KYUNGPOOK Math. J. 47(2007), 239-251

Minimization Method for Solving a Quadratic Matrix Equation

Hyun-Min Kim
Department of Mathematics, Pusan National University, Busan, 609-735, Korea
e-mail : hyunmin@pusan.ac.kr

Abstract. We show how the minimization can be used to solve the quadratic matrix equation and then compare two different types of conjugate gradient method which are Polak and Ribiére version and Fletcher and Reeves version. Finally, some results of the global and local convergence are shown.

1. Introduction

In this paper we consider numerical methods based on nonlinear minimization to solve the quadratic matrix equation

$$Q(X) = AX^2 + BX + C = 0, \qquad A, B, C, X \in \mathbb{C}^{n\times n}. \tag{1.1}$$

Some minimization methods for the general nonlinear equations, for example, the conjugate gradient method [7, Chap. 4], [9, Chap. 4], [15, Sec. 5.2] and nonlinear least-squares problems [5, Chap. 10], [15, Chap. 10] have been much studied. Before considering these methods for solving the quadratic matrix equation we consider two vital questions:

- What kind of minimizer of $f_S(x)$ can we find?
- Is the minimizer of $f_S(x)$ a solvent of $f_G(x)$?

where $f_G(x) = 0$ is the general nonlinear equation from $\mathbb{R}^n$ to $\mathbb{R}^n$ and $f_S(x) = \dfrac{1}{2}\|f_G(x)\|_2^2$ is the objective function from $\mathbb{R}^n$ to $\mathbb{R}$. For the answer of the first question we can consider the global and local minimizers. The global minimizer would be the best one but it can be difficult to find, because our information of $f_S(x)$ is usually restricted. The second choice would be the local minimizer. The next theorem gives a necessary condition for a local minimizer.

Theorem 1.1 [15, Thm. 2.2]**.** *If x^* is a local minimizer and $f_S(x)$ is continuously differentiable in an open neighbourhood of x^*, then*

$$\nabla f_S(x^*) = 0,$$

where

$$\nabla f_S(x) = \left(\frac{\delta f_S}{\delta x_i}\right) \in \mathbb{R}^n.$$

Received January 23, 2006.
2000 Mathematics Subject Classification: 65F30, 65H10.
Key words and phrases: quadratic matrix equation, conjugate gradient method, local and global minimizers.
This work was supported by Pusan National University Research Grant in 2003.

Note that $\nabla f_S(x) = 0$ is a necessary condition for a local minimizer but it is not sufficient. We call x a stationary point if $\nabla f_S(x) = 0$. Consider the function

$$f_S(x) = (\sqrt{x^3+1})^2 \qquad \text{for } x \geq -1.$$

This function has a global minimum at $x = -1$ but $f'_S(0) = 0$ and $f_S(0) = 1$ is not a minimum or a maximum. This example shows that a point $x \in \mathbb{R}$ can be a stationary point but not a local minimizer.

The next result shows that when the objective function $f_S(x)$ is convex, any local minimizer is a global minimizer.

Theorem 1.2 [15, Thm. 2.5]. *When the function f_S satisfies that*

$$f_S(\alpha x + (1-\alpha)y) \leq \alpha f_S(x) + (1-\alpha) f_S(y)$$

for all $x, y \in \mathcal{D}(f_S)$(the domain of the function f_S) and $0 \leq \alpha \leq 1$, any local minimizer x^* is a global minimizer of f_S. If in addition f_S is differentiable, then any stationary point x^* is a global minimizer of f_S.

The answer of the second question is if the general nonlinear equation $f_G(x) = 0$ has a solution x^* then

$$f_S(x^*) = \frac{1}{2}\|f_G(x^*)\|_2^2 = 0$$

and the global minimizer is clearly x^*. However, there may be local minima for which $f_S > 0$.

We now define the gradient and Hessian for general nonlinear matrix equation $G(X) = 0$, where $G : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$. Suppose we wish to solve the nonlinear matrix equation, $G(X) = 0$. One approach is to apply a minimization method to the function

$$g(X) = \frac{1}{2}\|G(X)\|_F^2,$$

where $g : \mathbb{R}^{n\times n} \to \mathbb{R}$. We can also regard g as a mapping $\mathbb{R}^{n^2} \to \mathbb{R}$ if we write the variables as $\xi = \mathrm{vec}(X)$, where

$$\mathrm{vec}(X) = [x_{11}, \cdots, x_{n1}, x_{12}, \cdots, x_{n2}, \cdots, x_{1n}, \cdots, x_{nn}]^T \in \mathbb{R}^{n^2}.$$

The gradient of g can be written as the matrix

$$\nabla g(X) = \left(\frac{\delta g}{\delta x_{ij}}\right) \in \mathbb{R}^{n\times n}$$

and the Hessian

$$\nabla^2 g(X) = \left(\frac{\delta^2 g}{\delta \xi_i \delta \xi_j}\right) \in \mathbb{R}^{n^2\times n^2}.$$

We now obtain an expression for the gradient of the function $g(X)$. Assuming that G is twice continuously differentiable we have the expansion

$$G(X+E) = G(X) + G'_X(E) + N_X(E), \tag{1.2}$$

where $G'_X(E) : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is the Fréchet derivative of G at X in the direction E and $N_X(E) = O(\|E\|)^2$. Applying the vec operator gives

$$\mathrm{vec}(G(X+E)) = r + d + n,$$

where $r = \mathrm{vec}(G)$, $d = \mathrm{vec}(G'_X(E))$ and $n = \mathrm{vec}(N_X(E))$. Thus

$$\begin{aligned} g(X+E) &= \frac{1}{2}\|G(X+E)\|_F^2 \\ &= \frac{1}{2}(r+d+n)^T(r+d+n) \\ &= \frac{1}{2}(r^T r + 2r^T d + O(\|E\|^2). \end{aligned} \tag{1.3}$$

This expansion must agree with the Taylor series

$$\begin{aligned} &g(X+E) \\ =\ & g(X) + (\mathrm{vec}(\nabla g(X))^T \mathrm{vec}(E) + (\mathrm{vec}(E))^T \nabla^2 g(X)\mathrm{vec}(E) + O(\|E\|)^3 \\ =\ & g(X) + \mathrm{trace}(\nabla g(X)^T E) + (\mathrm{vec}(E))^T \nabla^2 g(X)\mathrm{vec}(E) + O(\|E\|)^3. \end{aligned} \tag{1.4}$$

Therefore

$$\mathrm{trace}(\nabla g(X)^T E) \equiv r^T d = (\mathrm{vec}(G(X)))^T \mathrm{vec}(G'_X(E)),$$

which can be rewritten as

$$\mathrm{trace}(E^T \nabla g) = \mathrm{trace}(G(X)^T G'_X(E)).$$

We can use this equation to determine ∇g by setting $E = e_i e_j^T$ and using $\mathrm{trace}(AB) = \mathrm{trace}(BA)$:

$$\begin{aligned} (\nabla g)_{ij} &= \mathrm{trace}(e_j e_i^T \nabla g) \\ &= \mathrm{trace}(G(X)^T G'_X(e_i e_j^T)). \end{aligned} \tag{1.5}$$

By equating the second order terms in and we find that, writing $\mathrm{quad}(y)$ for the quadratic part of y in variable E,

$$\begin{aligned} (\mathrm{vec}(E))^T \nabla^2 g(X)\mathrm{vec}(E) &= \frac{1}{2}(d^T d + 2r^T \mathrm{quad}(s)) \\ &= \frac{1}{2}\mathrm{trace}\big(G'_X(E)^T G'_X(E) \\ &\qquad + 2G(X)^T \mathrm{quad}(N_X(E))\big). \end{aligned} \tag{1.6}$$

We will not attempt to simplify this expression further for general G. However, we can answer the important question of whether the Hessian is positive definite at a solution. At a solution X we have $G(S) = 0$ and so

$$(\mathrm{vec}(E))^T \nabla^2 g(X)\mathrm{vec}(E) = \frac{1}{2}\mathrm{trace}(G'_X(E)^T G'_X(E)).$$

Hence the Hessian of g is positive definite at X if and only if $\mathrm{trace}(G'_X(E)^T G'_X(E)) > 0$ for all nonzero E, that is, if and only if G'_X is nonsingular.

Using the gradient of $g(X) = \|G(X)\|_F^2$, $\nabla g(X)$, we can apply several numerical methods based on minimization.

2. Conjugate gradient method for $Q(X) = 0$

The conjugate gradient method for solving a nonlinear equation was introduced by Fletcher and Reeves [6]. This nonlinear conjugate gradient method can be adapted to solve the quadratic matrix equation with the function

$$f(X) = \frac{1}{2}\|Q(X)\|_F^2, \tag{2.1}$$

where $f : \mathbb{R}^{n\times n} \to \mathbb{R}$, and use the conjugate gradient method to find

$$\min_{X\in\mathbb{R}^{n\times n}} f(X).$$

To adapt the conjugate gradient method for solving $Q(X) = 0$ we first define the gradient and Hessian of the objective function $f(X)$. By expanding $Q(X+E)$ we find that $D_X(E) = AEX + (AX + B)E$. Therefore, from

$$\begin{aligned}(\nabla f)_{ij} &= \text{trace}(Q(X)^T(Ae_ie_j^TX + (AX+B)e_ie_j^T))\\ &= e_j^TXQ(X)^TAe_i + e_j^TQ(X)^T(AX+B)e_i\end{aligned}$$

and hence

$$\nabla f(X) = A^TQ(X)X^T + (AX+B)^TQ(X). \tag{2.2}$$

Also, from a result of Chu [2] on the generalized Sylvester equation it follows that D_X is nonsingular if the pair $(A, AX + B)$ is regular (that is, $\det(A - \lambda(AX + B))$ is not identically zero in λ) and the eigenvalues of the pair are distinct from the eigenvalues of $-X$.

To obtain the Hessian we note that $N_X(E) = AE^2$ and so can be written

$$\begin{aligned}&(\text{vec}(E))^T\nabla^2 f(X)\text{vec}(E)\\ &\quad= \frac{1}{2}\text{trace}\big((AEX + (AX+B)E)^T(AEX + (AX+B)E) + 2Q(X)^TAE^2\big).\end{aligned}$$

Setting $E = e_ie_j^T$ and $r = (j-1)n + i$ we have

$$\begin{aligned}&e_r^T\nabla^2 f(X)e_r \\ =\ &\frac{1}{2}\text{trace}\big([X^Te_je_i^TA^T + e_je_i^T(AX+B)^T][Ae_ie_j^TX + (AX+B)e_ie_j^T]\\ &\qquad +2Q(X)^TAe_ie_j^T(e_j^Te_i)\big)\\ =\ &\frac{1}{2}\text{trace}(X^Te_j(e_i^TA^TAe_i)e_j^TX + 2X^Te_je_i^TA^T(AX+B)e_ie_j^T\\ &\qquad +e_je_i^T(AX+B)^T(AX+B)e_ie_j^T + 2(e_j^Te_i)e_j^TQ(X)^TAe_i)\\ =\ &\frac{1}{2}\big((A^TA)_{ii}(XX^T)_{jj} + 2(A^T(AX+B))_{ii}x_{jj}\\ &\qquad +((AX+B)^T(AX+B))_{ii} + 2(e_j^Te_i)(Q(X)^TA)_{ji}\big).\end{aligned} \tag{2.3}$$

This determines the diagonal elements of the Hessian. From the identity

$$(e_i + e_j)^T A(e_i + e_j) = a_{ii} + 2a_{ij} + a_{jj} \tag{2.4}$$

it follows that we can determine the off-diagonal entries of $\nabla^2 f$ by evaluating $(e_i + e_j)^T \nabla^2 f(X)(e_i + e_j)$ for all $i \neq j$.

The conjugate gradient method algorithm for solving the quadratic matrix equation (1 1) can be simply defined as follows.

Algorithm 2.1. Given $X_0 \in \mathbb{R}^{n\times n}$ and $f : \mathbb{R}^{n\times n} \to \mathbb{R}$ this algorithm attempts to minimize $f(X)$.

```
*Evaluate* $f_0 = f(X_0)$, $\nabla f_0 = \nabla f(X_0)$
$k = 0$; $\mathcal{D}_0 = -\nabla f(X_0)$
**while** $\nabla f_k \neq 0$
(2.5)    *find* $t_k$ *such that* $\min_{t_k} \|Q(X_k + t_k\mathcal{D}_k)\|_F^2$ *over all* $t_k \in \mathbb{R}$
         $X_{k+1} = X_k + t_k\mathcal{D}_k$
         $\mathcal{U}_{k+1} = -\nabla f_{k+1} + \beta_k\mathcal{D}_k$
**end**
```

The constant β_k in Algorithm 2.1 has two possible forms suggested by Fletcher and Reeves [6] and Polak and Ribiére [16], respectively,

$$\beta_k^{\mathrm{FR}} = \frac{\|\nabla f_{k+1}\|_F^2}{\|\nabla f_k\|_F^2}, \tag{2.6}$$

$$\beta_k^{\mathrm{PR}} = \frac{\mathrm{trace}((\nabla f_{k+1} - \nabla f_k)^T \nabla f_{k+1})}{\|\nabla f_k|_F^2}. \tag{2.7}$$

We now call the conjugate gradient method with β_k^{FR} the $\mathrm{CG}_{\mathrm{FR}}$ method and the conjugate gradient method with β_k^{PR} the $\mathrm{CG}_{\mathrm{PR}}$ method. Note that Algorithm 2.1 can be implemented with exact line searches for a step length t_k. Recalling the merit function for Newton's method with line searches and from $Q(X + t\mathcal{D}) = Q(X) + tD_X(\mathcal{D}) + t^2 A\mathcal{D}^2$ we have a quartic polynomial

$$\begin{aligned} p_{\mathrm{CG}}(t) &= \|Q(X + t\mathcal{D})\|_F^2 \\ &= a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0, \end{aligned}$$

where

$$\begin{aligned} a_4 &= \|A\mathcal{D}^2\|_F^2, \\ a_3 &= \mathrm{trace}(D_X(\mathcal{D})^T A\mathcal{D}^2 + (A\mathcal{D}^2)^T D_X(\mathcal{D})), \\ a_2 &= \mathrm{trace}(Q^T A\mathcal{D}^2 + (A\mathcal{D}^2)^T Q) + \|D_X(\mathcal{D})\|_F^2, \\ a_1 &= \mathrm{trace}(Q^T D_X(\mathcal{D}) + D_X(\mathcal{D})^T Q), \\ a_0 &= \|Q(X)\|_F^2. \end{aligned}$$

Since $p_{\mathrm{CG}}(t)$ is quartic and the coefficient a_4 is positive.

Finally, note that exact line searches always satisfy the equation

$$\operatorname{trace}((\nabla f_{k+1})^T\mathcal{D}_k) = \operatorname{vec}(\nabla f_{k+1})^T\operatorname{vec}(\mathcal{D}_k) = 0. \tag{2.8}$$

By applying vec to the equation (2.6) and premultiplying by $\operatorname{vec}(\nabla f_{k+1})^T$ we have

$$\operatorname{vec}(\nabla f_{k+1})^T\operatorname{vec}(\mathcal{D}_{k+1}) = -\|\nabla f_{k+1}\|_F^2 + \beta_k\operatorname{vec}(\nabla f_{k+1})^T\operatorname{vec}(\mathcal{D}_k)$$

and hence by the exact line search the condition (2.8) we have $\operatorname{vec}(\nabla f_{k+1})^T\operatorname{vec}(\mathcal{D}_{k+1}) < 0$, which means that $\mathcal{D}_{k+1}$ is a descent direction.

Now we consider operation counts for solving quadratic matrix equation using the conjugate gradient method and compare with Newton's method. Each step of Newton's method requires $102n^3$ flops using the generalized Schur decomposition and if Hessenberg-triangular decomposition is used then only $52n^3$ flops. For exact line searches we need $5n^3$ flops more [10], [11]. Each conjugate gradient method substep requires the evaluation of the gradient, ∇f, and the exact line searches. To evaluate the coefficients of p_{CG} ($A\mathcal{D}$, $A\mathcal{D}^2$, $A\mathcal{D}X$, $(AX+B)\mathcal{D}$, $D_X(\mathcal{D})^TA\mathcal{D}^2$, $Q^TA\mathcal{D}^2$, $Q^TD_X(\mathcal{D})$) 7 matrix multiplications are required. Note that we need the multiplication of A and X for computing $D_X(\mathcal{D})$, which appears in computing coefficients of p_{CG}, a_1, a_2 and a_4, but AX is already available from the evaluation of ∇f. There are three symmetric matrices so for these matrix multiplications we need n^3 flops each. Table 1 gives the operation counts for each substep of the conjugate gradient method . We can see that exact line searches for the conjugate gradient method requires $11n^3$ flops, which is relatively more expensive than for Newton's method. In some practical examples of quadratic matrix equation the coefficient matrices can be large and sparse, for instance, for the damped mass-spring system A is diagonal and B and C are symmetric tridiagonal. Suppose A, B and C are banded, however, only the matrix A affects reducing flops with bandedness during multiplications. Let the matrix A be banded with bandwidth ω. Then Table 2 shows the comparison of operation counts of banded and dense matrices.

From Table 2 if $\omega \ll n$ (which is a reasonable assumption because the damped mass-spring system gives only $\omega = 1$), we can save

$$\frac{19n^3 - 13n^3}{19n^3} = \frac{6n^3}{19n^3} = 32\%$$

than in dense case. So the conjugate gradient method with banded A can save

$$\frac{102n^3 - 13n^3}{102n^3} = \frac{89n^3}{102n^3} = 87\%$$

than Schur algorithm of Newton's method and

$$\frac{52n^3 - 13n^3}{52n^3} = \frac{39n^3}{52n^3} = 39\%$$

than Hessenberg-Schur algorithm of Newton's method. Although the bandedness is almost destroyed during computing ∇f and the coefficients of p_{CG} for exact line searches, the conjugate gradient method with banded coefficient matrix A saves operation counts significantly comparing with Newton's method.

Table 1: Number of flops for using conjugate gradient method.

		flops
∇f		$8n^3$ flops
Line search	$A\mathcal{D}^2$	$4n^3$ flops
	$D_X(\mathcal{D}) = A\mathcal{D}X + (AX + B)\mathcal{D}$	$4n^3$ flops
	$D_X(\mathcal{D})^T A\mathcal{D}^2 + (A\mathcal{D}^2)^T D_X(\mathcal{D})$	n^3 flops
	$Q^T A\mathcal{D}^2 + (A\mathcal{D}^2)^T Q$	n^3 flops
	$Q^T D_X(\mathcal{D}) + D_X(\mathcal{D})^T Q$	n^3 flops
Total		$19n^3$ flops

Table 2: Comparison of operation counts for with banded and dense coefficient matrices.

	Dense	Banded with bandwidth ω
∇f	$8n^3$	$4\mathcal{D}n^2 + 4n^3$
Line searches	$11n^3$	$2\mathcal{D}n^2 + 9n^3$
Total	$19n^3$	$6\mathcal{D}n^2 + 13n^3$

3. Global convergence

Powell [18] established the global convergence for the $\mathrm{CG_{FR}}$ method with the general nonlinear equation, $f_G : \mathbb{R}^n \rightarrow \mathbb{R}^n$, assuming exact line search. Using similar argument we can prove a global convergence result for the $\mathrm{CG_{FR}}$ method with the function $f(X) : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}$ in (2 1). We start with two results which play crucial roles in the proving of global convergence.

Lemma 3.1 [7, Thm. 4.1.1], [12]**.** *The descent directions $\mathcal{D}_k$ and ∇f in Algorithm 2.1 satisfy the following result:*

$$\frac{\mathrm{vec}(\nabla f_k)^T \mathrm{vec}(\mathcal{D}_k)}{\|\nabla f_k\|_F^2} = -1.$$

Lemma 3.2 [15, Thm. 3.2], [12]**.** *Consider Algorithm 2.1. Suppose that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set*

$$\mathcal{L} := \{X : f(X) \leq f(X_0)\}, \tag{3.1}$$

where X_0 is starting matrix. Assume also that there is a constant Ω such that

$$\|\nabla^2 f(X)\| \leq \Omega, \qquad \text{for all} \quad X \in \mathcal{L}.$$

Then

$$\sum_{k=0}^{\infty} \frac{\left(\mathrm{vec}(\nabla f_k)^T \mathrm{vec}(\mathcal{D}_k)\right)^2}{\|\mathcal{D}_k\|_F^2} < \infty. \tag{3.2}$$

Inequality (3.2) in Lemma 3.2 can be rewritten by

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f_k\|_F^2,$$

where

$$\cos \theta_k = \frac{-\mathrm{vec}(\nabla f_k)^T \mathrm{vec}(\mathcal{D}_k)}{\|\nabla f_k\|_F \|\mathcal{D}_k\|_F}$$

and it is called the Zoutendijk condition [19].

We now remind that if the level set $\mathcal{L}$ in (3.1) is bounded and the gradient ∇f satisfies a Lipschitz condition in an open set $\mathcal{N}$, which is in some neighbourhood of $\mathcal{L}$, that is, there exists a constant $L > 0$ such that

$$\|\nabla f(X) - \nabla f(\tilde{X})\|_F \leq L\|X - \tilde{X}\|_F, \qquad \text{for all} \quad X, \tilde{X} \in \mathcal{N}, \tag{3.3}$$

then there exists a constant γ such that

$$\|\nabla f(X)\|_F \leq \gamma, \qquad \text{for all} \quad X \in \mathcal{L}. \tag{3.4}$$

Theorem 3.3 [18, Sec. 4], [12]**.** *Suppose that all assumptions in Lemma 3.1. and 3.2. hold and the level set $\mathcal{L}$ in* (3.1) *is bounded. Assume also that f is Lipschitz continuously differentiable in some neighbourhood $\mathcal{N}$ of $\mathcal{L}$ and Algorithm is implemented by CG$_{FR}$ method. Then*

$$\lim_{k\to\infty} \inf \|\nabla f_k\|_F = 0. \tag{3.5}$$

We note that Theorem applies to any general nonlinear matrix equation. However, it may be too expensive to apply the exact line searches for the general nonlinear equation. So, the global convergence for conjugate gradient method with inexact line searches can be considered. The global convergence of the CG$_{\mathrm{FR}}$ method with inexact line searches was considered for the general nonlinear equation by Al-Baali [1] and we can apply this to the general nonlinear matrix equation. First, we consider the strong Wolfe conditions, which avoids $\mathrm{vec}(\nabla f_k)^T \mathrm{vec}(\mathcal{D}_k) > 0$:

$$f_{k+1} \leq f_k + c_1 t_k \nabla f_k^T \mathcal{D}_k, \tag{3.6a}$$

$$|(\mathrm{vec}(\nabla f_{k+1}))^T \mathrm{vec}(\mathcal{D}_k)| \leq c_2 |(\mathrm{vec}(\nabla f_k))^T \mathrm{vec}(\mathcal{D}_k)|, \tag{3.6b}$$

where $0 < c_1 < c_2 < \dfrac{1}{2}$. Lemmas 3.1. and 3.2. can be rewritten as follows.

Lemma 3.4. *Suppose that Algorithm is implemented with a step length t_k that satisfies the strong Wolfe conditions* (3.6)*. Then the method generates a descent direction $\mathcal{D}_k$ that satisfies the inequalities*

$$-\frac{1}{1-c_2} \leq \frac{\mathrm{vec}(\nabla f_k)^T \mathrm{vec}(\mathcal{D}_k)}{\|\nabla f_k\|_F^2} \leq \frac{2c_2 - 1}{1 - c_2}, \qquad \textit{for all} \quad k = 0, 1, \cdots.$$

Proof. See [1, Thm. 1] and [15, Lem. 5.6]. □

Lemma 3.5 [15, Thm. 3.2], [12]**.** *Suppose that Algorithm is implemented with a step length t_k that satisfies the strong Wolfe conditions* (3.6)*. Assume also that in some neighbourhood $\mathcal{N}$ of $\mathcal{L}$ in* (3.1) *f satisfies the Lipschitz condition* (3.3)*. Then*

$$\sum_{k=0}^{\infty} \frac{\left(\mathrm{vec}(\nabla f_k)^T \mathrm{vec}(\mathcal{D}_k)\right)^2}{\|\mathcal{D}_k\|_F^2} < \infty.$$

Finally, from two lemmas we have the global convergence for the $\mathrm{CG_{FR}}$ method with inexact line searches.

Theorem 3.6. *Suppose that all the assumptions of Lemma 3.4. and Lemma 3.5. hold, and that Algorithm 2.1 is implemented by the CG_{FR} method with a line search that satisfies the strong Wolfe conditions* (3.6)*. Then*

$$\lim_{k\to\infty} \inf \|\nabla f_k\|_F = 0.$$

Proof. See [1, Thm. 2] and [15, Thm. 5.8]. □

The rate of convergence of conjugate gradient methods was consider by Crowder and Wolfe [3] with exact line searches. They considered the two-dimensional quadratic function

$$f_V(x) = \frac{1}{2} x^T V x,$$

where V is an $n \times n$ symmetric positive definite matrix. They obtained the rate of convergence with conjugate gradient method

$$f_V(x_{k+1})/f_V(x_k) \leq [(\mathcal{A}-1)/(\mathcal{A}+1)]^2,$$

where $\mathcal{A}$ is the condition number of the matrix V. It shows that the convergence of the conjugate gradient method is linear. For this reason it is desirable to consider preconditioning for improving the ratio.

5. Using `nleqn` based on hybrid method

If we consider the function f in (2.1) as n^2 nonlinear equations by defining $r(x) = \mathrm{vec}(Q(X))$ with n^2 variables, which are x_{ij} for $i, j = 1, 2, \cdots, n$, we can use minimization methods based on the Gauss-Newton and the Levenberg-Marquardt methods. For the Gauss-Newton method and the Levenberg-Marquardt method the direction d_k can be obtained by solving

$$J_k^T J_k d_k = -J_k^T r_k,$$

where J_k is the Jacobian of $r(x)$

$$J(x) = \left(\frac{\delta r_j}{\delta x_i}\right) \quad \text{for } i, j = 1, \cdots n^2.$$

A Matlab function, `nleqn`, for solving a system of m nonlinear equation in m variables using the Gauss-Newton method and the Levenberg-Marquardt method was implemented by Reichelt and Shampine. The function `nleqn` is based on the Fortran program HYBRD1 of More, Garbow and Hillstrom which is originally based on the program CALFUN of Powell [17, Chap. 7].

But these method needs to evaluate the Jacobian matrix which is $n^2 \times n^2$ and it means that for large n this method is not very suitable.

6. Numerical Experiments and Conclusions

In this section we show and compare some experimental results the CG$_{\mathrm{FR}}$ method, CG$_{\mathrm{PR}}$ method and a MATLAB function for solving general nonlinear equation, `nleqn`. Our experiments were done in MATLAB.

Our default starting matrix is, as in [4],

$$\text{Default\_}X_0 = \left(\frac{\|B\|_F + \sqrt{\|B\|_F^2 + 4\|A\|_F\|C\|_F}}{2\|A\|_F} \right) I,$$

which is designed to have norm roughly of the same order of magnitude as a solvent. The first plus sign avoids the starting value $X_0 = 0$ and the second plus sign is to avoid the possibility of a complex X_0. Iterations for CG$_{\mathrm{FR}}$ and CG$_{\mathrm{PR}}$ methods are terminated when the residual $Q(X_k)$ is of the same order of magnitude as the rounding error in computing it, namely when the relative residual $\rho(X_k)$ satisfies

$$\rho(X_k) = \frac{\|fl(Q(X_k))\|_F}{\|A\|_F\|X_k\|_F^2 + \|B\|_F\|X_k\|_F + \|C\|_F} \le nu, \tag{6.1}$$

where $u = 2^{-53} \simeq 1.1 \times 10^{-16}$ is unit roundoff. However, the iteration for `nleqn` is terminated with tolerance

$$\|\mathrm{vec}(Q(X_k))\|_2 = \|Q(X_k)\|_F \le 1.0 \times 10^{-6} \tag{6.2}$$

and maximum function evaluations $200 \times (n^2 + 1)$.

The first example is

$$Q_1(X) = X^2 + X + \begin{bmatrix} -6 & -5 \\ 0 & -6 \end{bmatrix} = 0, \tag{6.3}$$

from [4]. With starting matrices I_2 and the default starting matrix Default_X_0 CG$_{\mathrm{FR}}$ and CG$_{\mathrm{PR}}$ methods, and `nleqn` converge to the solvent, $S_1 = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$. Figure 1 illustrates that the CG$_{\mathrm{PR}}$ method gives faster convergence than the CG$_{\mathrm{FR}}$ method with both starting matrices, I_2 and Default_X_0.

We now consider the quadratic matrix equation

$$Q_2(X) = X^2 + \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix} X + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = 0, \tag{6.4}$$

which has two real solvents

$$S_1 = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{6.5}$$

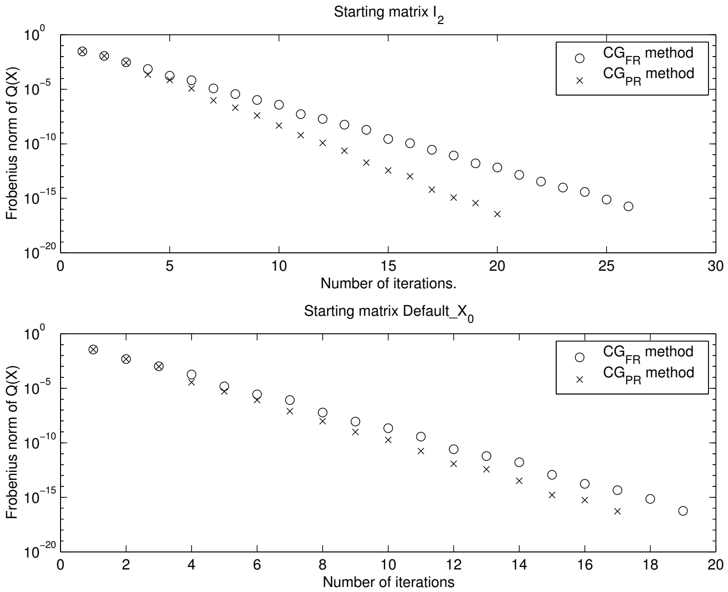

Figure 1: Convergence for problem (6.3) with CG_{FR} and CG_{PR} methods.

and infinitely many complex solvents, which have the forms

$$S_3 = \begin{bmatrix} -z-i+1 & i(z-1) \\ iz-1 & z \end{bmatrix}, \qquad S_4 = \begin{bmatrix} -z+i+1 & -i(z-1) \\ -iz-1 & z \end{bmatrix} \tag{6.6}$$

for all $z \in \mathbb{C}$. Applying our methods with the default starting matrix Ddfault_X_0 and $X_0 = 10^j I$, $j = 1, 5, 10$, gives the result in Tables 3. Note that `nleqn` does not converge with $X_0 = 10^{10} I$.

Finally, a practical examples is considered. It is based on a quadratic equation problem from [8, Sec. 10.11], with numerical values modified as in [13, Sec. 5.3], modelling

Table 3: Number of iterations for convergence for problem (6.4) using minimization methods.

X_0	CG_{FR}	CG_{PR}	nleqn
DdfaultX_0	17	7	9
$10I$	83	8	17
$10^5 I$	34	8	48
$10^{10} I$	39	10	-

oscillations in an aeroplane wing:

$$(6.7a)\qquad A = \begin{bmatrix} 17.6 & 1.28 & 2.89 \\ 1.28 & 0.824 & 0.413 \\ 2.89 & 0.413 & 0.725 \end{bmatrix}, \qquad B = \begin{bmatrix} 7.66 & 2.45 & 2.1 \\ 0.23 & 1.04 & 0.223 \\ 0.6 & 0.756 & 0.658 \end{bmatrix},$$

$$(6.7b)\qquad C = \begin{bmatrix} 121 & 18.9 & 15.9 \\ 0 & 2.7 & 0.145 \\ 11.9 & 3.64 & 15.5 \end{bmatrix}.$$

There are no real solvents because the 6 eigenvalues come in 3 complex conjugate pairs and any solvent must have 3 eigenvalues chosen from the 6. $\mathrm{CG}_{\mathrm{FR}}$ and $\mathrm{CG}_{\mathrm{PR}}$ methods need over 1000 iterations with same starting matrices. All methods including `nleqn` converge to the same solvent and the eigenvalues of the computed solvent are

$$\begin{aligned} &\texttt{-8.8483e-001} + \texttt{8.4415e+000i}, \\ &\texttt{9.4722e-002} + \texttt{2.5229e+000i}, \\ &\texttt{-9.1800e-001} + \texttt{1.7606e+000i}. \end{aligned}$$

Their conjugates are the eigenvalues of the quadratic equation problem.

We showed that the minimization can be used to solve the quadratic matrix equations. Two different types of conjugate gradient method a which are Polak and Ribiére version and Fletcher and Reeves version were introduced. Finally, finding a suitable preconditioner without too much expense remains an open problem.

Acknowledgements
This work is based on the Ph.D. thesis [12] so I would like to thank Professor Nicholas J. Higham for introducing for many very helpful discussions and comments on this work.

References

[1] Mehiddin Al-Baali, *Descent property and global convergence of the Fletcher-Reeves method with inexact line search*, IMA J. Numer. Anal., **5**(1985), 121-124.

[2] King-wah Eric Chu, *The solution of the matrix equation $AXB - CXD = E$ and $(YA - DZ, YC - BZ) = (E, F)$*, Linear Algebra and Appl., **93**(1987), 93-105.

[3] H. P. Crowder and P. Wolfe. *Linear convergence of the conjugate gradient method*, IBM Journal of Research and Development, **16**(1972), 431-433.

[4] George J. Davis. *Numerical solution of a quadratic matrix equation*, SIAM J. Sci. Stat. Comput., **2(2)**(1981), 164-175, 1981.

[5] J. E. Dennis, Jr. and Robert B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983, ISBN 0-13-627216-9.

[6] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Computer J., **7**(1964), 149-154.

[7] Roger Fletcher, Practical Methods of Optimization, Wiley, Chichester, UK, 2nd edition, 1987, ISBN 0-471-91547-5.

[8] R. A. Frazer, W. J. Duncan, and A. R. Collar, Elementary Matrices and Some Applications to Dynamics and Differential Equations, Cambridge University Press, 10th edition, 1938, 1963 printing.

[9] Philip E. Gill, Walter Murray, and Margaret H. Wright, Practical Optimization, Academic Press, London, 1981, ISBN 0-12-283952-8 (paperback).

[10] Nicholas J. Higham and Hyun-Min Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., **20(4)**(2000), 499-519.

[11] Nicholas J. Higham and Hyun-Min Kim, *Solving a quadratic matrix equation by Newton's method with exact line searches*, SIAM J. Matrix Anal. Appl., **23(2)**(2001), 303-316.

[12] Hyun-Min Kim, Numerical methods for solving a quadratic matrix equation, Ph.D. Thesis, University of Manchester, 2000.

[13] Peter Lancaster, Lambda-Matrices and Vibrating Systems, Pergamon Press, Oxford, 1966, xiii+196 pp.

[14] Jorge Nocedal, Conjugate gradient methods and nonlinear optimization In Loyce Adams and J. L. Nazareth, editors, Linear and Nonlinear Conjugate Gradient-Related Methods, pages 9-23, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

[15] Jorge Nocedal and Stephen J. Wright, Numerical Optimization, Springer-Verlag, New York, 1999, xx+636 pp, ISBN 0-387-98793-2.

[16] E. Polak and G. Ribiére, *Note sur la convergence des méthodes de directions conjugées*, Revue Fr. Inf. Rech. Oper., **16(R1)**(1969), 35-43.

[17] M. J. D. Powell, A Fortran subroutine for solving systems of nonlinear algebraic equations, In Philip Rabinowitz, editor, Numerical Methods for Nonlinear Algebraic Equations, pages 115-161, London, 1970, Gordon and Breach Science.

[18] M. J. D. Powell. Nonconvex minimization calculations and the conjugate gradient method, In D. F. Griffiths, editor, Numerical Analysis, Dundee 1983, pages 122-141, Lecture Notes in Mathematics, 1066, Springer, Berlin-New York, 1984.

[19] G. Zoutendijk, Nonlinear programming, computational methods, In J. Abadie, editor, Integer and Nonlinear Programming, pages 37-86, North-Holland, Amsterdam, 1970.