Vol.5.Issue.3.2017 (July-Sept)
©KY PUBLICATIONS

BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH

A Peer Reviewed International Research Journal

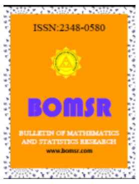
http://www.bomsr.com
Email:editorbomsr@gmail.com

INTERNATIONAL STANDARD SERIAL NUMBER 2348-0580

RESEARCH ARTICLE

LEAST SQUARES SOLUTIONS OF NONLINEAR MATRIX EQUATIONS

MINGHUI WANG , YUNCUI ZHANG, LUPING XU
Qingdao University of Science and Technology
Qingdao, Shandong, China



ABSTRACT

In this paper, we study the least squares solutions of nonlinear matrix equations, the least squares solutions of the nonlinear equations is solved by the steepest descent method and the Newton algorithm.

Keywords*:* Least squares solutions, steepest descent method, Newton method.

1. INTRODUCTION

In this paper, we consider the nonlinear matrix equations:

$$\begin{cases} X + AY^{-1}B = E \\ Y + CX^{-1}D = F \end{cases}, \qquad (1.1)$$

where A, B, C, D, E, F are n order complex matrices, X, Y are n order nonsingular matrices.

In 1970s, more and more scholars began to study the nonlinear matrix equations, J. Ortega [4] and W. Rheinbold [5] researched the theoretical methods and numerical methods of the nonlinear matrix equations (1.1). The nonlinear matrix equations are more complicated than linear matrix equations, then we need further research about nonlinear matrix equations. Newton algorithm is a classical method to solve nonlinear matrix equations, many new methods are improved by Newton algorithm. The Newton algorithm and the Newtonian algorithm have many results, many scholars have made improvement and amendment of Newton algorithm.

2. PRELIMINARIES

Definition2.1[9]According the raw(or line) of matrix, forming a long vector raw after raw (or line by line), then carry out certain operations, we called it Matrix Vec Operator. In this paper we use raw of matrix vec operator, if $A = [a_{ij}]$ is $m \times n$ matrix, then matrix A can be expressed as:

$$vec(A)=\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \\ a_{12} \\ \vdots \\ a_{m2} \\ a_{13} \\ \vdots \\ a_{mn} \end{bmatrix}(mn\times 1).$$

Definition2.2[8] Suppose F is a true value function defined on the open domain D of the n-dimensional Euclid space R^n, if F can be guided of x at D, the partial derivative $\partial_j f_i(x)\equiv\frac{\partial f_i(x)}{\partial x_j}, i=1,2,\cdots,m$ and each component function $f_1, f_2,\cdots, f_m$ is present at x of F, and we have:

$$F'(x)=(\partial_j f_i(x))\equiv\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix},$$

the matrix in the above is the Jacobi matrix of F .

Property2.1[6] For any matrix $A,B,C\in C^{n\times n}$, k is a constant, we have the following properties :

(1) $(A\otimes B)\otimes C=A\otimes(B\otimes C)$;

(2) $(kA)\otimes B=A\otimes(kB)=k(A\otimes B)$;

(3) normally, we have $A\otimes B\neq B\otimes A$.

Property2.2[6] For any matrix $A,B,C\in C^{n\times n}$, k is a constant, we have the following properties :

(1) $vec(A\pm B)=vec(A)\pm vec(B)$;

(2) $vec(kA)=kvec(A)$;

(3) $vec(ABC)=(C^T\otimes A)vec(B)$.

3. SOME RESULTS

We use the least squares method to solve the nonlinear matrix equations (1.1), which can be transformed into the following form:

$$\left\|X+AY^{-1}B-E\right\|_F+\left\|Y+CX^{-1}D-F\right\|_F=\min.$$

First, we let the real valued function:

$$
\begin{aligned}
& f(X,Y) \\
&= \|X + AY^{-1}B - E\|_F^2 + \|Y + CX^{-1}D - F\|_F^2 \\
&= \|E\|^2 + \|F\|^2 + tr(X^T X) - 2tr(E^T X) + 2tr(X^T AY^{-1}B) \\
& -2tr(E^T AY^{-1}B) + tr(B^T Y^{-T} A^T AY^{-1}B) + tr(Y^T Y) - 2tr(F^T Y) \\
& +2tr(Y^T CX^{-1}D) - 2tr(F^T CX^{-1}D) + tr(D^T X^{-T} C^T CX^{-1}D)
\end{aligned}
$$

Theorem3.1 If $A,B,C,D,E,F \in C^{n\times n}$, the least squares solution of the nonlinear matrix equations $\begin{cases} X + AY^{-1}B = E \\ Y + CX^{-1}D = F \end{cases}$ is equal to minimum point of the real valued function $f(X,Y)$.

Proof: we can get partial derivative of the real valued function $f(X,Y)$:

$$
\begin{aligned}
p_X &= \frac{\partial f(X,Y)}{\partial X} \\
&= -2E + 2AY^{-1}B + 2X - 2X^{-T}C^T Y D^T X^{-T} \\
& +2X^{-T}C^T F D^T X^{-T} - 2X^{-T}C^T CX^{-1}DD^T X^{-T} \\
&= 2(AY^{-1}B + X - E) - 2X^{-T}C^T(CX^{-1}D + Y - F)D^T X^{-T}
\end{aligned}
$$

$$
\begin{aligned}
p_Y &= \frac{\partial f(X,Y)}{\partial Y} \\
&= -2F + 2CX^{-1}D + 2Y - 2Y^{-T}A^T X B^T Y^{-T} \\
& +2Y^{-T}A^T E B^T Y^{-T} - 2Y^{-T}A^T AY^{-1}BB^T Y^{-T} \\
&= 2(Y + CX^{-1}D - F) - 2Y^{-T}A^T(X + AY^{-1}B - E)B^T Y^{-T}
\end{aligned}
$$

If $f(X,Y)$ reaches extreme value at a point, we have:

$$
\begin{cases}
\left.\dfrac{\partial f(X,Y)}{\partial X}\right|_{X=X_*,Y=Y_*} = 0, \\
\left.\dfrac{\partial f(X,Y)}{\partial Y}\right|_{X=X_*,Y=Y_*} = 0.
\end{cases}
$$

We also have:

$$
\begin{cases}
X_* + AY_*^{-1}B - E = X_*^{-T}C^T(Y_* + CX_*^{-1}D - F)D^T X_*^{-T}, & (3.2) \\
Y_* + CX_*^{-1}D - F = Y_*^{-T}A^T(X_* + AY_*^{-1}B - E)B^T Y_*^{-T}. & (3.3)
\end{cases}
$$

Substituting (3.2) into (3.3), we get:

$$Y_* + CX_*^{-1}D - F = Y_*^{-T}A^T X_*^{-T}C^T(Y_* + CX_*^{-1}D - F)D^T X_*^{-T}B^T Y_*^{-T},$$

$$Y_* + CX_*^{-1}D - F = (CX_*^{-1}AY_*^{-1})^T(Y_* + CX_*^{-1}D - F)(Y_*^{-1}BX_*^{-1}D)^T.$$

So we have $Y_* + CX_*^{-1}D - F = 0$, similarly, we can get $X_* + AY_*^{-1}B - E = 0$, X_*, Y_* are the least squares solution of nonlinear matrix equations.

Then we can get the least squares solutions of the nonlinear matrix equation $\begin{cases} X + AY^{-1}B = E \\ Y + CX^{-1}D = F \end{cases}$ is equal to the minimum point of solving real valued function $f(X,Y)$.

3.1 The steepest descent method for solving equations(1.1)

The steepest descent method is also known as the gradient method, it was proposed by famous mathematician Cauchy in 1867. It is the oldest way of analytic methods and the basis of the optimization method.

The process of steepest descent method is like blind down the mountain. We determine the initial matrix X_0, Y_0 and the direction p_{X_0}, p_{Y_0}, the line $\begin{cases} X = X_0 + \alpha p_{X_0} \\ Y = Y_0 + \alpha p_{Y_0} \end{cases}$ passing (X_0, Y_0)

and its downhill direction is p_{X_0}, p_{Y_0}, then we makes a point on the line for all real numbers α, we get:

$$f(X_0 + \alpha_0 p_{X_0}, Y_0 + \alpha_0 p_{Y_0}) \le f(X_0 + \alpha p_{X_0}, Y_0 + \alpha p_{Y_0}).$$

In the line, $f(X,Y)$ reaches a minimum value at X_1, Y_1, then we determine the downhill p_{X_1}, p_{Y_1} and continue next step operation on the line $\begin{cases} X = X_1 + \alpha p_{X_1} \\ Y = Y_1 + \alpha p_{Y_1} \end{cases}$, we find α_1 that make $f(X,Y)$ reaches a minimum value on the line $\begin{cases} X_2 = X_1 + \alpha_1 p_{X_1} \\ Y_2 = Y_1 + \alpha_1 p_{Y_1} \end{cases}$, we let:

$$f(X_1 + \alpha_1 p_{X_1}, Y_1 + \alpha_1 p_{Y_1}) \le f(X_1 + \alpha p_{X_1}, Y_1 + \alpha p_{Y_1}),$$

and so on, we can get $\alpha_0, \alpha_1, \cdots$ and $p_{X_0}, p_{Y_0}, p_{X_1}, p_{Y_1}, \cdots$.

p_{X_k}, p_{Y_k} are search direction, α_k is step length.First, we find the downhill direction p_{X_k}, p_{Y_k} of point X_k, Y_k, then we determine the step length α_k of the line $\begin{cases} X = X_k + \alpha p_{X_k} \\ Y = Y_k + \alpha p_{Y_k} \end{cases}$,

and we have:

$$f(X_k + \alpha_k p_{X_k}, Y_k + \alpha_k p_{Y_k}) \le f(X_k + \alpha p_{X_k}, Y_k + \alpha p_{Y_k}),$$

Finally, we can get $X_{k+1} = X_k + \alpha_k p_{X_k}$.

First, we determine the direction of the downhill: the fastest direction of real value function $f(X,Y)$ is the gradient direction, so the negative gradient direction should be the reduce fastest direction of $f(X,Y)$, we can choose the direction of the downward for the negative gradient:

$$\begin{aligned} p_{X_k} &= -\left.\frac{\partial f(X,Y)}{\partial X}\right|_{X=X_k, Y=Y_k} \\ &= 2X_k^{-T} C^T (CX_k^{-1} D + Y_k - F) D^T X_k^{-T} - 2(AY_k^{-1} B + X_k - E) \\ p_{Y_k} &= -\left.\frac{\partial f(X,Y)}{\partial Y}\right|_{X=X_k, Y=Y_k} \\ &= 2Y_k^{-T} A^T (X_k + AY_k^{-1} B - E) B^T Y_k^{-T} - 2(Y_k + CX_k^{-1} D - F) \end{aligned}$$

Then determine the step length α_k: we assume X_k, Y_k and select downhill direction p_{X_k}, p_{Y_k} on the line $\begin{cases} X = X_k + \alpha p_{X_k} \\ Y = Y_k + \alpha p_{Y_k} \end{cases}$,then we let α_k make $f(X,Y)$ in line $\begin{cases} X_{k+1} = X_k + \alpha_k p_{X_k} \\ Y_{k+1} = Y_k + \alpha_k p_{Y_k} \end{cases}$ reach minimum, and we let:

$$\frac{\partial f(X_k+\alpha p_{X_k}, Y_k+\alpha p_{Y_k})}{\partial \alpha}=0.$$

It is difficult to find out α of the trace in matrix, using dynamic change and adaptive method to α, finding the best value of α.

3.2 The Newton method for solving equations(1.1)

Newton put forward iterative method for solving the nonlinear equation, this method is called Newton algorithm. People have been using and improving Newton algorithm over the past three hundred years and expanding the application of Newton algorithm. When we use Newton algorithm solve the nonlinear equation, the convergence rate of the nonlinear equation is very fast. Newton algorithm is still an important method for solving problems until now.

Newton algorithm is an important linearization method, its basic idea is to transform nonlinear equation into linear equation step by step. Solving nonlinear equations is only an extension of nonlinear equation.

We transform $\begin{cases} X+A^*Y^{-1}A=E \\ Y+B^*X^{-1}B=F \end{cases}$ to the general form of the two equations changed by the two variables :

$$\begin{cases} f_1(X,Y)=X+AY^{-1}B-E=0 \\ f_2(X,Y)=Y+CX^{-1}D-F=0 \end{cases},$$

if we let $F(X,Y)=(f_1(X,Y), f_2(X,Y))^T, 0=(0,0)^T$.

Then the above nonlinear equations can be rewritten as:

$F(X,Y)=0,$

$$\begin{aligned} f_1(X,Y+\Delta Y)-f_1(X,Y) &= X+A(Y+\Delta Y)^{-1}B-E-(X+AY^{-1}B-E) \\ &= A(Y+\Delta Y)^{-1}B-AY^{-1}B \\ &= A[Y(I+Y^{-1}\Delta Y)]^{-1}B-AY^{-1}B \\ &= A[(I+Y^{-1}\Delta Y)^{-1}Y^{-1}]B-AY^{-1}B \\ &= A[(I-Y^{-1}\Delta Y+\cdots)Y^{-1}]B-AY^{-1}B \\ &= A(Y^{-1}-Y^{-1}\Delta YY^{-1}+\cdots)B-AY^{-1}B \\ &= -AY^{-1}\Delta YY^{-1}B+\cdots \end{aligned}.$$

From the above derivation we can get:

$$f_1(X,Y)-f_1(X,Y_k)=-AY_k^{-1}(Y-Y_k)Y_k^{-1}B\ ;$$

$$\begin{aligned} f_1(X,Y) &= f_1(X,Y_k)-AY_k^{-1}(Y-Y_k)Y_k^{-1}B \\ &= X+AY_k^{-1}B-E-AY_k^{-1}(Y-Y_k)Y_k^{-1}B \\ &= X+2AY_k^{-1}B-AY_k^{-1}YY_k^{-1}B-E \\ &= X+A(2Y_k^{-1}-Y_k^{-1}YY_k^{-1})B-E \\ &= 0 \end{aligned}.$$

Similarly, $f_2(X,Y)=Y+C(2X_k^{-1}-X_k^{-1}XX_k^{-1})D-F=0$.

Then we can get:

$$F(X,Y)=\begin{pmatrix} f_1(X,Y) \\ f_2(X,Y) \end{pmatrix}=\begin{pmatrix} X+A(2Y_k^{-1}-Y_k^{-1}YY_k^{-1})B-E \\ Y+C(2X_k^{-1}-X_k^{-1}XX_k^{-1})D-F \end{pmatrix}=0.$$

So we can get the following formula:

$$\begin{cases} X - AY_k^{-1}YY_k^{-1}B = E - 2AY_k^{-1}B & (3.4) \\ Y - CX_k^{-1}XX_k^{-1}D = F - 2CX_k^{-1}D & (3.5) \end{cases}.$$

Substituting X of formula (3.4) into formula (3.5) yields:

$$Y - CX_k^{-1}(E - 2AY_k^{-1}B + AY_k^{-1}YY_k^{-1}B)X_k^{-1}D = F - 2CX_k^{-1}D \;;$$

Substituting Y of formula (3.5) into formula (3.4) yields:

$$X - AY_k^{-1}(F - 2CX_k^{-1}D + CX_k^{-1}XX_k^{-1}D)Y_k^{-1}B = E - 2AY_k^{-1}B.$$

Then we can get：

$$\begin{cases} Y - CX_k^{-1}AY_k^{-1}YY_k^{-1}BX_k^{-1}D = F - 2CX_k^{-1}D + CX_k^{-1}(E - 2AY_k^{-1}B)X_k^{-1}D \\ X - AY_k^{-1}CX_k^{-1}XX_k^{-1}DY_k^{-1}B = E - 2AY_k^{-1}B + AY_k^{-1}(F - 2CX_k^{-1}D)Y_k^{-1}B \end{cases}.$$

If we let

$$P = CX_k^{-1}AY_k^{-1}, Q = Y_k^{-1}BX_k^{-1}D, R = F - 2CX_k^{-1}D + CX_k^{-1}(E - 2AY_k^{-1}B)X_k^{-1}D;$$

$$S = AY_k^{-1}CX_k^{-1}, T = X_k^{-1}DY_k^{-1}B, W = E - 2AY_k^{-1}B + AY_k^{-1}(F - 2CX_k^{-1}D)Y_k^{-1}B,$$

then the above iterative equations are transformed into:

$$\begin{cases} Y - PYQ = R & (3.6) \\ X - SXT = W & (3.7) \end{cases}.$$

So nonlinear matrix equations can be transformed linear matrix equations . For getting X, Y, we can do the following:

Straighten the operation with (3.6) on the left and right sides of the matrix:

$$vec(Y) - vec(PYQ) = vec(R)$$

$$vec(Y) - (Q^T \otimes P)vec(Y) = vec(R)$$

$$(I - Q^T \otimes P)vec(Y) = vec(R)$$

$$vec(Y) = (I - Q^T \otimes P)^{-1}vec(R)$$

Straighten the operation with (3.7) on the left and right sides of the matrix:

$$vec(X) = (I - T^T \otimes S)^{-1}vec(W).$$

Then we can get:

$$\begin{cases} vec(Y) = (I - Q^T \otimes P)^{-1}vec(R) \\ vec(X) = (I - T^T \otimes S)^{-1}vec(W) \end{cases}.$$

The iterative equation of the Newton algorithm is as follows:

$$\begin{cases} vec(Y_{k+1}) = [I - (Y_k^{-1}BX_k^{-1}D)^T \otimes (CX_k^{-1}AY_k^{-1})]^{-1}vec(\delta_k + CX_k^{-1}\gamma_k X_k^{-1}D) \\ vec(X_{k+1}) = [I - (X_k^{-1}DY_k^{-1}B)^T \otimes (AY_k^{-1}CX_k^{-1})]^{-1}vec(\gamma_k + AY_k^{-1}\delta_k Y_k^{-1}B) \end{cases}.$$

where $\delta_k = F - 2CX_k^{-1}D, \gamma_k = E - 2AY_k^{-1}B$.

Algorithm3.1(Newton algorithm)

(1)Initialize: we give the initial approximation matrix X_0, Y_0 and iterative number N ;

(2) Iteration: for $k = 0,1,2\cdots$, if $k < N$, then following the iteration equation,

$$\begin{cases} vec(Y_{k+1}) = [I - (Y_k^{-1}BX_k^{-1}D)^T \otimes (CX_k^{-1}AY_k^{-1})]^{-1}vec(\delta_k + CX_k^{-1}\gamma_k X_k^{-1}D) \\ vec(X_{k+1}) = [I - (X_k^{-1}DY_k^{-1}B)^T \otimes (AY_k^{-1}CX_k^{-1})]^{-1}vec(\gamma_k + AY_k^{-1}\delta_k Y_k^{-1}B) \end{cases}'$$

where $\delta_k = F - 2CX_k^{-1}D, \gamma_k = E - 2AY_k^{-1}B$, restore the matrix X_{k+1}, Y_{k+1} before the operation of matrix vec operator;Otherwise, we turn to (5);

(3)If we have iterated k steps as above,we get $f_1(X_k, Y_k)$, $f_2(X_k, Y_k)$ and $f_1(X_{k+1}, Y_{k+1})$, $f_2(X_{k+1}, Y_{k+1})$;

(4) If $\| f_1(X_k, Y_k)\|_2 + \| f_2(X_k, Y_k)\|_2 \le \| f_1(X_{k+1}, Y_{k+1})\|_2 + \| f_2(X_{k+1}, Y_{k+1})\|_2$, we let:

$$\bar{X} = \lambda X_k + (1-\lambda) X_{k+1},$$

$$\bar{Y} = \lambda Y_k + (1-\lambda) Y_{k+1},$$

where $\lambda \in (0,1]$, the initial value of λ is 1, we halve the processing means $\lambda = \frac{\lambda}{2}$, output $\bar{X}, \bar{Y}$ and let $f_1(X_{k+1}, Y_{k+1}) = f_1(\bar{X}, \bar{Y})$, $f_2(X_{k+1}, Y_{k+1}) = f_2(\bar{X}, \bar{Y})$, until the conditions are not satisfied and then jump out of this cycle (2);

(5)End.

4. NUMERICAL EXAMPLE

In this section, we give a numerical example to illustrate the efficiency of Algorithm 3.1, and all the tests are performed by MATLAB, version 7.0.

Example In the solution(1.1), A, B, C, D, X, Y are defined by $rand(n,n)$, then we get $E = X + AY^{-1}B$, $F = Y + CX^{-1}D$, we have initial matrix $X_0 = I_{n\times n}$, $Y_0 = I_{n\times n}$, $y = \log 10\left(\|E - X - A^*Y^{-1}A\|_F + \|F - Y - B^*X^{-1}B\|_F\right)$, Fig 4.1 is depicts $n=10$ of change curve between iteration steps k and residual y .

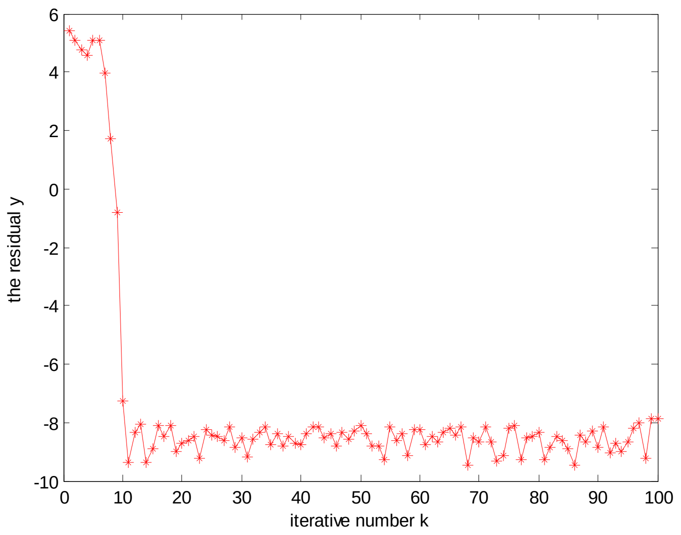


Fig 4.1 The iterative number k with respect to residual y when $n=10$

From Figure 4.1 we can get Newton algorithm for solving (1.1), the nonlinear matrix equations is convergent and converges rate is very fast.

From the Newton algorithm and the numerical examples, we get the convergence rate of Newton algorithm is more faster than other methods for solving nonlinear matrix equations, but the dependence on the initial value is very high, and the initial value of iteration X_0, Y_0 need very close to X_*, Y_*. In this paper, we improve the Newton algorithm and overcome its dependence on the initial value, the large amount of calculation make each step need to calculate Jacobi matrix $F'(X_k, Y_k)$, when Jacobi matrix $F'(X_k, Y_k)$ singular or ill conditioned, it can not continue calculations.

5.ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Shandong Province in China (ZR2013FL014, ZR2016AB07), the Science and Technology Program of Shandong Universitues of China (J13LN73) and the National Natural Science Foundation of China(11771188).

REFERENCES

[1]. W. N. Anderson, T. D. Morley, G. E. Trapp, 1990, Positive solutions to $X = A- BX^{-1}B^*$ .J, *Linear Algebra Appl*, 134: 53-62.

[2]. J. F. Wang, Y. H. Zhang, B. R. Zhu, 2004,The Hermitian positive definite solutions ofmatrix equation $X + A^* X^{-q} A = I(q<1)$. *J. Numer. Sinica*, 2004, 26: 31-73.

[3]. X. Duan, A. Liao. On the existence of Hermitian positive definite solutions of matrix.

[4]. J. Ortega and W. Rheinboldt, 1970, Iterative solution of nonlinear equation in several variables, Academic press, New York, .

[5]. W. Rheinboldt, 1974, Methods for solving systems of nonlinear equations, *J.SIAM Book Series*, phila.

[6]. Shufang Xu, Li Gao, PingwenZhang, 1999, Numerical linear algebra. *M. Peking University Press*

[7]. Jinfang Wang, Yuhai Zhang, Renben Zhu, 2004, The Hermitian positive definite solution of Matrix equation, *M*, 26(1): 61-72.

[8]. Qingyang Li, Zizhong Mo, Liqun Qi, 1987, Numerical solution of nonlinear system of matrix equations, *M*, 38-81.

[9]. Kaiyuan Zhang, 2015, Iterative algorithm for solving matrix equation constraints, *M*, 156-209.

[10]. Xiaomei Cui, Lihui Tan,2012, Study on the positive definite solutions of matrix equation $X + A^* X^{-1} A + B^* X^{-2} B = I$. *J. Journal of jilin institute of chemical technology*, 29(1): 82 -84.

[11]. S. M. El-sayed , A.M. Al-Dbiban, 2005, A new inversion free iteration for solving the equation $X + A^* X^{-1} A = Q$. *J. Comput. Appl. Math*. 181: 148-156.

[12]. Aijing Liu, Guoliang Chen, 2014, On the Hermitian positive definite solutions of nonlinear matrix equation $X^s + \sum_{i=1}^{m} A_i^* X^{-t_1} A_i = Q$. *J. Applied Mathematics and Computation*, (243): 950-959.