried out by MATLAB R2011b 7.1.3 on a PC equipped with an Intel(R) Core(TM) i7-2670QM, CPU running at 2.20GHZ with 8 GB of RAM in Windows 7 operating system.

Example 5.1. We first consider the algebraic Riccati equation (1.1) with the following form [13]:

$$A = \begin{pmatrix} 4 & 1-d & \cdots & 1 \\ 1-d & 4 & 1-d & \cdots \\ \ddots & \ddots & \ddots & \ddots \\ 1 & \cdots & 1-d & 4 \end{pmatrix}_{n\times n},$$

Algorithm 6 CMRH method with the variable preconditioner for solving continuous algebraic Riccati equation

Input: Given matrix $A,\ B,\ C \in \mathbb{R}^{n\times n}$, initial guess matrix $X_0 \in \mathbb{R}^{n\times n}$, such that $X_0 = (X_0)^T$ and $BX_0 - A$ is stable. Let the dimension of the Krylov subspace be K and the tolerance error $\varepsilon > 0$.
for $k = 0, \cdots, \mathrm{Iter}_{\max}$ **do**
  Compute $R_0 = \widetilde{C} - \mathcal{Q}(X_0)$ where $\widetilde{C} = C - X_0BX_0$. Set $\mathcal{A}_k = BX_k - A$;
  Determine i_0, j_0 such that $|(R_0)_{i_0,j_0}| = \max\{|(R_0)_{i_0,j_0}|\}_{i=1,\cdots,n}^{j=1,\cdots,m}$;
  Set $\beta = (R_0)_{i_0,j_0}$, $R_1 = R_0/\beta$, $p_{1,1} = i_0$, $p_{1,2} = j_0$;
for $j = 1, \cdots, K$ **do**
  $Y_j = M_jR_j$;
  $W = \widehat{\mathcal{Q}}(Y_j)$;
  for $i = 1, \cdots, j$ **do**
    $h_{ij} = W_{p_{i,1},p_{i,2}}$;
    $W = W - h_{ij}R_i$;
  end for
  Determine i_0, j_0 such that $|W_{i_0,j_0}| = \max\{|W_{i_0,j_0}|\}_{i=1,\cdots,n}^{j=1,\cdots,m}$;
  Set $h_{j+1,j} = W_{i_0,j_0}$, $R_{j+1} = W/h_{j+1,j}$, $p_{j+1,1} = i_0$, $p_{j+1,2} = j_0$;
  Denote $\widetilde{H}_K = [h_{i,j}] \in \mathbb{R}^{K+1,K}$, and $\mathcal{Y}_K = [Y_1, Y_2, \cdots, Y_K]$;
  Solve the problem

$$u^K = \arg\min_{u^K\in\mathbb{R}^K} \|\beta e_1^{K+1} - \widetilde{H}_Ku^K\|_2; \tag{4.5}$$

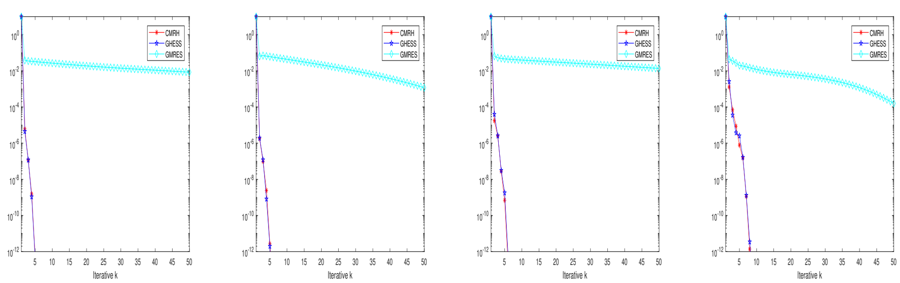
  Compute the approximate solution $X_K = X_0 + \mathcal{Y}_K(u^K \otimes I_m)$;
  Compute $R_K = \widetilde{C} - \widehat{\mathcal{Q}}(X_K)$;
  if $\|R_K\|_F < \varepsilon$ **then**
    output the approximate solution X_K;
  else
    $R_0 = R_K,\ \ X_0 = X_K$;
  end if
end for

where d (a constant in interval $[0, 1]$) and matrix B are given in Table 1. $C = \widetilde{C}^T\widetilde{C}$ is a low-rank random matrix, where $\widetilde{C}$ is an $n \times k$ random matrix whose entries are given in $[0, 1]$ with uniform distribution.

We give four different cases for different matrices A, B with various sizes. B is chosen by identity matrix or scalar matrix. Parameter d is selected as 0.5 or 0.8. Dimensions of matrices are set from 100 to 1000. The numerical results show that CMRH and GHESS are nearly similar efficiency, but both of them overmatch the GMRES, which can be seen from convergence performances in the Table 1. The fact further bears out the conclusions in [21], one of which clarifies a point that CMRH performs more accurately and reduces the residual norm compared with GMRES for linear systems. From the residual trend chart with the changing numbers of iteration in Figure 1, one can demonstrably find the desired performance of the proposed methods. In this example, it seems that CMRH method equally matches with GHESS method. To confirm this observation, we will give more examples below.

Table 1. Numerical results for Example 5.1.

Case	Size		CMRH	GHESS	GMRES
	$d = 0.8$	It	5	5	100
1	$B = 2I$	CPU	$6.5879e-02$	$4.4593e-02$	$1.8461e+00$
	$n = 100, k = 5$	RES	$2.5392e-13$	$1.8566e-12$	$7.3285e-04$
	$d = 0.5$	It	6	6	100
2	$B = I$	CPU	$9.3140e-02$	$4.5456e-02$	$1.9344e+00$
	$n = 100, k = 5$	RES	$5.8532e-16$	$5.1960e-16$	$1.4234e-05$
	$d = 0.8$	It	5	5	100
3	$B = 2I$	CPU	$2.630124e-01$	$1.456257e-01$	$8.626567e+00$
	$n = 200, k = 5$	RES	$2.0026e-13$	$6.5526e-13$	$2.7603e-03$
	$d = 0.5$	It	8	8	100
4	$B = I$	CPU	$1.4887e+01$	$1.4629e+01$	$1.0355e+02$
	$n = 1000, k = 10$	RES	$2.7925e-16$	$2.7850e-16$	$6.2269e-09$



(a) case1 (b) case2 (c) case3 (d) case4

Figure 1. The relative residual for CMRH, GHESS and GMRES mehtods with difference cases in Table 1.

Example 5.2. We consider the algebraic Riccati equation (1.1) with the test matrix matrix A is obtained by discretizing the following operators [3, 14]

$$L_A(u) := \triangle u - f_1(x, y)\frac{\partial u}{\partial x} - f_2(x, y)\frac{\partial u}{\partial y} - f_3(x, y)u$$

on the unit square $\Omega = [0, 1]\times[0, 1]$ with homogeneous Dirichlet boundary conditions where $f_1(x, y)$, $f_2(x, y)$, $f_3(x, y)$, and matrix B are given in Table 2. $C = \widetilde{C}^T\widetilde{C}$ is also a low-rank random matrix, where $\widetilde{C}$ is an $n \times k$ random matrix whose entries are given in $[0, 1]$ with uniform distribution. The matrix A is generated by the function of 'fdm-2d-matrix' from the LYAPACK toolbox [20].

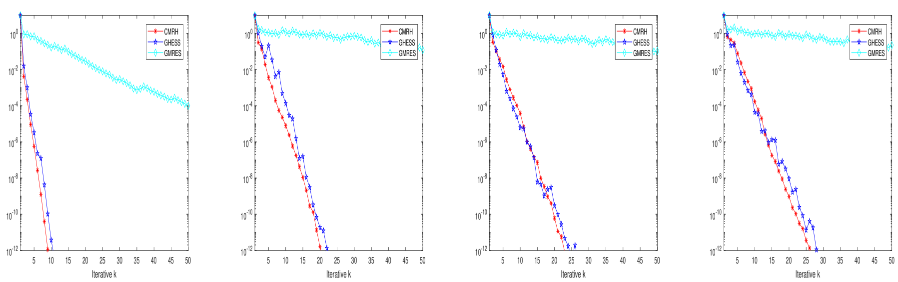
In this example, the selection of matrix A depends on functions $f_1(x, y)$, $f_2(x, y)$, $f_3(x, y)$, and the matrix B is just chosen with two cases, i.e., identity matrix and $\frac{\widetilde{B}^T\widetilde{B}}{\|\widetilde{B}\|_F}$, where $\widetilde{B} = rand(n, k)$. Three functions $f_1(x, y)$, $f_2(x, y)$, $f_3(x, y)$ are set with four cases in Table 2. All the numerical results are depicted in Table 3 and Figure 2. From the elapsed CPU and numbers of iteration, the CMRH is more stable slightly than the GHESS method in most cases, which distinctly outperform the GMRES method from all aspects.

Table 2. Different cases for Example 5.2 with $\widetilde{B} = rand(n, k)$.

Case	1	2	3	4
B	$B = I$	B=$\frac{\widetilde{B}^T\widetilde{B}}{\|\widetilde{B}\|_F}$	$B = I$	B=$\frac{\widetilde{B}^T\widetilde{B}}{\|\widetilde{B}\|_F}$
$f_1(x, y)$	e^{x^2+y}	$cos(xy)$	$sin(xy)$	$2xy$
$f_2(x, y)$	$sin(x^2 + 2y)$	e^{x^2y}	e^{x^2y}	e^{xy}
$f_3(x, y)$	$cos(xy)$	$x - y$	$x + y^2$	xy

Table 3. Numerical results for Example 5.2.

Case	Size		CMRH	GHESS	GMRES
	$n_0 = 9$	It	9	10	100
1	$n = 81$	CPU	$2.4519e-01$	$9.3002e-02$	$1.3383e+00$
	$k = 5$	RES	$9.4482e-16$	$1.0060e-15$	$1.2663e-08$
	$n_0 = 20$	It	20	22	100
2	$n = 400$	CPU	$4.8016e+00$	$4.5642e+00$	$5.7404e+01$
	$k = 5$	RES	$7.6813e-15$	$2.2402e-14$	$6.5506e-03$
	$n_0 = 25$	It	23	26	100
3	$n = 625$	CPU	$1.6791e+01$	$1.7032e+01$	$2.0781e+02$
	$k = 5$	RES	$1.0745e-14$	$1.5410e-14$	$2.0127e-02$
	$n_0 = 30$	It	26	28	100
4	$n = 900$	CPU	$6.0244e+01$	$4.5223e+01$	$3.0686e+02$
	$k = 10$	RES	$1.3342e-14$	$1.3471e-14$	$2.2768e-02$



(a) case1 (b) case2 (c) case3 (d) case4

Figure 2. The relative residual for CMRH, GHESS and GMRES mehtods with difference cases in Table 3.

Example 5.3. The test matrices A for this example are bcsstk02, bcsstk06, bcsstk15, bcsstk16 which stem from the Matrix Market, where bcsstk15, bcsstk16 is the data from module of an offshore platform and U.S. Army Corps of Engineers dam, respectively. For more detail, see [18].

The parameter conditions of matrix A of four data set introduced in this example are shown in the Table 4. The dimensions of matrices form small size 66×66 to large size 4884×4884. The condition number and Frobenius norm and number of nonzero imply the slight sparsity for matrix A. These test problems all can be obtained from Website http://math.nist.gov/MatrixMarket/. From Table 5, we find that, for bcsstk02 and bcsstk06, GMRES can work well for solving the continuous algebraic Riccati equation, however for test problems bcsstk15 and bcsstk16, GMRES don't implement successfully due to exceed the maximum CPU limit. For CMRH and GHESS methods, the convergence performance also illustrate expected result, i.e., the efficiency and feasibility for solving the continuous algebraic Riccati equation.

Table 4. Different parameters of matrix A for Example 5.3.

Test Problem	Size	Frobenius norm	Nonzero	Condition number
bcsstk02	66×66	$5.3e+04$	2211	$1.3e+04$
bcsstk06	420×420	$2.1e+10$	4140	$1.2e+07$
bcsstk15	3948×3948	$5.7e+10$	60882	$8.0e+09$
bcsstk16	4884×4884	$6.0e+10$	147631	$65e+09$

Table 5. Numerical results for Example 5.2.

Test Problem		CMRH	GHESS	GMRES
	It	2	2	100
bcsstk02	CPU	$2.9521e-02$	$1.3012e-2$	$9.1087e-01$
	RES	$3.5986e-11$	$5.8162e-11$	$4.6891e-04$
	It	2	2	100
bcsstk06	CPU	$2.5120e-01$	$2.4731e-01$	$8.2353e+01$
	RES	$3.6600e-14$	$2.3812e-14$	$9.4524e-04$
	It	2	2	–
bcsstk15	CPU	$4.2106e+01$	$4.2317e+01$	–
	RES	$1.5174e-10$	$1.4356e-10$	–
	It	2	2	–
bcsstk16	CPU	$1.2274e+02$	$1.3277e+02$	–
	RES	$1.7008e-14$	$2.5103e-14$	–

6. Conclusion

In this paper, Krylov subspace methods of Hessenberg based, i.e., CMRH and GHESS methods, are investigated for solving the continuous algebraic Riccati equation. In view of this, these approaches are quite efficient not only for linear matrix equations but also for nonlinear matrix equations. The proposed approaches have been demonstrated to be superior to the classical global GMRES, which can be fully validated in our numerical experiments section. Meanwhile, we point out some ideas for our future research work about the extension in linear tensor systems and its preconditioning method.

References

[1] L. Amodei and J. M. Buchot, *An invariant subspace method for large-scale algebraic Riccati equation*, Appl. Numer. Math., 2010, 60, 1067–1082.

[2] A. C. Antoulas, *Approximation of large-scale dynamical systems*, SIAM, Philadelphia, 2005.

[3] S. Agoujil, A. H. Bentbib, K. Jbilou, and E. M. Sadek, *A minimal residual norm method for large-scale Sylvester matrix equations*, Elect. Trans. Numer. Anal., 2014, 43, 45–59.

[4] L. Bao, Y. Lin and Y. Wei, *A modified simple iterative method for nonsymmetric*

algebraic Riccati equations arising in transport theory, Appl. Math. Comput., 2006, 181(2), 1499–1504.

[5] P. Benner, J. Li and T. Penzl, *Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems*, Numer. Linear Algebra. Appl., 2008, 15, 755–777.

[6] J. P. Chehab and M. Raydan, *Inexact Newton's method with inner implicit preconditioning for algebraic Riccati equations*, Comp. Appl. Math., 2017, 36, 955–969. DOI 10.1007/s40314–015–0274–8.

[7] E. Chu, H. Fan, W. Lin and C. Wang, *Structure-preserving doubling algorithms for periodic discrete-time algebraic Riccati equations*, Int. J. Control, 2004, 77, 767–788.

[8] D. Chu, W. Lin and R. Tan, *A Numerical Method for a Generalized Algebraic Riccati Equation*, SIAM J. Control Optim., 2006, 45(4), 1222–1250.

[9] B. N. Datta, *Numeritcal methods for linear control systems design and analysis*, Elservier Press, Amsterdam, 2003.

[10] R. Freund, G. H. Golub and N. M. Nachtigal, *Iterative solution of linear systems*, Acta Numer., 1992, 1, 57–100.

[11] X. Guo, W. Lin and S. Xu, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation*, Numer. Math., 2006, 103, 393–412.

[12] M. Heyouni and H. Sadok, *A new implementation of the CMRH method for solving dense linear systems*, J. Comput. Appl. Math., 2008, 213(2), 387–399.

[13] K. Jbilou, *Block Krylov subspace methods for large algebraic Riccati equations*, Numer. Algorithms, 2003, 34, 339–353.

[14] K. Jbilou, *Low rank approximate solutions to large Sylvester matrix equations*, Appl. Math. Comput., 2006, 177(1), 365–376.

[15] K. Jbilou, A. Messaoudi, and H. Sadok, *Global FOM and GMRES algorithms for matrix equations*, Appl. Numer. Math., 1999, 31(1), 49–63.

[16] J. Juang, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 1995, 230, 89–100.

[17] L. Lu, *Solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory*, SIAM J. Matrix Anal. Appl. 2005, 26, 679–685.

[18] Matrix Market. http://math.nist.gov/MatrixMarket/.

[19] V. Mehrmann, *The autonomous linear quadratic control problem, theory and numerical solution. Lecture Notes in Control and Information Sciences*, Springer, Berlin, 1991, 163.

[20] T. Penzl, *LYAPACK: A MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems, users guide*, 2000, 1(0). Available at: www.tu-chemnitz.de/sfb393/lyapack.

[21] H. Sadok, *CMRH: A new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm*, Numer. Algorithm, 1999, 20, 303–321.

[22] H. Sadok and D. B. Szyld, *A new look at CMRH and its relation to GMRES*, BIT Numer. Math., 2012, 52(2), 485–501.

[23] N. Sandell, *On Newton's method for Riccati equation solution*, IEEE Trans. Auto. Control, 1974, 19, 254–255.

[24] V. Simoncini, *On two numerical methods for the solution of large-scale algebraic Riccati equations*, IMA J. Numer. Anal., 2013, 1–17.

[25] K. Zhang and C. Gu, *Flexible global generalized Hessenberg methods for linear systems with multiple right-hand sides*, J. Comput. Appl. Math, 2014, 263, 312–325.