Symmetric low-rank corrections to quadratic models

Jiang Qian\textsuperscript{1, *}, Shu-Fang Xu\textsuperscript{2} and Feng-Shan Bai\textsuperscript{1}

\textsuperscript{1}Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
\textsuperscript{2}LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, China

SUMMARY

In this paper, we study the quadratic model updating problems by using symmetric low-rank correcting, which incorporates the measured model data into the analytical quadratic model to produce an adjusted model that matches the experimental model data, and minimizes the distance between the analytical and updated models. We give a necessary and sufficient condition on the existence of solutions to the symmetric low-rank correcting problems under some mild conditions, and propose two algorithms for finding approximate solutions to the corresponding optimization problems. The good performance of the two algorithms is illustrated by numerical examples. Copyright \textcopyright{} 2008 John Wiley & Sons, Ltd.

Received 20 July 2007; Revised 6 August 2008; Accepted 22 August 2008

KEY WORDS: quadratic model; symmetric low-rank correcting; numerical method

1. INTRODUCTION

Quadratic models are used extensively in industry to model the behavior of physical structures. While these models provide a convenient means of understanding the performance of structures under a variety of user-defined load conditions, there are often discrepancies between analytical results from a quadratic model and actual experimental results. To reduce the discrepancies, attempts are made to adjust the quadratic model using the experimental data.

There have been many previous attempts to update an analytical quadratic model of a physical structure using the experimentally measured data (see [1] and references therein). One approach is to use the eigenstructure assignment method from control theory to update the quadratic model. An analytical quadratic control model can be described as

\[ M_a \ddot{q}(t) + C_a \dot{q}(t) + K_a q(t) = B_0 u(t) \]  

(1)

\*Correspondence to: Jiang Qian, Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China.
\textsuperscript{1}E-mail: jqian104@gmail.com

Contract/grant sponsor: Project NSFC; contract/grant number: 10571007
Contract/grant sponsor: RFDP; contract/grant number: 2003001103

Copyright \textcopyright{} 2008 John Wiley & Sons, Ltd.
where $M_a$, $C_a$ and $K_a \in \mathbb{R}^{n \times n}$ are all symmetric with $M_a$ symmetric positive definite, which represent the mass, damping and stiffness matrices, respectively, $q(t)$ is the $n \times 1$ vector of displacements, $B_0$ is the $n \times m$ actuator influence matrix, $u(t)$ is the $m \times 1$ vector of control force and the overdots represent differentiation with respect to time $t$. In addition, the $r \times 1$ output vector $y(t)$ of sensor measurement is given by

$$y(t) = C_0q(t) + C_1\dot{q}(t) \quad (2)$$

where $C_0$ and $C_1$ are $r \times n$ output influence matrices. In control engineering, the matrices $B_0$, $C_0$ and $C_1$ are given, whereas in model updating these matrices have to be chosen. The problem is then to design a control law

$$u(t) = F_0y(t) \quad (3)$$

given by the $m \times r$ feedback gain matrix $F_0$, such that the closed-loop system has the desired eigenvalues and eigenvectors. Substituting (2) and (3) into (1) gives

$$M_a\ddot{q}(t) + (C_a - B_0F_0C_1)\dot{q}(t) + (K_a - B_0F_0C_0)q(t) = 0 \quad (4)$$

From (4) we can see that the matrix triple products $B_0F_0C_1$ and $B_0F_0C_0$ can be viewed as low-rank correcting matrices to the damping and stiffness matrices such that the updated model matches the experimental model data. Some recent related works can be found in [2–6]. However, these matrices are generally nonsymmetric when standard eigenstructure assignment techniques are used, which suggests that the adjusted damping and stiffness matrices are also nonsymmetric. In order to force the pseudo-controller to yield symmetric updating matrices, the approaches in [7–9] are to produce an adjusted finite-element model with symmetric low-rank updating on the damping and stiffness matrices, which matches the experimental model data. The number of pseudo-sensors and pseudo-actuators are taken to be equal, that is, $r = m$, and a new influence matrix $B \in \mathbb{R}^{n \times m}$ and pseudo-controllers $F = F^T$, $G = G^T \in \mathbb{R}^{m \times m}$ are designed such that the feedback controllers in (4) have the following symmetric forms:

$$B_0F_0C_1 = BFB^T, \quad B_0F_0C_0 = BGB^T \quad (5)$$

Once such $B$, $F$ and $G$ are found, $B_0$, $F_0$, $C_0$ and $C_1$ can be easily designed by (5). As a simple example, we may take $B_0 = B$, $F_0$ as any nonsingular matrix and $C_0 = F_0^{-1}GB^T$, $C_1 = F_0^{-1}FB^T$.

Substituting (5) into (4), the closed-loop system becomes

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = 0 \quad (6)$$

where

$$M = M_a, \quad C = C_a - BFB^T, \quad K = K_a - BGB^T$$

The problem then becomes finding $B$, $F$ and $G$ such that the adjusted system (6) matches the partially assigned eigenvalues and eigenvectors that are determined experimentally, which is described as

**SLCP (Symmetric low-rank correcting problem)**

Given $M_a = M_a^T > 0$, $C_a = C_a^T$, $K_a = K_a^T$, the measured matrix of eigenvalues $\Lambda \in \mathbb{R}^{m \times m}$ and the matrix of eigenvectors $X \in \mathbb{R}^{n \times m}$ ($m < n$), find $B \in \mathbb{R}^{n \times m}$, $F = F^T$ and $G = G^T \in \mathbb{R}^{m \times m}$ such that

$$MX\Lambda^2 + CX\Lambda + KX = 0 \quad (8)$$

where $M = M_a$, $C = C_a - BFB^T$ and $K = K_a - BGB^T$. 

The solution to the SLCP is generally underdetermined. Therefore, the question arises as to how this freedom is exploited. One way is to make the corrections minimal, which leads to the following problem:

**MSLCP (Minimizing symmetric low-rank correcting problem)**

Let

\[ S = \{(B, F, G) \in \mathbb{R}^{n \times m} \times \mathbb{R}^{m \times m} \times \mathbb{R}^{m \times m} \mid \text{SLCP solves the SLCP}\} \]

and \( q = \|C_a\|_F/\|K_a\|_F \). Solve the optimization problem

\[
\min \{\|BFB^T\|_F^2 + q \|BGB^T\|_F^2 \mid (B, F, G) \in S\}
\]

where \( \mathbb{S}^{m \times m} \) denotes the set of \( m \times m \) symmetric matrices and \( \| \cdot \|_F \) denotes the Frobenius norm.

Recently, in [9, 10], an algorithm is developed to solve the SLCP, which needs to solve a general algebraic Riccati equation and several linear systems. However, the computed \( F \) and \( G \) are only the necessary but not sufficient condition for the symmetric low-rank correcting in (5). In [7], based on the general solution to Equation (8) (see [11] for details), a general solution to the SLCP is found under the assumption that the matrix \( X(M_aX^2 + C_aX + K_aX) \) is nonsingular, and hence an efficient algorithm is developed to solve the MSLCP by solving only one linear equation. However, this assumption is only a sufficient condition for the existence of solutions to the SLCP and is often not satisfied (as an example, see Examples 1 and 2 in Section 4). In view of this case, in this paper this assumption is removed and a necessary and sufficient condition for the existence of solutions to the SLCP is derived under some mild conditions and, moreover, a general solution to the SLCP is obtained under these conditions. By using the general solution, two algorithms are developed for finding approximate solutions to the corresponding MSLCP.

This paper is organized as follows. In Section 2, we first review some related work in [11], and then give a necessary and sufficient condition for the existence of solutions to the SLCP under some mild conditions. In this section a general solution to the SLCP is also given under these conditions. In Section 3, by using the general solution obtained in Section 2, the MSLCP can be reduced into an unconstrained optimization problem. Two algorithms are then developed for finding approximate solutions to this optimization problem. Some numerical examples are shown in Section 4 to illustrate the behavior of the algorithms. Conclusions are drawn in Section 5.

### 2. GENERAL SOLUTION TO THE SLCP

Now we turn to solve the SLCP. Without loss of generality, we may assume that matrices \((\Lambda, X) \in \mathbb{R}^{m \times m} \times \mathbb{R}^{n \times m}\) have the following form:

\[
\Lambda = \text{diag}(\lambda_1^{[2]}, \ldots, \lambda_\ell^{[2]}, \lambda_{2\ell+1}, \ldots, \lambda_m)
\]

with

\[
\lambda_j^{[2]} = \begin{bmatrix} x_j & \beta_j \\ -\beta_j & x_j \end{bmatrix}, \beta_j \neq 0, \text{ for } j = 1, \ldots, \ell, \text{ and }
\]

\[
X = [x_{1R}, x_{1I}, \ldots, x_{\ell R}, x_{\ell I}, x_{2\ell+1}, \ldots, x_m]
\]
Here, and subsequently, we assume that the eigenvector matrix $X$ in (10b) has full column rank ($\text{rank}(X) = m$) and the eigenvalue matrix $\Lambda$ in (10a) has only simple eigenvalues. Let $X = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}$ (11) be the $QR$-factorization of $X$, where $Q \in \mathbb{R}^{n \times n}$ is orthogonal, $Q_1 \in \mathbb{R}^{n \times m}$ and $R \in \mathbb{R}^{m \times m}$ is nonsingular and upper triangular, and let $S = RA R^{-1}$. Partition $Q^T M_a Q$ and $\hat{B} = Q^T B$ as $Q^T M_a Q = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, $\hat{B} = \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix}$ (12) where $M_{11}, \hat{B}_1 \in \mathbb{R}^{m \times m}$.

It has been shown in [7] that $(B, F, G)$ is a solution to the SLCP if and only if

\[
\hat{B}_1 F \hat{B}_1^T = M_{11} S + S^T M_{11} + R^{-T} DR^{-1} + Q_1^T C a Q_1
\]

(13a)

\[
\hat{B}_1 G \hat{B}_1^T = -S^T M_{11} S - R^{-T} DAR^{-1} + Q_1^T K a Q_1
\]

(13b)

\[
\hat{B}_2 (G \hat{B}_1^T + F \hat{B}_1^T S) = M_{21} S^2 + (Q_2^T C a Q_1) S + Q_2^T K a Q_1 \equiv W_2
\]

(13c)

where, with $\xi_i$ and $\eta_i$ being arbitrary real numbers,

\[
D = \text{diag}\left(\begin{bmatrix} \xi_1 & \eta_1 \\ \eta_1 & -\xi_1 \end{bmatrix}, \ldots, \begin{bmatrix} \xi_\ell & \eta_\ell \\ \eta_\ell & -\xi_\ell \end{bmatrix}, \xi_{\ell+1}, \ldots, \xi_m\right)
\]

(14)

In addition, it is also shown in [7] that

\[
\hat{B}_1 (G \hat{B}_1^T + F \hat{B}_1^T S) = M_{11} S^2 + (Q_1^T C a Q_1) S + Q_1^T K a Q_1 \equiv W_1
\]

(15)

See [7] for more details.

Since $B$ is generally required to be full column rank, we require $\hat{B}_1$ to be nonsingular here or equivalently $X^T B$ to be nonsingular. If $\hat{B}_1$ is chosen to be nonsingular, then the matrices $F$ and $G$ can be obtained from (13a) and (13b) for any given $D$ defined by (14). In addition, it follows from (15) that

\[
G \hat{B}_1^T + F \hat{B}_1^T S = \hat{B}_1^{-1} W_1
\]

(16)

Substituting (16) into (13c) we get

\[
\hat{B}_2 \hat{B}_1^{-1} W_1 = W_2
\]

(17)

It is worthwhile to note that we do not require $W_1$ to be nonsingular here.

Let $W_1$ have the singular value decomposition (SVD)

\[
W_1 = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} [V_1, V_2]^T
\]

(18)
where $U, V \in \mathbb{R}^{m \times m}$ are orthogonal with $V_1 \in \mathbb{R}^{m \times k}$ and $\Sigma \in \mathbb{R}^{k \times k}$ ($k \leq m$) is a diagonal matrix with positive diagonal elements. Partition $\hat{B}_2 \hat{B}_1^{-1} U$ as

$$
\hat{B}_2 \hat{B}_1^{-1} U = [B_{11}, B_{12}]
$$

(19)

where $B_{11} \in \mathbb{R}^{(n-m) \times k}$. Thus, (17) can be expressed as

$$
[B_{11}, B_{12}] \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} = W_2 V = [W_2 V_1, W_2 V_2]
$$

(20)

Clearly, this equation is solvable if and only if

$$
W_2 V_2 = 0
$$

(21)

and if it is solvable, the solution is given by

$$
B_{11} = W_2 V_1 \Sigma^{-1}, \quad B_{12} \text{ arbitrarily chosen}
$$

(22)

And, hence, $B$ solved from (19) and (12), $F$ from (13a) and $G$ from (13b) form a solution to the SLCP.

Now we take a deep look at the condition (21). From (18) it is easy to derive that

$$
\mathcal{R}(W_2) = \mathcal{N}(W_1)
$$

(23)

where $\mathcal{R}(\cdot)$ and $\mathcal{N}(\cdot)$ denote the range and the null space of a matrix, respectively. Then the condition (21) is equivalent to

$$
\mathcal{N}(W_1) \subseteq \mathcal{N}(W_2)
$$

(24)

From the definitions of $W_1$ and $W_2$ we can get that

$$
W_1 = Q_1^T H_a R^{-1}, \quad W_2 = Q_2^T H_a R^{-1}
$$

where $H_a = M_a X A^2 + C_a X A + K_a X$. Thus, (24) means that if a vector $x$ satisfies that

$$
W_1 x = Q_1^T H_a R^{-1} x = 0
$$

then

$$
W_2 x = Q_2^T H_a R^{-1} x = 0
$$

which implies that $H_a R^{-1} x = 0$, since $Q = [Q_1 \ Q_2]$ is orthogonal. Therefore, (24) is equivalent to

$$
\mathcal{N}(Q_1^T H_a R^{-1}) \subseteq \mathcal{N}(H_a R^{-1})
$$

Notice that $\mathcal{N}(H_a) \subseteq \mathcal{N}(Q_1^T H_a)$, $X = Q_1^T R$ and $R$ is nonsingular, then we can get that (20) is solvable if and only if

$$
\mathcal{N}(X^T H_a) = \mathcal{N}(H_a)
$$

We have finally completed our main result, which is summarized below.
Theorem 1
Given $M_a = M_a^T > 0$, $C_a = C_a^T$, $K_a = K_a^T$ and the eigenmatrix pair $(\Lambda, X) \in \mathbb{R}^{m \times m} \times \mathbb{R}^{n \times m}$. If $A$ has only simple eigenvalues, and $X$ is of full column rank, then the SLCP is solvable with $X^T B$ nonsingular if and only if

$$
\mathcal{N}(X^T H_a) = \mathcal{N}(H_a)
$$

(25)

with $H_a = M_a X \Lambda^2 + C_a X \Lambda + K_a X$. In addition, the solution to the SLCP is given by

$$
B = Q[\tilde{B}_1^T, \tilde{B}_2^T]^T = Q[\tilde{B}_1^T, [[W_2 V_1^{-1}, B_{12}^T]U^T \tilde{B}_1]^T]^T \in \mathbb{R}^{n \times m}
$$

$$
F = \tilde{B}_1^{-1}(M_{11} S + S^T M_{11} + Q_1^T C_a Q_1 + R^{-T} L R^{-1}) \tilde{B}_1^{-T} \in \mathbb{R}^{m \times m}
$$

$$
G = \tilde{B}_1^{-1}(-S^T M_{11} S + Q_1^T K_a Q_1 - R^{-T} L R^{-1}) \tilde{B}_1^{-T} \in \mathbb{R}^{m \times m}
$$

where $\tilde{B}_1 \in \mathbb{R}^{n \times m}$ is arbitrary and nonsingular, $B_{12} \in \mathbb{R}^{(n-m) \times (m-k)}$ is arbitrary and $D$ defined by (14) is arbitrary. Here all other notations are defined as above.

Remark 2.1
The assumption that $W_1$ is nonsingular in [7] implies that $X^T H_a$ is nonsingular, which means that $H_a$ has full column rank, and so (25) holds naturally. But conversely, (25) cannot guarantee that $W_1 = X^T H_a R^{-1}$ is nonsingular, generally.

Remark 2.2
As mentioned before, we are actually concerned about symmetric low-rank updating for quadratic models. After some calculation, we can see that although $(B, F, G)$ are determined by $\tilde{B}_1$, $B_{12}$ and $D$, the symmetric low-rank correcting matrices $BFB^T$ and $BGB^T$ are independent of $\tilde{B}_1$. And hence the inverse of $\tilde{B}_1$ in the formulas for $F$ and $G$ will not cause any numerical difficulties actually.

3. SOLVING THE MSLCP

Assume that the conditions in Theorem 1 hold and $(B, F, G)$ are described as above. Let $J = \|BFB^T\|_F^2 + q \|BGB^T\|_F^2$, where $q = \|C_a\|_F/\|K_a\|_F$. Then, by (19) and (22), it follows that

$$
\tilde{B}_1^{-1} = [W_2 V_1^{-1}, B_{12}] U^T = W_2 V_1^{-1} U_1^T + B_{12} U_2^T \equiv \phi(B_{12})
$$

(26)

where $U = [U_1, U_2]$ with $U_1 \in \mathbb{R}^{n \times k}$ and, hence,

$$
J = \left\| \begin{bmatrix} \tilde{B}_1 & \tilde{B}_2 \end{bmatrix} F[\tilde{B}_1^T, \tilde{B}_2^T] \right\|_F^2 + q \left\| \begin{bmatrix} \tilde{B}_1 & \tilde{B}_2 \end{bmatrix} G[\tilde{B}_1^T, \tilde{B}_2^T] \right\|_F^2
$$

$$
= \left\| \begin{bmatrix} I & \phi(B_{12}) \end{bmatrix} \tilde{F}[I, \phi(B_{12})^T] \right\|_F^2 + q \left\| \begin{bmatrix} I & \phi(B_{12}) \end{bmatrix} \tilde{G}[I, \phi(B_{12})^T] \right\|_F^2
$$

(27)
Then, the optimization problem (9) is equivalent to minimizing the following function:

\[ J(B_{12}, D) = \| F_c + R^{-T}D\Lambda^{-1} R^{-1} \|_F^2 + q \| G_c + R^{-T}D\Lambda^{-1} R^{-1} \|_F^2 \]

\[ + 2\| \phi(B_{12})F_c + \phi(B_{12})R^{-T}D\Lambda^{-1}R^{-1} \|_F^2 + 2q \| \phi(B_{12})G_c + \phi(B_{12})R^{-T}D\Lambda^{-1}R^{-1} \|_F^2 \]

\[ + \| \phi(B_{12})F_c \phi(B_{12})^T + \phi(B_{12})R^{-T}D\Lambda^{-1}R^{-1} \phi(B_{12})^T \|_F^2 \]

\[ + q \| \phi(B_{12})G_c \phi(B_{12})^T + \phi(B_{12})R^{-T}D\Lambda^{-1}R^{-1} \phi(B_{12})^T \|_F^2 \] (29)

where \( B_{12} \) can be arbitrarily chosen and \( D \) is the undetermined block diagonal matrix as in (14). We will show two approaches for finding approximate solutions to this optimization problem.

### 3.1. Approach 1

Notice that for a fixed \( D \), if \( \phi(B_{12})=0 \), then the function \( J \) is minimized. Based on this, as an approximation, we take \( B_{12} \) such that \( \| \phi(B_{12}) \|_F \) is minimized. From (26) we can see that it is equivalent to choosing \( B_{12}=0 \). Write \( \phi(0)=H \), then the function to be minimized becomes

\[ J_1(D) = \| F_c + R^{-T}D\Lambda^{-1} R^{-1} \|_F^2 + q \| G_c + R^{-T}\Lambda^T\Lambda^{-1} R^{-1} \|_F^2 \]

\[ + 2\| F_1 + HR^{-T}D\Lambda^{-1}R^{-1} \|_F^2 + 2q \| G_1 + HR^{-T}\Lambda^T\Lambda^{-1}R^{-1} \|_F^2 \]

\[ + \| F_2 + HR^{-T}D\Lambda^{-1}H^T \|_F^2 + q \| G_2 + HR^{-T}\Lambda^T\Lambda^{-1}H^T \|_F^2 \] (30)

where \( F_1=HF_c, F_2=HF_cH^T, G_1=HG_c \) and \( G_2=HG_cH^T \). Let

\[ x = [\xi_1, \eta_1, \ldots, \xi_\ell, \eta_\ell, \xi_{2\ell+1}, \ldots, \xi_m]^T \] (31)

corresponding to the matrix \( D \) in (14), and express

\[ R^{-1} = [r_1, r_2, \ldots, r_m] = (r_{ij})_{m \times m} \] (32a)

\[ R^{-1}H^T = [h_1, h_2, \ldots, h_m] = (h_{ij})_{m \times m} \] (32b)

Then it is easily seen that the vectors \( Dr_j \) and \( Dh_j \) can be, respectively, rewritten as

\[ Dr_j = \Gamma_j x, \quad Dh_j = \Sigma_j x \] (33)

for \( j=1, \ldots, m \), where

\[ \Gamma_j = \text{diag} \left( \begin{bmatrix} r_{1,j} & r_{2,j} \\ -r_{2,j} & r_{1,j} \end{bmatrix}, \ldots, \begin{bmatrix} r_{2\ell-1,j} & r_{2\ell,j} \\ -r_{2\ell,j} & r_{2\ell-1,j} \end{bmatrix}, r_{2\ell+1,j}, \ldots, r_{m,j} \right) \] (34a)
Substituting (33) into (30) we get

\[
\nabla J_1(x) = 2 \sum_{j=1}^{m} (\Gamma_j^TR^{-1}F_c(:, j) + \Gamma_j^TR^{-1}R^{-T}\Gamma_jx + q\Gamma_j^TAR^{-1}G_c(:, j) \\
+ q\Gamma_j^TAR^{-1}R^{-T}\Lambda^T\Gamma_jx + 2\Gamma_j^TAR^{-1}H^THR^{-T}\Gamma_jx \\
+ 2q\Gamma_j^TAR^{-1}HG_1(:, j) + 2q\Gamma_j^TAR^{-1}H^THR^{-T}\Lambda^T\Gamma_jx) \\
+ 2\sum_{j=1}^{n-m} (\Sigma_j^TR^{-1}H^TR_2(:, j) + \Sigma_j^TR^{-1}H^THR^{-T}\Sigma_jx) \\
+ q\sum_{j=1}^{n-m} \Sigma_j^TR^{-1}H^THR^{-T}\Lambda^T\Sigma_j)
\]

Substituting (33) into (30) we get

\[
\nabla J_1(x) = 2 \sum_{j=1}^{m} (\Gamma_j^TR^{-1}F_c(:, j) + \Gamma_j^TR^{-1}R^{-T}\Gamma_jx + q\Gamma_j^TAR^{-1}G_c(:, j) \\
+ q\Gamma_j^TAR^{-1}R^{-T}\Lambda^T\Gamma_jx + 2\Gamma_j^TAR^{-1}H^THR^{-T}\Gamma_jx \\
+ 2q\Gamma_j^TAR^{-1}HG_1(:, j) + 2q\Gamma_j^TAR^{-1}H^THR^{-T}\Lambda^T\Gamma_jx) \\
+ 2\sum_{j=1}^{n-m} (\Sigma_j^TR^{-1}H^TR_2(:, j) + \Sigma_j^TR^{-1}H^THR^{-T}\Sigma_jx) \\
+ q\sum_{j=1}^{n-m} \Sigma_j^TR^{-1}H^THR^{-T}\Lambda^T\Sigma_j)
\]

Here we use the Matlab notation $A(:, j)$ to represent the $j$th column of a matrix $A$.

Setting $\nabla J_1(x) = 0$ leads to the linear equation for $x$ as follows:

\[
AX = b
\]

where

\[
A = \sum_{j=1}^{m} (\Gamma_j^TR^{-1}R^{-T}\Gamma_j + 2\Gamma_j^TR^{-1}H^THR^{-T}\Gamma_j) + \sum_{j=1}^{n-m} \Sigma_j^TR^{-1}H^THR^{-T}\Sigma_j \\
+ q\sum_{j=1}^{m} (\Gamma_j^TAR^{-1}R^{-T}\Lambda^T\Gamma_j + 2\Gamma_j^TAR^{-1}H^THR^{-T}\Lambda^T\Gamma_j) \\
+ q\sum_{j=1}^{n-m} \Sigma_j^TAR^{-1}H^THR^{-T}\Lambda^T\Sigma_j
\]

\[
b = -\sum_{j=1}^{m} (\Gamma_j^TR^{-1}F_c(:, j) + 2\Gamma_j^TR^{-1}H^TR_1(:, j)) - \sum_{j=1}^{n-m} \Sigma_j^TR^{-1}H^TR_2(:, j) \\
- q\sum_{j=1}^{m} (\Gamma_j^TAR^{-1}G_c(:, j) + 2\Gamma_j^TAR^{-1}H^TR_1(:, j)) \\
- q\sum_{j=1}^{n-m} \Sigma_j^TAR^{-1}H^TR_2(:, j)
\]

In summary, we have the following algorithm:

**Algorithm 3.1**

**Input:** $M_a = M_a^T > 0$, $C_a = C_a^T$, $K_a = K_a^T$ and $(\Lambda, X) \in \mathbb{R}^{m \times m} \times \mathbb{R}^{n \times m}$ as in (10).

**Output:** $B \in \mathbb{R}^{n \times m}$, $F = F^T$, $G = G^T \in \mathbb{R}^{m \times m}$. 

1. Compute the QR-factorization of $X$:

$$X = QR = \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}$$

and $S = RAR^{-1}$

2. Compute $M_{11} = Q_1^TM_aQ_1$, $M_{21} = Q_2^TM_aQ_1$;

3. Compute $W_1$ and $W_2$ by (15) and (13c), respectively, and compute the SVD decomposition of $W_1$ as in (18);

4. Compute $H = \phi(0), F_c, F_1, F_2$ and $G_c, G_1, G_2$, and compute $r_{ij}, h_{ij}, i, j = 1, 2, \ldots, m$, as in (32);

5. Solve $Ax = b$ for $x$, where $A, b$ are evaluated by (37), in which $\Gamma_j$ and $\Sigma_j, j = 1, \ldots, m$, are given by (34);

6. Form $D$ as in (14), and compute

$$B = Q \begin{bmatrix} I \\ H \end{bmatrix}$$

$$F = M_{11}S + S^TM_{11} + Q_1^TC_aQ_1 + R^{-T}DR^{-1}$$

$$G = -S^TM_{11}S + Q_1^TK_aQ_1 - R^{-T}DAR^{-1}$$

3.2. Approach 2

As a simple approach, $B_{12}$ is set to zero and then the objective function $J$ is minimized with respect to $D$ in Algorithm 3.1. As another approach, we can let one parameter($B_{12}$ or $D$) fixed, and minimize $J$ with respect to the other parameter ($D$ or $B_{12}$) alternately. That is, for a fixed $B_{12}^{(k)}$, $D^{(k)}$ is solved such that the following function is minimized:

$$J_1(D) = \|F_c + R^{-T}DR^{-1}\|_F^2 + q\|G_c + R^{-T}A^TDAR^{-1}\|_F^2$$

$$+ 2\|F_1^{(k)} + H^{(k)}R^{-T}DR^{-1}\|_F^2 + 2q\|G_1^{(k)} + H^{(k)}R^{-T}A^TDAR^{-1}\|_F^2$$

$$+ \|F_2^{(k)} + H^{(k)}R^{-T}DR^{-1}(H^{(k)})^T\|_F^2$$

$$+ q\|G_2^{(k)} + H^{(k)}R^{-T}A^TDAR^{-1}(H^{(k)})^T\|_F^2$$

(38)

where $H^{(k)} = \phi(B_{12}^{(k)})$, $F_1^{(k)} = H^{(k)}F_c$, $F_2^{(k)} = H^{(k)}F_c(H^{(k)})^T$, $G_1^{(k)} = H^{(k)}G_c$, $G_2^{(k)} = H^{(k)}G_c(H^{(k)})^T$.

Then, for the resulted $D^{(k)}$, $B_{12}^{(k+1)}$ is solved such that the following function is minimized:

$$J_2(B_{12}) = \|F^{(k)}\|_F^2 + q\|G^{(k)}\|_F^2 + 2\|\phi(B_{12})F^{(k)}\|_F^2 + 2q\|\phi(B_{12})G^{(k)}\|_F^2$$

$$+ \|\phi(B_{12})F^{(k)}\phi(B_{12})^T\|_F^2 + q\|\phi(B_{12})G^{(k)}\phi(B_{12})^T\|_F^2$$

(39)

where $F^{(k)} = F_c + R^{-T}D^{(k)}R^{-1}$ and $G^{(k)} = G_c + R^{-T}D^{(k)}AR^{-1}$. In addition, such process is repeated until some stop criterion is satisfied.
While finding \( D^{(k)} \) that minimizes \( J_1(D) \) described as in (38), the objective function is quite similar to the one described as in (30), except that \( H \) is replaced by \( H^{(k)} \). Hence we can use similar methods to solve for \( D^{(k)} \). While finding \( B_{12}^{(k+1)} \) that minimizes \( J_2(B_{12}) \) described as in (39), we use the deepest descent method with exact linear search here. In brief, in each inner iteration, \( B_{12}^{(k,j)} \) is modified to \( B_{12}^{(k,j+1)} \) by

\[
B_{12}^{(k,j+1)} = B_{12}^{(k,j)} - \alpha_{k,j} G_{k,j}
\]

where \( k \) and \( j \) denote the outer- and inner-loop steps, respectively, \( G_{k,j} = \nabla J_2(B_{12}^{(k,j)}) \) is the gradient matrix of \( J_2(B_{12}) \) on \( B_{12}^{(k,j)} \) and \( \alpha_{k,j} \) is the descent step solved by exact linear search.

After some calculation, we have

\[
G_{k,j} = \nabla J_2(B_{12}^{(k,j)}) = 4\phi(B_{12}^{(k,j)}) F^{(k)} F^{(k)} U_2 + 4q \phi(B_{12}^{(k,j)}) G^{(k)} G^{(k)} U_2
\]

\[
+ 4\phi(B_{12}^{(k,j)}) F^{(k)} \phi(B_{12}^{(k,j)})^T \phi(B_{12}^{(k,j)}) F^{(k)} U_2
\]

\[
+ 4q \phi(B_{12}^{(k,j)}) G^{(k)} \phi(B_{12}^{(k,j)})^T \phi(B_{12}^{(k,j)}) G^{(k)} U_2
\]

(41)

where \( \phi(\cdot) \) and \( U_2 \) are defined as in (26). Such an iteration is repeated until

\[
\| G_{k,j} \|_F < \text{tol}
\]

(42)

for some certain \( j_0 \), where \( \text{tol} \) is a given tolerance, and the resulted \( B_{12}^{(k,j_0)} \) is set to \( B_{12}^{(k+1)} \).

In summary, we have the following algorithm:

**Algorithm 3.2**

*Input: \( M_a = M_a^T > 0 \), \( C_a = C_a^T \), \( K_a = K_a^T \) and \((A, X) \in \mathbb{R}^{m \times m} \times \mathbb{R}^{n \times m} \) as in (10).*

*Output: \( B \in \mathbb{R}^{n \times m} \), \( F = F^T \), \( G = G^T \in \mathbb{R}^{m \times m} \).*

1. **Compute the \( QR \)-factorization of \( X \):**

\[
X = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} \quad \text{and} \quad S = R A R^{-1}
\]

2. **Compute** \( M_{11} = Q_1^T M_a Q_1 \), \( M_{21} = Q_1^T M_a Q_2 \);

3. **Compute** \( W_1 \) and \( W_2 \) by (15) and (13c), respectively, and compute the SVD decomposition of \( W_1 \) as in (18);

4. **Set** \( k = 1 \), and \( B_{12}^{(1)} = 0 \);

5. **Compute** \( H_k, F_c, F_1^{(k)}, F_2^{(k)}, G_c, G_1^{(k)}, G_2^{(k)} \) as described above, and compute \( D^{(k)} \) that minimizes \( J_1(D) \) described as in (38);

6. **Compute** \( F^{(k)} \) and \( G^{(k)} \) as described above, and compute \( B_{12}^{(k+1)} \) that minimizes \( J_2(B_{12}) \) described as in (39);

7. **If** \( |J(B_{12}^{(k+1)}, D^{(k)}) - J(B_{12}^{(k)}, D^{(k)})| > \text{tol} \cdot |J(B_{12}^{(k)}, D^{(k)})| \), set \( k = k + 1 \), and go to Step 5;
8. Set \( D = D^{(k)} \), and \( H = \phi(B_{12}^{(k+1)}) \), and compute

\[
B = Q \begin{bmatrix} I \\ H \end{bmatrix}
\]

\[
F = M_{11}S + S^T M_{11} + Q_1^T C_a Q_1 + R^{-T} D R^{-1}
\]

\[
G = -S^T M_{11} S + Q_1^T K_a Q_1 - R^{-T} D A R^{-1}
\]

**Remark 1**
In Step 6, the deepest descent method with exact linear search is used here. Other optimization methods can also be applied to solve for \( B_{12}^{k+1} \), and it deserves further study.

**Remark 2**
In Step 7, \( \text{tol}_1 \) is a given tolerance such that when the relative change of the objective function is too small, the outer iteration is stopped.

**Remark 3**
In fact, we can see that Algorithm 3.1 is just one step in Algorithm 3.2. The reason why we propose Algorithm 3.1 separately is that Algorithm 3.1 is much easier to implement, and it is a good enough choice in some cases.

### 4. NUMERICAL RESULTS

To illustrate the performance of the present algorithms, in this section we give some numerical examples, which are carried out using Matlab 6.5 with machine epsilon \( \varepsilon \approx 2.22 \times 10^{-16} \), and the tolerances in Algorithm 3.2 are set to \( \text{tol} = 10^{-4} \), \( \text{tol}_1 = 10^{-4} \).

**Example 1**
Consider an analytical five-DOF system [7, 9] with mass, damping and stiffness matrices given by

\[
M_a = \text{diag}(1, 2, 5, 4, 3)
\]

\[
\begin{bmatrix}
11 & -2 & 0 & 0 & 0 \\
-2 & 14 & -3.5 & 0 & 0 \\
0 & -3.5 & 13.0 & -1.2 & 0 \\
0 & 0 & -1.2 & 13.5 & -4 \\
0 & 0 & 0 & -415.4 & 0
\end{bmatrix}
\]

\[
C_a = \begin{bmatrix}
100 & -20 & 0 & 0 & 0 \\
-20 & 120 & -35 & 0 & 0 \\
0 & -35 & 80 & -12 & 0 \\
0 & 0 & -12 & 95 & -40 \\
0 & 0 & 0 & -40 & 124
\end{bmatrix}
\]

\[
K_a = \begin{bmatrix}
100 & -20 & 0 & 0 & 0 \\
-20 & 120 & -35 & 0 & 0 \\
0 & -35 & 80 & -12 & 0 \\
0 & 0 & -12 & 95 & -40 \\
0 & 0 & 0 & -40 & 124
\end{bmatrix}
\]
Suppose that the assigned matrix of eigenvalues and matrix of eigenvectors are

\[
\Lambda = \begin{bmatrix}
-1.116 & 3.057 & 0 & 0 \\
-3.057 & -1.116 & 0 & 0 \\
0 & 0 & -2.854 & 6.240 \\
0 & 0 & -6.240 & -2.854
\end{bmatrix}
\]

\[
X = \begin{bmatrix}
0 & 0 & -0.0155 & 0.0004 \\
0.3708 & 0.0048 & -0.0409 & 0.0009 \\
1 & 0 & -0.0213 & 0.0005 \\
0 & 0 & 0.3956 & 0.009 \\
0 & 0 & -0.9168 & 0.0210
\end{bmatrix}
\]

In this example, \( W_1 \) defined as in (15) is singular; thus, the assumption that \( W_1 \) is nonsingular in [7] does not hold, and hence the algorithm proposed there cannot be applied to this problem. Apply Algorithms 3.1 and 3.2 to this problem with \( q = \| C_a \|_F / \| K_a \|_F \). Algorithm 3.1 computes matrices \( B, F, G \) with

\[
B = \begin{bmatrix}
0.1803 & 0.5594 & -1.483 & 0.0067 \\
-0.3477 & -0.9376 & 0 & 0 \\
-0.9376 & 0.34767 & 0 & 0 \\
0 & 0 & -0.3961 & -0.9182 \\
-0.0030 & -0.0094 & 0.9433 & -0.3963
\end{bmatrix}
\]

\[
F = \begin{bmatrix}
0.4905 & -0.2536 & -0.2319 & -49.32 \\
-0.2536 & -107.4 & -96.59 & 27.28 \\
-0.2319 & -96.59 & -6.116 & 6.302 \\
49.32 & 27.28 & 6.302 & 803.0
\end{bmatrix}
\]

\[
G = \begin{bmatrix}
12.90 & -4.790 & -5.371 & -64.15 \\
-4.790 & 60.15 & -252.7 & 3501 \\
-5.371 & -252.7 & -16.08 & 1186 \\
-64.15 & 3501 & 1186 & -852.6
\end{bmatrix}
\]

The updated finite-element model \( (M, C, K) \equiv (M_a, C_a - BFB^T, K_a - BGB^T) \) satisfies the error bound

\[
\| MX\Lambda^2 + CXA + KX \|_F = 2.82 \times 10^{-11}
\]
and the value of $J = \|BFB^T\|_F^2 + q\|BGB^T\|_F^2$ is $3.88 \times 10^8$. Algorithm 3.2 converges after six iterations and computes matrices $B, F, G$ with

$$B = \begin{bmatrix}
-0.01235 & 0.05501 & -1.6941 & 0.0079 \\
-0.3477 & -0.9376 & 0 & 0 \\
-0.9376 & 0.3477 & 0 & 0 \\
0 & 0 & -0.3961 & -0.9182 \\
0.0002 & -0.0009 & 0.9469 & -0.3963
\end{bmatrix}$$

$$F = \begin{bmatrix}
0.4884 & -1.297 & -0.2643 & -49.31 \\
-1.297 & 8.088 & -93.05 & 25.75 \\
-0.2643 & -93.05 & -5.781 & -2.606 \\
-49.31 & 25.75 & -2.606 & -29.04
\end{bmatrix}$$

$$G = \begin{bmatrix}
12.89 & -4.469 & -5.361 & -64.15 \\
-4.469 & 944.0 & -225.3 & 35003 \\
-5.361 & -225.3 & -15.53 & 1075 \\
-64.15 & 35003 & 1075 & 10.39
\end{bmatrix}$$

The updated finite-element model $(M, C, K) \equiv (M_a, C_a - BFB^T, K_a - BGB^T)$ satisfies the error bound

$$\|MXA^2 + CXA + KX\|_F = 1.58 \times 10^{-11}$$

and the value of $J = \|BFB^T\|_F^2 + q\|BGB^T\|_F^2$ is $3.08 \times 10^8$.

We also apply both algorithms to this problem with different $q$’s, and list the results in Table I, where ‘Res’ denotes $\|MXA^2 + CXA + KX\|_F$ and ‘$J$’ denotes the value of $J = \|BFB^T\|_F^2 + q\|BGB^T\|_F^2$.

From the table, we can see that the value of $q$ does not affect the residual much, while the value of $J$ varies with $q$. It should be natural, since $q$ just plays a role in the optimization problem of MSLCP. We should also mention that Algorithm 3.2 always converges after a few steps.

<table>
<thead>
<tr>
<th>$q$</th>
<th>Algorithm 3.1</th>
<th>Algorithm 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>$9.74 \times 10^{-12}$</td>
<td>$3.11 \times 10^7$</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$2.35 \times 10^{-11}$</td>
<td>$3.79 \times 10^5$</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>$2.77 \times 10^{-11}$</td>
<td>$7.13 \times 10^4$</td>
</tr>
<tr>
<td>$10^{-7}$</td>
<td>$2.10 \times 10^{-11}$</td>
<td>$6.84 \times 10^4$</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>$2.18 \times 10^{-11}$</td>
<td>$6.81 \times 10^4$</td>
</tr>
</tbody>
</table>
This is because Algorithm 3.1, as the first step of Algorithm 3.2, has produced a good initial to Algorithm 3.2, as we mentioned above.

**Example 2**
Consider the system [12] with mass, damping and stiffness matrices given by

\[
M_a = \begin{bmatrix}
3.587 & 0.217 & 0.225 & -1.346 & 0.170 & 1.714 \\
0.217 & 3.397 & -0.128 & -0.470 & -0.304 & -1.094 \\
0.225 & -0.128 & 4.626 & -0.660 & 0.707 & -0.502 \\
-1.364 & -0.470 & -0.660 & 1.952 & 0.192 & 0.196 \\
0.170 & -0.304 & 0.707 & 0.192 & 4.955 & -0.1060 \\
1.714 & -1.094 & -0.502 & 0.196 & -0.106 & 3.707 \\
\end{bmatrix}
\]

\[
C_a = \begin{bmatrix}
5.424 & 0.052 & -0.667 & 0.666 & 0.242 & 0.078 \\
0.052 & 4.852 & 0.153 & 0.440 & -0.433 & 0.201 \\
-0.667 & 0.153 & 4.836 & -0.396 & 0.098 & -0.451 \\
0.666 & 0.440 & -0.396 & 4.983 & -0.074 & -0.646 \\
0.242 & -0.433 & 0.098 & -0.074 & 5.070 & 0.255 \\
0.078 & 0.201 & -0.451 & -0.646 & 0.255 & 5.256 \\
\end{bmatrix}
\]

\[
K_a = \begin{bmatrix}
1.809 & 0.285 & 0.218 & 0.226 & 0.623 & -1.902 \\
0.285 & 4.447 & 0.591 & 0.336 & -0.934 & -0.608 \\
0.218 & 0.591 & 4.572 & 0.917 & 0.889 & 0.196 \\
0.226 & 0.336 & 0.917 & 3.248 & 0.877 & 0.679 \\
0.623 & -0.934 & 0.889 & 0.877 & 5.125 & -0.081 \\
-1.902 & -0.608 & 0.196 & 0.679 & -0.081 & 4.491 \\
\end{bmatrix}
\]

Suppose that the assigned matrix of eigenvalues and matrix of eigenvectors are

\[
\Lambda = \begin{bmatrix}
-0.6019 & 0.6584 & 0 \\
-0.6584 & -0.6019 & 0 \\
0 & 0 & -2 \\
\end{bmatrix}
\]

\[
X = \begin{bmatrix}
-0.1634 & 0.4142 & -0.5720 & 0.2501 & 0.3949 & 0.0659 \\
0.1168 & 0.0520 & -0.3993 & -0.1084 & 0.2399 & 0.0816 \\
1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}^T
\]
Again $W_1$ is singular, thus the algorithm proposed in [7] cannot be applied to this problem. Apply Algorithms 3.1 and 3.2 to this problem. Algorithm 3.1 computes matrices $B, F, G$ with

$$B = \begin{bmatrix} -0.1183 & 0.4543 & 0.2466 \\ 0.4991 & -0.3407 & 0.1735 \\ -0.7133 & -0.4242 & 0.6893 \\ 0.3442 & -0.6304 & -0.0024 \\ 0.3526 & 0.3059 & 0.8538 \\ 0.0862 & 0.1449 & 0.3964 \end{bmatrix}, \quad F = \begin{bmatrix} 0.1123 & -1.055 & 1.033 \\ -1.057 & 1.307 & -1.959 \\ 1.025 & -1.933 & -3.967 \end{bmatrix}, \quad G = \begin{bmatrix} -0.1900 & -0.5611 & 0.3657 \\ -0.5616 & 2.3930 & -2.563 \\ 0.3638 & -2.544 & 2.185 \end{bmatrix}$$

The updated finite-element model $(M, C, K) = (M_a, C_a - B F B^T, K_a - B G B^T)$ satisfies the error bound

$$\|M X A^2 + C X A + K X\|_F = 9.42 \times 10^{-10}$$

and the value of $J = \|B F B^T\|_F^2 + q \|B G B^T\|_F^2$ is 91.54. Algorithm 3.2 converges after two steps and computes matrices $B, F, G$ with

$$B = \begin{bmatrix} -0.1060 & 0.3544 & 0.2671 \\ 0.5186 & -0.4998 & 0.2062 \\ -0.7158 & -0.4036 & 0.6851 \\ 0.3383 & -0.5828 & -0.0121 \\ 0.3389 & 0.4181 & 0.8308 \\ 0.0765 & 0.2234 & 0.3803 \end{bmatrix}, \quad F = \begin{bmatrix} 0.0760 & -1.017 & 1.048 \\ -1.018 & 1.479 & -1.930 \\ 1.039 & -1.904 & -3.819 \end{bmatrix}, \quad G = \begin{bmatrix} -0.2112 & -0.4810 & 0.3864 \\ -0.4815 & 2.380 & -2.581 \\ 0.3845 & -2.562 & 2.466 \end{bmatrix}$$

The updated finite-element model $(M, C, K) = (M_a, C_a - B F B^T, K_a - B G B^T)$ satisfies the error bound

$$\|M X A^2 + C X A + K X\|_F = 9.92 \times 10^{-10}$$

and the value of $J = \|B F B^T\|_F^2 + q \|B G B^T\|_F^2$ is 90.71.

We also apply both algorithms to this problem with different $q$’s, and list the results in Table II, where ‘Res’ denotes $\|M X A^2 + C X A + K X\|_F$ and ‘$J$’ denotes the value of $J = \|B F B^T\|_F^2 + q \|B G B^T\|_F^2$. 

Table II. Residual and $J$ of both algorithms with different $q$’s (Example 2).

<table>
<thead>
<tr>
<th>$q$</th>
<th>Algorithm 3.1</th>
<th>Algorithm 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Res</td>
<td>$J$</td>
</tr>
<tr>
<td>5</td>
<td>$8.46 \times 10^{-10}$</td>
<td>188.7</td>
</tr>
<tr>
<td>1</td>
<td>$5.25 \times 10^{-10}$</td>
<td>86.49</td>
</tr>
<tr>
<td>0.1</td>
<td>$4.65 \times 10^{-10}$</td>
<td>35.38</td>
</tr>
<tr>
<td>0.01</td>
<td>$8.55 \times 10^{-10}$</td>
<td>18.94</td>
</tr>
<tr>
<td>0.001</td>
<td>$7.78 \times 10^{-10}$</td>
<td>16.72</td>
</tr>
<tr>
<td>0.0001</td>
<td>$6.64 \times 10^{-10}$</td>
<td>16.50</td>
</tr>
</tbody>
</table>

It is quite similar to Example 1. The residual does not change much, while the value of $J$ varies with $q$. Again Algorithm 3.2 converges after two to three steps. We can see from Table II that the value of $J$ computed by Algorithm 3.1 is quite close to that computed by Algorithm 3.2; therefore, we may conclude that the reason why Algorithm 3.2 converges so fast is that Algorithm 3.1 has produced quite a good initial to Algorithm 3.2. In addition, it should also be mentioned that when $W_1$ is nonsingular, our algorithms are exactly the same as the algorithm presented in [7].

5. CONCLUSIONS

We have studied the symmetric low-rank correcting problems for second-order systems. In [7], an efficient algorithm is developed to incorporate the measure model data into a quadratic eigenvalue problem of an analytical quadratic model so that the adjusted model closely matches the experimental data, and some measurement of the updating matrices is minimized. However, the algorithm proposed there is based on the assumption that the matrix $W_1$ defined in Section 2 is nonsingular, which is often not satisfied. In this paper, this assumption is removed and a necessary and sufficient condition on the existence of solutions to the symmetric low-rank correcting problems is derived under some mild conditions. In addition, moreover, two algorithms are proposed for solving the corresponding optimization problem. Numerical examples show that the two algorithms are reliable and attractive.

ACKNOWLEDGEMENTS

The research of the first author was partly supported by Project NSFC 10571007. The research of the second author was partly supported by RFDP (2003001103) and Project NSFC 10571007.

REFERENCES


