Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

ELA

THE MINIMUM-NORM LEAST-SQUARES SOLUTION OF A LINEAR SYSTEM AND SYMMETRIC RANK-ONE UPDATES*

XUZHOU CHEN† AND JUN JI‡

Abstract. In this paper, we study the Moore-Penrose inverse of a symmetric rank-one perturbed matrix from which a finite method is proposed for the minimum-norm least-squares solution to the system of linear equations $Ax = b$. This method is guaranteed to produce the required result.

Key words. Finite method, Linear system, Moore-Penrose inverse, Symmetric rank-one update.

AMS subject classifications. 15A06, 15A09, 65F05.

1. Introduction. Throughout this paper we shall use the standard notations in [2, 14]. $\mathbb{C}^n$ and $\mathbb{C}^{m\times n}$ stand for the n-dimensional complex vector space and the set of $m \times n$ matrices over complex field $\mathbb{C}$, respectively. For a matrix $A \in \mathbb{C}^{m\times n}$ we denote $R(A)$, $N(A)$, A^*, and $A^\dagger$ the range, null space, conjugate transpose, and Moore-Penrose generalized inverse of A, respectively. It is well known that $A^\dagger = A^{-1}$ for any nonsingular matrix A.

Let $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$. We consider the solution to the following system of linear equations

$$Ax = b. \tag{1.1}$$

When A is nonsingular, the system of linear equations (1.1) has a unique solution $A^{-1}b$. In general, the system may not have a unique solution or there may be no solution at all. We are interested in those x, called the "least-squares" solutions, which minimize

$$\|b - Ax\|_2.$$

Among all the least-squares solutions, there is a unique x^* such that $\|x^*\|_2 < \|x\|_2$ for any other least-squares solution x. The x^* is called the minimum-norm least-squares (MNLS) solution to (1.1). It is shown [2, Theorem 2.1.1] that $A^\dagger b$ is the the MNLS solution to (1.1).

*Received by the editors on November 22, 2010. Accepted for publication on May 8, 2011. Handling Editor: Daniel Szyld.

†Department of Computer Science, Fitchburg State University, Fitchburg, MA 01720, USA (xchen@fitchburgstate.edu).

‡Department of Mathematics and Statistics, Kennesaw State University, Kennesaw, GA 30144, USA (jji@kennesaw.edu).

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

ELA

For a nonsingular matrix B, $B + cd^*$ is also nonsingular if $1 + d^*B^{-1}c \neq 0$ for vectors c and d. Moreover, we have

$$(B + cd^*)^{-1} = B^{-1} - \frac{B^{-1}cd^*B^{-1}}{1 + d^*B^{-1}c}. \tag{1.2}$$

The identity in (1.2) is the well-known Sherman-Morrison formula for the inverse of rank-one perturbed matrix and was later generalized to the case of rank-k perturbation. The Moore-Penrose inverse of rank-one modified matrix $B + cd^*$ for a general matrix B is also obtained in literature [2, Theorem 3.1.3]. More generalizations can be found in [3], [12] and [13].

Based on Sherman-Morrison formula, a recursive iterative scheme for the solution to the system of linear equations $Ax = b$ with a nonsingular A was recently developed by Maponi [10]. Unfortunately, his algorithm breaks down even for a very simple problem (see Example 2 of Section 4) and a modification based on pivoting techniques has to be introduced to avoid the failure. In this paper, following the same idea of using rank-one updates employed by Maponi, we will develop a finite method which is guaranteed to carry out all the steps until the required solution is reached.

Our approach is to construct a sequence of linear systems in such a way that the coefficient matrix of any linear system in the sequence is just a symmetric rank-one modification of that of its previous system. By employing the rank-one update formula for generalized inverse, the MNLS solutions to all systems of linear equations in the sequence can be successively computed. Like the algorithms in [9, 11], there is no need to compute the generalized inverses of all the intermediate matrices in the procedure.

The proposed algorithm is different from the classic Greville finite algorithm [6] and its variant [15] in several aspects. One difference is that the algorithm in this paper is not "streamlined" while the other two algorithms are. However, the major difference, which makes our algorithm better than the algorithm in [15], lies in the way of handling the intermediate matrix sequences durng the iteration. We observe that the intermediate sequence P_l of [15] for the MNLS solution is indeed $A_l^\dagger$, where A_l is defined as in (2.4). Unlike the algorithm by Zhou et al. in which both P_i and Q_i of [15, Theorem 2.2] must be updated and matrix multiplication must be performed at each iteration, the sequence $\{A_l^\dagger\}$ in our notations is never computed explicitly even though the Sherman-Morrison type of formula for $\{A_l^\dagger\}$ is used in the derivation. Thus, our algorithm utilizes less required memory locations and has a lower computational complexity than the classic Greville algorithm and the algorithm by Zhou et al. [15]. The strategy of symmetric rank-one updating is also adopted in the online and real-time kernel recursive least-squares algorithm of [4].

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

The rest of the paper is organized as follows. The existing results on Moore-Penrose inverse of symmetric rank-one perturbed matrix will be summarized in Section 2. An effective finite method for computing the MNLS solution of a general linear system will be proposed in Section 3. Some examples will also be included in Section 4.

2. Symmetric rank-one update and its M-P inverse. Notice that the general linear system (1.1) is equivalent to the normal equation

$$A^*Ax = A^*b \tag{2.1}$$

so far as the MNLS solution is concerned.

Let $a_i \in \mathbb{C}^n$ be the ith column of the matrix A^*. Thus, we have

$$A^* = (a_1|a_2|\cdots|a_m)\,. \tag{2.2}$$

Therefore, we can rewrite the normal equation (2.1) as the following equivalent linear system

$$\left(\sum_{i=1}^{m} a_i a_i^*\right) x = A^*b. \tag{2.3}$$

Obviously, the problem of obtaining the MNLS solution of linear system (1.1) becomes that of obtaining the Minimum-norm solution of linear system (2.3) whose coefficient matrix is the sum of m rank-one matrices.

Define

$$\hat{b} = A^*b,\ A_0 = 0, \quad \text{and} \quad A_l = \sum_{i=1}^{l} a_i a_i^* \quad \text{for } l = 1, \ldots, m. \tag{2.4}$$

Our approach is to construct a finite sequence

$$x_l = A_l^\dagger \hat{b}, \quad l = 0, 1, 2, \ldots, m, \tag{2.5}$$

where x_l is the MNLS solution of $A_l x = \hat{b}$ $(l = 0, 1, 2, \ldots, m)$. However, we will not construct the sequence (2.5) by directly obtaining the MNLS solution for each linear system from cold-start. Note that $A_l = A_{l-1} + a_l a_l^*$ is the rank-one modification of A_{l-1}, we would compute x_l in terms of x_{l-1} starting from $x_0 = A_0^\dagger \hat{b}$. We hope that the calculation of x_0 should be an easy one. In fact, the first linear system $A_0 x = \hat{b}$ of the sequence has a trivial MNLS solution $x_0 = A_0^\dagger \hat{b} = 0$. Consequently, the sequence $\{x_l\}_{l=0}^m$ can be constructed recursively and effectively. After m iterations, $x_m = A_m^\dagger \hat{b}$ will be finally reached. We note that the linear system (2.1) is the same as $A_m x = \hat{b}$.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

Therefore,

$$x_m = A_m^\dagger \hat{b} = \left(\sum_{i=1}^{m} a_i a_i^* \right)^\dagger A^* b = (A^*A)^\dagger A^* b = A^\dagger b,$$

which is the MNLS solution of (1.1).

Observe that $A_l = A_{l-1} + a_l a_l^*$. We have $x_l = A_l^\dagger \hat{b} = (A_{l-1} + a_l a_l^*)^\dagger \hat{b}$. By utilizing the existing results of Moore-Penrose inverse of rank-one modified matrix A_l and exploring its special structure, we will propose an effective method for computing the MNLS solution of the linear system $A_l x = \hat{b}$ from that of $A_{l-1} x = \hat{b}$ in the next section. To this end, we need to establish a relation between $(A_{l-1} + a_l a_l^*)^\dagger$ and $A_{l-1}^\dagger$.

For each $l = 1, 2, \ldots, m$, define

$$k_l = A_{l-1}^\dagger a_l, \quad h_l = a_l^* A_{l-1}^\dagger, \quad u_l = (I - A_{l-1} A_{l-1}^\dagger) a_l,$$

$$v_l = a_l^* (I - A_{l-1}^\dagger A_{l-1}), \quad \text{and} \quad \beta_l = 1 + a_l^* A_{l-1}^\dagger a_l.$$

THEOREM 2.1. *Let A_l ($l = 0, 1, 2, \ldots, m$) be defined as in (2.4), A be partitioned as in (2.2), and $\beta_l = 1 + a_l^* A_{l-1}^\dagger a_l$. Then $\beta_l \geq 1$ for each l.*

Proof. Obviously, $\beta_1 = 1 + a_1^* A_0^\dagger a_1 = 1$. We now only need to prove the result for $2 \leq l \leq m$. Observe that $A_l = \sum_{i=1}^{l} a_i a_i^*$ is positive semidefinite for $1 \leq l \leq m$. Let r be the rank of A_{l-1}. Then A_{l-1} is unitarily similar to a diagonal matrix, i.e.,

$$A_{l-1} = UDU^*,$$

where U is a unitary matrix and $D = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. We can write $A_{l-1}^\dagger = UD^\dagger U^*$ which is also positive semidefinite due to the fact that $D^\dagger = \text{diag}(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_r, 0, \ldots, 0)$. Hence, we have $\beta_l = 1 + a_l^* A_{l-1}^\dagger a_l \geq 1$. $\square$

The general result for the Moore-Penrose inverse of a rank-one modified matrix can be found in [2, Theorem 3.1.3]. For general $A \in \mathbb{C}^{m \times n}$, $c \in \mathbb{C}^m$, $d \in \mathbb{C}^n$, the Moore-Penrose inverse of $A + cd^*$ can be expressed in terms of $A^\dagger$, c, and d with six distinguished cases. Due to the six distinct conditions and expressions, it is extremely difficult, if not impossible to apply the result for a general matrix directly to construct a recursive scheme for solving the MNLS solution of a linear system. However, when applied to our specially structured sequence (2.5), in view of Theorem 2.1, β_l is real and nonzero which eliminates three cases. Due to the fact that $\left(A_{l-1}^\dagger\right)^* = \left(A_{l-1}^*\right)^\dagger = A_{l-1}^\dagger$, we also have $h_l = k_l^*$ and $v_l = u_l^*$ from which two of

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

three other cases can be combined into one. Thus, the six cases of [2, Theorem 3.1.3] are reduced to only two cases.

THEOREM 2.2. *Let $A_l = A_{l-1} + a_l a_l^*$ be defined as in (2.4).*

1. *If $u_l \neq 0$, then*

$$(2.6)\qquad \begin{aligned}(A_{l-1} + a_l a_l^*)(A_{l-1} + a_l a_l^*)^\dagger &= (A_{l-1} + a_l a_l^*)^\dagger (A_{l-1} + a_l a_l^*)\\ &= A_{l-1}A_{l-1}^\dagger + u_l u_l^\dagger\end{aligned}$$

and

$$(A_{l-1} + a_l a_l^*)^\dagger = A_{l-1}^\dagger - k_l u_l^\dagger - u_l^{*\dagger} k_l^* + \beta_l u_l^{*\dagger} u_l^\dagger.$$

2. *If $u_l = 0$, then*

$$(2.7)\qquad \begin{aligned}(A_{l-1} + a_l a_l^*)(A_{l-1} + a_l a_l{}^*)^\dagger &= (A_{l-1} + a_l a_l^*)^\dagger (A_{l-1} + a_l a_l^*)\\ &= A_{l-1}A_{l-1}^\dagger = A_{l-1}^\dagger A_{l-1}\end{aligned}$$

and

$$(A_{l-1} + a_l a_l^*)^\dagger = A_{l-1}^\dagger - \frac{1}{\beta_l} k_l k_l^*.$$

Proof. The result follows directly from [2, Theorem 3.1.3] and its proof. Details are omitted. □

3. The finite method for the MNLS solution. Though the rank-one modified formulas stated in Theorem 2.2 involve the Moore-Penrose inverses of matrices, our method to be proposed will generate a sequence of vectors $x_1, x_2, \ldots, x_m$ recursively without explicitly computing the Moore-Penrose generalized inverses of all the intermediate matrices A_l $(l = 1, \ldots, m)$. To establish the iterative scheme from x_{l-1} to x_l, we first need to define two auxiliary sequences of vectors $y_{s,t} = A_s^\dagger a_t$ and $\tilde{y}_{s,t} = A_s y_{s,t} = A_s A_s^\dagger a_t$. It is easily seen from

$$A_s^\dagger A_s = \left(A_s^\dagger A_s\right)^* = A_s^* \left(A_s^\dagger\right)^* = A_s \left(A_s^*\right)^\dagger = A_s A_s^\dagger$$

that $y_{s,t}$ and $\tilde{y}_{s,t}$ are the MNLS solutions of the following two auxiliary linear systems respectively

$$A_s y = a_t, \quad A_s \tilde{y} = A_s a_t.$$

In what follows, we will frequently employ the fact that $a^\dagger = a^*/||a||^2$ for any non-zero column vector a. We will distinguish two cases in our analysis in view of Theorem 2.2.

Case 1 $(u_l \neq 0)$. It is easily seen from Theorem 2.2 that

$$(3.1)\qquad x_l \;=\; (A_{l-1} + a_l a_l^*)^\dagger \hat{b} \;=\; x_{l-1} - k_l u_l{}^\dagger \hat{b} - u_l{}^{*\dagger} k_l^* \hat{b} + \beta_l u_l{}^{*\dagger} u_l{}^\dagger \hat{b}.$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

Note that

$$
\begin{aligned}
k_l {u_l}^\dagger \hat{b} &= A_{l-1}^\dagger a_l[(I - A_{l-1}A_{l-1}^\dagger)a_l]^\dagger \hat{b} \\
&= \tfrac{A_{l-1}^\dagger a_l(a_l^* - \tilde{y}_{l-1,l}^*)\hat{b}}{a_l^*\left(a_l - A_{l-1}A_{l-1}^\dagger a_l\right)} \\
&= \tfrac{y_{l-1,l}(a_l^*\hat{b} - \tilde{y}_{l-1,l}^*\hat{b})}{a_l^*(a_l - \tilde{y}_{l-1,l})}.
\end{aligned}
\tag{3.2}
$$

Similarly, we have

$$
{u_l}^{*\dagger} k_l^* \hat{b} = \frac{a_l^* x_{l-1}\left(a_l - \tilde{y}_{l-1,l}\right)}{a_l^*\left(a_l - \tilde{y}_{l-1,l}\right)},
$$

and

$$
\begin{aligned}
\beta_l {u_l}^{*\dagger} {u_l}^\dagger \hat{b} &= (1 + a_l^* A_{l-1}^\dagger a_l)\tfrac{(a_l - \tilde{y}_{l-1,l})(a_l^* - \tilde{y}_{l-1,l}^*)\hat{b}}{\left[a_l^*(a_l - \tilde{y}_{l-1,l})\right]^2} \\
&= \tfrac{(1 + a_l^* y_{l-1,l})(a_l^* - \tilde{y}_{l-1,l}^*)\hat{b}}{\left[a_l^*(a_l - \tilde{y}_{l-1,l})\right]^2}\left(a_l - \tilde{y}_{l-1,l}\right).
\end{aligned}
\tag{3.3}
$$

Combining (3.1) through (3.3), we have

$$
\begin{aligned}
x_l &= x_{l-1} - \tfrac{y_{l-1,l}(a_l^*\hat{b} - \tilde{y}_{l-1,l}^*\hat{b})}{a_l^*(a_l - \tilde{y}_{l-1,l})} - \tfrac{a_l^* x_{l-1}(a_l - \tilde{y}_{l-1,l})}{a_l^*(a_l - \tilde{y}_{l-1,l})} \\
&\quad + \tfrac{(1 + a_l^* y_{l-1,l})(a_l - \tilde{y}_{l-1,l})^*\hat{b}}{\left[a_l^*(a_l - \tilde{y}_{l-1,l})\right]^2}\left(a_l - \tilde{y}_{l-1,l}\right).
\end{aligned}
\tag{3.4}
$$

Following the same lines as in (3.2)–(3.3), we can write

$$
k_l {u_l}^\dagger a_t = \frac{y_{l-1,l} a_l^*(a_t - \tilde{y}_{l-1,t})}{a_l^*(a_l - \tilde{y}_{l-1,l})},
$$

$$
{u_l}^{*\dagger} k_l^* a_t = \frac{(a_l - \tilde{y}_{l-1,l}) a_l^* y_{l-1,t}}{a_l^*\left(a_l - \tilde{y}_{l-1,l}\right)},
$$

$$
\beta_l {u_l}^{*\dagger} {u_l}^\dagger a_t = \frac{(1 + a_l^* y_{l-1,l})}{\left[a_l^*\left(a_l - \tilde{y}_{l-1,l}\right)\right]^2}(a_l - \tilde{y}_{l-1,l}) a_l^*(a_t - \tilde{y}_{l-1,t}).
$$

It is seen from Theorem 2.2 that

$$
y_{l,t} = A_l^\dagger a_t = (A_{l-1} + a_l a_l^*)^\dagger a_t = (A_{l-1}^\dagger - k_l {u_l}^\dagger - {u_l}^{*\dagger} k_l^* + \beta_l {u_l}^{*\dagger} {u_l}^\dagger) a_t.
$$

Thus, we have an iterative scheme for $y_{l,t}$:

$$
\begin{aligned}
y_{l,t} &= y_{l-1,t} - \tfrac{y_{l-1,l} a_l^*(a_t - \tilde{y}_{l-1,t})}{a_l^*(a_l - \tilde{y}_{l-1,l})} - \tfrac{(a_l - \tilde{y}_{l-1,l}) a_l^* y_{l-1,t}}{a_l^*(a_l - \tilde{y}_{l-1,l})} \\
&\quad + \tfrac{(1 + a_l^* y_{l-1,l})}{\left[a_l^*(a_l - \tilde{y}_{l-1,l})\right]^2}(a_l - \tilde{y}_{l-1,l}) a_l^*(a_t - \tilde{y}_{l-1,t}).
\end{aligned}
\tag{3.5}
$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

For the auxiliary sequence $\tilde{y}_{l,t} = A_l y_{l,t}$, we could multiply A_l on both sides of (3.5) and then simplify the resulted expression. However, in our derivation we employ (2.6) instead:

$$\begin{aligned} \tilde{y}_{l,t} &= A_l A_l^\dagger a_t = \left(A_{l-1}A_{l-1}^\dagger + u_l u_l^\dagger\right) a_t \\ &= \tilde{y}_{l-1,t} + u_l u_l^\dagger a_t \\ &= \tilde{y}_{l-1,t} + \tfrac{\left(I-A_{l-1}A_{l-1}^\dagger\right)a_l a_l^*\left(I-A_{l-1}A_{l-1}^\dagger\right)a_t}{a_l^*\left(I-A_{l-1}A_{l-1}^\dagger\right)a_l}. \end{aligned}$$

Therefore, we have

$$\tilde{y}_{l,t} = \tilde{y}_{l-1,t} + \frac{\left(a_l - \tilde{y}_{l-1,l}\right) a_l^* \left(a_t - \tilde{y}_{l-1,t}\right)}{a_l^* \left(a_l - \tilde{y}_{l-1,l}\right)}. \tag{3.6}$$

Case 2 ($u_l = 0$). Observe that

$$\beta_l = 1 + a_l^* A_{l-1}^\dagger a_l = 1 + a_l^* y_{l-1,l} \tag{3.7}$$

and

$$k_l k_l^* \hat{b} = y_{l-1,l} a_l^* x_{l-1}. \tag{3.8}$$

It is seen from Theorem 2.2 that

$$x_l = (A_{l-1} + a_l a_l^*)^\dagger \hat{b} = \left(A_{l-1}^\dagger - \frac{1}{\beta_l} k_l k_l^*\right) \hat{b} = x_{l-1} - \frac{1}{\beta_l} k_l k_l^* \hat{b},$$

which, together with (3.7) and (3.8), implies

$$x_l = x_{l-1} - \frac{1}{1 + a_l^* y_{l-1,l}} y_{l-1,l} a_l^* x_{l-1}. \tag{3.9}$$

By following the same token, we can develop an iterative scheme for $y_{l,t}$:

$$y_{l,t} = y_{l-1,t} - \frac{1}{1 + a_l^* y_{l-1,l}} y_{l-1,l} a_l^* y_{l-1,t}. \tag{3.10}$$

For $\tilde{y}_{l,t}$, in view of (2.7), we have

$$\tilde{y}_{l,t} = \tilde{y}_{l-1,t}. \tag{3.11}$$

Finally, since $A_0 = 0$, we have $x_0 = 0$, $y_{0,t} = 0$, and $\tilde{y}_{0,t} = 0$ ($t = 1, 2, \ldots, m$) from which we can compute the MNLS solution x_m by applying (3.4) or (3.9) repeatedly with the help of auxiliary sequences $y_{l,t}$ and $\tilde{y}_{l,t}$ ($t = l+1, l+2, \ldots, m$ and $l < m$). We summarize this procedure as follows.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

ELA

Procedure for the MNLS solution of $Ax = b$.

Step 0 Input: b and A. Let a_i be the ith column of A^*;
Step 1 Initialization: compute $\hat{b} = A^*b$. Set $x_0 = 0$, $y_{0,t} = 0$ and $\tilde{y}_{0,t} = 0$ for all $t = 1, \ldots, m$;
Step 2 For $l = 1, 2, \ldots, m$,
(a) if $a_l - \tilde{y}_{l-1,l} \neq 0$, then
compute x_l using (3.4); compute $y_{l,t}$ and $\tilde{y}_{l,t}$ using (3.5) and (3.6) respectively for all $t = l+1, l+2, \ldots, m$ and $l < m$;
(b) if $a_l - \tilde{y}_{l-1,l} = 0$, then
compute x_l using (3.9); compute $y_{l,t}$ and $\tilde{y}_{l,t}$ using (3.10) and (3.11) respectively for all $t = l+1, l+2, \ldots, m$ and $l < m$;
Step 3 Output: the minimum-norm least squares solution x_m.

Let us analyze two cases in our method further. If $u_l = 0$ for some l, then we have

$$a_l = A_{l-1}A_{l-1}^{\dagger}a_l = \left(\sum_{i=1}^{l-1} a_i a_i^*\right) A_{l-1}^{\dagger}a_l = \sum_{i=1}^{l-1}(a_i^* A_{l-1}^{\dagger}a_l)a_i.$$

This means that if $u_l = 0$ for some l, a_l is a linear combination of $\{a_i :\ i = 1, 2, \ldots, l-1\}$. Now, we end up this section with the following interesting result which shows that the opposite is also true.

THEOREM 3.1. *Let u_i and a_i ($i = 1, 2, \ldots, m$) be defined as before. Then, $u_l = 0$ if and only if a_l is a linear combination of $\{a_1, a_2, \ldots, a_{l-1}\}$.*

Proof. The definition of $u_l = (I - A_{l-1}A_{l-1}^{\dagger})a_l$ indicates that $u_l = 0$ is equivalent to $a_l \in N(I - A_{l-1}A_{l-1}^{\dagger})$. Let $\hat{A}_{l-1} = [a_1|a_2|\cdots|a_{l-1}]$. Obviously, we have $A_{l-1} = \hat{A}_{l-1}\hat{A}_{l-1}^*$. It is easily seen that

$$R(\hat{A}_{l-1}) = R(\hat{A}_{l-1}\hat{A}_{l-1}^*) = R(A_{l-1}) = R(A_{l-1}A_{l-1}^{\dagger}) = N(I - A_{l-1}A_{l-1}^{\dagger}).$$

Therefore, $u_l = 0$ is equivalent to $a_l \in R(\hat{A}_{l-1})$, i.e., a_l is a linear combination of $\{a_1, a_2, \ldots, a_{l-1}\}$. $\square$

As a consequence of Theorem 3.1, if A has full row rank, then A^* has full column rank which implies that $\{a_1, a_2, \ldots, a_l\}$ is linearly independent for all l. Thus $u_l \neq 0$ for all l in this case.

Although the procedure for the MNLS solution does not impose any restriction on the rows of matrix A, we believe that it will be better off if we could first prune linearly dependent rows of A as was discussed in [7] and then apply the procedure. By doing so we will actually remove the cases of $u_l = 0$ and thus will be able to reduce computational error and possibly storage as well as computational complexities.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

4. Some examples. From the derivation of our method, we see that we have never formed the A_l explicitly. Also, this method is free of matrix multiplication. One major advantage of this recursive method is that it will always carry out a result successfully. The method provides an alternative way for directly finding MNLS solution to a system of linear equations.

We illustrate our method with three examples.

Example 1. Let

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ and } b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{4.1}$$

By using our method, we generate a sequence of vectors

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \; x_1 = \begin{bmatrix} 23/98 \\ 23/49 \\ 69/98 \end{bmatrix}, \; x_2 = \begin{bmatrix} -1/2 \\ 0 \\ 1/2 \end{bmatrix}.$$

x_2 is the minimum-norm solution to the system of linear equations (4.1).

Example 2. Let

$$A = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 0 \\ -1 & 0 & -1 \end{bmatrix} \text{ and } b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{4.2}$$

This time, we have

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \; x_1 = \begin{bmatrix} 5/9 \\ 5/9 \\ -5/9 \end{bmatrix}, \; x_2 = \begin{bmatrix} -1/4 \\ -1/4 \\ 5/2 \end{bmatrix}, \; x_3 = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}.$$

x_3 is the solution to the system of linear equations (4.2). Note this is a nonsingular matrix which is taken from [10]. As was pointed out by the author, Algorithm 1 in [10] breaks down and a modification based on pivoting techniques has to be introduced to avoid the failure. However, as we can see, our method solves it easily.

In either Examples 1 or 2, A has full row rank. The sequence $\{x_i\}$ is indeed constructed by our method without calling Step 2(b) which confirms Theorem 3.1. In the following example, both Step 2(a) and Step 2(b) are called up.

Example 3. Let

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \text{ and } b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{4.3}$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 22, pp. 480-489, May 2011

ELA

By using our method, we generate a sequence of vectors

$$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ x_1 = \begin{bmatrix} 33/25 \\ 66/25 \end{bmatrix},\ x_2 = \begin{bmatrix} 3 \\ -3/2 \end{bmatrix},\ x_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

x_3 is the MNLS solution to the system of linear equations (4.3). In this example, we do observe that $u_1 \neq 0, u_2 \neq 0$ but $u_3 = 0$, i.e., x_3 is computed in Step 2(b).

REFERENCES

[1] A. Bjorck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[2] S.L. Campbell and C.D. Meyer. *Generalized Inverses of Linear Transformations*. Pitman, London, 1979.

[3] X. Chen. The generalized inverses of perturbed matrices. *Int. J. Comput. Math*, 41:223–236, 1992.

[4] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process*, 52:2275–2285, 2004.

[5] T.N.E. Greville. The pseudo-inverse of a rectangular or singular matrix and its application to the solution of systems of linear equations. *SIAM Rev.*, 1:38–43, 1959.

[6] T.N.E. Greville. Some applications of the pseudoinverse of a matrix. *SIAM Rev.*, 2:15–22, 1960.

[7] V. Lakshmikantham, S.K. Sen, and S. Sivasundaram. Concise row-pruning algorithm to invert a matrix. *Appl. Math. Comput.*, 60:17-24, 1994.

[8] V. Lakshmikantham and S.K. Sen. *Computational Error and Complexity in Science and Engineering*. Elsevier, Amsterdam, 2005.

[9] E.A. Lord, V.Ch. Venkaiah, and S.K. Sen. A concise algorithm to solve under-/over-determined linear systems. *Simulation*, 54:239–240, 1990.

[10] P. Maponi. The solution of linear systems by using the Sherman-Morrison formula. *Linear Algebra Appl.*, 420:276–294, 2007.

[11] S.K. Sen and S. Sen. $O(n^3)$ g-inversion-free noniterative near-consistent linear system solver for minimum-norm least-squares and nonnegative solutions. *J. Comput. Methods Sci. Eng.*, 6:71–85, 2006.

[12] S.K. Sen and E.V. Krishnamurthy. Rank-augmented LU-algorithm for computing Moore-Penrose matric inverses. *IEEE Trans. Computers*, C-23, 199–201, 1974.

[13] S.R. Vatsya and C.C. Tai. Inverse of a perturbed matrix. *Int. J. Comput. Math*, 23:177–184, 1988.

[14] G.R. Wang, Y.M. Wei, and S. Qiao. *Generalized Inverses: Theory and Computations*. Science Press, Beijing, 2004.

[15] J. Zhou, Y. Zhu, X.R. Li, and Z. You. Variants of the Greville formula with applications to exact recursive least squares. *SIAM J. Matrix Anal. Appl.*, 24:150–164, 2002.