World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:5, No:8, 2011

A Fast Cyclic Reduction Algorithm for A Quadratic Matrix Equation Arising from Overdamped Systems

Ning Dong and Bo Yu

Abstract—We are concerned with a class of quadratic matrix equations arising from the overdamped mass-spring system. By exploring the structure of coefficient matrices, we propose a fast cyclic reduction algorithm to calculate the extreme solutions of the equation. Numerical experiments show that the proposed algorithm outperforms the original cyclic reduction and the structure-preserving doubling algorithm.

Keywords—Fast algorithm, Cyclic reduction, Overdamped quadratic matrix equation, Structure-preserving doubling algorithm

I. Introduction

In a quadratic eigenvalue problem (QEP) [17], the eigenvalues λ and eigenvectors x are to be find to satisfy

$$Q(\lambda)x = (\lambda^2 M + \lambda D + K)x = 0, \quad M, D, K \in \mathbb{C}^{n\times n}. \quad (1)$$

In this paper, we are interested in a QEP arising from the mass-damped system, that is,

$$\begin{aligned} M &= mI_n, \\ D &= P_n \text{diag}(d, \ldots, d, 0) P_n^T + \tau I_n, \\ K &= P_n \text{diag}(k, \ldots, k, 0) P_n^T + \kappa I_n \end{aligned} \quad (2)$$

and $P_n = (\delta_{ij} - \delta_{i,j+1})_{i,j=1}^n$ with δ_{ij} the Kronecker delta, i.e. $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$. The parameters in (2) have the following physical meaning. The i-th mass weight m is connected to its $(i+1)$-th neighbour by a spring and a damper with constants k and d, respectively. The i-th mass is also connected to the ground by a spring and a damper with constants κ and τ, respectively. We refer to [16] for more details.

The overdamped case of system (1) often need to be detected in many applications and has the following definition [6].

Definition 1.1. *If $M > 0$, $D > 0$, $K \geq 0$ and $D > \mu M + \mu^{-1} K$ for some $\mu > 0$, the QEP (1) from mass-spring system is called overdamped.*

Here and hereafter, the matrix inequality $M_1 \geq M_2 (M_1 > M_2)$ for Hermitian matrices M_1 and M_2 means that matrix $M_1 - M_2$ is positive semidefinite (definite).

Guo and Lancaster [7] recently showed that the overdamping condition can be checked efficiently by computing two eigenvalues of the extremal solutions[1] of the quadratic matrix equation (QME)

$$Q(S) = MS^2 + DS + K = 0 \quad (3)$$

with $M, D, K \in \mathbb{R}^{n\times n}$ the same as those in (2). Therefore, the detection of a overdamped system (1) relies on obtaining the extremal solution of (3) efficiently.

Generally, the fixed-point methods [11], Newton's method [11], [12], cyclic reduction (CR) [2] and different structure-preserving doubling algorithms (SDA) [4, 14, 18–20] are efficient algorithms for computing the extreme solutions of the QME (3). Although they all share the complexity of $O(n^3)$ flops per iteration, fixed-point methods are linearly convergent and latter three in general provide quadratic convergence. Moreover, the CR algorithm and the SDA algorithm are considered more effective than Newton's method since a matrix decomposition and several matrices multiplications are only required [14], [20].

In this paper, we reconsider the CR algorithm applied into (3). By taking full advantage of the special structure of coefficient matrices in (2), we extend them to a class of centrosymmetric Toeplitz-plus-Hankel $(T + H)$ matrices, and then devise an algorithm for performing the CR iteration with $O(n^2)$ flops per step. This algorithm is based on a suitable modification of the fast inverse formula for $T + H$ matrices developed in [10]. The numerical experiments show that the proposed fast CR algorithm outperforms the CR algorithm [2] and the SDA [4].

The rest of this paper is organized as follows. We extend the coefficient matrices of the overdamped QME (3) to a class of centrosymmetric $T + H$ matrices in the next section. In Section 3, we review the CR algorithm in [2], [6] and develop a fast CR algorithm based on the recursively fast inverse formula. We obtain a similar fast inverse formula for another class of overdamped QMEs in Section 4. Section 5 is devoted to test the proposed algorithm and compare its numerical performance with the CR algorithm [2], [6] and the SDA [4]. We conclude the paper by discussion in Section 6.

II. Preliminaries

In this section, we do some preliminaries. We first introduce some properties on $T + H$ matrices. Let $T = (t_{i-j})_{i,j=1}^n$ and $H = (h_{i+j-2})_{i,j=1}^n$ be Toeplitz matrix and Hankel matrix,

N. Dong is with the College of Mathematics and Econometrics, Hunan University, Changsha, 410082, P. R. China. Email:dnsx216@tom.com

Corresponding author. B. Yu is with the College of Mathematics and Econometrics, Hunan University, Changsha, 410082, P. R. China. Email:wenyubwenyub@yahoo.com.cn, boyu@hun.edu.cn

[1]The extreme solutions are two solutions $S^{(1)}$ and $S^{(2)}$ which have as their eigenvalues the n largest eigenvalues and the n smallest eigenvalues in the corresponded quadratic eigenvalue problems [7].

respectively. The next lemma gives an equivalent description of $T+H$ matrix [9].

Lemma 2.1. *A matrix M is a $T+H$ matrix if and only if the $(n-2)\times(n-2)$ submatrix in the center of the matrix $\nabla_{W_n}(M) = W_nM - MW_n$ is the zero matrix, where* $W_n = \begin{bmatrix} 1 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & 1 \\ & & & 1 & 1 \end{bmatrix}$.

Note that the matrix W_n such that the $(n-2)\times(n-2)$ submatrix of $\nabla_{W_n}(M)$ is zero matrix is not unique [9]. We use W_n of the form in Lemma 2.1 so that it is convenient to derive a fast inverse formula in the next section.

It follows from Lemma 2.1 that the $T+H$ matrix N satisfies the following displacement structure (see also [10], [9])

$$W_nN - NW_n = GF, \tag{4}$$

where $G = (g_1, g_2, g_3, g_4)$ and $F = (f_1, f_2, f_3, f_4)^T$ with $g_i, f_i \in \mathbb{R}^n$, $i = 1, 2, 3, 4$. That is, the W_n displacement rank of $T+H$ matrix does not exceed 4. Specially, if the RHS of (4) equals zero matrix, the matrix M is called *W_n-commutable $T+H$ matrix*. The following lemma describes the W_n-commutable $T+H$ matrix in the component-wise sense.

Lemma 2.2. *Let $M = T + H = (t_{ij})_{i,j=1}^n + (h_{i+j-2})_{i,j=1}^n$. M is a W_n-commutable $T+H$ matrix if and only if*

$$\begin{cases} t_1 = t_{-1}, \quad h_n = h_{n-2}, \\ t_{1-i} + h_{i-1} = t_{-i} + h_{i-2}, \\ t_{i-1} + h_{2n-1-i} = t_i + h_{2n-i}, \quad i = 2, \cdots, n-1. \\ t_{i-1} + h_{i-1} = t_i + h_{i-2}, \\ t_{1-i} + h_{2n-1-i} = t_{-i} + h_{2n-i}, \end{cases} \tag{5}$$

Proof. It follows from Lemma 2.1 that the nonzero entries of M can only appear in its first and last rows and its first and last columns. Direct computations and the definition of W_n-commutable $T+H$ matrix complete the proof. □

Denote by $J = (\delta_{i,n+1-j})_{i,j=1}^n$ the counter-identity matrix. A $T+H$ matrix M is called *centrosymmetric* if $JMJ = M$. Evidently, if the Toeplitz part of the matrix M is symmetric and the Hankel part of M is persymmetric[2], then M is a centrosymmetric $T+H$ matrix, but the inversion is not true.

The following lemma indicates some interesting properties about (cenrtosymmetric) W_n-commutable $T+H$ matrices.

Lemma 2.3. *(a) Let $A \in \mathbb{R}^{n\times n}$ be nonsingular, A is a (centrosymmetric) W_n-commutable $T+H$ matrix if and only if A^{-1} is a (centrosymmetric) W_n-commutable $T+H$ matrix.*

(b) Let A, B, C and $D \in \mathbb{R}^{n\times n}$ be (centrosymmetric) W_n-commutable $T+H$ matrices, then $AB+CD$ is a (centrosymmetric) W_n-commutable $T+H$ matrix.

Proof. Since the inverse, product and sum of centrosymmetric matrices are all centrosymmetric, the conclusion is direct by the definition of W_n-commutable $T+H$ matrix. □

[2]A matrix $A \in \mathbb{R}^{n\times n}$ is persymmetric if it symmetric about its northeast-to-southwest diagonal[5].

The above lemma shows that the inverse, product, sum and their combinations of W_n-commutable $T+H$ matrices are still W_n-commutable $T+H$ matrices. This structure-preserving property provides the base for designing a fast CR algorithm for QME (6) in the next section.

III. Fast CR algorithm

A. The CR algorithm

The cyclic reduction (CR) algorithm is a very efficient algorithm for solving some nonlinear matrix equations (see [2], [15] for example). Attractive properties of the CR algorithm include its quadratic convergence rate, low computational cost per iteration and nice numerical reliability. The general iteration scheme for finding the extremal solutions of QME

$$Q(X) = AX^2 + BX + C = 0 \tag{6}$$

is as follows:

Algorithm 3.1. The Cyclic Algorithm.

step 0 : $S_0 = B,\ A_0 = A,\ B_0 = B,\ C_0 = C.$
step 1 : *For* $k = 0, 1, 2 \ldots,$ *until convergence, do*
$$\begin{aligned} S_{k+1} &= S_k - A_kB_k^{-1}C_k, \\ A_{k+1} &= A_kB_k^{-1}A_k, \\ B_{k+1} &= B_k - A_kB_k^{-1}C_k - C_kB_k^{-1}A_k, \\ C_{k+1} &= C_kB_k^{-1}C_k. \end{aligned}$$

When the QEP corresponding to the QME (6) is a overdamped, Guo, Higham and Tisseur [6] showed that the matrix inequalities

$$A_k > 0, \quad C_k \geq 0, \quad B_k \geq \mu^{2^k}A_k + \mu^{-2^k}C_k, \quad k \geq 0 \tag{7}$$

hold with some positive real constant μ. Thus the iterations (7) are well defined.

The convergence of iteration (7) in this case are summarized as follows [6, Theorem 3.1, Corollary 4.7].

Theorem 3.1. *Let QME (6) be overdamped. Consider the iteration (7). We have*

(a) $\|A_k\|\|C_k\|$ converges quadratically to zero with any matrix norm $\|\cdot\|$.

(b) $\{S_k\}$ converges quadratically to a nonsingular matrix $\hat{S}$. Moreover, the two extreme solutions of QME (6) are given by

$$S^{(1)} = -\hat{S}^{-1}C, \quad S^{(2)} = -A^{-1}\hat{S}^T. \tag{8}$$

(c) $\{B_k\}$ converges quadratically to a nonsingular matrix $\hat{B} = A(S^{(1)} - S^{(2)})$.

As far as the complexity at each iteration is concerned, Algorithm 3.1 requires one matrix inverse and several matrix multiplications. The computational cost can be specified as follows [7]. Let

$$B_k = L_kL_k^T \tag{9}$$

be the Cholesky factorization. Let

$$V_k = L_k^{-1}A_k \ \text{ and } \ U_k = L_k^{-1}C_k. \tag{10}$$

Then $A_kB_k^{-1}C_k = V_k^TU_k$, $C_kB_k^{-1}A_k = (A_kB_k^{-1}C_k)^T$, $A_kB_k^{-1}A_k = V_k^TV_k$, and $C_kB^{-1}C_k = U_k^TU_k$. Therefore, the computational work required for one iteration is about $19n^3/3$.

B. The fast algorithm

In this subsection, we shall give a fast implementation of Algorithm 3.1 by the use of special structures of coefficient matrices in (2).

It is not difficult to see that coefficient matrices M, D and K in (2) fall in a class of $T+H$ matrices

$$R=\Big(r_{|i-j|}+\begin{cases} r_{i+j-1} & i+j\le n+1,\\ r_{2n+1-i-j} & i+j>n+1,\end{cases}\Big)_{i,j=1}^{n}. \quad (11)$$

In fact, we have $R=M$ with $r_0=m, r_1=\cdots=r_n=0$, $R=D$ with $r_0=2d+\tau, r_1=-d, r_2=\cdots=r_n=0$ and $R=K$ with $r_0=2k+\kappa, r_1=-k, r_2=\cdots=r_n=0$.

Note that the Toeplitz part and the Hankel part of the matrix R are symmetric and persymmetric, respectively, thus by Lemma 2.2, R is a centrosymmetric W_n-commutable $T+H$ matrix. It is well known that the inverse and the Schur complement of displacement structure matrices can preserve the low rank property [13], so does R.

Let $A=M$, $B=D$ and $C=K$ in Algorithm 3.1, It follows from Lemma 2.3 that iteration sequences $\{A_k\}$, $\{B_k\}$, $\{C_k\}$ and $\{S_k\}$ are all centrosymmetric W_n-commutable $T+H$ matrices and, thereby, the limit matrices $\hat{S}$ and $\hat{B}$ (hence the extreme solutions $S^{(1)}$ and $S^{(2)}$) are centrosymmetric W_n-commutable $T+H$ matrices due to Theorem 3.1. With this observation, we can develop a fast algorithm with $O(n^2)$ to fulfill the CR iteration as follows.

Algorithm 3.2. Fast cyclic reduction algorithm.

step 0: $S_0=B,\ A_0=A,\ B_0=B,\ C_0=C.$
step 1: For $k=0,1,2\ldots,$ *until convergence, do*
1.1. *Compute fast inverse* B_k^{-1} *with* $O(n^2)$.
1.2. *Compute products* $A_kB_k^{-1}C_k,\ A_kB_k^{-1}A_k$ *and* $C_kB_k^{-1}C_k$ *with* $O(n^2)$.
1.3. *Fulfill step 1 in Algorithm 3.1.*

It is clear that the fast implementation of Algorithm 3.2 depends on the fast inversion of matrix B_k. Moreover, we notice that each B_k is of the form (11), so that the fast inversion can be done by a suitable modification of the algorithm proposed by Heinig, Jankowski and Rost [10]. Let R be nonsingular, from (4) one obtains

$$W_nR^{-1}-R^{-1}W_n=-\sum_{i=1}^{4}x_iy_i^T,$$

where x_i, y_i are the solutions of equations

$$Rx_i=g_i,\quad R^Ty_i=f_i\quad (i=1,2,3,4).$$

Since R is a centrosymmetric $T+H$ matrix, the above equations are reduced to

$$Rx_1=e_1,\quad Rx_2=e_n,\quad Rx_3=Rx_4=0 \qquad (12)$$

and

$$y_1=x_3,\quad y_2=x_4,\quad y_3=x_1,\quad y_4=x_2,$$

where e_1 and e_n are the first and the last column of identity matrix I_n.

The following lemma gives the inverse formula of the matrix R by using the solution in (12).

Lemma 3.2. *Let R in (11) be nonsingular. Then the columns $u_j(j=1,\ldots,n)$ of R^{-1} can be determined by the solution of $Rx_2=e_n$ and the recursion*

$$u_n=x_2,\ \ u_{j-1}=\begin{cases} W_nu_n-u_n, & j=n\\ W_nu_j-u_{j+1}, & 2\le j\le n-1\end{cases} \qquad (13)$$

Proof. We prove the lemma by induction. It is clear that $u_n=x_2$ for $j=n$. Assume that $Ru_j=e_j(j\le n)$, where e_j is the j-th column of the identity matrix. By $W_nR-RW_n=0$, we have

$$W_ne_j-RW_nu_j=0.$$

This together with

$$W_ne_j=\begin{cases} e_{n-1}+e_n, & j=n\\ e_{j-1}+e_{j+1}, & 2\le j\le n-1\end{cases}$$

yields $R(W_nu_n-u_n)=e_{n-1}$ and $R(W_nu_j-u_{j+1})=e_{j-1}(2\le j\le n)$. The proof is complete. □

The computation of formula (13) is $2n^2$. Since R^{-1} is centrosymmetric, the cost can be reduced to $n^2/2$.

To complete the computation of R^{-1}, we need to solve the equation $Rx_2=e_n$ in (12) with $O(n^2)$. This can be done by a recursion procedure. Rewrite R in (11) as $R=(r_{|i-j|}+r_{i+j-1})_{i,j=1}^{n}$ and consider the sequence of principal sections of order m of R

$$R^{(m)}=(r_{|i-j|}+r_{i+j-1})_{i,j=1}^{m}\quad (1\le m\le n).$$

Let $u^{(m)}$ and $v^{(m)}$ be vectors with the dimension m such that

$$R^{(m)}v^{(m)}=-g^{(m)},\quad R^{(m)}u^{(m)}=e_m^{(m)}, \qquad (14)$$

where

$$\begin{aligned} g^{(m)} &= (r_m+r_{m+1}-r_{m-1}-r_m,\\ &\quad r_{m-1}+r_{m+2}-r_{m-2}-r_{m+1},\ldots,\\ &\quad r_1+r_{2m}-r_0-r_{2m-1})^T\in\mathbb{R}^m \end{aligned} \qquad (15)$$

and $e_m^{(m)}$ is the last column of identity matrix of the dimension m.

Lemma 3.3. *Assume R to be strongly nonsingular. Then the solutions $v^{(m)}$ of (14) have the recursion*

$$\begin{aligned} v^{(m+1)} &= \Big(W^{(m+1)}-(1+\frac{\gamma_m}{\delta_m}-\frac{\lambda_m}{\delta_{m-1}})I_{m+1}\Big)\\ &\cdot\begin{bmatrix} v^{(m)}-e_m^{(m)}\\ 1\end{bmatrix}-\frac{\delta_m}{\delta_{m-1}}\begin{bmatrix} v^{(m-1)}-e_{m-1}^{(m-1)}\\ 1\\ 0\end{bmatrix}, \end{aligned} \qquad (16)$$

where

$$\delta_m=(f^{(m)})^T(v^{(m)}-e_m^{(m)})+r_0+r_{2m+1}\neq 0,$$

$$\gamma_m=(g^{(m+1)})^T\begin{bmatrix} v^{(m)}-e_m^{(m)}\\ 1\end{bmatrix},$$

$$\lambda_m=(f^{(m+1)})^T\begin{bmatrix} v^{(m)}-e_m^{(m)}\\ 1\end{bmatrix}$$

with

$$f^{(m)}=(r_m+r_{m+1},r_{m-1}+r_{m+2},\ldots,r_1+r_{2m})^T\in\mathbb{R}^m$$

and $W^{(m)} \in \mathbb{R}^{m\times m}$ has the same structure with W_n in Lemma 2.1.

Proof. The definition of δ_m implies it is nonzero, otherwise $(v^{(m)}-e_m^{(m)},1)^T$ would be a nontrivial vector of the kernel of $R^{(m+1)}$. It follows from the equality $g^{(m)}=f^{(m)}-R^{(m)}e_m^{(m)}$ that

$$R^{(m+1)}\begin{bmatrix} v^{(m)}-e_m^{(m)} \\ 1 \end{bmatrix}=\begin{bmatrix} 0^{(m)} \\ \delta_m \end{bmatrix} \tag{17}$$

and

$$R^{(m+1)}\begin{bmatrix} v^{(m-1)}-e_{m-1}^{(m-1)} \\ 1 \\ 0 \end{bmatrix}=\begin{bmatrix} 0^{(m-1)} \\ \delta_{m-1} \\ \lambda_{m-1} \end{bmatrix}. \tag{18}$$

Rearranging (17) and (18) yields

$$\begin{aligned} & R^{(m+1)}(1-\frac{\lambda_{m-1}}{\delta_{m-1}})\begin{bmatrix} v^{(m)}-e_m^{(m)} \\ 1 \end{bmatrix} \\ & +\ R^{(m+1)}\frac{\delta_m}{\delta_{m-1}}\begin{bmatrix} v^{(m-1)}-e_{m-1}^{(m-1)} \\ 1 \\ 0 \end{bmatrix} \\ = & \begin{bmatrix} 0^{(m-1)} \\ \delta_m \\ \delta_m \end{bmatrix}. \end{aligned} \tag{19}$$

On the other hand, we have the following displacement equation

$$\begin{aligned} & W^{(m+1)}R^{(m+1)}-R^{(m+1)}W^{(m+1)} \\ = & g^{(m+1)}(e_{m+1}^{(m+1)})^T-e_{m+1}^{(m+1)}(g^{(m+1)})^T \end{aligned} \tag{20}$$

with $g^{(m)}$ defined in (15). A postmultiplying (20) by $\begin{bmatrix} v^{(m)}-e_m^{(m)} \\ 1 \end{bmatrix}$ yields

$$\begin{aligned} & R^{(m+1)}\left\{(W^{(m+1)}-\frac{\gamma_m}{\delta_m}I_m)\begin{bmatrix} v^{(m)}-e_m^{(m)} \\ 1 \end{bmatrix}\right\} \\ & -\begin{bmatrix} 0^{(m-1)} \\ \delta_m \\ \delta_m \end{bmatrix} \\ = & -g^{(m+1)}. \end{aligned}$$

This together with (19) completes the proof. □

Lemma 3.3 and (17) directly yield the following theorem.

Theorem 3.4. *The solution of linear system $Rx_2=e_n$ in (12) is given by*

$$x_2=\frac{1}{\delta_{n-1}}\begin{bmatrix} v^{(n-1)}-e_{n-1}^{(n-1)} \\ 1 \end{bmatrix}. \tag{21}$$

The computational cost of the formula (21) is about $13n^2/2$. Thus with (13) and (21), the fast inverse of B_k^{-1} in step 1.2 of Algorithm 3.2 can be derived in about $7n^2$.

We now turn to the products in step 1.3 of Algorithm 3.2. Consider two matrices R_1 and R_2 of the structure (11), their product can be obtained as follows.

1. Compute the last column of the product R_1R_2. The cost is $2n^2$.

2. The elements of R_1R_2 (denote by r_{ij}) with the subscript satisfying $i+j\geq n+1$ and $j\geq i$ can be recovered by

$$r_{ij}=r_{i-1,j+1}+r_{i+1,j+1}-\begin{cases} r_{i,j+1}, & j=n-1 \\ r_{i,j+2}, & j<n-1 \end{cases}. \tag{22}$$

The cost is $n^2/2$.

3. The remainder elements in R_1R_2 can be recovered by the symmetry and persymmetry.

With the above scheme, the computational cost of products $A_kB_k^{-1}C_k$, $A_kB_k^{-1}A_k$ and $C_kB_k^{-1}C_k$ in step 1.3 of Algorithm 3.2 is about $25n^2/2$. Hence the whole complexity of fast CR algorithm per step is about $45n^2/2$.

IV. Another special overdamped QME

In this section, we consider another special case as an example in [16]. The springs (dampers) connect each mass to its neighbor and to the ground have the same constant κ (τ), except the first and last ones for which $\kappa_1=\kappa_n=2\kappa$ ($\tau_1=\tau_n=2\tau$). It is easy to see that such coefficient matrices fall in another class of $T+H$ matrices

$$\tilde{R}=\tilde{r}_{|i-j|}-\begin{cases} \tilde{r}_{i+j} & i+j\leq n+1 \\ \tilde{r}_{2n+2-i-j} & i+j>n+1 \end{cases}.$$

Let $\tilde{W}_n=(\delta_{i+1,j}+\delta_{i,j+1})_{i,j=1}^n\in\mathbb{R}^{n\times n}$. We can similarly define the *$\tilde{W}_n$-commutable* matrix if a matrix M satisfies $M\tilde{W}_n=\tilde{W}_nM$. Analogous to the Lemma 2.2, the following result gives the equivalent conditions of a $\tilde{W}_n$-commutable $T+H$ matrix.

Lemma 4.1. *Let $M=T+H=(t_{ij})_{i,j=1}^n+(h_{i+j-2})_{i,j=1}^n$. M is a $\tilde{W}_n$-commutable $T+H$ matrix if and only if*

$$\begin{cases} t_1=t_{-1}, \quad h_n=h_{n-2} \\ -t_i=-t_{-i}=h_{i-2}=h_{2n-i}, \quad i=2,\cdots,n-1. \end{cases} \tag{23}$$

Different with W_n-commutable $T+H$ matrix, a $\tilde{W}_n$-commutable $T+H$ matrix is definitely centrosymmetric. Indeed, Lemma 4.1 shows that the Toeplitz part and the Hankel part of a $\tilde{W}_n$-commutable $T+H$ matrix is symmetric and persymmetric, respectively. Thus it is centrosymmetric.

By Lemma 4.1, $\tilde{R}$ defined in (23) is a $\tilde{W}_n$-commutable $T+H$ matrix. Following the same way with the above section, we can similarly develop a fast CR algorithm which is based on the next fast inverse formula. Since the proof is similar to that of Lemma 3.2 and Lemma 3.3, we omit it.

Lemma 4.2. *Let $\tilde{R}$ in (23) be nonsingular. Then the columns $\tilde{u}_j(j=1,\ldots,n)$ of $\tilde{R}^{-1}$ can be determined by the solution of $\tilde{R}\tilde{u}_n=e_n$ and the recursion*

$$\tilde{u}_{j-1}=\tilde{W}_n\tilde{u}_j-\tilde{u}_{j+1}, \quad 2\leq j\leq n. \tag{24}$$

Theorem 4.3. *The solution $\tilde{u}_n$ in Lemma 4.2 can be obtained by*

$$\tilde{u}_n=\frac{1}{\tilde{\delta}_{n-1}}\begin{bmatrix} \tilde{v}^{(n-1)} \\ 1 \end{bmatrix}, \tag{25}$$

where $\tilde{v}^{(n-1)} \in \mathbb{R}^{n-1}$ *has the recursion*

$$\tilde{v}^{(m+1)} = \left(\tilde{W}^{(m+1)} + (\frac{\tilde{\lambda}_{m-1}}{\tilde{\delta}_{m-1}} - \frac{\tilde{\lambda}_m}{\tilde{\delta}_m})I_{m+1}\right)\begin{bmatrix} \tilde{v}^{(m)} \\ 1 \end{bmatrix} - \frac{\tilde{\delta}_m}{\tilde{\delta}_{m-1}}\begin{bmatrix} \tilde{v}^{(m-1)} \\ 1 \\ 0 \end{bmatrix} \quad (26)$$

with

$$\tilde{\delta}_m = (\tilde{g}^{(m)})^T\tilde{v}^{(m)} + \tilde{r}_0 + \tilde{r}_{2(m+1)} \neq 0,$$

$$\tilde{\lambda}_m = (\tilde{g}^{(m+1)})^T\begin{bmatrix} \tilde{v}^{(m)} \\ 1 \end{bmatrix}$$

and

$$\tilde{g}^{(m)} = (\tilde{r}_m - \tilde{r}_{m+2}, \tilde{r}_{m-1} - \tilde{r}_{m+3}, \ldots, \tilde{r}_1 - \tilde{r}_{2m+1})^T \in \mathbb{R}^m.$$

V. Numerical experiments

The purpose of this section is to show the effectiveness of the proposed fast CR algorithm. We compared the numerical performance of Algorithm 3.2 (FCR) with that of Algorithm 3.1 (CR) and SDA (see SDA-2 in [4]). Our experiments were implemented in Fortran 90 and tested on a PC with AMD 3600+ processor and 512M memory, which had unit roundoff $u = 2^{-53} \approx 1.1 \times 10^{-16}$. In CR algorithm and SDA algorithm, we used Fortran subroutines DSPTRF and DTRSM in LAPACK [1] to compute the Cholesky factorization in (9) and solve the triangular matrix equation for U_k and V_k in (10), respectively. The stop criterion of all three algorithms is

$$\frac{\|S_{k+1} - S_k\|_1}{\|S_k\|_1} \leq nu, \quad (27)$$

where n is the dimension of the problem. When (27) was satisfied, we took S_{k+1} as an approximation to $\widehat{S}$.

Example 5.1 Consider the QME (3) with

$$M = I,$$
$$D = \beta \cdot \text{tridiag}(-10, 30, -10),$$
$$K = \text{tridiag}(-5, 15, -5),$$

where $\beta > 0$ is a real parameter to determine the overdamped degree of (3) [6].

Following the detecting method proposed in [6], QME (3) is weakly overdamped for some $\beta \in (0.447213, 0.447214)$. We took the dimension n varying from 500 to 3000 and $\beta = 1$, 0.4473 (i.e different overdamped degrees) to test all algorithms. We reported the CPU time elapsed for obtaining $S^{(2)}$ in Figure 1 and the relative residual, calculated as

$$\text{Res} = \frac{\|Q(S_k)\|_F}{\|A\|_F\|S_k\|_F^2 + \|B\|_F\|S_k\|_F + \|C\|_F},$$

in Figure 2.

We can see from Figure 1 that all algorithms need more CPU time to obtain the solvent when the overdamped degree of the QME (3) is weaker (i.e. β is smaller). However in any case, the FCR algorithm outperforms the CR algorithm and SDA algorithm in CPU time. We also note that the used time of FCR algorithm increase largely when $n > 2000$, this may be caused by the insufficient memory in our PC. In terms of accuracy, Figure 2 shows that the CR algorithm and SDA algorithm perform better than FCR algorithm.

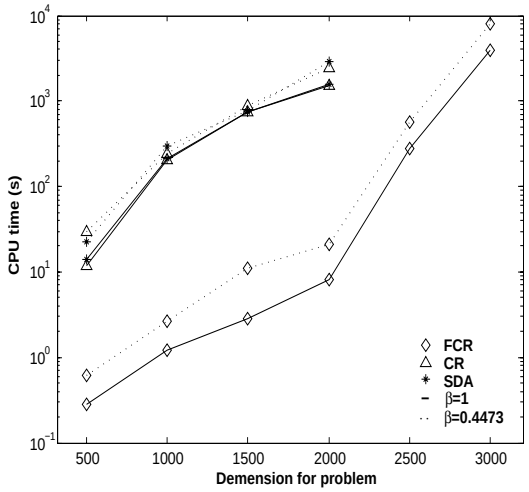

Fig. 1. CPU time for different algorithms and μ in Example 5.1.

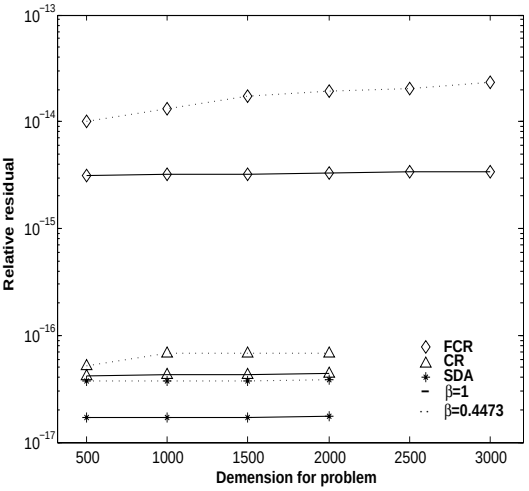

Fig. 2. Relative residual for different algorithms and μ in Example 5.1.

Example 5.2 We consider the QME (6) with A and C of the form R defined in (2), where r_m $(1 \leq m \leq n)$ are random numbers distributed in $(-1, 0)$ and $r_0 = 2n$. Let $B = \mu A + \mu^{-1}C + 10^{-3}I_n$ with $\mu > 0$. It follows from Definition 1.1 that such QME is overdamped.

We took $\mu = 1, 0.5$ to test the CPU time used for different algorithms. The stop criterion and the computation of the relative residual are the same with Example 5.1. Figure 3 and 4 give the total time to obtain $S^{(2)}$ and the calculated relative residual, respectively.

We can see from Figure 3 that the CPU time used by FCR algorithm was less than the CR algorithm and the SDA algorithm for different μ. The same with Example 5.1, Figure

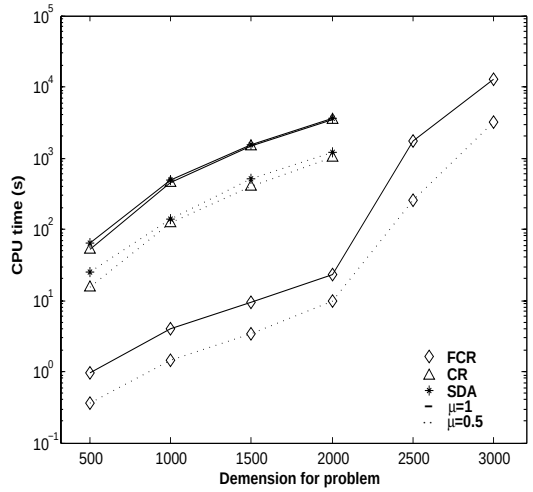

Fig. 3. CPU time for different algorithms and μ in Example 5.2.

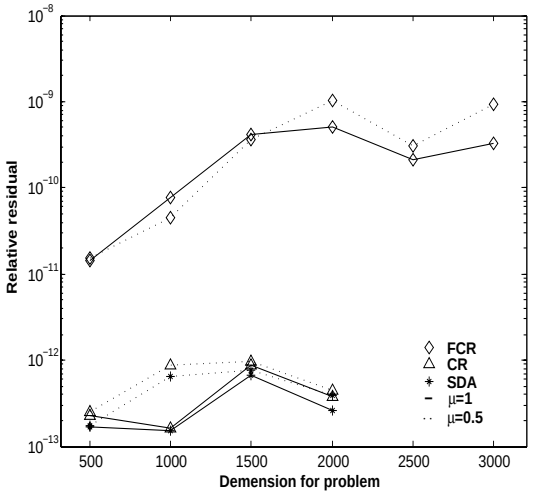

Fig. 4. Relative residual for different algorithms and μ in Example 5.2.

4 shows that the relative residual of FCR algorithm is larger than the other two non-structured algorithms.

VI. Conclusion

We have presented a fast cyclic reduction algorithm for obtaining extremal solutions of quadratic matrix equations arising from the overdamped mass-spring system. This method is based on recursive formula derived by exploring the structure of the coefficient matrices. The preliminary numerical results show that the proposed method outperforms the non-structured CR algorithm and SDA algorithm. At the moment, we are not aware if it is possible to devise a more general fast CR algorithm when the damper (spring) constants are different. We leave it as a topic for further study.

Acknowledgment

This work was supported in part by the major project of the Ministry of Education of China granted 309023 and the open fund project of key research institute of philosophies and social sciences in Hunan universities granted 11FEFM03.

References

[1] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *LAPACK Users' Guide*, (3rd Edition), SIAM, Philadelphia, PA, (1999).

[2] D.A. Bini, L. Gemignani and B. Meini, *Computations with infinite Toeplitz matrices and polynomials*, Linear Algebra Appl., 343/344 (2002), pp. 21-61.

[3] D. A. Bini, B. Meini, F. Poloni, *Fast solution of a certain Riccati equation through Cauchy-like matrices*, ETNA., 33 (2009), pp. 84-104.

[4] C.-Y. Chiang, E.K. W. Chu, C.-H. Guo, T.-M. Huang, W.-W. Lin, S.-F. Xu, *Convergence analysis of the doubling algorithm for several nonliear matrix equations in the critical case*, SIAM J. Matrix Anal. Appl., 31, 227-247, 2009.

[5] G.H. Golub and C.F. Van Loan, *Matrix Computations*, (3rd Edition), Johns Hopkins University Press, Baltimore, MD, (1996).

[6] C.-H. Guo, N. J. Higham and F. Tisseur, *Detecting and solving hyperbolic quadratic eigenvalue problems*, SIAM J. Matrix Anal. Appl., 30 (2009), pp. 1593-1613.

[7] C.-H. Guo and P. Lancaster, *Algorithms for hyperbolic quadratic eigenvalue problems*, Math. Comp., 74 (2005), pp. 1777-1791.

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

[9] I. Gohberg, T. Kailath and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557-1576.

[10] G. Heinig, P. Jankowski and K. Rost, *Fast inversion algorithm of Toeplitz-plus-Hankel matrices*, Numer. Math., 52 (1988), pp. 665-682.

[11] N.J. Higham and H.-M. Kim, *Numerical ananlysis of a quadratic matrix equation*, IMA. J. Numer. Anal., 20 (2000), pp. 499-519.

[12] N.J. Higham and H.-M. Kim, *Solving a quadratic matrix equation by Newton's method with exact line search*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 303-316.

[13] T. Kailath and A.H. Sayed, *Fast Reliable Algorithms for Matrices with Structure*, SIAM, PA, (1999).

[14] W.-W. Lin and S.-F. Xu, *Analysis of structure-preserving doubling algorithms for Riccati-type matrix equations*, SIAM J. Matrix Anal. Appl. (2005).

[15] B. Meini, *Efficient computation of the extreme solutions of* $X + A^*X^{-1}A = Q$ *and* $X - A^*X^{-1}A = Q$, Math. Comp., 239 (2002), pp. 1189-1204.

[16] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339-361.

[17] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problems*, SIAM Rev., 43, (2001), pp. 235-286.

[18] F.-H. Wen, X.-G. Yang, *Skewness of return distribution and coeffcient of risk premium*. J. Syst. Sci. Complex., 22, (2009), pp. 360-371.

[19] F.-H. Wen, Z.-F. Liu, *A copula-based correlation measure and its application*. Int. J. Inform. Tech. Decis. Making, 8, (2009), pp: 1-15.

[20] B. Yu and N. Dong, *A structure-preserving doubling algorithm for extreme solvents of quadratic matrix equations*, Advanced Modeling and Optimization, 12, (2010), pp. 85–100.