

Research Article

A Fast Algorithm of Moore-Penrose Inverse for the Loewner-Type Matrix

QiuJuan Tong,^{1,2} SanYang Liu,¹ and Quan Lu³

¹ School of Science, Xidian University, Xi'an 710071, China

² School of Science, Xi'an University of Post and Telecommunications, Xi'an 710061, China

³ Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to QiuJuan Tong, xiaotong@xupt.edu.cn

Received 25 July 2011; Accepted 8 October 2011

Academic Editor: Tadeusz Kaczorek

Copyright © 2012 QiuJuan Tong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we present a fast algorithm of Moore-Penrose inverse for $m \times n$ Loewner-type matrix with full column rank by forming a special block matrix and studying its inverse. Its computation complexity is $O(mn) + O(n^2)$, but it is $O(mn^2) + O(n^3)$ by using $L^+ = (L^T L)^{-1} L^T$.

1. Introduction

Loewner matrix was first studied by Loewner in 1934 in [1]. He studied the relations of various Loewner matrices via the characteristic of the monotone matrix function and the problem of rational interpolation at that time. Since then, more further studies had been carried out by many scientists in [2–5], such as Belevitch, Donoghue Jr., Fiedler, Chen and Zhang. From their mentioned works, we can find various properties of Loewner matrix and its application in the rational interpolation. In 1984, Vavřín presented a fast algorithm for the inverse of Loewner matrix in [6], then the fast algorithm for the Loewner system was got accordingly. Rost and Vavřín presented a fast algorithm for the system whose coefficient is a Loewner-Vandermonde matrix from 1995 to 1996 in [7, 8]. Lu gave a fast triangular factorization algorithm for the symmetrical Loewner-type matrix in 2003 in [9]. Xu et al. and so forth gave a fast triangular factorization algorithm for the inverse of symmetrical Loewner-type matrix in 2003 in [10]. In 2009, Tong et al. gave a fast algorithm of the Moore-Penrose inverse for $m \times n$ symmetrical Loewner-type matrix [11]. In this paper, we generalize the fast algorithm of symmetrical Loewner-type matrix to the Loewner-type matrix. The theory and computation of generalized inverse matrix arise in various applications such as optimization theory, control theory, computation mathematics, and mathematical statistics.

Therefore, there are very important theoretical and practical significance when we study the fast algorithm of Moore-Penrose inverse for the Loewner-type matrix.

This paper is organized as follows: in Section 1, we present some preliminaries results. The fast algorithm of Moore-Penrose inverse for Loewner-type matrix is driven in Section 2. We give some numerical examples to illustrate the fast algorithm obtained in Section 3.

2. Preliminaries

A Loewner matrix \mathbf{L}_1 is a matrix of the form $\mathbf{L}_1 = (l_{ij})_{m \times n} = ((\xi_i - \eta_j) / (\alpha_i - \beta_j))_{i,j=1}^{m,n}$, where α_i , β_j , ξ_i , and η_j ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$) are given numbers and $\alpha_i \neq \beta_j$. A Loewner-type matrix \mathbf{L} is a matrix which satisfies

$$\tilde{\mathbf{D}}\mathbf{L} - \mathbf{L}\mathbf{D} = \sum_{j=1}^l \mathbf{p}^{(j)} \mathbf{q}^{(j)\top}, \quad (2.1)$$

where $\tilde{\mathbf{D}} = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_m)$, $\mathbf{D} = \text{diag}(\beta_1, \beta_2, \dots, \beta_n)$, $\mathbf{p}^{(j)} = (p_1^{(j)}, p_2^{(j)}, \dots, p_m^{(j)})^\top$, and $\mathbf{q}^{(j)} = (q_1^{(j)}, q_2^{(j)}, \dots, q_n^{(j)})^\top$. Loewner matrix \mathbf{L}_1 is a special case of Loewner-type matrix, and it satisfies $\tilde{\mathbf{D}}\mathbf{L}_1 - \mathbf{L}_1\mathbf{D} = (\xi_1, \dots, \xi_m)^\top (1, \dots, 1) - (1, \dots, 1)^\top (\eta_1, \dots, \eta_n)$.

Let n be the rank of $m \times n$ Loewner-type matrix \mathbf{L} . Forming an $m + n$ matrix

$$\mathbf{M} = \begin{pmatrix} -\mathbf{I}_m & \mathbf{L} \\ \mathbf{L}^\top & \mathbf{0} \end{pmatrix}, \quad (2.2)$$

we obtain

$$\mathbf{M}^{-1} = \begin{pmatrix} \mathbf{L}(\mathbf{L}^\top \mathbf{L})^{-1} \mathbf{L}^\top - \mathbf{I}_m & \mathbf{L}(\mathbf{L}^\top \mathbf{L})^{-1} \\ (\mathbf{L}^\top \mathbf{L})^{-1} \mathbf{L}^\top & (\mathbf{L}^\top \mathbf{L})^{-1} \end{pmatrix} = \begin{pmatrix} \mathbf{L}\mathbf{L}^+ - \mathbf{I}_m & (\mathbf{L}^\top)^+ \\ \mathbf{L}^+ & (\mathbf{L}^\top \mathbf{L})^{-1} \end{pmatrix}. \quad (2.3)$$

It is obvious that we may obtain the Moore-Penrose inverse \mathbf{L}^+ of Loewner-type matrix \mathbf{L} by (2.3) if we can get \mathbf{M}^{-1} .

Now let us begin to find \mathbf{M}^{-1} .

The following result will be useful for getting \mathbf{M}^{-1} .

Lemma 2.1 (see [12]). *Let all the leading principal submatrices \mathbf{S}_i ($i = 1, 2, \dots, n$) of matrix $\mathbf{S} = (s_{ij})_{n \times n}$ whose order is n be invertible. Linear system $\mathbf{S}\mathbf{x} = \mathbf{b}$ be given, where $\mathbf{b} = (b_1, b_2, \dots, b_n)^\top$. Note that $\mathbf{b}_i = (b_1, b_2, \dots, b_i)^\top$, and let $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ii})^\top$, $\mathbf{u}_i = (u_{1i}, u_{2i}, \dots, u_{ii})^\top$ be solution vectors of linear systems $\mathbf{S}_i \mathbf{x}_i = \mathbf{b}_i$, $\mathbf{S}_i \mathbf{u}_i = \mathbf{e}_i^{(i)}$ differently, then*

$$\mathbf{x}_i = \begin{pmatrix} \mathbf{x}_{i-1} \\ 0 \end{pmatrix} + \sigma_i \mathbf{u}_i, \quad (2.4)$$

where $\sigma_i = b_i - \sum_{j=1}^{i-1} s_{ij} x_{i-1,j}$.

3. Fast Algorithm of the Moore-Penrose Inverse for Loewner-Type Matrix

By using (2.1), we know that \mathbf{M} satisfies

$$\begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D} \end{pmatrix} \mathbf{M} - \mathbf{M} \begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D} \end{pmatrix} = \sum_{j=1}^l \left[\begin{pmatrix} \mathbf{p}^{(j)} \\ \mathbf{0}_n \end{pmatrix} (\mathbf{0}_m^T \mathbf{q}^{(j)T}) - \begin{pmatrix} \mathbf{0}_m \\ \mathbf{q}^{(j)} \end{pmatrix} (\mathbf{p}^{(j)T} \mathbf{0}_n^T) \right]. \quad (3.1)$$

Let all the leading principal submatrices \mathbf{M}_i ($i = 1, 2, \dots, m+n$) of \mathbf{M} be invertible. If $i > m$, by virtue of (3.1), we have

$$\begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D}_{i-m} \end{pmatrix} \mathbf{M}_i - \mathbf{M}_i \begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D}_{i-m} \end{pmatrix} = \sum_{j=1}^l \left[\begin{pmatrix} \mathbf{p}^{(j)} \\ \mathbf{0}_{i-m} \end{pmatrix} (\mathbf{0}_m^T \mathbf{q}_{i-m}^{(j)T}) - \begin{pmatrix} \mathbf{0}_m \\ \mathbf{q}_{i-m}^{(j)} \end{pmatrix} (\mathbf{p}^{(j)T} \mathbf{0}_{i-m}^T) \right]. \quad (3.2)$$

Let $\mathbf{g}_i^{(j)} = (g_{i1}^{(j)}, g_{i2}^{(j)}, \dots, g_{ii}^{(j)})^T$, $\mathbf{h}_i^{(j)} = (h_{i1}^{(j)}, h_{i2}^{(j)}, \dots, h_{ii}^{(j)})^T$ be solution vectors of linear systems

$$\mathbf{M}_i \mathbf{g}_i^{(j)} = \begin{pmatrix} \mathbf{p}^{(j)} \\ \mathbf{0}_{i-m} \end{pmatrix}, \quad \mathbf{M}_i \mathbf{h}_i^{(j)} = \begin{pmatrix} \mathbf{0}_m \\ \mathbf{q}_{i-m}^{(j)} \end{pmatrix} \quad (i = m, m+1, \dots, m+n) \quad (3.3)$$

differently, then using $\mathbf{M}_m = -\mathbf{I}_m$, we obtain $\mathbf{g}_m^{(j)} = -\mathbf{p}^{(j)}$, $\mathbf{h}_m^{(j)} = \mathbf{0}_m$. By virtue of Lemma 2.1 we have

$$\mathbf{g}_i^{(j)} = \begin{pmatrix} \mathbf{g}_{i-1}^{(j)} \\ 0 \end{pmatrix} + \sigma_i^{(j)} \mathbf{u}_i, \quad \mathbf{h}_i^{(j)} = \begin{pmatrix} \mathbf{h}_{i-1}^{(j)} \\ 0 \end{pmatrix} + \tau_i^{(j)} \mathbf{u}_i, \quad (3.4)$$

where $\sigma_i^{(j)} = -\sum_{k=1}^m l_{k,i-m} g_{i-1,k}^{(j)}$, $\tau_i^{(j)} = q_{i-m}^{(j)} - \sum_{k=1}^m l_{k,i-m} h_{i-1,k}^{(j)}$, and $\mathbf{u}_i = (u_{1i}, u_{2i}, \dots, u_{ii})^T$ is a solution vector of $\mathbf{M}_i \mathbf{u}_i = \mathbf{e}_i^{(i)}$. Multiplying (3.2) by \mathbf{M}_i^{-1} on the left and on the right differently, we obtain

$$\mathbf{M}_i^{-1} \begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D}_{i-m} \end{pmatrix} - \begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D}_{i-m} \end{pmatrix} \mathbf{M}_i^{-1} = \sum_{j=1}^l (\mathbf{g}_i^{(j)} \mathbf{h}_i^{(j)T} - \mathbf{h}_i^{(j)} \mathbf{g}_i^{(j)T}). \quad (3.5)$$

Multiplying (3.5) by $\mathbf{e}_i^{(i)}$ and noting that $\begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D}_{i-m} \end{pmatrix} \mathbf{e}_i^{(i)} = \beta_{i-m} \mathbf{e}_i^{(i)}$, we have

$$\beta_{i-m} \mathbf{u}_i - \begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D}_{i-m} \end{pmatrix} \mathbf{u}_i = \sum_{j=1}^l (h_{ii}^{(j)} \mathbf{g}_i^{(j)} - g_{ii}^{(j)} \mathbf{h}_i^{(j)}). \quad (3.6)$$

Substituting (3.4) in (3.6) and noting that $g_{ii}^{(j)} = \sigma_i^{(j)} u_{ii}$, $h_{ii}^{(j)} = \tau_i^{(j)} u_{ii}$, we have

$$\beta_{i-m} \mathbf{u}_i - \begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D}_{i-m} \end{pmatrix} \mathbf{u}_i = u_{ii} \sum_{j=1}^l \left(\tau_i^{(j)} \begin{pmatrix} \mathbf{g}_{i-1}^{(j)} \\ 0 \end{pmatrix} - \sigma_i^{(j)} \begin{pmatrix} \mathbf{h}_{i-1}^{(j)} \\ 0 \end{pmatrix} \right), \quad (3.7)$$

and hence,

$$\mathbf{u}_{ki} = \begin{cases} \frac{u_{ii} \sum_{j=1}^l (\tau_i^{(j)} \mathbf{g}_{i-1,k}^{(j)} - \sigma_i^{(j)} \mathbf{h}_{i-1,k}^{(j)})}{\beta_{i-m} - \alpha_k} & (k = 1, 2, \dots, m), \\ \frac{u_{ii} \sum_{j=1}^l (\tau_i^{(j)} \mathbf{g}_{i-1,k}^{(j)} - \sigma_i^{(j)} \mathbf{h}_{i-1,k}^{(j)})}{\beta_{i-m} - \beta_{k-m}} & (k = m+1, m+2, \dots, i-1). \end{cases} \quad (3.8)$$

Now, let us look for u_{ii} . Choosing the i th of $\mathbf{M}_i \mathbf{u}_i = \mathbf{e}_i^{(i)}$, namely

$$\sum_{k=1}^m l_{k,i-m} u_{ki}, \quad (3.9)$$

and using (3.8), we have

$$u_{ii} = \frac{1}{\sum_{k=1}^m l_{k,i-m} / (\beta_{i-m} - \alpha_k) \left[\sum_{j=1}^l (\tau_i^{(j)} \mathbf{g}_{i-1,k}^{(j)} - \sigma_i^{(j)} \mathbf{h}_{i-1,k}^{(j)}) \right]}. \quad (3.10)$$

Note that $\mathbf{M}^{-1} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m+n})$, where $\mathbf{v}_i = (v_{1i}, v_{2i}, \dots, v_{m+n,i})^T$ is the i th column of \mathbf{M}^{-1} , then

$$\mathbf{v}_{m+n} = \mathbf{u}_{m+n}. \quad (3.11)$$

Letting $i = m+n$ in (3.5) and multiplying it by $\mathbf{e}_k^{(m+n)}$ ($k > m$) on the right, we have

$$\beta_{k-m} \mathbf{v}_k - \begin{pmatrix} \tilde{\mathbf{D}} \\ \mathbf{D} \end{pmatrix} \mathbf{v}_k = \sum_{j=1}^l \left(h_{m+n,k}^{(j)} \mathbf{g}_{m+n}^{(j)} - g_{m+n,k}^{(j)} \mathbf{h}_{m+n}^{(j)} \right). \quad (3.12)$$

So, we obtain the fast algorithm of Moore-Penrose inverse \mathbf{L}^+ of Loewner-type matrix \mathbf{L} by (2.3), (3.4), and (3.8)~(3.12).

Algorithm

Step 1.

$$\mathbf{g}_m^{(j)} = -\mathbf{p}^{(j)}, \quad \mathbf{h}_m^{(j)} = \mathbf{0}_m \quad (j = 1, 2, \dots, l), \quad (3.13)$$

For $i = m + 1, m + 2, \dots, m + n$,

$$\begin{aligned}
\sigma_i^{(j)} &= -\sum_{k=1}^m l_{k,i-m} g_{i-1,k}^{(j)}, & \tau_i^{(j)} &= q_{i-m}^{(j)} - \sum_{k=1}^m l_{k,i-m} h_{i-1,k}^{(j)} \quad (j = 1, 2, \dots, l), \\
t_{ik} &= \sum_{j=1}^l \left(\tau_i^{(j)} g_{i-1,k}^{(j)} - \sigma_i^{(j)} h_{i-1,k}^{(j)} \right) \quad (k = 1, 2, \dots, i-1), \\
\lambda_i &= \sum_{k=1}^m \frac{l_{k,i-m}}{\beta_{i-m} - \alpha_k} t_{ik}, & u_{ii} &= \frac{1}{\lambda_i}, \\
u_{ki} &= \frac{u_{ii}}{\beta_{i-m} - \alpha_k} t_{ik} \quad (k = 1, 2, \dots, m), \\
u_{ki} &= \frac{u_{ii}}{\beta_{i-m} - \beta_{k-m}} t_{ik} \quad (k = m+1, m+2, \dots, i-1), \\
\mathbf{g}_i^{(j)} &= \begin{pmatrix} \mathbf{g}_{i-1}^{(j)} \\ 0 \end{pmatrix} + \sigma_i^{(j)} \mathbf{u}_i, & \mathbf{h}_i^{(j)} &= \begin{pmatrix} \mathbf{h}_{i-1}^{(j)} \\ 0 \end{pmatrix} + \tau_i^{(j)} \mathbf{u}_i \quad (j = 1, 2, \dots, l).
\end{aligned} \tag{3.14}$$

Step 2.

$$v_{i,m+n} = u_{i,m+n} \quad (i = 1, 2, \dots, m). \tag{3.15}$$

For $k = m + 1, \dots, m + n - 1$,

$$v_{ik} = \frac{1}{\beta_{k-m} - \alpha_i} \sum_{j=1}^l \left(g_{m+n,i}^{(j)} h_{m+n,k}^{(j)} - h_{m+n,i}^{(j)} g_{m+n,k}^{(j)} \right) \quad (i = 1, \dots, m), \tag{3.16}$$

then

$$\mathbf{L}^+ = \begin{pmatrix} v_{1,m+1} & \cdots & v_{m,m+1} \\ \vdots & & \vdots \\ v_{1,m+n} & \cdots & v_{m,m+n} \end{pmatrix}. \tag{3.17}$$

The algorithm requires $(7l-3)mn + ln^2 + O(m)$ multiplication and division operations and $(5l-3)mn + (l/2)n^2 + O(m)$ addition and subtraction operations, and the computation complexity is $O(mn) + O(n^2)$, but it is $O(mn^2) + O(n^3)$ by using $\mathbf{L}^+ = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$.

4. Numerical Examples

We get the Moore-Penrose inverse matrix of Loewner-type matrix with Fortran program in computer, and the results of the partial numerical examples are given as follows in Table 1 (the error is measured by the 2-norm of vector, and the time is measured by second).

Table 1

Orders of matrix		Fast algorithm		$\mathbf{L}^+ = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$	
m	n	Err.	Tim.	Err.	Tim.
10000	20	$7.254e-16$	2	$8.109e-14$	11
20000	20	$2.398e-15$	2	$2.788e-14$	31
30000	20	$2.019e-15$	5	$2.106e-14$	89
40000	20	$9.108e-14$	8	$1.001e-14$	201
60000	20	$4.901e-14$	36	$3.760e-14$	705

Example 1. Consider

$$\begin{aligned}
 \alpha_i &= \frac{(i-1)\pi}{m-n+1}, & \beta_j &= \frac{(j+1)\pi}{m+n-1}, & \xi_i &= (-1)^i(i-mn), & \eta_j &= j^{j-m}. \\
 p^{(1)}(i) &= \frac{\xi_i}{2}, & p^{(2)}(i) &= 1, & p^{(3)}(i) &= \frac{\xi_i}{2}, & p^{(4)}(i) &= 1; \\
 q^{(1)}(j) &= 1, & q^{(2)}(j) &= -\eta_j, & q^{(3)}(j) &= 1, & q^{(4)}(j) &= -\eta_j. \\
 & & & & i &= 1, 2, \dots, m; & j &= 1, 2, \dots, n.
 \end{aligned} \tag{4.1}$$

From above examples and many more examples not given, we find that the stability of the fast algorithm is very good, and it needs shorter time than that of $\mathbf{L}^+ = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$.

Acknowledgment

This work is being supported by the National Natural Science Foundation Grants of China (no. 60974082).

References

- [1] K. Loewner, "Über monotone matrixfunktionen," *Math*, vol. 38, pp. 177–216, 1934.
- [2] V. Belevitch, "Interpolation matrices," *Philips Research Report*, vol. 25, no. 5, pp. 337–369, 1970.
- [3] W. F. Donoghue Jr., *Monotone Matrix Functions and Analytic Continuation*, Springer, New York, NY, USA, 1974.
- [4] M. Fiedler, "Hankel and Loewner matrices," *Linear Algebra and Its Applications*, vol. 58, pp. 75–95, 1984.
- [5] G. N. Chen and H. P. Zhang, "More on Loewner matrices," *Linear Algebra and Its Applications*, vol. 203–204, pp. 265–300, 1994.
- [6] Z. Vavřín, "Inverses of Loewner matrices," *Linear Algebra and Its Applications*, vol. 63, pp. 227–236, 1984.
- [7] K. Rost and Z. Vavřín, "Recursive solution of Loewner-Vandermonde systems of equations. II," *Linear Algebra and Its Applications*, vol. 223–224, pp. 597–617, 1995.
- [8] K. Rost and Z. Vavřín, "Recursive solution of Loewner-Vandermonde systems of equations. I," *Linear Algebra and Its Applications*, vol. 233, pp. 51–65, 1996.
- [9] Q. Lu, "A fast algorithm for the triangular factorization of the symmetric Loewner type matrix," *Journal of Engineering Mathematics*, vol. 20, no. 2, pp. 139–142, 2003 (Chinese).
- [10] M. Xu, Z. Xu, and Z.-K. Shi, "A fast triangular factorization algorithm for the inverse of symmetric loewner-type matrix," *Journal of China Three Gorges University*, vol. 6, pp. 552–554, 2003 (Chinese).

- [11] Q. Tong, Q. Lu, S. Liu, and J. Chai, "A fast algorithm of Moore-Penrose inverse for the symmetric Loewner-type matrix," in *Proceedings of the International Conference on Information Engineering and Computer Science (ICIECS '09)*, pp. 2155–2158, December 2009.
- [12] Z. Xu, K.-Y. Zhang, and Q. Lu, *Fast Algorithm of Toeplitz-Like Matrix(in Chinese)*, Northwest Polytechnical University Press, Xi'an, China, 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

