



Research article

# Double iterative algorithm for solving different constrained solutions of multivariate quadratic matrix equations

Shousheng Zhu\*

Department of science, Shandong Jianzhu University, Jinan 250101, China

\* **Correspondence:** Email: zhushousheng18@sdjzu.edu.cn.

**Abstract:** Constrained solutions are required in some practical problems. This paper studies the problem of different constrained solutions for a class of multivariate quadratic matrix equations, which has rarely been studied before. First, the quadratic matrix equations are transformed into linear matrix equations by Newton’s method. Second, the different constrained solutions of the linear matrix equations are solved by the modified conjugate gradient method, and then the different constrained solutions of the multivariate quadratic matrix equations are obtained. Finally, the convergence of the algorithm is proved. The algorithm can be well performed on the computers, and the effectiveness of the method is verified by numerical examples.

**Keywords:** multivariate quadratic matrix equations; Newton’s method; modified conjugate gradient method; different constrained solutions

**Mathematics Subject Classification:** 15A24, 65F45, 65H10

## 1. Introduction

Quadratic matrix equation has many different forms, such as  $AX^2+BX+C = O$  arising in quasi-birth-death processes [1, 2] and Riccati equation  $XCX - XE - AX + B = O$  arising in transport theory [3–6]. There are also some coupled quadratic matrix equations with two or three variables [7, 8]. This article will study the general form of these equations. As can be seen, the linear part of these equations can be expressed in the form  $\sum_{i=1}^3 C_i^{(l)} X_i D_i^{(l)}$ , and the quadratic part of them can be expressed as  $\sum_{i,j=1}^3 X_i E_{ij}^{(l)} X_j$ . Therefore, we study the following coupled quadratic matrix equation:

$$\sum_{i=1}^3 C_i^{(l)} X_i D_i^{(l)} + \sum_{i,j=1}^3 X_i E_{ij}^{(l)} X_j = S^{(l)} \quad (l = 1, 2), \tag{1.1}$$

where all matrices are  $n \times n$  real matrices. Each equation in (1.1) consists of three linear terms and nine quadratic terms. Besides, Eq (1.1) have three variables and only two equations, so the solution is not unique.

As we know, we always need some special kind of solutions in practical applications, such as symmetric solutions are widely used in control theory [8, 9] and reflexive solutions which are also called generalized centro-symmetric solutions are used in information theory, linear estimate theory and numerical analysis [10, 11]. Liao, Liu, etc. have studied the problem of different constrained solutions of linear matrix equations [12, 13]. In this article, we will design a method to obtain different constrained solutions of a class of quadratic matrix equations.

Many researchers have studied quadratic matrix equations. For example, Bini, Iannazzo and Poloni gave a fast Newton's method for a quadratic matrix equation [4]. Long, Hu and Zhang used an improved Newton's method to solve a quadratic matrix equation [14]. Convergence rate of some iterative methods for quadratic matrix equations arising in transport theory was also described by Guo and Lin [15]. Zhang, Zhu and Liu have studied the constrained solutions of two-variable Riccati matrix equations based on Newton's method and modified conjugate gradient (MCG) method [16, 17]. This article will further study the problem of different constrained solutions of coupled quadratic matrix equations with three matrix variables. The algorithm designed in the paper is superior in computing different constrained solutions.

Notation:  $R^{n \times n}$  denotes the set of  $n \times n$  real matrices. The symbols  $A^T$  and  $\text{tr}(A)$  represent the transpose and the trace of the matrix  $A$  respectively.  $A \otimes B$  stands for the Kronecker product of matrices  $A$  and  $B$ ,  $\overline{\text{vec}}(\cdot)$  is an operator that transforms a matrix  $A$  into a column vector by vertically stacking the columns of the matrix  $A^T$ . For example, for the  $2 \times 2$  matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

the vectorization is  $\overline{\text{vec}}(A) = [a, b, c, d]^T$ . We define an inner product of two matrices  $A, B \in R^{n \times n}$  as  $[A, B] = \text{tr}(A^T B)$ , then the norm of a matrix  $A$  generated by this inner product is Frobenius norm and denoted by  $\|A\|$ , i.e.  $\|A\| = \sqrt{[A, A]}$ .

Let  $\Omega_1$  be the set of symmetric matrices.  $P_1, P_2 \in R^{n \times n}$  are said to be symmetric orthogonal matrices if  $P_i = P_i^T$  and  $P_i^2 = I$  ( $i = 1, 2$ ).  $X \in R^{n \times n}$  is said to be a reflexive matrix with respect to  $P_1$  if  $P_1 X P_1 = X$ . Let  $\Omega_5$  be the set of reflexive matrices.  $X \in R^{n \times n}$  is said to be a symmetric reflexive matrix with respect to  $P_2$  if  $X^T = X = P_2 X P_2$ . Let  $\Omega_9$  be the set of symmetric reflexive matrices. We call  $(X_1, X_2, X_3)$  a constrained matrix in  $\Omega_{1-5-9}$  when  $X_1 \in \Omega_1$ ,  $X_2 \in \Omega_5$  and  $X_3 \in \Omega_9$ . Besides, if the symmetric orthogonal matrices  $P_1$  and  $P_2$  are changed, we will get different constrained matrices in  $\Omega_{1-5-9}$ .

The paper is organized as follows: First, we use Newton's method to convert the quadratic matrix equations into linear matrix equations. Second, MCG method [10, 13, 16, 18] is applied to solve the derived linear matrix equations. Finally, numerical examples are presented to support the theoretical results of this paper.

## 2. Newton's method for solving quadratic matrix equations

As a matter of convenience, we first introduce some notations.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}, \quad X^{(k)} = \begin{bmatrix} X_1^{(k)} \\ X_2^{(k)} \\ X_3^{(k)} \end{bmatrix}, \quad X^* = \begin{bmatrix} X_1^* \\ X_2^* \\ X_3^* \end{bmatrix}.$$

$Y$ ,  $Y^{(k)}$  and  $Y^*$  are defined in the same way as  $X$ ,  $X^{(k)}$  and  $X^*$  respectively. Then let

$$\begin{aligned}\psi^{(l)}(X) &= \sum_{i=1}^3 C_i^{(l)} X_i D_i^{(l)} + \sum_{i,j=1}^3 X_i E_{ij}^{(l)} X_j - S^{(l)}, \\ \phi_X^{(l)}(Y) &= \sum_{i=1}^3 C_i^{(l)} Y_i D_i^{(l)} + \sum_{i,j=1}^3 X_i E_{ij}^{(l)} Y_j + \sum_{i,j=1}^3 Y_i E_{ij}^{(l)} X_j,\end{aligned}$$

we can obtain

$$\psi^{(l)}(X + Y) = \psi^{(l)}(X) + \phi_X^{(l)}(Y) + \sum_{i,j=1}^3 Y_i E_{ij}^{(l)} Y_j \quad (l = 1, 2), \quad (2.1)$$

where  $\phi_X^{(l)}(Y): \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$  is the Fréchet derivative of  $\psi^{(l)}(X)$  at  $X$  in the direction  $Y$  [1].

**Lemma 2.1.** Finding the solution  $(X_1^*, X_2^*, X_3^*) \in \Omega_{1-5-9}$  of (1.1) can be transformed into finding the corrected value  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$  of  $\psi^{(l)}(X + Y) = 0$  ( $l = 1, 2$ ). We linearize and solve, to find  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$  from the coupled linear matrix equation

$$\phi_X^{(l)}(Y) = -\psi^{(l)}(X) \quad (l = 1, 2). \quad (2.2)$$

*Proof.* Supposing that the approximate solution  $(X_1, X_2, X_3) \in \Omega_{1-5-9}$  of Eq (1.1) has been obtained. Let  $X_i^* = X_i + Y_i$  ( $i = 1, 2, 3$ ), then finding  $(X_1^*, X_2^*, X_3^*) \in \Omega_{1-5-9}$  of (1.1) is transformed into finding the corrected value  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$  from

$$\psi^{(l)}(X + Y) = O \quad (l = 1, 2). \quad (2.3)$$

The Eq (2.3) is quadratic equations about  $Y_i$ . As is known, when the norm of  $Y_i$  is small enough, the quadratic parts  $\sum_{i,j=1}^3 Y_i E_{ij}^{(l)} Y_j$  about  $Y_i$  in (2.1) can be discarded according to Newton's method. In this way, we can get a linear approximation

$$\psi^{(l)}(X + Y) \approx \psi^{(l)}(X) + \phi_X^{(l)}(Y).$$

Therefore, finding the solution  $(X_1^*, X_2^*, X_3^*) \in \Omega_{1-5-9}$  of (1.1) is transformed into finding  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$  from  $\psi^{(l)}(X) + \phi_X^{(l)}(Y) = O$  ( $l = 1, 2$ ), that is, to solve (2.2).  $\square$

According to [14], Newton's method (algorithm 1) is introduced to find constrained solutions in  $\Omega_{1-5-9}$  of (1.1). Let

$$\psi(X) = \begin{bmatrix} \psi^{(1)}(X) \\ \psi^{(2)}(X) \end{bmatrix}, \quad \phi_X(Y) = \begin{bmatrix} \phi_X^{(1)}(Y) \\ \phi_X^{(2)}(Y) \end{bmatrix}.$$

---

**Algorithm 1 :** Newton's method solves the solution  $X$  of Eq (1.1)

---

Step 1. Choose an initial matrix  $(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}) \in \Omega_{1-5-9}$  and set  $k := 1$ .

Step 2. If  $\psi(X^{(k)}) = O$ , stop, else, solve for  $(Y_1^{(k)}, Y_2^{(k)}, Y_3^{(k)}) \in \Omega_{1-5-9}$  from

$$\phi_{X^{(k)}}(Y^{(k)}) = -\psi(X^{(k)}). \quad (2.4)$$

When (2.4) hasn't constrained solutions in  $\Omega_{1-5-9}$ , solve for  $(Y_1^{(k)}, Y_2^{(k)}, Y_3^{(k)}) \in \Omega_{1-5-9}$ , such that

$$\|\phi_{X^{(k)}}(Y^{(k)}) + \psi(X^{(k)})\| = \min. \quad (2.5)$$

Step 3. Compute  $X^{(k+1)} = X^{(k)} + Y^{(k)}$ , set  $k := k + 1$  and go to step 2.

---

The convergent properties about Newton's method can be obtained as follows according to [14] (The proof is similar to Lemma 2.1 in [14]).

**Theorem 2.1.** *Assume that the real matrix  $X^*$  is a simple root of (1.1), i.e.  $\psi(X^*) = O$  and  $\phi_{X^*}(Y)$  is regular. Then if the starting matrix  $X^{(1)}$  is chosen sufficiently close to the solution  $X^*$ , the sequence  $\{X^{(k)}\}$  generated by Newton's method converges quadratically to the solution  $X^*$ .*

### 3. MCG method for solving linear matrix equations

In algorithm 1, when  $X^{(k)}$  is known, then  $Y^{(k)}$  needs to be solved. In this section, MCG method will be used to solve  $Y^{(k)}$  from Eq (2.2), that is, to solve Eq (2.4) or Eq (2.5). Consider the general form of Eq (2.2)

$$\sum_{i=1}^3 \sum_{j=1}^7 A_{ij}^{(l)} Y_i B_{ij}^{(l)} = F^{(l)} \quad (l = 1, 2), \quad (3.1)$$

where all matrices in Eq (3.1) are  $n \times n$  real matrices. Let

$$h(Y) = \begin{bmatrix} h^{(1)}(Y) \\ h^{(2)}(Y) \end{bmatrix}, \quad F = \begin{bmatrix} F^{(1)} \\ F^{(2)} \end{bmatrix}, \quad R = F - h(Y) \stackrel{\text{def}}{=} \begin{bmatrix} R^{(1)} \\ R^{(2)} \end{bmatrix},$$

$$p(R) = \begin{bmatrix} p_1(R) \\ p_2(R) \\ p_3(R) \end{bmatrix}, \quad q(Y) = \begin{bmatrix} q_1(Y) \\ q_2(Y) \\ q_3(Y) \end{bmatrix},$$

where

$$h^{(l)}(Y) \stackrel{\text{def}}{=} h^{(l)}(Y_1, Y_2, Y_3) = \sum_{i=1}^3 \sum_{j=1}^7 A_{ij}^{(l)} Y_i B_{ij}^{(l)} \quad (l = 1, 2),$$

$$p_i(R) = \sum_{l=1}^2 \sum_{j=1}^7 (A_{ij}^{(l)})^T R^{(l)} (B_{ij}^{(l)})^T \quad (i = 1, 2, 3),$$

$$q_1(Y) = \frac{1}{2}(Y_1 + Y_1^T), \quad q_2(Y) = \frac{1}{2}(Y_2 + P_1 Y_2 P_1),$$

$$q_3(Y) = \frac{1}{4}(Y_3 + Y_3^T + P_2(Y_3 + Y_3^T)P_2),$$

and  $P_1, P_2 \in R^{n \times n}$  are symmetric orthogonal matrices.

In order to solve Eq (3.1), the following two questions will be considered.

**Problem 3.1.** *Assume that (3.1) has constrained solutions, find  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$  from (3.1).*

**Problem 3.2.** *Assume that (3.1) hasn't constrained solutions, find  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$ , such that*

$$\|h(Y) - F\| = \min. \quad (3.2)$$

#### 3.1. MCG method for solving problem 1

Based on the MCG method, we establish the following algorithm (algorithm 2) to solve problem 3.1.

From algorithm 2, we can easily see  $(Y_1^{(k)}, Y_2^{(k)}, Y_3^{(k)}) \in \Omega_{1-5-9}$  for  $k = 1, 2, \dots$  and have the following convergent properties (The proof is similar to Theorem 2.1 in [10]):

**Theorem 3.1.** *Assume that Eq (3.1) has constrained solutions in  $\Omega_{1-5-9}$ . Then for an arbitrary initial matrix  $(Y_1^{(1)}, Y_2^{(1)}, Y_3^{(1)}) \in \Omega_{1-5-9}$ , a solution of problem 3.1 can be obtained by algorithm 2 within finite number of iterations, which is also a constrained solution in  $\Omega_{1-5-9}$  of (3.1).*

**Algorithm 2** : MCG method to solve problem 3.1

Step 1. Choose an arbitrary initial matrix  $(Y_1^{(1)}, Y_2^{(1)}, Y_3^{(1)}) \in \Omega_{1-5-9}$ , set  $k := 1$  and compute

$$R_k = F - h(Y^{(k)}) \stackrel{\text{def}}{=} \begin{bmatrix} R_k^{(1)} \\ R_k^{(2)} \end{bmatrix}, \quad \tilde{R}_k = p(R_k) \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{R}_k^{(1)} \\ \tilde{R}_k^{(2)} \\ \tilde{R}_k^{(3)} \end{bmatrix}, \quad Z_k = q(\tilde{R}_k).$$

Step 2. If  $R_k = O$ , or  $R_k \neq O$  and  $Z_k = O$ , stop, else, compute

$$\alpha_k = \frac{\|R_k\|^2}{\|Z_k\|^2}, \quad Y^{(k+1)} = Y^{(k)} + \alpha_k Z_k.$$

Step 3. Compute

$$R_{k+1} = F - h(Y^{(k+1)}) \stackrel{\text{def}}{=} \begin{bmatrix} R_{k+1}^{(1)} \\ R_{k+1}^{(2)} \end{bmatrix}, \quad \tilde{R}_{k+1} = p(R_{k+1}) \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{R}_{k+1}^{(1)} \\ \tilde{R}_{k+1}^{(2)} \\ \tilde{R}_{k+1}^{(3)} \end{bmatrix},$$

$$\beta_{k+1} = \frac{\|R_{k+1}\|^2}{\|R_k\|^2}, \quad Z_{k+1} = q(\tilde{R}_{k+1}) + \beta_{k+1} Z_k.$$

Step 4. Set  $k := k + 1$  and go to step 2.

**3.2. MCG method for solving problem 2**

Algorithm 2 will break if  $R_i \neq O$  and  $Z_i = O$ , which means that Eq (3.1) hasn't constrained solution in  $\Omega_{1-5-9}$  according to Theorem 3.1. Therefore, we need to solve problem 3.2, that is, to find constrained least-squares solutions of (3.1).

We replace the problem of finding least-squares solutions in  $\Omega_{1-5-9}$  of (3.1) with finding solutions in  $\Omega_{1-5-9}$  of equivalent linear matrix equations by the Theorem 3.2, and then an iterative algorithm to find constrained least-squares solutions in  $\Omega_{1-5-9}$  of (3.1) is constructed according to algorithm 2.

As a matter of convenience, we introduce some notations:

$$u(Y) = h(Y_1, Y_2, \frac{1}{2}(Y_3 + P_2 Y_3 P_2)) \stackrel{\text{def}}{=} \begin{bmatrix} u^{(1)}(Y) \\ u^{(2)}(Y) \end{bmatrix},$$

$$v(Y) = h(Y_1^T, P_1 Y_2 P_1, \frac{1}{2}(Y_3^T + P_2 Y_3^T P_2)) \stackrel{\text{def}}{=} \begin{bmatrix} v^{(1)}(Y) \\ v^{(2)}(Y) \end{bmatrix},$$

$$g(Y) = \begin{bmatrix} g_1(Y) \\ g_2(Y) \\ g_3(Y) \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1(F) \\ Q_2(F) \\ Q_3(F) \end{bmatrix},$$

where

$$g_1(Y) = p_1(u) + p_1^T(v), \quad g_2(Y) = p_2(u) + P_1 p_2(v) P_1,$$

$$g_3(Y) = \frac{1}{2} (p_3(u) + p_3^T(v) + P_2 (p_3(u) + p_3^T(v)) P_2),$$

$$Q_1(F) = p_1(F) + p_1^T(F), \quad Q_2(F) = p_2(F) + P_1 p_2(F) P_1,$$

$$Q_3(F) = \frac{1}{2} (p_3(F) + p_3^T(F) + P_2 (p_3(F) + p_3^T(F)) P_2),$$

$u$  and  $v$  are functions of  $Y$ . Then, according to [16, 17], we have the following theorem.

**Theorem 3.2.** *Iterative algorithm for solving problem 3.2 can be replaced by finding constrained solutions in  $\Omega_{1-5-9}$  from*

$$g(Y) = Q. \quad (3.3)$$

Indeed, (3.3) has constrained solutions in  $\Omega_{1-5-9}$ .

*Proof.* When  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$ , we have  $Y_1 = Y_1^T$ ,  $Y_2 = P_1 Y_2 P_1$  and  $Y_3 = Y_3^T = P_2 Y_3 P_2$ . Therefore, solving problem 3.2 is equivalent to solving  $(Y_1, Y_2, Y_3) \in \Omega_{1-5-9}$  from

$$\left\| \begin{bmatrix} u(Y) \\ v(Y) \end{bmatrix} - \begin{bmatrix} F \\ F \end{bmatrix} \right\| = \min. \quad (3.4)$$

Now we have to prove that solving the problem (3.4) is equivalent to finding constrained solutions in  $\Omega_{1-5-9}$  of (3.3). We let multiply operation prior to Kronecker product operation between matrices. Let

$$G_i^{(l)} = \sum_{j=1}^7 A_{ij}^{(l)} \otimes (B_{ij}^{(l)})^T,$$

$$H_{im}^{(l)} = \sum_{j=1}^7 A_{ij}^{(l)} P_m \otimes (P_m B_{ij}^{(l)})^T, \quad (i = 1, 2, 3; l, m = 1, 2),$$

$$M = \begin{bmatrix} G_1^{(1)} & G_2^{(1)} & \frac{1}{2}(G_3^{(1)} + H_{32}^{(1)}) \\ G_1^{(2)} & G_2^{(2)} & \frac{1}{2}(G_3^{(2)} + H_{32}^{(2)}) \\ G_1^{(1)} T_{n,n} & H_{21}^{(1)} & \frac{1}{2}(G_3^{(1)} + H_{32}^{(1)}) T_{n,n} \\ G_1^{(2)} T_{n,n} & H_{21}^{(2)} & \frac{1}{2}(G_3^{(2)} + H_{32}^{(2)}) T_{n,n} \end{bmatrix},$$

$$y = \begin{bmatrix} \overline{\text{vec}}(Y_1) \\ \overline{\text{vec}}(Y_2) \\ \overline{\text{vec}}(Y_3) \end{bmatrix}, \quad f = \begin{bmatrix} \overline{\text{vec}}(F) \\ \overline{\text{vec}}(F) \end{bmatrix},$$

where  $T_{n,n}$  denotes a commutation matrix such that  $T_{n,n} \overline{\text{vec}}(A_{n \times n}) = \overline{\text{vec}}(A^T)$  [19] and let  $T_{n,n}$  only work on  $\overline{\text{vec}}$ . Then applying  $\overline{\text{vec}}$  to the following equations:

$$\begin{cases} u(Y) = F, \\ v(Y) = F, \end{cases} \quad (3.5)$$

we can get the equivalent equation:  $My = f$ . Besides,  $M^T My = M^T f$ , the normal equation of  $My = f$ , is the vectorization of (3.3). Therefore, the least-squares solution of  $My = f$  is also a solution of  $M^T My = M^T f$ , and the vectorization of the solution of (3.3). So the solution of (3.4) is also a solution of (3.3), and vice versa.

Above all, iterative algorithm for solving problem 3.2 can be replaced by finding constrained solutions in  $\Omega_{1-5-9}$  of (3.3).

As we all know, normal equations always have solutions, and the vectorization of Eq (3.3) is a normal equation, so Eq (3.3) also has solutions. Suppose  $\tilde{Y} = (\tilde{Y}_1^T, \tilde{Y}_2^T, \tilde{Y}_3^T)^T$  (whether  $\tilde{Y} \in \Omega_{1-5-9}$  or

not) is a solution of (3.3), then  $g(\tilde{Y}) = Q$ . Let  $Y_i^* = q_i(\tilde{Y})$  ( $i = 1, 2, 3$ ), then  $(Y_1^*, Y_2^*, Y_3^*) \in \Omega_{1-5-9}$  and  $g(Y^*) = Q$ . Hence, (3.3) has constrained solutions in  $\Omega_{1-5-9}$ .  $\square$

We use the MCG method to find constrained solutions in  $\Omega_{1-5-9}$  of (3.3) by algorithm 3, which is also a process to solve the problem 3.2.

---

**Algorithm 3** : MCG method to solve problem 3.2

---

Step 1. Choose an arbitrary initial matrix  $(Y_1^{(1)}, Y_2^{(1)}, Y_3^{(1)}) \in \Omega_{1-5-9}$ , set  $k := 1$  and compute

$$R_k = Q - g(Y^{(k)}), \quad \tilde{R}_k = g(R_k), \quad Z_k = \tilde{R}_k.$$

Step 2. If  $R_k = O$ , stop, else, compute

$$\alpha_k = \frac{\|R_k\|^2}{\|Z_k\|^2}, \quad Y^{(k+1)} = Y^{(k)} + \alpha_k Z_k.$$

Step 3. Compute

$$R_{k+1} = Q - g(Y^{(k+1)}), \quad \tilde{R}_{k+1} = g(R_{k+1}),$$

$$\beta_{k+1} = \frac{\|R_{k+1}\|^2}{\|R_k\|^2}, \quad Z_{k+1} = \tilde{R}_{k+1} + \beta_{k+1} Z_k.$$

Step 4. Set  $k := k + 1$  and go to step 2.

---

From algorithm 3, we can see that  $(Y_1^{(k)}, Y_2^{(k)}, Y_3^{(k)}) \in \Omega_{1-5-9}$  for  $k = 1, 2, \dots$  and have the following convergent properties (The proof is similar to Theorem 2 in [13]).

**Theorem 3.3.** *For an arbitrary initial matrix  $(Y_1^{(1)}, Y_2^{(1)}, Y_3^{(1)}) \in \Omega_{1-5-9}$ , a solution of problem 3.2 can be obtained by algorithm 3 within finite number of iterations, and it is also a constrained least-squares solution in  $\Omega_{1-5-9}$  of (3.1).*

#### 4. Numerical experiments

In this section, we design two computation programmes to find constrained solutions of (1.1). Then two numerical examples are given to illustrate the proposed results. All computations are performed using MATLAB. Because of the influence of roundoff errors, we regard a matrix  $A$  as zero matrix if  $\|A\| \leq 10^{-7}$ .

Let  $n$  be the order of the matrix  $X_i$ ,  $k, k_1, k_2$  be the iteration numbers of algorithm 1, algorithm 2 and algorithm 3 respectively, and  $t$  be the computation time (seconds).

**Programme 1.**

- (1) Choose an initial matrix  $(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}) \in \Omega_{1-5-9}$  and set  $k := 1$ .
- (2) If  $\psi(X^{(k)}) = O$ , stop, else, solve for  $(Y_1^{(k)}, Y_2^{(k)}, Y_3^{(k)}) \in \Omega_{1-5-9}$  from (2.4) using algorithm 2. When algorithm 2 breaks, that is (2.4) hasn't constrained solution in  $\Omega_{1-5-9}$ , solve for  $(Y_1^{(k)}, Y_2^{(k)}, Y_3^{(k)}) \in \Omega_{1-5-9}$  from (2.5) using algorithm 3.
- (3) Compute  $X^{(k+1)} = X^{(k)} + Y^{(k)}$ , set  $k := k + 1$  and go to step 2.

**Programme 2.**

- (1) Choose an initial matrix  $(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}) \in \Omega_{1-5-9}$  and set  $k := 1$ .
- (2) If  $\psi(X^{(k)}) = O$ , stop, else, solve for  $(Y_1^{(k)}, Y_2^{(k)}, Y_3^{(k)}) \in \Omega_{1-5-9}$  from (2.5) using algorithm 3. Especially, when (2.4) has constrained solutions in  $\Omega_{1-5-9}$ , the constrained least-squares solutions in  $\Omega_{1-5-9}$  are also its constrained solutions in  $\Omega_{1-5-9}$ .
- (3) Compute  $X^{(k+1)} = X^{(k)} + Y^{(k)}$ , set  $k := k + 1$  and go to step 2.

**Example 4.1.** Consider (1.1) with the following parameters:

$$P_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$C_i^{(l)} = C + l \times \text{ones}(3), \quad D_i^{(l)} = (C_i^{(l)})^T, \quad E_{ij}^{(l)} = -u_i u_j^T,$$

$$S^{(l)} = \sum_{i=1}^3 C_i^{(l)} X_i^* D_i^{(l)} + \sum_{i,j=1}^3 X_i^* E_{ij}^{(l)} X_j^* \quad (i, j = 1, 2, 3; l = 1, 2),$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix}, \quad X_1^* = \begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0 \\ 0.5 & 0 & 2 \end{bmatrix},$$

$$X_2^* = \begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & -0.5 \\ 0 & 0 & 2 \end{bmatrix}, \quad X_3^* = \begin{bmatrix} 1 & 0 & 0.25 \\ 0 & 1 & 0.25 \\ 0.25 & 0.25 & 2 \end{bmatrix},$$

$$u_1 = (1, 1, 0)^T, \quad u_2 = (0, 1, 1)^T, \quad u_3 = (0, 0, 1)^T.$$

We can easily see that (1.1) has the constrained solution  $(X_1^*, X_2^*, X_3^*) \in \Omega_{1-5-9}$ . By applying programmes 1 and 2 with the initial matrix  $X_i^{(1)} = \text{eye}(3)$ ,  $Y_i^{(1)} = \text{zeros}(3)$  ( $i = 1, 2, 3$ ), we obtain the constrained solution in  $\Omega_{1-5-9}$  of (1.1) as follows:

$$X_1^{(5)} = \begin{bmatrix} 1.0000 & -0.0000 & 0.5000 \\ -0.0000 & 1.0000 & 0.0000 \\ 0.5000 & 0.0000 & 2.0000 \end{bmatrix},$$

$$X_2^{(5)} = \begin{bmatrix} 1.0000 & -0.0000 & 0.5000 \\ -0.0000 & 1.0000 & -0.5000 \\ -0.0000 & 0.0000 & 2.0000 \end{bmatrix},$$

$$X_3^{(5)} = \begin{bmatrix} 1.0000 & -0.0000 & 0.2500 \\ -0.0000 & 1.0000 & 0.2500 \\ 0.2500 & 0.2500 & 2.0000 \end{bmatrix}.$$

The iteration numbers and computation time are listed in Table 1.



**Table 1.** The iteration numbers and computation time of programmes 1 and 2.

Results	$n$	$k$	$k_1$	$k_2$	$t$
Programme 1	3	5	97	0	0.4521
Programme 2	3	5	0	184	3.4310

From the results in Table 1, we see that programme 1 is more effective when the derived linear matrix equations are always have constrained solutions in  $\Omega_{1-5-9}$ .

**Example 4.2.** Consider (1.1) with the following parameters:

$$C_i^{(l)} = i \times C + l \times \text{ones}(3), \quad D_i^{(l)} = (C_i^{(l)})^T,$$

$$E_{ij}^{(l)} = -u_i u_j^T \quad (i, j = 1, 2, 3; l = 1, 2),$$

$$S^{(1)} = \begin{bmatrix} -18.5 & 32 & -29 \\ 36 & 50.5 & 4.5 \\ -43 & -5.5 & -37.5 \end{bmatrix}, \quad S^{(2)} = \begin{bmatrix} 2.5 & 68 & -11 \\ 81 & 83.5 & 29.5 \\ -22 & 17.5 & -22.5 \end{bmatrix},$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix},$$

$$u_1 = (1, 1, 0)^T, \quad u_2 = (0, 1, 1)^T, \quad u_3 = (0, 0, 1)^T.$$

By applying programmes 1 and 2 with the initial matrix  $X_i^{(1)} = \text{ones}(3)$ ,  $Y_i^{(1)} = \text{zeros}(3)$  ( $i = 1, 2, 3$ ),  $P_1$  and  $P_2$  are identity matrices, we obtain a special constrained solution (now,  $X_1$  and  $X_3$  are symmetric matrices,  $X_2$  is a general matrix) in  $\Omega_{1-5-9}$  of (1.1) as follows:

$$X_1^{(4)} = \begin{bmatrix} 1.1352 & 1.5249 & 1.5924 \\ 1.5249 & 1.0188 & 0.9904 \\ 1.5924 & 0.9904 & 1.1223 \end{bmatrix},$$

$$X_2^{(4)} = \begin{bmatrix} 0.9332 & 0.8411 & 1.0152 \\ 1.8411 & 0.9730 & 0.9549 \\ 2.0152 & 0.9549 & 1.0135 \end{bmatrix},$$

$$X_3^{(4)} = \begin{bmatrix} 0.9769 & 1.6245 & 1.4150 \\ 1.6245 & 1.0128 & 1.0299 \\ 1.4150 & 1.0299 & 0.9482 \end{bmatrix}.$$

When  $X_i^{(1)} = \text{eye}(3)$  ( $i = 1, 2, 3$ ), others remain unchanged, we obtain another special constrained solution in  $\Omega_{1-5-9}$  of (1.1) as follows:

$$X_1^{(6)} = \begin{bmatrix} 1.8147 & 1.8833 & 2.4692 \\ 1.8833 & 1.1089 & 1.0227 \\ 2.4692 & 1.0227 & 2.6495 \end{bmatrix},$$

$$X_2^{(6)} = \begin{bmatrix} 1.2581 & -0.0773 & 0.6145 \\ 0.9227 & 0.6873 & 0.5880 \\ 1.6145 & 0.5880 & 0.2754 \end{bmatrix},$$

$$X_3^{(6)} = \begin{bmatrix} 0.3466 & 2.2276 & 1.1261 \\ 2.2276 & 1.3542 & 1.0098 \\ 1.1261 & 1.0098 & 1.1756 \end{bmatrix}.$$

Thus it can be seen that if the constrained solution of Eq (1.1) is not unique, we can get different constrained solutions in  $\Omega_{1-5-9}$  when choosing different initial matrices.

## 5. Conclusions

In this paper, an iterative algorithm is studied to find different constrained solutions. By using the proposed algorithm, we compute a set of different constrained solutions in  $\Omega_{1-5-9}$  of multivariate quadratic matrix equations. The provided examples illustrate the effectiveness of the new iterative algorithm.

There are still some results we can obtain by changing the initial matrices  $X_i^{(1)}$  and  $Y_i^{(1)}$ , the direction matrix  $Z_k$  in algorithm 2 and Eq (3.3) in algorithm 3. In this way, we can get other kind of constrained solutions, which are not only interesting but also valuable. It remains to study in our further work.

## Acknowledgments

This research was funded by Doctoral Fund Project of Shandong Jianzhu University grant number X18091Z0101.

## Conflict of interest

The author declares no conflict of interest.

## References

1. N. J. Higham, H. M. Kim, Solving a quadratic matrix equation by Newton's method with exact line searches, *SIAM J. Matrix Anal. Appl.*, **23** (2001), 303–316. doi: 10.1137/S0895479899350976.
2. C. R. Chen, R. C. Li, C. F. Ma, Highly accurate doubling algorithm for quadratic matrix equation from quasi-birth-and-death process, *Linear Algebra Appl.*, **583** (2019), 1–45. doi: 10.1016/j.laa.2019.08.018.
3. V. Angelova, M. Hached, K. Jbilou, Sensitivity of the solution to nonsymmetric differential matrix Riccati equation, *Mathematics*, **9** (2021), 1–18. doi: 10.3390/math9080855.
4. D. A. Bini, B. Iannazzo, F. Poloni, A fast Newton's method for a nonsymmetric algebraic Riccati equation, *SIAM J. Matrix Anal. Appl.*, **30** (2008), 276–290. doi: 10.1137/070681478.
5. P. Benner, Z. Bujanović, P. Kürschner, J. Saak, RADI: A low-rank ADI-type algorithm for large scale algebraic Riccati equations, *Numer. Math.*, **138** (2018), 301–330. doi: 10.1007/s00211-017-0907-5.

6. C. L. Liu, J. G. Xue, R. C. Li, Accurate numerical solution for shifted M-matrix algebraic Riccati equations, *J. Sci. Comput.*, **84** (2020), 1–27. doi: 10.1007/s10915-020-01263-4.
7. A. G. Wu, H. J. Sun, Y. Zhang, A novel iterative algorithm for solving coupled Riccati equations, *Appl. Math. Comput.*, **364** (2020), 1–14. doi: 10.1016/j.amc.2019.124645.
8. S. Koskie, C. Coumarbatch, Z. Gajic, Exact slow-fast decomposition of the singularly perturbed matrix differential Riccati equation, *Appl. Math. Comput.*, **216** (2010), 1401–1411. doi: 10.1016/j.amc.2010.02.040.
9. E. Burlakov, V. Verkhlyutov, I. Malkov, V. Ushakov, Assessment of cortical travelling waves parameters using radially symmetric solutions to neural field equations with microstructure, In: *Proceedings of international conference on neuroinformatics*, Cham: Springer, 2020, 51–57. doi: 10.1007/978-3-030-60577-3\_5.
10. M. Dehghan, M. Hajarani, An iterative algorithm for solving a pair of matrix equations  $AYB = E$ ,  $CYD = F$  over generalized centro-symmetric matrices, *Comput. Math. Appl.*, **56** (2008), 3246–3260. doi: 10.1016/j.camwa.2008.07.031.
11. T. X. Yan, C. F. Ma, The BCR algorithms for solving the reflexive or anti-reflexive solutions of generalized coupled Sylvester matrix equations, *J. Franklin I.*, **357** (2020), 12787–12807. doi: 10.1016/j.jfranklin.2020.09.030.
12. A. P. Liao, Z. Z. Bai, The constrained solutions of two matrix equations, *Acta Math. Sinica*, **18** (2002), 671–678. doi: 10.1007/s10114-002-0204-8.
13. X. M. Liu, K. Y. Zhang, MCG method for a different constrained least square solution of two-variables linear matrix equations for recurrent event data (in Chinese), *Acta Math. Appl. Sin.*, **34** (2011), 938–948.
14. J. H. Long, X. Y. Hu, L. Zhang, Improved Newton's method with exact line searches to solve quadratic matrix equation, *J. Comput. Appl. Math.*, **222** (2008), 645–654. doi: 10.1016/j.cam.2007.12.018.
15. C. H. Guo, W. W. Lin, Convergence rates of some iterative methods for nonsymmetric algebraic Riccati equations arising in transport theory, *Linear Algebra Appl.*, **432** (2010), 283–291. doi: 10.1016/j.laa.2009.08.004.
16. K. Y. Zhang, S. S. Zhu, X. M. Liu, An iterative method to find symmetric solutions of two-variable Riccati matrix equation (in Chinese), *Acta Math. Appl. Sin.*, **36** (2013), 831–839.
17. S. S. Zhu, K. Y. Zhang, An iterative method for different constrained solutions of two-variable Riccati matrix equation (in Chinese), *J. Syst. Sci. Math. Sci.*, **33** (2013), 197–205. doi: 10.12341/jssms12039.
18. C. Q. Lv, C. F. Ma, A modified CG algorithm for solving generalized coupled Sylvester tensor equations, *Appl. Math. Comput.*, **365** (2020), 1–15. doi: 10.1016/j.amc.2019.124699.
19. X. D. Zhang, *Matrix analysis and applications*, 1 Ed., Cambridge: Cambridge University Press, 2017.

