ISSN: 2754-6705

Journal of Mathematical & Computer Applications

SCIENTIFIC Research and Community

Review Article

Open Access

A New Matrix Decomposition Method for Inverting the Comrade Matrix

B Talibi[1]*, A Aiat Hadj[2] and D Sarsri[1]

[1]Laboratory of Innovative Technologies, National School of Applied Sciences of Tangier, Abdelmalek Essaâdi University, B. P. 1818, Tangier, Morocco

[2]Regional Center of the Trades of Education and Training (CRMEF)-Tangier, Avenue My Abdelaziz, Souani, BP: 3117, Tangier. Morocco

ABSTRACT

Herein,we decompose the Comrade matrix based on its lower triangular L and triangular T factorization to calculate its inverse. We then show the proposed method is efficient with two algorithms.

***Corresponding author**

B Talibi, Laboratory of Innovative Technologies, National School of Applied Sciences of Tangier, Abdelmalek Essaâdi University, B. P. 1818, Tangier, Morocco.

Received: January 21, 2025; **Accepted:** January 24, 2025, **Published:** February 04, 2025

Keywords: Comrade Matrices, Decomposition, LT Factorization, Matrix Inverse

Introduction

In recent years a great deal of attention has been attracted to structured matrices in mathematics because they appear in numerous applications in engineering, physics and computational science. One of the most important classes of them, whichhave been found extremely useful to solve wide range of complex problems is Comrade matrices. Due to their unique structure that allows computing, they are essential in manydomains such as systems theory, polynomial computations, numerical analysis, etc.

Comrade matrices are a generalisation of manyother familiar structured matrices (in particular, arrowhead matrices and tridiagonal matrices), and are also a special instance of bordered ma- trices. This is a richer theoretical relationship that widens their circle of applicability. Comrade matrix is frequently used in systems theory as a means to help model dynamic systems and study their stability. In polynomial computations they are used to divide generalized polynomials and they provide a nice setting for problems of eigenvalues.

While they have such general usefulness, the study of Comrade matrices has been left relatively unexplored compared to the simpler case. We hope to contribute to this new field by providing new results and methods that exploit their structural properties. In particular, we present a decomposition image model using LT factorization, which enables fast inverse computation. This is a major step toward theorical and practical use of quantum computers, including applications such as solving linear systems and alkorithms for polynon biniding.

Moreover, we conduct a comparison of our approach to existing methods to assess the performance of our proposed method, including the Toeplitz Hessenberg algorithm and the LT decomposition method. We further analyze its performance in detail to illustrate the benefits of our method, in terms of the running time.

The structure of the rest of this paper is as follows, Section 2 describes the decomposition LT of the comrade matrix. After proposing our decomposition method in Section 3, we present various decomposition $T+K$ for it in Sections 4-5-6. Numerical experiments and a comparison with other algorithms are presented in Section7.

A New Decomposition LT for the Comrade Matrix

Consider the comrade matrix of the form:

$$C_n = \begin{pmatrix} a_0 & b_1 & c_1 & c_2 & \cdots & c_{n-1} \\ d_1 & a_1 & b_2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & d_{n-2} & a_{n-2} & b_{n-1} \\ 0 & & \cdots & 0 & d_{n-1} & a_{n-1} \end{pmatrix} \quad (1)$$

If the matrix Cn is multiplied by PC_nP, we convert the matrix C_n in its first form into the form C_n with δ_i on last row, where:

$P = (pij)$ and $pij = 1$, if $i + j = n$, and zero otherwise.

And C'_n take the form:

$$C'_n = \begin{pmatrix} a_{n-1} & b_{n-1} & 0 & \cdots & \cdots & 0 \\ d_{n-1} & a_{n-2} & b_{n-2} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & d_2 & a_1 & b_1 \\ c_{n-1} & c_{n-2} & \cdots & c_1 & d_1 & a_0 \end{pmatrix} \quad (2)$$

Theorem 1 *Hence, any matrix C'_n is equal to the product*

$$C'_n = LT \quad (3)$$

With:

$$L = \begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 & 1 \\ 1 & \frac{x_{n-1}}{b_{n-1}} & \cdots & & \frac{x_2}{b_2} & \frac{x_1}{b_1} \end{pmatrix} \quad (4)$$

$$T = \begin{pmatrix} x_n & & & & & \\ a_{n-1} & b_{n-1} & 0 & \cdots & \cdots & 0 \\ d_{n-1} & a_{n-2} & b_{n-2} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ 0 & & \ddots & d_2 & a_1 & b_1 \end{pmatrix} \quad (5)$$

With:

$$x_1 = a_0$$

$$x_n = (-1)^{n-1}(\prod_{i=1}^{n-1} b_i)^{-1} det(C_n)$$

$$x_r = (-1)^{n-1}(\prod_{i=n-r+1}^{n-1} b_i)^{-1} C_r \;\; (r = 2, ..., n-1)$$

Proof 1 *By (4) we obtain clearly $x_1 = a_0$ also*

$$det(C_n) = det(L)(\prod_{i=1}^{n-1}(b_i)x_n = (-1)^{n-1}(\prod_{i=1}^{n-1}(b_i)x_n$$

In addition, through L,T direct product, we obtain the following recurrence relationships (or "Gaussian"):

$$x_r + \sum_{k=1}^{r-1} b_k \frac{x_k}{b_k} = b_r \quad (6)$$

for $r = 2, ..., n-1$ we obtain:

$$x_2 = -b_2^{-1} = det \begin{bmatrix} a_1 & b_1 \\ d_1 & a_0 \end{bmatrix} = -b_2^{-1} C_2$$

$$x_3 = (\prod_{i=n-2}^{n-1} b_i)^{-1} C_3$$

And we conclude the formula of x_r

A Decomposition T + K of a Comrade Matrix

Let C_n an $n \times n$ real or complex matrix:

$$C_n = \begin{pmatrix} a_0 & b_1 & c_1 & c_2 & \cdots & c_{n-1} \\ d_1 & a_1 & b_2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & d_{n-2} & a_{n-2} & b_{n-1} \\ 0 & & \cdots & 0 & d_{n-1} & a_{n-1} \end{pmatrix} \quad (7)$$

$$= \begin{pmatrix} a_0 & b_1 & 0 & & \cdots & 0 \\ d_1 & a_1 & b_2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & d_{n-2} & a_{n-2} & b_{n-1} \\ 0 & & \cdots & 0 & d_{n-1} & a_{n-1} \end{pmatrix} + \begin{pmatrix} 0 & 0 & c_1 & \cdots & \cdots & c_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 0 & 0 & 0 \\ 0 & \cdots & & & & 0 \end{pmatrix} \quad (8)$$

We denote that:

$$T = \begin{pmatrix} a_0 & b_1 & 0 & & \cdots & 0 \\ d_1 & a_1 & b_2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & d_{n-2} & a_{n-2} & b_{n-1} \\ 0 & & \cdots & 0 & d_{n-1} & a_{n-1} \end{pmatrix} \quad (9)$$

And

$$K = \begin{pmatrix} 0 & 0 & c_1 & \cdots & \cdots & c_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 0 & 0 & 0 \\ 0 & \cdots & & & & 0 \end{pmatrix} \quad (10)$$

As we show we obtain a tridiagonal matrix plus an $n \times n$ matrix. We applied a CL factorization on the tridiagonal matrix T:

Theorem 1. Such tridiagonal matrix T can be written as the product $T = CL$, where:

$$C = \begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ x_1 & \frac{x_2}{b_1} & \cdots & \cdots & \frac{x_{n-1}}{b_{n-2}} & \frac{x_n}{b_{n-1}} \end{pmatrix} \in \mathbb{M}_n(\mathbb{C})$$

And

$$L = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ a_1 & b_1 & \ddots & & & \vdots \\ d_1 & a_2 & b_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{n-2} & a_{n-1} & b_{n-1} \end{pmatrix} \in \mathbb{M}_n(\mathbb{C})$$

With:

$$x_n = a_n \quad and \quad x_r = (-1)^{n-r} \left(\prod_{k=r}^{n-1} b_k \right)^{-1} \Delta_r$$

$$where \;\; r = 1, 2, 3, ..., n-1$$

And

$$\Delta_r = \det \begin{pmatrix} a_r & b_r & 0 & \cdots & 0 \\ d_r & a_{r+1} & b_{r+1} & \ddots & \vdots \\ 0 & b_{r+1} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & 0 & d_{n-1} & a_n \end{pmatrix}.$$

Proof. Immediatly we obtain $x_n = a_n$, and

$$det(T) = \Delta_1 = (-1)^{n-1} \left(\prod_{k=1}^{n-1} b_k \right) x_1$$

Moreover, by direct multiplication of C and L we get the recurrence formulas:

$$x_{n-1} + \frac{x_n a_{n-1}}{b_{n-1}} = d_{n-1}$$

$$x_r + \frac{x_{r+1} a_r}{b_r} + \frac{x_{r+2} d_r}{br+1} = 0 \; for \; r = 1, 2, ..., n-2$$

We obtain:

$$x_{n-1} = \frac{1}{b_{n-1}} (d_{n-1} b_{n-1} - a_n a_{n-1})$$

$$= -\frac{1}{b_{n-1}} \Delta_{n-1}$$

And

$$x_{n-2} = -\left(\frac{x_{n-1}}{b_{n-2}} a_{n-2} + \frac{x_n}{b_{n-1}} d_{n-2}\right)$$

$$= -\frac{1}{b_{n-2} b_{n-1}} (x_{n-1} a_{n-2} b_{n-1} + a_n d_{n-2} b_{n-2})$$

$$= \frac{1}{b_{n-2} b_{n-1}} (a_{n-2} \Delta_{n-1} + a_n b_{n-2} b_{n-2})$$

$$= \frac{1}{b_{n-2} b_{n-1}} (a_{n-2} \Delta_{n-2})$$

Inductively, we conclude the formula for x_r For this reason, it remains to prove equation for $r = 1$. In fact, we have

$x_1 + \frac{x_2 a_1}{b_1} + \frac{x_3 d_1}{b_2} = 0$

$$x_1 = -\left(\frac{x_2 a_1}{b_1} + \frac{x_3 d_1}{b_2}\right)$$

$$= -\left(\frac{-1}{b_1 b_2} (x_2 a_1 b_2 + x_3 b_1 d_1)\right)$$

$$= -\frac{1}{b_1 b_2} \left((-1)^{n-2} a_1 \left(\prod_{k=2}^{n-1} b_k \right)^{-1} \Delta_2 + (-1)^{n-3} b_1 d_1 \left(\prod_{k=3}^{n-1} b_k \right)^{-1} \Delta_3\right)$$

$$= -(-1)^{n-2} \left(\prod_{k=1}^{n-1} b_k \right)^{-1} (a_1 \Delta_2 - b_1 d_1 \Delta_3)$$

$$= (-1)^{n-1} \left(\prod_{k=1}^{n-1} b_k \right)^{-1} \Delta_1$$

Inverse of the Comrade Matrix C_n: Algorithm 1

Any comrade matrix C_n can be written as:

$$C_n = LT \tag{11}$$

$$L = \begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 & 1 \\ 1 & \frac{x_{n-1}}{b_{n-1}} & \cdots & & \frac{x_2}{b_2} & \frac{x_1}{b_1} \end{pmatrix} \tag{12}$$

A companion matrix, then:

$$L^{-1} = \begin{pmatrix} -\frac{x_{n-1}}{b_{n-1}} & \cdots & \cdots & \cdots & -\frac{x_1}{b_1} & 1 \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{pmatrix} \tag{13}$$

We have also:

$$T^{-1} = \sum_{k=0}^{n-1} a_k J^k \tag{14}$$

With: $a_0 = \frac{1}{b}$, $a_1 = -\frac{a_0 a}{b}$ and $a_k = -\frac{1}{b}(a_{k-1} a + a_{k-2} d)$ for $k \geq 2$.

$$J = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 1 & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \tag{15}$$

Inverse of the Comrade Matrix C_n: Algorithm 2

As the result we can write the comrade matrix C_n on the form:

$$C_n = CL + K \quad (16)$$

Applying the Sherman-Morrison-Woodbury formula we get:

$$C_n^{-1} = (CL)^{-1} - (CL)^{-1}K(CL)^{-1}(1 + K(CL)^{-1})^{-1} \quad (17)$$

$$C_n^{-1} = L^{-1}C^{-1} - L^{-1}C^{-1}KL^{-1}C^{-1}(1 + KL^{-1}C^{-1})^{-1} \quad (18)$$

Where

$$C^{-1} = \begin{pmatrix} -\frac{x_2}{x_1 b} & -\frac{x_3}{x_1 b} & \cdots & -\frac{x_{n-1}}{x_1 b} & -\frac{x_n}{x_1 b} & 1 \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{pmatrix}$$

$$L^{-1} = \Sigma_{k=0}^{n-1} a_k J^k, \; a_0 = \frac{1}{b}, \; a_1 = -\frac{a_0 a}{b} \text{ and}$$

$$a_k = -\frac{1}{b}\left(a_{k-1}a + a_{k-2}d\right) \text{ for } k \geq 2$$

Where

$$J = \begin{pmatrix} 0 & & \cdots & & 0 \\ 1 & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

Inverse of the comrade matrix C_n: Algorithm 3

In this section we consider the comrade matrix as:

$$C_n = T + K$$

With T is a tridiagonal matrix. Applying the Sherman-Morrison-Woodbury formula we get:

$$C_n^{-1} = T^{-1} - T^{-1}KT^{-1}(1 + KT^{-1})^{-1}. \quad (19)$$

We supose that T is non-singular and from :$T^{-1}T = I$ we obtain the column vector for the inverse Ts^{-1} as:

$$Dn - 1 = \frac{1}{a_{n-1}}(E_n - d_n D_n)$$

$$Dj - 1 = \frac{1}{a_{j-1}}(E_n - d_j D_j - b_{j+j} D_{j+1}) \quad j = n-1, ..., 2$$

Consider the sequence of numbers $(A_i)_{0 \leq i \leq n}$ and $(B_i)_{0 \leq i \leq n}$ characterized by a term recurrence relation:

$$A_0 = 1$$

$$d_1 A_0 + a_1 A_1 = 0$$

And

$$b_{i+1}A_{i-1} + d_{i+1}A_i + c_{i+1}A_{i+1} = 0 \quad for \;\; 1 \leq i \leq n-1,$$

Theorem 2 Suppose that $A_n \neq 0$, then Tn is invertible and the column Dn will be

$$D_n = [\frac{-A_0}{A_n}, ..., \frac{-A_0}{A_n}]$$

Numerical Experiments:

In this section, we give a numerical example to illustrate the effectiveness of this algorithms. Our algorithm is tested by MATLAB R2019a.

Consider the following 5-by-5 comrade matrix

$$C = \begin{bmatrix} 7/2i+1 & -5i+1 & 9/4 & 9/4 & 9/4 \\ -9/7+i & 7/2i+1 & -5i+1 & 0 & 0 \\ 0 & -9/7+i & 7/2i+1 & -5i+1 & 0 \\ 0 & 0 & -9/7+i & 7/2i+1 & -5i+1 \\ 0 & 0 & 0 & -9/7+i & 7/2i+1 \end{bmatrix}$$

The inverse of the matrix C is given as:

$$C^{-1} = \begin{bmatrix} 0.0875+0.1827i & -0.0858-0.1628i & 0.2114+0.0733i & -0.1181-0.0834i & 0.2681+0.0200i \\ -0.0028+0.0637i & 0.0789+0.1187i & -0.0621-0.1103i & 0.1151+0.0697i & -0.1303-0.0934i \\ -0.0113+0.0173i & 0.0024+0.0460i & 0.0736+0.1439i & -0.0891-0.1010i & 0.1480+0.1348i \\ -0.0050+0.0032i & -0.0047+0.0123i & 0.0024+0.0460i & 0.0708+0.1179i & -0.1095-0.1542i \\ -0.0026-0.0002i & -0.0050+0.0032i & -0.0113+0.0173i & -0.0056+0.0613i & 0.0767+0.1795i \end{bmatrix}$$

In the table we give a comparison of the running time between this three algorithms and LU method in MATLAB R2020a.

The running time (in seconds) of two algorithms in MATLAB R2020a.

Table 1: The Running Time

Size of the matrix (n)	Algorithm 1	Algorithm 2	Algorithm 3	LU method
100	0.045385	1.281990	0.168303	0.343213
200	0.160370	5.992268	0.024366	0.590798
300	0.305907	22.665200	0.052737	1.211636
500	1.059106	336.329051	0.125830	4.252555

Conclusion

So far with all so much valuable and insightful results we have obtained using the LT decomposition method. In the decomposition, L is a lower triangular matrix with nonzero entries below and on the main diagonal and T is a tridiagonal matrix with nonzero entries on the main diagonal and two adjacent diagonals. This scheme not only makes it easier to represent the original matrix, but also improves computational efficiency and helps with obtaining analytical solutions for some problems. It turns out this structure of the decomposition is especially useful for analyzing and inverting structured matrices such as the Comrade matrices [1-13].

References

1. A Hadj, M Elouafi (2008) A fast numerical algorithm for the inverse of a tridiagonal and pentadiagonal matrix. Appl Math Comput 202: 441-445.
2. B Talibi, A Hadj, D Sarsri (2018) A numerical algorithm for computing the inverse of a Toeplitz pentadi- agonal matrix. Applied Mathematics and Computational Mechanics 17: 83-95.
3. B Talibi, A Hadj, D Sarsri (2021) A numerical algorithm to inversing a Toeplitz heptadiagonal matrix. Palestine Journal of Mathematics 10: 242-250.
4. El Mikkawy MEA (2004) A Fast Algorithm for Evaluating nth Order Tri-Diagonal Determinants. Journal of Computational and Applied Mathematics 166: 581-584.
5. El Mikkawy MEA, Rahmo E (2010) Symbolic Algorithm

for Inverting Cyclic Pentadiagonal Matrices Recursively-Derivation and Implementation. Computers and Mathematics with Applica- tions 59: 1386-1396.
6. Kavcic A, Moura JMF (2000) Matrices with Banded Inverses: Inversion Algorithms and Factorization of Gauss Markov Processes. IEEE Transactions on Information Theory 46: 1495-1509.
7. Wang X B (2009) A New Algorithm with Its Scilab Implementation for Solution of Bordered Tridiagonal Linear Equations. IEEE International Workshop on Open-Source Software for Scientific Computation (OSSC) Guiyang 11-14.
8. Golub G, Van Loan C (1996) Matrix Computations. Third Edition the Johns Hopkins University Press Baltimore and London https://pages.stat.wisc.edu/~bwu62/771/golub1996.pdf .
9. El-Mikkawy MEA, Atlan F (2014) Algorithms for Solving Doubly Bordered Tridiagonal Lin- ear Systems. British Journal of Mathematics and Computer Science 4: 1246-1267.
10. Burden R L, Faires JD (2001) Numerical Analysis. Seventh Edition Books and Cole Publishing Pacific Grove https://www.scirp.org/reference/referencespapers?referenceid=1519495 .
11. Karawia AA (2013) Symbolic Algorithm for Solving Comrade Linear Systems Based on a Modified Stair-Diagonal Approach. Applied Mathematics Letters 26: 913-918.
12. Karawia A A (2012) A New Recursive Algorithm for Inverting a General Comrade Matrix. 1210-4662.
13. Karawia A A, Rizvi QM (2013) On Solving a General Bordered Tridiagonal Linear System. International Journal of Mathematics and Mathematical Sciences. 33: 1160-1163.